Flow and Frustration – an assessment of the impact of software design on the usability of applications within a working environment

STEPHEN CLOUGH

A thesis submitted to
The University of Gloucestershire
in accordance with the requirements of the degree of
Doctor of Philosophy
in the Faculty of Media, Arts and Technology

December 2016

Abstract

Many people spend a lot of their time on computers in work environments, and their usage of these is different from their usage in leisure environments. This study addresses how to make the interactions in this type of task work for the user and thereby enable the user to be more efficient and productive at work.

The literature survey considers the dichotomy of flow – as defined by Csikszentmihalyi – and frustration (defined here in Chapter 2). Through a series of surveys and experiments, this study examines how people think and process information during work-type activities. This is crucial to understanding how the “flow” state can be achieved because it is a central thesis that this is where these human processes are working at their best.

The initial survey uses a questionnaire about the experience of work and leisure by the participants. The aim is to understand and define the difference between these experiences. Following this, interviews are used to draw out some significant aspects of positive and negative experience of computer usage. These features are then verified in a second empirical study using a different data source, which enables a more comprehensive questionnaire to be produced aimed at providing more detailed data for analysis to understand the types of users and the circumstances in they have used computer systems. This provides some insights into the experience that users have, against some demographics, demonstrating that, broadly, the categorisations are appropriate

This study then explores the frustration aspect (and, by contrast, how to avoid frustration) using two experimental studies. The first looks specifically at the design of a user interface and the impact that this has on the usage. The second examines the processing and achievement of tasks with varying levels of “built in” frustration-inducing elements, including impossible tasks. This experiment shows that tasks that are simple to specify but difficult to achieve build up levels of frustration that are carried into further tasks and cause users to be less productive, even for simpler-seeming tasks.

My thesis provides strong evidence for the validity of the flow concept as defined by Csikszentmihalyi and that this can be contrasted with frustration in a predictive dichotomy. It indicates that the concept works within the field of computer system usage, and there is some indication that providing an IT environment that encourages flow will provide a far more productive working environment than one which inhibits flow. The flow-frustration dichotomy is identified as a valid and productive one for computer interaction design.

By understanding the flow-frustration dichotomy designers can aim to ensure interfaces enable user tasks that flow naturally enabling them to be free from frustration and therefore more efficient and productive.

Acknowledgement

I would like to thank my lead supervisor, Dr. Kevin Hapeshi, who has helped and supported me over ten years, and secondary supervisor, Dr. Nina Reeves, as well as all those at the university who have helped along the way.

I would also like to acknowledge Dr. Jean Russel, who helped me with SPSS to assist in the statistical analysis of the data.

I would also like to thank all those who I have interviewed, who have filled out questionnaires or who have just chatted about their experiences with me. This has helped me appreciate that what I am doing has real-world implications.

Declaration

I declare that the work in this thesis was carried out in accordance with the regulations of the University of Gloucestershire and is original except where indicated by specific reference in the text. No part of the thesis has been submitted as part of any other academic award. The thesis has not been presented to any other education institution in the United Kingdom or overseas. Any views expressed in the thesis are those of the author and in no way represent those of the University.

Signed .... Date ....

List of Figures

List of Tables

Contents

Chapter 1 - Introduction

Project background.

For any interactive computer system, a good user experience is an important factor in terms of both productive and content users (see Krug, 2000). Software applications that are difficult to work with, which lead to negative user experiences, will also lead to a waste of effort and often either an abandonment or under-use of the system (Zuboff, 1988 provides some examples of this wasted effort). This study addresses key issues that might lead to a poor user experience, in particular focusing on the features of interface designs that may lead to increased or decreased productivity for the users. The concepts of Flow and Frustration introduced by Csikszentmihalyi (2002) are used here and are defined as the positive and negative engagement of a person in a task-orientated process respectively.

This thesis will examine how these two concepts relate to software usability and how they can be used to indicate good and poor system interaction. It emphasises how task-induced frustration can lead to a feeling of anger and dehumanisation in users during practical use and why this may be considered less of a factor when selection of software for work-related tasks is made.

Motivations.

The reason for this exploration of these topics is, as will be explored more later, there has been much work in the area of customer-facing improvements, and there is continuing work there. However, this has not been reflected back into non-commercial interfaces (what will be seen as work-like interactions). Additionally, the Flow/Frustration dichotomy is one that has been applied to creative areas of work, but not in this area, of user interaction, at least before this study was commenced. Because the Flow concept seems to be one that relates to work-time very well, this study seeks to understand whether there is a use here.

There is also a concern or question for the way that people are expected to interact with software in a work-like environment, where the individuals may have little or no say over the software being used. There is a core concern about the dehumanization that software can include, as well as the demeaning that someone who is capable and competent in their own field, but not computer literate often feels. There is an acceptance that computer systems have often been built assuming a higher degree of computer literacy that is reasonable.

Research Questions.

The core hypothesis explored in this thesis is that "**The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems**".

The contest of this exploration is work-like tasks. We will see that work applications tend to have less advanced interfaces – more basic WIMP. The second hypothesis is that "**Using this flow/frustration dichotomy for interfaces definition will produce financial benefits for organisations**" – that is, the financial benefit will be seen by applying the ideas to traditional WIMP interfaces, not just more advanced approaches.

In this work-like context, it is also hypothesized that "**A frustration experience can have a statistically significant impact on a participant's task-completion experience.**"

The driving hypothesis for this study is that **a flow/frustration dichotomy is a valid way of addressing user experience of computer interaction**. Therefore, this thesis has addressed these two terms and explored plans and processes as the ways in which we manage tasks and processes. This thesis has studied in more detail the concept of **flow**, which has a more formal definition by Csikszentmihalyi, who coined the term. Frustration is less well formally documented but is taken, for the purposes of this thesis, as the opposite to a flow situation (see Figure 1-1), being in either the boredom or anxiety sections of the diagram. This diagram compares the challenge of a task against the skills possessed, both of these from the perspective of the person attempting the task. Where the challenge is significantly greater than the skill possessed, the person will (according to Csikszentmihalyi) become anxious. Where the skill possessed is significantly greater than the challenge presented, the person will become bored. However, in the middle is the channel where “Flow” occurs, where the challenge presented and the skills required match. On this hypothesis, the anxiety and boredom aspects of the diagram represent frustration.

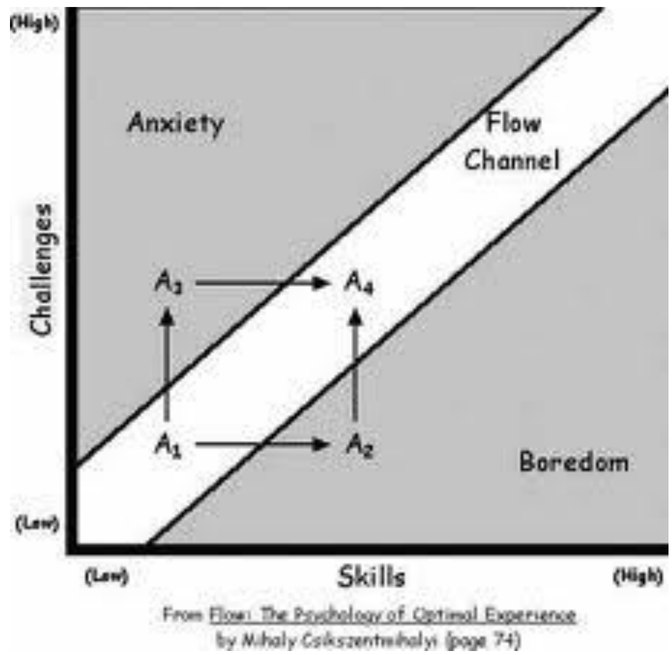


Figure 1-1 - Flow/frustration dichotomy, as pictured by Csikszentmihalyi

Research Objectives

Drawing the connections between the work of Csikszentmihalyi and the understanding of the work-task environment as well as the concern for the emotional responses of the participants makes this work distinctive. While much work is done on software efficiency and customer conversion strategies, this is attempting to take a more rounded or holistic position, looking at whether the same concepts can help to make the working tasks a more positive experience for the user. This research aims to provide guidance on areas that will make interface designs more flow-enabling. This includes measures of tasks that can be tested against an interface.

A second outcome is to validate the work of Csikszentmihalyi against the specific domain of work-like software development. To identify a way that this work can be utilised in the design and development of software enables his work to be further utilised over time.

Finally, the aim of this research is to establish that frustration experiences are not simply inconvenient but do have knock-on effects to further task processing. The implication of this is that poor interface design can lead to inefficiency in terms of a whole work environment – that people who have to work with frustrating software are liable to be inefficient in their whole work day.

Contributions to knowledge

The concepts of flow and frustration are explored in some detail in order to develop a system for recording and measuring these to help properly understand what they mean in the context of work-related tasks. The starting point is an evaluation of the history of interaction development and, to understand how we have arrived where we are, an assessment of the core interaction concepts on inputs and outputs, along with how these have been assessed. In addition, this study compares both the work and non-work (or leisure) context of interactive computer applications, along with an assessment of why a positive work environment is important.

The main contribution to knowledge in this thesis is the use of the Flow hypothesis of Csikszentmihalyi in the specific context of processing in work-type of environments. Additionally, it is exploring what the impact of this hypothesis as a means of assessing user interaction might be on the performance of system users. The opposite side of this is that where users are frustrated in their use of computer systems, they will perform less well.

Research Structure

This research explores a number of these areas to further the available material on these, however it always focuses on the flow and frustration dichotomy. In particular, it will explore the various aspects covered above, to see what each area can contribute to the understanding of what makes for frustrating experiences and what might enhance the flow experience.

Literature review

The literature review (chapter 2) will cover a range of areas providing a background to the remainder of the study. Understanding the **historical background** behind computer interaction design can assist us in understanding why we currently do what we do. This is partly because people do seem to like familiarity: even if the familiar experience is less than ideal, it is sometimes better to retain this than to change (which is the idea of Krug). The mass use of computers is no more than thirty years old, and by non-technical people less than a decade. While this will change, it does mean that for many people, computer usage is a recent and significant change, and the experience that people bring from other areas needs to be considered.

Computer interaction is not a single process, not least because there are two quite distinct and separate aspects – the inputs and the outputs. While they differ, and should be considered on their own, they are also fundamentally connected. When typing in a word processor, the keys pressed reflect in the characters displayed on the screen. These are physically and conceptually separated, but the secondary aspect for this study is that **interaction is about engaging with the user inputs and outputs**. In other words, while it is important to consider these separately, they both need to work together to serve the interactions of the user. An understanding of the input and output processes is critical to appreciate and create suitable user experiences.

The third aspect that is core to this study is that **work and leisure computer usage is different**. Because there are different drivers involved, the considerations for the user experience need to differ. In particular, approaches and ideas that work in a leisure environment may not always apply in a work-like environment. In the context of a flow/frustration dichotomy this is critical because frustrations with a leisure usage has (by the definitions above) less abandonment cost than work frustration. On the other side, enabling flow in a work environment can make a significant difference to total resource cost for the task, thus being significantly more efficient.

Studies

The studies will then take from these ideas and explore whether any empirical evidence can be found for assertions. The first study (chapter 3) is a questionnaire, looking at the **division of work and leisure** and identifying the differences between them. This signifies these categories as valid and important ones to build on.

The next stage (chapter 4) of gathering of data from participants uses an interview process, and these responses are validated using online forum postings. The context of this work is considering **"positive" and "negative" experiences**, rather than "flow" and "frustration", because it is important at this stage to see whether these categorizations match actual experience.

Studies three and four (chapter 5 and 6) use more questionnaires, drawing on the initial categorization as a starting point. Study three draws explicitly from the interview responses and uses a set of propositions drawn from as the core questions. This seeks to confirm these concepts across a far wider population and provide more **definition to the flow and frustration concepts**, in terms of actual experience.

Study four is explicitly looking at **the flow experience** and seeking to ascertain whether the participants do experience Csikszentmihalyi's flow and whether they recognise it. This also looks at whether computer applications are a positive or negative factor in this experience.

Experiments

Then there are two experiments conducted on willing participants. These are both looking at the points where a frustration-like experience takes over from a flow-like experience. The first experiment (chapter 7) is explicitly designed to demonstrate **the importance of application interface design to enhance and enable the flow experience**. By looking at basic visual and processing attributes, the intention is to see whether appropriate changes in the application design can either help or frustrate the user.

The second experiment (chapter 7) is explicitly designed to identify **a point of frustration** in the processing of specific tasks, and ascertain if and where this point occurs. By asking participants to undertake problematic and impossible tasks, the point at which they abandon them can be identified and analyzed, knowing that they must become frustrated at some point.

Chapter 2 - Literature review

The introduction of computers from the very start has presented questions in how to provide appropriate interaction. They primarily are machines requiring some form of man-machine interaction, some degree of communications with people. The Bombe, built by Alan Turing in Bletchley Park and one of the earliest computation machines, used a "menu" – a set of starting instructions, that were wired in – to provide the initial programming for the machine (see Ellsbury, 2012). The "readout" was taken from the disks, and the positions in which they stopped were noted down. Similarly, the Colossus – the first programmable computer – was "programmed" using plugs and switches (see Copeland, 2012). These distinctly mechanical processes reflected the fact that these machines were built for single purposes, albeit with differing inputs as starting points.

There is some continuity with the early commercially available mainframe computers, which were often also started by entering boot programs on switches, or loading operating systems via paper tape. Even when running as more accessible programmable computers, the interaction with them was very often using punched cards, paper tape or some similar form of coding instructions in a way that the computer would understand, not the end user, a mediation requiring operations staff to assist with. The origins of these were from the industrial looms, like the Jacquard loom, which used a concept very similar (see Jones, 2011). While the use of easily disposable cards was a significant improvement over previous interface methods, it still, critically, required intermediaries to prepare requests and to operate the card readers.

However, one significant improvement was that the output of these machines was usually in the form of printed reports, rather than a set of dials or similar – far more accessible to non-technical people. This is one of the important factors in driving the commercial use for Data Processing (see Caminer, 2002, for the early uses of the LEO computers in Lyons, which heralded the major use of computer systems for many years ahead).

The development of the screen and keyboard as an interface to it was important. In essence, a keyboard for entering directly into the computer was little different from a card punch, except that it no longer required intervention from an operator. A monitor as the output was not radically different – it simply provided the feedback that used to be provided to the operators when they processed the same things. But the removal of their intervention meant that less technical users would often see the messages produced. What is more, many non-technical people might be using the system at the same time. In many respects, this was the birth of the HCI discipline. Although Dumas, 2007, indicates – quite correctly – that the **profession** of addressing and resolving HCI issues probably started in the early 1980s, the **issues** that HCI addresses arose much earlier, as soon as non-specialists were using computers. This is reflected in the change in terminology from HCI – focusing on the specific interactions – to UX or User Experience, which is a more holistic phrase covering the entirety of the experience of using the software application. This is why the early history has relevance for this study, because we are focusing on **non-specialists** using computer systems, and for a lot of the history, the only people who actually used computers were technically minded and were prepared to learn how to get the most out of the machines. Additionally, our second hypothesis is about the use of flow for more traditional (not cutting edge) interface work.

The importance of the keyboard as the core interaction technique can be seen in the work of Card, Moran and Newell (1983), who were some of the earliest researchers to look at performance and who used **keystroke-level analysis** as the core of their work. This technique involves identifying the complexity of tasks based on the number of keystrokes to achieve them. They also admit that their model of interaction "is not applicable to problem solving" (Card, Moran and Newell, 1983, p. 430), a limitation that was valid at the time because the majority of the tasks they were looking at and analysing were manual, repetitive data-management tasks, not problem-solving activities. Today, this is more of a problem for three reasons.

Firstly, the use of the keyboard as the primary interaction method is far less than it was. While it is still the main interaction for actual typing, most activities undertaken involve use of the mouse as well, and the "optimal solution" to a task may not be as straightforward as it was. According to Malizia and Bellucci (2012), the current interfaces are "pseudo-natural". The more natural they become, the more individual.

Secondly, much of the use of computer systems today is more problem-solving related than data-entry related, especially within business usage. While the GOMS model may still have value for some tasks, in terms of modelling the computer usage for an individual as they tackle a specific task, this is far too complicated for the model as it stands.

Finally, the approach that Card, Moran and Newell took is only one possible approach. So Bertelsen, 1994, critiques their core approach, because it fails to take into account other possible approaches to cognitive understanding. Today, other ways of seeing and understanding the interaction between the individual and the computer system – largely driven by the universal presence and nature of computers – may be more appropriate. Therefore, this is rejected as a valid approach for this research.

The use of keyboards as the primary input devices meant that it was substantially secretarial staff (who already had typing skills) who used the computer applications, not the management, office staff or engineers. Seeley and Targett (1999) indicate that even then, usage of computer systems among senior executives was "debatable", the anecdotal evidence suggesting that is was not always embraced personally. The standard 80-character, 25-line display, with line editing tools that often required "secret knowledge", was standard for the time, and interacting with this was a skill that users had to learn, often slowly. The Card, Moran and Newell (1983) study demonstrates that the ideal routes or quickest keystroke actions were not always used, even by experienced users. As noted above, the more complex interactions we see with computer systems today are likely to mean that most people use non-optimal approaches to completing a task. We will see later, in Krug (2000), that users tend to *satisfice*, or find a sufficiently good solution to a task. The standard GOMS model is of no use when there is no "best" route to a solution.

Use of tools by human operators

The industrial revolution, when the usage of new tools – machines, often power driven – was introduced, was an era when the machine was initially dominant (see Dempsey, Wogalter and Hancock, 2000). The challenges to interaction then are very similar to the more modern challenges of introducing a computer system (see Zuboff, 1988, p. 8), as the information technology revolution has some similarities to the industrial technology revolution. The reactions of workers in Zuboff's (1988, pp. 140-150) study into the introduction of technology at Perry Wood (a wood pulping plant) included comments from workers made after the introduction of computerised technology that they felt like "prisoners, blinkered, mute and impersonal", because the social interaction of the work had gone, and this was very dehumanising. This approach by the management of Perry Wood is an indication that the frustration experience is often ignored when introducing technology – the third hypothesis indicates that this is dangerous in terms of achieving task completion.

These negative approaches have echoes of Marx (1946, p. 460): "In the form of machinery, the instrument of labour immediately enters into competition with the worker." He was talking of the industrial revolution, where the new automated tools had supplanted the workers, whose responses were sometimes aggressive and violent (Marx, 1946, p. 458). For the Perry Wood workers, without resorting to such violence, there was definitely dissatisfaction identified by Zuboff (1988, p. 301). While acknowledging the differences, the reactions in the early stages of the new technological revolution can cast some light on the reactions of the first industrial revolution. In terms of modern-day acceptance of technology, Zuboff is dated, but for an understanding of the early implications, and the impact today of radical changes in working environments, it is invaluable because these studies cannot be replicated today, due to the substantial computer literacy in the population.

The age of Zuboffs study is – as well as being a critical insight into an earlier time – the biggest challenge to contemporary use. The problems encountered in Perry Wood would not be encountered in a modern environment, because of two crucial reasons. Firstly, the technology has progressed to a point where this type of initial automation would not happen. Any such place would already have some degree of computer usage, so the progression to a more automated environment would be less dramatic (and, undoubtedly, less unexpected). Secondly, the computer literacy of most people today is so much higher that this type of change would be less threatening. Our world has embraced computer systems, so the revolution that occurred in Perry Wood was far more challenging than any equivalent computerisation today.

None of this is to dismiss Zuboff (as Burton-Jones, 2014 points out, her study still repays further investigation). It can help us to see the challenges that some people still face with technology change, even if they may not be so open about it. The "emotional and mental demands on workers in an infomated environment" (Burton-Jones, 2014, p. 36) are still significant and important, even if the nature of these demands is different. Most critically, it shows the Frustration aspect of our primary hypothesis in practice.

Graphical interfaces.

At the end of 1980s, many software applications started to adopt a graphical interface approach to integrate with the operating systems they were starting to work on. The leading word processor of the time – WordPerfect – made a significant move into a windows-hosted, graphical approach. The move of this major business tool – one of the most significant uses of personal computers in the office at the time – to a new paradigm for interaction marked a significant development in the role of User Experience as a profession, but it also highlighted the change in the world of interaction. In the DOS and mainframe paradigm, WordPerfect was the world leader (see Pachal, 2012), a lead that plummeted after Microsoft started to dominate the software market in commercial environments. It was not necessarily the most important change, but it was an indicator of a sea-change in the way users interacted with software applications in work environments.

The drive for this move to graphical approaches was clear in the last DOS versions of WordPerfect – the need to learn techniques and tricks to perform various common actions. As with so many developments in software, it was attempting to deal with the frustration of requiring specialist application knowledge to use. Understanding command line interfaces, or even complicated GUI interfaces, can be frustrating to users whose skills lie elsewhere. In simple terms, if a typing pool could intuitively learn how to use a word processor without specialist training, that would be a major sales point. In terms of our second hypothesis the financial benefit to a business is not in terms of novel interactions, it is in terms of improving existing interactions.

The introduction of GUIs like Windows, Lisa and GEM meant that a new approach to user interaction was needed. The concept of a WIMP environment covered both GEM and Windows (Lisa preceded the acceptance of this style) and became the de facto standard in the years to come (see van Dam, 1997). The WIMP paradigm was –and still is – significant to user interaction. WIMP stands for Windows, Icons, Menus and Pointers.

Windows. The core development that made these graphical environments productive was the introduction of windows, or separate processes running in parallel. Even van Dam (1997), who is looking towards a post-WIMP environment, accepts the continued use of some form of windowing system. In essence, a windows system applies to any form of parallel processing of applications shown on the desktop. The issue for interaction is that with multiple processes all being run for a single user, how can we appropriately provide relevant interaction – in particular, providing appropriate messages back when they may be working on another process.

Where users are entering into flow – which is a single-process state – having a multi-process environment can be distracting. However, by allowing multiple activities at once, this should enable a user to enhance their flow experience by following their own paths, if these involve multiple applications. It might also be a positive for flow processing if the user knows that any important interruptions can be received and dealt with, they are not preventing appropriate breaks in their flow. It may be that what van Dam calls a “butler-style” interface is more appropriate, where there is a seemingly intelligent interface, which can leave you alone when you need it, but will interrupt you for something urgent.

Of course, multiple processes are nothing new. The early mainframes all ran with multiple processes concurrently (or apparently concurrently – the reality of process swapping is not pertinent here, but see, for example, Krishnamurthi and Young, 1984). However, for most users there was only a single interactive process for them to work against. The windows paradigm involves multiple processes for the same user, running concurrently, and all potentially requiring the involvement of the same user.

Icons. These are small pictures that are used to represent processes (programs) or actions (tools). They have become more advanced with time, and as the graphical abilities of monitors and processors have improved, but they are still a critical part of much interfacing – although how they are used and interacted with may change. If we consider icons as "visual clues", then they are a critical part of interaction and will remain so, despite van Dams (1974) beliefs that they will eventually be unnecessary and therefore disappear from use. Users, it would seem, continue to need visual clues as to what to do, and icons – in some form – are therefore liable to stay, although they are changing in form. Because people are good at pattern recognition, seeing an image and interpreting what it means can be very productive. Producing appropriate iconography will continue to be important in the design of user interfaces, even if they will extend beyond visual to a broader instant sensory indicator – as van Dam (1974) suggested.

It is also relevant to look at the iconography we use, because this tells us a lot about how people interact with systems. As Bedi and Hrustek (2010) argue, the use of the floppy disk as the save icon is very anachronistic, but it is universally used. It works, not because people know what the icon shows but because people recognize what it represents. People who have only started to use computers in the last 10 years or so may never have used a real floppy disk, and so the picture reflects an action, not a physical item. According to Bedi and Hrustek (2010), the icon still represents the process to people using it.

The issue that will present itself is with more significant post-WIMP interfaces, where icons to represent processes might be replaced by gestures (see Angelini, lalanne et. al. 2015), especially on small-form devices like phones. For individuals in a flow state, all that is needed is some means for them to make a process happen, and they are less liable to care exactly how this is indicated. If we consider an office environment, having people gesticulating to action everything may not be appropriate. But for people to be in a flow state, they need some intuitive means of performing the processes they require. It would seem that there is a lot more to be done on this to replace the icon paradigm.

Menus. This refers to the drop-down controls, providing lists of options and tools that the particular application can perform. This has meant that applications have been able to provide significantly more features and options than previously, while making them all readily available to the user. However, the problem comes when substantial numbers of options or choices are required.

One of the responses to this that has been tried (in Microsoft Office 2000) is **adaptive** menus. These are menu lists that change based on what the user is doing and what they have chosen before. Microsoft used this approach in some of its software applications. However, as Mitchell and Shneiderman (1989) had previously found, users do not like changing menu systems. They get confused and take longer to perform tasks when menus are not in static positions. What was liked, post the Microsoft Office 2000 approach, were **adaptable** menus (Findlater and McGrenere, 2004), where the user can adapt menus to match their own specific requirements. Research is continuing into optimal menu design (see Bailly and Oulasvirta, 2014), in particular to identify the way to define better adaptable menu systems.

Van Dam (1974) may have correctly predicted that menu-driven systems are not going to stay. He suggests that more gestural interaction would supersede the menu approach, however there is currently no approach that can provide the full functionality of a menu system (whether words or graphical). While menus will change, maybe become more adaptive, there is no evidence that they will disappear, because users get frustrated when they are unable to find functionality (which is why the adaptive menus failed).

Pointer. The pointer was one of the most radical developments of this time. While this substantially refers to a mouse, it also covers any input device that can be used position a curser anywhere in the monitor window, like trackballs and joysticks, and can also be used to simulate a "click" process. It was the first new interaction device since the monitor was introduced, and it allowed a completely new approach to interaction. Users were no longer limited to the keyboard with its limited ability to indicate movement around the screen. The mouse allowed the user to see a location on the screen, move there, and interact.

Once again, van Dam (1974) predicted that a mouse controller will start to disappear for this type of interaction. Smartphones – one of the biggest growth areas in computer interaction – are not controlled by a mouse but through touch-sensitive screens. However, this does not mean that the pointer concept will disappear. Even if, as van Dam implied, interaction becomes more 3D, some form of pointing process is liable to be involved, even if it is simply 3D finger tracking. In fact, the ability to point and click an interface is still an important approach, and one that will probably persist, however it is achieved, not least because it does appear to be a very usable concept. Seeing something and then interacting with it seems to be a natural form of engagement, which is why it is still used.

This WIMP interface has been explored in depth because it is still the basis of the majority of our interaction and seems liable to be for the foreseeable future. In terms of the second hypothesis, it is still – in a reasonably basic form – the standard means of interacting for business users. The paradigm does not appear to be particularly frustrating inherently and so should not impede a flow experience, but this may be because of familiarity (which is an important aspect of a flow experience). At the same time, there are important developments (see Argyros and Lourakis, 2006) because the technology is becoming more advanced, and the needs of specific groups are becoming more diverse. Other post-WIMP technologies will become significant, but for now, especially as this study is focusing on work – and so often office – processes, the WIMP paradigm will be assumed. Because there is no clear post-WIMP approach, we will stick with an assumption that the WIMP style will be used. It is beyond the scope of this study to assess other interaction styles at this time, but interface methods that engage with users more naturally – more seamlessly – may improve the flow experienced by users in their usage of computer systems.

What this paradigm started to produce was what Schneiderman (2002, p. 11) calls "made-for-users" applications – applications that were there to assist people in performing their work. This is distinct from "made-for-purpose", which are applications that force people to use them rather than other tools (examples might be the software drivers for monitors). This is a continuing development, the aim being for applications to support your work, rather than define your work, according to Schneiderman (2002, p. 13). However, he also suggests (op. cit., p. 24, 25) that up to 10% of working time may be spent on computer problems, which would suggest that he is somewhat optimistic. Time spent resolving computer problems is time where the applications in use are not made-for-users. If a software application *demands* your involvement, then it is not made-for-users.

The Internet

A more recent stage in this development has been the growth of the internet and the changes and developments this has forced on user interaction work. Neilson (2006) points out that people use the internet as a whole, not as a collection of separate sites, which is how it was used in the early days. The significant change is in software applications that are web-based or web-fronted, rather than desktop applications.

The accessibility and use of the internet can be a positive factor. There is now much more help and assistance available for problems, especially for common software. However, there is a possibility of frustration, especially with proprietary software, if no help can be found – this can be exacerbated because the user-base is no longer a few friends but all users across the world.

In terms of user interaction, the most important aspect of this is the separation of the delivery mechanism – the display of the interface in a browser – from the processing – on a server at an unknown or irrelevant location. This opens up the possibility also of **asynchronous** processing, where the server and display are both able to process at the same time, providing a more responsive interaction. So as Kludge, Kargl and Weber (2007) show, using AJAX processing (**A**synchronous **J**avascript **A**nd **X**ML, a set of technologies that enable websites to process asynchronously and appear more responsive) the page speed and smoothness can be significantly improved, providing RIA (Rich Internet Applications – see Wahib and Khoon, 2015). This can help with user's immersion in an interaction, or "flow" in performing their task (Hudson 2007). Properly designed AJAX interactions do improve usability (Holzinger, Mayr, et. al. 2010), however, there are downsides identified in all of the previously cited research. Some of the most serious include a lack of standards and potential hang-ups.

There is one other relevant aspect of the "Web 2.0" paradigm, which is the importance of User Generated Content (or UGC). As Oh, Susarla and Tan (2008) make clear, user content is closely related to being "hyper-networked" because people read or watch what others generate. The rise of social media is an important aspect of the web – maybe the most important when considering social trends (Asur and Huberman, 2010).

The combination of the display process being much more detached from the processing and the expectations of users to be involved in their applications has meant that the concepts and methodologies of the User Experience discipline are starting to come into their own. Dix and Cowen (2007) discuss "Social Experience as Product", which they use to mean how networking with other users can be a valuable reason for an application, and "Democratising of Media", which is the importance of the individual's experience, along with the ability to share this experience, as a reflection of what is actually going on in the world. However, the hyperreality that Baudrillard (1995) discusses, where the only reality is the simulacrum or image presented, would seem to define the social media paradigm perfectly (see Jerina, 2010 "People are increasingly separated from their own experience" and "For social media the projection of the self on Facebook and Twitter *precedes* the object of the self – we are left only with a simulacrum of the self, a self which refers to nothing") and demonstrates the problem that we never actually know reality because we always see it reflected through others experiences. It is significant that some more recent work by Brooker (2011-12) and Curtis (2016) reflects the hyperreality paradigm as how an increasing number of people experience the world, while identifying some of the risks and dangers of this.

How does this changing style and approach to interfaces impact on the flow and frustration thesis being explored here? The importance of Baudrillard is that **perception** is becoming seen as **reality**. This change means that as long as the perception of a product or interface is working for a particular person, then they will perceive it as the reality, however close this is to actuality. There is potential here for frustration when the actuality that others see doesn't match your own (for example when an email gets lost, the sender knows that they have sent it, but the recipient hasn't received it) but also for flow because the reality underlying the interaction can be out of step but without interrupting the user.

Primarily, it means that **user perception** of a task processing is far more important than more objective measurements. This in turn means that more automated testing and evaluation becomes more difficult, and actual user engagement with applications is critical. A new version of an application may be objectively quicker, but if the user does not perceive this improvement, they will be unimpressed. Of course, the other side to this is that perceived improvements are far more important in gaining acceptability. The hyperreality paradigm argues that nothing is real (see Stolze, 2016). From a software perspective, the increased engagement and power in the hands of users does make objective reality less significant. The "truth" is whatever is presented to a particular user. I do not feel we are yet ready to embrace Žižeks "reality of the virtual", that there is no actual reality behind it. There is something real – the Baudrillardian approach simply argues that people only see that reality though filters of various sorts.

Plans, processes and flow.

To understand what Csikszentmihalyi (2006) refers to as flow, we need to understand the concepts of **plans** and **processes**. These are short term stories, with definitive (even if not always perfectly clear) aims. The dynamics of a process flow are often a key aspect of computer applications. Most applications are intended to be enabling or supporting of some form of task-orientated plan or process, which can be a whole range of tasks. The typing of a document on a computer is a sequence of plans or a range of processes. The playing of a game is a process – to achieve success in the game certain tasks need to be completed. The calculation of entries for a report is a process, involving a range of different actions. The checking of your email is a process, a task leading to a goal that may start off other processes. They all involve a set of steps leading to a goal, with some of the steps at least involving interaction with a computer system.

The essence of the argument presented by Csikszentmihalyi (1997, 2002) is that the experience of using software will relate to how well it supports the plans and processes we perform. He argued that peoples most useful and productive time is when processes flow and we achieve our tasks smoothly. Suchman (1991, p. 3) is useful in understanding what the activity of a process is and how it can be made to flow better. She explains that there are two approaches to achieving an aim: either making a plan that directs everything that is done towards the final goal or taking decisions at various points in a process based on what appears to be the right thing at the time – what she calls “situated actions” (op. cit.). Of course, the first of these assumes that you know how to get to the end of the process at the start.

Krug (2000) discusses the importance of making actions very simple for users, and cites the example of the Amazon.co.uk *1-click* ordering which enables customers who have already registered their details to buy an item by simply clicking one button – all of the checkout process is then completed and the product delivered. This *1-click* process has been extremely successful for them, according to Krug, because the process of item choosing and selection has, by that point, been completed, and people who are focussed on the goal want a simple one step towards completion – all they want is to get to the goal of having the items ordered. This works (to a large extent) because users can only use this after they have already checked out and know what the process is. So, according to Krug, customers are confident in the process, which makes it work.

Of course, Krug’s perspective is very much what is good for the business behind the application – usually a website. His aim is providing a positive user experience so that the user buys more and returns to buy again. This does not discount his work or ideas, but it can mitigate against the situated actions. The title of Krug’s work, “Don’t Make Me Think”, underlies his idea that making users think is an indication of bad design. However, for a user to take situated actions – actions at any point that are based on the best option – they need to think, they need to have the range of options and choices provided to them. If we have no choices, no involvement, then we fall into the “boredom” section of the Csikszentmihalyi graph shown in Figure 1-1.

Krug’s observations and recommendations have been influential in the development of web site design. His core principle is that everything should be self-evident – that is, if it is a link to jobs, it should say “Jobs” and be something that is *clearly clickable* (Krug, 2000, p. 14-15). The term “clearly clickable” is not easily defined; it may change over time and may be different between different user groups, but the user of the system should be able to identify an important hypertext link as something they can click. As he points out, on the computer screen users do not read web pages in the order we build them; they scan in a fairly random way (pp. 21-23) (while this is specifically applied to web pages, the same principles will apply to desktop applications too). In a general sense, Shreshtha and Lenz (2007) and Neilson (2006) show that for most web pages, the reading pattern is an F-shape. This means that the users are scanning the pages across the top, down the left and across the middle, not reading them in depth and not necessarily reading or seeing them in the same way that the designer intended. Users of computer applications do not always see things as obviously as the developers see them – it may be obvious to a developer who used a lot of Microsoft tools that “Print” is on the “File” menu, and how to get to it, but to a user who does not use as many of these Microsoft tools, this may be obtuse.

The problem with Krug’s view in the wider context of software application design is that his focus is (deliberately and correctly) on consumer web sites and so based on the requirements of the business behind these sites. The principles Krug defines as important to him – not having to think; scan reading; “satisficing” – are all valid for any software, but what they mean may depend which side of the application you are on – are you the business wanting more sales, or the user wanting a positive experience? This third idea of *satisficing* is actually critical to software usability, that users of software do not look for optimal routes to achieve their goals – that is, in the Card, Moran and Newel (1983) terminology, use the best selection rules to find the optimal method. Rather they find a solution that does the job for them (see Krug, 2000, pp. 24-27). The solution people find for a task is one that *satisfies* them and *suffices* to achieve the goal.

In terms of the theses we are considering, this has significance because users will find their own flow route through an application. This need for satisfaction may mean that users' frustration at not being able to achieve their task is then continued – they remain frustrated by a task that has passed them by.

Krug's *satisficing* (p. 26-27) is interesting in that users "muddle through" (according to Krug). This is a critical understanding because users do find ways of achieving solutions irrespective of how developers and designers expect them to. Software development organisations often find users who need a new solution because they are currently making do with something in ways that the original developers would not have contemplated. When a problem has occurred, they have found a solution that does what they need, however obscure. To encourage users towards an optimal method, they need to be able to find it without "thinking" (meaning, in this context, conscious cognitive processing). If solutions are not glaringly obvious, users will *satisfice*. Within a commercial web environment, this may mean that users go somewhere else, however, within a constrained work environment, it may just be that users have to think, break their flow, and perform their tasks less efficiently and less enjoyably. It may be that the growth of the internet since Krug wrote his ideas has changed this significantly, as it is much easier today to find the good way to achieve a solution, at least for commercial software products. It may be that today, we muddle through less than we did: at the same time, we may find ourselves less satisfied when we cannot achieve solutions.

The Amazon 1-click feature, where a single user click processes the whole of the ordering, is a good example of how applications have improved over technically orientated, command-line-type interfaces to an apparently "easier" point and click environment. Clicking an image or text link makes the processes happen, whatever they are. In most cases, it is irrelevant to the user what goes on behind the scenes. For them it doesn't matter how the process is being done, what is important is that the goal of the user at this point is achieved by clicking this link. It should be clear from the link what it will do for the user, whether it is a button or a clickable link or any other visual indicator. Importantly, it should then go and do it. The user only really needs to know that it is happening and when it is complete. This is Krug's core point and why the 1-click button is important for commercial web pages, according to him.

However, most tasks undertaken (with or without computers) are not so amenable to single-click processing – they require a series of processes or actions to achieve a goal. For example, a purchase may require approval, or a report may require information to be obtained from other sources. Some actions may differ depending on the precise circumstances or situation – *situated actions* are the actions taken in a situation which lead to some sort of goal (however loosely defined). The actions in situations are not necessarily based on the goal, but people "*fall back on whatever embodied skills are available to you*" (Suchman, 1991, p. 53). However, to make the sensible decisions in a situation, it is important to know 1) what the options are, 2) what the state is and 3) where we are in reaching for the goal. "The user expectation ... is that each system response conveys ... an assessment of the last action ... and a recommendation for what to do next" (Suchman, 1991, p. 154-5). The context of where we are is important, and Chapter 7 of Suchman's book shows some of the problems with having only goals, not situated responses. An analogy for this is driving. When people get in a car, they have an aim or goal – whether this is a place to get to or just to have a pleasant day's driving. This aim is important to keep in mind, but at every junction, a decision needs to be made about where to go – this is situated action. In some cases, decisions need to be made about how to negotiate a junction or what to do if a road is closed or busy. For Suchman (1991, p. 34), the important principle is that actions can be accounted for as part of a plan *post hoc* but not from the intent of the actor. So Suchman's theory is that we cannot know what someone is going to do based on what they intend but only on what they have done – decisions in a situation cannot be predicted, but they can be justified or understood when the aim is reached.

In terms of computer applications, this would mean that the set of options presented to a user needs to be clear, appropriate and relevant so that they can make the appropriate decisions as they progress – what Norman (1988, see pp. 85-

90) calls *affordances*. Affordances (or perceived affordances technically, because they are just graphics) are the actions that an object seems to have. For example, a handle on a door suggests that you should pull it to open the door. The Amazon 1-click process only works if the set of decisions has already been made, then the affordance of the button, which is clearly marked, does exactly what it says. Normally the set of processes or actions needs to be clear, including all of the information needed at that point to make the appropriate decision.

Within the context of work environments, it would seem that Suchman, with her Situated Actions (that is, providing all of the information needed to clearly make an informed decision), is a more appropriate model than Krug's hiding of the processes. For users to be a valuable part of the process, they need to have an understanding at one level of what they are doing – of what, from a business process perspective, their actions mean. They don't need to know the technical detail (save in a database, email someone) but at a level appropriate to their function (request authorisation).

As Kersten and Gail explore, the idea of "situated action" has developed into a wider context of "context-aware" software. That is, software that knows what your current situation is (as much as is relevant) and so can present you with appropriate actions at that point. As the UbiComp concept (ubiquitous computing – the idea of computers being in everything and everywhere and less noticeable) has developed, the need for software to understand the context better has become important. This is in support of the Suchman model, rather than the Krug model, by giving users the right set of options, not assuming the software knows what is needed.

There are problems with the Suchman model – as Nardi (1996, p. 45) points out, her use of "situation" as the driving focus is not necessarily always the right approach. It focusses on one core action, not the wider process, which Nardi explored in Activity theory (which we will cover more below). From a working environment, the context – including a chronological context – can be critical to understanding how an individual works and processes. Suchmans concept is important in terms of making the situated actions work and function as smoothly and cleanly as possible, but this is not the whole picture.

Flow

According to Csikszentmihalyi (2002, p. 143ff) **flow** is the positive engagement of a person in a task-orientated process. He introduces the concept that interaction is a psychological process in which there is a changing dynamic as the individual engages and responds and as the application responds (or not). Csikszentmihalyi also introduces and discusses the concept of "happiness", which is an important one for Csikszentmihalyi because it includes states where task achievement is productive (Csikszentmihalyi, 2002, p. 154): "If workers really enjoyed their jobs ... they would almost certainly produce more efficiently." For Csikszentmihalyi, this emphasis is important because much of his work focuses on how people can be happy themselves and get more out of life by functioning more efficiently. This is Csikszentmihalyi's core concept, however, he offers no direct evidence for it, and while this is a good aim in itself, in commercial terms, making people happy is not a primary function of the working environment. It is also not clear that efficient processing of work tasks actually makes people happy in general terms. While they may (as he argues) find an individual *task* more fulfilling, he does not explore how this relates to a more general happiness in life.

How does this relate to interface design? Where does worker contentment link to interface design and flow? In simple terms, Csikszentmihalyi's ideas suggest that the software that is straightforward to use not only makes the worker feel better but also makes them feel valued. The question here is whether the human aspect is one of the drivers behind designing interfaces. This thesis is seeking to identify whether the flow/frustration dichotomy can be a way of understanding the design of interfaces that makes them more amenable to the flow experience and whether this would lead to worker contentment.

Ostroff (1992) does start by casting doubt on this idea: "The bulk of evidence shows the correlation between [job] satisfaction and performance [is] relatively low." Does this invalidate Csikszentmihalyi's argument? Ostroff explores what the parameters of these studies are. In particular, Csikszentmihalyi argues slightly differently – not that satisfied employees are more efficient, but that a flow state tends to make individuals both more satisfied and efficient in fulfilling that task. Ostroff does find that, at an organisational level, satisfied employees "[tend] to be more effective than" less satisfied employees. More recent studies (for example, Yee, Gui and Yeung, 2015) continue to support Ostroff, contra Csikszentmihalyi, that there is a direct relationship between work enjoyment (or happiness) and productivity. Rather, this study will explore the negative approach, whether frustration leads to less productivity. If so, it would suggest that flow would lead to greater productivity. While less frustrated workers may be happier, this will not explicitly be tested, because the connection would appear to be far more complex.

Csikszentmihalyi (1975, p. 49, 120) identifies the flow state as occurring where the challenges of a task are well matched to the capabilities of the person engaged in it. This implies that, in a good working environment, this is possible – where employees are being given tasks that are suited to their abilities. If the match is not right, it will lead to either boredom – where the tasks are too simple – or anxiety – where they are too complex. However, this is not the full story. There is more to a person maintaining a flow state or achieving it for a significant portion of their day. He also distinguishes between *autotelic* and *exotelic* activities, that is, ones where the reward is from a nicely completed task or where the rewards are external. He also makes it clear that different people reach a flow state in different ways from the same activity (1975, p. 62). This means that a system needs to be flexible to a particular user's needs in order to enable flow for a range of users.

The argument by Csikszentmihalyi is that identifying the tasks and the set of processes that a particular individual would find most conducive to a flow situation is not straightforward. And, within a commercial environment, there are tasks that need to be achieved whatever. This will be addressed in more detail later, but where an individual is in control of their own tasks, then they are more likely to experience the flow state. But it is also true that proper tool design – especially the software in use, which is often the primary tool – can make a big difference. "Extremely procedural" processes help to block other thoughts and so enhance flow. It is all about the now, where each action naturally leads to the next (Csikszentmihalyi, 1975, 85). This should therefore be a design aim for computer software interfaces, so as to enable the flow process. Krug (2000), states that "a typical element of flow activities is the presence of direct, unambiguous, feedback" – the context here is not computer systems, but this is a positive aim for their design. As we saw earlier, context-specific applications should allow for straightforward activities at each stage in a process and so should be more able to enable a flow experience.

This principle implies that some of the features and constraints of interfaces that make flow less possible cause problems and irritation. It is about providing more quantification to the "direct, unambiguous feedback" and the "each action naturally leads to the next" concepts. If the software interface is very "flow aware", and enables rather than interrupts flow, then the tasks being achieved should be more enjoyable and fulfilling, whatever the environment. A flow-enabling application is one that does not frustrate the user, does not provide either too much or too little information at any point.

Csikszentmihalyi (1975, p. 165) also did some interesting work in depriving people of flow for a period of time. The subjects reported that they were "More tired, more sleepy [and had] more headaches", that they felt "less relaxed and less healthy" after 48 hours of flow deprivation. The increased use of computers today, and the substantial time that many people spend on them, mean that the interfaces need to be much improved, otherwise people engaged in non-flow work activities and non-flow leisure computer use are liable to be significantly negatively affected. Flow deprivation tends to make people more irritable and so less tolerant of – for example – poorly designed software interfaces. They also report being less creative or innovative, and Csikszentmihalyi identifies that for this type of work or task, where leaps of innovations are significant, flow is vitally important to the "creative juices" (Csikszentmihalyi, 1996, see p. 8, but this is the essence of the whole book). The implications are that in a flow-deprived state, people will be less creative, less innovative, and, in areas where this is important, less productive. Finally, Haidt (2006, p. 221) links these ideas of fulfilment and productivity with the understanding that Marx brought to working environments.

Frustration

The alternative to a feeling of flow is the frustration that users can feel with computer systems when their goal is blocked or difficult to achieve. Picard (1999) identifies that this is a key issue in human-computer interaction: “Not only do many people **feel** frustration with technology, but they **show** their frustration”, the evidence for which she explores in this paper. There exists a small body of research into user frustration which indicates that users do tend to get frustrated when their task achievement process is blocked, especially when this is caused by technology failures. However, the scope tends to be either based on systems that the user is comfortable with (see Ceaparu, Lazar, Bessiere, 2004) or focused around more physical interfaces (see Reynolds, 2001). Both of these are important and significant aspects of interactions, but this research aims to look at the software features and task-based processes, addressing more how software can be improved to make the working life of users better. This is a different approach to Ceaparu et. al. (2004), who "felt that it was more powerful to let users perform tasks that were relevant and important to the users themselves, rather than pre-assigned tasks chosen by the researchers."

