EVALUATING THE VISUAL STUDIO C# INTEGRATED DEVELOPMENT ENVIRONMENT USING HUMAN-COMPUTER INTERACTION PRINCIPLES.

Molaodi Innocent Bosele: Student Number 23290730

Theresia Monyake: Student Number 24488844

Research Project

In

BSc Honours Computer Science

in the

SCHOOL OF INFORMATION TECHNOLOGY

in the

FACULTY OF ECONOMIC SCIENCES AND INFORMATION TECHNOLOGY

at the

North West University (Vaal Triangle Campus)

Supervisor: Malie Zeeman

Co-supervisor: Imelda Smit

21 November 2016

TABLE OF CONTENTS

List of figures

List of Tables

CHAPTER 1: INTRODUCTION

Key Words: *Human-Computer Interaction (HCI), HCI Principles, Visual Studio C#, Graphical User Interface (GUI), integrated development environment (IDE).*

1.1 BACKGROUND

Evaluation of the Visual studio C# (VSC#) Integrated Development Environment (IDE) using Human Computer Interaction (HCI) principles is key to the study and emphasis will be placed on how efficient beginner programmers are able to interact with the VSC# IDE and to what degree the HCI affects their ability to progress in the development of programming skills. The study target first year students who are enrolled for programming module in Visual Studio C# with no previous programing experience. The aim of the study is to develop clear user guideline to promote the application of good HCI principles with the development of an IDE for a programming language.

The study focuses on the application of the HCI principles to promote a clear and desirable user interface for beginner-programmers in a C# programming development environment. Furthermore the study outlines the challenges which the beginner programmer experiences within the C# development environment and aim to show how user guidelines can be used to promote an improved learning environment. For example, with a clear understating of programing IDE layout we can promote a better user experience and ways to promote growth and interests within the c# development space.

.according to Dix *et al.* (2004:3) HCI can be seen as an approach to ensure that computers perform their various tasks while t maintaining usability in the process of interaction with the end-user. Dix *et al.* (2004:3) further elaborates by describing HCI as "*a study of people and computer technology*" and "*how they influence each other*". HCI is commonly associated with developing some sort of a prototype to improve user interface design and user experience (Hinze-Hoare, 2007:3).

Human Computer Interaction consists of a set of principles and theories stated by Hinze-Hoare (2007:3) which are formulated as rules that assess and ensures better computer system usability. The principles and theories furthermore intend to address the challenges faced during the design phase and how they can overcome (Terblanche, 2014b:23).

C#" C-sharp "programming language is one of the most preferred development tool in the Microsoft Visual Studio software suite for its Object Orientated programming for business applications. This study investigates the human computer interaction principles that are applied in the Visual Studio C# IDE programmers used to develop software (Sharp, 2013:1).

1.2 CONCEPTS KEY TO THE STUDY

Human Computer Interaction, Visual Studio C#, Integrated Development Environment, Graphical User Interface, HCI Principles, Learnability.

1.2.1 Human Computer Interaction

The study of how computer systems are implemented through interaction, design. Human Computer Interaction considers how individuals, communities and different organizations are influenced by computer systems. It also encompasses how information can be easily accessed to enhance communication and how to control various computer activities. HCI constantly aims for better supporting interaction methods for end-user activity (Myers, 1998:45).

1.2.2 Human computer interaction principles

Human computer interaction principles are a set of design rules that address the compatibility of user interfaces during interaction with users. HCI principles assume that developers are not well equipped with background knowledge regarding design rules. These principles serve as the intermediary between user and developers, where users expect a user friendly IDE or user interface and developers design. HCI principles allow developers to be considerate of the user’s abilities (Terblanche, 2014b:23).

1.2.3 Integrated development environment

According to Konsynski *et al.* (1984:66) an Integrated Development Environment is an environment that has been packaged as an application program, typically consisting of language-aware editing, project definition facilities, integrated compilation and a debugger. Thus provides various features to help manipulate objects. Furthermore IDEs provides all necessary features within the package and all specifications that the user need for successful interaction. Moreover, an IDE supports and facilitate object-orientated languages (Konsynski *et al.*, 1984:66).

1.2.4 Graphical User Interface

Graphical User Interface Also referred to as direct manipulation interfaces, which require programmers to deal with elaborative graphics and multiple ways of giving the same command, multiple irregular input devices. GUI has multiple graphics that allow for better user interaction for example a mouse and keyboard are necessary components for successful interaction (REIMER, 2005:1).

1.2.5 Visual Studio C#

C# Is pronounced "C sharp" a programming language that is designed for building a variety of applications that run on the .NET Framework. C# is simple, powerful, type-safe, and object-oriented. Microsoft Visual Studio C# is widely used programming language and is well-known for its extremely powerful features (Sharp, 2013:2). The Visual Studio C# 1.0 version was released in 2001, Visual Studio C# 2.0 version in 2005 and Visual Studio C# 3.0 version in 2008. However, it has been very powerful that more and more versions were released and finally the latest is version was released in 2015 (Sharp, 2013:1).

1.3 PROBLEM STATEMENT

First year students find it difficult to interact with the Visual Studio C# IDE and therefore fail to progress in required time of the development of their programming skills. Furthermore, an analysis on the Microsoft Visual Studio C# IDE can reveal why students cannot transition into the actual programming.

1.4 OBJECTIVES OF THE STUDY

1.4.1 Primary Objectives

The primary objective of this study is to evaluate the Visual Studio C# IDE using the HCI principles.

1.4.2 Theoretical Objectives

The theoretical objectives required in order to achieve the primary objective of this study is formulated as follows:

- Gain a better understanding of HCI framework,
- Observe the application of HCI principles,
- Clear background of HCI theories,
- Explore on IDE's in programming,
- Understand in depth the, Microsoft Visual Studio C# programming language by focusing on the graphical user interface.

1.4.3 Empirical Objectives

The empirical objectives for this study aims to collect data on how the implementation of HCI principle in visual studio C# IDE helps to enhance student experience for better learning. These are the empirical objectives:

- Interview Guide Compilation
- A pilot Interview
- Qualitative Interview

1.5 RESEARCH DESIGN AND METHODOLOGY

Research methodology can be defined as a technique used to solve the problem statement of a research (Rajasekar *et al.*, 2006:3). A methodology can be seen as the study of how research can be conducted. However, research methodology is not particularly limited to research methods only, but also involves the context in which the techniques are used and to which extent their relevance is to the study.

A chosen paradigm for the study is an interpretative research paradigm that will address student's perspective based on their interactions using the Microsoft Visual Studio C# IDE.

The study will use a Grounded Theory as a research method. This process will include the evaluation visual studio C# IDE using HCI principles and will require researcher to use conduct interviews using an infinity approach. These are the phases that should follow after the conduction of interviews.

Stage 1: Transcription

In this stage it would be ideal to have decent quality audio device for the interview process. Transcriptions should have huge margins and the line spacing should be sufficient to allow for effective coding as well as note making.

Stage 2: Be familiar with the interview

Familiarization with interview accompanied by good audio advice for transcribing is crucial for interpretation of findings. It helps to also re-listen to the recordings, thus allows the researcher to understand the different perspectives of the research.

Stage 3: Coding

The researcher at this stage is familiar with the concept as well understanding the findings. The researcher will read thoroughly through the transcripts, pay attention to every detail. Therefore, the researcher should be able to classify or find a pattern within the transcripts and recordings. Coding allows researcher to make possible categories that can be used and compared at the end of the research for interpretation purposes.

Stage 4: Develop a working analytical framework

This stage takes place once all researchers have analysed the first set of the transcripts. Researchers all meet and compare specific labels that they came up with, they compare the labels and choose a set of categories of the codes that they agree with. This stage is rather iterative as all the data should be seen as relevant as possible.

Stage 5: Applying the analytical framework

In this stage, the codes and categories that were agreed on are then applied by indexing the transcripts. A code will typically be assigned an alphabet as an easy identification and will be transferred to the transcripts. Atlas T.I is data analysis tool that can be used to speed up the process in terms of quicker retrieval of the data, allows for storage and organization of data as well.

Stage 6: Chart the data into framework matrix

This stage is necessary if the interview takes more an hour and there might be redundant or unnecessary information. Therefore it is important to be as narrow as possible for interview to be of value as well as to reduce unnecessary data.

1.5.1 Data collection methods

Qualitative interviews and observations will be conducted to understand how the students' personal experiences when interacting with the Visual Studio C# IDE and the interview will be semi-structured.

Participants from different educational background and gender are the preferred candidates. Each and every interview will be undertaken by asking the participants questions based on Visual Studio C# experience.

Each interview be properly prepared and planned for.

Participation observation will also be crucial through obtaining of data, on whether the participants are able to show application of HCI principles.

The researchers will start by interviewing C# first year student to evaluate their educational background and level of experience in programing specifically using Visual studio C# IDE.

The position of the study is aligned with the interpretative paradigm. According to Nicola K Gale (2013) an interpretative research has six (6) stages to be followed as procedure to analyse data:

1.5.2 Participants and participant selection

Participants of the study will be randomly selected from a first year IT students In C# based on a test a random selection of participants will be based on number of students who took the pre-test. 10 students will be selected participate in the interviews and observation.

1.5.3 Data analysis methods

The axial coding method, which is “set of procedures” where data is collected and placed together in categories to help make connections (Kendall, 1999:746). This method will assist where different answers to each question of the interview will be compared to gather understanding of the situation. Atlas T.I a coding tool will be utilized in this study for interview interpretations and to classify information on transcripts into categories and possible themes.

1.6 ETHICAL CONSIDERATION

All student that will be chosen as participants will not have to mention their personal particulars. Each individual will be treated fairly and the same with others. If a participant successfully completes the interview, they will be greatly appreciated. The privacy of students is accommodated for as well as security of the students during the conduction of interviews. Relevance as consideration as well to ensure that the relevant students who are directly affected by the Microsoft Visual Studio C# IDE which are ITRW 123 students. Lastly compliance is considered in order to get approval for conducting the research from the Ethics board.

1.7 CHAPTER CLASSIFICATION

Chapter 1 Introduction and background to the study: Purpose of the study, an overview of HCI, problem identification and brief description of logical flow of the study.

Chapter 2 Literature review: Theoretical background of HCI and concepts of the study explained in depth.

Chapter 3 Research design and methodology: Consists of the interpretative paradigm, case study as a research method, participant selection, qualitative interviews and axial coding as a data analysis technique.

Chapter 4 Empirical Study: Conduction of Qualitative interviews as data collection methods and interpretation of information using coding. This is the process of findings.

Chapter 5 Conclusions and Recommendations: Based on the findings, guidelines and recommendation will be addressed.

1.8 WORK LOAD DIVISION

Work is distributed amongst the researchers (Bosele and Monyake) with a follow up peer evaluation at the completion of the study.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

The objective of this chapter is to establish a theoretical framework in Human-Computer Interaction (HCI) and review the literature relating to the critical focal section of the research area the design rules focusing on principles affecting system learnability. According to Boote and Beile (2005:2) a literature review is the evaluation report of the studies relating to the area of the research. Luciano (2011:303) further defines a literature review as a process of gathering the study objectives that explains the specific approach to the research topic.

The main goal for the literature review for this study is to have a clear and broad overview and understanding of the theoretical concepts relevant to the study. The main concepts the study will be focusing on are HCI, HCI principles; Graphical User Interfaces (GUI), Integrated Development Environment (IDE) and the computer programming language Visual Studio C#.

The secondary objective for this study include understanding the relationship among the HCI design rule on learnability as a HCI concept and to have a clear view on how the HCI framework influence the design phase of the programming environment for HCI purposes.