The reasons for frustration identified by Ceaparu et. al. (2004) are important, in that they find that error messages, dropped connections, freezes, long downloads, and features are key issues. These match, in broad terms, many of the findings of this research too. These issues fall into two broad categories: those under the control of the application developer and those not under their control.

Two of these issues are under the control of the software developer: misleading error messages and hard to find features. These two can be improved by developers paying attention to heuristic guidelines such as those provided by Nielsen (2001) and Schneiderman (1982) for error messages:

> **Visible and highly noticeable**, both in terms of the message itself and how it indicates which dialogue element users must repair.
>
> **Preserve** as much as the user's work as possible. Let users correct errors by editing their original action instead of having to do everything over again.
>
> **Reduce the work** of correcting the error. If possible, guess the correct action and let users pick it from a small list of fixes.
>
> (Nielsen, 2001)

These are a starting point rather than an end, but Nielsen raises the point that the communication is for human beings to read and understand, and that they have to be able to act on it. These indicate that there are rules that can start to address some of the issues raised by Ceaparu et. al. (2002) if software developers decide to follow them. Schneiderman's (1982) first principle seems to cover a lot of the area "Increase attention to Message Design". Although when originally written it was more about the wording, today it covers the positioning, colouring, font and connection to the error location, but the principle still applies.

There is a problem with this in the context of supporting and enabling flow. The idea that errors should be visible and highly noticeable works well if the problem is with what the user is actually doing at the time. However, prominent messages from other concurrent tasks can be disruptive to a flow experience. Additionally, prominent messages that prevent you from completing a sub-task can be frustrating – it may sometimes be better to alert but permit the user to continue and fix problems before proceeding. Providing appropriate feedback at the level required by the user (not the application) is still a major challenge in software development.

The problem of features being hard to find, highlighted above as an issue, is another major challenge but one that can be achieved by the right application design but requires careful design. Some applications do this by having very few features – they are single function applications. Phone apps are often examples of this: simple, straightforward, and doing one thing. As Nielsen and Loranger (2006, p. 80) point out, users are less interested in features, more in benefits. As they argue, if they can see that a product will benefit them – solve their problems or help their work – they are more liable to be interested than knowing that the product has some features that they might use occasionally.

Those reasons for frustration not under the control of the developer (as detailed above) are often ignored, but they are still aspects of frustration to a user – the fact that they may not be under the control of the application developer is not particularly pertinent to the end user. Tools that are core to the web 2.0 development, like AJAX, make some of these much easier to handle (see O'Reilly, 2007), and desktop applications need to handle interactions in a similar way. Why should the user need to know what is hosting their application? As O'Reilly points out, the focus of the Web 2.0 world – which includes all application interaction – is of core user focus, concentrating on the users' interactions, and handling everything else in the background.

Information and communication

As highlighted in the previous section, history shows us that the development of computer interaction has been a core part of the progress of computer systems. The three aspects of computer systems – the inputs, the processing and the outputs – have developed as separate but connected streams. Interaction covers two of these: the provision of input and the production of outputs from a computer system. These are the places where people interact with computer systems, so they need to be as smooth as possible so as not to interrupt the flow state. A proper and deep understanding of how we interact helps to appreciate why it is so complex and personal. A flow experience can occur when the interaction is *sufficiently good* not to interrupt the processing, not necessarily perfect or even as good as it could be.

The need for interaction to be sufficiently good, not perfect, is why the Card, Moran and Newell (1983) studies are rejected because they worked on a perfect route. As they discovered, people don't need a perfect solution but one that is reasonably good because (as we will see) the human brain is good at dealing with patterns.

While a lot is often made of the processing power and speed of computer systems, this is meaningless without some form of interaction with users. People make up an important part of the wider "system" of which the computer program is a part. Processing power is important to do the data manipulation that the computer is best at, as well as to make the interactions work more smoothly. Processing power is required to enable interaction designers to provide the best interaction possible.

Within this study, there is an assumption that user control is through basic input and output devices for GUI interfaces: that is a keyboard, mouse and screen. However, there is an acknowledgement that the choices for input and output have been evolving, especially over the last decade. The principles of interaction should apply as much whatever the devices used are, and the historical development of these still applies. Within a work environment it is likely that keyboard, mouse (or other pointer device) and screen (or screens) are likely to be a significant focus of interaction for some time to come.

One factor in interaction development is the learning time for new forms of input, as West (1998) discusses with reference to failed attempts to change the standard keyboard input with a Dvorak keyboard. The Dvorak demonstrates improved speeds (maybe not as large as some proponents claim, but still improved – see Rogers, 2003, p. 9); the time for adoption (to retrain existing users to using the new keyboard and bring them up to speed) does not justify the benefits. Changes, especially in the working environment, need to be done without loss of performance, and so without a significant learning period. Dvorak may, in time, improve the flow experience, but many people who use computers daily have learned the standard Qwerty keyboard, and so it is *sufficiently good* for a flow experience.

The presentation of information to the user (the outputs) is also of significance to the interaction because this is where the processing presents itself to the user, providing information back to the system users. Today, this is mainly done through a screen display, and it is vital that this is appropriately and properly used to communicate information accurately and concisely. This is more significant in phone displays because they are smaller but becoming far more significant as interactive devices. Because so much of the screen "real estate" is required to support interaction, it needs to provide the appropriate visual clues to enable the user to interact with the application. This, in turn, requires an understanding of how we perceive and interpret information we see.

Natural Language

An alternative to manual interfaces is to use natural language and speech input. One of the earliest examples of natural-language input to a computer (and the related natural language output) was ELIZA, written by Joseph Weizenbaum (See Weizenbaum, 1966, referred to in Suchman, 1991, p. 22f). Although this was not spoken but typed, it still demonstrated some of the major issues of natural language inputs. The simple mechanism used in ELIZA was to exhibit two of the features of a good listener – repeating back what is said and picking out specific words to further the conversation (see Norman, 2004, pp. 188-9) – ELIZA utilized a database of counselling psychology phrases. Because a good counselling psychologist is also a good listener, this was sufficiently convincing for the purposes it was used for. The problem is that any apparent intelligence in the conversation came from the human participant, not the computer (see Suchman, 1991, pp. 71-72). ELIZA had absolutely no emotional response or engagement in the conversation (see Picard, 1997), something which may have helped give a sense of "professional detachment" from the perspective of the human participant but is a core aspect of genuine natural language interaction.

ELIZA, while being significant in the development of natural language interaction, was not very good at it (see Shah, Warwick et. al. 2016, where it scored very poorly on their scale). It provided surprising insight into the field but was not a productive route for more general natural language interaction.

One of the problems of trying to utilize natural language interfacing is that this type of communication is extremely complex (Suchman, 1991, p. 41). Norman (1993, p. 9) discusses the fact that without an interpretive background, normal speech is highly ungrammatical and very difficult to understand or replicate. Suchman (1991, p. 46) refers to an experiment where individuals were asked to express what was said to them in as much detail as possible, with as little assumed background as they could, and the experiment demonstrated that this was impossible to achieve because we do not (in fact, cannot) use language in that way. This is similar to the challenge of trying to talk to a computer in natural language.

One of the reasons (Suchman, 1991, p. 60) is that communication between humans is very subtle – gestures, including facial expressions, the way the person stands, the moves they make are all a critical part of a primarily verbal communication. Tones and pauses, the merging of words and the dropping of endings make the verbal expression complex to interpret in a literal fashion. Additionally, overlapping voices, which are part of most normal conversations, add a new layer of subtlety. These subtle rules of conversation are (Suchman, 1992, p. 80) “inexorably meaningful”; they cannot be ignored, because they are part of the whole communication. They change, sometimes in small but important ways, the precise meaning of the words used.

Other challenges to utilizing natural language in the interface are the changes in an individual’s speech when they are tired or ill and the even more complex differences between regional accents and word usage (see Forsberg, 2003). As Tufte (2006, p. 160) points out, the human mind is able to process very large amounts of information and to find patterns in it. We do this far better than computers are able to, which is why, largely, we can understand a range of accents and even other languages.

Finally, Suchman, (1991, pp. 78, 80) points out that accidental *non sequiturs* are actually very rare in real conversation because of the wider context – because the participants share some contextual knowledge that helps them communicate. Understanding the route from one comment to another will often involve knowing and understanding the background of all participants. Without language context, which is something exceptionally difficult to include in a computer model, most human conversations are obscure and obtuse. The majority of the information in a conversation is not in the words alone but in the words set in a wider context known to the participants. So, for structured data (see Kaufman, Sheehan et. al. 2016), voice input can work well (ELIZA was a form of structured exchange), but not for wider or less contextualised use.

All of these reasons are what makes the introduction of Spoken Dialog Systems (see Hirschberg and Manning, 2016) like Siri© so remarkable. However, these systems are not handling true natural speech – they handle phrases, words and short expressions spoken clearly and directly. They still lack the ability to engage naturally in conversation.

Communication

Successful communication with computers relies on good information being provided to the user by the computer system. There are issues in computers communicating with people similar to those the other way around. For all of the reasons discussed above, natural language is not a viable choice for rich information supply from a computer. One of the most important aspects of this is our ability to cope with very high levels of sensory input, as long as the information is related and contextualised (Tufte, 2006, p. 160). This echoes to the work of Miller (1956), who devised the well-known idea that we can hold seven (plus or minus two) items of information in our short-term memory at one time. However, what is crucial to maximise this is the context of the items. Bruner (1986, p. 46-7) states, “You can get a lot of expected information into 7 slots, but much less unexpected information.” He also showed how important context is, using a playing card experiment (Bruner, 1986, p. 47). In this, he showed playing cards with the right images, but some of them had the wrong colours, and they took significantly longer to be recognised. Context plays an important part of our perception, not just the immediate context but the wider historical context of images.

Context is an important part of how we receive information. The context of what the computer wishes to communicate with us will influence what we understand by information presented. Some of this context is what Tufte (1990, 2001 generally, 2006 p. 152) refers to as the “Data/ink ratio” or “chartjunk”, both of which are reflecting that the amount of presented detail should reflect the amount of information being presented. If there is too much “junk”, it is harder to find the significant information. Tufte discusses charts from Pravda, the Russian news agency, which typically overemphasised the improvements being made in the country. This was often done by increasing the volume of presented chart bars (where it is only the height that is significant) or other ways of misrepresenting the information being shown. He compares this with railway timetables, of which he has a number of examples (see Tufte, 1990, pp. 24-26 and pp. 46-7) where very large amounts of data are being shown in a (relatively) confined space. What is significant is that we can interpret the timetables – we can take in the quantity of data being displayed and interpret it in the context of the rest. Additionally, the wider context of the data displayed changes the way we see the individual data items. Context not only helps us interpret individual data items, it also changes the way in which we interpret the totality of what is presented to us.

The importance of context in interpretation was further explored by Bruner (1986, p. 45). The view and interpretation of a castle changes when the spectator realizes that it is Hamlet’s castle – the context (history and associations) change the way we see and interpret the castle. Similarly, the play Copenhagen (Frayn, 1988) is based around this concept – that the way you interpret certain specific events of 1941 depends on how you want to interpret them. In fact, as Bruner says, “We characteristically assume that what someone says *must* make sense, and we will, when in doubt about *what* sense it makes, search for or invent an interpretation of the utterance to give it sense”. Our minds will make sense of received information wherever possible. In the context of our thesis, this means that we must be provided with appropriate information in a context that enables us to interpret it correctly. It is not only the raw information that is relevant but the context that should enable the user to appropriately identify the next action to take.

While Tufte (1990) provides excellent insights into static and printed media, he does not cover specifically the more dynamic computer interface medium. Here, the idea of a story (also explored by Bruner, 1986, p. 42-3) can provide the context that can aid the communication. The achieving of a task is a narrative or story with the participants moving from one scene or action to another and taking actions appropriate to the new situation. Bruner (2003) calls this “Intentional State Entailment”. It is a contextually consistent communication, or “Hermeneutic Composability” as Bruner expresses it. Bruner also is clear that stories engage with us as people because as humans, we relate to others and our history in terms of stories.

Perception.

It is a peculiarity, according to Gordon, 2004, p. 219, that “There has not yet been a satisfactory theory of vision.” What this means is that we do not fully understand the intricacies in how we actually see and interpret images. Perception is about what we **understand** about what we see, rather than the straightforward image our eyes receive. Our brain interprets and models the image on our retinas into the interpretation we think we see. At the core of this interpretation is, once again, context (Gordon, 2004, pp. 17, 62, 100, 156) because the messages our eyes receive are put into a context and the interpretation formed out of that context. This context consists of what we have just recently seen, what our other senses might tell us, and what we know about the world.

The challenge of this for software display is that the perception will not necessarily be as the software developer intended it or even the same between different people. This might be one justification for allowing users to customise their own interfaces (to an extent) to aid **their** perception of what is going on. There are positives to this uncertain perception processing, as Brunswick (in Gordon, 2004, p. 58) points out. What he explains is that uncertainty is important; that we do not need to see things perfectly for the brain to correctly interpret the reality. Tufte (2006, p. 50) puts this more succinctly, that it is better to get displays “approximately right, rather than exactly wrong”, because the brain’s perception mechanisms will sort out the details. As Hoffman, Singh and Prakash (2015) argue, “Perception is an adaptive interface.” The better we understand this, the better our ability to use the perception process to our advantage – that is, the less work we need to put in to trick the user’s brain into seeing what we want it to see. There are four main theories currently that try to explain the perception process, each of which has an impact on flow.

1. **Gestalt theories**. These theories are based on the **context** of what we see, and they argue that the context affects everything we perceive. The extra that the context brings is the Gestaltqualität, from which the theory gets its name (Gordon, 2004, p. 13). It is a "phenomenological" theory, based on what people actually see, not just on what they are supposed to see. A computer screen, for example, is just a whole set of coloured pixels, but we perceive something significantly more than just colours because we interpret the images we see.

   These theories tend to break down when trying to explain **why** we perceive as we do (Gordon, 2004, p. 21-27), but they do help when trying to explain **what** we actually perceive. One insight that is very useful (Gordon, 2004, p. 48) is that expected visual information is easier to see than unexpected information – we can perceive what we expect to see easier than what we don't expect (See Krug, 2000, p. 34-36). The why questions are answered more by neurophysical ideas (Gordon, 2004, p. 86-88), but the reductionism in these approaches needs to be balanced with other approaches.

   The Gestalt principles, as outlined by Few (2006) are that we perceive:

   a. Proximity: Things close to each other tend to be related.
   b. Closure: We like closed, finished shapes, so we can sometimes display less and let the brain finish it off.
   c. Similarity: Similarly-shaped/orientated items are related.
   d. Continuity: We fill in missing parts of lines. This means we can use alignments to group things.
   e. Enclosure: Enclosed object are related.
   f. Connection: Connected things are related.

We like to see patterns in things, and so we recognize these patterns rather than the reality of what they present. Patterns can be used to improve interactions, using these ideas to give the user visual clues. More significantly, they can be used in a consistent way to enable users to more clearly understand where they are and what they need to do next. As an example, keeping action objects (like buttons) all looking the same helps to identify them; drawing lines around sets of objects helps to see them as a whole (and so as one object, not as multiple objects).

2. **Empiricist theories**. These argue that we work from hypothesis about what we see and interpret the visual information in terms of whether it supports or dismisses each hypothesis (Gordon, 2004, Ch. 5 passim). This means the perception process starts from our minds and understanding rather than, as with Gestalt theories, the visual input. Of course, the question this raises is how we verify or dismiss various hypotheses (Gordon, 2004, p. 135-6) and does not manage to explain how some things are very quickly perceived, ruling out a cognitive process of this sort. These ideas can be used in interaction by presenting what people expect.

   In terms of supporting or enabling flow, this means that we should present information to a user as they expect it to be. Once again, this may differ between users, so allowing them to modify the interface is important so that their hypothesis about what they are seeing is either reinforced or negated. This may be as simple as displaying a negative balance in red at the top of a screen and a positive one in black, or as complex as arranging clear graphical presentation of a mechanical system so the user can see at a glance anything that doesn’t match their hypothetical “perfect” state.

3. **Direct Perception theories**. These argue that “understanding perception requires a joint study of the organism and its environment” (Gibson, in Gordon, 2004, p. 146, 155). As de Wit, van der Kamp and Withagen (2015) put it, “[O]ne and the same environmental situation can cause different retinal images.” In other words, we cannot understand perception without

taking into account the rest of the organism and environment, specifically the other sensory inputs.

This holistic approach to receiving information matches well with the approaches we looked at earlier for communicating to another person. Significantly, it implies that there are certain limitations to the way computers can communicate back, at least with current technology, because computers are not yet able to use the full range of senses to communicate with us or to fully understand the environment, although mobile devices will often reflect orientation and movement. At the same time, mobile devices are also used in unusual situations – outside, in tunnels or caves – which can impact on the usability of software.

4. **Computation theories**. These treat the brain as a complex computer. They interpret the way we perceive as purely computational (Marr, in Gordon, 2004, p. 205). This idea of the mind as a complex computer has been heavily criticised, as more insights into the mind have been discovered. In particular, while computational processes might be able to mimic some brain functions, that does not demonstrate that this is the reality (see Wien and Pona, 2015, especially the Chinese Room Mental Experiment). Reducing the human aspect to another computational process is not only dehumanising but fails to appreciate the distinctions in the human aspect of the human-computer system. This approach will not be considered.

These theories give insights into the best way of presenting information to a user in a way that can either enable a flow experience (or conversely, frustrate the user). Using the gestalt principles, ensuring that the display is what people expect and considering the environment that the user is liable to be in can help to enable a flow environment. Understanding the theories and the way that we perceive what is shown us can help to understand how visual information should be most appropriately displayed to provide the maximum content with the minimum cognitive challenge, enabling the flow experience. Being able to obtain the information needed immediately is important. But the variation in the theories makes providing a single style difficult, so flexibility is important so that an individual can produce an interface that works for them.

Activity theory

While it is important to look at the practicalities of communication and interaction, we also need to consider the whole picture, the interplay of the different players in an interaction. This is what Activity Theory (Nardi, 1996, p. 7) seeks to understand, working with the “asymmetry between people and things”. This means that the human and computer partners of the interaction are not interchangeable; each has a distinct role to play. Nardi also states that “[P]eople and artefacts [that is, tools or machines] do not occupy the same ontological space.” In other words, they are fundamentally and conceptually different. The interaction between them should therefore not just be focussed around what the computer can do but also around what the user or operator can do. Interaction ideas need to understand the full scope of the interaction and how the different players – the people and the artefacts – can contribute to the entire exchange.

The experiences that Zuboff (1988, p. 6) describes from his observations in the Perry Wood pulp mill reflect the problems when the user’s role and tasks are not taken into account. People who were displaced by the introduction of computers experienced “profound disorientation and loss of meaning” because they were being treated ontologically in the same space. As the project continued, the importance of “intuition” or “instinct” (which is a distinctly human factor) was noted (Zuboff, 1988, p. 55). Csikszentmihalyi (1975, p. 169-170) did some work on flow deprivation, the participants using phrases like “tense”, “slept badly”, “more irritable”, “listless, worthless”, reflecting similar ontological angst as was observed in the Perry Wood workers. This would suggest that ensuring computer system users are in the right ontological space is important to the people themselves for their well-being.

The loss of a sense of worth seems from the Csikszentmihalyi and Zuboff studies to be related to an increase in frustration in quite fundamental terms. The Zuboff study indicates that in a work environment, this appears to be a more significant aspect. The skills and knowledge obtained and used in the work environment appear to be more significant to the sense of self-worth. Mediation theory (Verbeek, 2015) sees people and artefacts “not … as two ‘poles’ between which there is an interaction, they are the result of the interaction”. The context is also considered, that one particular interaction is part of the entire human-technology relationship.

Enabling flow is, in part, about understanding the ontological place of the participants (not necessarily just one person and one application – the participants may be multiple people and applications, each with a specific impact on the interaction). As Clemmensen, Nardi and Laptelinin (2016) show, the use of activity theory in HCI research is proving productive, in particular in studies of flow (Young, 2008), although as this relates to children, it is not of direct use in the current study. This can provide a more holistic view than the Situated Actions of Suchman, as discussed above, and between them they can provide approaches to grasp the whole picture as well as the situated picture and find ways that the entire experience is a positive one.

Work and Leisure

One significant development in computer usage was the expansion into the home PC market, simply because of the increased number of users. The real challenge was understanding what computers would be used for: “While the most frequently mentioned intended applications were for business and education, in reality, families tended to use their PCs more for word processing and games” (National Science Foundation, 2001). The way that people engaged with their computers changed when people used the computers at home (for leisure) rather than in their work environment. While the basic tools are the same, our second hypothesis is that there is a difference in terms of usage and interaction.

At one time, this distinction would have been straightforward. Today, the ubiquity of computer usage, and the ubiquity of some applications across both of these areas, makes this distinction more difficult to define simply. There are various authors who look at how "work-like tasks" are identified, however we will initially look at the categories identified by Csikszentmihalyi (1975) as tasks that are not voluntary, time-dependent, externally defined and unpleasant. These will be considered and modified to reflect a wider sense of these tasks. It is important to get a clear definition here because "the blurring of work and non-work activities is clearly a complex issue for contemporary HCI research" (Grönvall, Ciolfi et. al. 2016). Providing definitions that draw from a range of sources, it is hoped to clarify the scope and context of this work.

1. **Non-voluntary/abandonment cost**.

   Csikszentmihalyi (1975, pp. 3, 25) expresses this by saying, "The same activity is experienced differently if we originate it or are forced to do it." While this is true, it is not necessarily the case that work-like tasks need to be non-voluntary. In particular, unpaid "voluntary" work would be included in a definition of "work", even though it fails to meet this criterion. Haworth and Veal (2004, p. 2) refer to tasks that are required to retain the work, which reflects better as it includes unpaid work. They also criticise the sense of non-voluntary because it implies that work as such may not be meaningful. They argue that work can be, and is better when it is, a meaningful experience.

   This is not to dismiss the reality that for many, work lacks meaning. However, this is not an essential aspect of it, and (as Bailey, 2016, discusses), people often find the meaning is that they would miss work if they didn't have it. This indicates that simply describing it as "non-voluntary" misses an important aspect of work experiences.

   On the other side, as Costikyan (2002) points out, leisure activities can also involve requirements and rules, especially, for example, when participating in a sport or a corporate activity.

   Csikszentmihalyi's argument that such tasks cannot be enjoyable is mistaken, even by his own findings. Csikszentmihalyi (2006) covers a range of people who, by and large, enjoy their work. He is right that there is a difference between us initiating it and being forced to do it, but this does not imply that it cannot be enjoyable. As Costikyan (2002) points out, in the converse, sometime leisure work also involves challenge and conflict, but this does not mean that it is not enjoyable, despite being restricted or driven by others.

For these reasons, a different wording will be used here, which is about **abandonment cost**. This idea is taken partly from Krug (2000) and Nielsen and Loranger (2006), relating to the decreased cost of abandonment for users of commerce web sites. A customer searching for a product can leave one site and move to another with little or no cost (to them), something which the designers of sites need to be aware of.

If a task is broadly voluntary and leisure-based, then the cost of abandoning the task tends to be low. More particularly, if one tool or website is not working, there is a choice of using another one. Within a working environment, this is usually not possible, so the cost of abandoning the task is high because performing that task, using those tools, is part of the job. If the user choses to abandon that task, they may lose their job.

2. **Time-Dependent/resource dependency**.
   Time criticality is covered by two aspects of Csikszentmihalyi's analysis. Firstly, there may be deadlines imposed on the tasks externally. This means that someone else requires you to achieve this task, and it is part of your working agreement. Secondly, there is a resource cost to taking longer over the task – the longer it takes, the more it costs the organisation which has imposed this task on you. "Cost" does not just mean a financial cost, as it could also be a loss of client confidence or failure to meet regulatory requirements.

   While this is an important aspect of work-related tasks, non-work tasks can also show some of these features. Costikyan (2002) discusses various games, some of which have time restrictions or time pressures. There can be financial penalties to not achieving a purchase at a particular time, even for leisure usage, and there are regulatory requirements on individuals as well as organisations that have deadlines (such as tax returns).

For these reasons, we will use **resource dependency** instead of a (more specific) time limitation. Higher resource dependency would indicate work-like tasks, whereas lower dependency, leisure-like tasks. This is analogous, on a personal level, to Resource Dependence Theory (Pfeffer and Salancik , 2003, quoted in Klein and Pereira, 2016) applied across businesses, based on the idea that an organisation's survival depends on their ability to acquire critical resources from the external environment. In the same way, a work-like task is a task that requires critical resources from outside the direct control of the individual. This can be deadlines or other people's contributions to complete the task.

Externally defined/External control.

This expands on the non-voluntary aspect, but emphasises that the tasks undertaken are defined by other people. In the context of computer applications, this may mean that the decision about software or hardware is out of your control and is what Rogers (2003, p. 403) calls an "Authority Innovation Decision".

There are always those who actually make these decisions, who may have far more influence in what they are using than others, but even they are almost always driven by external requirements. These might be money or feature requirements of software. They may want to use one spreadsheet package but be restricted because it costs too much or because other software only interacts with a different spreadsheet package. These decisions may then impact the usability of the purchased software.

It is interesting to consider the van Dam "butler" model in this context (van Dam, 1997). The butler knows your personal needs and requirements, and "discreetly does [your] bidding by anticipating [your] needs". This is what van Dam holds as the goal to strive for in terms of interfaces. The problem with this model in terms of work-like tasks is that someone else's butler may have a different set of priorities and be allowed to override yours. This butler override is defined as **external control** over a task, which would define it as work-like.

For leisure-like tasks, there is much more control over what is required – you can purchase from any web site you want, buy and use any word processor or spreadsheet package that you find most appropriate. You can, in this context, direct your butler to work purely for you and not have any other pressures driving your choices or working environment.

3. **Unpleasant**.

Csikszentmihalyi says (1975, p. 1) that "because our assumptions are that work is unpleasant, it tends to be", a statement that he doesn't provide support for and is not supported elsewhere. He goes on to say (pp. 3, 25) that "What we **must** do cannot be enjoyable – this is the assumption built into our psyche."
Contra this, both Bovier and Perneger (2003) and Glenn and Weaver (1982), show that a good proportion of workers that they studied are satisfied or enjoy their work. This clearly indicates that "unpleasantness" is not a core aspect of "work" tasks, although these studies do focus on specific sections of the employment market and ones that might be more satisfied than average.

This is not to dismiss the fact that, for some people, work is difficult and unpleasant. The images that Marx (1946) showed of the working environments in the 1900s are history at least in the West. This is partly because the West has moved the difficult and dangerous work to other countries like China, where some of these problems may still exist. Some aspects of work, and some industries as a whole, will be unpleasant to work in. However, this does not serve within a **definition** of work tasks, and so this criterion will be dismissed entirely. It is hoped that by properly exploring the role of the human and computer or machine aspects of a task achievement, some of the more disagreeable aspects of work can be removed by appropriately valuing the human participant, something that we will look at below.

Other studies do define work in different ways. Dubin, 2017, defines work as "Continuous employment for the production of goods and services for remuneration". However, the "continuous" nature of this – that is, something that is performed as a career, over a long period of time – while relevant for his study does ignore many casual and part-time workers, those who work on a contractual basis and so may not have continuous work. The "remuneration" aspect removes voluntary workers from consideration. The tasks we will use are relevant across these boundaries.

Work and workers

What does "appropriately valuing the human participant" mean, particularly within the context of a work situation? As Haworth and Veal (2004) explore, work is an aspect of a society, of a culture, of how an individual sees themselves, particularly in respect of their own society. By considering the significance of an individual's well-being and looking at why we work and what the meaning of work might be, we can find some direction for valuing the human participant. This is also a direct result of considering Activity Theory and the "asymmetry between people and things" (Nardi, 1996, p. 7).

This is important, because we are looking at whether the flow context can make software users "happier" in a work context, and part of this happiness, as well as part of the flow context, is individuals feeling valued. Anderson, Crous, and Schepers, 1996, show that flow scores and the quality of work life are correlated, and part of the quality of work life is being valued in the workplace.

Haworth and Veal (2004, p. 29) show that the time people spend in work is now lower than it has been, at 2000-2500 hours a year, but substantially more of this time is spent using computer systems. If we are to deal with the human aspect of the computer-human interaction as something more than just a component in a system, the interaction with computers within the work environment is a crucial aspect. In particular, do we identify the human aspect as distinctive or just another component in a larger system?

Zuboff (1988, pp. 3-4) indicates the fear that "in 15 years there will be nothing for the worker to do", because the nature of jobs and work was changing rapidly at the time she was writing. This is still the case, and yet she also says, “Without the technology, the company would not remain competitive”, so the rejectionist approach that Marx would advocate does not work. Without an appropriate engagement with technology, the business will not survive, so the challenge is to engage with the technology without the dehumanising of people that Marx saw (and rightly challenged).

Identifying the human participant as ontologically different to the computer participant does lead to a religious perspective. Within Christianity, which is the predominant faith influencing ideas in the West, there is an important strand of thought that gives humanity a value because human beings are made in the image of God – the *imago dei*. This is an important concept, even if not always clearly understood. This is not the place to explore the *imago dei* doctrine fully, as it is complex and controversial, simply to say that the guiding passage is Gen 1:26, where Adam is said to have been creating in the image of God. Through Adam, all other humans also have something of the image or characteristics of God within them – humans are special or distinct because there is something of the divine within them (see Myers, 1987, p. 515, article “Image of God”). Even when this is not explicit, the idea of humans being made to “emulate” God, or to attempt to be more “Godlike” (Hill, 1997, p. 14) reflect a similar perspective, that humans are divinely related beings, and so our responsibility to each other needs to reflect this.

The ontological perspective of humanity is distinct from the ontological view of computers and even other animals (Nardi, 1996). This is an understanding of humanity that values people but does not dismiss everything else. As Bannon (1991) puts it, “Emphasis is placed on the person as an autonomous agent that has the capacity to regulate and coordinate his or her behaviour, rather than simply being a passive element in a human-machine system.” The human aspect includes the importance of emotion. Cañamero (1999) argues that emotion is an important part of the interactions and that it helps when the computer appears to demonstrate emotions. This is reflected in Peter, Crane, Axelrod and Beale (2006), Crane, Shami and Peter (2007) and Peter and Blythe (2005), who all argue that an emotional engagement with the software application makes the interaction appear more "human" or "natural". This is not a question of making applications that emulate human emotions as much as designing applications that take into account the importance of emotion in humans and the significance of an emotional response. This is about mediation theory and understanding the full, wide context of an interaction.

The change in knowledge basis

Zuboff (1988) discusses the progression from business knowledge (p. 255) to education (p. 227-8), which has had the effect that work becomes information-based rather than business-knowledge-based. This means that the data (un-interpreted information) that is provided by the line workers is used – manipulated by the reporting systems – to provide information to the management (see p. 234-5). This has meant that the power has moved from the knowledgeable workers (see Zuboff, 1988, p. 164) not to the management, but to the IT staff who control the flow of information (p. 250).

The result of this is that the **control** and the **knowledge** are separated because those who work at the front end (in Perry Wood, which is the example from Zuboff, 1988, this means the people who manage the machinery) know what is needed, but may not be able to control the systems well enough. The management are often at too much of a distance (physically or chronologically) from involvement, and the IT staff are unlikely to know enough about the intricate working of the plant without help. An acknowledgement of their skills and understanding is empowering to the workers, and giving them responsibility, not just treating them as local button-pushers, improved the quality of the work at the Perry Wood plant. As Bannon (1991) says, "People are more than a sum of parts, be they information-processing subsystems or physiological systems, they have a set of values, goals and beliefs about life and work." These goals and beliefs will impact the way that they approach their work and their interaction with the system.

This change, where the computer systems are actually in control (because they provide all of the information to the decision makers) reflects Baudrillard (see Kent) and his concept of "hyperreality". This is a philosophical theory that the presentation of information becomes the reality, whether or not it truly reflects any more objective truth. As Baudrillard himself puts it (1995, pp. 14-15), “Establishment of the truth is impossible where our means of seeing what happened are controlled by others.” While the original was referring to the media, the same principle would appear to apply to computer generated information, because the control is similarly in the hands of others.

Baudrillard also shows that the truth as shown through the computer system reports becomes the truth for everyone. The aim of the business is to improve the figures on the reports, because they are the only way people have of ascertaining what is going on. Any sense of an “objective” truth becomes more and more distant (Baudrillard, 1995, p. 51). This is where Zuboffs (1988, pp. 349, 354) argument that data is in some way “objective” is wrong. The processes by which data is gathered are subject to all sorts of influences, because the producers of the data have some appreciation of what the results are expected to be. There is a tendency to provide information that fits with the expectations, ensuring that the “system” shows all to be in order. The definition of “everything being OK” is no longer that someone has looked to check that all parts are working fine; it is now that the data says all parts are working fine. This is as accurate as the gathering and processing tools make it.

There is another level of this disconnect from reality, which is the interfaces between the user and the back end of the computer system. The interface that is presented is trusted to represent the data or information that backs it up. While making sure the interface is one that engages appropriately with the user, as the user trusts the interface to be representing the state of the underlying system, it must be clearly accurate. Enabling flow in a work environment means, among other things, that the interaction is natural and accurate so that the user can concentrate on their task, not on the tools they are using, and know that the information that they are presented with reflects a reality that they can engage with. Contra Csikszentmihalyi (1975, p. 197), with the right tools, it should be possible to experience flow in a work environment (something that we will explore more later in this study). In fact, if the tools are appropriate, it should be the normal experience for many people.

Introducing technology

One of the important aspects of work and computer applications is the introduction of new technology. Bannon (2011) is one of the few people to question how little discussion there is on the moral level in “augmented memory” projects, which are applications aimed at remembering everything for us. Our ability to forget (or at least ignore) things is important to our interaction with the world around us and to better being able to process the important things (Kuhl, Dudokovic, Kahn and Wagner, 2007). According to them, we need to forget things so that the cognitive resources are not used in identifying the relevant information. Reducing this forgetting ability can impact our capacity to make decisions. Introducing this technology without consideration on the potential impacts may have significant and serious consequences.

In Zuboff’s study of Perry Wood (1988, p. 210), the organisers admit that “implementation planning was not our strength”, so despite large amounts of money spent on technology, the issues of how people and computers would interact and work together was not addressed. Bainbridge (1983) explains one of the problems they found at Perry Wood: that control systems are put in place because they are more “efficient” or “reliable”. However, this ignores the fact that they require monitoring by humans who are more skilled rather than less, as they need to be able to deal with all of the unexpected events, often on their own. The skills that would have been spread over a team, and utilised as a team, are often now required in one person, without others to help in the decision making. They may have experienced flow, using their skills to keep things flowing, and only occasionally broken out of this to adjust or correct something. Now they are liable to be permanently fixing or checking and never achieve a flow state.

Bannon (2001) makes the point that the human factors in a computer-involved environment are the most resilient and adaptive parts of the system. The people can resolve error situations and improvise where needed, something that the computer parts of the system are not able to do. The focus, according to Bannon, should be on using computer technology to support people doing their jobs, rather than, as at Perry Wood, trying to replace people with computer systems. Interestingly, this approach is supported by Schröedinger (1992, p. 116) in his exploration of the reality of mind, where he says “The machine must take over the toil for which the man is too good, not man the work for which the machine is too expensive” – an approach which, even though written before the impact of computers, treats the machine as the subservient partner, there to help the person fulfil their task.

Roger’s work (2003) on innovations is an important insight into how people accept new ideas and new technology, especially new technology that changes the way people work. Zuboff (1988) found the same in Perry Wood, that peer views and opinions weigh strongly with people. They often have more weight than the formal rules and demands from management, in terms of how people respond to new innovations. Telling the Perry Wood workers to use the systems was not as productive as winning them over, getting the innovators and early adopters to support it, who will then slowly win over the rest. According to Roger, if these people are not persuaded **for** the innovation, then they are more likely to persuade others – who are less excited about new ideas and changes – **against** it.

The response to change is important, because people do not always respond well to change – to the introduction or modification of our environment (see Makridakis, 2017), and there is fear of change (see Frey and Osborne, 2017). This fear will have a negative impact on acceptance and tend (with Rogers above) to be against changes, rather than for them. This fear, will have a negative impact on the flow experience, because flow requires a focus on the task at hand, not on the possibilities ahead.

Leisure

For the purposes of this study, “leisure” and “work” are defined as opposites. Leisure is defined as “not-work”, starting from the definition we have adopted above for “work-like” tasks.

1. **Low cost of abandonment**.
   That is, the cost or consequences of abandoning the task is low. This is not to suggest that the tasks are unimportant, just that the implications of failing to achieve them are less significant. It might be that their significance is for only a few people or that failing to achieve them is simply inconvenient.

2. **Low resource usage**.
   This means that the activities do not generally take external resources. They don’t delay other people or involve costs for the time taken, and they don’t involve large amounts of money.

3. **Low external control**.
   This means that the activities or resources required are very significantly controlled or defined by the person engaging in them. In the context of this study, this means that the person can make significant choices in terms of software they can use.

All of these reflect the position of Stebbins (2005), who rejects the more traditional assessments of “choice” and “freely chosen” activities in favour of what he calls “uncoerced behaviour”. The low levels of these three signifiers indicate a lack of coercion in the fulfilment of the activities. While they are not reduced to zero, as there is a degree of what Stebbins (2005) calls “agreeable obligation”, that there are obligations to participate or engage in the activities, but that this has an autotelic payback; it provides positive feedback to the participant (an aspect that is important in a flow experience). However, the boundary is not clear-cut. It is a spectrum, where activities will fall either towards the leisure end or the work end rather than being clearly divided.

The distinction between work- and leisure-like tasks is not based on a colloquial understanding of this, as paid employment. Buse (2009) points out that one of her interviewees, “William”, was in retirement but still doing voluntary accounting work. This activity would be more work-like tasks, especially as he was bringing his professional experience into his voluntary work and because he had professional responsibilities. This is contra the position of OEDC (2009), where leisure is defined as “the sum of non-compulsory activities”; the compulsory state is one factor in the spectrum, but not the only one. Defining leisure is more complex than this, especially in terms of computer usage, which often blurs the distinction between working and leisure. The definition of a task as “work” or “leisure” is not, primarily, about the nature of the task itself, but the context that the task is set and performed in.

Chapter 3 - Methodology

This study contains a range of studies with differing methodologies. This chapter will examine the different approaches and why they are taken.

Methodology overview

The process and structure of this study follows two related threads. Firstly, there is the process of looking at how people interact with software. The literature review contains a number of areas relating to understanding the human aspect of interactions and processing of tasks. This is all considered in the context of task-based environments, and draws in investigation about the distinction between work and leisure-based contexts.

The other thread is more definitively flow-orientated, and investigates the practical, real-life experience of flow, before feeding this into both of the experimental phases. Both the process breaks and frustration experiments are driven by the concept of flow, and the experiences that participants have of flow experiences.

These experiments are intended to provide more insight into the hypothesis for this study.

Empirical data collection methodology

This study explores the experience of both flow and frustration, within work-like tasks. The first part of the study involves conducting pilot interviews, drawing from some of the questions considered above but seeking to ascertain from scratch whether the issues people actually experience match with the theoretical issues. As Moggridge (2006, p. 131) says, “Design ought to start from the problem people are having and also from ideals”, and this study starts to look at the problems, whereas the theoretical ideals have been identified in the literature review below. These interviews and questions are to ascertain that these theoretical and real-life issues match up. Especially, “do the problems that people actually report match with the flow and frustration issues addressed above”.

The questions asked in the initial interviews are positive and negative aspects of computer applications, which are indications of flow (positive) and frustration (negative) problems. Additionally, looking at where participants go to find resolutions provides some insight into how engaged they are with the product and the computer system. The second experiment takes these categorised positive and negative and explores whether they work in a different environment – posts on internet forums. Once again, this is looks at whether or not the planning and processing of computer mediated tasks are enabled or supported by the products under consideration.

The next part of the research consists of a number of questionnaires, covering topics of work and leisure, computer usage, and flow. The first of these is intended to identify differences between work and leisure usage of software. The aim is to confirm that usage is distinct and that the categories are valid. One important issue is the question of work specific applications, which relates to the issues of new technology, and of learning software packages for new employees. This also explores whether there are significant work-only software tools and whether the valuing of the people aspect of the system is an issue. For leisure usage, where it is not mandated, there is a strong financial drive to make them usable. For mandated work usage, there is less of a drive, so the human aspect can easily get lost. There is also a possibility that work-related software tools may be more complex in use because the work tasks are liable to be more complex. What is more, generic tools will work for the simpler tasks, so work-specific tools will need to be more complex and function with more difficult problems.

The computer usage questionnaire will ascertain the positives and negatives to obtain more details about the responses that individuals have to computer applications, drawing from Csikszentmihalyi's work and the previous studies. It is drawing further into the issues that have already been raised but with a larger group of people enabled by using online questionnaires rather than face-to-face work. Finally, the flow questionnaire looks at whether people have experienced flow in their normal lives and whether computers are positive or negative agents in this.

The next study is an experiment around the perception of inputs, drawing from the issues of how we interact and communicate as well as our perception of the screens. This ascertains whether the visual display of process breaks makes a difference to the way that people work through an input screen. It looks at the way that people handle screens, and how they pause in their processing (subconsciously, and only for a brief time). The thesis here is that good breaks, in sensible places, will help to reduce frustration, whereas bad breaks that interrupt the flow of work will increase frustration.

The final study is on frustration points, looking at how people try to pursue tasks that are set, and where their points of frustration and abandonment are. This is drawing from the previous studies to set tasks that will have levels of frustration, and timing the responses. Focusing on the frustration side, this identifies the type of interaction tasks that cause frustration.

This experiment will quite deliberately stress people, because it will be asking them to attempt tasks deliberately chosen to be either straightforward or impossible. It will be made clear at the start that the aspect of interest is the **process** of performing the tasks, not necessarily the **success** of these tasks. Overall, the various experiments offer evidence as to whether treating people as humans with a set of skills to utilise a computer tool is a more efficient way of working towards achieving tasks than treating them as functional resources, interchangeable and without processing abilities. It is interesting that such a result would fit in well with the aims of researchers like Leffingwell (see Zuboff, 1988, p. 118) to fill the working day efficiently; however, it would go against the approach that he took to making this happen, "[eliminating] time spent on co-ordination and communication". Rather, allowing people to communicate and engage with others would seem to be the way to encourage them to work efficiently.

Questionnaires

A number of the studies use questionnaires to gather the information. These were all designed and developed using the guidelines in Bradburn, Sudman and Wansink (2004). Details about the specific questions and scales will be outlined in the specific studies. These studies were done in areas where quantitative questions were being asked, and statistical analysis could be done on the results.

The other justification for using them in the way that they were was to obtain a wide range of responses to the questions. While there was some variation in numbers or respondents obtained, this did get the questions out to a range of people. They were all self-completion, online questionnaires using SurveyMonkey, an industry standard survey development platform. As well as simplifying the distribution and collection processes, this made the gathering anonymous.

While they have been used significantly in this research, their limitations have been acknowledged. As Gillham (2008) points out, they "[get] information from people … by posing direct or indirect questions". But also "one of the weaknesses of questionnaires is that they seek to get answers *just* by asking questions". They are used where the questions are limited but the quantity of data can help to improve the results, alongside other methods of data gathering.

Surveys and interviews

"As a rule of thumb, if you are not familiar enough with the research area … to predict the range of possible responses … you should first use a qualitative approach" (Boynton and Greenhalgh, 2006). For some of the early work in this research, interviews were used to extract more free-form information from users, without the structure of a formal questionnaire. The purpose of interviews in a research process is "to explore the views, experiences, beliefs and/or motivations of individuals on specific matters .... Qualitative methods, such as interviews, are believed to provide a 'deeper' understanding of social phenomena than would be obtained from purely quantitative methods, such as questionnaires" (Gill, Stewart et. al. 2008). It is precisely this qualitative information that is required from these sessions, as an insight into the experiences of people using software applications. At the same time, the difficulties of collecting this data (especially from larger numbers of participants) and analyzing it to provide definitive actions meant that the technique was used sparingly.

Interviews were used by Zuboff, 1988, in a similar way to provide qualitative information about the problems that workers were experiencing. Zimmerman, Forlizzi and Evenson (2007), also used interviews, again to provide the basics as a starting point for their ongoing research. They are working within the HCI community, and talk about "co-discovery", the sense of working together to understand the problems and issues for exploration.

The sense of social interaction (as explored also by Myers and Newman 2007) is important because the expression of problems can be difficult, so the coaching and prompting that a semi-structured interview provides is very useful.

Experiments

There was also some experimental work done towards the end of the research process. Building on some of the insights gained from the questionnaires, these were developed to provide more constrained environments than the previous work. More details of the specific requirements in each case are found in the relevant chapters; however, in general, they allowed a controlled process to take place, with clear and comparable statistics taken from the interactions.

By designing and developing experiments that collected data invisibly and silently (while ensuring that they have given informed consent), the aim of these was to gather information without interrupting the flow (if any) or add to the frustration, where applicable. Additionally, by collecting data directly, this did not involve the physical presence of researchers, who might influence the results.

Statistical analysis

In the statistical analysis of the results, I have used SPSS substantially to provide a range of analysis. This is a standard tool used for statistical work and provides a set of tools for exploring the data gathered.

Identifying the appropriate tests to carry out was done using Field, 2013, as a starting point, and some further exploration beyond this. In some cases, the data was well distributed and provided useful insights from parametric tests. However, in some cases, these were proving less conclusive, so non-parametric tests (which make less assumptions about the data distribution) were also used to determine correlations and insights.

Hypotheses

As computers have become more widely used, and a core tool of business, there has been some frustration because of the lack of comprehensive approaches to allowing wider access to non-technical people. The tools used in early research (like GOMS) have laid a foundation for work today, but are no longer relevant as such because of the increase in WIMP interfaces and the internet, which provides asynchronous processing. The first hypothesis, "**The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems**", is a way of quantifying and categorising this frustration by contrasting it to the concept of flow.