2.2 INTERGRADED DEVELOPMENT ENVIRONMENT

The Intergraded Development Environment provides comprehensive facilities to system developers for application development. The package available includes tools such as debuggers, code editor, build in automation and compilers.

IDE requires a considerate critic in order to determine whether the tool is fit for purpose by considering aspects such as the infrastructure for support and end user experience for optimal usage. Konsynski *et al.* (1984:65) states that Integrated Development Environment represents undefined set of information system development, computer aided and system development techniques. The software application aid and provide comprehensive object oriented facilities for software development. IDEs provide all tools needed into one workspace by so creating programming environment awareness and representation of an ideal development set for software developers (Konsynski *et al.*, 1984:66).

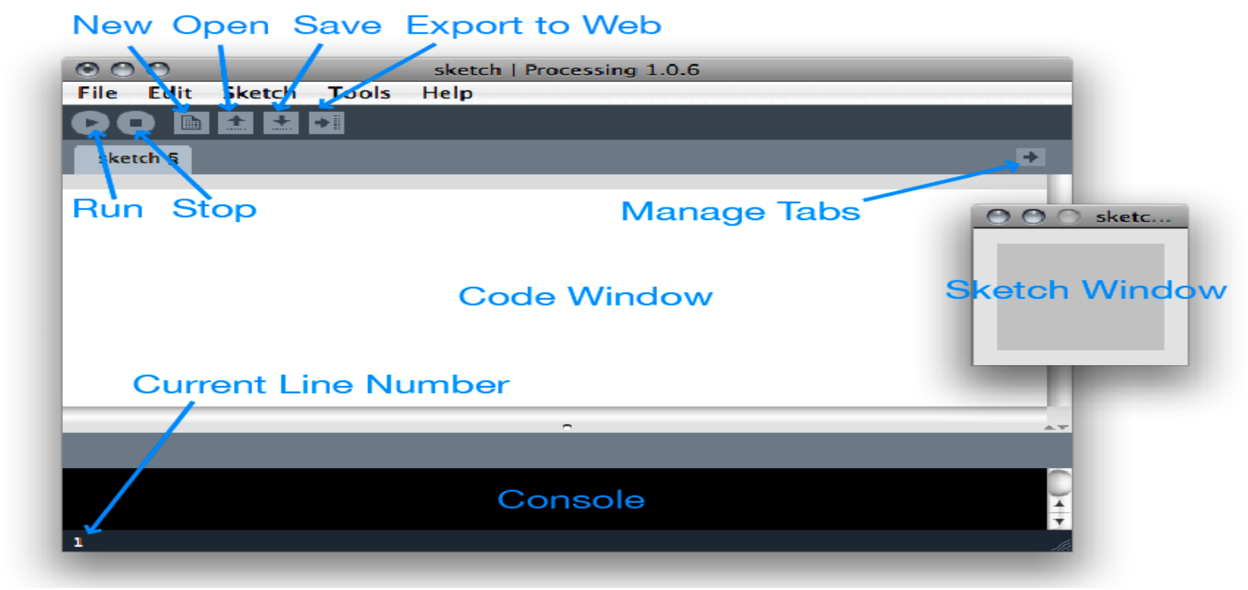
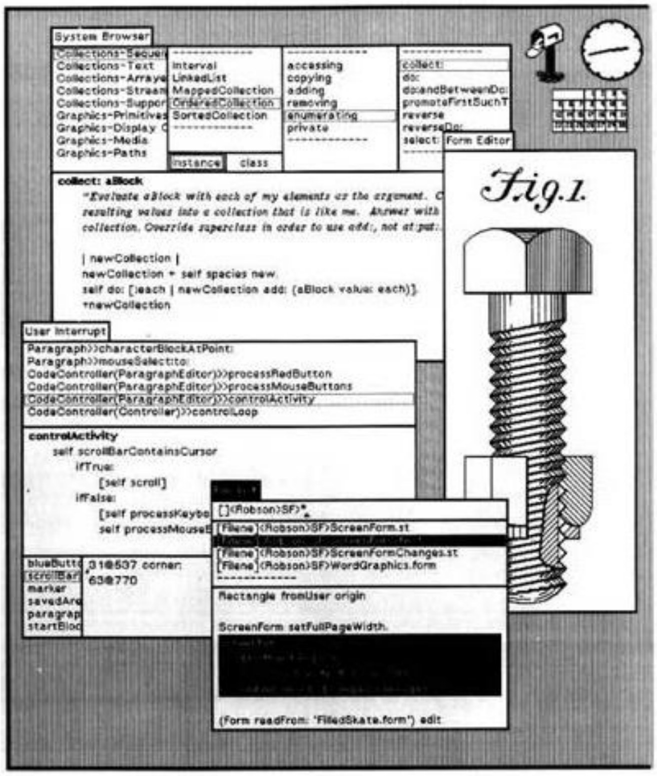


Figure 2.1: IDE Design Computation (Nee &Nest, 2012:1)

2.3 GRAPHICAL USER INTERFACE

Graphical User Interface can be classified as a type of a user-interface and therefore have led to speculations as to why programmers decided to adopt GUI features and how they would become the primary interaction or medium between the end-user and computer. According to Chen and Schoeneman (2004:2) in the 1930s Vannevar Bush was one of the few people who were inquisitive about the concept of GUI and during that era a device called “Memex” was written. The device would look like a table with touch screens representations, a scanner and keyboard attached to it (REIMER, 2005:2). Memex would allow the end-user to obtain all knowledge through interaction, but the digital computer at the time had not been created. However, in the 1937 digital computers' construction began and Douglas Englebart who was an engineer at the time decided to think about ways in building a machine that could enhance a person thinking and application. He became well known for believing that the fastest way to improve people's abilities to solve complex problems was through a digital computer.

The first modern GUI that was developed was the Smalltalk and had been perceived as a programming language that anyone could understand. It was globally recognized as the first object-oriented programming language. It was also well-known for improved Java-like features that assisted the programmer such as automatic memory management (REIMER, 2005:2). (See figure 2.2)



```
collect: aBlock
    "Evaluate aBlock with each of my elements as the argument. C
    resulting values into a collection that is like me. Answer with
    collection. Override superclass in order to use add:, not at:put:.

    | newCollection |
    newCollection ← self species new.
    self do: [:each | newCollection add: (aBlock value: each)].
    ↑newCollection
```

```
Paragraph>>characterBlockAtPoint:
Paragraph>>mouseSelect:to:
CodeController(ParagraphEditor)>>processRedButton
CodeController(ParagraphEditor)>>processMouseButtons
CodeController(ParagraphEditor)>>controlActivity
CodeController(Controller)>>controlLoop
```

```
controlActivity
    self scrollBarContainsCursor
        ifTrue:
            [self scroll]
        ifFalse:
            [self processKeybo
            self processMouseB
```

```
31@537 corner:
63@770
```

```
[]<Robson>SF>*
[Filene]<Robson>SF>ScreenForm.st
[Filene]<Robson>SF>ScreenFormChanges.st
[Filene]<Robson>SF>WordGraphics.form
-------------
Rectangle fromUser origin

ScreenForm setFullPageWidth.

(Form readFrom: 'FilledSkate.form') edit
```

Figure 2.2: The First modern GUI (REIMER, 2005:3)

The graphical user Interface has evolved throughout the years and was called "The Lisa user interface" by Apple since 1979 in its early development phases (REIMER, 2005:3).

The user interface had made numerous improvements such as, shortcuts on a keyboard for frequently utilized commands and checkmarks or well known as checkboxes next to menu items on the “On-Line System” that was developed. The Lisa user interface was the first interface that could display all types of files in the file system, where each file could be browsed through a directory (REIMER, 2005:3). Moreover, it developed the concept of “drag-and-drop” to manipulate files. All these improvements have led to the advanced GUIs used today and this figure below show how GUIs have evolved throughout the years.

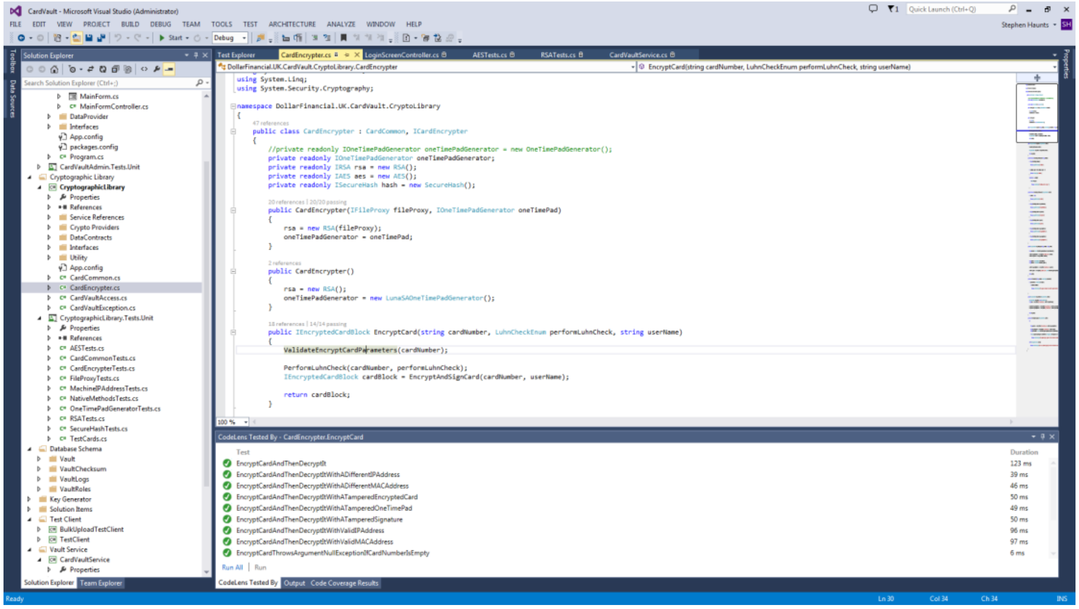
GUI is stated by Chen and Schoeneman (2004:2) as consumer piece of a program where end-users are able to communicate with the program. There are two parts involved in the GUI namely, the representation and the behavior. Representation explains the methods in which input devices are presented however, the behavior defines the relationship between end-user and the system’s responses (Chen & Schoeneman, 2004:2).

2.4 VISUAL STUDIO C#

Microsoft’s Visual Studio C# is a well-known dominant language that is widely used by mostly developers to create various applications. A big portion of the program’s features and components is inherited from Microsoft Visual Basic and C++ languages. The Visual Studio C# 1.0 version was released in 2001, Visual Studio C# 2.0 version in 2005 and Visual Studio C# 3.0 version in 2008. However, it has been very powerful that more and more versions were released and finally the latest is version was released in 2015 (Sharp, 2013:1).

Visual Studio C# has many features from being able to build extremely accessible application to its ability to allow data sharing, connectivity and collaboration among users (Sharp, 2013:1). It has powerful features that were built to be less complex and ambiguous. Visual Studio C# is has strong visualized IDE and is considerate in terms of user interaction. C# developers continuously allow users to critique and give feedback and strive to deliver the best language that will allow users to be comfortable yet productive.

Visual Studio C# IDE was chosen in this study due to the fact that is one of the core programming modules at North-West University and generally widely known programming language.