Providing an interaction with a computer system is a complex process, especially if we are seeking to maintain a flow experience. In particular, in work environments where the usability may not be a core factor (so modern paradigms like cloud-based and metro ideas – where applications are available on any device – may not be considered viable or sufficiently secure), the nature of the frustration and the impacts of it may be different, so the second hypothesis is "**Using this flow/frustration dichotomy for interfaces definition will produce financial benefits for organisations.**" This distinction is not purely that the work and leisure environments are different but that the nature of task completion is different. We have explored in some depth already why there is a difference between work and leisure and how we can understand it. The experimental studies within this thesis are attempts to simulate work-like tasks within a controlled environment to enable the gathering of statistics in a way that would not be possible within the scope of this research in a real work environment.

Finally, while the questionnaire information is very productive to understand self-assessed responses, and the interviews provide important and in-depth insights into a number of participant's experiences, the experimental work was necessary to assess the impact later on, to support the final hypothesis, "**A frustration experience can have a statistically significant impact on a participant's task-completion experience.**" This idea has arisen (at least in part) from the exploration of perception and communication, as well as the look into Activity Theory. These have given an indication that interaction is not simply a one-off situation but that there are expectations raised (and sometimes lowered). The "whole picture" of both activity theory and mediation theory implies that actions and experiences are not separate from a broader interaction and engagement with technology. A proper understanding of the ontological position and role of the human participant is important to appreciate that frustration can have long-term impacts.

Ethical issues

This study does not have a substantial ethical assessment requirement, and so was not identified as requiring submission to an ethics committee. All of the questionnaires and experiments have instructions at the start that they are optional and that participants can stop at any point if they wish to. All information is anonymised and there no identifiable information is reported on or retained. Participants are told of this.

In the process break experiment, there is an ethical challenge, because some of the tasks being set are not possible to achieve (this is an important aspect of the study). Participants are told that the purpose is to understand their tasks processing, not the completion and fulfilment of the tasks. However, they are not informed that some tasks are (deliberately) not possible.

Chapter 4 - Initial surveys.

Work and Leisure survey

Our second hypothesis is that "**Using this flow/frustration dichotomy for interfaces definition will produce financial benefits for organisations**". To begin to identify work tasks, a questionnaire was sent out attempting to identify the distinction in terms of computer usage for these two areas. Haworth and Veal (2004, p. 29) show that the time people spend in work is now lower than it has been, at 2000-2500 hours a year, but substantially more of this time is spent using computer systems. This questionnaire is an attempt to explore how much computer usage people have in their work environments and also to compare against their leisure usage.

Some research was done to identify existing studies covering this area. Much of the work consists of either work or leisure usage, relating to other aspects of their lives, however, I want to consider the same individuals in both domains. The increase in computer usage in work environments (as noted by Haworth and Veal above) seems to be matching the more general increased usage (see ONS, 2016). While this report is relating to internet usage, the fact that "Regular internet use continues to rise, with more than 8 in 10 people going online almost every day in 2016", is an indication that usage overall is increasing.

Method

This survey aims to identify the differences between usage of computer systems in a working environment and in a leisure environment as reported by users. Because of the complexities of differentiating clearly, the participants were allowed to self-identify what they considered work and leisure activities with some indicators. The aim was not to find out if people identified these correctly but rather to ascertain differences in usage between what they classify as "work" and "leisure" uses.

Participants

51 people participated in this survey, of whom 41 answered all of the questions. The participant list was drawn from the researcher's contacts, including social media contacts, with a request to pass the link on to others. The results came in over a reasonably short period (1 month), however no distinction was recorded between primary and secondary contacts. The timescale would suggest that most were primary and that there were no tertiary participants. The only restriction specified was that the participants should have a "reasonably full-time job, that is, one that occupies them for a significant portion of the week". This wording was deliberately chosen to include people who may not have what is technically a "full-time" job but who did work sufficient hours to differentiate working time from leisure time.

There were no requirements to identify the type of work engaged in or any aspect of the socio-economic class that they would assign themselves to. However, the means of communication (email and social media) meant that they did have access to a computer and connections to the researcher or his network, implying that they are largely of higher socio-economic grouping. By only analyzing the results comparatively, rather than absolutely, it is hoped that this bias may be made irrelevant. However, there is further work that could be done to ascertain whether the results obtained were largely consistent across a range of social and economic indicators.

No detailed data was collected to explain why some people did not answer all of the questions. It might be surmised that the second grid of questions for leisure usage, similar to a previous one already filled out, may have made some participants give up. Of the 10 participants who abandoned the questionnaire, 7 stopped at the start of the leisure options, and 3 stopped at the "number of applications used in a week" within the leisure section.

The data does indicate that none of the participants are working exceptionally long days – more than 9 hours on computer applications. Along with the initial restriction, it would seem that the participants are all working fairly normal hours, providing us with good quality data for analyzing.

Materials and apparatus

The questionnaire was built and distributed using SurveyMonkey (2011). The questions were divided into work-related and leisure-related and designed to match up so that comparisons could be made. However, the ranges used differed to reflect the different expectation of usage.

For the time spent using computer applications, the work-related question had percentage ranges. While a pre-existing scale was looked at, no appropriate one was found, however, they did indicate that more granularity around the 50-90% range was likely to be important, as the expectation was that this would be where most participants would register (this came substantially from SurveyMonkey example questions). A percentage range was used because it was expected that the participants would have a better idea of the **proportion** of their time than the actual **hours**. For someone who is always working on a computer system, the 90-100% range was considered less threatening than having to say 8 hours a day, as there is still some stigma attached to heavy computer usage, especially for women (see Bernstein, 1991).

For the leisure time, the number of hours per day was used, as the expectation was that participants would have a better idea of this and would be able to calculate on this basis. These ranges were based on a SurveyMonkey question but extended to cover up to 14 hours (they original only covered 10 hours but were related to specific social networking sites).

For the question about the number of computer applications used in work, options of 1, 2, 3, 4, 5-10 and 10+ were provided. The expectation that drove this was that most participants would be using a small number of applications in the working environment, so detailing this at the lower level made sense, while providing a means of capturing the higher usage. This was drawn from some exploration into application numbers used, but the results were based on a range of sources. The follow up question regarding the number of these that were work-only tools had a more restricted range. The expectation here was that most people would have some work-only applications, but there were unlikely to be very many of these for one person. Work-specific was clarified to mean those computer tools used purely in the work environment, purely for work-like tasks.

In the questions about leisure usage, the expectation was that participants would also have a few applications that they used regularly, and the range of 1-6 was used. Again, this was drawn from a variety of sources that indicated this was a reasonable range to use. Those at the higher end may be using multiple applications, many different tools for getting full use out of their computer, or they may simply be heavily using many social networking applications. However, they are also liable to be more experienced with the range of applications that they use.

For their most commonly used application, that is, the software tool they use most in each environment, the number of hours they use in in a day (for work) or a week (for leisure) was gathered. This was designed to identify whether they are using a single tool very substantially or a range of tools. The expectation is that many users will be using a single tool for a high portion of their time because that is what they mainly use the computer for. Other tools are important and necessary, but there is a focus on one particular process. It was expected that this would be true in both work and leisure environments, and the survey was intending to provide either support of refutation for this.

The final questions in the work section are about whether the application is seen in a positive or negative light. The expectation is that most participants would be using applications that are a positive benefit to them, however, it was important to understand where applications may be benefitting others but not the people expected to use them. A free text entry was provided to offer explanations and reasons as to why this was the case, either way.

The final questions for the leisure part of the survey were to seek the reasons for a specific choice. The expectation here was that most would choose the software they wanted to use – an indication of a less externally-defined task. However other options like “it came with the computer” were also offered, to see whether the choice was deliberate or accidental. The options within this section were taken from an internet search for reasons to purchase a particular application, although, once again, no specific source. It is hoped that the questionnaire would serve to clarify the possibilities and ascertain whether the “leisure” use conformed to the definition above.

Procedure

A link to the questionnaire was sent out, and it was completed online. To assess the timing and the wording of the questions, 2 volunteers were asked to complete the questionnaire and provide feedback, which was incorporated into the final questions. Their results were not used in the analysis, as the wording of some of the questions was slightly modified, and it was important that all participants should answer the same set of questions. There were 9 questions, and the initial testing indicated that completion time should be around 10-15 minutes. As an unsupervised task, this was considered acceptable.

The questionnaire was made available for 6 weeks, to allow time for people to receive the request and complete it. Results were downloaded from SurveyMonkey at the end.

Results

After an outline presentation of the results, and exploration of how representative these are, there is cross-interpretation of the data, to extract as much information as it has to give, in particular to better see the nature of the participants’ lifestyles as revealed in the questionnaire. The statistical analysis is done using PAST (Hammer, Harper and Ryan, 2001) and SPSS (IBM Corp, 2013).

Working questions

How much time do people spend using computer applications? The results are shown inTable 4-1, and there is one division of users that appears immediately. Some respondents use computers relatively little of the time, and some use them a significant proportion of the time. In the analysis below, we will refer to these as light users and heavy users. The others in-between are also significant, as the high spots are not substantial. It would appear that the lower end of the proportion should have been more granular, but the expectation was that in most jobs, there is a reasonable proportion of computer usage. At the very least, a category of "none" may have proved more useful – however, this entire group are those who are not highly dependent on computers for their work. The sizes of the categories are uneven to provide more granularity in the middle of the range. Merged figures providing values across more even ranges are shown in the table alongside, for comparison.

Table 4-1 – Reported proportion of work time spent using computer applications.

Proportion of the time	Percentage of respondents		Number of respondents	
0-30%	25.5%	25.5%	13	13
31-50%	11.8%	23.6%	6	12
51-60%	11.8%		6	
61-75%	9.8%	51.0%	5	26
76-90%	25.5%		13	
91-100%	15.7%		8	

The **number of applications used** results are shown in Table 4-2, but the lower end has been combined, as the numbers were very low. The immediate results indicate that the number of applications tends to be higher than expected.

Table 4-2 – Number of distinct applications used by participants in work time environments, divided by ranges.

Number of Applications	Response Percentage	Response Count
1-4	21.5%	11
5-10	47.1%	24
11+	31.4%	16

Against this, **the number of work-specific applications** is shown in Table 4-3, which seems to indicate that slightly fewer applications are work-specific. There are some outliers which skewed the results and comparisons which have been removed. They may be anomalous results because certain people have unusual working patterns or because the questionnaire was incorrectly filled out. While these individuals may provide valid information, they will skew the statistical analysis, and as they may be mistakes in entry. Having excluded these, and comparing the actual results between these two, there is a mean of 70% of applications used are work only.

Table 4-3 – Number of work specific applications used in work time environments.

Work Specific Applications	Response Percentage	Response Count
0	17.6%	9
1	5.9%	3
2	5.9%	3
3	19.6%	10
4	21.6%	11
5+	29.4%	15

The **time used per day for the most common application** is shown in Table 4-4 which shows that a substantial proportion of people use their most common application for less than half of the day. However, there is a significant proportion (18.2%) of respondents who use the applications for 6-9 hours, implying that they are using this all day. These are very heavy users, spending their working day permanently on a single application. In the analysis below, these will be referred to as dedicated users.

Table 4-4 – Hours spent per day on most used computer application in work time.

How long per day	Response Percentage	Response Count
1-3 hours	54.5%	24
4-5 hours	27.3%	12
6-9 hours	18.2%	8
more than 9 hours	0.0%	0

On the question of **whether the application makes the performance of the work easier of harder**, 95.5% indicated that it made it easier. The comments indicated that they “could not do their work without it” and that the work tasks were very strongly focussed around the computer applications. The two responses that it made it harder included comments suggesting that other people using computer applications rather than the actual participants were driving this usage. This is clearly indicative of work-like usage.

Leisure questions

The numbers who made it through to the leisure section was reduced to 41, which is still sufficient to give these results validity. The questions mirror the work questions but with important variations.

How long do people spend on leisure computer use? Not surprisingly, the answers as shown in Table 4-5 are less than for work usage but still a reasonable amount of time spent. 12 respondents spent 4-8 hours of their leisure time per day on computer applications. This is a considerable portion of available time for people who are working reasonable hours (a requirement of participating).

Table 4-5 – Reported portion of leisure time spent per day on computer applications.

Answer Options	Response Percent	Response Count
0-3 hours	68.3%	28
4-8 hours	29.3%	12
9-14 hours	2.4%	1
more than 14 hours	0.0%	0

The number of different applications used in a week is also significant, and the expectation was it would be substantially reduced (maybe an email client, Facebook or similar, and a word processor). However, the results in Table 4-6 show that many are using considerably more than this. In fact, nearly half are using six or more applications on a regular basis, which implies that this usage is probably more complex or intense than simple email checking or similar. However, there are some who use just a few and so may be performing basic tasks. We will see later if the time and the number of applications show any correlation.

Table 4-6 – Number of different applications used in leisure activities.

Answer Options	Response Percent	Response Count
1	7.3%	3
2	4.9%	2
3	12.2%	5
4	17.1%	7
5	9.8%	4
6+	48.8%	20

For the **most common application, how long are people using it** for? This is shown in Table 4-7, where most people are spending around an hour a day on their main application, which is a reasonable amount of time. Dedicating 10 hours a week to this activity is a significant investment, and, in terms of the usage of the application, suggests that it needs to be well designed and built, to make that time as productive and useful as possible.

Table 4-7 – Total time spent per week on most used computer application for leisure usage.

Answer Options	Response Percent	Response Count
1-5 hours	41.5%	17
6-10 hours	24.4%	10
11-20 hours	17.1%	7
21-35 hours	14.6%	6
35-50 hours	2.4%	1
more than 50 hours	0.0%	0

The final question is intended to ascertain whether this usage matches a profile of “leisure use” or whether there are significant external drivers for the use a particular software application. As we see in Table 4-8, the vast majority are using software out of personal choice, not from external pressure, and even those who are not tend to go with a simple choice rather than having a particular package forced on them.

Table 4-8 – Why do you use this particular software application, for leisure usage.

Answer Options	Response Percent	Response Count
Personal choice	78.0%	32
Requirements from other people (i.e. members of a club)	0.0%	0
Software that came with the computer	12.2%	5
Application that someone recommended	7.3%	3
Saw it advertised, or downloaded it when I needed something	2.4%	1
Other (please specify)	0.0%	0

Time spent with number of applications

The first exploration is to analyse the relationship between the time spent and the number of applications used, for work and for leisure. The raw data for working applications is shown in Table A1-50 where the proportion of time is a mean of the ranges selected. The number of applications is also set to 7 where
5-10 was selected as a mean value. When applying a Pearson Correlation to this data, we find a coefficient of 0.447, which is significant at the 0.01 level (2-tailed). So we find there is a positive correlation here: that the more time the participants spend with computers in a work environment, the more applications they use.

Attempting a curve fit on this data, to analyse what the correlation is gives us the results in Figure 4-2. The Quadratic and Inverse curves provided the best fits against this data, with R^2 values of 0.25 and 0.28. What does become clear from this analysis is that there is an increase – a degree of correlation – but at the higher levels this breaks down. So those who spend a lot of time using computer systems are more variable in terms of their application count than those who use only a few.

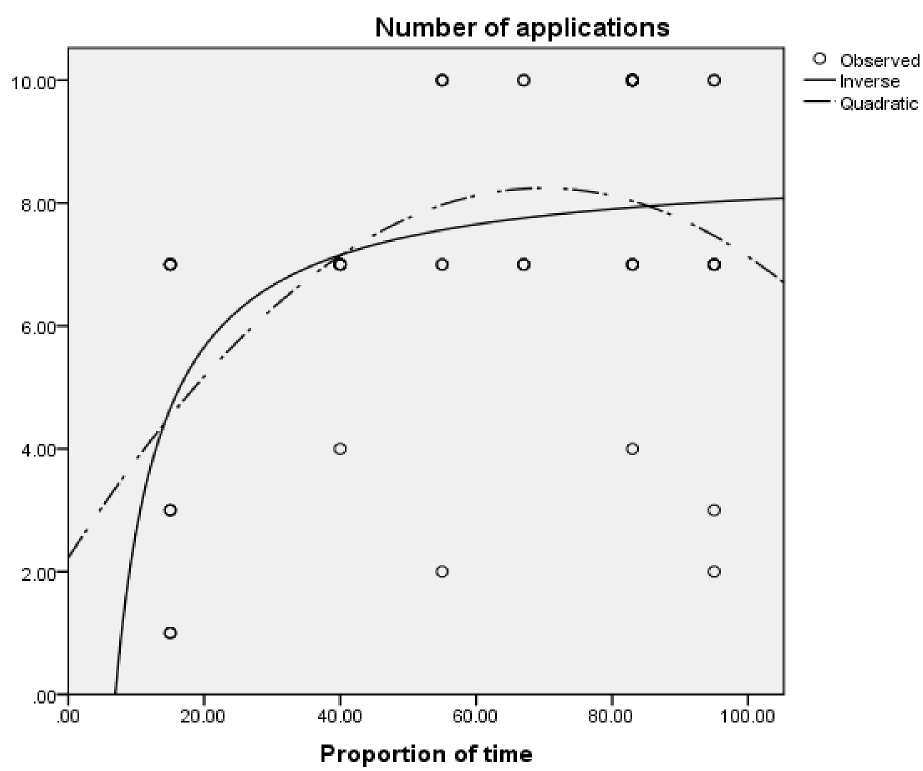


Figure 4-2 - Curve fit for Inverse and Quadratic correlations of Time spent in applications against the number of applications for work situations.

There are four broad categories:

1. People who spend very little time using computers, who (because they use less than the mean number of applications) appear to be using just a few tools for doing their work, as required.
2. People who spend a moderate amount of their time using computers, who are adding more applications to consume more and more of their time. They are not just spending longer using the same tools but are making more use of computers in their work, utilising more and more tools.
3. People who spend a large amount of their time using computers and use a large number of different tools on them.
4. People who spend a lot of time on computers, but are using just a few tools on them. They are likely to be very heavily invested in their few tools, using them for a large portion of the day.

Next, we need to look at the same interpretation for the leisure results, and these are shown in

Table A1-51. Once again, a Pearson Correlation has a coefficient of 0.318, significant at the 0.05 level (2-tailed). We also find that the Kendal's Tau correlation coefficient of 0.312 and the Spearman's Rho correlation coefficient of 0.346 are both significant at the 0.05 level (2-tailed). This all indicates that these values are correlated, that the more time people spend using their computers in their leisure time, the more applications they use.

Attempting a curve fit on this data, to analyse what the correlation is gives us the results in Figure 4-3. The Quadratic and Logarithmic curves provided the best fits against this data, both with R^2 values of 0.1.

While it is clear that there is a lot less variation here, there are three broad categories of usage:

1. Those who use a lot of applications spend a lot of time using them. These are likely to be enthusiasts who use a variety of applications and make full use of their computers.

2. Those who use a lot of applications, but not for long. These are likely to be people who make good use of their computers, but for functional use, not much of the time.

3. Those who use very few applications and not for long. These are liable to be people simply checking email occasionally or similar.

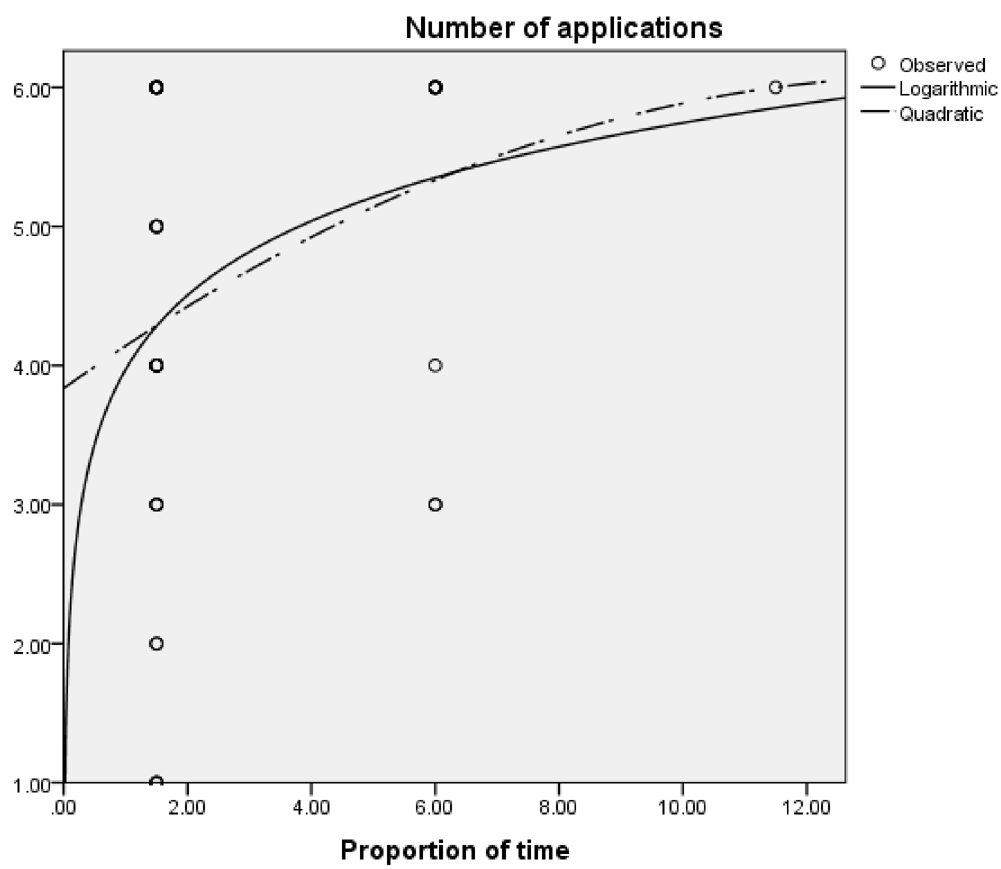


Figure 4-3 - Curve fit for Logarithmic and Quadratic correlations of Time spent in applications against the number of applications for leisure use.

It may also be informative to explore the comparison between the two sets of results. Do those who spend more time in work environments using computers also spend more time in their leisure using computers? Attempting to correlate on the outcome variables produces no significance. A partial correlation also produces nothing significant. This would indicate that usage of computer systems in work and leisure environments are not related.

Proportion of work specific over total

The second area to explore correlations in is the proportion of work-specific applications over the total number of applications. This follows from the identification above that the participants spend significantly more time on work applications, so it is worth noting whether these are written for the workplace or not. This raw data is shown in Table A1-52. This has a significant correlation using Pearson, with a value of 0.386, significant at the 0.05 level. Looking at the curve fit, we find an s-curve provides the best fit (see Figure 4-4).

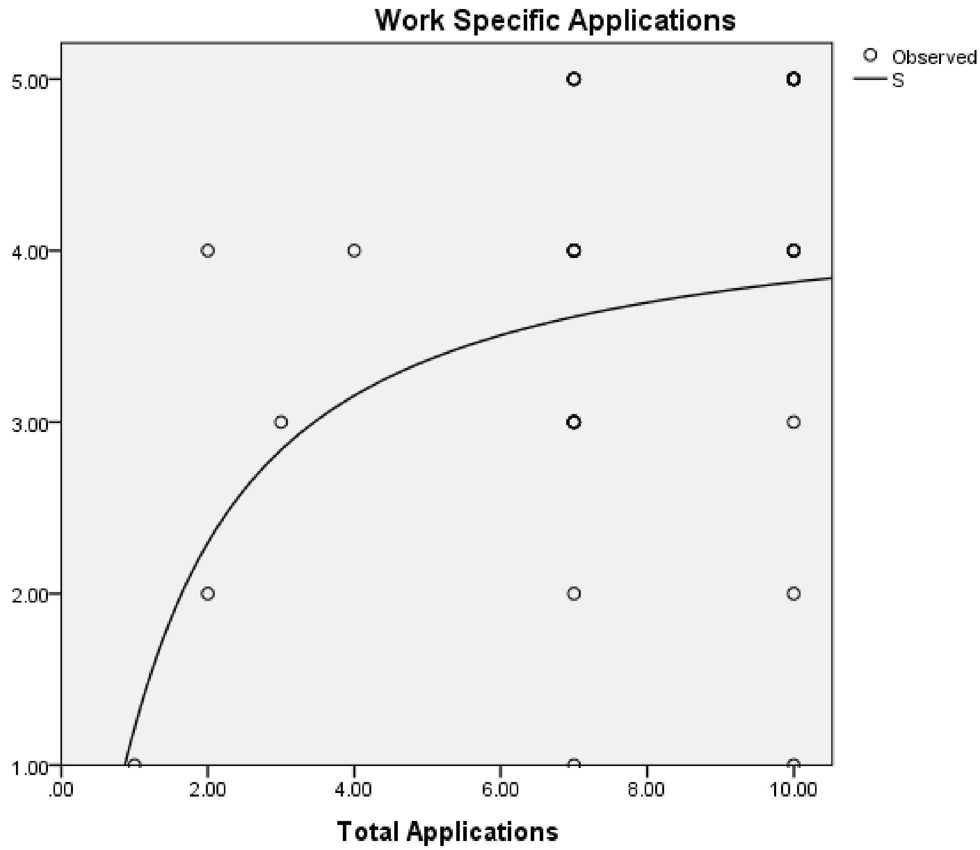


Figure 4-4 - S-Curve fit of total applications against work specific applications in the work environment.

The fit is not particularly good, especially at the higher levels. But it gives some indication that at the lower end, people may be forced to use more work-specific applications, whereas at the higher levels, where participants are using more applications, they are then using them for more general usage, or in ways that are not purely work-related.

If we adjust the values to reflect non-work applications and correlate these against the total applications, we do find a Pearson correlation of 0.865, which is very strong. Applying the curve fit (see Figure 4-5), we find the correlation appears to be very much linear. In other words, the extra applications people are using are largely non-work-specific tools.

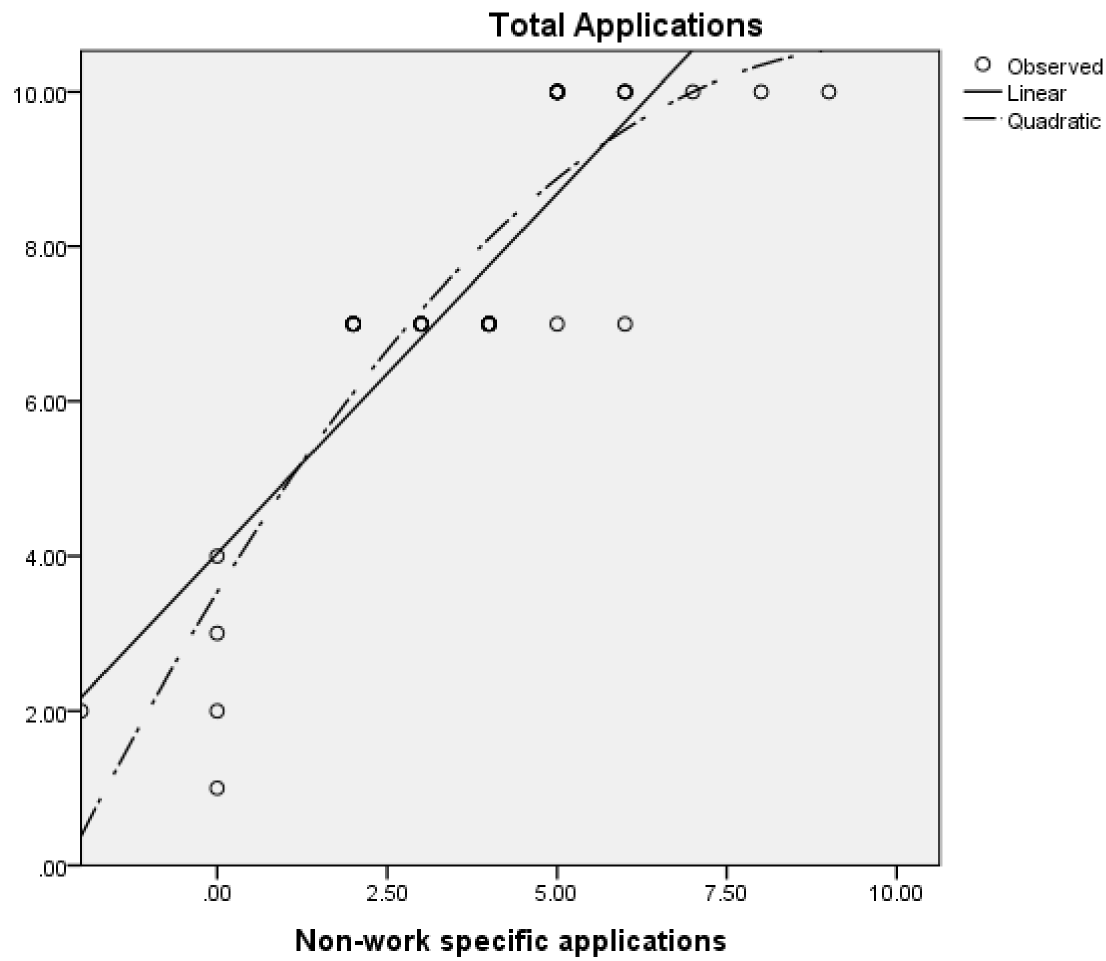


Figure 4-5 - Linear and Quadratic curve fits of total applications against non-work specific applications in the work environment.

Many of those participating are clearly using a significant portion of work-driven applications. While it would be interesting to understand how these are used, the questionnaire does not provide that information, and it is difficult to draw any speculative conclusions. However, a mixture analysis of the results does indicate that the figures cluster around the one third to one half area, indicating that a significant portion of applications for many people are not work specific. What we find from the inverse analysis is that the extra applications are non-work tools. We could surmise that participants are starting by using work applications and then finding more tools that they can use. However, without more information, we cannot identify the meaning of this correlation.

<u>Time spent on most common application as proportion of total time</u>

How much of the total work time is spent on the most common applications? This indicates whether the time was spent on a few or many applications. The raw data of the working times are shown in

Table A1-53. The hours per day are calculated by taking the proportion of the time the respondents indicated and recording this as a proportion of an 8-hour day. This is only an estimate but provides a valid baseline. Of these, 16 record a value of over 1, indicating that the approximations have not worked here and that a quarter of the respondents spend a very substantial proportion of their time on their main application. Applying a correlation test to this data, we find a Pearson correlation of 0.404, which is significant at the 0.01 level (two tailed). The Kendal's Tau correlation is at 0.366, and the Spearman's rho at 0.416, all indicating that the data is correlated.

The curve fit for this data, however, doesn't give a clear indication of the nature of the correlation. There is a slight tendency for longer days to be reflected in longer time on the main application, but this is not decisive. Some people will spend their entire day on their main applications, others hardly any time.

For leisure use, the raw data is shown in Table A1-54. The correlations here are again strong – Pearson correlation of 0.607, which is significant at the 0.01 level. However, once again, there is no clear curve fit, mainly because the data is distinct, not continuous.

Comparing these two sets of figures, we find no correlation between the work and leisure times meaning that (as we saw before) participants have no connection between their work and leisure usage.

Number of applications, and reasons for usage.

The final comparison is between the number of applications used for work and leisure. The raw data for this is shown in Table A1-55, and it is clear that the number of applications used in the workplace is higher than the number used in leisure time. The Pearson correlation is 0.501 (Significant at the 0.01 level, 1-tailed), Kendal's tau producing 0.434 and Spearman's rho 0.487. This is a high correlation, and we see a clear pattern in the curve fit (see Figure 4-6) that the increasing number of applications used in work environments are reflected in an increased number of applications used in leisure usage.

This is interesting as there were no other correlations between work and leisure usage, in terms of time spent. So participants varied in their time usage, but there is a clear connection in terms of applications, which may be an indication that participants get more use from their computers in leisure environments when they learn what they can do in their work environments.

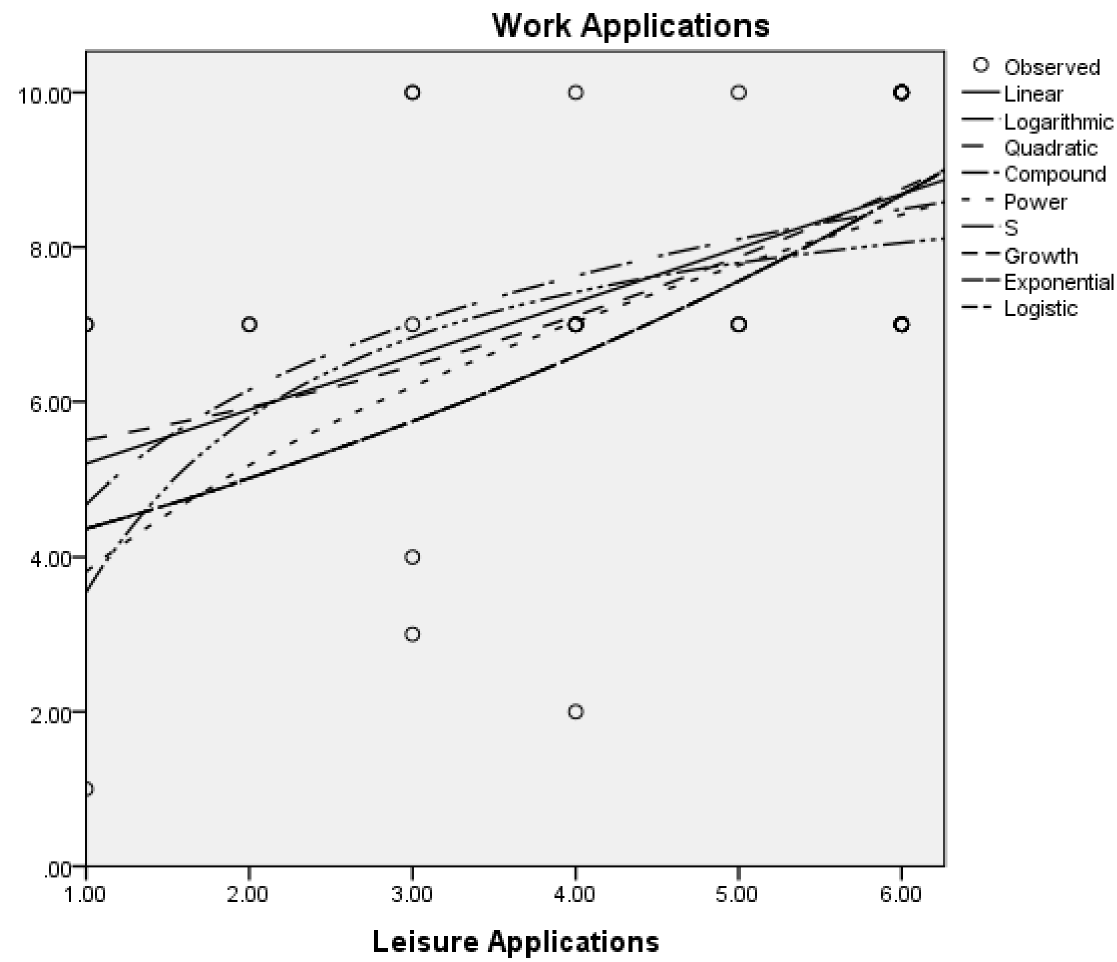


Figure 4-6 – Curve fitting for comparison of work and leisure application counts.

One final assessment of the data is to consider the relationship between the time spent and the reason for choosing. This is to see if people who choose their software spend longer on it. It may be because they are more knowledgeable or because they were able to make a choice, and so find something they were happy with. We see the results in Table 4-9, but there is no clear connection, partly due to the small number of respondents. There is no clear focus on personal choice, the highest value being that it came with the computer.

Table 4-9 – Mapping the mean of the time spent on computers with the reason for choosing.

Choice	Requirement	Came with computer	Recommended	Advertised
10.6	-	12.2	3.0	15.0

Discussion.

The first thing we can assert from this study is that the second hypothesis here has some support – that "**Using this flow/frustration dichotomy for interfaces definition will produce financial benefits for organisations**". There is clearly a spread of different usage, and within work and leisure environments, some people are using a few applications extensively, whereas others use a wider variety or use a few different ones. If we are looking at improving the user satisfaction and flow within the work environment, it is not sufficient to simply make one application work well, as most people were using a number of applications and they all need to work well together. The abandonment cost (See chapter 2) of tasks is high, but the usage of many different applications, some of which are not work specific, means that the standard is set very high (see Grönvall, Ciolfi et. al. 2016 as quoted in chapter 2). The blurring of work and leisure lines actually puts pressure on work-specific applications to match up in terms of usability with leisure applications. However, it is also important to make individual applications highly usable, as some people are spending considerable time on individual applications. Making the working environment a more flow-friendly one is a large challenge, and a holistic approach is needed to progress this. The insights into perceptions above show that the entire environment is important. We don't simply see one specific place, we experience the entire environment, so if one part of the environment is distracting, it will impact our entire perception and so our entire experience.

In the working environment, participants spend considerably more time using applications than in their leisure time. That makes sense because people spend a lot longer at work than at leisure. It does confirm that the work environment is a vital one for many people and that computer systems do play an important role in this environment. More, we see that around half the time is spent on work specific applications, so those systems that are written for business are being used substantially. It does make some sense to focus on the interaction with these as a route to improving the overall experience that people have with computer systems. We see from Activity Theory (Nardi, 1996 and Zubhof, 1998, both referenced in Chapter 2 above) that the overall experience is of crucial importance to the individual application experience. Respondents use more applications at work than in their leisure, so a focus on the entire computerised work environment will also have a significant impact.

We did identify correlations between the time spent and the number of applications, identifying a number of different usage patterns. However, the work and leisure usage was not correlated, so there is nothing we can draw from this comparison. On the other hand, we can identify that work and leisure usage is distinct and different, and that particular work patterns don't then relate to leisure patterns. It justifies the hypothesis but also indicates that when studies do assessments of computer usage, is it important to identify whether work-type or leisure-type environments are being considered.

Following on from this, our next study will be looking at the positive and negative aspects of computer usage in general but also looking at any differences identified between work and leisure usage. If we can identify the **reasons** people become frustrated with computer applications, this serves as important groundwork for quantifying these problems, and this is what we explore in the next two studies.

Interviews

The previous study looked at one of the important distinctions made in the initial work, between work and leisure usage. Having identified some of the distinctions between these two, we now move to consider is the perception of the usage of software applications by actual users. As Moggridge (2006, p. 131) says, "Design ought to start from the problem people are having and also from ideals.", so it is important to assess these problems at an early stage. Can we identify the perceived frustrations and flow experiences and understand how people actually view their usage of software? This will also identify the categories and principles we will use to define these flow and frustration concepts. This is important so that they are not just based on theoretical ideas but on the views and considerations of software users. This is to attempt to demonstrate the first hypothesis "**The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems.**" To understand this, we need to identify the flow and frustration signifiers, and this is the first stage of defining these from user experiences.

If we can identify the types of area that users consider frustrating or flow-inducing, then we can use these to understand what the effects of these experiences are on user perspective. The frustration aspect is highlighted by Shneiderman (2002, p. 24, 25), which indicates that up to 10% of time may be spent on computer problems. These frustrations may be more significant for work tasks because of the external control over software decisions. Following Shneiderman again, the concept of made-for-user software is software that is made for people to actually use, and made-to-go-wrong software will, one assumes, frustrate users.

Method

The purpose of these interviews is to identify those issues that the participants identified with using computer systems. The questions were very open-ended, aiming to understand and record general perceptions of usage, and utilise these responses to formalize the potential categories that we might use.

Participants

7 interviews were conducted by the researcher between September 2008 and March 2009. They were drawn from the researcher's personal contacts.

It should be noted that the selection of participants was very much self-selection – those who responded to a request. The range of respondents was not defined to cover specific groups, but the comments below indicate the range of respondents.

There were a range of roles and positions across the participants. They had a cross-section of educational levels; included both male and female participants; one was American, the remainder British. Two of the volunteers were disabled in some way which impacted their computer usage, and they provided interesting insights into issues that may not explicitly affect everyone but touch on the edges of usability questions. One participant had RSI, meaning that long-term typing was not possible. Some participants had a lot of experience with computer systems, whereas one described themselves as a "technophobe".

Their working roles included a business woman who is usually on the road or at client offices; a mother and housewife who uses a computer for leisure activities, including a lot of social interaction; and a teacher. They all use computers within their life somewhere, either substantially for their work or for a significant amount in their leisure activities. Some of them use computers at work and some at home.

The number of participants is not high, but the intention was to get a few insights from individuals to give some pointers to possible issues and positives, to help form ideas for the next stage. Statistical analysis of the volunteer attributes is limited because the number of participants is low, but the **individual** information is important rather than statistical summary assessment across the participants.

Materials and Procedure

There were two starter questions, asked to all participants: "What do you find good in computer applications?" and "What do you find bad in computer applications?". The term "computer applications" is intended to cover a wide range of software tools, leaving the interpretation open to the interviewee. The terms "Good" and "bad" are intended as explicitly value-laden terms, again leaving the interviewee to define what their experience of computer usage is.

All of the other questions and exploration came out from these initial questions – identifying the reasons and the issues, the reasons behind the answers. So it was important to understand what participants were trying to do, to grasp why applications were good or bad and to understand what the real gaps for bad applications or the real solutions for good applications were. These open-ended questions were used to allow the participants to explore their own thoughts and not be guided by suggested responses.

The interviews took place in a range of locations, one reason that this is an exploratory pilot, as the environment is not well controlled. One was in a coffee shop, several were in the participants' homes, and one was done over an internet chat forum. The locations were chosen for the convenience of the participants and in places where they felt comfortable to talk. The timing of the interviews was open-ended but took around 45-60 minutes. The focus was on the specific problems that individuals had, with a range of products referred to. The specific products may not have been widely used or the problems of wide significance, but they were issues that these individuals experienced. Similarly, the positives were areas that the specific participants found significant for them. The importance of these interviews was identifying the issues that caused these individuals problems.

After an introduction to the discussion, the core question of "What was bad about your applications?" was asked. This preceded the positives to allow participants to explore their frustrations first and to leave the interview on a positive note. The terms "good" and "bad" were chosen deliberately as emotional but non-specific terms – the question was not about whether something worked or not but about experiences that were positive and negative. It left a chance for the volunteers to explore for themselves what they interpret as "good" and "bad" because these words are more likely to produce responses based on experiences of products and systems than words like "what worked?" or "what did you enjoy?" The aim was to identify usage experience rather than application functionality – that is, whether the user experience was a positive or negative one rather than whether the requirement was eventually achieved.

One of the standard follow-up questions used when volunteers indicted that they had bad experiences was where they went for help and advice. This was asked to try to identify whether the issues were actually usability issues or whether they were problems with understanding what they were trying to do. This was attempting to ensure that the problems were specifically with the computer system rather than the task being attempted.

Results

To analyse this, the three questions asked and responded to were used as the core categories: what is good, what is bad and what did you do to resolve problems. Each interview was transcribed manually, read through, and all comments relating to the good or bad experiences noted. All points where interviewees mentioned **why** they liked or disliked products a note was taken, relating it (where possible) to previous mentions of a similar reason. Where possible, specific products were kept out of the assessment, although in some cases this was unavoidable – these references tended to be marginal, largely because they are too specific. Additionally, the purpose of this study is not to look at specific products but to attempt to identify some rules or principles for general application development.

The units of analysis were “utterances”, identified as a set of comments related to each other. For the face-to-face interviews, this was indicated where one topic ended and another started, although the nature of such conversations means that this was not always a clear and definitive break. In some cases, the discussions overlapped, so decisions were made about assigning comments to one or the other. For the messaging interviews, identifying the “utterances” was more complex, as the nature of the conversations was even more interconnected. The same type of process was undertaken of separating out one set of comments (from both sides) about one product or set of experiences and treating these as an utterance.

The context of the responses was noted, not to include in this analysis but to ensure that the responses were properly categorised. The texts are what Krippendorff

(2004, p. 63) calls “Social Interaction” rather than “Interview Transcripts”, which changes the nature of the texts (requiring more careful assessment of the context) but not the value. The insights achieved are as valuable despite them not being more formal interviews with a clearly defined structure. The challenge is to spend more time on the texts to give more careful consideration to them rather than simply using word counting.

Because of this, the analysis has been done by manual assessment of each interview, not using automated techniques. Each interview transcript was read through with any comments made about the usability of software noted. For each comment noted, the context was identified and the potential relationship to other comments made. The factors reported below did come out straightforwardly, with many of the interviews following a similar approach and with similar comments made. The wider context of the comments and the information in the interviews is not recorded in this content analysis but has influenced the further development within this project because they give more subtle clues as to the real issues that the volunteers were experiencing. This content analysis is simply trying to identify some of the clearer signposts from this initial work.

The results section here will deal with the general assessment and comments of the interviewers to assess these general views. The more detailed consideration of the three questions – what is good, what is bad and what people did to find help – will be considered in the discussion section below.

<u>General Comments</u>

The first comment that appeared in a number of the interviews in various forms was a question of training or learning the usage of the various software systems being an issue. Some felt that they had not been trained in the usage of a particular tool and so did not feel that they were getting the most out of it in their working with it. This aspect was most significant with work-based software but also featured in personal or leisure tools. This indicates that sometimes people find ways of doing things with software tools that may not be the best because they have never been taught the “right” way to do them. They may be doing it right, or they may not (and may be wasting a lot of time by this), but they have no idea because they know too little about the capabilities and expectations.

There is, within these answers, an implication that “complicated” software requires some training to use. Of course, this may be a tautology, in that software is defined as “complicated” if it requires training and straightforward if it doesn’t There is an implication within this that some form of formalised training is a valid and important part of larger software products and systems. Some user experience practitioners argue that this is mistaken — some time to familiarise oneself with software may be needed. However, if it is not intuitive and able to be self-taught to a level that gives the user confidence in what they are doing, then this may be a fundamental fault with the software system, not an inherent part of complex applications. Some of the software that is defined as “straightforward” may be as complicated and powerful as other systems. Because it is well designed from a User Experience perspective, and the complexity hidden, it is seen as a “simple” or intuitive system. A number of those interviewed used the word “intuitive” as a positive indication of usability for a system.

Complexity involving more options and decisions actually makes us less able to make a choice (Weinschenk, 2009, pp. 50-52). In fact, if we have just a few choices, we are more engaged and more likely to make and pursue a decision through. This is, of course, more important when we are looking at optional applications, but it is relevant for mandatory applications as well because it is important that software users have confidence in their decisions. In a business situation, it is important that users can have confidence in their decisions and be able provide easy justification for them.

One volunteer referred to training on software in a very positive manner. This person was a teacher who was using a package designed to assist in IT instruction. The process of learning the package from scratch, and therefore understanding how the students would learn it, was vital. For them, the process of learning the software was intimately connected with lesson preparation, and the simplicity of learning the package from both his perspective and his students was very important. Implicit in this was the fact that he felt his students would be able to learn the package safely and easily and would probably exceed his understanding of it. The package

encouraged learning about it and experimenting with it in a way that was vital for this particular environment.