CardVault - Microsoft Visual Studio (Administrator)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Solution Explorer
Test Explorer
CardEncrypter.cs
LoginScreenController.cs
AESTests.cs
RSATests.cs
CardVaultService.cs
DollarFinancial.UK.CardVault.CryptoLibrary.CardEncrypter
EncryptCard(string cardNumber, LuhnCheckEnum performLuhnCheck, string userName)
CodeLens Tested By - CardEncrypter.EncryptCard
Run All | Run
CodeLens Tested By Output Code Coverage Results
Ln 30 Col 34 Ch 34 INS

```
using System.Linq;
using System.Security.Cryptography;

namespace DollarFinancial.UK.CardVault.CryptoLibrary
{
    47 references
    public class CardEncrypter : CardCommon, ICardEncrypter
    {
        //private readonly IOneTimePadGenerator oneTimePadGenerator = new OneTimePadGenerator();
        private readonly IOneTimePadGenerator oneTimePadGenerator;
        private readonly IRSA rsa = new RSA();
        private readonly IAES aes = new AES();
        private readonly ISecureHash hash = new SecureHash();

        20 references | 20/20 passing
        public CardEncrypter(IFileProxy fileProxy, IOneTimePadGenerator oneTimePad)
        {
            rsa = new RSA(fileProxy);
            oneTimePadGenerator = oneTimePad;
        }

        2 references
        public CardEncrypter()
        {
            rsa = new RSA();
            oneTimePadGenerator = new LunaSAOneTimePadGenerator();
        }

        18 references | 14/14 passing
        public IEncryptedCardBlock EncryptCard(string cardNumber, LuhnCheckEnum performLuhnCheck, string userName)
        {
            ValidateEncryptCardParameters(cardNumber);

            PerformLuhnCheck(cardNumber, performLuhnCheck);
            IEncryptedCardBlock cardBlock = EncryptAndSignCard(cardNumber, userName);

            return cardBlock;
        }
```

```
Test                                                        Duration
✓ EncryptCardAndThenDecryptIt                               123 ms
✓ EncryptCardAndThenDecryptItWithADifferentIPAddress        39 ms
✓ EncryptCardAndThenDecryptItWithADifferentMACAddress       46 ms
✓ EncryptCardAndThenDecryptItWithATamperedEncryptedCard     50 ms
✓ EncryptCardAndThenDecryptItWithATamperedOneTimePad        49 ms
✓ EncryptCardAndThenDecryptItWithATamperedSignature         50 ms
✓ EncryptCardAndThenDecryptItWithValidIPAddress             96 ms
✓ EncryptCardAndThenDecryptItWithValidMACAddress            97 ms
✓ EncryptCardThrowsArgumentNullExceptionIfCardNumberIsEmpty 6 ms
```

Figure 2.3: Visual Studio C# IDE (Haunts, 2013)

2.5 HUMAN COMPUTER INTERACTION

This section introduces the concept of HCI principles and how it can be applied on software development in Visual Studio.NET. The context of HCI defines the interaction between man and machine in general scope. HCI can be defined as the interaction between the system user and the system which focuses on how users experience system functionality (Dix *et al.*, 2004:3). The research in the field of HCI focus on how human interact with computers in attempt to create software application so that users have better computer interaction experience. The origin of HCI is derived from the early 1980. The concept started to have an impact late in the 1990s on the functional operations when compute Graphic User interface were established (Myers, 1998:46).

Only a number of individuals which included computer engineers and scientist had the privilege to access computers. However, Myers (1998:48) asserts that the computer scientists and engineers had not been aware that they were already practicing interaction back then. By the 1980s computers had immensely improved and interaction was gradually becoming a practice. Computers had more usability features and became a powerful tool for businesses (Myers, 1998:49)

MacKenzie (2012:5) states that the rapid growth on the implementation of HCI principles it's due remarkable timeliness user experience and resulted in a major growth in software development. Figure 2.4 demonstrates the timeline in HCI history.

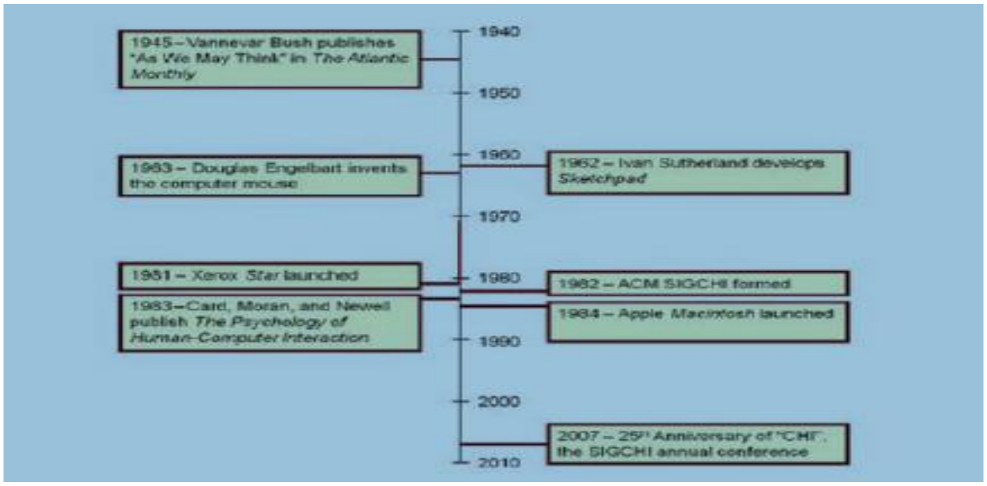

1940
1945 – Vannevar Bush publishes "As We May Think" in The Atlantic Monthly
1950
1960
1963 – Douglas Engelbart invents the computer mouse
1962 – Ivan Sutherland develops Sketchpad
1970
1980
1981 – Xerox Star launched
1983 – Card, Moran, and Newell publish The Psychology of Human-Computer Interaction
1982 – ACM SIGCHI formed
1984 – Apple Macintosh launched
1990
2000
2007 – 25th Anniversary of "CHI", the SIGCHI annual conference
2010

Figure 2.4: HCI overview timeline (MacKenzie, 2012)

2.6 INTERFACE DESIGN RULE

The study will outline the fundamental building blocks of the Interface design rules in HCI.

The themes include Learnability, Flexibility, Robustness and HCI patterns, however for the purpose and application of this study, the focus is Learnability as we use and aim to improve student Visual Studio C# programing experience.

The design rule can be defined as the sets of instruction which offers developers a framework to meet targeted development standards with the aim to improve user experience by making learnable, interactive and consistent Graphical interfaces.

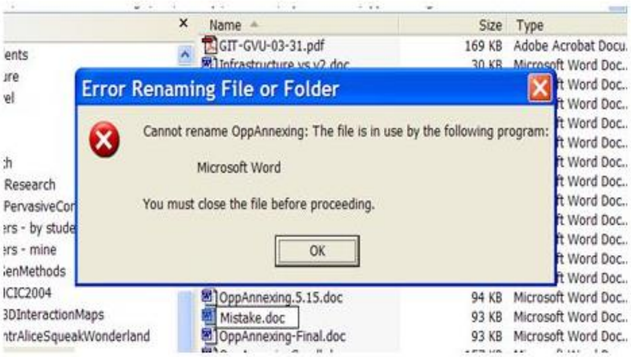
Learnability is the ability for new users to easily use and interact with the system to achieve maximum usage performance and study will focus on the five (5) principle affecting learnability Dix *et al.* (2004:260).

2.7. HCI PRINCIPLES

HCI principles were developed from various theory-based knowledge, common sense and user experience (www.baddesigns.com, 2011). These principles are guidelines that developers must consider during user interface design. The guidelines are aimed at eliminating not only ambiguous design but complexities that end user come across in various interactive designs (Terblanche, 2014b:23). Furthermore, HCI principles encompasses a set of *design rules* that which are required that assist designers to be aware of usability issues that arise when interaction takes place. In most cases, designers do not often have the background knowledge in cognitive science, design rules and ergonomics which should be carefully understood prior designing any interactive user interfaces (Dix *et al.*, 2004:260). This study aims to understand the most common principles effecting learnability (Synthesizability, Consistency, Predictability, Familiarity and Generalizability)

2.7.1 Synthesizability

The ability for the user to build on how a system operates. In other words it is the process of how the user can understand what can happen after a certain operation or certain changes. The system has to be able to be informative in terms of any changes the user makes and guide the user if there are potential errors.



Error Renaming File or Folder

Cannot rename OppAnnexing: The file is in use by the following program:

Microsoft Word

You must close the file before proceeding.

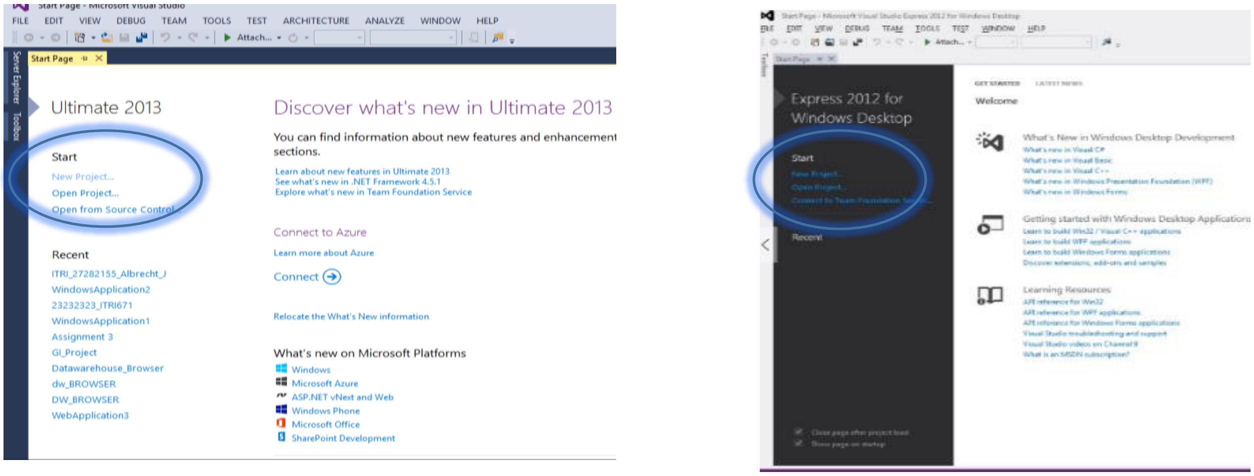
OK

Figure 2.5: Example of Synthesizability

The figure demonstrates a message dialogue box that communicates with the user when changes or errors are detected.

2.7.2 Consistency

This principle concentrates on the comparability in information and yield conduct rising up out of equivalent circumstances or task objectives (Hinze-Hoare, 2007:10). What is truly inferred here is that when the layout is constantly b the same, due to the fact that users rely on consistent user interfaces, the user will gradually be familiar with the IDEs. The user can perceive what will happen in light of before experiencing another type of IDE or user interface. Moreover consistency can apply to IDEs in terms of various layouts and designs (See in figure 2.8)



Discover what's new in Ultimate 2013

You can find information about new features and enhancement sections.

Learn about new features in Ultimate 2013
See what's new in .NET Framework 4.5.1
Explore what's new in Team Foundation Service

Connect to Azure

Learn more about Azure

Connect

Relocate the What's New information

What's new on Microsoft Platforms

What's New in Windows Desktop Development

What's new in Visual C#
What's new in Visual Basic
What's new in Visual C++
What's new in Windows Presentation Foundation (WPF)
What's new in Windows Forms

Getting started with Windows Desktop Applications

Learn to build Win32 / Visual C++ applications
Learn to build WPF applications
Learn to build Windows Forms applications
Discover extensions, add-ons and samples

Learning Resources

API reference for Win32
API reference for WPF applications
API reference for Windows Forms applications
Visual Studio troubleshooting and support
Visual Studio videos on Channel 9
What is an MSDN subscription?