The second general factor that came out from a number of the interviews in various sometimes subtle ways was the significance of problems experienced with the software systems used. What this shows is that perception of problems or issues does not necessarily relate directly to the scale of the problems. Software that has errors will not necessarily be rejected, dependant substantially on the nature and location of the issues found.

The location of faults was important. If there were some errors in parts of a package or system that were not often used and were not part of the core functionality, then they were viewed as less significant than core issues. What is defined as “core” and “marginal” may differ from person to person because if a package is bought for a specific purpose which it cannot achieve, then this functionality is core for that purchaser, whereas it may be marginal for others who utilise other features of the package. One example mentioned was an image manipulation program that worked perfectly well, except that it did not have the ability to import images, meaning that all of the manipulation was only available on existing images in the system. This meant that while the manipulation features may have been astounding, the application was useless for this participant’s requirements because they involve manipulation of personal images.

There seemed to be an assumption (possibly an unconscious one) that software was tested in the same way that it was used. This means that if the software does not do the “basic” tasks (something that is actually different for different users because it depends on what the expectations are), then the chances of it being able to do the more complex ones is remote. The truth is that often, the more complex features are thoroughly tested, but some of the basic UI is not. These features will never be used by someone who decides that they do not trust the software.

A second issue that mitigated in favour of a package was how problems identified could be fixed. In general, if readily available updates or workarounds were available, then this was considered less of a problem. One participant indicated that they had found problems, but there had been workarounds on the web site, so this issue could be resolved for them. Also, if advice from real people or web sites was clearly available to assist progress, then this was a positive factor. Assistance in the usage of the software was generally a positive factor. An implication of this is that the software producer has to acknowledge problems and support users progressing resolutions, whether through free updates or advice services. Admitting to problems would appear to be a better strategy than ignoring them, and it is worth noting (as we will see later) that solutions were often looked up on the internet using a search engine rather than directly to the manufacturer's web site, meaning that the manufacturer has no control over what is being said. If they are perceived as being unhelpful, then a web search will, in all likelihood, produce some of the negative comments as well as resolutions. The user perception of the software and the manufacturer can be significantly affected by this sort of insight, brought about by problems not being readily solved.

Thirdly, if the impact of problems was significantly outweighed by the advantages of the software, then the software was generally more acceptable according to the participants, even though the problems and errors may appear to be quite substantial. So if the software is generally very good, provides some significant and useful features, even if more advanced facets do not completely work, then the application still gives significant benefit and the problems are accepted. Applications that actually do a few things well and reliably are more acceptable than applications that try to do a whole lot more but do none of it particularly well.

Good software experiences.

The first reason identified by a number of the interviewees was that the software application **worked when needed**. Software applications that are work-related need to work at the point when they are required because waiting for a day or so is not viable. This means that the software needs to work in disconnected mode if appropriate (if it needs a network connection but one is not available), it needs to work on various platforms (for example, all current versions of Windows, Mac, and Linux-based systems – "Some of the online jobsites don't work with Safari [the Mac browser] or don't allow you to edit resumes or cover letters"), and it needs to work consistently across these. This requirement was also valid for leisure software, but the frustration level was not as high as when work deadlines were involved.

One interviewee had to use their computer with voice software, and so providing full functionality through this medium was also important ("When writing [dictating] an email, I usually end up writing it in Word and then copying and pasting because Word is so much better [than the email system]."). Software that just worked and handled the back-end processing automatically was considered to be good and usable. On a related note, software that handled formats across different versions was also appreciated because, in a similar way, they are back-end processing and should be invisible to the software user. This principle is reflected in other feedback too, that the software should be handling its own processing and not involve the front-end user in this. So, **working when needed** includes the idea of interacting with other software properly and working without exceptional interaction.

A related area was the comments that a number of people made that the application **focussed on the data**, the information, and they were "quick", "simple", and "straightforward" to use. In context, these comments were about the focus being on the action, the task, the process rather than the use of the application ("It is clear and easy to add content", "A lot of detail is quickly added"). For users of applications who do not consider themselves technically adept, they expressed that they didn't need to know about the internal workings or the processing of the data. They wanted it to relate to their tasks and the functions that they were trying to perform. Exactly what this meant did vary because the tasks that people were seeking to achieve were different, but the essence from the further explorations was about focussing on the information and the task not the application.

In one case this meant that it worked while disconnected, processed work to the server when connected, and remembered what was been done previously ("I do my work on the laptop, and when I reconnect to the network, it updates automatically"). In another, it meant that it supported the process of teaching a Systems course to Junior School pupils, so the focus was on the teaching process, not the application. This issue is all about the software being as close to invisible as possible while understanding what the user requires. Updating back onto the network is important for business users who often work remotely and only synchronise periodically. This needs to be done automatically because the software knows or can identify the changes far better than the user remembering. Teachers need to spend their time teaching, not trying to get software working. Their focus within their work environment is on getting their job done.

A third aspect that also relates to the same aspects was that the applications were **fit for purpose** and enhanced the process. Google was mentioned because it enables the users to find a whole lot of information (everything on the internet, in fact) simply and without fuss. It was mentioned without explanation because everyone knew about it and everyone knew what it did. It is notable that the phrase “to Google” has entered the language (see Wikipedia, 2002) because it is so ubiquitous, so much a part of the language of internet users (“I sourced a tray for a Tripp trap chair, made in Sweden but reviewed on a Canadian website … I found a supplier in England”). Another interviewee mentioned a route-finding application that she uses regularly because she has to visit people in their homes all around the area, and again, this does the job she needs it to, without any fuss. Once again, it performs when needed, provides just the information that is wanted, and nothing else. The concept of **fit for purpose** relates not just to the ability to do what is wanted but also to do this straightforwardly and without the application getting in the way. As Olsen (2012, p. 7) states, “As professionals, we are not interested in the tools *per se*, but in how we can use them to perform tasks.” He relates it to software tools, but the principles apply to any software application that is designed to assist in a task.

The issue of **features on applications** will come up again in the next section, but comments came up as positives where features were provided that were simple to define but enhanced the functionality of the product in the context of what it was meant to do. The features provided were helpful and enhanced the usage of the application, the task that the application was there to enhance. It should be noted that these “killer functions” were tasks that were a natural part of the task, simply done within the application quicker or easier than they could be done without it. This means that the software application was clearly and definitively enhancing the ability of the user to achieve their overall task, according to the respondents. One example was the teacher, who found that the flexibility of the software was significant in making it easy to use for the complex and flexible task of teaching others: he commented that it was easy for his pupils to pick up, and so stretch it beyond what he had thought of doing.

Other comments made include the importance of software developing, adapting to the requirements and capabilities of user and computers. In particular, the comment was, “It is much better than it was 15 years ago” – with the implication that they had used various versions of the software over a number of years. Sticking with it does suggest that it fulfilled their needs over time. Another comment made by a few interviewees was that WYSIWYG functionality (meaning that what was seen in the processing was what was seen in hard copy) was an important part of a good software application. Once again, this is about being able to focus on the results, not the process. If a user wants to create a document, they need to be able to focus on the document, not the word processor.

There is a place to consider upgrade processes, why people take or don’t take upgrades to software. According to Shove, Watson, Hand et. al., (2007, p. 23) it is not just a “consumerist” drive (people just wanting the latest product). Rather there is a sociological desire for change and improvement, people want something that does the job better. Focussing on doing things better may be less exciting than clever new features but may prove a more positive upgrade path.

Bad software experiences

There was a lot more discussion and comment on the problems than the good aspects. Two people made comments similar to “where do I start?” when asked about problems. This indicates that problems are very much what is expected of software (although see the note above on problems and their impact). However, there were two factors that were reported more often than any others.

Firstly, the **problem of incompatibility** of different versions of software was raised a number of times. This was either between home and work computers or between differing home computers or even between different people emailing documents within a work environment. In one case, it was a secretary who sent documents to people across the company (some nearby, others a long way away) who had a different version of the software. This participant had to remember (rather than being prompted) to save the right version, otherwise it caused problems and needed to be redone. While the task of redoing the file is not significant, when the files are transferred by email or memory stick and there is the additional task of alerting the sender to the problem and resending a copy or of having to wait another day as they go home to resave the file, the resolution to this problem becomes substantial. What is also relevant is that when this happens once or twice, or only once a month, it is just something to work with and ignore. When these types of problem occur daily or weekly, it is a major problem. Even though the impact may not be clearly seen in terms of resources and time, it is a major problem in terms of user satisfaction with the software.

A related issue is the **differing functionality across versions** of software. This was particularly raised where people use one version of software at work, and another at home; they get used to the functionality in one version, which makes it difficult and slow when switching to another version. In some cases, this was when an upgrade was done, and the functionality (or the process to access the functionality) had changed. The difficulty is seen as needing to re-learn the process, something that may be quite a challenge (see below). This was evidenced in these interviews for software that is used across home and work usage (or example, word processors) where a version can be significantly different in terms of user interface. This is the case even if (sometimes especially if) the new functionality is more powerful than the previous one. It is so easy to see this from a software development perspective that providing more functions or better ways of doing things is often seen as an improvement; however, from the perspective of a user of the software, if they have already learnt how to do something, changing this should be done very carefully. If it is made far more intuitive but based on the existing process, then this may be a positive development, even if the original process is flawed. Changing to a new process may be received and seen as a poor development. It is possible that usability is more important with upgrades than with new features (although new features should also be usable). One interviewee highlighted the problem as poor communication with the upgrades, resulting in documents being sent out to clients with amendments visible because defaults had changed. The fact of this change had not been communicated either by the IT department or the software. This was a changed process that was not clear or obvious, resulting in problems that appear trivial but were a major embarrassment for this participant and their employers.

The second factor raised by a number of people was that **applications were too complex**, had too many functions that were not useful or appropriate. This is the contrary aspect to applications that were “fit for purpose” (a phrase three of the interviewees used), that did one thing well. Applications that were trying to do everything tended to fail because they were unable to do the important things sufficiently well. While lacking functionality does not seem to be clearly related to extra functionality, the core problem was the same, that the application did not do its core requirement properly because it was trying to do too many other things. A number of authors comment (in passing) on the problem of phones that become so complicated that the prime task of making a phone call becomes complex. The same problem can impact other software, where the core task becomes lost within peripheral processes. While the peripheral processes may be excellent, the impact on the core functionality means that the software application as a whole deteriorates from the perspective of the user.

A further aspect of the “too complex” response was that processes took too many steps to complete. This is not an absolute count but a sense that the number of steps to complete a process was more than the participants felt the process “should” take. This is really about the relationship between the task process and the application process. When these match well, this is acceptable; when the application takes fewer steps, this is where it is seen as being helpful and really positive. Where it has more, especially significantly more, a very negative perception is gained of the application. We will be looking further into some of what comprises a “step” later in this study.

The next aspect raised by interviewees was where company rules that ignore technical issues for **conformity** or **standardisation** and enforce upgrades without considering the implications of the upgrade for those who use the applications. The problem here is that assumptions are made about the usage of applications on a global/corporate level that are not necessarily reflected in the actual use of these applications at a personal level. Companies make corporate level decisions that a new package or upgrade does everything needed and there are no ill effects, based on an incomplete picture of what is used and needed across the company. This means that the impact of the changes is not fully taken into account.

The fourth major issue was software that was being **overly helpful**, making assumptions that are unwarranted, interrupting the participants' work flow by trying to assist then in doing the task. Also included here are features that automatically adjust the working environment, modifying the processing actions to perform a task. Applications that are flexible in their configuration for use are viewed positively, and applications that learn or adjust based on user activity are also good. Software that doesn't remember what the user was doing or did last time is frustrating, meaning that a balance between understanding and assuming needs to be found.

This issue emphasises the problems in trying to make software responsive to the user without being either too proscriptive or too "cute" (as one interviewee put it) and "chatty". Appreciating when a user wants help with their task is a more complex challenge than appears to be demonstrated by most of the software mentioned. Doing this in a rule-driven way is often a problem because the software appears to interact with the user like a mindless person. There is some indication that it might be appreciated, but it needs to be managed in a careful way; it needs to engage with the user, not alienate them. If it is to be present, it needs to enable interactions like a thinking and considerate person, who knows what you might want but accepts that you may want something else this time. There is a sense that, because these types of learning and intelligent interaction are difficult and complex, software designers do not bother to include them. However, all users really want what Cooper (2004, p. 162ff) calls "Polite software", which he defines in 14 specific requirements. These requirements do broadly match the participants' comments.

Two other aspects were noted by 3 interviewees and can be combined into one. The first was **perceived slowness of applications**, and the second was **having to remember or write down information** during the processing. The speed of an application is an important issue, but it was the **perceived** speed or performance of the application that was actually being highlighted. In the first case where this was mentioned, the real problem occurred because the application slowed down at one point in the day (because of backup work being done). This meant that the speed was unreliable and unpredictable, resulting in the perceived performance deterioration even if the absolute performance was still within acceptable limits. The issue with this is that the user may have expectations (either of their own or set by other people) which they then cannot meet. As we will see later, delayed or interrupted processing can have proportionally larger effects on the completion of tasks. A system being twice as slow at a crucial time may result in certain tasks taking far more than twice as long to achieve.

Having to write information down is sometimes caused by applications performing too slowly and not providing sufficient feedback or information. The time that information is required to be remembered is beyond a reasonable time for the human brain to remember it, so it has to be written. Another reason is that information is not properly passed between different parts of an application, so it needs to be noted outside of the application to be re-entered somewhere else. In both these cases, the use of memory or paper to do things that the computer should be doing is frustrating and irritating and results in a negative opinion. In fact, this applies in a wider way, that if the computer is not doing those things which it is expected to be better at (remembering, calculating), then the overall perception of the application is less favourable.

Resolutions

The third part of these responses is a short discussion of the ways that people use to resolve problems they find in their applications. The first way is to **use the internet** ("use Google" as most people said) to find whether anyone else has had similar issues (they normally have). This is interesting because it carries with it the implication that the use of one application (in fact, two: Google and a web browser) for finding the answers to problems with another application is a natural step for most participants. Most participants did indicate that they found the answers through a Google search, so it did what it was being asked to do, which is one of the positive aspects identified above. Of course, this does not apply to in-house written applications or very small-market work-specific applications, but for the mass-market software, this does seem to be the commonest solution.

The second way that people find answers is similar but in a different sphere, which is to **ask colleagues** or specialists (which is a narrowed-scope version of looking on Google), depending on who is available. This option was found to be useful and productive too, as the people asked were chosen to be those who could help. In both these ways the basic premise of asking someone else who has either experienced the problems or who knows the software well enough to be able to troubleshoot is pursued. This obviously does work if the software is in-house or very specific, as long as the user can be put in contact with someone who knows the software.

One final option for what to do when people hit a problem was to **move on to something else** – to do some other task. This is not always possible, of course, but avoiding the problems by changing the task is a perfectly valid option in some cases. From the perspective of the software provider it may not be so good: doing something else if an e-commerce site does not work sufficiently probably means a lost customer. In a business environment, it may mean that complex processes get delayed and the business suffers from this or that expected savings from new software are not realised. It may mean lost business or lost contracts, either for the business or the software suppliers or supporters. It is an interesting and very challenging answer to the problem but one which people are often prepared to take. If the software does not work suitably, people will not use it.

Discussion

What can we learn from these results relating to the core themes of flow and frustration? As an initial pilot study, there is very little statistical analysis that is available on these results alone, but it provides a starting point for further work. The statistical work will be done in the next study as this is expanded and progressed, and we can draw comparisons and utilize this information further.

The core question here, however, is whether this study provides initial evidence for the flow/frustration dichotomy thesis we are proposing. We have identified that there are features that the participants consider positive and those which they consider negative. At this point, we cannot definitively connect these to our core proposal. The comment noted above of "where do I start" with respect to the level of frustration suggests that this is seen as a core aspect and expectation of using computer systems.

There is an important insight here that solving problems on specific applications is just one part of a bigger solution. Ensuring that people's entire computer usage experience is positive involves improvement to the entire environment – physical hardware, networking and software applications. In simple terms, there is no point in having a fantastic web-fronted application if the web browsers available are problematic. There is no point in designing fantastic new software if the hardware needed to use it will frustrate the user. Perception of a computer system is not limited to a particular part of the package but to all of it, meaning that design and development needs to consider and address the whole environment.

Another aspect that caused frustration was the incompatibility of software between versions or environments. This is a frustration based on the learning curve required to properly utilize a particular piece of software in a particular environment. There is specific frustration here when that learning is made irrelevant for new situations. The user may have spent a long time learning the way to do their tasks in one situation or using one version of the program, and it is frustrating when they cannot transfer this. Introducing frustration into an upgrade process undermines trust in the software. The problem here is that users are being made to think, contra Krug (2000), to have to work out how to do something, whereas it should be "obvious" (whatever that means in context). There is frustration when learned obviousness is rendered obsolete.

The third aspect worth noting is the complexity problem. Once again, users are being made to think when they shouldn't have to. This is, significantly, about users reporting complexity experienced, not necessarily an objective measure of complexity. The tasks themselves may be complex or not; what is at issue here is the complexity of the computer usage to enhance the task. As with most of these observations, it is **perception** of complexity compared to perception of how complicated the computer enhancement of the task should be. This is a reflection of Baudrillard's insights (see the literature review above) that the perceived reality is what is important, that this becomes reality. Further, the exploration of perception above makes it clear that what our minds see is not necessarily the same as what is in front of us. This means that the individual is a part of the system, and so it is their experiences that are critical, not a purely objective interpretation of the application.

The issue of perception is crucial to these interviews. The nature of the question is about what participants find easy or difficult, something that might differ for different people. This reflects the prime thesis of this study, because flow is a different experience for each individual, as is frustration. When the tools "work when needed", and "[focus] on data", this means that they are assisting the task accomplishment, not intruding into it, thus potentially enhancing the flow experience. When the computer system demonstrates inconsistency or excessive complexity as perceived by the participant or is slower than the thought process of the participant, then they feel frustrated in their task fulfillment.

Having identified a set of factors that seem to reflect these positive or negative experiences – factors that will enhance flow or cause frustration – the next stage is to confirm these aspects by utilizing a different source of information to provide support for these categories as valid.

Confirmation from discussion boards.

From previous chapters, we have been able to identify some significant issues for users both positive and negative. The next study examines whether these categories hold up in a different but related environment. The purpose of this is to examine whether the findings from the previous study also hold in a different environment, identify the matches, and explore the possible reasons for mismatches.

If we see the same categories in a different context, we can conclude that they do hold some validity in terms of the reasons for positive or negative software choice. On the other hand, as this is a confirmatory study, if we do not find matches, we need to identify reasons for this and ascertain if there is some reason that in this new context, such factors will not be highlighted.

This study is a follow up to the previous one, taking the broad categories identified there (see below) and re-assessing them to see if these can be used to test the first hypothesis "**The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems.**"

Method

This study utilises the categories identified through the interviews in the previous study, detailed above. This investigation examines posts across forums on the internet in order to see if the categorisations appear to be valid in this different context. By passively observing users interacting with each other (rather than the researcher) it avoids any bias towards the categories in the questioning. The aim was to attempt to accurately reflect how people classify and explore their categorisation when they are not being asked to do so as part of a research project.

Participants

A total of 8 posting threads were identified, with 4-5 relevant postings per thread, making a total of 32 postings. They were made by a variety of people, none of whom had the same self-chosen monikers. While it is possible that some individuals posted under multiple monikers, this is considered unlikely.

None of the forums were places that the researcher frequented, so there was no prior knowledge of bias. This also means that there is less likely to be any overlap with the respondents in the previous study. Because the respondents are anonymous, there is nothing else we can identify about them.

Materials and apparatus

One of the big challenges in doing this comparison is to ensure that we can gather a comparable set of data, while asking subtly different questions (because the medium is different, so there are none of the extra clarification questions that can be used in the interviews). However, because of the nature of the original interviews, starting from two core questions, a similar approach can be taken for finding online forum postings.

This study is confirmatory, the core purpose being to identify whether the areas identified in the first study hold up or not, whether anything crucial has been missed or anything irrelevant has been included.

Procedure

The comments were taken as a one-off download on 26th May, 2009, based on the first few pages of results from Google.co.uk, from a search for “Good Software Application”. The resulting pages were then assessed, and only those with multiple comments on them were used, to ensure that there was some opportunity for discussion or debate. This means that when people have to argue for an application being good, we can see what sort of arguments they use.

A similar approach to the previous study was taken where each posting is treated as a unit, and where each reference to a positive or negative reason for using certain software was made. It is therefore trying to reflect as closely as possible the information style gathered in the previous study.

There is a degree of overlap, so in some cases postings are counted multiple times. It is noticeable that the information density was much higher in the posting than in the interviews, probably reflecting the fact that a person typing a response to a question will often focus more on the details because typing is not such an innate, natural ability as talking. Comments that did not fit in with the categorisations were noted separately and considered as to whether they reflect distinct categories or whether they are merely the result of the medium used.

Because this was confirmatory, a slightly different approach was taken to the analysis from the initial study. Copies of the analysis process are shown in Appendix 6.

Results

As with previous studies, the results are divided into good and bad software experiences. There were no general comments because of the nature of the data. Because the aim is to define categories, these specific comments are the most useful.

Good software experiences

The positive responses found in the data were analysed for conformity to the categories previously identified. This was done by identifying the number of responses (utterances) that fit into each category.

The first category was **worked when needed**, that the product did what was needed in the circumstances that it was expected to. A number of the responses featured this as a reason for the products being good. This is partly what is expected, given that a number of the questions are along the lines of "what is good software for [this task]", that the responses often came along the lines of "[this software] is good for [this task]", which obtained marks here. Some of the responses indicated that **working when needed** was significant, in that the request for advice was specific enough for a certain package to be relevant, when in a broader request this may not have been the case.

It may be more significant that the responses were not higher here – that there were not more responses indicating that certain software was the thing that worked in the specified situation. This is possibly because sometimes the questions were too broad, or the responses were seeking to give more general advice. It is also possible that when the question is turned around, as it often was, into "what software does this” rather than “what is good about this software doing this”, it results in multiple solutions, and other factors will play a part in the choice of a specific package.

The second category is **focussed on data**, that the software was focussed on the data or the process that it was dealing with, and did not force the user to understand the software interaction. Comments included **convenient** (the software would just work without technical knowledge) and **set and forget** (it can be run up and then ignored without having to constantly engage with it). In the particular situations, this was seen as a positive feature, although it may not always be the case to the same degree. However, the ability to simply install it or run it and then focus on the required task is clearly being built into some software and getting a good response for it.

The third category, and the one most with most references, was **fit for purpose.** This is a category indicating that the software does what it is supposed to do or claims to do. That this was the highest scoring does partly reflect the nature of the questions being asked but also reflects that many people see the requirement of doing the full range of facilities is more important than other aspects like working across platforms and environments. It is not that these are irrelevant, but if someone specifies “the best software for doing X on platform Y”, then the environment is clearly specified. The nature of these types of forums is that single-platform answers are normally the most useful because people seeking tools to work across multiple platforms will not use this as a means of identifying appropriate systems and applications. Some of the comments were that the software was “highly functional”, “easy to use” and “did everything needed”. These comments reflect a number of different factors in reality, but the context indicated that the software did what was expected. “Highly functional” is important for some types of software, where people need to be able to access significant functionality, as long as it is also “easy to use” or at least easy enough for someone who may need to use these features.

This category also covers comments that indicate the software was particularly good for what it was trying to do. Where the requirement was for high functionality, this was important. The high number of comments in this category reflects even more than a simple “worked when needed”. The answers were along the lines of “[this software] is good in [this task] in [this situation]", often reflecting individuals' usage and adaption to the best environment and setup for the best software. Sometime people have adapted to the best software by changing their environment and hardware and now recommend this configuration to others, who hopefully will not have to adapt so much. It raises the question rarely asked in business environments of what is most important. Is it finding a software solution that works on an existing platform, or finding the best software configuration for the job? The nature of computers as general-purpose tools may be impacting their ability to function well at specific roles and tasks.

The final category is **functional and adaptable.** This addresses whether the software was highly featured and could change and adapt to related requirements. The few comments here tended to reflect that the functions were right for the needs, so the software was **not too complex**. Some of the comments in the "fit for purpose" category are relevant here, that the high level of functionality must be accessible.

One comment was critical that certain software lacked more advanced features, indicating that features are important, and there are often sets of features that are "expected" of certain categories of software. It is a fine line that all software developers need to draw between providing too much functionality that makes full usage complex and not providing features that are often unspoken but expected.

Bad software experiences

There were fewer comments in the negative categories, reflecting the nature of the questions being asked. There is less enticement to criticise products where the question is being asked for recommendations. However, there were some, which probably (given these considerations) reflect strong views.

The first negative category was **incompatibility**, which had a number of comments. In particular, the inability to import or export from the application (where this is a relevant and appropriate requirement of the software) was criticised. It is very rare these days for personal software to be completely free-standing, generating its own data storing and using this. It is much more common for applications to have to load previous or external information. Whether this is image files, music or more bespoke interactions, the boundaries of a software package are less easily defined than they used to be.

Another comment was that the software was a **resource hog**, meaning that it was not appropriate for lower specification computers. This is a problem particularly where users may want to run the package on home computers, but it may also be an indication of more significant problems with the software development. This is another case where careful consideration of the wider hardware environment may need to be taken into account when software is being developed and recommended.

The second category identified in the initial study was **differing functionality**, where software has significant changes between versions. This category was not raised at all by the posters in this study, almost certainly because the questions were asking about new software for a function, not updates. The lack of comments here should not be taken as an indication that this category is not valid, just that it requires a different approach to identify it.

The third category was **too complex**, that the software was too difficult to use. One comment from these participants was that it was “impossible to use”, another that it was “unbearable”. While these are subjective comments, they are from people who have purchased the software and tried to use it without success. They are not providing detailed objective feedback for review, but subjective responses as to whether the software is able to allow them to do what the software was advertised as doing. It is significant that software has reached customers in a state that makes it unusable, and attracts these sorts of comments on a public forum. It is a clear demonstration that the ideas of user experience research have not yet worked their way into all areas of software development.

The final category is that the **software makes assumptions** about the user and tries to be overly helpful. Once again, there were no reports of this in the sample. It is not clear whether this reflects the difference in the data sources or whether this is not a valid category. The suspicion is that this is becoming less of a problem as software vendors are listening to users who indicate that this is not wanted.

<u>Comparison of the two assessments</u>

The core question for this study is whether it confirms or refutes the finding from the previous study. The tables below show the comparisons as percentages of the total responses for that medium. As the numbers of respondents are different, this provides a consistent comparison. Table 4-10 shows the comparison of the interview and the online positive assessments (correlation is significantly > 0.05). Only 2 of the categories have similar values (“Worked when needed” and “Functions”). The other two differ significantly, with the data focus being more significant for the interviews and the fitness for purpose more for the postings. The standard deviations are included in this as an indication of how wide apart the values are.

This discrepancy is not significant in terms of the validity of the settings. Both of the categories are represented, and the different media may well be reflecting a different focus for the question. It is significant that for the interviews, there was a chance to provide immediate feedback and follow-up questions, whereas in the forum posts, while this is possible, it is less likely. The purpose of the posts and the questions asked is often to decide why to buy a particular product rather than feedback on actual usage. Purchasing decisions may have different requirements to ongoing assessment.

The mean values are included because the overall challenge of these two studies was to identify categories that are significant for software users, so merging these two provides some insight into how significant they are over both populations. They all appear to be represented reasonably, the variation not being that significant given the number of participants. We can conclude from this that all of these categories are valid, although they will have differing significance to differing populations.

Table 4-10 – Comparison of categorised responses from interviews and forum posts, regarding positive software experiences. Percentage of total responses recorded.

	Interviews	Posts	Standard Deviation
Worked when needed	21.4	23.5	1.48
Focussed on data	42.9	23.5	13.72
Fit for purpose	14.3	35.3	14.85
Functions	21.4	17.6	2.69

When we look at the negative experiences and responses, we see the values in Table 4-11. Once again, analysis does not provide any real confidence of correlation between these values (correlation is significantly > 0.05). However, the relatively small number of responses makes this less significant, as there are no conclusions that can be drawn from the small amount of data.

The same applies to the lack of any responses for **different functionality** and **too assumptive** categories. This may be significant, and the low mean value for these, especially the latter, does indicate that they may be less significant than the other categories. The major difference between the Interview values and the mean values is the different functionality category, which seems to be more significant for the interviewees than the postings respondents. This probably reflects, once again, the difference in population. When considering purchasing software, the expected platform is usually known, whereas when a product has been bought, and is in use, then some of the incompatibilities may become noticeable and significant.

Table 4-11 - Comparison of categorised responses from interviews and forum posts, regarding negative software experiences. Percentage of total responses recorded.

	Interviews	Posts	Standard Deviation
Incompatibility	25.0	50.0	17.68
Different functionality	31.3	0.0	22.13
Too complex	25.0	50.0	17.68
Assumptive	18.8	0.0	13.29

Discussion

The crucial question for this study is whether this supports the results from the previous study. On the positive aspect, applications that are **straightforward** in terms of doing the task that they are intended for, that are **focussing on the data** and the information being utilised and not on the processing of the information, are liked in both studies.

On the other hand, the negative experiences focussed around applications that **needed attention** paid to them. This included where they changed between versions or the facilities provided were not implemented properly or were not necessary. The issue of applications **being too helpful** is very similar, in that they are intruding into the process, rather than assisting the management of the task. Focus on the task is good; distraction from the task is bad.

The two studies indicate that the categories identified seem to be valid ones and cover a good range of the issues. There is some support for these conclusions from other sources, for example Norman (2009, p. 65), where some of the incompatibility issues are considered. It is, however, worth noting that the negative issues have less support and need to be considered less definitive, although not less important. This study confirms the first hypothesis, that **The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems.** It further supports the categorization I have identified previously. These three studies also give some support to the third thesis, that **a frustration experience can have a statistically significant impact on a participant's wider experience**. While not explicit, some of the comments raised did indicate that problems have a wider impact.

This study is limited very much by the time constraints, and the findings are thereby limited. Additionally, the time since the study was conducted means that some of these results are out of date, and a new study, taking more recent discussions and sites, could produce different but also more reliable results. It is considered that the range of studies in total do provide valid insights for this research and an approach for any future similar studies.

The next stage of this research is to use these areas and issues in a wider survey to establish the usage of computer systems. This should identify the dichotomy more clearly, with re-usable tools. It will also serve to identify any issues with particular social categorization groups and provide more insight into the experience of computer usage.

Chapter 5 - Computer usage survey.

The previous studies have identified categories of positive and negative software usage experiences. Examples include software that was straightforward to use or that focused on the data (not the process) was viewed in a positive light, whereas software that needed attention or was seen as trying to be too invasively helpful was viewed negatively. This study will utilize this information and quantify the categories into specific questions (see below). This provides the basis for a broader questionnaire where the results can be assessed against various demographics. The aim of this study is to see whether there is a difference in user's perceptions of the flow experience and the frustration experience and whether that is dependent on reported academic ability, self-classified expertise or the area a subject works in.

The intention of this is to ensure that these categories can be used across all users of computer systems, with consideration for the differences in populations, and valid results obtained from them in support of the first hypothesis, that **The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems**. In particular, this survey is intended to ascertain whether other demographics are at play or whether there is any difference across demographics that might be of interest.

The previous surveys identified a number of areas that represent feedback about positive and negative reactions to computer systems. These have been quantified into a series of questions (see Table 5-18 and Table 5-19), which are used as the basis of the survey.

Method

Participants

A total of 107 people were invited to participate in the questionnaire, drawn from the researchers' friends, relatives, and clients, including social media contacts, with a request to pass the link on to others. Of these, 63 completed the entire questionnaire. 29 abandoned at the first major set of questions, with a few at each further list.

The participants were anonymous, so the only demographics are in the self-determined answers provided, they did cover at least 2 countries and a range of ages and backgrounds. There were explicitly no further restrictions on the participants, in an attempt to draw from the widest possible group. However, the ability to complete the questionnaire online indicated a moderate level of computer skill.

1. Qualifications.

The choice of qualification categories is a difficult one – as we can see in Jenkins and Sabates (2007), the range of qualification is very broad, but a classification was needed that was straightforward to understand for the participants but also covered the range expected (it was anticipated that higher degrees would be held by a reasonable portion of the participants).

The list was based on the six-point NVQ levels, ranging from none to a degree. However, the distinctions between poor GCSE and good GCSE were removed, as well as the foundation degree and degree. This was to allow the extra Doctorate categorization to be included, opening up the higher end, in exchange for less detail at the lower end. The results would seem to bear out this division across the participants for this survey. It was also a range that would work for people from across the UK at least (and with some explanation even internationally), across all ages (whatever the actual qualification was).

The first demographic assessment was the academic qualifications of the participants. The results of this are shown in Table 5-12. We find from this that 75.2% of the participants were educated to degree level, a higher proportion than in the national population. The national average is 31.6%, according to ONS, 2012, where NQF level 4 and above indicates degree level. There are 24.8% who are not that well qualified, so there is sufficient representation of these in the remaining results.

Table 5-12 – Qualifications of questionnaire participants based on highest educational qualification achieved.

Highest qualification	Number of participants	Percentage of participants
None	4	3.7
GCSE or equivalent	4	3.7
A levels	19	17.4
Degree or Masters	73	67.0
Doctorate	9	8.2

If we exclude those who did not participate further in the questionnaire, then the figures change, and are shown in

Table 5-13. This is a more accurate representation of those who actually engaged with the questionnaire as a whole exercise. The proportion of these who are degree educated is virtually unchanged, at 75.7%. The most significant difference is that the non-degree educated participants are now more likely to have A levels, with only 3.8% not achieving this level, around half the figure for all participants. While it may be that participants in this group were intimidated by the questionnaire, we cannot say any further, especially as the initial group size was only 8.

Table 5-13 - Qualifications of questionnaire participants who completed at least one question based on highest educational qualification achieved.

Highest qualification	Number of participants	Percentage of participants
None	2	2.6
GCSE or equivalent	1	1.3
A levels	16	20.5
Degree or Masters	52	66.7
Doctorate	7	9.0

2. <u>Computer expertise</u>

The second demographic question should be more useful in categorizing the results, as it relates to confidence or expertise in usage of computer applications. The problems of identifying a measure for computer expertise are also challenging. Studies like Ternus and Schuster (2008) and Grant, Malloy and Murphy (2009) demonstrate an approach to identifying skills where the applications are limited. However, this study needs to have definitions that are applicable over work and leisure usage, as well as over an indeterminate range of applications. Therefore, these approaches have been rejected.

Approaches like Potosky and Bobko 1988 were considered, but the 12-point CUE scale was too complicated for this study. Assessments like this and the Scala scale (Scala 2018) tend to focus on specific skill types (can you use a database, for example), whereas this study needs less specific detail, but more general categories, based on an individual's perception of their expertise.

After exploring other possibilities, the scale was a combination of various other scales and lists of expertise indicators (including the levels of expertise identified by some software installations) to produce a range of options including both time (occasional and regular) as well as competent and expert indicators. The concept of an "expert" user – as this is self-determined – is an indication that others might ask the participant for help (a concept that came up in a few of the scales explored). The "software developer" was included because the expected audience would include many who were involved in writing software and providing a very high-level skillset.

Because this is self-selected, and not determined as a result of testing or assessment of specific capabilities, it should be treated with some caution. Having said that, the groups were all designed to express positive features, so being honest should not carry any perceived stigma with regard to the questionnaire.

The levels used were defined for this project, aiming to cover those who use computers mainly in their leisure time as well as those who work with them. Entries 1-3 cover those who use occasional applications or have to use them as part of their work but whose work is not driven by computers. Entries 3-5 cover those who use computer applications for significant portions of their work and use substantial functionality within this. Entries 5-6 cover those who are involved in software development within their normal work, which can include the writing of macros for word processors or spreadsheets. The results – taking into account only those who completed at least one question in the rest of the survey – are shown in Table 5-14.

Table 5-14 – Self-reported computer experience, based on categories defined for this survey.

Computer experience	Number of participants	Percentage of participants
None	0	0.0
Occasional usage	1	1.3
Regular usage	12	15.4
Competent user	38	48.7
Expert application user	14	17.9
Expert software developer	13	16.7

It is clear from this that most of the users have a reasonable degree of capability with computer systems. Again, this is important because there is only one participant who might be expected to struggle due to lack of regular usage, so the expectation might be that these participants should have few problems with their systems.

Can we make any comparisons between these two data sets, the education and the computer experience? Are the more educated participants also more computer literate? If the qualifications are ranked 1-5, and the experience response 0-5, then we have two equivalent values that we can compare. When we do this, we find a comparison mean of 1.01, with a standard deviation of 0.71, which is inconclusive (this is taken across the whole data set). The high standard deviation by comparison to the mean indicates that the data is quite varied, and more complex analysis is appropriate. We can also look at reducing the ranges by dividing the qualifications by 5 and the experience by 6, giving a mean of 1.09 with a standard deviation of 0.31, which, while better and indicating that there may be some correlation, is still not conclusive.

Is there any means of analysing them that can give insights into how they compare? If we look at Table 5-15, these are the results of subtracting the qualifications from the experience values to identify whether there is a step up in experience or a step down and how large the steps are. The details are cross-tabulated and shown in Table 5-16. When we look at all of the values, as previously, they are so widely dispersed that there is no clear pattern. However, about 1/3 (33%) of the participants have an equal level of education and experience. The highest proportion (45%) of the participants indicate a higher level, where the difference is greater than zero. This indicates that many participants have a level of computer skill higher than their educational level indicator, and when we calculate the figures, we find the mean value of these is 1.6, with a deviation of 0.67, which would suggest that many of the participants are finding a level of expertise within their computer usage slightly greater than their education would indicate. This suggests that they are utilising their educational skills to gain computer expertise. This is the top right-hand quadrant of the table.

Table 5-15 – Comparison of self-reported experience values against educational qualifications, identifying the mean and sum over all respondents, and the number and difference sum based on whether the skill level is higher, lower, or equal to the qualification level. z-values using Kolmogorov–Smirnov > 0.05 for both samples, comparative Kolmogorov–Smirnov $p<0.05$.

	All values		Equal	Skill>qualification	Skill<qualification
Mean	0.50	**Count**	35	48	24
Std. Dev.	1.18	**Sum**		78	-25

Table 5-16: Comparison of self-reported experience against educational qualifications.

	None	Occasional usage	Regular usage	Competent user	Expert application user	Expert software developer	**Total**
None	0	1	0	1	0	0	**2**
GCSE or equivalent	0	0	2	0	2	0	**4**
A levels	0	0	1	9	7	2	**19**
Degree or Masters	2	2	17	33	10	13	**73**
Doctorate	0	0	1	6	1	1	**9**
Total	**0**	**1**	**21**	**49**	**20**	**16**	**107**

There are also a number (22%) whose level of expertise is less than their education would indicate. The calculations on these figures give a mean of -1.04, with a deviation of 0.20, suggesting that a small portion may not have utilised their level of academic skill in their expertise in computer systems. However, we can assess this information against the actual academic achievement level, and it becomes clear that those who are indicating achievement less than their education level are mostly at the degree or doctoral level, which suggests that these participants may be experts in their fields, but they are not as expert in the computer systems they are using.

While the direct comparison of educational achievement against self-reported computer expertise may seem unreasonable, there is evidence that many of the participants are achieving a reasonably high level of expertise in the computer applications that they use, irrespective of actual academic qualification. Those with a lower level of qualification are stretching their skills to work with computer systems, whereas those with higher level of education are utilising their skills with computer applications.

3. Employment area

The third demographic is employment area because the usage of computer applications in work is significantly different from that in pure leisure use. These results are shown in

Table 5-17, only utilising the data from those who continued the questionnaire.

The categorisation here required some consideration because the participants may not all be traditionally employed or may be part-time or even variously employed. One approach that has become more common recently is to obtain a job title (or nearest fit) and identify area from this. This approach was not considered because a wider area of work was of interested, rather than a more specific job title, which may not properly reflect some more diverse or transient work situations.

One approach is to use the classification such as Levine, Salmon and Weinberg (1999), but this has a lot of classifications and is not considered suitable for user selection. The final classifications were based on information from the Office for National statistics ns-sec classifications. However, see the discussion for more insight here. Unfortunately, the ns-sec has been superseded and references are no longer valid.

As expected, a significant portion of the participants are in office or managerial roles (37.6%) and so are liable to be using computer applications for significant aspects of their work. There is also a substantial group in academia (students or teachers) who make up 27% of the participants. These are an important group because they are more liable to be utilising software in more obscure ways rather than the more typical business application usage. It is also significant, because some of these will be using software to teach others and will be teaching them about computer applications and the attitudes to them.

There is also a smaller portion (25%) whose involvement is liable to be leisure usage (unemployed, either voluntary or not) or as an assistance to their work rather than an integral part of it (artistic, caring and retail). These people may be using computer applications to perform certain tasks (email, web searching) but who could achieve their work without the applications, albeit with more difficulty. All of these assessments include some guesswork, but they do cover some of the different ways in which the computer applications may be used across different types of job area. We will see later how the responses are reflected in these groupings.

Table 5-17 – Participants in questionnaire, job areas self-identified.

Job area	Number of participants	Percentage of participants
Student	10	12.8
Teaching/academic	13	14.3
Unskilled/Manual	0	0.0
Skilled/Office	20	21.3
Managerial	15	16.3
Artistic/Creative	5	6.4
Caring	4	5.1
At home/Voluntarily unemployed	6	7.2
Involuntarily unemployed	4	5.1
Retail	1	1.3

Materials and apparatus

The survey was delivered using SurveyMonkey (2009) as an online-only survey. The questions were arranged in two sections: the demographic questions, already covered, and the computer usage questions. The computer usage questions required responses on a scale ranging from “Strongly Agree” to “Strongly Disagree”, guidance for these being taken from Bradburn, Sudman and Wansink (2004), who suggest these wordings. There was also a “Not Applicable” option to allow participants to opt out completely. Bradburn et. al. (2004) do offer some evidence that extreme values are less likely to be chosen in this sort of survey, and so the analysis is done bearing this in mind.

There were 14 statements, generated by the researcher out of the responses from the previous studies for the participants to rate regarding software they felt positive about (seeTable 5-18) and 12 for software they felt negative about (see

Table 5-19).

Table 5-18 – List of statements about computer applications that participants had a positive reaction to.

It was straightforward to start using
It did what I wanted it to
It was easy to work out what I had to do to make it work
It informed me of what I needed to do clearly and in my language
It helped me focus on the information, not the application
It did what it said it would
It provided me with exactly what I needed from it
It had features that I might need, but didn't use
It encouraged me to think of possible uses I could put it to
It was highly compatible with other software applications I use, or worked alongside or with other software applications
While there were many complex options, I could use it simply too
The complexity of the software application was about what I expected and related to what I had to do
The software application understood what I was trying to do
The software application handled minor errors sensibly

Table 5-19 – List of statements about computer applications that participants had a negative reaction to

It ran on only specific computers
I required help or assistance to use it
It was always getting in my way, or displaying messages
It required significant effort to get it to do what I wanted
It provided me with information in an unhelpful format, or requiring further work from me
One or more features I expected were not present
It seemed more restricted than I had expected
It required effort to work with other software applications
The software application was far too complex for what I was trying to do
The software application did far more than I needed and so was less useful for my specific task.
The format of information required was very rigid
Making minor mistakes or forgetting things caused big problems.

The questions were split into two groups, of 7 and 6 respectively, and the groups alternated to ensure that the participants switched from a positive to a negative experience and did not get into a routine in answering the statements. The option was also given to change the software being considered at the switch, in case the statements had prompted them to consider a different application that they would also like to comment on. Because this questionnaire was not focussing on the specific application, rather the responses to applications, this should not impact the overall quality of responses.

The generation of the questions was taken from comments and summaries made in the previous studies, identifying areas that the participants there considered significant. They are all formatted versions of comments that were made by at least 2 participants in one of the previous studies.

There were two final questions, one a free text entry, “What one factor would make the biggest difference to your usage of software applications?”, used as a catch-all to see if there were any other issues or questions that the survey had raised in the participants’ minds. The second was an optional email address, where a simple acknowledgement was sent when provided.

Procedure

A link to the questionnaire was emailed with a basic request to complete (the questionnaire included most of the details needed), and it was completed online. To assess the timing and the wording of the questions, 3 volunteers were asked to complete the questionnaire and provide feedback, which was incorporated into the final questions. Their results were not used in the analysis, as the questionnaire was amended slightly before being sent out.

The initial testing indicated a completion time of 10-15 minutes, which was considered acceptable for an unsupervised test of this sort. The questionnaire was made available for 2 months, terminated when the total number of responses was just over 100, which was considered sufficient for the purposes. Results were downloaded from SurveyMonkey at the end.

Results

What did people like about software applications?

The raw results of the 14 questions about positive applications are show in Table A2-56, and it is clear from this that the second set of responses are less definitive than the first set. This may indicate that the participants are considering their responses. The data can be analysed numerically by scoring the responses as 1 for “Strongly Agree” and 5 for “Strongly Disagree”. If we take the mean responses from each section, we find that part 1 has a mean of 2.1 with a standard deviation of 0.97, whereas the second half has a mean of 2.7 with a standard deviation slightly lower at 0.95. This indicates that the second set of responses is producing results slightly more towards the “neither” response, although not in itself indicating any definitive conclusions. A t-test calculation gives a value of t=3.903 with df=76, indicating that there is some significance in the distinction, but we need to explore further into the results to ascertain what these results can tell us.

Applying a Friedman’s 2-way ANOVA doesn’t indicate any clear patterns in the data, however a Spearman’s correlation does indicate that most of the questions correlate in their responses (at either the 0.05 or 0.01 level, 2-tailed). The only two that didn’t correlate were the responses “It had features that I might need, but didn’t use” and “It encouraged me to think of possible uses I could put it to”. These two did find some correlation with each other and some correlation with some other variables, which would indicate that these questions were less appropriate than the others. The full details are shown in Table 5-21, with the levels of correlation significance found. In many cases, there is a high correlation identified, to the 0.01 level, with a few more indicted at the 0.05 level. This means that the responses to many of the question was consistent, although there is a very notable that two of the questions have very little correlation with others. These seem therefore to stand out and may not be as clear an indication as the others.

A simplified analysis shown in Table 5-20 identifies the divergence of the responses per question. This shows that the first responses of part 2 (questions 8 and 9) are the more divergent for most people. What is more, the divergence for question 8 from the participants’ mean response is substantial (2 points different). Which is “It had features I might use, but didn’t need” – as we saw above, these answers don’t correlate well with the other answers given.