☑ Close page after project load
☑ Show page on startup

Figure 2.6: Microsoft visual studio in 2012 and 2013 (Zak, 2015:129)

The Microsoft Visual studio C# IDE for both versions has a consistent layout, similar design and user is able to relate to each package. The user must be able to have the same understanding throughout thus crucial for IDEs to be similar in functionality

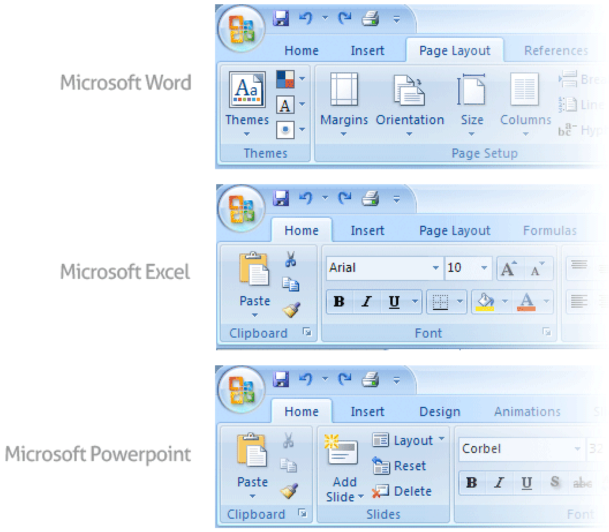


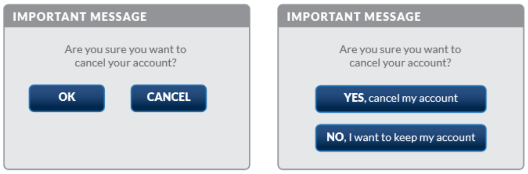
Figure 2.7: Example of Consistency

The figure demonstrates a consistent menu bar in Microsoft Office.

2.7.3 Predictability

Predictably can be defined as the reference to the level of degree at which correct prediction can be made regarding the system. According to (Dix *et al.*, 2004:262) predictability determines the operational visibility of the system.

The user's knowledge of previous system history is sufficient to help determine the system's future behaviour.



IMPORTANT MESSAGE

Are you sure you want to cancel your account?

IMPORTANT MESSAGE

Are you sure you want to cancel your account?

The one on the right has labels that are **meaningful**; it uses active verbs. This combination allows the user to easily predict the outcome.

Figure 2.8: Example of predictability

In most cases during interaction, users need to have some knowledge by having the ability to remember how they interacted with an IDE. The user's knowledge must be sufficient enough to allow the user to remember the functionality and usability of the system based on past experience (Noyes, 2008:3). Thus interaction improves gradually as user engages in future. However, predictability is often centred on the user and not highly dependent on the ability of the system to predict for the user, it is therefore crucial that the user can identify and easily remember certain actions that they applied in the past based on the simplicity and usability of an IDE.

2.7.4 Familiarity

Familiarity has to do with the first impression that the user encounters when they interact with a system and how the user can be able to initiate interaction (Dix *et al.*, 2004:262). Familiarity principle encompasses some of the generalizability principle as well, whereby a user's has the ability to interact with a system based on other knowledge gained from systems that users interacted with in the past. Therefore, the knowledge that the user gained in previous interactions can offer insight for better interaction in the future (Terblanche, 2014b:23).

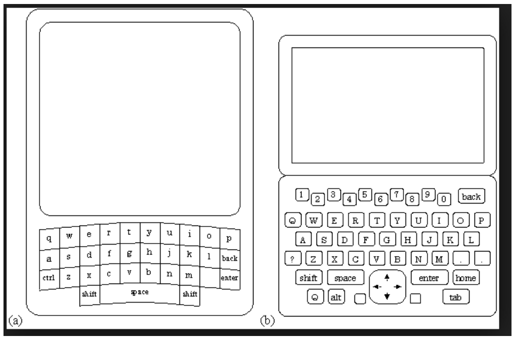


Figure 2.9: Example of familiarity (Mackenzie, 2012)

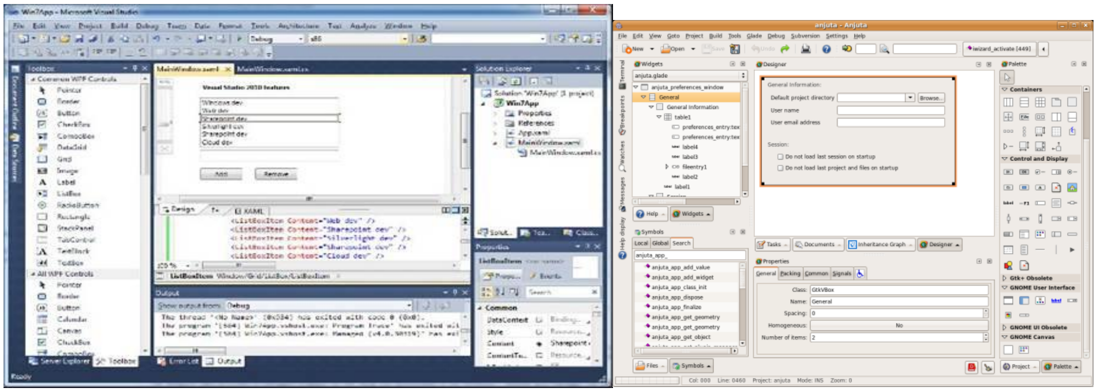


Figure 2.10: Matching Users expectations (Aspainall, 2007)

The above example applies on how prior knowledge compare to the new system , in this case students will be tested on their previous knowledge of systems development and how it impact on understanding the visual studio C# programming language.

2.7.5 Generalizability

Generalizability can be seen as the extension of research results and conclusions from a specific study. According to (*Dix at al*, 2004:262) generalizability is defined as the process of extending specific interaction knowledge to a new but similar   situation.

Generalizability principle improves user's cognitive capabilities and helps to develop predictive system models. It also provide a platform for consistency.

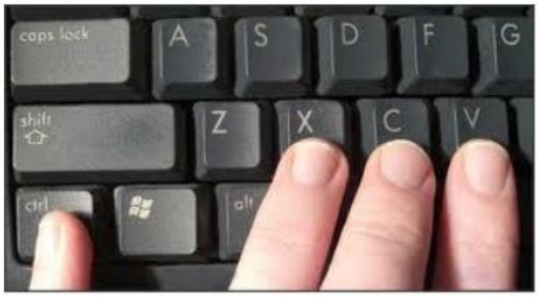


Figure 2.11: Example of Generalizability Cut/copy/paste (Aspinall, 2007)

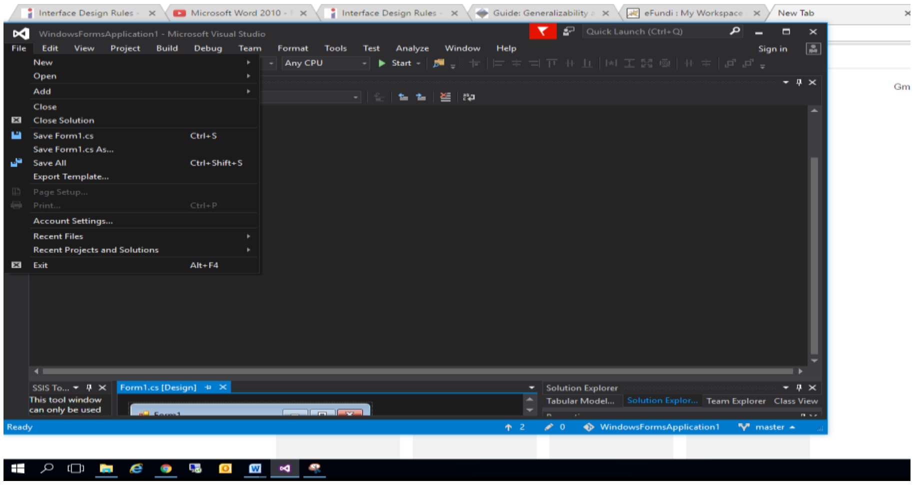


Figure 2.12: Example of short cuts keys in Visual Studio C# IDE (Visual studio, 2016)

The figure shows generalizability functions available on Visual Studio for users to use and can be aided by having prior usage experience form other vendor products in the market.

2.8 CONCLUSION

The purpose of this chapter was to explain in detail the purpose of literature review with specific inference form available resource. Firstly the review explains the context of literature, on what is about and why is it relevant to this study. Secondly the review explains in details on how different concepts fit together in terms of definitions, keywords, graphical presentation and give insights about the chosen field of study and relevancy. The next chapter will be focused on literature review on the research methodology and specific to empirical analysis on data collection methods to be conducted.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 INTRODUCTION

A research methodology is a structured approach to address a research problem (Kothari, 2004:8). The researchers have to know which methodology is applicable for their particular research. The research methodology includes number of aspects i.e. to understand the concept; purpose of the research methodology and to understand the concept of research as well as the research process. There are four research paradigms that serve as a basic structure of a research paper, which will be discussed. In the process of the methodology the interpretative research principles should reveal some insight on how to implement an interpretative research. There are various types of research methods and the study considers grounded theory, which will be elaborated on. Data collection methods and analysis methods will also be discussed in this chapter.

3.2 THE CONCEPTS OF RESEARCH

According to Rajasekar *et al.* (2006:2) research is an orderly or scientific search for valid information on a particular topic for a study. Wilson (2008) states that research is a focused or tailored investigation that aims to discover new facts in a specific field. Ultimately, research is said to be common in having the “inquisitive” aspect (Wilson, 2008). Research contains various elements from defining to understanding a problem as well as redefining them to develop a hypothesis or research statement. The elements also involve data collection to analyze results in order to recommend or give guidelines.

Every research has a certain flow. Figure 3.1 illustrates the overall standard guidelines of the research process. The figure indicates how the research process has a number activities and how the activities are sequenced. In most cases they tend to overlap with each other. But the general procedure of the tasks is as follows; formulate of the research problem, literature review, formulate the hypothesis, research design, sample design, collection of data, implementation of the project, clear analysis data, test the hypothesis, interpret the results and lastly prepare the report.

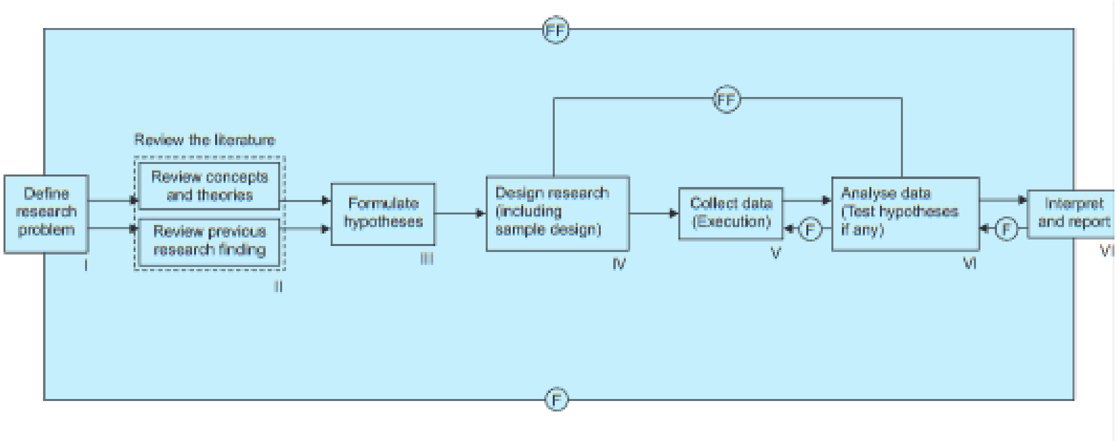

FF
FF
Review the literature
Define research problem
I
Review concepts and theories
Review previous research finding
II
Formulate hypotheses
III
Design research (including sample design)
IV
Collect data (Execution)
V
F
Analyse data (Test hypotheses if any)
VI
F
Interpret and report
VI
F

Figure 3.1: The research process (Kothari, 2004:11)

3.3 RESARCH PARADIGMS

A paradigm is a structure or model about the nature of knowledge and its existence generated from a global view or a belief system (Cohen & Crabtree, 2008:331). Paradigms give guidelines on a number of ways which a research can be conducted and these paradigms are shared among various communities. Table 3.1 below provides types of paradigms and in which context they should be used. The table summarizes the paradigms.