Table 5-20 - Outliers in positive question answers, with the number of occurrences where this question is the outlier, and the mean variation from the mean for these entries. The occurrences column indicates how many for whom this question is most different from their mean response – and therefore could be considered to be there most unreliable response.

Question	Occurrences	Mean difference
1. It was straightforward to start using	5	1.40
2. It did what I wanted it to	2	0.89
3. It was easy to work out what I had to do to make it work	3	1.48
4. It informed me of what I needed to do clearly and in my language	8	1.34
5. It helped me focus on the information, not the application	8	1.85
6. It did what it said it would	0	-
7. It provided me with exactly what I needed from it	2	1.61
8. It had features that I might need, but didn't use	19	2.21
9. It encouraged me to think of possible uses I could put it to	11	1.71
10. It was highly compatible with other software applications I use, or worked alongside or with other software applications	9	1.60
12. While there were many complex options, I could use it simply too	5	1.84
13. The complexity of the software application was about what I expected and related to what I had to do	0	-
14. The software application understood what I was trying to do	4	1.39

Table 5-21 - Spearmans correlation coefficients for positive questions from the Computer usage questionnaire

	It was straightforward to start using	IIt did what I wanted it to	It was easy to work out what I had to do to make it work	IIt informed me of what I needed to do clearly and in my language	It helped me focus on the information, not the application	It did what it said it would	IIt provided me with exactly what I needed from it	It had features that I might need, but didn't use	It encouraged me to think of possible uses I could put it to	It was highly compatible with other software applications.	While there were many complex options, I could use it simply too	The complexity of the software application was about what I expected and related to what I had to do	The software application understood what I was trying to do
It did what I wanted it to	0.60												
It was easy to work out what I had to do to make it work	0.81	0.52											
It informed me of what I needed to do clearly and in my language	0.64	0.57	0.74										
It helped me focus on the information, not the application	0.65	0.50	0.66	0.64									
It did what it said it would	0.51	0.72	0.48	0.48	0.66								
It provided me with exactly what I needed from it	0.58	0.78	0.47	0.50	0.53	0.70							
It had features that I might need, but didn't use	0.15	0.04	0.05	0.06	0.18	0.12	-0.09						
It encouraged me to think of possible uses I could put it to	0.21	0.21	0.16	0.06	0.22	0.19	0.19	0.28					
It was highly compatible with other software applications I use, or worked alongside or with other software applications	0.35	0.47	0.31	0.34	0.35	0.36	0.27	0.28	0.27				
While there were many complex options, I could use it simply too	0.31	0.36	0.32	0.33	0.39	0.32	0.24	0.34	0.40	0.52			
The complexity of the software application was about what I expected and related to what I had to do	0.34	0.54	0.32	0.31	0.35	0.50	0.57	-0.06	0.52	0.31	0.33		
The software application understood what I was trying to do	0.36	0.49	0.38	0.39	0.60	0.59	0.49	0.11	0.23	0.39	0.43	0.54	
The software application handled minor errors sensibly	0.45	0.27	0.38	0.26	0.41	0.37	0.26	0.19	0.32	0.34	0.30	0.37	0.44

No correlation identified

Correlation is significant at the 0.01 level (2-tailed)

Correlation is significant at the 0.05 level (2-tailed)

There are also two entries with higher values – "It informed me of what I needed to do" and "The application understood what I was trying to do", both of which scored 2.5 mean with a standard deviation of 1.05. While the variation is one of the largest in the study, the mean figure points to some significant disagreement on these statements. This is interesting and relevant, as they are both relating to communication issues between the application and the user, the core material of User Experience work. It suggests that even though the software was considered good, the user experience aspect was not necessarily the one that made it good. This may reflect expectations of software, but is also a reminder that making software that interacts well with people, while important, should not be done at the expense of making software that works and does what it should.

There are two more responses with substantially low scores: "It did what I wanted it to" and "It did what it said it would". Again, these may be related, implying that the participants wanted it to do what it said it would. These questions recorded a mean of 1.8 and 1.9 with moderate deviations of 0.76 and 0.79. The comparison between these responses and the previous ones does suggest that if software does what it should, then the user interaction process seems to be less significant. However, this may also be the case that "doing what it said it would" means that the interaction and usage of the interface is not noticed as much. It is when applications fail to do what they should that the interaction issues surface explicitly.

Note that in the Kruskal-Wallis test results shown below, the summary is included as a table, but the more detailed information is considered and analysed for the purpose of identifying any correlations of information.

We can also analyse these responses against the demographic information. A basic correlation provides nothing of significance, which is unsurprising, as the academic levels are ordinals, and there is no reason to suppose that more academically qualified people would be more extreme. When we compare the academic qualification using a Kruskal-Wallis test (Table 5-22), we also find very little correlation indicated, except for the question of whether the tools "understood what I was trying to do". A few others do have reasonable significance figures, but nothing definitive. This means that the academic level is not an indication to a particular set of answers. This is to be expected and means that users are spreading their answers across the ranges whatever level of academic achievement they have.

Table 5-22 - Kruskal Wallis Test Statistics - Grouping Variable: Academic level. Shaded area is Monte Carlo Significance figures (Based on 10000 sampled tables with starting seed 2000000). Based on positive responses to computer applications.

		It was straightforward to start using	It did what I wanted it to	It was easy to work out what I had to do to make it work	It informed me of what I needed to do clearly and in my language	It helped me focus on the information, not the application	It did what it said it would	It provided me with exactly what I needed from it	It had features that I might need, but didn't use	It encouraged me to think of possible uses I could put it to	It was highly compatible with other software applications I use, or worked alongside or with other software applications	While there were many complex options, I could use it simply too	The complexity of the software application was about what I expected and related to what I had to do	The software application understood what I was trying to do	The software application handled minor errors sensibly
χ^2		4.566	7.685	3.916	8.575	4.483	6.260	9.493	2.637	2.558	1.926	2.021	5.825	13.629	3.559
Asymptotic Significance.		0.335	0.104	0.417	0.073	0.345	0.181	0.050	0.620	0.634	0.749	0.732	0.213	0.009	0.469
Significance.		0.354	0.077	0.461	0.046	0.349	0.169	0.024	0.725	0.729	0.840	0.805	0.212	0.002	0.518
99% Confidence Interval	Lower Bound	0.342	0.070	0.448	0.040	0.337	0.160	0.020	0.714	0.718	0.830	0.795	0.202	0.001	0.505
	Upper Bound	0.367	0.084	0.474	0.051	0.361	0.179	0.027	0.737	0.741	0.849	0.815	0.223	0.004	0.531

Table 5-23 - Kruskal Wallis Test Statistics - Grouping Variable: Experience level. Shaded area is Monte Carlo Significance figures (Based on 10000 sampled tables with starting seed 726961337). Based on positive responses to computer applications.

		It was straightforward to start using	It did what I wanted it to	It was easy to work out what I had to do to make it work	It informed me of what I needed to do clearly and in my language	It helped me focus on the information, not the application	It did what it said it would	It provided me with exactly what I needed from it	It had features that I might need, but didn't use	It encouraged me to think of possible uses I could put it to	It was highly compatible with other software applications I use, or worked alongside or with other software applications	While there were many complex options, I could use it simply too	The complexity of the software application was about what I expected and related to what I had to do	The software application understood what I was trying to do	The software application handled minor errors sensibly
χ^2		2.012	1.913	3.128	3.782	7.844	0.269	1.302	4.827	5.002	3.246	0.667	1.749	4.874	2.337
Asymptotic Significance.		0.570	0.591	0.372	0.286	0.049	0.966	0.729	0.185	0.172	0.355	0.881	0.626	0.181	0.505
Significance.		0.574	0.594	0.379	0.288	0.045	0.974	0.731	0.188	0.171	0.358	0.888	0.637	0.177	0.516
99% Confidence Interval	Lower Bound	0.561	0.581	0.367	0.276	0.040	0.969	0.719	0.178	0.161	0.345	0.880	0.625	0.168	0.503
	Upper Bound	0.587	0.607	0.391	0.300	0.051	0.978	0.742	0.198	0.180	0.370	0.896	0.650	0.187	0.529

We can also compare, in the same way, the responses against the recorded experience. Again, a standard correlation provides no correlation, for the same reasons. The Kruskal-Wallis analysis information is shown in Table 5-23, and here we see no correlation indicated at all. This suggests that the experience of computer usage is not impacting the responses being given. The significances are varied but not low enough to indicate a genuine correlation.

The lowest value is for “It helped me focus on the information, not the application”. The Spearman’s correlation for this is negative, so the more experienced, the more negative the response to this. There may be many reasons for this, and we do not have sufficient information from this questionnaire to provide further details as to why this is.

Applying the same approach to the job area, we see the results in Table 5-24. Once again, as expected, there is no significant correlation from a basic Spearman’s, which is reflected again in the Kruskal-Wallis results, where little correlation is indicated. The most significant question response was for “It had features that I might need, but don’t use”, which might be an indication that within certain job groups the relevance of feature availability is consistent. However, it broadly indicates that job area is influencing the results.

<u>What did people not like about their computer applications?</u>

The 12 questions about applications that participants found “unhelpful” or negative, which are handled in much the same way as for the positive responses above and the raw data is shown in

Table A2-57. The responses for these questions were much more varied that for the positive comments above, giving an overall mean for the first part of 2.5 with a standard deviation of 1.21. This indicates less agreement with the suggestions as a whole and with each other from a total of 73 participants. The second half returned a mean of 2.7 with a standard deviation of 1.13 from 64 participants, which again indicates more diverse views.

Applying the Spearman's correlation to this data, we see these results in Table 5-25. We see here far less correlation than we saw for the positive questions, although more in the later questions. This shows that participants are not being as consistent in their answers, but it might also reflect the reduced number of respondents here.

Table 5-24 - Kruskal Wallis Test Statistics - Grouping Variable: Job level. Shaded area is Monte Carlo Significance figures (Based on 10000 sampled tables with starting seed 1660843777). Based on positive responses to computer applications.

		It was straightforward to start using	It did what I wanted it to	It was easy to work out what I had to do to make it work	It informed me of what I needed to do clearly and in my language	It helped me focus on the information, not the application	It did what it said it would	It provided me with exactly what I needed from it	It had features that I might need, but didn't use	It encouraged me to think of possible uses I could put it to	It was highly compatible with other software applications I use, or worked alongside or with other software applications	While there were many complex options, I could use it simply too	The complexity of the software application was about what I expected and related to what I had to do	The software application understood what I was trying to do	The software application handled minor errors sensibly
χ^2		4.464	6.418	5.902	7.038	3.499	7.366	9.736	12.418	6.440	4.617	5.181	8.376	4.684	3.193
Asymptotic Significance.		0.813	0.601	0.658	0.533	0.899	0.498	0.284	0.133	0.598	0.798	0.738	0.398	0.791	0.922
Significance.		0.842	0.624	0.691	0.565	0.923	0.521	0.282	0.111	0.635	0.837	0.78	0.414	0.821	0.941
99% Confidence Interval	Lower Bound	0.833	0.611	0.679	0.552	0.916	0.508	0.270	0.103	0.623	0.827	0.769	0.402	0.811	0.934
	Upper Bound	0.851	0.636	0.703	0.578	0.930	0.534	0.294	0.119	0.647	0.847	0.791	0.427	0.831	0.947

Table 5-25 - Spearmans correlation coefficients for negative questions from the Computer usage questionnaire

	It ran on only specific computers	I required help or assistance to use it	It was always getting in my way, or displaying messages	It required significant effort to get it to do what I wanted	It provided me with information in an unhelpful format, or requiring further work from me	One or more features I expected were not present	It seemed more restricted than I had expected	It required effort to work with other software applications	The software application was far too complex for what I was trying to do	The software application did far more than I needed and so was less useful for my specific task.	The format of information required was very rigid
I required help or assistance to use it	-0.084										
It was always getting in my way, or displaying messages	0.007	0.278									
It required significant effort to get it to do what I wanted	-0.033	0.538	0.451								
It provided me with information in an unhelpful format, or requiring further work from me	0.065	0.305	0.506	0.477							
One or more features I expected were not present	0.221	0.224	0.264	0.319	0.474						
It seemed more restricted than I had expected	0.076	0.022	-0.017	-0.017	0.049	0.158					
It required effort to work with other software applications	0.121	0.085	0.003	0.157	0.102	0.061	0.494				
The software application was far too complex for what I was trying to do	-0.111	0.245	0.108	0.051	-0.006	-0.245	0.395	0.411			
The software application did far more than I needed and so was less useful for my specific task.	-0.039	0.077	-0.019	-0.111	-0.098	-0.317	0.308	0.371	0.794		
The format of information required was very rigid	-0.125	0.126	0.048	-0.047	0.055	0.084	0.595	0.449	0.565	0.427	
Making minor mistakes or forgetting things caused big problems.	0.058	0.149	0.058	-0.023	0.028	0.034	0.662	0.463	0.548	0.469	0.829

No correlation identified

Correlation is significant at the 0.01 level (2-tailed)

Correlation is significant at the 0.05 level (2-tailed)

Table 5-26 - Kruskal Wallis Test Statistics - Grouping Variable: Academic. Shaded area is Monte Carlo Significance figures (Based on 10000 sampled tables with starting seed 213175432). Based on negative responses to computer applications.

		It ran on only specific computers	I required help or assistance to use it	It was always getting in my way, or displaying messages	It required significant effort to get it to do what I wanted	It provided me with information in an unhelpful format, or requiring further work from me	One or more features I expected were not present	It seemed more restricted than I had expected	It required effort to work with other software applications	The software application was far too complex for what I was trying to do	The software application did far more than I needed and so was less useful for my specific task.	The format of information required was very rigid	Making minor mistakes or forgetting things caused big problems.
χ^2		1.681	1.393	2.189	2.998	2.349	3.944	4.009	2.927	8.140	5.531	7.477	8.197
Asymptotic Significance.		0.794	0.845	0.701	0.558	0.672	0.414	0.405	0.570	0.087	0.237	0.113	0.085
Significance.		0.885	0.904	0.777	0.626	0.758	0.441	0.443	0.650	0.055	0.202	0.073	0.049
99% Confidence Interval	Lower Bound	0.876	0.896	0.766	0.613	0.747	0.428	0.430	0.637	0.049	0.192	0.066	0.044
	Upper Bound	0.893	0.911	0.788	0.638	0.769	0.454	0.456	0.662	0.061	0.213	0.080	0.055

The question with the highest correlation is “I needed help or assistance”, indicating that people who needed to obtain help more also responded with a similar degree to the other questions. Needing help might be an indicator that the user of the software is finding it frustrating to use, and so is responding similarly to the other questions.

One interesting result is the coefficient for “It only ran on specific computers” because of the issues that sometimes arise between Windows and Mac software, or older versions of the operating system. The concept of flow means that a user should be able to perform a function without having to think about the device they are on, so for some people who use different computer environments in different situations, this can be an issue. This returned a response mean of 2.9 with a reasonably high deviation, indicating that a significant portion of the respondents are ambivalent about this. However, there are numbers for whom this is a problem – with a total of 20 disagreeing and 27 agreeing. This is the only question that has no relevant correlation with any other questions. It may be that cross-compatibility is not seen as a usage issue as such but as a separate matter. It is important for a number of people but may not be impacting the entire perception of the software.

If we look as we did above at the comparisons with the demographic information, the Kruskal-Wallis analysis against academic level is shown in Table 5-26. As expected, most of the questions don’t show any correlation with the academic level, which is good, but two do come under the 0.1 significance level: “The software was too complex” and “Minor problems caused big problems”. These indicate that the questions are found to more closely reflect the academic level – the higher the academic level, the stronger the agreement with these questions. This may be that such users are more demanding of their software, which would be reflected in these questions.

Table 5-27 - Kruskal Wallis Test Statistics - Grouping Variable: Experience. Shaded area is Monte Carlo Significance figures (Based on 10000 sampled tables with starting seed 1660843777). Based on negative responses to computer applications.

		It ran on only specific computers	I required help or assistance to use it	It was always getting in my way, or displaying messages	It required significant effort to get it to do what I wanted	It provided me with information in an unhelpful format, or requiring further work from me	One or more features I expected were not present	It seemed more restricted than I had expected	It required effort to work with other software applications	The software application was far too complex for what I was trying to do	The software application did far more than I needed and so was less useful for my specific task.	The format of information required was very rigid	Making minor mistakes or forgetting things caused big problems.
χ^2		1.933	1.322	7.452	1.984	1.739	2.438	2.318	0.642	0.894	3.808	3.211	1.722
Asymptotic Significance.		0.813	0.601	0.658	0.533	0.899	0.498	0.284	0.133	0.598	0.798	0.738	0.398
Significance.		0.842	0.624	0.691	0.565	0.923	0.521	0.282	0.111	0.635	0.837	0.780	0.414
99% Confidence Interval	Lower Bound	0.833	0.611	0.679	0.552	0.916	0.508	0.270	0.103	0.623	0.827	0.769	0.402
	Upper Bound	0.851	0.636	0.703	0.578	0.930	0.534	0.294	0.119	0.647	0.847	0.791	0.427

Table 5-28 - Kruskal Wallis Test Statistics - Grouping Variable: Job Area. Shaded area is Monte Carlo Significance figures (Based on 10000 sampled tables with starting seed 1429851888). Based on negative responses to computer applications.

		It ran on only specific computers	I required help or assistance to use it	It was always getting in my way, or displaying messages	It required significant effort to get it to do what I wanted	It provided me with information in an unhelpful format, or requiring further work from me	One or more features I expected were not present	It seemed more restricted than I had expected	It required effort to work with other software applications	The software application was far too complex for what I was trying to do	The software application did far more than I needed and so was less useful for my specific task.	The format of information required was very rigid	Making minor mistakes or forgetting things caused big problems.
X^2		6.228	2.843	11.195	9.621	8.552	5.168	8.994	4.347	4.645	6.945	8.261	11.193
Asymptotic Significance.		0.513	0.899	0.130	0.211	0.286	0.639	0.253	0.739	0.703	0.435	0.310	0.130
Significance.		0.545	0.926	0.112	0.200	0.289	0.692	0.256	0.778	0.742	0.457	0.311	0.110
99% Confidence Interval	Lower Bound	0.532	0.920	0.104	0.190	0.277	0.680	0.245	0.767	0.730	0.444	0.299	0.101
	Upper Bound	0.557	0.933	0.120	0.211	0.300	0.703	0.267	0.789	0.753	0.470	0.323	0.118

The next comparison is with the experience level. The Kruskal-Wallis information is in Table 5-27, and it is showing that the experience is also not impacting the results found, although there is some closer correlation for “It required effort to work with other applications”. It may well be that the more experienced users are working with more applications or more complex tools than those with lesser experience, and so they find more issues here.

Finally, the comparison with job level is shown in Table 5-28, showing that there is no correlation, although again, there are two questions with relatively low significance levels: “Getting in my way” and “Minor issues caused big problems”. However, the low numbers of respondents for some of the job categories (especially those that are the higher end of the list) means that it is dangerous to take too much from these results.

Other comments

52 respondents provided some information in response to the open-ended question “What one factor would make the biggest difference to your usage of software applications?”.

The participants identified an issue with the platform – software needed to be used on Mac and PCs and continue working through platform upgrades. One participant responded that everyone should use Linux because windows XP was slow and unreliable. There is expressed here a desire to have platform software that is invisible, meaning that you never know or need to know what you are actually running because everything runs on everything. In User Experience terms, invisible software is part of the ubiquitous computing idea, that the computing applications are everywhere, but they are not noticed. It would seem for these participants at least, we are a long way from that.

A significant number of participants (15) identified an issue with help or assistance in using the software. One request was for a printed manual rather than the difficult to use help system, and another was for the system as a whole to be proofread by someone with limited computer knowledge. Others expressed a desire for a “bigger picture” approach to help rather than just detailed assistance, or “cookbook” processes, to allow a user to grasp an entire task. One wanted better and more specific training. While this is not related to the software product itself, it is still an important part of the system introduction process. While training should not be needed for a good and well-written application, specific training for how particular people achieve their tasks within an organisation is very relevant. Another related comment was for the system not to crash when they tried things out. In this case, they wanted to learn by doing but the system was not robust enough for this. All of these relate to users attempting to understand how a system works but not having the resources available to them when they needed them to manage this. The increase in internet-provided support may improve this to an extent, but users need to have immediate access to good quality, comprehensive and flexible material. The delivery mechanism is actually less important than the immediate and appropriate availability. Most of the user’s questions were probably answered within the system, but they could not access them in a way that worked for them at that point.

There were 18 responses that were about doing the things that it says easily. These are questions about the software not doing what the user bought it to do. One responded “not trying to be too clever by second guessing”, although another commented that “Google applications seem to have a kind of ‘sixth sense’ … this is fantastic.” This implies that preparing for possible actions can work if done well. Google applications do not have any sixth sense, but clearly their user interaction appeals to this particular participant. Others commented that it should be “easy” or “straightforward” to use, and should “cope with minor error without tripping me out of the whole system”. So safe and simple to use are important features for these. Another commented that “they like to feel that they are in charge of the computer” and that the system is doing what they want, not the other way around. A few comments include phrase like “clear instructions”, which may refer to the help systems noted above, but may also refer to assistance and instructions while using the application, the various texts around the functional processing to guide and indicate the next steps.

There are a number of other comments, indicating that issues like price are also important. One comment is that they would like someone to tell them what products to use for various tasks. This is an indication that there is still confusion and disillusionment in terms of trying to find the right tool for the job. Other people don't like software changing between versions and losing functionality. What is important to understand from all of these comments, however, is that while they are useful and valuable insights, they are comment responses, and so each comment is a unique response. While the assessment above tries to group these, this is less definitive than the self-selected groupings of the preceding questions. It is dangerous to take conclusions from such scant evidence without significantly more to back this up.

The spread of responses across the various demographics is shown in Table 5-29, Table 5-30 and Table 5-31. We can see that the more educated the participant, the higher the chance that they would respond in general, with over half of those with degrees providing some feedback. This may be simply that the more educated feel that they have something to contribute or that they are thinking more about the feedback than others. Their academic achievement may give them more confidence in their replies. There is very little in the experience breakdown of significance, especially given the low respondent numbers in the lower groups. However, the job area does show one interesting factor, that students have generally not provided comments. In all other groups, at least half and often two thirds of respondents have given some feedback.

While it would be possible to do more analysis on this against the positive and negative responses, this has not been done, mainly because only half of the respondents have given any feedback, and many of the responses are too brief to provide opportunities for proper textural analysis.

Table 5-29 - Response count and proportions for open-ended responses against academic achievement

Academic	Response Count	Percentage Responded
None	1	25
GCSE or equivalent	0	0
A levels	8	42
Degree or Masters	37	51
Doctorate	6	67

Table 5-30 - Response count and proportions for open-ended responses against computer experience

Experience	Response Count	Percentage Responded
None	0	-
Occasional usage	0	0
Regular usage	7	58
Competent user	24	63
Expert application user	11	79
Expert software developer	10	77

Table 5-31 - Response count and proportions for open-ended responses against job area

Job area	Response Count	Percentage Responded
Student	3	30
Teaching/academic	9	69
Unskilled/Manual	0	-
Skilled/Office	13	65
Managerial	12	80
Artistic/Creative	5	100
Caring	3	75
At home/Voluntarily unemployed	3	50
Involuntarily unemployed	3	75
Retail	1	100

Grouped analysis

For the next part of the analysis, we need to split the respondents into groups and analyze the responses within these groups. There are three groups that naturally appear, based on the job description (and were originally reflected in the sources for these groups):

Group 1 – Academic – Student, Teacher, Academic

Group 2 – Office or Manual – Unskilled/Manual, Skilled/Office, Managerial

Group 3 – Caring, creative or voluntary – Artistic/Creative, Caring, At Home, Unemployed, Retail.

The Retail entry does not quite fit into the final category, but there was only one respondent here, and the group as a whole then makes up a reasonable group of people for whom traditional computer usage is unlikely to be a fundamental part of their usual work.

If we first look at the qualifications against the expertise reported by the participants, represented on a hi-lo chart, we see the results in Table 5-32. It is clear from this that the hi-hi quarter is the most common one, indicating that those with high qualification and high expertise do make up the most significant portion of the participants. At the same time, the hi-lo quarters are stronger than the lo-lo, indicating that this is not a direct correlation between experience and qualification for this grouping. Further, this is the case all the way through the groups. The caring group is the one showing the least fit to this pattern.

Table 5-32 - hi-lo chart representing expertise against qualifications. lo represents a score of 3 or less, hi of greater than 3.

		Academic		**Office**		**Caring**	
		Experience		Experience		Experience	
		lo	hi	Lo	hi	lo	Hi
Qualification	Lo	1	6	0	11	3	4
	Hi	3	18	6	35	8	10

The implications of this are that the participants tend to have significant levels of either expertise or qualification, or both. In fact, across the groups, only 4 participants have low levels of both, whereas 63 have high levels of both. We can also see that for the academic and office groups, the high experience low qualification rate is higher than the high qualification low experience sectors (the opposite is the case for the caring group). This means that experience tends to slightly outstrip qualifications, whereas with the caring group, computer experience tends to be reduced, as might be expected based on the nature of these roles.

The next hi-lo diagram is across the same groups but for a comparison of the positive and negative responses. These details are shown in Table 5-33. While the expectation is that the hi-lo axis should be the most significant one, indicating a degree of agreement across the two sides, this is not what is seen. The lowest entry across all groups is the hi-hi quadrant, but the hi positive-lo negative is also lower than expected. What we find across all groups is that the lo positive, lo negative quadrant is more significant, indicating that a significant number of people chose lower positive but also lower negatives: less commitment in either category.

Table 5-33 - hi-lo chart representing positive responses against negative responses. lo represents a score of 2.5 or less, hi of greater than 2.5.

		Academic		Office		Caring	
		Negative		Negative		Negative	
		lo	hi	Lo	hi	Lo	Hi
Positive	Lo	9	6	7	17	7	6
	Hi	2	2	5	4	4	1

There is also some value in taking a look at a survival analysis across the questions, especially given that there were a number of participants who dropped out during the survey. The reason for considering this is to understand whether any particular statistics are substantially undermined by being partial results. It may prove useful to assess these by the groups, identifying whether there are any differences in survival between these groups. The results of this are shown in Table 5-34, a basic analysis to identify where people have dropped off from the total, if at all. It seems here that the academic and office groups have reasonably high mean drop off, although there are substantial deviations in these figures. We should point out that the big divisions come at 7, 13, 20 questions, which is the end of each page. This would seem to suggest that the Academic and Office groups are stopping at the end of pages 2 or 3.

Table 5-34 - Survival analysis data showing mean drop-off question, out of 26, divided by job grouping. Values are the point at which the participants answered their final question, out of 26.

	Academic	Office	Caring	All
Mean	18.6	16.3	5.1	17.0
Std Dev	10.5	12.2	3.0	11.6
Total entries	28	52	7	107

The caring group has a very different profile, with a mean drop-off point in the first page, which is an indication that these participants were very poor at finishing. While there are only 7 participants in this group, this is a very strong indication that the groups were not very happy with filling out this on-line questionnaire. We need a further analysis to look at how many failed to finish, and how many failed to start. This analysis is shown in Table 5-35, including the total participants from the previous table, and a count of participants who completed zero of these questions. This provides some additional useful information about all of the groups. What is clear from the Academic and Office groups is that the non-starters (that is, those who completed no questions) are a significant proportion of the non-completers.

Table 5-35 - Survival analysis of participants who failed to complete the questionnaire, divided by job grouping. Values are the fian question number answered.

	Academic	Office	Caring	All
Mean	2.9	1.3	5.1	4.1
Std Dev	6.1	4.4	3.0	6.6
Mean of starters	11.8	11.4	5.1	11.6
Non-starters	22	47	0	29
All entries	**28**	**52**	**7**	**107**

For the academic group, of the 28 non-completers, 22 answered no questions, so there are only 6 participants who started and gave up. The mean of all non-completers is very low because these non-starters reduce it substantially: the mean of starters is 11.8, implying that they have dropped off after page 1 or 2.

The office group shows a very similar pattern, with only 5 of non-finishers completing any of the answers and a mean very similar. In fact, looking at the actual data of the non-completing starters for these two groups, we find that all of the drop offs are at the end of pages (7, 13 or 20).

The caring group is a different pattern entirely. In fact, all of these non-finishers actually started the questionnaire, and, examining the data in more depth, none of these finished at the end of a page. This would suggest that these people made decisions part way through a set of questions to give up the questionnaire. Their behavior is quite different to the other two groups.

Overall, it does appear that the majority of non-finishers started the questionnaire and decided to abandon once they saw the first set of questions. Of those who did continue to answer, most of them made it to the end. This would indicate that an improved response might have been achieved by further warnings about the nature of the questionnaire, but there was no clear place within the questions that people would especially drop off.

Discussion

This survey identifies some key issues in relation to the flow/frustration dichotomy hypothesis. On the frustration side, this supports the assertion made by Cooper, Reinmann and Cronin's (2007) "…commonly interactive products irritate us because they aren't considerate, not because they lack features" (Cooper, Reinmann and Cronin, 2007, p. 250). The results support the notion that features are less important than an application that works well (that 'behaves nicely' for the user).

We have found here that users report high levels of frustration across the board with applications that did not do what was expected. This was the case regardless whether these expectations were reasonable or otherwise. The frustration felt by users appears to be because applications failed to enable the flow that participants were seeking in achieving their tasks and activities. This sense of dehumanisation – the lack of a proper appreciation of the ontological space occupied by the user (see Chapter 2) – reflects still some of the concerns expressed in Zuboff (1988, as referred to in Chapter 2 above). And, as with that study, this will have negative results on the individuals concerned and will feed into frustrations in their work environments.

Conversely, good applications enable flow. One participant credited Google with having a "sixth sense". This could be interpreted as simply a way of describing software behaving as expected, doing what the person wants and what naturally flows from what they are already doing. Achieving this level of acceptance from users is not a mystical art; it is the principle of good software design. As Cooper, Reinmann and Cronin say (2007), "Good design makes users more effective" (p. 16). Software that gets out of the way of the user (enabling them, but not interfering) will score higher from the user's point of view.

What is also clear is that adaptive software is important. When it comes to help and assistance different people want different ways of resolving this. Some people want immediate, hands-on assistance and help, whereas others find it more useful to have none of this, but be told clearly what an issue is. Some people want to learn by examples and "cookbooks", whereas others enjoy simply experimenting with software to see what it can do. Providing the wrong solutions to an individual problem is frustrating, whereas providing the right one is flow-enhancing.

This study does have flaws. The first one is the lack of any control group, any group against whom the results can be compared. There is also an issue with the classification of job area, which would be better (if redone now) against the SOC classification (SOCPC, 2010). However, the results are valid (bear in mind that the SOC classification was produced after this survey was done) against the classification used.

Additionally, these experiences are fluid and changing, as work experience changes and the usage of computer systems change, so this type of study would be valuable to repeat periodically to assess changes in usage experience.

This study has provided support of the first and third hypotheses. The distinctions identified within this study offer support to the hypothesis that "**The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems**" because the questions did appear to be valid across all of the categories. There is also some support for the third hypothesis that "**A frustration experience can have a statistically significant impact on a participant's task-completion experience**" because the frustrations seem to have wider implications – as mentioned above, the frustration will have implications for the individual's entire experience of work.

This study has assessed the usage of computer systems, drawing from the analysis done previously, and providing more insight into how people use computers, especially drawing demographics on computer usage. Having understood the general usage of computer systems, it is now time to look into the concepts of "flow" and "frustration" as perceived by users of systems. The next study is to look at the concept of flow by users.

Chapter 6 - Flow Questionnaire.

The first and primary hypothesis in this study is that the dichotomy of flow and frustration is a valid one. Having addressed the core questions of how people use computers, what they feel that their frustrations and positive experiences are, this study moves on to an assessment of these experiences in more formal terms. The first part of this is to look at the experience of Flow, in order to examine whether this leads to improved efficiency for workers in terms of work completion.

The purpose of this study was to ascertain whether the concept of Flow as defined in other parts of this study is something that participants experience at all and whether this applies to the use of computer systems or not. As Csikszentmihalyi (2002) says, “In the course of my studies, I tried to understand as exactly as possible how people felt when they most enjoyed themselves and why.” In terms of the hypothesis, the purpose of this study is to ascertain whether the participants experience flow at all as one half of the dichotomy and whether Csikszentmihalyi can be shown to be right.

Method

Participants

A total of 25 people responded, of which only one failed to answer all of the questions and so will be excluded for later parts of the analysis. There were no restrictions on the participants: they were male and female, from the UK and the US, with a range of skills and computer usage. The participant list was drawn from the researcher’s contacts (friends, family and colleagues), including social media contacts, with a request to pass the link on to others.

Materials and apparatus

The questionnaire (reproduced in Appendix 3) was built and distributed using SurveyMonkey (2012). There are three parts of the survey: computer usage, non-computer usage and micro-flow. For the computer usage, the participants were asked to identify a situation where they might have experienced Flow. This was done without using the word “Flow”, which might be perceived as a more technical term than is appropriate for this and might confuse the participants. For this situation, a number of questions were asked to identify whether the indication of Flow was justified. Some questions were asked about how important the computer was to the experience, to ascertain whether this experience was enhanced or hindered by the computer involvement.

For the non-computer usage, they were similarly asked to identify a Flow-type experience, this time one that did not involve a computer. The same questions were then asked of the participants. Finally, for the micro-flow questions, a few insights into when smaller processes seem to flow were asked. Micro-flow is a term Csikszentmihalyi uses in various places for experiences of Flow over very short periods, for very short-term tasks (see Csikszentmihalyi, 1975).

The set of questions asked are listed in Table 6-36.These have been gleaned from the range of indicators that Csikszentmihalyi uses in his works, as well as the indicators from the previous studies. The problem with Csikszentmihalyi’s writings is that many of his studies are from creative environments (see 2008 in particular), and much later study (for example Koehn, 2017) apply to non-technical environments (sporting is a prime area where the Flow thesis has been applied) and they are not necessarily appropriate to transfer over to the technical realm. So, the questions are drawn from the originals, but made appropriate to the technology realm.

These references will be used in the remaining analysis to refer to the questions. Questions B, E and F have the results reversed in the analysis because they are expressed negatively, so this enables direct comparison. These were chosen as the questions most used across the works to reflect whether flow was being experienced.

Table 6-36 – References for question in flow questionnaire, with negatively expressed questions highlighted.

Question Reference	Question Text
A	You felt in control of the process
B	**You were bored by the task, or felt disconnected from it**
C	I was easily able to achieve the task
D	You had a clear focus on the achievement of the task
E	**It seemed to take far longer than it actually did**
F	**The task was really beyond my ability, although I managed it**
G	There was a sense of personal reward having achieved it

Procedure

An email link to the questionnaire was sent out, and it was completed online. To assess the timing and the wording of the questions, 2 volunteers were asked to complete the questionnaire and provide feedback, which was incorporated into the final questions. Their results were not used in the analysis, as the questionnaire was then amended before being sent out.

The initial testing indicated a completion time of 10 minutes, which was considered acceptable for an unsupervised test of this sort. The questionnaire was made available for 2 months, and the results were downloaded from SurveyMonkey at the end.

Results

The questions can be combined to give a “flow score” for each participant, which are used in the result details shown below. These figures can be then compared across the two sets of questions. The questions are marked from zero to four, where four indicates the most agreement with Csikszentmihalyi’s analysis of flow, and 0 indicates least agreement.

Computer usage

The raw data for the computer usage flow questions are shown in Table A3-58. The mean of all the values gives a value of 3.3, with a relatively low standard deviation of 0.45. This high mean figure indicates that there is a strong degree of flow, as indicated by these pointers, experienced by the participants when using a computer. The correlations are explored in

Table 6-37. We see there are a few significant correlations, indicating that these questions are more definitive than others. The question of "There was a sense of personal reward" – the autotelic payback, reward form this activity rather than from outside – seems to be important.

There are two questions “I was easily able to achieve the task” and “It was really beyond my ability”, which had no correlation with anything else (and actually didn’t have a particularly close correlation with each other). The small size of the sample means that this is difficult to analyze in any depth, but it may be about participants not wishing to admit that their tasks were either too easy or too difficult for them.

Table 6-37 - Correlations between questions for flow questions – computer usage.

	You felt in control of the process	You were bored by the task, or felt disconnected from it	I was easily able to achieve the task	You had a clear focus on the achievement of the task	It seemed to take far longer than it actually did	The task was really beyond my ability, although I managed it
You were bored by the task, or felt disconnected from it	0.394					
I was easily able to achieve the task	0.292	-0.264				
You had a clear focus on the achievement of the task	0.577	0.592	0.019			
It seemed to take far longer than it actually did	0.363	0.482	0.012	0.390		
The task was really beyond my ability, although I managed it	0.000	0.126	-0.065	-0.056	-0.061	
There was a sense of personal reward having achieved it	0.623	0.696	-0.093	0.510	0.436	0.000
No correlation identified						
Correlation is significant at the 0.01 level (2-tailed)						
Correlation is significant at the 0.05 level (2-tailed)						

This section also includes a question about how important the computer was to the process and to making it flow. The raw data for this is shown in Table A3-59 (in the appendix). The questions for this section are shown in Table 6-38, with one question having the responses reversed because it is expressed negatively. When we analyze the values, the mean value for all participants is 2.9 with a 0.3 standard deviation, indicating that the mean is probably a good indicator. This would indicate that in this survey, the computer is substantially important to the processes being undertaken.

Table 6-38– References for importance of computer question in flow questionnaire, with negatively expressed questions highlighted

a	The computer was critical to the task
b	I hardly noticed the computer usage, although it was important
c	**The computer application was an irritation**
d	The computer application was a real help

The correlation statistics are shown in Table 6-39, where it is clear that there is significant agreement across the questions, so the mean value of 2.9 above is a good indicator of the significance of computers across the questions.

Table 6-39 - Correlation between answers on whether the computer was a significant factor in flow questionnaire

	The computer was critical to the task	I hardly noticed the computer usage, although it was important	The computer application was an irritation
I hardly noticed the computer usage, although it was important	-0.686		
The computer application was an irritation	-0.462	0.702	
The computer application was a real help	0.708	-0.538	-0.551
. Correlation is significant at the 0.01 level (2-tailed).			
. Correlation is significant at the 0.05 level (2-tailed).			

<u>Non-computer use</u>

The raw data for the non-computer usage flow question are shown in Table A2-57 (in the appendix – there is little data we can take form this directly). The mean of all the scores (scored the same way as above) gives a value of 3.3, the same as for the computer questions. There is a little less variation in these figures, with a standard deviation of 0.33. This still indicates strong agreement with the flow hypothesis.

Table 6-40 - Mean values per question for Flow question with non-computer questions Higher values indicate stronger agreement. This data can be compared to the computer values in Table 6-38

Question	Mean value	Std Dev
A. You felt in control of the process	3.0	0.69
B. You were bored by the task, or felt disconnected from it	3.6	0.58
C. I was easily able to achieve the task	3.0	0.54
D. You had a clear focus on the achievement of the task	3.5	0.65
E. It seemed to take far longer than it actually did	3.0	0.68
F. The task was really beyond my ability, although I managed it	3.3	0.46
G. There was a sense of personal reward having achieved it	3.6	0.50

The correlations are shown in Table 6-41, and it is clear here that there is very little of relevance in this – indicating that there is a high divergence of answers. The only really significant correlation is found between the three questions on achievement, which may indicate that task achievement is an important aspect – in comparison with the results in Table 6-40, these three questions have comparatively high values, showing strong agreement with the statement, although the standard deviations on these questions are not notable.

Table 6-41 - Correlations between questions for flow questions – non-computer usage

	You felt in control of the process	You were bored by the task, or felt disconnected from it	I was easily able to achieve the task	You had a clear focus on the achievement of the task	It seemed to take far longer than it actually did	The task was really beyond my ability, although I managed it
You were bored by the task, or felt disconnected from it	0.063					
I was easily able to achieve the task	0.453	-0.058				
You had a clear focus on the achievement of the task	0.239	0.516	0.180			
It seemed to take far longer than it actually did	0.132	0.364	0.119	0.140		
The task was really beyond my ability, although I managed it	0.286	0.103	0.155	0.182	0.286	
There was a sense of personal reward having achieved it	-0.052	0.514	0.093	0.536	0.049	-0.030
No correlation identified						
Correlation is significant at the 0.05 level (2-tailed).						
Correlation is significant at the 0.01 level (2-tailed).						

Micro-flow

The final part of the questionnaire was about "micro-flow" and was the one part where the term "Flow" was actually used as the simplest means of describing what was being asked for. The raw data from this section are shown in

Table A4-61. Four of the questions are range questions again, as the previous ones, and can be scored as previously. They are all about where and when some form of "micro-flow" occurs. The mean values of these can indicate whether this is a situation that encourages flow or not. Analysis on this, and how it relates to the other questions, is shown below.

The final question is a simple selection of how often this sort of micro-flow experience occurs, according to the participant. The aim of the scale here is to cover three broad frequencies: Rarely, Often and Constantly. Each of these has two or three subdivisions, which will provide appropriate detail for this study.

The occurrence is rated from "Regularly and repeatedly every day", recording a value of 7, through "Every day, at least once", "Most days", "Once or twice a week", "Every few weeks", "Every month or so", "Infrequently – several times a year" down to "Very infrequently", scoring 0. Significantly, there are no responses under 3, and only 2 responses under 4. The breakdown is shown in Table 6-42, where is shows clearly that for most people they do experience what they describe as a flow experience every few weeks or once or twice a week. 17 out of 24 respondents came into this category.

Table 6-42 - Frequency of micro-flow occurrence as reported by flow questionnaire respondents.

Frequency	Number of responses
Very infrequently	0
Infrequently – several times a year	0
Every month or so	0
Every few weeks	2
Once or twice a week	8
Most days	9
Every day, at least once	2
Regularly and repeatedly every day	3

Finally, there was one respondent who did not answer any of the questions in this section. There is no reason given, and they have not missed any of the other questions. They will be taken out of any analysis that includes comparisons to this data. All of the other respondents answered all of the questions in this section.

Comparison between the two question sets.

Do we find any correlation between the answers for specific questions across the computer use and non-computer use? A summary of the details is in Table A3-60. The p-values, taken from a one-way ANOVA on the two data sets, indicate that in most cases, there is no correlation between the values. Those questions which do have relevant p-values also have the most discrepancy in the mean values, which probably accounts for the difference. It would appear that, overall, using a computer makes the participants feel slightly less in control of the process, probably because the computer is involved. Additionally, using a computer increases the sense of the task being within the participant’s ability. This may reflect that participants do not feel comfortable (and so in the flow state) when attempting computer tasks that are stretching their ability.

Applying a correlation analysis to the two sets of data (from computer usage and non-computer usage) shows no significant correlation. However, the Wilcoxon Signed Rank test does indicate that the median of differences between the computer and non-computer usage is zero with a significance level of 0.05. Details of this assessment show both positive and negative differences indicated, and the sample size is not high enough to take much else from this.

Do we find anything if we compare the two mean flow indices for each participant from the two different question sets? Once again, we find that the p-values are high, and there is no correlation between the values. This is not very surprising, as the range of values is not great, and, because the values are discrete entries, not a continuous scale. A small change in the response given does not necessarily reflect as significant a change in the participants feeling.

Overall, there is no clear correlation across the two sets of answers, but this is most likely to be because of the integral nature of the answers, and arithmetic comparisons always run the risk of not showing the true picture.

Can we compare the flow questions against the importance of the computer? Rather than comparing each individual question, can we combine these to produce a single flow-factor for each participant? Will we lose substantial information if we do this?

To see whether there is a substantial variation, we need to compare the standard deviation of each participant's responses against the mean of their responses. When we do this, we find very low deviations, against relatively high mean figures. Similarly, when we compare the means from each of the question sets, we find that the differences are not substantial against the values, so by combining them, we do not generally lose significant detail. We can therefore take means of these values and provide a flow-factor – or f_p - for each participant, which reflects the participants' responses well.

For the importance of the computer, we need just the computer entries, so if we only use the computer flow-factor f_{pc} for this comparison, what do we find? We find that only question b shows any correlation, and this is a negative correlation. The less the person seems to exhibit flow, the less they indicated noticing the computer, although it was important. It may be that this reflects the fact that more computer usage tends to go with less flow. When we take a mean of the computer importance values and use this to compare with the f_{pc} value, we do find a clear correlation, with p-values under 0.05 and a slope of 0.32 (see Figure 6-7), meaning that overall increased flow does seem to be associated with increased importance of the computer system.

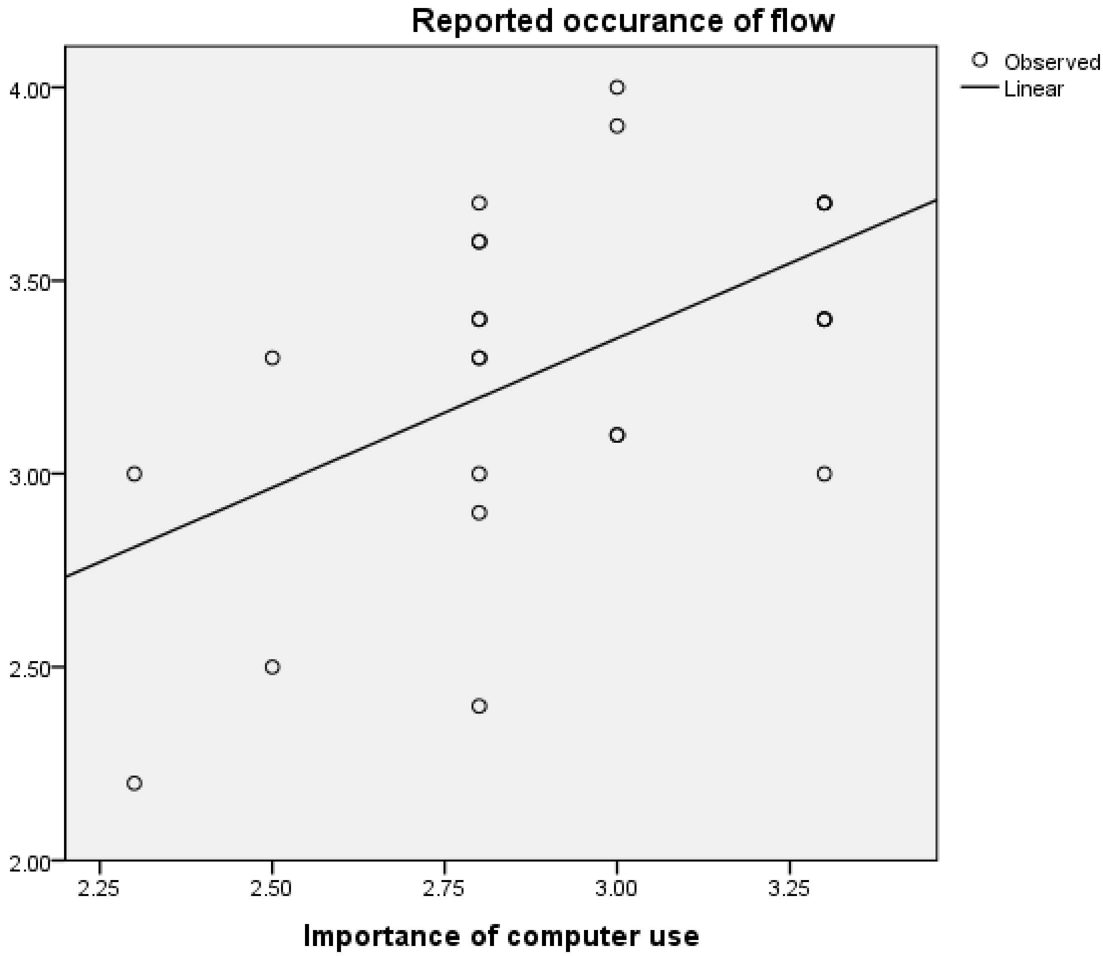


Figure 6-7 – Linear curve fit for f_{pc} and importance of computer usage mean value (which provides the closest fit)

What about the comparisons of the f_p against the micro-flow responses? Comparing the f_p value against each individual micro-flow question does show p-values under 0.05, but there is no definitive relationship that can be established, mainly because (again) the individual question answers are integral, which does not make a good comparison. If we take the mean of these values (the f_m value), however, there is a correlation. We find that the presence of micro-flow does appear to be linked to the increased identification of flow in specific situations. While there is definite correlation (that increasing instances of flow and micro-flow seem to be related), far more data would be needed to draw further conclusions from this.