Table 3.1: Research Paradigms (Cohen & Crabtree, 2008:332)

Paradigm	**Ontology**	**Epistemology**	**Question**	**Method**
Positivism	Hidden rules govern teaching and learning process	Focus on valid and reliable tools to undercover rules	What works?	Quantitative

Interpretive/ constructivist	Reality is created by people in groups	Discover the underlying meaning of events and activities	Why do you act in this manner?	Qualitative
Critical	Society is common with inequalities and injustice	Help undercover injustice and empowering citizens	How can I change this situation?	Ideological review, Civil actions.
Pragmatic	The truth is what is useful	The best method is one that solves problems	Will this intervention improve learning?	Mixed Method, design-based.

3.3.1 Positivism research paradigm

Positivistic research is also known as quantitative research and is conducted using experimental techniques, quantitative measures in order to test or examine the hypothesis (Golafshani, 2003:598). The hypothesis is the researcher's prediction of the outcome; usually the null hypothesis will have a negative or opposite prediction that is expected. This paradigm is also concerned with the analysis and the measurement of the relationships among the variables. The variables are measured through understanding how changes occur as more variables are added or eliminated from the experiment. Golafshani (2003:599) further elaborates that the variables consist of independent and depend variables whereby the relationship determines the final decision. The final decision depends on the null hypothesis: if it is rejected, the researcher's prediction is correct; if the null hypothesis is accepted the researcher's prediction is incorrect.

3.3.2 Interpretative Research Paradigm

Interpretative research is also known as qualitative research, associated with understanding a concept in a holistic view. The objective of interpretative research is to magnify one understands when inquisitive on a subject. Interpretative research is used often because it allows researchers to intensely understand a phenomenon; interpret overall the study by having a wide range of meanings that could be understood. Johari (2009:26) asserts that interpretative is a stronger paradigm than positivist research as it can accommodate more age groups. Furthermore, Johari (2009:27) states that interpretive research is even more imperative due to the fact that the information and knowledge gained revolves not only around the study of information systems but also considers other contexts such as cultural aspects of the study.

3.3.3 Critical Social Theory paradigm

Critical social theory focuses not only on general theoretical challenges but the solid current affair challenges. This theory is a school of thought and its main objective is develop the human circumstance (Ngwenyama, 1991:271). This research approach is intended to be a different approach that considers individuals to critique an existing system or an artifact from an information systems perspective. Ngwenyama (1991:272) declares that CST concentrates on finding the alternatives to social circumstances that will consider human desires. It considers the challenges faced by society and ultimately comes to a conclusion that addresses and gives guidelines on ways to improve the social conditions. This approach asks individuals to extend their opinions to technology and ensures that there is some rationality when artifacts continuously advance.

3.3.4 Design Science Research Paradigm

Design can be defined as to make and to “bring into being” (Vaishnavi & Kuechler, 2015) . Design involves creation of original artifacts that do not already exist. The design of artifacts has been acknowledged as an activity throughout the years and has been significant from education, medicine to law which are all focused on the design of artifacts. DSR is a gradually growing field as more researchers could not only analyse existing artifacts but would rather design better suitable artifacts that will serve as better technology. According to Vaishnavi and Kuechler (2015) DSR is the process of learning and building simultaneously based on the learnings gained.

3.3.5 Position of the Study

In the process of evaluating of the Visual Studio C# IDE, the interpretative research paradigm is relevant, as the study analyses the IDE as participants are required to share their experience and to observe certain tasks. Interpretative research is used to describe the Grounded Theory as a method and to further describe qualitative research methods. The focus of this study is to evaluate the Integrated Development Environment (IDE) in C# using HCI Principles through understanding the user's view on Visual studio C# through qualitative data gathering from Interview and observations..

3.4 RESEARCH METHODOLOGY

A research methodology can be defined as a technique used to solve a problem statement of a research (Rajasekar *et al.*, 2006:3). Research methodology can be seen as the study of how research can be conducted. However, research methodology is not particularly limited to research methods only, but also involves the context in which the techniques are used and to which extent their relevance is to the study.

In this study the researcher's aim to evaluate how students interact with the visual studio C# IDE using the interpretative paradigm. During the interaction process the researchers gain insight on certain struggles that students face and overall understanding the C# IDE from the user's perspective. The participant is a large contribution to the study and interpretivists believe that knowledge is gained through interaction and language. The interpretative paradigm proposes that it is crucial to analyze social activity from user's point of view on the subject of the study (Tracy, 2012:14).

The Interpretative paradigm consist of a seven interpretative research principles that are fundamental for the interpretive research process, Klein and Myers (1999:72) discuss the principles as follows:

3.4.1 The fundamental principle of the hermeneutic circle

The aim of the principle is to understand certain complexities based on presumptions of meanings and how these complexities relate to other parts of the study. This principle can be used as approach to elaborate on wide variety of research questions such as the primary and secondary questions of the study. The hermeneutic circle thus suggests that a study can be broken down into parts and can be understood individually that allows the researchers to understand the whole study (Klein & Myers, 1999:72). For example how the researchers can understand the students' point of view in different domains, through interviews and observations.

3.4.2 The principle of contextualization

This principle is concerned with understanding what is meant by the author of a text and that of the interpreter of the text. The solution here is to understand the meaning of the text in the modern context of the study and can be done by understanding of the domain of the study (Klein & Myers, 1999:73). For example understanding the literature of the study and interpreting it by asking viewers about what they can comprehend of the study.

3.4.3 The principle of interaction between the researcher and subjects

This interaction principle addresses how collected data and interpretation have a direct impact on each other. This principle will be carefully considered when interviews and observations take place in order to collect data. The data will then be compared to how it is interpreted based on the findings (Miskon *et al.*, 2015:6).

3.4.4 The principle of abstraction and generalization

This principle requires the researchers to find a link between how data is interpreted based on the application of theory. The data collected is then explained in terms of the concepts of the study that can accommodate human understanding (Miskon *et al.*, 2015:5).

3.4.5 The principle of dialogical reasoning

This principles entails the ability of the researchers to challenge their biased reasoning that lead to the original research design. It also supposes that the findings during data collection process may differ to the original preconceptions of the study. This can possibly lead to the findings not having ability to support the study ultimately (Miskon *et al.*, 2015:6). For example, the interview questions should be biased.

3.4.6 The principle of multiple interpretations

This principle aims to ensure that data collected from numerous sources including sources from interviews to other supporting documents, is interpreted correctly throughout. It also addresses the importance of types of coding and how they may affect interpretations ultimately. It claims that interpretations of data is sensitive and should be treated with utmost attentiveness (Miskon *et al.*, 2015:6). For example considerations of other methods of coding such as manual coding

3.4.7 The principle of suspicion

This principle emphasis the importance of data collection and the analysis where the researchers does not consider the interviewee’s answers at first glance, but considers a follow up and other insights that support the study. This can be achieved by triangulating around the study to find as much information as possible from the same person (Miskon *et al.*, 2015:6). It is important for the researchers to consider all the participants views on the subject of the research.

3.5 RESEARCH METHODS

Research method for this study applies grounded theory principles. Grounded theory is explained in terms of how data will be collected and analysed. It also touches on the various stages to be followed by the researchers in an interpretative research and how the empirical study take place.

3.5 .1 Grounded Theory

Grounded theory is a qualitative method that is used to assist the researchers with the collection and analysis of data (Charmaz & Smith, 2003a:54). Grounded theory is a repetitive process and it also compares information obtained from interviews to relate it to the study during the empirical process. It is an iterative process that allows the researchers to critically revise the literature review especially if the literature has some unnecessary content that will not yield any value during data analysis (Charmaz & Smith, 2003b:54). Grounded theory aims to re-assure that every detail is featured in the empirical study by addressing the gaps identified in the findings on the HCI principles. The theory suggests that the type of data that will be collected is aligned with the literature. It asserts that there is a possibility that after a first cycle of interviews, there might be random information revealed.

The information from students may differ immensely and possibly be contradicting. Therefore, grounded theory notices information that may not form part of an expected pattern during data analysis; it allows random answers to be included in the empirical study. In some instances during data analysis, grounded theory acknowledges that some answers reveal valid insights to a study. Thus the new insight can be of great value to a study. The study applies grounded theory in terms of continuous cycles of interviews through an infinity approach, where there are reflections on the type of data gathered; addressing data obtained that may differ and contradict from other data gathered.

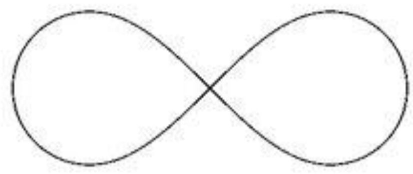


Figure 3.2: Grounded Theory (Botes, 2014)

Figure 3.2 demonstrates the infinity approach, For example, after a cycle of interviews one participant can mention a different factor such as *the structure* in terms of the consistency on the Visual studio C# IDE which had not been mentioned in the interviews at that point. *The structure* perhaps has not been considered as a code in the consistency principle. Grounded theory also suggests that the researchers revises the literature and validate whether or not *the structure* is elaborated on or not. It proposes that the researchers refer to the literature and considers how the factor should be addressed. Through the infinity approach the researchers become aware and random factors can be reflected on as possible codes in the analysis. Thus infinity approach proposes a continuous process of reflecting and revising the study.

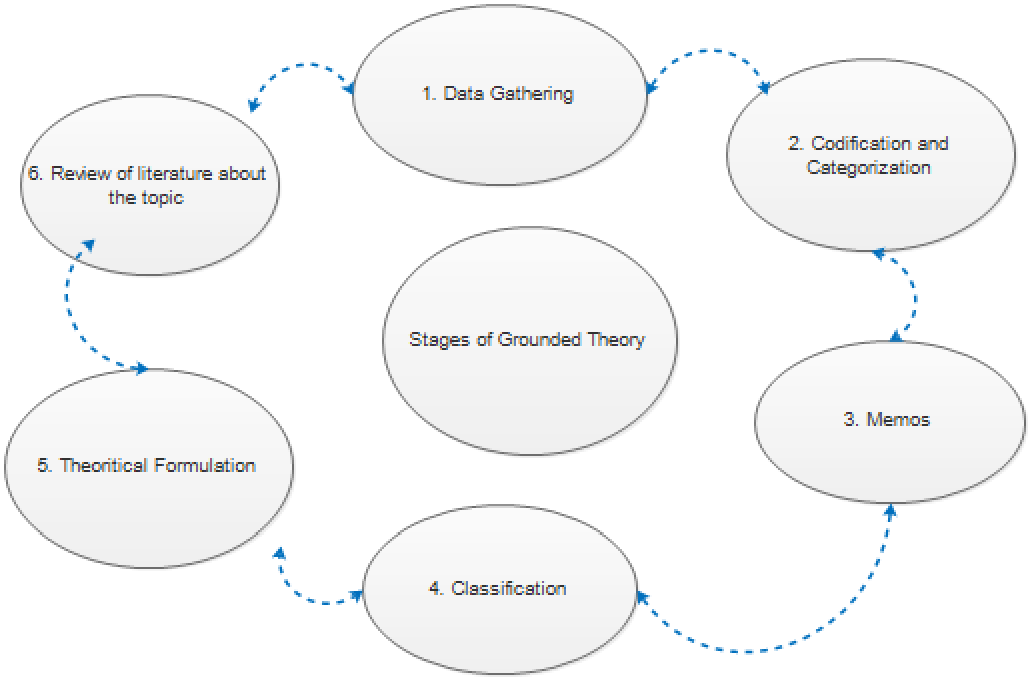

1. Data Gathering
2. Codification and Categorization
3. Memos
4. Classification
5. Theoritical Formulation
6. Review of literature about the topic
Stages of Grounded Theory

Figure 3.3: Grounded Theory (Cutcliffe, 2000)

3.5.2 Grounded theory phases

Figure 3.3 highlights the grounded theory through various phases. Data gathering as the first phase, this phase in an important part of the research, according to Yuan (2001:272) data collection is a vital aspect to minimize chance of unreliable data which maybe hinder research failure. Qualitative data gathering will be conducted through qualitative Interviews refer to Section 3.7.1.

The interviews are followed by coding and categorization phase, where data is categorized accordingly. The memos phase involves reflecting on the meaning of codes and how the codes should relate to each other in order for them to be classified according to various groups (Tracy, 2012:196). Memos serve as an intermediary between writing codes and writing a rough draft of the analysis. Memos play a vast role in understanding how the codes can affect each other and eventually the whole analysis.

The classification phase is where the researchers combines the related codes. Theoretical formulation forms part of interpretations and analysis by ensuring that the findings are a true reflection of the research. The final phase includes reviewing the literature to ensure that the questions are revised according to the literature, adding the necessary features that will be relevant to the study.

3.6 DATA COLLECTION METHODS

Data collection methods are used by the researchers to report on the research questions and support the overall literature of the study. This section elaborates on how the researchers can create a sound interview guide in order to conduct face-to-face qualitative interviews. This section elaborates on how participants were selected using a criteria. Lastly the section explains how data is analyzed and protected for ethical purposes.

3.6.1 Qualitative Interviews

A qualitative interview aims to understand mainly what the interviewee’s view is based on the study. Interviews seek to investigate and uncover meanings about a subject not only from the researchers but users. The users have to relate or most likely be influenced by the subject of the study. Qualitative interviews seek to understand the user’s experience on the subject. According to Kvale (2006:481) interviews allow individuals to voice their opinions and gives them freedom to explain their experiences in relation to a subject. Qualitative interviews will be used to gather the necessary data for the study. Grounded theory principle are applied in the process of qualitative interviews where after each cycle of interviews the researchers can reflect on the questions and possibly include more in order to collect data that has depth. Kvale (2006:484) emphasizes the power dynamics during qualitative research as follows:

- ***The interviewer rules the interview:*** The interviewer is responsible to clearly define the plan, implementation and outcomes the interviews. The interview must have a set ground rules and should be communicated to each interviewee.
- ***An Interview is a one-way dialogue:*** The interviewer must understand that they ask the questions and interviews answer the question. Roles must be clearly defined as to who is entitled to speak in which manner. The interviewer must drive the interview and not be lost in a dialogue, in order to reach the goal.
- ***The interview is an instrumental dialogue:*** An interview is referred to be instrumental when it is not about have a good long conversion with interviewee, but for interviewer to attain meaningful information that they need.
- ***Interview may be a manipulative dialogue:*** The interviewer may have a hidden agenda to some degree by asking particular questions in a vague manner.
- ***Counter control:*** It is crucial for the interviewer to address the fact that the interviewee should answer all the questions, cannot randomly withdraw during the interview with no reason and to being dishonest.
- ***Membership research****:* Some interviewers may attempt to reduce their dominance of the interview by allowing the interviewee to amend the interpretation. Therefore research interviewers may seek validation from the interviewee and will often ask the interviewee to take part in deciding if the interpretation represent the overall interview well.

According to DiCicco-Bloom and Crabtree (2006:315) an interview guide structure uses a certain technique. There are 3 types of techniques that can be used namely:

- ***Unstructured interviews:*** Interviews that are driven my conversation. The interviews are often associated with observation where by the interviewer notices some behavior while observing. The interviewer documents the behavior and randomly creates follow-up questions that are not in the interview guide.

- ***Semi-structured interviews:*** Interviews that are scheduled on time, appropriate venues and generally organized. The interview is prearranged on the set of questions and can also have questions evolving from conversation, mostly in terms of clarity or rephrasing of a question.
- ***Structured interviews:*** Are the most widely used technique for interviews. The questions are structured for certain groups and are very narrow and specific. The research question will be the priority when formulating the interview guide. The first set of questions is strongly associated to the research question of the study.

3.6.2 Creating an interview guide

An interview guide is necessary when the research interviewers need a structure and order of the questions that will be asked in an interview. The interview guide must include *what* type of questions will be asked and consider *how* relevant the questions are to the study (Tracy, 2012:143). It is crucial for the researchers to re-visit the primary, theoretical objectives as well as the empirical objectives; most importantly the literature of the study before drawing up an interview guide. Researchers must consider the main objective of the interview guide in terms of what the questions should reveal about the study. The researchers must decide whether the questions will to only gather facts, to test a hypothesis, to discover a new focus on the study or to explore personal feelings and opinions (Tracy, 2012:143).

Good interviews should support the overall study and be aligned with the objectives. Some of the characteristics of good interview questions are discussed by Tracy (2012:144) as:

1. *The age group*: Are the questions accommodative in terms of feelings of the interviewee, are they ethically aligned with the age group and are the questions not too sophisticated for the interviewee to understand.

2. *The jargon:* Are the questions clearly defined and simple enough to be understood. The questions should consider the language of the interviewee especially if interviewee is not familiar with field of the particular research.

3. *Neutral and straight-forward:* The questions must be as narrow as possible. The researchers should know the type answers they are looking for. The questions should not be too vague.

4. *Promote open-ended answers:* The researchers must try to avoid "yes" and "no" answers. The researchers should rather ask on how the interviewee was able to rename the button.

5. *Ask one question at a time:* The researchers must not ask two or more questions in one. Try by all means to ask about a certain feature at a time. The interviewer must also try to avoid questions such "What caused the program to exit and do you know what you clicked and could you understand the error before it crashed".

3.6.3 Face-to-face interviews

The study uses face-to-face interviews as part of data collection. A good interview guide does not determine overall success of an interview. Face-to-face interviews contribute a large factor towards a good interview. The researcher should consider the following factors when conducting face-to-face interviews (Tracy, 2012:160).

Location: The interviewer must ensure that the location has access and that it is not hectic nor loud place. Be considerate of the interviewee in terms of safety. The comfort of the interviewee will allow them to be free and expressive during the interview.

Climate: The interviewer should consider an environment that will allow them to record thus a quiet and calm place should be a priority. A quiet space is necessary for a good interview because there will not be a lot of distractions. The temperature also contributes to a good interview, for example, if it is outdoors in the sun in the summer season then both interviewer and interviewee are unlikely focus on the interview but the heat.

Privacy: There should be sufficient privacy during the interview away from any distraction, co-workers and family members or generally anyone who is not going to add value to the interview.

Time: The interview guide will roughly estimate the time it will take for the interview. It is vital that interviewer makes it clear to interviewee of what time the interview will start and how long it will take. The interview must consider the time it takes for him/her to set up for interview to give them time to review the interview guide and set all his audio devices for recording ready. The interviewer should be late for the interview.

Generally, when an interview begins, the interviewer should brief the interviewer on the actual topic, the purpose of the interview, how the interviewer is contributing to the interview and what is expected from the interviewee.

3.6.4 Selecting participants

The selection criterion for participants required the researchers to formulate a test that will contain a number of design instructions. The instruction on the test will test HCI principles and a student has to follow the instruction. The aim of the test is to narrow down the top half and bottom half in terms of score results. The criteria will allow a fair evaluation of students and a balance in terms of how the students view the visual studio C# IDE. However, considering the HCI principles discussed in chapter 2, the principles aim to ease the learning process for the students. The chosen students will move to the face-to-face interviews. The interview contains a total number of 10 participants from ITRW123 lecture.

3.6.5 Ethics and Data Protection

The study prioritize data collection ethics, end-users will have option to exercise their confidentiality preferences during and after research interview process.

3.6.6 Data analysis

Initial stages of data analysis include arranging the data by preparing all fieldnotes, the interview transcripts all supporting data. Considerations should be made when deciding which information is important and is of value to the study. Data analysis is a crucial part of the study and the researchers should also know how to store the data. according to Huberman and Miles (1994:300) analysis of qualitative research the study aims to understand the holistic view of the focus area by using data to describe its meaning. The analysis will be done through the use of graphical representation tools for clear interpretation. Data will be compared and categorized based on user performance as according to Huberman and Miles (1994:301) there is a need for weighing of evidence in data to validate scale of distortions. Qualitative data analysis will be done during the last cycle of the interpretive study. During this phase the study will incorporate the ground theory. Data will be contextualized and study will be using semi structured interview; observation methods and recordings will be taken after the interview process to be transcribed. Interview transcripts are used for coding and should be written correctly. Coding is a process of classifying or labeling data. Coding can be achieved in using different resources; paper and highlights manually, excel spreadsheet or a qualitative data analysis tool. Atlas.ti is a data analysis software that will be used for this study. Excel is a tool that will be used to assist in tracking data from interviews and to find patterns within the data.

3.7 CONCLUSION

A research methodology is defined as an approach to addressing the research. The study will evaluate the Visual Studio C# IDE using interpretive research to understand the impact of HCI principle application on a system and end-user experience. Grounded theory principles is the chosen research method. Data collection methods used are qualitative interviews and guidelines for successful interviews. A selected number of participants using a criterion will participate in the interviews and the next chapter elaborates on the research findings for insights and analytics.

Chapter 4: Empirical Study

4.1 Introduction

The main objective of this study is to evaluate the visual studio C# IDE of a programming language using the HCI principles, by understanding how HCI principles were applied on the IDE and how they can help improve learning C# IDE programming language for first year ITRW 124 students. To address this objective, an interpretative research methodology is used. Grounded theory as a research method is applied throughout the process of data collection to data analysis phase.

Qualitative interviews are conducted to gather data and ATLAS.ti software is used in order to code and analyse the data. Face-to-face interview rules are applied in the process of interviews and the power dynamics during interviews are applied. The selection criterion used for participant selection is discussed in Chapter 3.

This chapter will discuss the initial stages of data collection where a pilot interview was conducted and how it revealed insights about the literature. A discussion on the tool used for data analysis (ATLAS.ti) is elaborated on as well coding as concept that is mostly used in this chapter. Each principle has a certain codes that highlight various phases from the participant's reviews on the C# IDE. Included are comparisons highlighting the number of code occurrences per HCI principle and thus indicate how the participants relate to parts of the C# IDE.

4.2 Interview Guide Compilation

The first version of the interview guide was compiled based on HCI principles that were randomly chosen by the researchers namely, synthesizability, predictability, consistency, generalizability, familiarity. The interview guide consisted of 18 questions shown below.