Finally, is there any correlation between the reported experience of micro-flow (f_m) and the frequency with which this occurs? Firstly, to bring them into the same scope, they are amended to range from 0 to 4, returning the f_{mo} value. There is definitively a correlation, as it returns a p-value significantly under 0.05. This is shown in Figure 6-8. What seems clear from this is that the higher the reported incidence of micro-flow, the more often this occurs, which may be related to the clearer recognition of the effect, but it appears to be stronger than this, especially as there are some outliers shown.

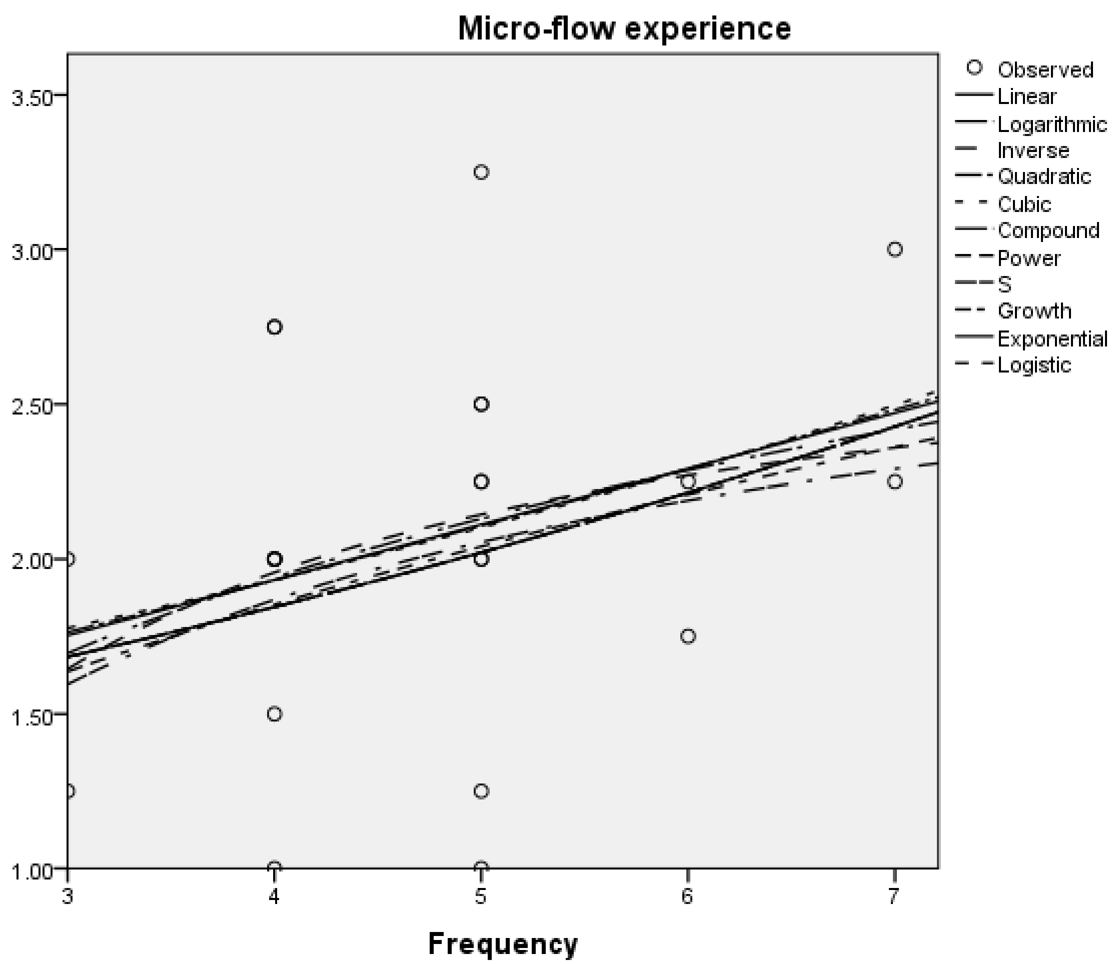


Figure 6-8 – Curve fit for reported experience of micro-flow f_m and frequency of reported micro-flow f_{mo} All models are shown as they all have a similarly low R^2 value.

Hi-lo diagrams

We can also do a simple hi-lo presentation of this data, comparing the computer usage values and non-computer usage values. The challenge here is to identify the break point, because when we put a mid-range division of 2.5, we see the results in Table 6-43. What is clear here is that the lo quadrants are being largely excluded, so a different point to divide is required to identify any significant correlation.

Table 6-43 - hi-lo chart representing computer usage flow score against non-computer usage flow score. Mid-point division is 2.5

		Non- computer usage	
		lo	hi
Computer	Lo	0	0
	Hi	2	19

Looking at the median values of the frequencies, and adjusting the break-line up to take account of the higher position, we do find the results shown in Table 6-44, where we see that, once again, the hi-hi quadrant is the most populous, but less so. The other quadrants are fairly even, indicating that while for most participants, high flow scores for both computer and non-computer tasks are the norm, those who do not experience this may find a degree of flow in some circumstances.

Table 6-44 - hi-lo chart representing computer usage flow score against non-computer usage flow score. Mid-point division is 3

		Non- computer usage	
		Lo	hi
Compu	Lo	4	3

Hi	2	12

Discussion.

Firstly, there do seem to be indications of flow experiences, and in fact some degree of micro-flow appears to be very common. This is a good indicator that the work of Csikszentmihalyi in this respect is being broadly supported. This seems to be in support of the first hypothesis "**The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems**".

Secondly, there is very little real indication of substantive differences between the appearance of flow when using a computer and when not, in appropriate activities. This means that we can utilize the concept of flow within the context of computer usage and also that we do not need to reassess the context or assessment of the use of flow for the computer application context.

What is significant is that, since this study was conducted, there has been a lot of investigation into the flow experience. Articles like Stormoen (2016), Wilson (2016), as well as more recent work by Csikszentmihalyi and others (Mao, 2016, and Csikszentmihalyi, 2017) indicate that the experience of flow is becoming much more common, and these studies do tend to support the findings in this study across different fields.

In conclusion, we do find that the participants report the indicators of flow in certain circumstances. While there are no differences between the situation using a computer and one not using a computer, there are some indications that computer activity may be less indicative of flow when they are stretching the participants a little more. Those who recognized a flow state also appeared to recognize the micro-flow state, where flow would occur for short periods or in small tasks. Additionally, the more that this is recognized, the more often it is reported as occurring. The participants do seem to experience flow reasonably frequently, with no-one indicating less than every few weeks

Further to this, it would seem that the categorization phrases used are satisfactory, because the results tend to be consistent. There are a few fluctuations, but the results are broadly similar across the participants. This experiment would seem to support the existence and reality of flow within the experience of most people, and it is something that people experience regularly. This would support the hypothesis I am investigating. They do experience it with and without computers. There is some indication that the use of computers is a slight negative correlation with the experience of flow.

There are limitations in this study, mainly in terms of the numbers of participants. However, it was conducted earlier than many others and can provide an early example of this, to be used alongside the other studies that are showing that the flow aspect of computer usage – never mind the flow concept that is being seen as a more important aspect of any process – is gaining hold in psychology work.

The next experiment will look at some of the important frustration aspects, which may provide some of the reasons why computer systems do not enhance flow. Having identified the experience of frustration and smoothness of process in the perception of users, this next study takes a closer look at the experience of "flow", although not emphasizing this word or the concept explicitly. We have identified that a significant portion of the participants do seem to experience some degree of flow at some time, and we explored the impact of computer usage within this. Next, we need to explore a basic perception of the frustration aspect, and see if we can identify a frustration indicator or point.

Chapter 7 - Experiments

Process Break experiment

Having examined the Flow aspect of the dichotomy and identified that this is part of the experience that participants have, we now need to look at the frustration side and explore this. This is being done in two separate experiments, the first one being an attempt to identify some visual or user interaction elements that interrupt the flow through a process or a web page. We are continuing to test the first hypothesis, “**The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems**”. Exploring the frustration aspect of user experience, this experiment will also explore the third hypothesis that that “**A frustration experience can have a statistically significant impact on a participant’s task-completion experience.**” In particular, this experiment will look into whether frustration continues to have further impacts on performance.

This experiment will explore whether cognitive process breaks actually occur while processing a task. These breaks are expected to equate to the short-term memory capacity and duration. If the short-term memory is being used to read and process actions on a screen, then once this is full, we will need to clear it before moving on. Similarly, if we split the process into smaller chunks, then this should give us the breaking points that we need, and so we should see some cognitive breaks that reflect the visual breaks. While the use of short-term vs longer-term memory is a working consideration for the design of the experiment, it is not directly testing this hypothesis – a full test would be beyond the scope of this study. The ideas of memory we have explored in Miller (1956) and Bruner (1986) as covered in Chapter 2.

These breaks in cognitive processing to allow a "reset" are termed here Process Breaks, and are important in terms of the design of applications, because if we can work with appropriate size chunks, this should help the "Flow" process, by not requiring breaks forced at unexpected places. In particular, the work of Kuhl, Dudokovic, Kahn and Wagner (2007), as discussed in Chapter 2 above, cover the importance of forgetting those things that are not important, to help focus and concentration on the important things. However, the process of why this chunking may work, or may produce differing results, is not within this scope, not least because there is still a lot of work on the functioning of memory (see Cowan, 2016 and Bearman and Jones, 2016, where the process of forgetting is also touched on).

Method

Participants

A total of 33 participants started the experiment, although only 18 completed it. The researcher was present when some participants were undertaking it, although this was not necessary, and the feedback from them was that the final phase of the experiment was extremely frustrating, which probably explains why so many did not complete this one in particular. It should be noted that 30 did complete the first two phases, and so comparatives can be done between these two. Participants were from the UK, the majority male and with a range of ages. The precise details of these demographics were not gathered.

The participants were drawn from the researcher's contacts, friends, family and associates. There were no restrictions on the participants, as the actual individual tasks are simple and straightforward additions.

Materials and apparatus

The experiment was written using Visual Studio in C# and asp.net and delivered from a commercial hosted environment. The coding made extensive use of AJAX processing, to asynchronously report to the server the times for each mini-task without interrupting the processing of the task as a whole, meaning that the participants' engagement was not disturbed by the measurement process.

Each phase of the experiment consisted of 25 very simple and straightforward arithmetic calculations, not requiring any undue skill. The intent was that they should all be identical in difficulty and the number choices were made randomly. The only difference expected between specific results is in the presentation of the task.

Procedure

There are two main ways of breaking up a task into parts: either by visually making one large task look like multiple tasks (visual breaks) or by making tasks separated in time so that they cannot be completed in one go. Both of these are being explored in this experiment.

The number of questions was chosen to represent a page-long form on other web applications. The task for each phase is a set of questions that would not be easy for most people to complete in one go (see Figure 7-9 and Figure 7-10). In phase 1, this is presented as a block, with no visual clues to divide the task. This phase serves as a control to see how users handle large screens of input, and any timing irregularities would be the participant's natural process dividing a larger task. The expectation is that the only restrictions are related to concentration levels, although clear explanations of this are not required. The purpose of this is as a control against the other phases. The screen layout is shown in Figure 7-9.

The second phase processes identically, however the visual display of this phase is different, with breaks, horizontal lines drawn every 5 questions. This is providing a visual indication to divide up the monolithic task, with the expectation that participants will view this more as a group of 5 separate processes with 5 tasks in each. The display for this is shown in Figure 7-10. Aside from the addition of the lines, the processing and display of this phase is identical to the phase 1 stage. There is an expectation that for this stage, the visual break up will show a distinct pattern to the response times, over and against the control.

The third phase returns to the visual display of the control phase, with 25 undivided questions, appearing like Figure 7-9. However, this time, between each answer and the next question, there is an ever-increasing time delay. Note that this time delay is not included in the recorded time to answer, which is still between the question being displayed and the answer being entered. The time delay increases by half a second for each entry.

The code is written so that only one question is displayed at once, and once the answer is provided, it is verified and when correct, the next question is displayed. This means that there is only one question being shown for answering at once, to focus the participants. If the answer is wrong, the time is also recorded and marked as a failure. In this case, no new question is generated, meaning that each question has to be answered correctly before moving on. This means that some cognitive work is needed, and random answers cannot be entered. The calculations have to actually be completed correctly.

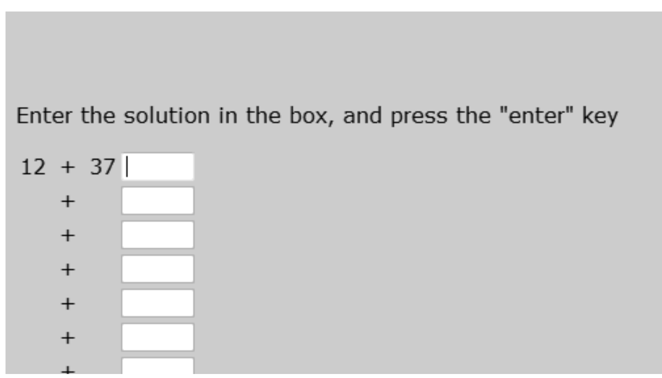


Figure 7-9- Process break experiment control screen, and time break screen.

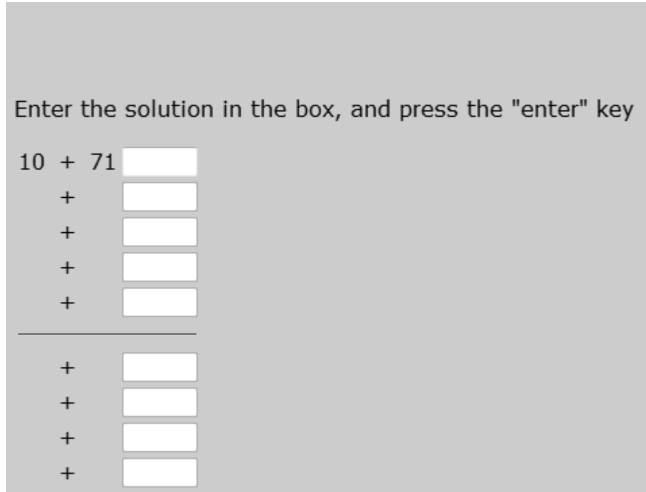


Figure 7-10 - process break screen with visual dividers

The participants were pointed to the web site to perform the experiment and asked to complete it. Some brief instructions were provided at the start to ensure that participants knew roughly what they were doing. No problems were reported in understanding the tasks, although no explanations were given as to the differences between the three versions. This was so that the participants would not modify their behavior for each but would respond to it as it appeared.

Results

Phases 1 and 2

The results returned are simply a set of times, and so some form of processing is required for these to provide any information. The information that we are interested in is the shape of the response graphs, in particular, the periodicity and the height of the peaks.

The processing on this data is in two stages. Firstly, to identify the peaks, which was done by highlighting the results that were higher than the 2 results on either side. Once these were identified, the second stage was to quantify the mean periods of these peaks, to produce a periodicity value for each participant and for each phase. Additionally, the depth of these peaks can be calculated by taking the difference between this peak value and the minimum values both before and after. Taking the mean of the before and after values, a figure can be arrived at for the height of these peaks, which should indicate the significance of them.

The raw data are shown in Table A4-62 and Table A4-63.

Phase 3

The third phase requires a different approach to analysis, because the expectation is not of periodicity but of increasing times. Therefore, the important figures are the rate of increase, identifying where the times increase and do not then reduce again. These raw figures are shown in Table A4-64, where the mean rise is included as well, to identify how much the time increases each step.

There is additionally a column indicating the question of maximum increase, which relates to the question number which prompted the maximum time increase across the experiment phase, to identify at what point the most significant change happened.

With and without dividers

The important figure from the data that needs to be calculated is the **periodicity**. As the participants perform the task, they will, at various points, take a brief break (because the pages of calculations have been designed to be too much for most people to manage in a single go). It is the timing of these breaks that is significant.

The raw data captured is the time to perform each calculation from the point at which it is displayed to the time when a solution is presented. These figures are irrelevant as absolute values. Some people will be quicker or slower than others, and there might be discrepancies in the participant’s environment that affect the absolute times. However, the comparative times are significant. The ranges of figures are plotted and there are peaks identified. The periodicity is the number of calculations between peaks (the time for these can also be used as the **periodic time**).

Exploring this data for normality, we find that the Kolmogorov-Smirnov significance is 0.29, which would indicate that the data does not follow a normal distribution. However, the Shapiro-Wilk test does give us a significance of 0.169, so the t-test is a valid interpretation of the spread of this data. Analyzing the outliers, there is one exceptional figure which, if removed, brings the Kolmogorov-Smirnov significance to 0.1 (the Shapiro-Wilk is unchanged). The t-test (t-value of 26.3) gives us a mean difference of 5.19 (±0.40). This gives a snapshot of the variation. If we look at the continuous change as a sparkline, we see it does find a level fairly early  , and this figure provides us with a good indication of the base that we can make comparisons against for the other tests.

The same processing applied to the data with visual breaks included gives us a comparative value. The Kolmogorov-Smirnov and Shapiro-Wilk both indicate that these are not normal distributions. Removing outliers (3 extreme values, 2 high, 1 low), we find that the Kolmogorov-Smirnov significance rises to 0.20, and the Shapiro-Wilk rises to 0.258, so we can now apply the t-tests (this results in a total of 6 missing values). The t-test (t-value of 33.3) gives us a mean of 4.52 (±0.28). The sparkline is , which once again shows a clear platform in the middle but also shows the outliers.

This figure is definitively lower, but we need to identify whether the differences for individuals are significant. An independent samples t-test produces a t-value of 2.65, significant at the 0.05 level. This indicates that the difference between individual scores are significant, while noting that the one-way ANOVA doesn't produce a significant difference because the individual differences are swamped in the range of values.

We have also recorded the error data - that is, the number of times that the participants entered the wrong solution. As we have no indications as to why they were wrong, the results can only indicate the level of error response. For phase 1, the level of errors is very low, with only questions 2, 4 and 14 producing a few errors. It may be that 2 and 4 are part of the learning process for the trial. The majority of these were from a few individuals who had a significant number of failures (7 and 5 being the highest). Analysis of the individuals' responses shows that most respondents answer all of their questions correctly. The total overall mean number of attempts for phase 1 is 1.054, indicating a very low level of errors, especially when taking into account the two abnormal entries.

The second phase errors show a similar pattern, with one abnormal entry recording 9 attempts but the remainder showing extremely low levels of error. The overall mean number of attempts for this phase is 1.055, similar to the phase 1 value. While a pictorial representation does indicate that one participant had a few problems, there were no other patterns of errors, and therefore nothing significant can be ascertained from this.

<u>With timed delays</u>

The requirement for this data set is similar to the data set for the visual breaks, in that we need to determine whether the periodicity is valid. The same approach is taken as above, and the core data is shown in Table A4-64. In this case, the data does not level off  but rather continues to increase, indicating that there is no clear level at which we can take a representative value.

Applying the same approach as before to this, we find the Kolmogorov-Smirnov significance is 0.19 and the Shapiro-Wilk is down at 0.026, which indicates that the data is only just hitting normality – as is indicated by the sparkline above. However, the t-test can still be applied, and it returns us a mean of 3.89 (±0.52) with a t-value of 16.2. This is not quite as clear as the other entries but is still a good indication that this mean value is significant.

When we use the same approach and compare these values to the base values using the independent t-test, this produces a t-value of 2.3, significant at the 0.05 level. Once again, this is a significant value, a significant difference for the individuals concerned. Again, the individual differences are swamped in the overall variation because individuals are each working in their own way. A curve fit is show in Figure 7-11 (note that no curve fit was shown for the breaks because there was no clear curve that fitted the pattern). The inverse nature of this would indicate that those who could concentrate on the initial experiment, and so complete it faster, are the people who find the timed break experiment most disruptive to their working. However, the small sample size makes this difficult to draw more than speculative assumptions on this.

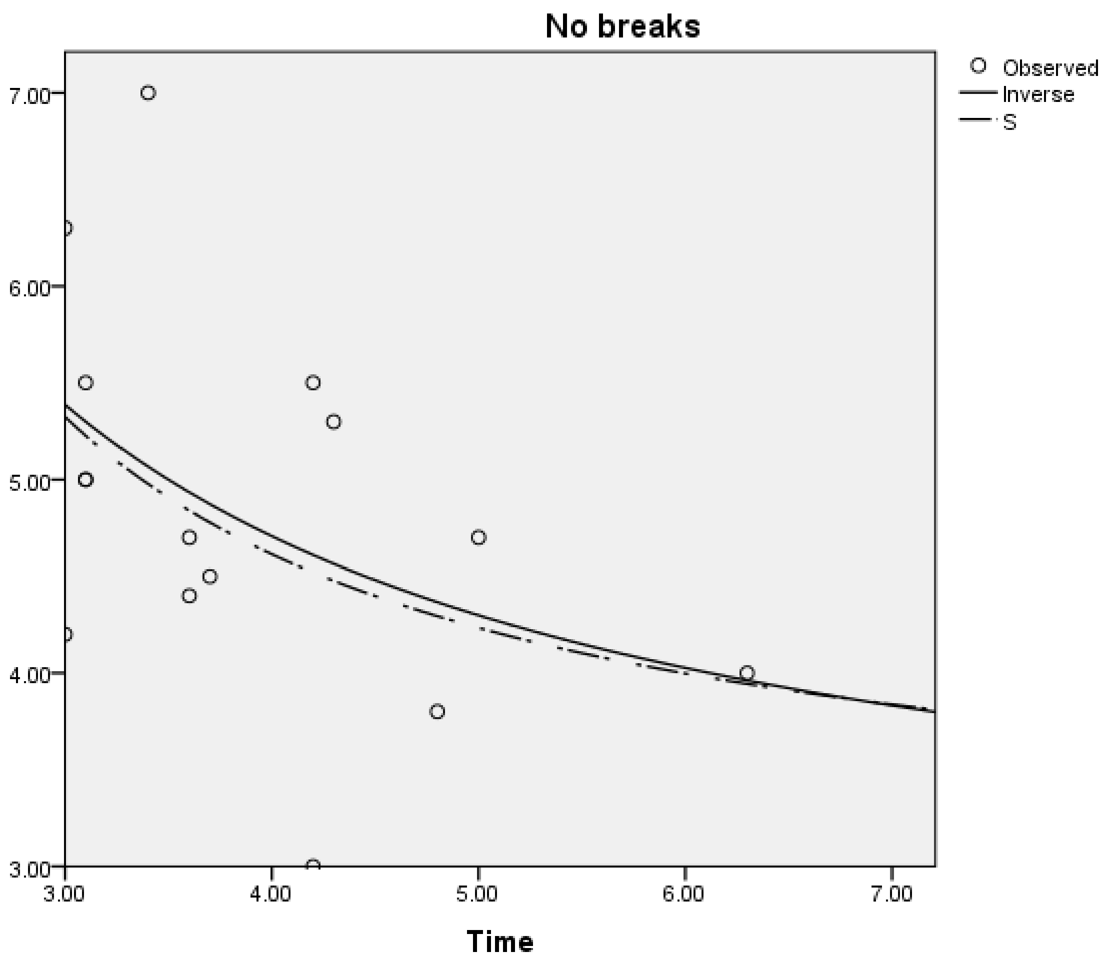


Figure 7-11 - Curve fit for control group (no breaks) against timed breaks.

If we now take a look at the level of increase that we found in this phase, the mean increase is 8.3s, so at each rise, on average, the participants take 8.3 seconds longer for the remaining questions. If we consider a mean of 3.2-3.6

between the peaks, which represents an increase in the delay of 1.6-1.8 seconds (half a second for each step), this indicates that if the delay in processing is over 2 seconds, then participants are liable to take another 8 seconds to return their focus to the task and process. Delays of this sort (around 2 seconds) therefore do appear to cause significant ongoing delays because of the interruption to the process flow.

Survival analysis.

Can we do a basic survival analysis on the data, assessing how people dropped out of the task? To some extent, we can, although it is more difficult with the smaller number of people and is the challenge of the final section, where many dropped out. What we find on a basic analysis is as below:

Phase 1: 34 participants, of whom 2 dropped out before the end.

Phase 2: 30 participants, of whom 2 dropped out before the end.

Phase 3: 24 participants, of whom 12 dropped out before the end.

What is clear is that there is very little dropout except at the start of another phase. We see 2 participants stopped after the first phase, while 2 dropped out in middle; similarly, 4 dropped out at the end of phase 2 and 2 during the phase. The majority dropped out during this final phase. What this would indicate is that, while people did drop out, it tended to be at the end of a phase (or, most likely, at the start of a new phase). There seems to be very little else that can be obtained from this analysis.

Discussion

The data indicates that there is some difference happening with changes in the experimental conditions. Having identified a control set of figures, we have discovered that with visual markers to indicate breaks, these do tend to drive the point at which participants pause. Additionally, there is some indication that by pausing sooner, the entire task seems to be achieved quicker.

When the time delay is introduced, the performance drops off, and the time to complete the task is significantly higher (having subtracted the extra delay time included). Delays of this nature break the concentration of the participant, and they then have much longer before they can continue the process. In terms of a flow hypothesis, their flow is broken significantly, so they have to readjust before returning to the task at hand. The frustration is shown in the high dropout rate for this phase.

The two phases that we can compare to the control do produce some interesting insights. Phase 2, with visual breaks, showed a reduced period, indicating that the visual indicators do have an effect on the processing of the questions. We can surmise that by dividing the display, the participants largely processed each group separately. Exactly how these match with the 4.3 periodicity would require more investigation. The dividing lines clearly have an effect, but it is not yet clear precisely what the effect it. While the line might appear to some to **divide** the groups, to others, it may serve to **merge** two sets together, and grouping is a very powerful visual clue.

The timed break phase indicated that time delays within the processing do have very significant impacts on the performance of the participants. In fact, by increasing the delay by less than 2 seconds, the time to complete the task increases by 8 seconds, which is substantial. In terms of commercial sites, this may not be a major problem, although users may decide to go elsewhere. For work-based tasks, where participants may be processing this task for significant portions of the day, adding 10 seconds onto every task, when the tasks are taking an average of 8.8 seconds (mean time taken from phase 1), is significant. Half of the time is being wasted.

In terms of the hypothesis, this does seem to support the third hypothesis "**A frustration experience can have a statistically significant impact on a participant's task-completion experience.**" The control phase indicates that there is a periodicity to the processing of this type of data. There is further work to do in terms of identifying what the differences are (the simple visual break lines were introduced as the most straightforward indication) and what the implication of other visual markers are. Using the Gestalt principles (see Few 2006, discussed above in Chapter 2), these could each be tested against a control environment to identify which aspects have the most impact. There would also be further work to explore the reactions of participants to this, what types of division work best for people. These aspects are beyond the scope of this study.

The timed phase 3 showed substantial increase in processing time, indicating that some frustration was being experienced. Watching some participants' action, this stage did also show some substantial frustration with this version of the task. The rapid drop off of successful participants for this phase suggests that this frustration is more widespread. Timed delays do appear to increase frustration quite substantially. In terms of design of computer applications, this means that ensuring specific tasks are completed without delays would appear to be significant. The recommendation from this work is that task completion should be achievable without delays (for example, in loading of new web pages), and, where appropriate, with the total task visually divided into sections.

Additionally, further work that would prove very profitable would be explorations into the physiological basis of this frustration and processing. The initial working idea was that the limit came based on the short-term memory storage; however, this was not the hypothesis being tested, and it was merely used as a scale to provide the visual divisions in the second part of the experiment. There is continued work exploring the functioning of memory (see, for example, Missaire et. al. 2017), meaning that the more simplistic (while still usable) ideas of Miller, while still serving as descriptive explanations, are being seen as less functionally accurate. As more functional explorations are developed, this is a valuable practical exploration of these – using a better understanding of the functioning of the brain to make user computer interaction better.

This experiment has shown that there are user decisions that can impact on the frustration experienced by users of a system. This implies that getting the interaction correct is a critical aspect of a user experiencing flow or frustration with a computer system. Next, we will undertake an experiment to look at the point where frustration turns into failure and the impact of this on the wider usage of the system.

Frustration Point experiment

We have looked at one experiment that identifies some of the visual elements that enhance or lessen the flow experience. Now we will look more at the explicit frustration direction, and see if we can identify a frustration point – that is, the point at which participants will become too frustrated to continue. This is to demonstrate the third hypothesis, that a **frustration experience can have further impact** on task achievement.

The purpose of this experiment is to quantify at what point the frustration with attempting certain tasks becomes too much, and people give up. Additionally, some insights are being gathered into what happens at the point of abandonment, how people respond at that point, and what it appears to be that triggers that abandonment.

Method

Participants

It was intended initially to video some 20-30 participants for this experiment, however the practicalities of organizing people and equipment into defined locations reduced this to 8 videoed participants. The participants were drawn from students and staff at two institutions where facilities were made available for this videoing.

The videoing approach was chosen because it was hoped this would provide more information than the simple timing data being automatically gathered, for further analysis of potential “rage” incidents and to see whether the body language would convey more about the approach to the questions.

The experiment was then adapted to be a self-administered task, available for access over the internet. This produced an additional total of 12 participants, 10 of whom completed the entire task. These participants were drawn from friends and associates of the researcher.

Materials and Apparatus

The experiment was put together as a c# web application with variable task list, meaning that the individual tasks being asked for could be changed while maintaining the harness intact to record the process. For each experiment, the set of tasks was manually selected or modified, across two different web sites, with around 5-6 tasks per site. This was intended to give sufficient tasks to provide a spread of information, but not too many that the overall experiment became long and participants became bored, which would be another factor to take into account. In initial tests, the time to finish the tasks appeared to be around half an hour. Links were provided to the web sites, so there was no necessity to search for them or to search the web as a whole for answers.

A decision was made to build a framework from scratch rather than utilising an existing one to ensure that all factors are controlled – all aspects of the framework are under the control of the researcher. Given that this was for timing interaction, it was important that the timings were not interfered with by other aspects of the framework or the process of loading the web pages to the client. For the same reasons, the page was explicitly designed simply (from a technical perspective) to ensure that page rendering was not impacting the time taken.

Choice of tasks.

A set of tasks were identified manually on the sites to be used. These were large and commonly used sites, which the participants may have been familiar with, although this was not significant. The tasks were manually ranked into 4 **challenge levels**, defined as follows:

1. A task where the solution could be found by using phrases in the definition entered into the search box of the site or where they appeared on the home page, possibly with a link. This might include a book title when looking at a book web site.

2. A task where clues in the description can be used for navigation around the site, eventually leading to a list that contains the required item. Additionally, this challenge level would include items where a search produces a substantial number of responses and the required response is not on the first page of results, requiring either a more advanced search or working through multiple pages. An example might be a finding a biography of a popular author that doesn't contain the author's name from a book web site.

3. A task where some other process is required to achieve the task or very significant navigational work is needed. It would include where a site needs to be logged in to or registered with before any searching can be done, or where the navigation route to be taken is not clear. It might include situations where an item is being sought, but it is not clear what the item actually is and so where to navigate, meaning that some additional searching may be needed to identify the right route to take. Any task which has another required prior task is included within this level.

4. The final level of task is one which is impossible within the parameters provided. Trying to find products that are not available, or information that is not provided are included here. An example would be trying to

find Melatonin for sleep disorders on a pharmacy site, as this is not allowed for sale in the UK. It is important that the task definitions here are **reasonable**, in the sense that they are tasks that appear as if they might be possible.

These levels were chosen as a simple approach to identifying complexity of task. There are more complex approaches that could be taken (for example White and Livonen, 2002), but the requirement here was for a few levels so that they could all be covered in an experiment that would last less than 30 minutes.

Pilot studies for this experiment indicated that this was the right number of levels to use. These challenge levels are also significant for a wider view of application design: all of the tasks that a user will perform on a particular application will fall into one or other of these challenge levels. For an application of any sort to be highly accessible and user-friendly, a high proportion of the tasks expected to be performed against it should fall into the first two levels. Some tasks (buying from an e-commerce site, for example) may fall into level 3, but if they too can be level 2, then it can improve the customer experience (see Spool, 2009, and Krug 2000). If there are reasonable and justified tasks that fall into level 4, then the user experience will be negative. In the previous experiment, we noticed that for many participants, finding appropriate and relevant help at a point in time was a challenge level 4 task, and the perception of the products was thereby damaged.

In all cases, the tasks are framed as **reasonable search requests**. This means that they are more or less vague product descriptions, or information requests, that someone might be attempting to resolve on the site. So this may be a very specific product definition, received on email (a level 1 request); a remembered conversation (“you can get rompers like this in blue on rompers.com”), which may be a level 2 or 3 request; or an expectation (“I will get the address off their web site”), which should be a level 1 but may well be a level 4 if the particular company does not advertise its location. To ensure that all requests are initially treated by the participants as valid, they should all have the appearance of reasonableness.

These challenge levels are strictly ordinal, reflecting increasing complexity, and a set of data is only relevant to one specific application. Comparing the performance of a challenge level 2 task on one application against a challenge level 2 task on another application does not yield any valid information. This is because the applications on which these tasks are being done are different, and so there is no clear baseline for direct comparison. It is possible to define tasks that are comparable, but the challenge level alone is not sufficient to achieve this. What would be more appropriate to compare would be the number of requests tackled on a particular application that are the lower levels or the proportion of requests that are challenge level 1. Within a specific application, across redesigns, these proportions can be assessed to ensure that they do not increase. For this experiment, however, the significance is that each site's data will be compared to itself and not to other sites. It is not about ascertaining which of the sites has a better user experience but about identifying the process of users attempting to achieve various levels of task.

Procedure

Recording information

As the tasks were undertaken, the time between starting the task and ending it (which is also the start of the next task) was recorded, along with the task indicator, so that an XML import of the timings against the various tasks (and their challenge level) could be taken. Additionally, comments about the performance of the tasks were taken. In later iterations of the experiment, the comments were codified and taken after each task; however, the core timing information is identical across all versions of the experiment. In some cases, where it was practical, the participants were videoed while performing the tasks to gather further information about their reactions and responses. However, as this was only possible for a small number, it is considered as additional insight rather than core data feed. It is critical that some of the participants were videoed to obtain this insight and see occasions of “web rage” that may occur.

Tullis and Albert (2008, p. 89) explore a concept called “lostness”, which is of relevance to this experiment. More details on this idea are in Smith (1996), and it relates to how suboptimal a particular participant is in finding the information that they are after. Their methods of recording and measuring participant’s performance are very similar to the ones being used in this experiment. They were very specifically comparing an actual participant’s path to completing a task against their defined “ideal” path. The reality of most applications today is that some of Smith’s assumptions are no longer appropriate. In particular, the idea that there is only one “best” way to achieve a goal is not so relevant to modern software. This is partly because counting keystrokes and mouse clicks is only one part of a modern interaction, and simply minimising these will not necessarily produce the quickest route for a particular participant to reach the goal.

This experiment is measuring the time taken to get to the end, irrespective of the path taken. Some people will prefer and naturally use site navigation as their starting point, whereas others will use search facilities. For some people, keyboard shortcuts will be the most important route, whereas others will prefer to use the mouse. If participants are utilising the route that they are most comfortable with, they will probably achieve the tasks quicker than if they are being forced to use different approaches (once again, this is a flow issue, that the most natural approach for an individual will make the interaction less invasive).

Tullis and Albert (2008, p. 94) also discuss the concept of **learnability**. This is the change in performance of specific tasks as the participants become more familiar with the tasks or the application. There is an important insight here that as the tasks within one site progress, there should be some learnt behaviour. This may mean that later tasks are performed quicker as the participants learn the peculiarities of the site. For this reason, the tasks were ordered with a variety of lower and higher challenge levels. It is not random, as this would make the assessment of the effect more difficult. By doing this, the impact of any learned behaviour (which is not expected to be substantial anyway, as the total time on a specific site should be 20 minutes or less) should be ameliorated.

For the videoed aspect of the experiment, participants were set up with a laptop to perform the experiment, and a camera was set up behind a one-way mirror to record the interaction. A brief introduction was given, including explaining about the camera, and a consent form signed before they were left alone with the experiment.

For the distributed version, the introductory text was enhanced to provide more clarification, and it was sent out by email and similar links. These participants had to understand it as it was presented, and there were no responses indicating that any problems were encountered.

For all of the tasks, the participants were told clearly that the important aspect of the experiment was the **process of attempting** to complete the tasks rather than the **actual success rate** of completing them. This is important, as they were not told that some of the tasks were not achievable. Believing that all of the tasks might be possible if they try hard enough would make the time for completion significantly higher. At the same time, they were not put under pressure of success and knew that they could abandon the tasks at any point. Continuation of a task beyond a reasonable point was down to the participant's decision, not the experimenter's demands. One videoed participant was reminded of this after trying one task for a substantial time.

Results

The expectation from this experiment was that at some challenge level, the attempt to achieve the task would be abandoned. In reality, the challenge level 4 tasks would have to be abandoned, as they were not achievable. The question was how long it would take before abandonment and what the effects of this would be. It was hoped that through this analysis, a **frustration point** could be identified, which is the point at which a task is abandoned when using a specific site. The later tasks would also indicate whether the frustration point alters depending on the previous task achievement.

<u>Videoed responses</u>

The videos served as backup information for the observations, as the participants were directly observed and notes taken of their behaviour. The combination of these two served to provide a fuller picture of the participants' engagement with the tasks at hand and served even when the video evidence on its own was not perfectly clear. These results are therefore a combination of details from the videos and manual notes.

The process of identifying participants and organising for them to be videoed took some time, which resulted in the data for the experiment having to be reworked to an extent. Some items being searched for were no longer available, and so the tasks had to each be reconsidered. This means that the timed information gathered from the videoed participants cannot be directly compared to the timed information from the other participants or even from the two stages of the videoing. Each videoing session did not produce sufficient results to warrant statistical analysis. This means that, while the data obtained is valuable, it is not possible to compare videoed statistical results with non-videoed statistical results, although internal statistics can be used based on individuals' responses. The videos themselves were watched through by the researcher, and aspects like demeanour, position in the chair and any facial expression were noted along with the point within the experiment that they took place.

There were a number of significant responses shown in the videoed experiments. One of the first was that most of the participants would follow a similar route to initially trying to achieve a task. This involved reading the task details and copying or typing the significant words into a search box on the site once they had found this (one site did not have a search box immediately visible) and assessing the results. For challenge level 1 tasks, this approach would generally find a solution fairly quickly. For level 2 tasks, this was often a good starting point, at least to demonstrate that some further research was required to find the solution.

A number of the participants were very dogged in their attempts to achieve each task. This may have been because they were under observation or just because they were anxious to demonstrate that they could achieve the tasks (and not have to admit failure). In some cases, they would continue way beyond what might normally be expected, spending up to 10 minutes on a single (level 3 or 4) task. While it may be seen as a positive that they were so determined, there was also significantly increased frustration at trying to achieve these. Follow up work would confirm this, including a whole set of study on *why* people continue pursuing tasks for as long as they do. This is not within the scope of this study.

When participants found a level 2 task that did not find a straightforward resolution, many of them would return to the question page to check and verify that they had correctly read the requirement before progressing. This would imply that there is some doubt about the nature of the task, and they require the task definition to be refreshed before continuing. This can also be expressed that they fail to find the required results on a first attempt and need the reassurance that they are looking for the correct information. What is significant from the perception of flow is that this is breaking their flow in trying to achieve the task and so will prove expensive. Introducing this doubt appeared to mean that the participants reset, returned to the start, and tried again having ensured that they understood the requirements.

Another behaviour observed when the initial search failed was manual searching around the site to find the information required. This is not a structured navigational search, as might be expected, but a more random site-map type search. They tried to identify what is available on the site and select from all of this what might be the most productive route to take. While doing this, they would often return to refresh their memory of the task, which is a break in their flow. This is a significant finding, as there is often discussion about the best way to organise navigation for a site (or any application for that matter) to provide users with what they require. This experiment would indicate that if users do use the navigation, they use it in a rather blunt way. What is more, the normal route that these participants were using was the search facilities. If this type of task-focussed processing is core to the site's functioning, then the most important feature is a very good search functionality rather than simple navigation. Challenge level 2 tasks are, it would appear, flow-breaking and so less productive, meaning that even these should, where possible, be enabled as level 1 tasks.

It was instructive to observe the participants when they had failed to easily complete the task (that is, when it was not a challenge level 1 or 2 task). At this point, a number of the participants would withdraw at least briefly from the screen and break their eye contact with the screen. In more serious cases, when they were presented with challenge level 3 tasks, they would sit back in the chair for a moment. They would often return by rechecking the task they were trying to complete, as if they were stopping everything they had done and starting afresh. In terms of task completion, this is a substantial break in momentum. If they had achieved flow in searching, the frustration of not finding the solutions broke that completely. In some cases, the frustration point was clearly indicated by a shout or throwing up their hands. These actions are clearly very negative when it comes to task achievement and might, in other situations, involve the user leaving the task for a while to make a drink or suchlike. Having said this, there was no clear anger demonstrated, but this may have been due to being videoed by a stranger. In a number of the experiments where the participant is known to the experimenter, more aggressive responses have been seen. This would clearly indicate that **computer rage** is a valid feature of some software application use. This is the point where the participant feels substantial anger towards the software that they are using.

The final observation is that once a participant has had to give up on a task, they become despondent and are less generous of the particular site (and the entire experiment).This means that they are more likely to give up with further tasks on the same site, something even more marked in those who spent longest before giving up. This **despondency** is an important aspect to explore because it will affect the participants' further usage of the site, possibly well into the future.

Unsupervised responses

This aspect of the study was a web-delivered, self-administered test, automatically providing timing feedback for analysis. Some changes were introduced into this experiment as noted above to ensure that the responses were able to be analysed clearly. A total of 24 responses were gathered, although some did have significant problems. Some of the experimental data is not viable; however, there were 12 participants who provided a full set of data from site 1 and 10 from site 2, giving us some statistically valuable information.

The raw data for sets of task against time spent are shown in Table A5-65 and Table A5-66. The times shown are in seconds, rounded, although they were all recorded in milliseconds. Applying correlation checks on these, there is very little correlation found in these, reflecting the fact that different participants would take different times for the same task, dependent on their persistence. We see the same pattern in both site samples.

In order to identify any patterns in this data, we need to integrate the other information we have, about task challenge level, participants response, and whether they abandoned the task. The questions have each been assigned a challenge level, based on the expected route to achieving them. By dividing the time taken by the challenge level, it is possible to gather whether there is any real correlation; that is whether more challenging tasks take longer, for a specific participant.

Exploring the data, we find that the Kolmogorov-Smirnov and Shapiro-Wilk tests indicate a reasonable normality fit. However, the nature of the data makes it exceptionally difficult to interpret in a statistical way. Some work on the data, restructuring it, does provide us with a way to analyse it. We see the estimated marginal means in Table 7-45, which show an increasing mean with the difficulty level, as expected.

Table 7-45 - Estimated Marginal Means table from frustration point experiment.

Ranked difficulty of task

Dependent Variable: Time Taken to complete task

ranked difficulty of task	Mean	Std. Error	95% Confidence Interval	
			Lower Bound	Upper Bound
1.00	81.146	27.689	26.021	136.271
2.00	82.143	35.532	11.403	152.883
3.00	132.148	24.488	83.396	180.901
4.00	155.845	24.488	107.093	204.597

What is significant here is that the 95% confidence level interval is quite substantial compared to the mean value, at least for the lower levels (reflected in the standard error values). We can take this further and apply a Tukey HSD test to give comparisons between each pair of task levels (see Table 7-46), showing more detail on the comparisons. It is clear that a number of the entries have wide ranges and differences that are not considered significant. This should warn us that the conclusions drawn are not conclusive with this set of data, but give an indication. Further experiments would be needed to validate these findings.

Table 7-46 - Tukey HSD comparison of all pairs or difficulty ranks for the Frustration test experiment.

Multiple Comparisons

Dependent Variable: time taken to complete task

Tukey HSD

(I) ranked difficulty of task	(J) ranked difficulty of task	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1.00	2.00	-2.4619	44.00735	1.000	-117.9941	113.0702
	3.00	-61.9402	36.11074	.323	-156.7415	32.8610
	4.00	-85.6368	36.11074	.091	-180.4380	9.1645
2.00	1.00	2.4619	44.00735	1.000	-113.0702	117.9941
	3.00	-59.4783	42.58673	.505	-171.2809	52.3243
	4.00	-83.1749	42.58673	.215	-194.9775	28.6277
3.00	1.00	61.9402	36.11074	.323	-32.8610	156.7415
	2.00	59.4783	42.58673	.505	-52.3243	171.2809
	4.00	-23.6966	34.36522	.901	-113.9153	66.5222
4.00	1.00	85.6368	36.11074	.091	-9.1645	180.4380
	2.00	83.1749	42.58673	.215	-28.6277	194.9775
	3.00	23.6966	34.36522	.901	-66.5222	113.9153

Based on observed means.