Question 1.1. *What is your initial impression when you saw the C# IDE?*

Question 1.2*. In the first two encounters with C# IDE, were you able to identify the following controls:*

- *Task Bar*
- *Menu Bar*
- *Solution Explorer*
- *Toolbox*
- *Data source*
- *Server Explorer*
- *Properties*

Question 1.3. *During interaction, do you feel that C# IDE gives fair responses and informative guidance (Warning signs when overwriting)? Give a practical example.*

Question 1.4. *Based on your interactions, have you recognized any similarities in terms the C# IDE layout or design? Give an example.*

Question 1.5. *Considering the items you identified on the C# IDE, would you say they share the same functionality? Give an example.*

Question 1.6. *Have you used different version of Microsoft Visual Studio C#? If so, what's your view on the general layout of the versions? Give an example**.***

Question 1.7. *Would you like to see anything different on C# IDE?*

Table 4.4: Source guide for interview questions

Each of the question were intended to address each of the principles and throughout the transcription and analysis process the interview was not as focused and lacked direction.

HCI Principle addressed	Meaning of a principle in literature and source.	Interview guide question.
1 Synthesizability	The ability of the user to make changes without overwriting and the systems being able to warn the user to make changes in order for them reflect with their permission (Hinze-Hoare, 2007:11)	## During interaction, do you feel that C# IDE IDE gives fair responses and informative guidance (Warning signs when overwriting)? Give a practical example.
2 Predictability	This is the ability of the user to use their past knowledge in order to determine the effect of future actions (Dix *et al.*, 2004:262).	## In the first few encounters with C# IDE IDE, were you able to identify the following controls? Toolbox, solution explorer, properties etc.?
3 Familiarity	The extent to which a user's past knowledge and real world experiences. The familiarity of users can be measured by the link between their existing knowledge and knowledge required for them to relate to the system (Hinze-Hoare, 2007:10)	## Based on your interactions, have you recognized any similarities in terms the C# IDE IDE layout or design? Give an example.
4 Generalizability	Is focused on the user's ability to apply their previous knowledge to an unknown or similar interaction (Terblanche, 2014a:23)	1.5 Considering the items you identified on the C# IDE IDE, would you say they share the same functionality? Give an example.
5 Consistency	Is the resemblance in the behaviour from similar activities. A design interface with similar capabilities or functionality (Terblanche, 2014a:24).	1.6 Have you used different version of Microsoft Visual Studio C# IDE? If so, what's your view on the general layout of the versions? Give an example.

4.3 Pilot Interview

The pilot interview was conducted on relevant participant which was a student that is enrolled for the module ITRW 123 (C# IDE). It is fair to conduct interviews on people who are exposed to the program and these students need to be accredited for the ITRW 123 module in order to progress in their studies.

As first time researchers the pilot interview was a learning experience. The interviewee was taken through the study why he was a relevant candidate. The researchers shared the questions equally and elaborated as much as possible on each question. There were practical questions and the researchers had to observe the entire process.

One of the purposes of the pilot was to determine if there are potential flaws and/or restrictions on the literature. The purpose of the pilot was to reveal possible weaknesses in terms of the interview guide. A pilot interview was conducted by the researchers and revealed insights on the interpretation of the literature; it revealed that there are weaknesses in the interview guide and there had to be revision on the literature of the study, as well as the compilation of the interview guide. From the analysis of the pilot interview's transcription, the researchers noticed that an included HCI principle, namely, consistency, although discussed in the literature, are not part of enhancing the learnability. According to (Dix *et al.*, 2004:262) there are six principles that form part of learnability of by applying them any particular interface of a system/program through HCI. The pilot thus allowed the researcher to revise the study and make the necessary refinements from literature. The refinement of the literature determined the type of questions included in the interview guide that addresses the study legitimately

4.4 Interview Guide Adjustments

The interview guide was restructured by removing the unambiguous questions in the guide and rather including questions that address each principle in depth. The questions that were removed are:

Question 1.3. *Which controls were you able to use?*

Question 1.4. *Can you comment on clarity (overall visibility) of IDE?*

Question 1.5 *Can you tell us how you resized the form? Why did you choose that specific method?*

Question 1.6. *Were you comfortable with the initial font size of the button/label?*

Question 1.7. *Based on recoverability functions c# has, you were asked you to recover server explorer. Were you able to?*

Question 1.8. *If you were asked to close/cancel the toolbox, would you able to recover it?*

Question 1.9.*How do you deal with errors? What are your main concerns with C# IDE?*

Question 1.10. *How did you overcome the problems you experience in C#?*

Question 1.11: *Based on your past experience, please comment on the familiarity?*

Question 1.12. *Would you like to see anything different on c# IDE?*

Question 1.13. *For example, if you are writing a test, and you get stuck, do you feel that c# gives fair responses and guidance?*

Question 1.14: *How do you get help with errors during tests?*

Question 1.15. *C# has a HELP tab in the menu bar, have you ever used it during errors?*

Question 1.16. *In your opinion is the IDE user friendly?*

An infinity approach in grounded theory was applied after the pilot interview and from that, it allowed the reserachers to use the correct set of HCI principles and ach of the questions in the revised interview guide addressed each principle. Each question was as specific as possible in order to validate weather a particular HCI principle is applied in the C# IDE. Therefore a new interview guide was that compromises of 6 questions that each addressed a principle and one questions that ask the participant on how they feel that the C# IDE can be improved, the questions are:

Question 1.1. *What is your initial impression when you saw the C# IDE?*

Question 1.2*. In the first two encounters with C# IDE, were you able to identify the following controls:*

- *Task Bar*
- *Menu Bar*
- *Solution Explorer*
- *Toolbox*
- *Data source*
- *Server Explorer*
- *Properties*

Question 1.3. *During interaction, do you feel that C# IDE gives fair responses and informative guidance (Warning signs when overwriting)? Give a practical example.*

Question 1.4. *Based on your interactions, have you recognized any similarities in terms the C# IDE layout or design? Give an example.*

Question 1.5. *Considering the items you identified on the C# IDE, would you say they share the same functionality? Give an example.*

Question 1.6. *Have you used different version of Microsoft Visual Studio C#? If so, what's your view on the general layout of the versions? Give an example**.***

Question 1.7. *Would you like to see anything different on C# IDE?*

From the analysis of the pilot interview's transcription, the researchers noticed that an included HCI principle, namely, consistency, although discussed in the literature, are not part of enhancing the learnability. According to (Dix, 2004) there are six principles that form part of learnability of by applying them any particular interface of a system/program through HCI. The pilot thus allowed the researcher to revise the study and make the necessary refinements from literature. The refinement of the literature determined the type of questions included in the interview guide that addresses the study legitimately

4.5 Interviews

The interview process went through a number of milestones that will be discussed in this section. Interviews were required to be transcribed, coded and analysed. A discussion of the software tool called ATLAS.ti as quantitative analysis tool to assist in the data analysis process. Coding and analysis is also elaborated on after the interviews had been transcribed. Lastly themes are gathered and a code book as a concept is explained and practically compiled in a table.

4.5.1 ATLAS.ti

The researchers make use of ATLAS.ti a qualitative analysis tool. ATLAS.ti has an import function where transcripts can be opened in the software. It is vital that the interviews are transcribed before importing them into the software. The software is structured in a way that the researcher can create a code and attach it to a quotation per question. It further allows a researcher to group the codes into possible themes. The researcher can store the transcriptions and continue amendment as much as possible.

4.5.2 Coding and analysis

The researchers used codes in ATLAS.ti, which is a process of integrate the collected data to generate themes, classifications as well ideas. The codes can be stored and retrieved on an ATLAS.ti project for any amendments as mentioned on Section 4.3. Coding the collected data ensures easy navigation to facilitate making comparisons and to see any trends in the data. Codes are based on phrases, keywords, ideas, concepts, topics and quotations that are found in the data. A code is usually supported by a phrase or quote. Codes are mostly given meaningful terms that give a sign of the concept that derives a theme. The number of codes that appear may highlight issues or insights regarding the study.

4.5.3 The Code Book

A code book is created for the researchers to understand in which context the codes are used in and what each code means. Every individual reading the empirical study without reading the research literature has to understand what a code represents.

Table 4.5.3: Code list and definitions.

CODES	CODE DEFINITIONS
Adaptable	1. Ability to adjust to new conditions: For this research specifications, this code monitors the ability of a participant to learn and progress thorough engraining with the MVS C# IDE IDE after the first two encounters.
Case sensitive help	2. Text or inputted text that is sensitive to program syntax: this code reflect the participant’s usage experience on help function while interaction with the C# IDEIDE.
Command feedback	3. Give an authoritative or peremptory order: For this research specification: this code reflect the participant’s usage experience on level of usage guidance while interacting with C# IDE IDE.
Comparable	4. Having features in common with something else to permit: For this research specification: this code reflect the participant’s cognitive ability related the current C# IDE IDE to other programing tools. Much emphasis is to measure the participant’s learning and reflective behaviour.
Consistency	5.This principle concentrates on the comparability in information and yield conduct rising up out of equivalent circumstances or task objectives (Hinze-Hoare, 2007:10)
Design conventions	6. Ability to change functionality of an object: this code reflect the participant’s ability to use different methods to perform similar task operations.
Easy to learn	7. For this research specifications, this code monitors the ability of a participant's ability to learn and progress through C# IDEIDE interaction.
Efficient to use	8. For this research specifications, this code monitors the ability of a participant's to use the C# IDE IDE at the most continent way considering time and accessibility.
Error detection	9. For this research specifications, this code monitors the participant's usage experience based on C# IDEIDE feedback functionality.
Memorable	15. For this research specifications, this code monitors the participant's ability to recall information for their previous knowledge based on C# IDEIDE functions.
Message warning	16. For this research specifications, this code monitors the participant's usage experience on the effectives of error communication.
Past Knowledge	17. For this research specifications, this code monitors the participant's ability to use previously learn skill to perform C# IDE functions.
Practice	18. For this research specifications, this code monitors the participant's usage experience based on C# IDEIDE feedback on providing consistent user guideline, suggestion and recommendations.
Predictability	19. Predictably can be defined as the reference to the level of degree at which correct prediction can be made regarding the system. According to (Dix at all,2014)
recognizable	20. For this research specifications, this code monitors the participant's usage experience and ability to perform task based on previous experience.
Recoverability	21. For this research specifications, this code monitors the participant's ability to manage and recover C# IDEIDE components.
Same Functionality	22. For this research specifications, this code monitors the participant's reflective capabilities

CODES	CODE DEFINITIONS
	across deferent development platforms.
Satisfaction	23. For this research specifications, this code monitors the participant's general impression on C# IDEI experience.
Similar capabilities	24. For this research specifications, this code monitors the participant’s ability to relate different design functions to C# IDEIDE.
Familiarity	10.Familiarity has to do with the first impression that the user encounters when they interact with a system and how the user can be able to initiate interaction (Dix et al., 2004:262)
Generalizability	11. Generalizability is defined as the extension of research results and conclusions from a specific study. According to (Dix at all, 2004:264)
Helpful suggestions	12. For this research specifications, this code monitors the participant's usage experience based on C# IDEIDE feedback based on tool's ability for give interactive guide on how to optimize system usage.
Interactive	13. For this research specifications, this code monitors the participant's usage experience based on C# IDEIDE feedback on providing consistent user guideline, suggestion and recommendations.
Learnable	14. For this research specifications, this code monitors the ability of a participant's to understand the general MVS C# IDE Framework.
Similar Window	25. For this research specifications, this code monitors the participant's ability to relate different design patterns to chide.
Subjectively pleasing	26. For this research specifications, this code monitors the participant's general impression on C# IDEIDE experience taking into consideration exceptions.
Synthesizability	27. This is the ability of the interface to allow the user to construct a predictive mental model of how it operates.
Understandable	28. For this research specifications, this code monitors the participant's ability to navigate through C# IDEIDE and incorporate different functions.
Useable	29. For this research specifications, this code monitors the participant's usage and ability to perform expected tasks on C# IDEIDE.
User Friendly	30. For this research specifications, this code monitors the participant's usage experience and ability to familiarize with the C# IDEIDE components.
Visibility	30. This is an important part of the psychological principle that it is easier to recognize images than to have to recall them.