The error term is Mean Square (Error) = 17124.038.

Using the means, we do, at this point, find that the graph shows a clear trend (see Figure 7-12 for the more difficult tasks. From this we can conclude that while more data would be required to support the thesis, we do have indications that more complicated tasks are taking longer, and, looking at the shape of the graph, the very complicated ones take a substantial amount longer.

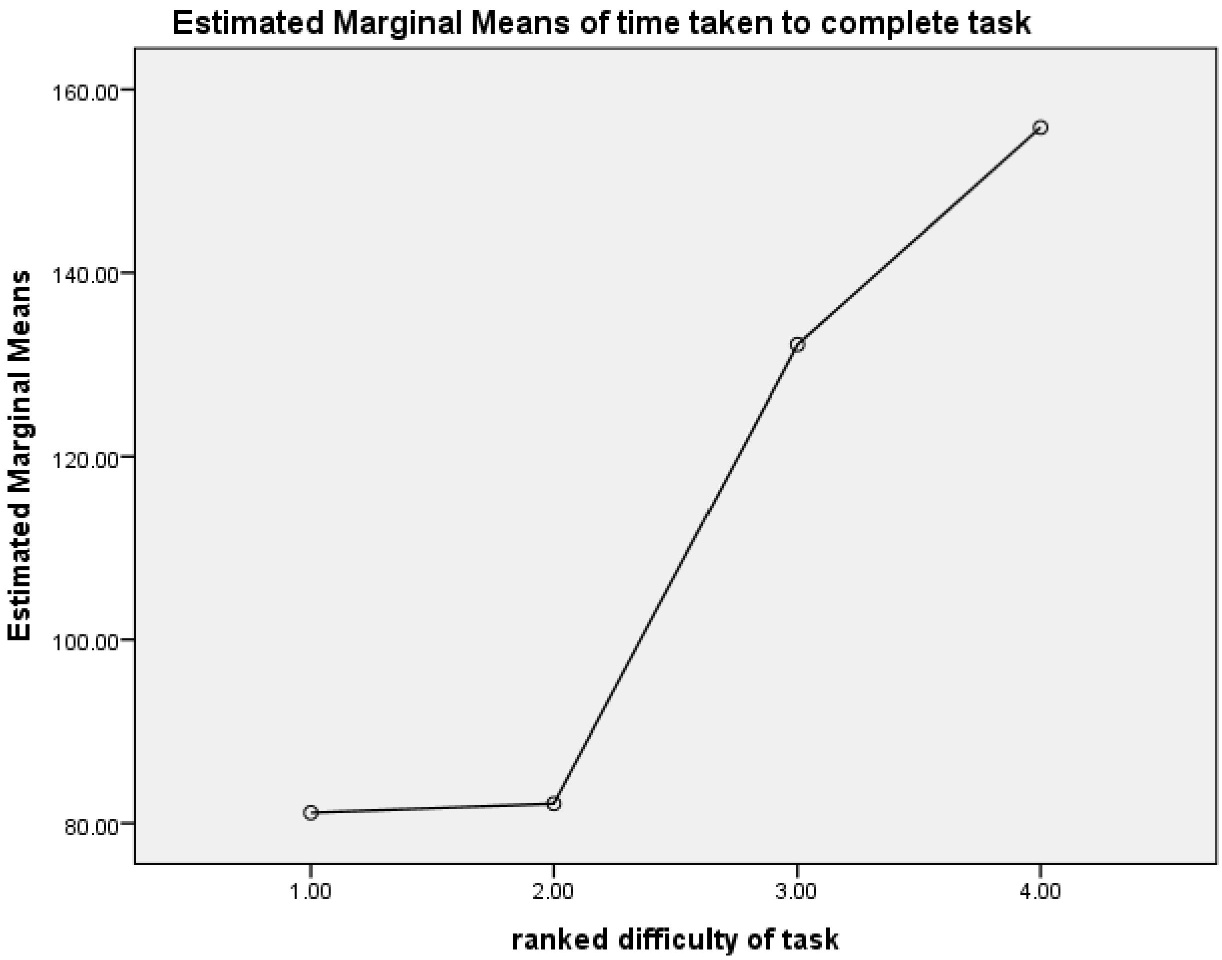


Figure 7-12 - Graph of means vs task difficulty, for frustration point experiment

Tasks at levels 1 and 2 do show a very similar timescale for completion. This makes sense, as these are both the types of query and task that many people are used to performing. It is the more complex ones – 3 and 4 – where we see a dramatic rise in the time. Because these tasks are taking more steps, more cognitive power to complete, we are seeing (as we did in the last experiment) that the time taken to achieve (or abandon) is much higher.

Can we identify any definitive results based on abandonment of the tasks? The mean time for an abandoned task is 81.2s, whereas for a completed task, it is 109.3s. What is more, the standard deviation is so high for these that taking definitive conclusions from these is risky. It is interesting that people, on average, took longer to complete a task than to decide to abandon, and we could conclude that some progress was being made, giving them reason to continue.

Looking at the raw data in more depth, there are some tasks (site 1, task 3 and site 2, task 6) where the response was significantly better than the expectation, which may indicate that the participants found ways of achieving them that were not planned for. There is no evidence that these were abandoned more than other tasks. Similarly, site 1, task 1 and site 2, task 2 returned particularly high values. Again, these were not especially highly abandoned. This indicates the participants found these tasks harder than expected. These discrepancies may be indicators of changes in the sites. Products being searched for that should be easy were removed or improvements were made in the search results. We can use this information to highlight the tasks that match their challenge level as more significant results because the setting and completion of the task seem to produce similar responses. We also need to be slightly wary of the results indicated, as these tasks may not have been at the right challenge level when the experiment was taken.

If we look at the challenge levels against abandonment, we see this information in Table 7-47, where the number of reported abandonments is matched against the challenge levels. The final row is the proportion of abandons against actual responses, divided by the challenge level, to ascertain whether there is a direct correlation here. The results do indicate that there may be some connection. A second order calculation was made, using the challenge level squared, as well as an inverse and inverse second order calculation, none of which indicated any closer correlations – in fact, the second order calculation indicated less correlation. Of course, because the challenge levels are ordinal values, they do not necessarily represent steps of complexity, so the resulting figures are, in themselves, meaningless. The purpose is simply to identify whether there is any form of correlation between more challenging tasks and abandonment levels.

Applying correlation tests to this data, there is no definitive correlation found (a coefficient of 0.275). Examining the data manually does show that is a slight indication of increased abandonment for more challenging tasks, but this is very slight. This would probably indicate that there are a range of reasons for abandoning a task, not just difficulty of completing it.

Table 7-47 – Frustration point experiment comparison of challenge level against abandonment counts. Challenge levels reflect the difficulty in completing the tasks, against the frequency of abandonment.

	Site 1					Site 2						
Question	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Challenge	1	2	2	3	4	2	1	1	2	4	4	2
Abandons	8	7	5	9	10	2	4	3	4	4	4	3

The final assessment on this data is a question of whether there are patterns in the behaviour shown. Specifically, there are three questions to ask here: Are participants taking more or less time after they have abandoned a task? Are participants taking more or less time on subsequent sites after abandoning a task? Are participants increasing or decreasing their time on a new site before they abandon? The first analysis is shown in Table A5-67, where, for a range of abandonments, the time of abandoned tasks along with the difference in average time before the abandoned task and the difference in average time after the abandoned task.

In most of the entries, 15 out of 26, the mean time after an abandoned task is greater than the mean time before. In the remaining 11 cases, the time is reduced. This would indicate that for a small majority of cases, abandoning the task then reduces the time that the participants will spend on remaining tasks. However, if we limit this to tasks on the same site, while we lose a significant number of data points, we do see that the post-abandonment time is always less than the pre-abandonment time. This would indicate that the participants are spending less time trying to achieve a task once they have had to abandon one, as expected. We also notice a significant number of negative values in the outside two columns, which are indication that the time spent trying to achieve a task before abandonment is often higher than the time spent on other tasks either side. On the other hand, the positive figure indicates that tasks that do not produce possible solutions within a reasonable time may be abandoned by other participants. We cannot draw any definitive conclusions from this, as the spread of data is too wide – both of these effects are probably coming into play for different participants.

Can we identify if participants are taking more or less time on subsequent sites? Where they completed both sites, we show the results in

Table A5-68. The majority (11 over 7) are taking less time on the second site than on the first, and the mean of the differences (55.4 against 24.3) suggests that there is a tendency to spend slightly less time per task on a second site, once a task has been abandoned. However, we need also to compare this to the differences among the entire dataset, where we actually find that most participants (13 against 7) took longer on the second site than on the first, and the mean difference is higher too. This would tend to support the assertion that once a user has had to abandon a task, they will probably be less tolerant of further sites and tasks.

Finally, are there any patterns in the first task of a new site after abandonment? That is, is a new site given more time than the old site after a task has to be abandoned? The data is very scant for this, as it only applies to participants who have completed both sites and abandoned a task other than the final task in the first site, to ascertain whether they are taking longer or shorter on further tasks within the site against the first task in the new site. The evidence from the two occasions where this occurs is that the first task of the new site is not getting more time – less in fact. However, because the dataset is so small on this, no inferences can be safely drawn from this information.

Hi-lo and survival analysis

Applying a simple hi-lo analysis to this data, comparing the results from the two phases, what do we find? The results of a basic analysis are shown in Table 7-48. Unfortunately, there are comparatively few participants, but even with this number, there is no clear correlation, no particular segments that stand out. In fact, moving the mid-point divisions produces a fairly even spread. This would suggest that the results from one site are not significantly impacting the results from the second site. In fact, the only quadrant that seems to be significant is the lo-lo quadrant, which continues to be slightly higher than the others. This would indicate that those who are quick in their completion or abandonment of the tasks in one site seem to be just as quick in the second site. Of course, the small sample size makes this difficult to substantiate beyond a general observation.

Table 7-48 - hi-lo chart representing hi and low response times for the two sites used in the frustration point experiments. Mid-point division is 90s

		Web site 2: lo	Web site 2: Hi
Web site 1	Lo	9	3
	Hi	8	4

What might prove more useful, although it is also limited by the small sample size, is a survival analysis on the participants. A summary of the point at which participants abandon the task is shown in Table 7-49. What we can see from this is that across the non-completing participants, the mean point of drop-out is just after question 5 – which is the end of part 1. This suggests that somewhere around here, the end of the first site or the start of the second, is where the participants are giving up.

Table 7-49 - Number of tasks completed by participants, for abandonment experiment. Mean response times, divided by completion status.

	All responses	Non-completers	Non-starters or completers
Mean completed	8.08	5.29	7.00
Std dev	4.61	4.16	3.68
Count	24	14	10

The non-starters column (actually those who continue beyond the first task) shows a slightly higher figure because a significant number (4) do seem to drop out after 1 task. Those who stay beyond one task but drop out by the end seem to drop out early in the second set of tasks. There is no evidence that there is a particular task that is frustrating participants, although there do seem to be a few who drop out at the very end. This may be a technical issue, for example a failure to record a final time if they conclude in a certain way. The total numbers of participants, however, mean that it is hard to draw anything from this.

Discussion

Have we identified a frustration point, where participants will give up trying to achieve a task, and be frustrated? The video results are clear that there is a point of frustration, which is where a task that appears straightforward turns out not to be and where the participant becomes irritated. There were clearly seen (in observation) two different breaks in process: where a participant has to return to the original task definition to confirm the requirements and where they come to a stop, retreat from the screen, and restart their attack on the task. The real question is whether these can be seen in the statistical analysis of the timings and feedback from the larger study.

Firstly, there is an indication that more challenging tasks do take longer, and tasks with a challenge level of 3 or 4 were abandoned by participants in a relatively short time. While it may not be surprising that level 4 tasks were abandoned (as they are impossible), the fact that level 3 tasks, which are possible but have distinct barriers to achievement, are also abandoned very nearly as quickly indicates that the participants are very ready to give up on tasks. This does indicate support for the first thesis "**The Flow/Frustration dichotomy is a valid and useful way of understanding the experience of using computer systems**" because frustration is being experienced by people attempting non-straightforward tasks.

Finally, the impact of the participants abandoning tasks is significant. It would appear that the participant's trust of a site is reduced when they have to abandon a task. The time they then spend on further tasks appears to be reduced, which implies that the frustration that occurs will persist for a while. In terms of the concept of flow that Csikszentmihalyi explores, it would appear that once the process flow is broken, it takes some significant time to regain this. The participants did not appear to regain this during the experiment. For task-orientated applications, many work applications included, a break in flow would appear to be a significant issue, and regaining this may take over 10 minutes and may require a break from the application.

These findings do reflect well the findings in Ceaparu, Lazar, Bessier et. al. (2004), who also found that:

> “Based on the data, it is clear that user frustration is a serious problem. The subjects reported high levels of frustration, as well as large quantities of wasted time. This wasted time comes at a cost in financial terms. In addition, increased levels of frustration can impact on how users interact with other people during the day.”

These findings match well with the findings from this experiment. Future work may be indicated by the likes of Portugal, Belk, Quintas, et. al. (2016) who use specially adapted devices to identify the sense of frustration. A biological approach to understanding this frustration – especially over periods of time – would be a useful future approach to identifying the instigators and results of computer use frustration.

This experiment would strongly indicate support for the third hypothesis, “**A frustration experience can have a statistically significant impact on a participant’s task-completion experience.**” The experience of struggling – being frustrated – in one task does appear to have further impact on later tasks and does have an impact on the later experience.

Chapter 8 - Conclusions and further work

In terms of our original hypotheses, have we identified support or refutation? Once I have explored each hypothesis and the support (or otherwise) we have identified (as the main achievements), I will look into the potential future directions of this research and finish with some recommendations for computer application work.

Main achievements.

Hypothesis 1.

The first hypothesis we set out to explore is that **"The Flow/Frustration dichotomy can be used as a measurement tool to understanding efficiency of computer systems."** This hypothesis was supported by the interviews that took place and the computer forum research that was used to provide further insight into this. This primary and critical research provided a lot of insight into how to quantify the flow and frustration experiences. It was significant that some participants indicated that the "bad" experiences would take longer than the good ones, indicating that frustration or irritation in computer use is an important aspect of the real-world use of computer software. However, they all provided positives as well, so this indicated that the dichotomy was a way of discussing the usage, even if not yet in full support of the hypothesis.

We found then that the result from the computer usage questionnaire was also supportive of the hypothesis by drawing more insight into the questions and the way that people interact with their software applications. The literature study into perception is significant here because the way that users perceive their software to be impacts significantly on their interpretation of it – people don't complain about software that they cannot see.

Although it has not been covered in this research, identifying an appropriate iconography for a particular user base is a significant part of making software applications usable and appropriate for their intended users. This relates to the concepts of flow because if a user does not have to change mindsets to understand what to do on the computer, they are more likely to stay in a flow state. Therefore, if the entire visual presentation to them is in an iconography that they already understand, they will be interacting with the software as if it wasn't there, and the dream of ubiquitous computing will be one stage further forward (see Wieser, 1994, for a little more on ubiquitous computing iconography). This concept of software applications that are invisible (because they are ubiquitous) can mean that problems are transferred to other things – something we may also see in this work, where frustrations with work tasks can be transferred to the software tools that are used. This is a weakness in the studies but one that does not invalidate them. Rather, it indicates that, as computers become more ubiquitous, it becomes more difficult to identify the actual cause of the problems identified.

The questionnaire investigating experience of flow provides a strong indication that the experience of flow is one that the participants experience. In terms of the hypothesis, this is a strong indication that people not only experience flow but also recognize the concept (in some sense) and so experience the positive aspects of a flow experience. This tells us that the flow aspect is definitely something that can be used to indicate positive factors of a computer usage experience, so it supports that side of the hypothesis. The evidence that the concept of flow is gaining some degree of acceptance may well be helped by this and be helping this. When we then look at the experiments into frustration, we find that this is an indicator of negative computer application experiences. Both the frustration point and process breaks experiments show that frustration is the experience when software applications do not respond or behave well.

Between these, it would seem that there is strong support for this hypothesis from these studies.

Hypothesis 2

Our second hypothesis is that "**Using this flow/frustration dichotomy for interfaces definition will produce financial benefits for organisations.**" The first work and leisure questionnaire did indicate that this was a valid division; the experience of work and leisure usage was different, and the individual's usage did not correlate between them.

The reason for this distinction was that, when drawing insights from other studies, it is important to understand whether the studies done are utilizing tasks that are work-based or tasks that are leisure-based. It should be pointed out that this relates purely to the tasks, not the actual environment being worked in – not least because these environments are becoming less easy to differentiate. Ludwig, Dax, Pipek, and Randall, (2016) talk of “consideration of the “situated” view of contextuality”, and they discuss more the issues of work and leisure time and space being more and more blurred. The context is no longer the physical location; the situation may transform anywhere into a work location or, presumably, a leisure location. For these reasons, the concept of task-based research may prove to be more useful in identifying differing expectations and usage patterns.

The financial benefits are shown because the efficiency and performance are improved, and the time spent using such systems is substantial (as shown form the questionnaire), so a small improvement will produce substantial efficiency – and so financial – benefits.

This hypothesis is largely seen to be supported and has driven the design of some of the later studies. However, it is accepted as a limitation of these studies that the distinction between work and leisure usage is a far wider topic than there has been scope for here.

Hypothesis 3.

Our third hypothesis is that “**A frustration experience can have a statistically significant impact on a participant’s task-completion experience.**” The questionnaire on computer usage gave some support to this hypothesis because the participants reported that frustrating experiences did have wider implications. However, the most support for this was from the Frustration Point experiment, where there was evidence that frustration in one task appeared to have further effects on later tasks.

Also very important, was the Process Break experiment, which gave some good qualitative evidence that forcing breaks in the processing, in particular by frustrating the flow experience, had an ongoing impact on later task processing. We can therefore state that this hypothesis is demonstrated.

The main achievements are to have found approaches to testing these hypotheses, and identifying that they are valid approaches to understanding user interfaces. Using the hypothesis of Flow, we can generate more insights into user interactions in a work-like environment.

Recommendations

This study has identified three aspects of user interaction – the three hypotheses under examination. These form the basis of the recommendations to be made.

Firstly, it is recommended that studies exploring user interaction address whether they are considering work or leisure types of interaction. These have been shown to be different, so studies need to identify whether this is a task-like, work style of interaction or a leisure style of interaction.

This division is not always as expected, and studies will happen that are apply across both, but there is a difference (see Csikszentmihalyi and LeFevre, 1989, where the surprising finding was that flow occurred more in work than in leisure experiences). While the distinctions may need honing – the ones identified here are working definitions – the reality of this distinction is important, and even where the environments may appear similar (similar software is used across them both), use may differ.

Secondly, it is recommended that the flow/frustration dichotomy is used as a way of understanding user experience of software. This has been shown to be a valid approach and so should be considered when assessing user experience. More specifically, the elements that produce frustration should be avoided, and the aspects that enhance flow promoted. However, as this differs between different people, this means that user interfaces need to be tested with people executing work-like tasks to identify whether the interface promotes flow or frustration. The importance of the task-based approach, simulating a more work-like environment, is important.

Finally, the fact that a frustration experience has a wider impact means that those purchasing software in work environments should consider the flow/frustration aspect of the software usage. This is not just because it makes those who are working feel better, although this is a laudable aim. It is also because there is a benefit in terms of longer-term performance and efficiency.

One area that has not been explored here is the feeling of the participant, but this would be valuable further work. The recommendation also applies to those using software in work environments, that their frustration may be due to software usage (although they may direct it in other ways).

Further work

There are developments that have happened since this work was started that are relevant and present opportunities to take this work further.

The first of these is the **gameful design paradigm**, which is the idea of bringing game concepts to non-game applications. In many ways, it is a logical route to take, as the time people spend on games is significantly higher than they like to spend on business applications but is a deliberate crossover from the leisure usage to the work usage and will have impacts on this research identifying the differences. This can be hidden, for example using game ideas to drive the design process (see Buxton, 2007, pp. 286ff), so treating the user's process through the application as a game process. In other cases, it is more explicit, where the actual process through the application is very like a game process, making the achievement of the task like the achievement of a game level. One example of the use of games for serious processing is found at fold.it. (See also Nacke and Deterding 2017; Smith, Gradisar, King and Short, 2017; Kaye, Monk, Wall, et. al., 2018, where the link between gaming and flow is explored.)

There is some further study to do to see whether this blurring of the work and leisure spaces is having a positive effect – in particular, whether it is making work applications more like leisure activities or simply taking work into the leisure space. It would be of interest to repeat the study but adding in more about the overlap to ascertain how well divided these spaces are.

Secondly, there has been a significant development in the managing of **Large Data**. The problems here are that this data is analyzed once and then lost, meaning that frustrations if the analysis is wrong are not easily rectified. This concept started with astrophysics but has become more significant today for social media data (see Kim, Lund and Dombrowski, 2013). Mostly, it is about presenting often ephemeral trends and has implications also for those on the receiving end of this analysis. Do the adverts or promotions that are presented in the midst of doing other things actually help the users flow or disturb it? If we apply a flow/frustration analysis to this, can these analysis results be used to be more effective?

There is also further research that would be productive in terms of the ongoing understanding of the findings from this work – what other, more practical, recommendations can be applied to help software development to allow for the flow experience? In the process break experiment, some aspects of this were identified, but a full assessment of the appropriate design paradigm for applications is outside the scope of this study.

Limitations of this work.

This study has limitations, some of which have been noted in the discussions previously. Firstly, it is important to understand that the ontological space this work has been conducted in is that of computing rather than psychology or biology. In particular, there would be an interesting and valuable line to be pursued from a biological perspective of the impacts of flow on a person. The approach taken here to identifying flow (and frustration) experiences are from a computer user experience perspective and so produce different information from other approaches.

There are also limitations in terms of budget and scope of this work, meaning that the sizes of the studies are smaller than ideal. It would be of interest to rerun them individually against significantly larger groups. It is expected that the results would be broadly the same but would provide more evidence and more subtlety in the results.

This work was also deliberately and explicitly focused on interactions through a mouse, keyboards and screen. These were the most widely used forms of interaction at the start of the study, although more recently phones and tablets have become far more significant in the interaction market. These devices have a different interaction paradigm, although still broadly WIMP-based, and produce limitations on the size of screen that can be used. They are mostly touch interactions, which mean that the size of displayed items needs to be sufficiently large to conveniently touch without overlapping. This makes for a limited space for interaction compared to a normal computer screen. The core principles of making interaction natural and flow-enhancing are still important and as valid, despite the fact that the practicalities of coding these are different.

One area that was not explored in this thesis. but is of critical importance to a proper understanding of Flow is the lack of a clear understanding of what Flow is as a physiological level. “People call it different names: in the zone, in the groove, on fire, in the moment, present, unified, tuned in, a high, a rush. All these terms describe a similar sensation.” (McCarther, S., 2018) indicate a wide range of acknowledgement of the idea, without any real understanding from a biological or neurological perspective. There is work to be done in exploring these aspects of the phenomena, which may provide more insights into both why and how it can be utilized for improved performance and user satisfaction.

Another important aspect of the Flow idea is that, over the last decade (while this research has proceeded), the concept has become far more common in the literature around UX. This is hard to reference, because Csikszentmihalyi has been referenced within the psychology area for longer. His ideas have progressed into other areas, becoming more accepted in a range of areas. This development is where this research fits in, as being very forward thinking when started, and very contemporary now.

Conclusions.

This study has identified the flow/frustration dichotomy as a valid and useful way to understand usage of software in a working environment. There is more work to be done to take this forward, however establishing this is a significant contribution.

This research has identified some areas that can make interface design more flow-enabled and utilized some tasks and measures that have contributed to this. While these may need to be improved, they provide a route towards this outcome.

In terms of validating the work of Csikszentmihalyi against a new domain of work-like software usage, this has been done and demonstrated. The fact that so many users considered that they experienced flow shows us that it is a valid area, but the wider research that indicates this is often tempered by the frustration aspect means it needs to be considered more. That this can have further, knock-on effects, has also been demonstrated, which should give financial incentive to producing more flow-enhancing experiences in software.

References

Anderson, D.R., Crous, F., and Schepers, J.M., 1996. Flow and quality of work life in a diverse workforce.

Angelini, L., Lalanne, D., van den Hoven, E., Mazalek A., Khaled O., and Mugellini, E. (2015) Tangible meets Gestural: Comparing and Blending Post-WIMP Interaction Paradigms in *Proceedings of the 9th International conference on Tangible, Embedded and Embodied Interaction* pp. 473-376 Stanford, USA

Argyros, A.A., and Lourakis, M.I., 2006, May. Vision-based interpretation of hand gestures for remote control of a computer mouse. In European Conference on Computer Vision (pp. 40-51). Springer, Berlin, Heidelberg.

Asur, S., and Huberman, B., (2010) *Predicting the Future with Social Media* Retrieved 1st June 2012 from http://arxiv.org/pdf/1003.5699v1.pdf

Bailey, K., (2016). How work can be made meaningful. *The Conversation*.

Bailly, G., and Oulasvirta, A., (2014) Towards Optimal Menu Design in *Interactions*, Vol 21, no 4, pp. 41-45

Bainbridge, L., (1983) Ironies of Automation *Automatica*, vol 19, no 6, pp. 775-779

Bannon, L., (1991) From Human Factors to Human Actors. In Greenbaum, J and Kyng, M., (Eds.) *Design at work: Cooperative Design of Computer Systems*. Hillsdale: Lawrence Erlbaum Associates pp. 25-44

Bannon, L., (2011) Reimagining HCI: Towards a more Human-Centred Perspective. *Interactions*. New York: ACM, pp. 50-57

Baudrillard, J., (1995) *The Gulf war did not take place (translated by Patton, P)* Sidney: Power Publications

Beaman, C. P., and Jones, D. M., (2016). The item versus the object in memory: on the implausibility of overwriting as a mechanism for forgetting in short-term memory. *Frontiers in psychology*, *7*.

Bedi, K., and Hrustek, N., (2010) Generic Icons – Yes or No? in *Journal of Information and Organizational Sciences*, Volume 34, Issue 1 pp. 17-26

Bernstein, B., (1991) Comform and experience with computing: are they the same for women and men *ACM SIGCSE Bulletin* Volume 23, Issue 3 pp. 57-60

Bertelsen, O., 1994, August. Fitts' law as a design artefact: A paradigm case of theory in software design. In International Conference on Human-Computer Interaction (pp. 11-18). Springer, Berlin, Heidelberg.

Bovier, P., and Perneger, T., (2003) Predictors of Work Satisfaction among Physicians *European Journal of Public Health*, Volume 13 Issue 4 pp. 299-305

Boynton, P. M., and Greenhalgh, T., (2004) Selecting, designing and developing your questionnaire *Bmj*, Volume 328 issue 7451, pp. 1312-1315

Brooker, C., (2011-12) in *Black Mirror*, TV Series, Channel 4 UK

Bradburn N., Sudman S., and Wansink B., (2004) *Asking Questions – a definitive guide to questionnaire Design*. San Fransisco: Jossey-Bass

Bruner, J., (1986) *Actual Minds, Possible Worlds*, Cambridge, Mass; London : Harvard University Press

Bruner, J., (2003) The narrative construction of reality in Mataes M and Sengers P (Eds.) *Narrative Intelligence* Amsterdam/Philaelphia: John Benjamins Publishing

Burton-Jones, A., 2014. What have we learned from the Smart Machine?. Information and Organization, 24(2), pp. 71-105.

Buse, C., (2006) When you retire, does everything become leisure: Information and Communication Technology use and the Work/Leisure boundary in retirement *New Media Society* Volume 11, pp. 1143-1161

Buxton B., (2007) *Sketching User Experiences* London: Elsevier

Cañamero, D., (1999) What Emotions are Necessary for HCI? In *Procedures of HCI:Ergonomics and User Interfaces* Volume 1, pp. 838-842 Lawrence Erlbaum Associates

Caminer, D., (2002) LEO and the Computer Revolution, in *Computing and Control Engineering Journal* Volume 13, Issue 6, pp. 273--280

Card, S., Moran, T., and Newell A., (1983) *The Psychology of Human Computer Interaction* London: Lawrence Erlbaum Associates

Ceaparu, I., Lazar, J., Bessiere, K., Robinson, J., and Shneiderman, B., (2004). Determining causes and severity of end-user frustration. *International journal of human-computer interaction*, *17*(3), 333-356.

Clemmensen, T., Nardi, B., and Kaptelinin V., (2016) Making HCI Theory Work: An Analysis of the use of Activity Theory in HCI work in *Behaviour & Information technology*, pp. 1-20

Cooper, A., (2004) *The Inmates are Running the Asylum* Indianapolis: Sams Publishing

Cooper, A., Reinmann, R., and Cronin, D., (2007) *About Face 3: The essentials of interaction design,* Indianapolis: John Wiley & Sons

Copeland, B., *A Brief History of Computing*, downloaded on 12th May, 2012 from www.alanturing.net/turing_archive/pages/Reference Articles/BriefHistofComp.html

Costikyan, G., (2002) I have no words & I must Design: Towards a Critical Vocabulary for Games *Proceedings of Computer Games and Digital Cultures Conference* pp. 9-33

Cowan, N., (2016). The many faces of working memory and short-term storage. *Psychonomic Bulletin & Review*, 1-13.

Crane E., Shami N., Peter C., (2007) Let’s get emotional: Emotion research in human computer interaction *CHI07 Extended abstracts on Human factors in Computer Systems*

Csikszentmihalyi, M., (1975) *Beyond Boredom and Anxiety* London: Jossey-bass Publishers

Csikszentmihalyi, M., and LeFevre, J., 1989. Optimal experience in work and leisure. Journal of personality and social psychology, 56(5), p. 815.

Csikszentmihalyi, M., (1997) *Finding Flow* New York: Basic Books

Csikszentmihalyi, M., (2002) *Flow: The classic work on how to achieve happiness* London: Rider

Csikszentmihalyi, M., (2006) *Creativity: Flow and the psychology of discovery and invention.* New York: HarperPerennial

Csikszentmihalyi, M., Khosla, S., and Nakamura, J., (2017). Flow at Work. *The Wiley Blackwell Handbook of the Psychology of Positivity and Strengths‐Based Approaches at Work*, 99-109.

Curtis, A., (2016) *HyperNormalisation* BBC

Dempsey, P., Wogalter, M., and Hancock, P., (2000) What's in a name? Using terms from definitions to examine the fundamental foundation of human factors and ergonomic science. *Theoretical Issues in Ergonomic Science*, Volume 1, Issue 1, pp. 3-10

de Wit, M., van der Kamp, J., Withagen, R., (2015) Visual illusions and direct perception: Elaborating on Gibson's insights in *New Ideas in Psychology*, Volume 36, pp. 1-9

Dix, A., and Cowen, L., (2007). HCI 2.0? Usability meets Web 2.0**.** In *Proceedings of HCI2007, Volume 2*

Dubin, R., 2017. The world of work: Industrial society and human relations. Taylor & Francis.

Dumas, J., (2007) The great leap forward: The birth of the Usability profession (1988-1993) in *The Journal of Usability Studies* Volume 2, Issue 2 pp. 54-60

Ellsbury, G., (2012) *The Enigma and the Bombe* downloaded 10th May, 2012 from http://www.ellsbury.com/enigmabombe.htm

Few, S., (2006) *Information Dashboard Design* Sebastopol CA: O'Reilly Media Inc

Field, A., 2013. Discovering statistics using IBM SPSS statistics. sage.

Findlater L., and McGrenere, J., (2004) A Comparison of Static, Adaptive and Adaptable Menus in *Proceeding of the SIGCHI conference on Human Factors in Computing Systems* pp. 89-96

Forsberg, M., (2003) *Why is speech recognition difficult?*, Chalmers University of technology, Göteberg

Frayn, M., (1998) *Copenhagen* London: Methuen

Frey, C.B., and Osborne, M.A., 2017. The future of employment: how susceptible are jobs to computerisation?. Technological Forecasting and Social Change, 114, pp. 254-280.

Gill, P., Stewart, K., Treasure, E., and Chadwick, B., (2008). Methods of data collection in qualitative research: interviews and focus groups. *British dental journal*, Volume *204, issue* 6, pp. 291-295.

Gillham, B., (2008) *Developing a questionnaire*. A&C Black

Glenn, N., and Weaver, C., (1982) Enjoyment of Work by Full-Time Workers in the US, 1955 and 1980 *Public Opinion Quarterly*, Volume 46, pp. 459-470

Gordon, I., (2004) *Theories of visual perception 3rd ed.* Hove: Psychology Press

Grant, D.M., Malloy, A.D., and Murphy, M. C., (2009). A comparison of student perceptions of their computer skills to their actual abilities. Journal of Information Technology Education, 8(1), 141-160.

Grönvall, E., Ciolfi, L., Avram, G., Rossitto, C., and Barkhuus, L., (2016). HCI at the boundary of work and life. *Personal and Ubiquitous Computing*, pp. 1-5.

Haworth, J., Veal, A., (2004) *Work and Leisure* London: Routledge

Hill, A., (1997) *Just Business: Christian Ethics for the Marketplace* Carlisle: Paternoster Press

Haidt, J., (2006) *The Happiness Hypothesis* London: William Heinemann

Hammer, Ø., Harper, D., Ryan, P., (2001) PAST: Paleontological statistics software package for education and data analysis. Palaeontologia Electronica 4(1): 9pp. http://palaeo-electronica.org/2001_1/past/issue1_01.htm

Hirschberg, J., and Manning, C., (2016) *Advances in Natural Language Processing* Downloaded from http://cs224d.stanford.edu/papers/advances.pdf Dec 2016

Hoffman, D., Singh, M., and Prakash, C., (2015) The Interface Theory of Perception in *Psychonomic bulletin & review* volume 22 issue 6

Holzinger, A., Mayr, S., Slany, W., and Debevc, M., (2010) The Influence of AJAX on Web Usability *Proceedings of the 2010 International Conference on e-Business* pp. 1-4

Hudson, W., (2007) AJAX Design and Usability *Proceedings of the 21st BCS HCI Group Conference*, Volume 2, pp. 227-8

IBM Corp. Released 2013. *IBM SPSS Statistics for Windows, Version 22.0*. Armork, NY

Jenkins, A., Sabates, R., (2007) *The Classification of Qualifications in Social Surveys* Institute of Education: London

Jerina, A., (2010) *Social Media and the Simulcrum of the self.* Retrieved June 1st, 2012, from http://researchgeek.wordpress.com/2010/10/10/simulacrum-of-the-self/

Jones, D., (2011) *Punched Cards – a Brief Illustrated Technical History* Downloaded on 11th May 2012 from http://www.divms.uiowa.edu/~jones/cards/history.html

Kaufman, D., Sheehan, B., Stetson, P., Bhatt A., Field, A., Patel C., and Maisel J., (2016) Natural Language Processing – Enabled and Conventional Data Capture Methods for Input to Electronic Health Records: A Comparative Usability Study in *JMIR Medical Informatics* Oct-Dec 2016

Kaye, L.K., Monk, R.L., Wall, H.J., Hamlin, I., and Qureshi, A.W., 2018. The effect of flow and context on in-vivo positive mood in digital gaming. International Journal of Human-Computer Studies, 110, pp. 45-52.

Kent, R., (n.d.) *Baudrillard and Hyperreality* Retrieved November 5, 2007, from http://www.ilovephilosophy.com

Kersten, M., and Gail C., "Reducing friction for knowledge workers with task context" *AI Magazine*, Summer 2015, p. 33+. *Academic OneFile*, Accessed 13 Dec. 2016.

Kim J., Lund A., and Dombrowski C., (2013) Telling the story in Big Data in *Interactions* vol 20.3, p. 48-51

Klein, L., and Pereira, B., (2016) The Survival of Interorganizational Networks: A proposal based on Resource Dependence Theory. *RAM. Revista de Administração Mackenzie*, Volume *17* issue 4, pp. 153-175.

Kludge J., Kargl F., and Weber M., (2007) The effects of the AJAX Technology on Web Application *Usability International Conference on Web Information Systems and Technologies (WEBIST 2007)*

Koehn, S., 2017. Flow State and Performance Patterns in Tennis Competition. Psychology and Behavioral Science International Journal, 7(3).

Krippendirff, K., (2004) *Content Analysis: An Introduction to its Methodology,* London: Sage

Krishnamurthi M., and Young R., (1984) A Multitasking Implementation of System Simulation: The Emulation of an asynchronous parallel processor for System Simulation using a single processor in *Proceedings of the 16th conference on Winter simulation* pp. 652-662.

Krug, S., (2000) *Don't make me think* Indianapolis: New Riders Publishing

Kuhl B., Dudokovic N., Kahn I., and Wagner A., (2007) Decreased demands on Cognitive control reveal the Neural Processing benefits of Forgetting *Nature Neuroscience* Volume 10 issue 7 pp. 908-914

Levine, C., Salmon, L., and Weinberg, D. H., (1999). Revising the standard occupational classification system. Monthly Labor Review, 122(5), 36-45.

Ludwig, T., Dax, J., Pipek, V., and Randall, D., (2016). Work or leisure? Designing a user-centered approach for researching activity “in the wild”. *Personal and Ubiquitous Computing*, *20*(4), 487-515.

Makridakis, S., 2017. The forthcoming artificial intelligence (AI) revolution: Its impact on society and firms. Futures, 90, pp. 46-60.

Malizia A., and Bellucci A., (2012) The artificiality of Natural User Interfaces. *Communications of the ACM Vol. 55, No 3*. New York, ACM Press, pp. 36-38

Mao, Y., Roberts, S., Pagliaro, S., Csikszentmihalyi, M., and Bonaiuto, M., (2016). Optimal experience and optimal identity: A multinational study of the associations between flow and social identity. *Frontiers in psychology*, *7*, 67.

Marx, K., (1946) *Capital*, London: Everyman

McCarther, S., (2018). Getting in the Zone, Part 1: Flow and Finding a State of Peak Performance. Journal of Singing, 74(3), pp. 329-334.

Miller, G., (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review 3*, 81-97

Missaire, M., Fraize, N., Joseph, M. A., Hamieh, A. M., Parmentier, R., Marighetto, A., and Malleret, G., (2017). Long-term effects of interference on short-term memory performance in the rat. *PloS one*, *12*(3), e0173834.

Mitchell J., and Shneiderman, B., Dynamic versus Static Menus: An Exploratory Comparison in *SIGHCI Bulletin, Volume 20, Issue 4* pp. 33-37

Moggridge, B., (2006) Designing Interactions, Cambridge, MA: MIT Press

Myers, A. C., (Ed) (1987) *The Eerdmans Bible Dictionary* Grand Rapids, Mich.: William B Eerdmans Publishing

Myers, M. D., and Newman, M., (2007). The qualitative interview in IS research: Examining the craft. Information and organization, 17(1), pp. 2-26.

Nacke, L. E., and Deterding, C.S., (2017). The maturing of gamification research. Computers in Human Behaviour, pp. 450-454.

Nardi, B., (1996) *Activity Theory and HCI*. Cambridge, Mass: MIT Press

Nardi, B.A., (1996). Studying context: A comparison of activity theory, situated action models, and distributed cognition. Context and consciousness: Activity theory and human-computer interaction, 69102.

National Science Foundation (2001) *The application and Implications of Information Technologies in the home: Where are the Data and what do they say?* Arlington. Downloaded from http://www.nsf.gov/statistics/nsf01313/patterns.htm Jan 2014

Nielsen, J., (2001) *Error Message Guidelines*. Retrieved May 2nd, 2012 from http://www.useit.com/alertbox/20010624.html

Nielsen, J., (2006) *F-Shaped Pattern For Reading Web Content*. Retrieved May 1st, 2012 from http://www.useit.com/alertbox/reading_pattern.html

Nielsen, J., and Loranger, H., (2006) *Prioritising Web Usability* Berkley: New Riders Publishing

Norman, D., (1988) *The Psychology of Everyday Things,* New York, Basic Books

Norman, D., (1993) *Things that make us smart*, Cambridge Mass.: Perseus Books

Norman, D., (2004) *Emotional Design*, New York, Basic Books

Norman, D., (2009) Compliance and Tolerance in *Interactions*, volume 16.3, pp. 61-65

OEDC (2009), Special Focus: Measuring Leisure in OECD Countries in *Society at a glance* Retrieved June 5th, 2012, from http://www.oecd.org/dataoecd/36/43/42675407.pdf

Oh, J., Susarla, A., and Tan Y., (2008) *Examining the Diffusion of User-Generated Content on Online Social Networks* Retrieved June 1st, 2012 from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1182631

Olsen, K., (2012) Invisible Stable Interfaces, in *Interfaces* issue 92, Autumn 2012.

ONS (2016) Internet access – households and individuals: 2016. retrieved Dec 2016 from https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2016

O'Reilly, T., (2007). What is Web 2.0: Design patterns and business models for the next generation of 20 software. Retrieved May 9th, 2012, from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1008839

Ostroff, C., (1992) The relationship between Satisfaction, Attitudes and Performance: An Organizational Level Analysis: Journal of Applied Psychology, volume 77, issue 6, 963-974

Pachal, P., *Bill Gates Testifies: Microsoft beat WordPerfect fair and square* downloaded from http://mashable.com/2011/11/21/bill-gates-microsoft-word-wordperfect/ on 15th May 2012

Peter C., Blyth, G., (2005) The Role of Emotions in Human-Computer Interaction. *Proceedings of the HCI 2005 Conference*, Volume 2, pp. 295--298.

Peter C., Crane E., Axelrod L., Beale R., (2006) Engaging with Emotions – the Role of Emotion in HCI in *Proceedings of the HCI 2006 Conference,* Volume 2, pp. 270-272, London; British Computer Society

Pfeffer, J., and Salancik, G., (2003) *The External Control of Organisations: A Resource Dependence Perspective* Stanford University Press.

Picard, R., (1997) *Affective Computing* Cambridge, Mass; London: MIT Press

Picard, R., (1999) Affective Computing for HCI, in *Proceedings of HCI International on Human-Computer Interaction: Ergonomics and User interfaces* Volume 1, pp. 829-833

Pona, N., (2015) *Computational Theory of Consciousness* Downloaded from https://logic.at/lvas/185054/Consciousness.pdf Dec 2016

Portugal, D., Belk, M., Quintas, J., Christodoulou, E., and Samaras, G., (2016, July). Identification of an Individual's Frustration in the Work Environment Through a Multi-sensor Computer Mouse. In *International Conference on Human Aspects of IT for the Aged Population* (pp. 79-88). Springer International Publishing.

Potosky, D., and Bobko, P., (1998). The Computer Understanding and Experience Scale: a self-report measure of computer experience1. Computers in Human Behavior, 14(2), pp. 337-348.

Reynolds, C. J., (2001) *The sensing and measurement of Frustration with Computers* SM thesis, MIT. Retrieved online form http://vismod.media.mit.edu/pub/tech-reports/TR-559.pdf May 1st, 2012

Rogers, E. M., (2003) *Diffusion of innovations 5th ed.*, New York; London: Free Press

Scala: http://www.scala-lang.org/old/node/8610 - downloaded April 2018

Seeley M., and Targett, D., (1999) Patterns of Senior Executives; personal use of computers in *Information and Management* Volume 35, issue 6, pp. 315-330

Shah, H., Warwick, K., Vallverdué, Wu, D., (2016) Can machines talk? Comparison of Eliza with modern dialogue systems in *Computers in Human Behavior* volume 58, pp. 278-295

Shove, E., Watson, M., Hand, M., Ingram, J., (2007) *The Design of Everyday Life*, Oxford:Berg

Shneiderman, B., (1982). System message design: Guidelines and experimental results. In A. Badre and B. Shneiderman (eds). *Directions in Human/Computer Interaction*. Norwood, NJ: Ablex Publishing, pp. 55-78

Shneiderman, B., (2002) *Leonardos laptop: human needs and the new computing technologies*, London: MIT Press.

Shreshtha, S., and Lenz K., (2007) Eye Gaze Patterns while Searching vs Browsing a website, in *Usability News*, Vol 9, issue 1

Schröedinger, E., (1992) *What is Life?,* Cambridge: Cambridge University Press

Smith, L.J., Gradisar, M., King, D.L. and Short, M., 2017. Intrinsic and extrinsic predictors of video-gaming behaviour and adolescent bedtimes: the relationship between flow states, self-perceived risk-taking, device accessibility, parental regulation of media and bedtime. Sleep medicine, 30, pp. 64-70.

Smith, P. A., (1996) *Towards a practical measure of hypertext usability* in Interacting with Computers, 8(4), 365-381

SOCPC (2010) *2010 SOC User Guide* US Bureau of Labor Statistics.

Spool, J. M., (2009) *The $300 Million Button* Retrieved Feb 22nd, 2012 from http://www.uie.com/articles/three_hund_million_button

Stebbins, R., (2005) Choice and experimental definitions of Leisure. *Leisure Sciences* Volume 27, pp. 349-352

Stolz, T., (22010016) Contradictions of Hyperreality: Baudrillart, Žižek and Virtual Dialectics. *International Journal of Žižek studies*, Volume 10, Issue 1

Stormoen, S., Urke, H.B., Tjomsland, H.E., Wold, B., and Diseth, Å., (2016). High school physical education: What contributes to the experience of flow?. *European Physical Education Review*, *22*(3), pp. 355-371.

Suchman, L., (1991) *Plans and Situated Actions*, Cambridge: Cambridge University Press

SurveyMonkey (2009) Retrieved Nov 29th 2009 from https://www.surveymonkey.com/home/

SurveyMonkey (2011) Retrieved Oct 24th 2011 from https://www.surveymonkey.com/home/

SurveyMonkey (2012) Retrieved Feb 4th 2012 from https://www.surveymonkey.com/home/

Ternus, M., and Shuster, G., (2008). Computer literacy in a traditional nursing program: A 7-year study to identify computer-based skills needed for success. *MERLOT Journal of Online Learning and Teaching*, 4(1), 24-36.

Tufte, E., (1990) *Envisioning Information*, Cheshire Conn: Graphics Press

Tufte, E., (2001) *The Visual Display of Quantitative Information*, Cheshire Conn: Graphics Press

Tufte, E., (2006) *Beautiful Evidence*, Cheshire, Conn: Graphics Press

Tullis, T., and Albert, B., (2008) *Measuring the User Experience*, Burlington MA: Morgan Kaufmann van Dam, A., (1997), *Post-WIMP User Interfaces*, CACM, Volume 40, Issue 2, pp. 63-67

Verbeek, P-P., (2015) Beyond Interaction: A Short introduction to Mediation Theory in *Interactions* Volume 21, issue 3

Wahid, S., and Khoon, Z., (2015) Rich Internet Application (RIA) New Dimension with HTML5, CSS3 and Javascript technology in *International Journal of advances in Computer Science and its Applications (IJCSIA)*, pp. 172-177, Volume 5 Issue 1.

Weinschenk, S., (2009) *Neuro Web Design* Berkley, CA; New Riders

Weiser, M., (1994) The world is not a desktop. *Interactions*; January 1994; pp. 7-8.

Weizenbaum, J., (1966) *ELIZA – a computer program for the study of natural language communication between man and machine* in CACM, volume 9, issue 1

White, M. D., and Livonen, M., (2002). Assessing level of difficulty in web search questions. *The Library Quarterly*, *72*(2), 205-233.

Wikipedia (2002) Retrieved March 2014 from http://en.wikipedia.org/wiki/Google_(verb)

Wilson, E. E., (2016). *Metacognitions of flow experience: towards an understanding of the self-regulation of optimal experience* (Doctoral dissertation, London Metropolitan University).

West, L. J., (1998) *The standard and Dvorak keyboards revisited: Direct measures of Speed* Retrieved May 29th 2013 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.6886&rep=rep1&type=pdf

Yee, R., Guo, Y., and Yeung, A., (2015) Being close or being happy? The relative impact of work relationship and job satisfaction on service quality in *International Journal of Production Economics*, Volume 169, pp. 391-400

Young, K., (2008) Towards a Model for the Study of Children’s Informal Internet use in *Computers in Human Behavior*, Volume 24 issue 2 pp. 173-184

Zimmerman, J., Forlizzi, J., and Evenson, S., (2007), April. Research through design as a method for interaction design research in HCI. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 493-502). ACM.

Zuboff, S., (1988) *In the age of the smart machine – the future of work and power*, Oxford: Heinemann Professional Publishing

Appendix 1 - Work and Leisure Questionnaire data.

Table A1-50 – Calculations of estimated time spent against application counts to produce mean application time for work environments. p<0.05, comparing the two data sets.

Proportion of time spent	Number of applications	Mean time per application
40	7	5.7
15	3	5.0
55	2	27.5
55	7	7.9
83	10	8.3
83	4	20.8
40	4	10.0
40	7	5.7
95	7	13.6
83	10	8.3
15	3	5.0
15	1	15.0
83	10	8.3
15	7	2.1
67	10	6.7
15	7	2.1
83	7	11.9
15	7	2.1
15	7	2.1
15	1	15.0
83	10	8.3
67	10	6.7
15	7	2.1
15	1	15.0
95	7	13.6
83	10	8.3
15	7	2.1
15	7	2.1
95	2	47.5
67	7	9.6
67	7	9.6
55	10	5.5
40	7	5.7
95	7	13.6
95	10	9.5
83	10	8.3
95	3	31.7
67	7	9.6
83	7	11.9
55	7	7.9
83	10	8.3
95	7	13.6
55	10	5.5
40	7	5.7
15	3	5.0
40	7	5.7
95	10	9.5
83	10	8.3
83	7	11.9
55	10	5.5
83	10	8.3
40	7	5.7

Table A1-51 - Calculations of estimated time spent against application counts to produce mean application time for leisure environments.

Proportion of time spent	Number of applications	Mean time per application
1.5	2	0.75
1.5	5	0.30
1.5	6	0.25
1.5	3	0.50
1.5	4	0.38
1.5	5	0.30
1.5	5	0.30
6	3	2.00
1.5	1	1.50
6	6	1.00
1.5	2	0.75
1.5	6	0.25
1.5	1	1.50
1.5	6	0.25
6	6	1.00
1.5	6	0.25
6	6	1.00
1.5	5	0.30
11.5	6	1.92
1.5	6	0.25
1.5	6	0.25
1.5	4	0.38
1.5	6	0.25
6	3	2.00
6	6	1.00
1.5	4	0.38
6	6	1.00
1.5	6	0.25
6	6	1.00
1.5	1	1.50
1.5	4	0.38
6	6	1.00
6	4	1.50
1.5	3	0.50
1.5	3	0.50
1.5	4	0.38
6	6	1.00
1.5	6	0.25
1.5	6	0.25
1.5	4	0.38
6	6	1.00

Table A1-52 – Proportion of work specific applications used compared to total applications used in a work situation.

Total applications used	Work specific	Percentage of work specific
7	3	42.9
3	-	-
2	4	200.0
7	5	71.4
10	5	50.0
4	4	100.0
4	-	-
7	3	42.9
7	3	42.9
10	5	50.0
3	-	-
1	-	-
10	4	40.0
7	3	42.9
10	5	50.0
7	3	42.9
7	4	57.1
7	3	42.9
7	5	71.4
1	1	100.0
10	1	10.0
10	5	50.0
7	1	14.3
1	-	-
7	4	57.1
10	3	30.0
7	-	-
7	-	-
2	2	100.0
7	-	-
7	5	71.4
10	5	50.0
7	2	28.6
7	3	42.9
10	5	50.0
10	2	20.0
3	3	100.0
7	5	71.4
7	4	57.1
7	4	57.1
10	4	40.0
7	5	71.4
10	5	50.0
7	4	57.1
3	-	-
7	3	42.9
10	5	50.0
10	4	40.0
7	5	71.4
10	5	50.0
10	4	40.0

Table A1-53 – Proportion of time spent on main computer application against total time spent on computer applications in work environment.

Total hours per day	Hours per day on main application	Proportion of time on main application
2.0	2.96	0.7
2.0	1.20	1.7
-	4.40	-
4.5	4.40	1.0
4.5	6.56	0.7
4.5	6.56	0.7
2.0	2.96	0.7
2.0	2.96	0.7
2.0	7.60	0.3
4.5	6.56	0.7
2.0	1.20	1.7
2.0	1.20	1.7
4.5	6.56	0.7
2.0	1.20	1.7
4.5	5.36	0.8
2.0	1.20	1.7
4.5	6.56	0.7
2.0	1.20	1.7
7.5	1.20	6.3
-	1.20	-
2.0	6.56	0.3
7.5	5.36	1.4
2.0	1.20	1.7
-	1.20	-
4.5	7.60	0.6
2.0	6.56	0.3
2.0	1.20	1.7
-	1.20	-
7.5	7.60	1.0
-	5.36	-
7.5	5.36	1.4
4.5	4.40	1.0
2.0	2.96	0.7
2.0	7.60	0.3
7.5	7.60	1.0
2.0	6.56	0.3
-	7.60	-
2.0	5.36	0.4
4.5	6.56	0.7
2.0	4.40	0.5
4.5	6.56	0.7
7.5	7.60	1.0
2.0	4.40	0.5
4.5	2.96	1.5
-	1.20	-
2.0	2.96	0.7
2.0	7.60	0.3
2.0	6.56	0.3
7.5	6.56	1.1
2.0	4.40	0.5
7.5	6.56	1.1

Table A1-54 –Proportion of time spent on main application against total time spent on computer applications for leisure environment.

Total hours per day	Hours per week on main application	Proportion of time on main application
3	10	0.3
-	-	-
-	-	-
8	10	0.8
3	10	0.3
3	10	0.3
-	-	-
3	10	0.3
15	10	1.4
27	10	2.6
27	42	0.6
3	10	0.3
15	42	0.4
8	10	0.8
8	10	0.8
-	-	-
8	10	0.8
3	10	0.3
15	42	0.4
-	-	-
15	10	1.4
27	42	0.6
3	10	0.3
-	-	-
42	77	0.5
15	10	1.4
3	10	0.3
-	-	-
8	10	0.8
-	-	-
3	10	0.3
3	42	0.1
8	42	0.2
3	10	0.3
8	42	0.2
3	10	0.3
-	-	-
15	42	0.4
3	10	0.3
8	10	0.8
27	42	0.6
27	42	0.6
3	10	0.3
8	10	0.8
-	-	-
3	10	0.3
8	42	0.2
3	10	0.3
15	10	1.4
3	10	0.3
27	42	0.6
3	10	0.3
27	42	0.6

Table A1-55 – Comparison of total number of applications in work environment against leisure environments. p>0.05 across the two data sets.

Work Applications	Leisure Applications	Proportion
7	2	3.5
3	-	-
2	-	-
7	5	1.4
10	6	1.7
4	3	1.3
4	-	-
7	4	1.8
7	5	1.4
10	5	2.0
3	3	1.0
1	1	1.0
10	6	1.7
7	2	3.5
10	6	1.7
7	-	-
7	1	7.0
7	6	1.2
7	6	1.2
1	-	-
10	6	1.7
10	6	1.7
7	5	1.4
1	-	-
7	6	1.2
10	6	1.7
7	6	1.2
7	-	-
2	4	0.5
7	-	-
7	6	1.2
10	3	3.3
7	6	1.2
7	4	1.8
10	6	1.7
10	6	1.7
3	-	-
7	6	1.2
7	1	7.0
7	4	1.8
10	6	1.7
7	4	1.8
10	3	3.3
7	3	2.3
3	-	-
7	4	1.8
10	6	1.7
10	6	1.7
7	6	1.2
10	4	2.5
10	6	1.7

Work And Leisure

1. Work and Leisure survey

Thank you for taking part in this survey. It is asking about your usage of computer systems at work and at leisure.

Work includes anything that you are contracted to do - it is not necessarily paid work, but whatever you would consider your job or career.

Leisure includes hobbies or interests, as well as computer usage for personal use - personal emails, writing letters using a word processor, or managing personal accounts.

All questions are required, please answer to the best of your ability, selecting the answer that most closely reflects your actual situation.

Thank you!

Work And Leisure

2. Work Computer usage

Firstly, I would like to ask about your work computer usage. This is any time you use a computer in your contracted (not necessarily paid) work.

1. Approximately what proportion of your working time do you spend using computing applications? Please give a percentage.

- 0-30%
- 31-50%
- 51-60%
- 61-75%
- 76-90%
- 91-100%

2. In a normal or average working week, how many different computer applications - including web sites - do you use.

- 1
- 2
- 3
- 4
- 5-10
- 10+

Work And Leisure

3. How many of these applications are work only - that is, applications that are only used within your work environment, or are specific to your work type.

This might include accounting programs for your company. It does not include office-type applications like Word. However, if your have an application that uses, say, Excel, but has been written for your organisation, this would count.

Note that computer applications can include web applications that are work-specific.

- 0
- 1
- 2
- 3
- 4
- 5+

Work And Leisure

3. Work computer usage

4. For the work only application that you use most of the time (if there is a choice, then please just choose one), how much of your day do you use it for?

- 1-3 hours
- 4-5 hours
- 6-9 hours
- more than 9 hours

5. In your opinion, does the application make your performance of your work easier or harder?

- Easier
- Harder

6. Please explain why this is the case, if possible.

Work And Leisure

4. Leisure computer usage

Secondly, I would like to ask about your leisure computer usage. This includes any usage that you use to manage your life, or for hobbies or interests.

7. As an average, how long do your spend per day using computer applications. Note that this includes browsing web sites, or using online mail (like google mail or yahoo).

- 0-3 hours
- 4-8 hours
- 9-14 hours
- more than 14 hours

8. How many different applications do you use in a week?

- 1
- 2
- 3
- 4
- 5
- 6+

9. For your most common application, how long in a week do you spend using it?

- 1-5 hours
- 6-10 hours
- 11-20 hours
- 21-35 hours
- 35-50 hours
- more than 50 hours

10. What drives you to these particular applications?

- Personal choice
- Requirements from other people (i.e. members of a club)
- Software that came with the computer
- Application that someone recommended
- Saw it advertised, or downloaded it when I needed something

Other (please specify)

Appendix 2 - Computer Usage Questionnaire data.

Table A2-56 – Responses to the statements by participants for software applications that they had positive experiences of. Respondents – 80 for part 1, 69 for part 2

	Strongly agree	Agree	Neither	Disagree	Strongly disagree	N/A
It was straightforward to start using	22	44	4	7	2	1
It did what I wanted it to	25	48	2	3	1	1
It was easy to work out what I had to do to make it work	19	38	11	8	3	1
It informed me of what I needed to do clearly and in my language	15	31	18	13	2	1
It helped me focus on the information, not the application	19	28	18	11	1	3
It did what it said it would	23	47	6	1	1	2
It provided me with exactly what I needed from it	21	33	17	4	3	2
It had features that I might need, but didn't use	27	39	2	0	0	1
It encouraged me to think of possible uses I could put it to	14	18	21	14	0	2
It was highly compatible with other software applications I use, or worked alongside or with other software applications	19	26	9	11	1	3
While there were many complex options, I could use it simply too	21	30	10	6	0	2
The complexity of the software application was about what I expected and related to what I had to do	13	40	9	6	0	1
The software application understood what I was trying to do	12	28	17	8	3	1
The software application handled minor errors sensibly	10	32	19	6	1	1

Table A2-57 - Responses to the statements by participants for software applications that they had negative experiences of. Respondents – 73 for part 1, 64 for part 2

	Strongly Agree	Agree	Neither	Disagree	Strongly disagree	N/A
It ran on only specific computers	9	18	14	17	3	12
I required help or assistance to use it	14	23	7	19	6	4
It was always getting in my way, or displaying messages	13	23	12	14	4	7
It required significant effort to get it to do what I wanted	25	27	7	5	4	5
It provided me with information in an unhelpful format, or requiring further work from me	16	27	10	11	4	5
One or more features I expected were not present	19	24	12	9	5	4
It seemed more restricted than I had expected	8	33	9	10	1	3
It required effort to work with other software applications	16	26	10	5	1	6
The software application was far too complex for what I was trying to do	6	11	15	25	4	3
The software application did far more than I needed and so was less useful for my specific task.	6	14	11	25	5	3
The format of information required was very rigid	14	19	15	11	2	3
Making minor mistakes or forgetting things caused big problems.	12	16	19	11	2	4

This survey is trying to assess how people work with, interact with, and use computers and computer systems. This is part of an academic research project, and the results will be incorporated into a research project.

Please note that no personal details are being collected, and that only the summary results will be published. Thank you for agreeing to participate in this survey.

First of all, a few questions about who you are

1. What is the highest academic qualification that you have achieved.

- None
- O levels or GCSEs
- A Levels
- Degree or masters
- Doctorate

2. How would you rate your computer experience? Please indicate the nearest if none of them seem to match

- None
- Occasional usage
- Regular usage
- Competent user
- Expert application user
- Expert software developer

3. How would you define your job area (if none match, please give the nearest one)

- Student
- Teaching/Academic
- Unskilled or manual
- Skilled or office
- Managerial
- Artistic/Creative
- Caring
- At home/voluntarily unemployed
- Involuntarily unemployed (long term)
- Retail

Now we want to ask about your experience with software applications. This means anything that you run on a computer, including Windows or MacOS, tools like Word and Excel, web sites that involve processing, like places you buy from. It even includes the web browser that you are currently using. This list is not exclusive - there are many others, but this is to give you an idea of the types of things we are looking for.

We also want to ask about "helpful" and "unhelpful" software. "Helpful" means that your experience of using the software was positive overall or generally, and that you would use it again willingly. "Unhelpful" means that you found problems and would consider using some other software if it was suggested - you were not happy with the experience. Please note that this is about your experience of using the software, and so any answers are valid. We are asking if you found it positive or negative to use

Please note that the sets of questions alternate between helpful applications and positive experiences and unhelpful applications and negative responses. Please check the wording of the question carefully.

4. Think about a software application that you have used in the last year that you have found helpful and positive. Please rate the following statements about this software application:

	Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree	Not relevant
It was straightforward to start using	○	○	○	○	○	○
It did what I wanted it to	○	○	○	○	○	○
It was easy to work out what I had to do to make it work	○	○	○	○	○	○
It informed me of what I needed to do clearly and in my language	○	○	○	○	○	○
It helped me focus on the information, not the application	○	○	○	○	○	○
It did what it said it would	○	○	○	○	○	○
It provided me with exactly what I needed from it	○	○	○	○	○	○

5. Think about a software application that you have used in the last year that you have found unhelpful and negative. Please rate the following statements about this software application:

	Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree	Not relevant
It ran on only specific computers	○	○	○	○	○	○
I required help or assistance to use it	○	○	○	○	○	○
It was always getting in my way, or displaying messages	○	○	○	○	○	○
It required significant effort to get it to do what I wanted	○	○	○	○	○	○
It provided me with information in an unhelpful format, or requiring further work from me	○	○	○	○	○	○
One or more features I expected were not present	○	○	○	○	○	○

6. Thinking about a software application that you found helpful again - this can be the same one as previously, or a different one. Please rate the following statements about this software application:

	Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree	Not relevant
It had features that I might need, but didn't use	○	○	○	○	○	○
It encouraged me to think of possible uses I could put it to	○	○	○	○	○	○
It was highly compatible with other software applications I use, or worked alongside or with other software applications	○	○	○	○	○	○
While there were many complex options, I could use it simply too	○	○	○	○	○	○
The complexity of the software application was about what I expected and related to what I had to do	○	○	○	○	○	○
The software application understood what I was trying to do	○	○	○	○	○	○
The software application handled minor errors sensibly	○	○	○	○	○	○

7. Thinking again about a software application that you have used in the last year that you have found unhelpful and negative. Please rate the following statements about this software application:

	Strongly Agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree	Not relevant
It seemed more restricted than I had expected	○	○	○	○	○	○
It required effort to work with other software applications	○	○	○	○	○	○
The software application was far too complex for what I was trying to do	○	○	○	○	○	○
The software application did far more than I needed and so was less useful for my specific task.	○	○	○	○	○	○
The format of information required was very rigid	○	○	○	○	○	○
Making minor mistakes or forgetting things caused big problems.						

Thank you for those answers. Just a few final questions to make sure that we haven't missed anything important from you.

8. What one factor would make the biggest difference to your usage of software applications?

9. If you are prepared to, please provide your email address. This will only be used for a simple acknowledgement, and will not be used for any further "spam", or be given or sold to anyone else. It will not appear in the research results at all.

Appendix 3 - Flow questionnaire data

Table A3-58 - Computer usage questionnaire, computer usage questions summary.

Participant number	A	B	C	D	E	F	G	Mean	Std Deviation
1	3	3	3	4	3	3	3	3.1	0.38
2	4	4	1	4	3	4	4	3.4	1.13
3	2	2	3	3	1		2	2.2	0.75
4	4	4	4	4	3	3	4	3.7	0.49
5	2	3	2	2	3		3	2.5	0.55
6	3	4	2	4	4	3	4	3.4	0.79
7	3	4	3	4	4	3	4	3.6	0.53
8	4	4	3	4	3	4	4	3.7	0.49
9	4	4	4	4	4	4	3	3.9	0.38
10	2	1	3	3	2	4	2	2.4	0.98
11	3	4	2	4	4	3	3	3.3	0.76
12	4	3	3	3	3	3	4	3.3	0.49
13	4	4	2	4	3	3	4	3.4	0.79
14	3	2	3	3	4		3	3.0	0.63
15	3	3	3	4	3	3	3	3.1	0.38
16	4		4	4	3	3	4	3.7	0.52
17	4	4	4	4	4	4	4	4.0	0.00
18	4	3	3	4	3	3	3	3.3	0.49
19	3	4	2	4	3	4	4	3.4	0.79
20	4	4	3	4	3	4	4	3.7	0.49
21	4	3	3	4	3	4	4	3.6	0.53
22	4	1	4	3	3	3	3	3.0	1.00
23	3	3	3	3	3	4	2	3.0	0.58
24	3	4	3	3	3	4	4	3.4	0.53
25	3	3	3	3	2	3	3	2.9	0.38

Table A3-59 - How important was the computer to your task, flow questionnaire

Participant	a	b	c	d	Mean	Std Dev
1	3	3	2	3	3.0	0.00
2	4	2	2	4	3.3	0.96
3	2	3	3	2	2.3	0.50
4	3	2	2	3	2.8	0.50
5	3	2	2	2	2.5	0.58
6	3	2	2	3	2.8	0.50
7	4	1	2	3	2.8	1.26
8	4	1	1	4	3.3	1.50
9	4	1	1	3	3.0	1.41
10	4	2	3	3	2.8	0.96
11	2	3	2	2	2.5	0.58
12	3	2	2	3	2.8	0.50
13	2	3	2	3	2.8	0.50
14	-	3	3	2	2.3	0.58
15	4	1	1	3	3.0	1.41
16	4	2	2	4	3.3	0.96
17	3	3	2	3	3.0	0.00
18	3	2	2	3	2.8	0.50
19	4	1	1	4	3.3	1.50
20	4	1	1	4	3.3	1.50
21	3	2	2	3	2.8	0.50
22	4	3	2	3	3.3	0.50
23	3	2	2	3	2.8	0.50
24	4	2	2	4	3.3	0.96
25	3	2	2	3	2.8	0.50
Mean	3.3	2.0	3.1	3.1		

Table A3-60 - Computer usage questionnaire, non-computer questions summary. 1 is a disagreement, 5 is an agreement

Participant number	A	B	C	D	E	F	G	Mean	Std Deviation
1	2	4	2	4	3		4	3.2	0.98
2	3	4	2	3	3	4	3	3.1	0.69
3	4	4	4	4	3	3	4	3.7	0.49
4	3	4	3	4	3	3	4	3.4	0.53
5	2	3	3	2	3	3	3	2.7	0.49
6	3	4	3	4	4	3	4	3.6	0.53
7	2	4	3	4	4	3	4	3.4	0.79
8	3	4	3	4	2	3	4	3.3	0.76
9	4	4	4	4	4	4	4	4.0	0.00
10		3	3	4	1		4	3.0	1.22
11	3	4	2	4	4		4	3.5	0.84
12	3	4	3	3	3	3	3	3.1	0.38
13	3	4	2	3	3	3	4	3.1	0.69
14	3	4	3	4	3	3	3	3.3	0.49
15	3	3	3	4	3	3	3	3.1	0.38
16	1	3	3	3	3		4	2.8	0.98
17	4	4	4	4	4	4	4	4.0	0.00
18	3	4	3	4	3	3	4	3.4	0.53
19	3	3	3	4	3	4	3	3.3	0.49
20	3	3	3	2	3	3	3	2.9	0.38
21	3	3	3	3	2	3	4	3.0	0.58
22	4	2	3	3	3	3	3	3.0	0.58
23	3	3	3	3	3	4	3	3.1	0.38
24	3	4	3	4	3	4	4	3.6	0.53
25	3	3	3	3	3	3	3	3.0	0.00

Table A4-61 –Flow questionnaire responses about micro-flow, when it happens, and how often.

Participant number	Working situation	Leisure situation	Computer applications	Computers help me	Frequency
1	1	3	1	2	5
2	3	1	3	3	5
3	2	3	0	0	5
4	1	1	1	1	3
5	2	1	1	1	3
6	2	2	2	2	4
7	3	0	2	3	7
8	2	3	2	2	5
9	3	3	4	3	7
10	3	3	3	3	6
11	3	2	1	1	5
12	2	1	3	3	4
13	2	2	2	2	4
14	2	2	1	1	4
15	3	3	1	1	4
16	2	2	2	2	5
17	1	3	2	3	6
18	-	-	-	-	-
19	2	2	3	2	4
20	3	2	3	3	5
21	1	3	2	4	4
22	1	1	1	1	4
23	3	3	2	3	5
24	2	2	2	2	5
25	2	1	2	3	7
Mean	2.1	2.0	1.9	2.1	4.8

Flow Survey

Introduction

This questionnaire is looking at those times when you have done something - undertaken a task of some form, or a mini-task, a part of larger task - when it goes really smoothly. You know the times, everything just seems to work, to focus, everything just happens well.

For this questionnaire, you will need to recall some times when this has happened. It does not need to be very major tasks or activities, or particularly long times. It would be helpful if you could identify times when this happened using a computer applications and times when not using a computer at all. If you can identify recent times when this happened, it would be most helpful, as we need to understand how you felt in these cases.

The final part of the questionnaire is about a whole range of times, and your very general experience of this. The theory is that you should experience this in small ways repeatedly, in all sorts of things that you do. We want to understand whether this is the case.

Flow Survey

Computer usage

You are thinking of times when you performed an activity that went really smoothly and well. Firstly, please can you consider a time when you felt this experience engaging in a task that involved a computer. This does not need to be completely computer usage, but should have a computer as an important part of the task

1. Please rate this experience in terms of the following characteristics:

	Strongly don't agree	Don't agree	Agree	Strongly agree	Not applicable
You felt in control of the process					
There was a sense of personal reward having achieved					
I was easily able to achieve the task					
You were bored by the task, or felt disconnected from it					
You had a clear focus on the achievement of the task					
It seemed to take far longer than it actually did					
The task was really beyond my ability, although I managed it					

2. Considering the computer usage within this task specifically, please rate this aspect of the process.

	Strongly don't agree	Don't agree	Agree	Strongly agree
The computer was critical to the task				
I hardly noticed the computer usage, although it was important				
The computer application was an irritation				
The computer application was a real help				

Flow Survey

Non computer usage

You are considering times when you engaged in smoothly flowing activities, things that went really well, however difficult they were. Next, please can you consider a time when you felt this experience engaging in a task that did not involve a computer. This may be when playing a game or some other activity.

3. Please rate this experience in terms of the following characteristics:

	Strongly don't agree	Don't agree	Agree	Strongly agree	N/A
You felt in control of the process					
You were bored by the task, or felt disconnected from it					
I was easily able to achieve the task					
You had a clear focus on the achievement of the task					
It seemed to take far longer than it actually did					
The task was really beyond my ability, although I managed it					
There was a sense of personal reward having achieved					

Flow Survey

More general experiences of micro-flow

You are still thinking about tasks or mini-tasks that progress really smoothly. Please consider now when these sort of experiences tend to occur, as a rule. Note that this can just include short periods when this is happening as a part of a larger task.

4. When you are achieving tasks that you enjoy, please rate these

	Never	Occasionally	Quite Often	Usually	Always
This normally happens in a working situation	◯	◯	◯	◯	◯
Computer applications help me in this	◯	◯	◯	◯	◯
This normally happens in a leisure situation	◯	◯	◯	◯	◯
This generally involves computer applications					

*5. Please indicate how often this occurs for you

Appendix 4 - Process break experiment data

Table A4-62 – Process break experiment no breaks, periodicity and peak height data.

Reference	Mean Period	STDEV of period	Mean peak height	STDEV Peak Height
1299341879917	7.3	2.52	31537	9250
1299468393778	5.0	0.82	16017	13257
1299499825698	5.5	0.58	5688	4332
1299583025827	3.0	0.63	7509	3577
1299585863794	4.2	1.30	10601	4890
1299587299671	5.3	1.71	8347	3593
1299588105023	3.8	0.84	8241	2148
1299588401639	7.3	2.08	9081	6179
1299589733944	5.3	2.22	19628	13945
1299590423563	5.3	2.22	4820	3279
1299590530643	4.4	2.51	10598	6487
1299592484718	4.7	0.58	8696	3515
1299594223422	7.0	1.00	14521	3354
1299597416515	4.7	0.58	9029	1685
1299601928005	7.3	1.53	24590	20246
1299604752021	5.5	1.29	32216	47773
1299605223484	5.0	2.31	8948	1601
1299607127833	4.0	1.22	5814	8167
1299614741640	5.5	2.38	12822	2920
1299615882999	3.8	2.39	18537	8672
1299619580057	4.4	1.14	9067	4665
1299621972366	6.3	0.58	9299	913
1299672002759	4.5	1.29	8344	1777
1299693601976	5.5	1.29	9243	1635
1299750221699	4.8	1.71	21509	17181
1299750349421	4.2	0.84	7656	3246
1299750674865	5.7	1.15	18623	4294
1299750892362	6.7	3.79	11947	2309
1299757433430	8.5	4.95	12419	5209
1299760254953	4.0	1.87	8209	4128
1299765977192	4.4	1.52	9730	3716
1299792313890	6.3	2.08	7474	3969
1300888437785	5.5	1.00	2998	738

Table A4-63 - Process break experiment with breaks, periodicity and peak height data

Reference	Mean Period	STDEV of period	Mean peak height	STDEV Peak Height
1299341879917	4.2	1.64	38854	8495
1299468393778	4.4	1.67	10018	4702
1299499825698	4.4	1.52	4592	3289
1299585863794	5.0	2.16	10544	8287
1299587299671	4.4	1.67	8403	4628
1299588105023	5.3	3.21	68701	101948
1299588401639	2.0	-	8235	-
1299589733944	4.0	1.22	11696	5898
1299590530643	4.0	-	13957	-
1299594223422	5.0	0.82	15560	15149
1299597416515	4.5	1.00	7651	4958
1299601928005	4.8	1.50	12790	5047
1299604752021	5.0	2.31	11638	1880
1299605223484	5.3	0.50	16956	4956
1299607127833	4.4	1.52	4768	3713
1299614741640	3.5	1.05	8357	2308
1299615882999	6.0	2.65	18896	13485
1299619580057	4.0	1.22	12357	8142
1299621972366	4.8	1.26	16978	9130
1299672002759	7.3	4.93	12776	7736
1299693601976	4.3	1.50	14083	12859
1299750221699	3.4	0.55	13492	8524
1299750349421	7.3	3.21	8065	5083
1299750674865	3.6	1.95	7274	5055
1299750892362	6.3	2.52	7293	4049
1299757433430	4.8	1.26	9583	4492
1299760254953	4.2	0.84	11670	2740
1299765977192	4.8	2.36	8984	2760
1299792313890	4.0	2.00	3861	3678
1300888437785	3.6	0.89	4616	4231

Table A4-64 – Process break experiment, where time between entries increases each time.

Reference	Mean Increase period	STDEV of period	Mean increase	STDEV of increase	Question of max increase
1299468393778	4.2	1.94	10233	7758	22
1299587299671	4.2	1.83	6021	4277	4
1299588105023	3.1	0.64	8797	6961	23
1299589733944	3.1	0.83	12118	14070	3
1299594223422	4.2	1.95	6717	8224	24
1299597416515	3.0	1.20	4745	3412	22
1299614741640	4.3	2.34	7318	6129	21
1299619580057	3.6	0.98	6092	6242	23
1299621972366	3.4	1.14	16413	15126	17
1299672002759	5.0	3.39	6934	6464	18
1299750221699	4.2	1.94	8798	6501	23
1299750349421	3.1	1.36	6094	4109	25
1299750674865	6.3	3.30	15633	6286	12
1299750892362	3.1	0.83	10304	4026	19
1299757433430	4.8	1.30	4428	3745	24
1299760254953	3.6	1.40	8826	5424	23
1299765977192	3.0	1.31	4408	3622	21
1299792313890	3.7	2.87	5447	4685	22

Appendix 5 - Frustration point experiments

Table A5-65 – Frustration point experiment, first site responses from unsupervised participants.

	Task 1	Task 2	Task 3	Task 4	Task 5	Mean	Std Dev
1	17.1	11.1	3.9	14.6	56.2	**20.6**	**20.5**
2	125.5	-	-	-	-	**125.5**	**-**
3	73.7	-	-	-	-	**73.7**	**-**
4	321.1	9.9	69.6	59.0	119.9	**115.9**	**121.2**
5	206.2	-	-	-	-	**206.2**	**-**
6	35.0	116.8	46.5	6.3	24.6	**45.8**	**42.3**
7	90.1	228.9	75.2	42.6	29.8	**93.3**	**79.6**
8	9.3	212.1	96.7	62.2	38.2	**83.7**	**78.6**
9	8.9	603.0	152.5	88.5	-	**213.2**	**266.4**
10	10.9	9.1	8.8	13.3	18.9	**12.2**	**4.1**
11	7.9	153.7	29.5	5.4	7.6	**40.8**	**63.8**
12	127.4	95.0	36.7	3.9	-	**65.7**	**55.7**
13	273.2	-	-	-	-	**273.2**	**-**
14	48.0	95.7	6.8	11.4	5.8	**33.5**	**38.9**
15	83.5	120.3	-	-	-	**101.9**	**26.0**
16	110.7	349.6	22.7	442.7	-	**231.4**	**197.2**
17	93.9	148.3	119.7	181.6	309.2	**170.5**	**84.1**
18	32.8	171.0	145.9	269.9	153.8	**154.7**	**84.4**
19	14.3	64.1	12.4	6.8	9.4	**21.4**	**24.1**
20	9.5	4.9	139.5	-	-	**51.3**	**76.4**
21	6.1	14.9	608.0	4.8	12.6	**129.3**	**267.7**
22	60.9	154.9	57.8	58.0	-	**82.9**	**48.0**
23	140.3	170.1	102.6	110.6	94.4	**123.6**	**31.2**
24	109.2	6.8	93.1	54.3	7.4	**54.2**	**47.4**
Mean	**84.0**	**137.0**	**96.2**	**79.8**	**63.4**		
Std Dev	**84.8**	**142.6**	**133.3**	**114.6**	**84.6**		

Table A5-66 - Frustration point experiment, second site responses from unsupervised participants.

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Mean	Std Dev
1	133.0	342.5	157.9	80.9	14.6	4.3	6.9	**105.7**	**121.5**
4	77.1	221.0	104.7	38.2	45.0	16.4	19.0	**74.5**	**71.9**
6	41.3	145.8	34.3	29.6	62.5	4.8	4.2	**46.1**	**48.5**
7	49.1	43.0	49.3	104.2	10.4	18.8	6.1	**40.1**	**33.6**
9	660.4	140.4	-	-	-	-	-	**400.4**	**367.6**
11	8.0	9.3	285.1	8.6	62.9	7.1	65.9	**63.8**	**101.1**
12	31.4	18.7	26.9	4.8	8.3	5.0	-	**15.9**	**11.6**
14	100.5	32.5	102.8	50.4	980.5	200.7	31.7	**214.2**	**343.0**
16	260.1	473.9	67.2	85.5	315.8	150.8	-	**225.5**	**155.8**
17	89.9	170.2	132.6	58.5	198.9	121.2	247.5	**145.6**	**64.9**
18	37.3	88.0	15.0	69.4	36.9	103.9	81.7	**61.7**	**32.5**
19	56.3	149.1	95.0	127.6	10.7	43.2	59.4	**77.3**	**48.9**
21	366.0	10.4	294.6	12.5	61.7	13.8	113.9	**124.7**	**146.7**
23	57.7	112.4	107.0	57.4	113.6	58.3	-	**84.4**	**29.2**
24	41.4	379.0	94.0	108.9	5.1	145.3	-	**128.9**	**132.2**
Mean	**134.0**	**155.8**	**111.9**	**59.7**	**137.6**	**63.8**	**63.6**		
Std Dev	**174.2**	**143.1**	**85.6**	**38.7**	**257.8**	**67.2**	**74.4**		

Table A5-67 - Times around abandonment – comparative time before and after an abandonment act.

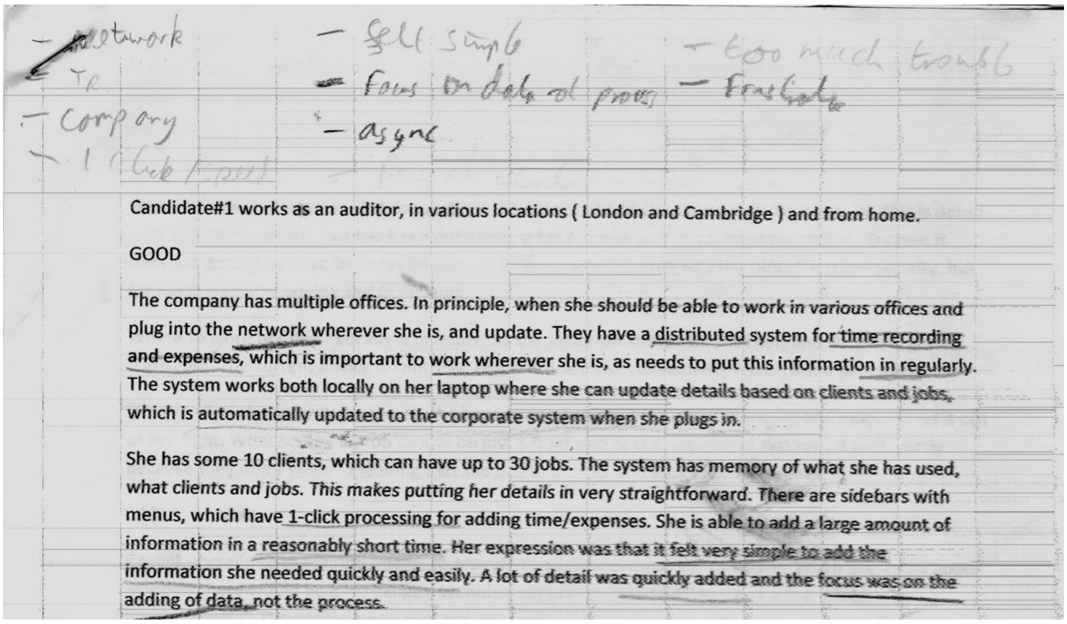
Time before abandoning	Time of abandoned task	Time after abandoning
-74.6	59.0	-15.5
138.8	229.0	184.2
-43.8	62.2	62.2
127.3	127.4	101.7
36.8	120.3	120.3
-207.5	22.7	-118.2
173.3	309.2	163.6
-1.1	153.8	92.1
49.8	64.1	-13.2
140.3	140.3	49.8
-15.3	54.4	-75.5
-13.6	119.9	45.4
-40.0	42.6	3.0
-67.8	38.2	38.2
212.5	442.7	301.8
-1.9	12.4	-65.0
170.1	170.1	79.7
-62.3	7.4	-122.5
-52.9	29.8	-8.0
233.7	473.9	372.7
-7.5	6.8	-70.5
-11.7	108.9	103.8
-28.5	43.0	-5.2
141.2	315.8	-165.0
-4.9	9.4	67.9
41.2	145.3	-145.3

Table A5-68 – Comparison of post-abandonment task activity times, between the two site tasks.

Mean site 1	Mean site 2	Difference
74.5	64.2	10.3
40.1	106.3	-66.2
15.9	57.5	-41.7
225.5	84.3	141.2
145.6	148.4	-2.8
61.7	102.9	-41.2
77.3	26.1	51.2
84.4	80.9	3.5
128.9	41.4	87.6
74.5	84.5	-10.0
40.1	41.8	-1.6
225.5	224.3	1.3
77.3	8.9	68.4
84.4	90.9	-6.5
128.9	25.7	103.2
40.1	37.5	2.6
77.3	7.0	70.3
77.3	7.9	69.4

Appendix 6 - Content analysis process.

The initial interviews analysis process



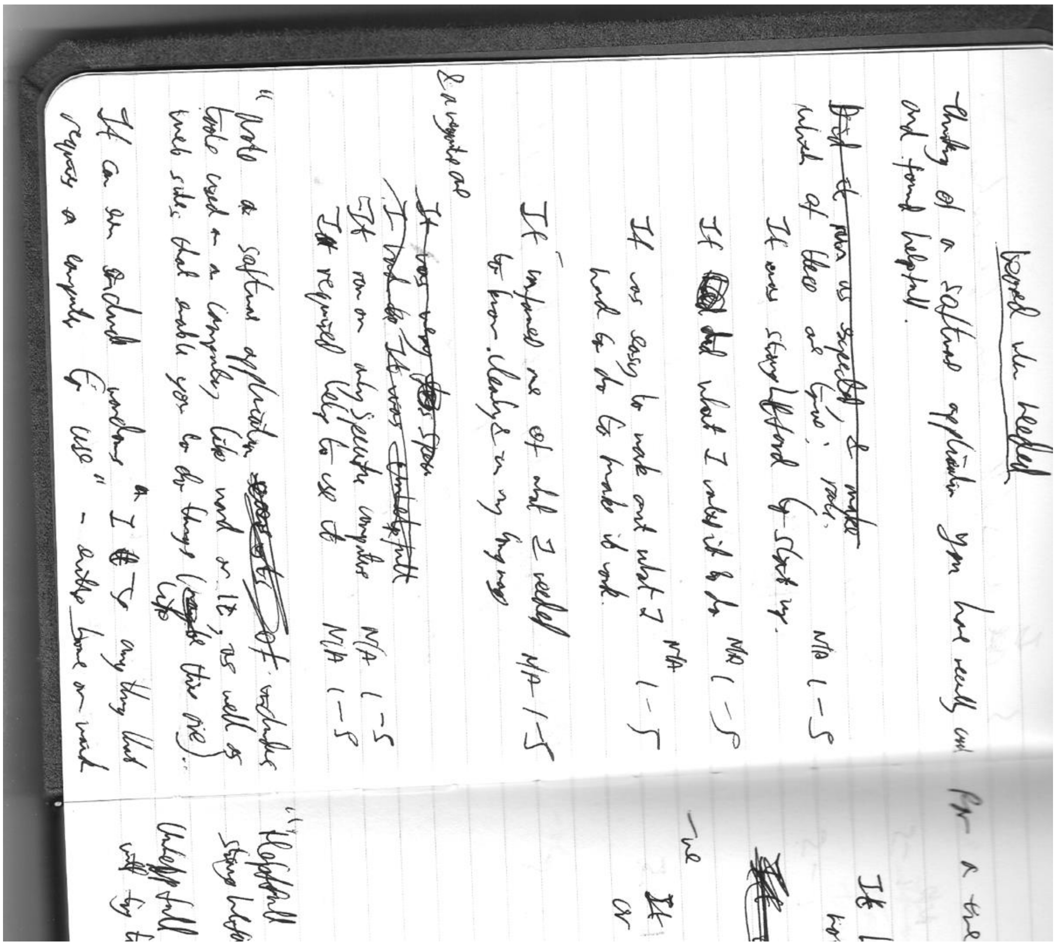
Candidate#1 works as an auditor, in various locations (London and Cambridge) and from home.

GOOD

The company has multiple offices. In principle, when she should be able to work in various offices and plug into the network wherever she is, and update. They have a distributed system for time recording and expenses, which is important to work wherever she is, as needs to put this information in regularly. The system works both locally on her laptop where she can update details based on clients and jobs, which is automatically updated to the corporate system when she plugs in.

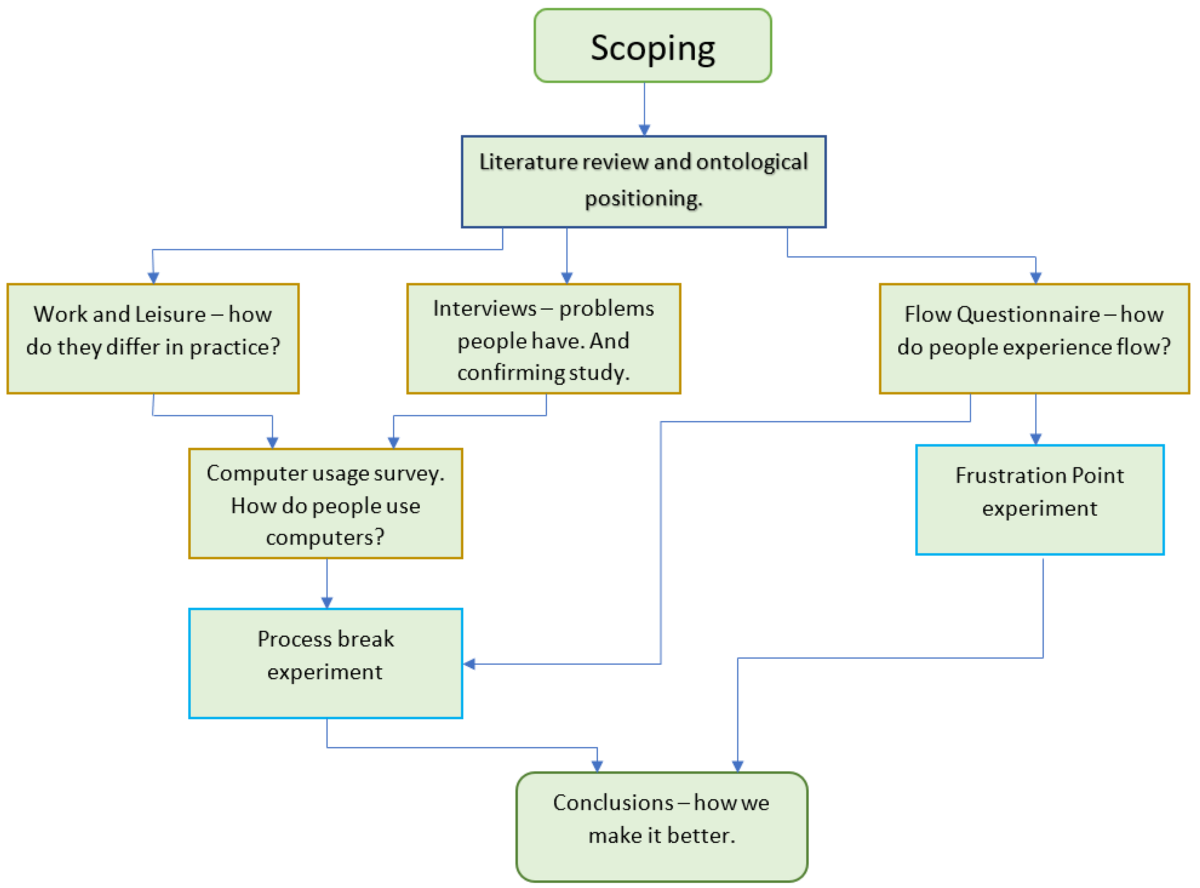
She has some 10 clients, which can have up to 30 jobs. The system has memory of what she has used, what clients and jobs. This makes putting her details in very straightforward. There are sidebars with menus, which have 1-click processing for adding time/expenses. She is able to add a large amount of information in a reasonably short time. Her expression was that it felt very simple to add the information she needed quickly and easily. A lot of detail was quickly added and the focus was on the adding of data, not the process.

The confirmatory analysis image:


Word the needed
Thinking of a software application you have recently used and found helpful.
Which of these are true; rate.
It was straightforward to start up. NA 1–5
It did what I wanted it to do NA 1–5
It was easy to work out what I had to do to make it work. NA 1–5
It informed me of what I needed to know clearly & in my language NA 1–5
It ran on only specific computers NA 1–5
It required help to use it NA 1–5
"Note a software application includes tools used on a computer, like word or IE, as well as web sites that enable you to do things (like this one).
It can be anything that requires a computer to use"
"Helpfull
Unhelpfull

120

Appendix 7 - Methodology flowchart.


Scoping
Literature review and ontological positioning.
Work and Leisure – how do they differ in practice?
Interviews – problems people have. And confirming study.
Flow Questionnaire – how do people experience flow?
Computer usage survey. How do people use computers?
Frustration Point experiment
Process break experiment
Conclusions – how we make it better.