4.5.4 Themes

A theme is also known as a ‘code family’, in this context lead back to the HCI principles that are the core of the literature of the study. The process of identifying a theme is where all codes are collected and it is discovered that each code can substantiates a certain principle. Therefore all codes that are similar or support a certain can be group together to justify how the HCI principle is applied on the C# IDE.

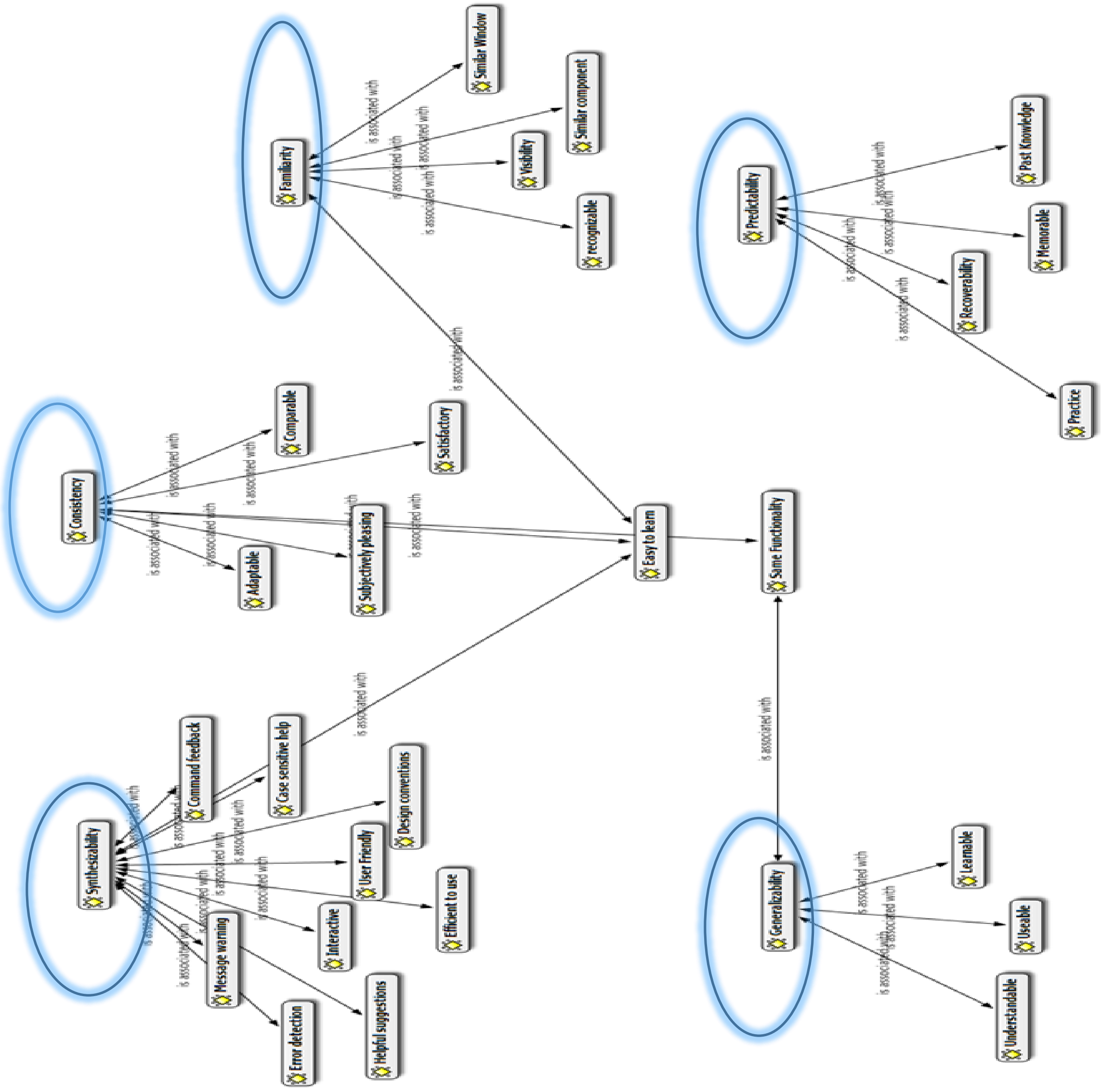


Figure 4.5.4: Analysis themes

4.6 Discussion of the analysis findings

This section consists of comparisons amongst each principle as well as the codes associate of each of them. Comparisons reveal the number of occurrences of each code per principle/theme. The aim of the comparison is to reveal the students' perspective on the Microsoft Visual Studio C# IDE. Therefore using perspective the researchers could collect a set of codes that can be supported and capture the code occurrence of each HCI principle.

4.6.1. Synthesizability

The ability for the user to build on how a system operates. In other words it is the process of how the user can understand what can happen after a certain operation or certain changes.

Table 4.6.1: Results for code occurrence for synthesizability

CODE	OCCURRENCE	EXAMPLE PHRASE
Error detection	8	"Gives error messages and there is also a help library that I learnt after"
Efficient to use	5	"having the functionality of dragging the controls and dropping them was easier than coding the actual object"
Design conventions	1	"You can change to Encryption view or Design view by clicking Code or Designer on the View menu"
Case sensitive help	1	"If I have misspelled a word highlighting of a red underline under the word and when you try to run the code it will simply tell you what is wrong with the error"
Command feedback	8	"When I code, it shows me errors and points to them directly"
Message warning	6	"It guides me in terms of closing the program, it will warn me via a dialogue message if I want to save"
Helpful suggestion	3	"I mistakenly deleted my tool box, it warned me, it helps a lot in terms of achieving the objective"
Easy to learn	4	"Gives error messages and there is also a help library that I learnt after"

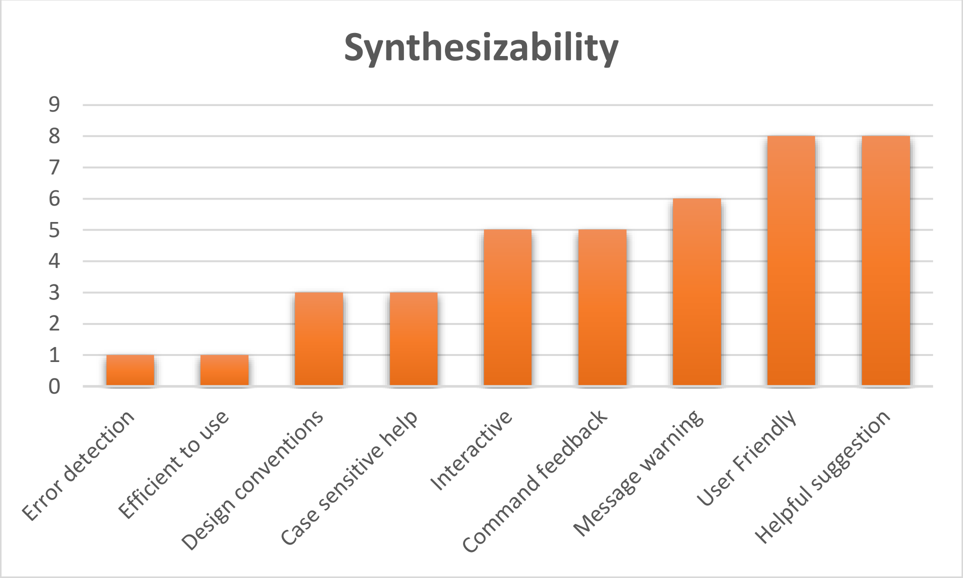


Figure 4.6.1: Code occurrence for synthesizability principle.

4. 6.2 Predictability

Predictably can be defined as the reference to the level of degree at which correct prediction can be made regarding the system. According to (Dix *et al.*, 2004:262) predictability determines the operational visibility of the system.

Table 4.6.2. Results for code occurrence for predictability

Code	Occurrence	Example phrase
Past knowledge	6	"One thing I have noticed, When the code is incorrect, it has a pop up functionality that shows what is missing"
Memorable	1	"I have seen a similar window"
Recoverability	4	"sometimes if there are hidden objects, I search and retrieve them in View tab"
Practice	3	"I started using it, and it is helpful and easy to use as I practised"

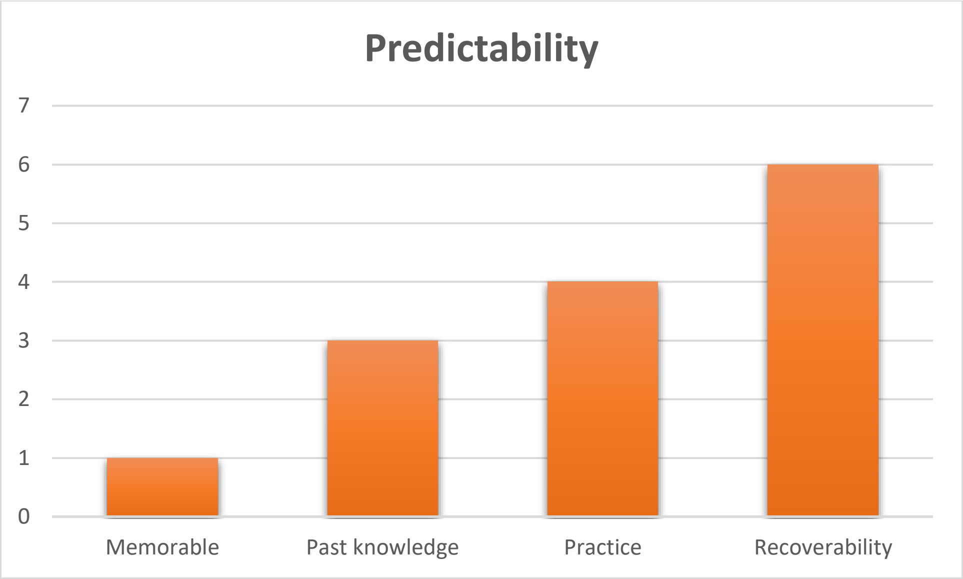


Figure 4.6.2: Code occurrence for predictability principle

4. 6.3 Familiarity

Familiarity has to do with the first impression that the user encounters when they interact with a system and how the user can be able to initiate interaction (Dix *et al.*, 2004:262).

Table 4.6.3: Results for code occurrence for familiarity

Code	Occurrence	Example phrase
Similar component	8	"Yes, In Java code blocks, File menu and connectivity objects are the same"
Similar Window	1	"In the file tab in MS word, I know I can find "new project", "open project", "save as" which also applies in C# IDE program. Also the VIEW tab u can retrieve some controls as well"

Visibility	10	*“ observation: The task bar, menu bar, solution explorer, toolbox, properties”*
Recognizable	8	“I have seen a similar window in Microsoft word, excel in terms of the menu bar”
Easy to learn	4	“I have seen a similar window in Microsoft word, excel in terms of the menu bar”

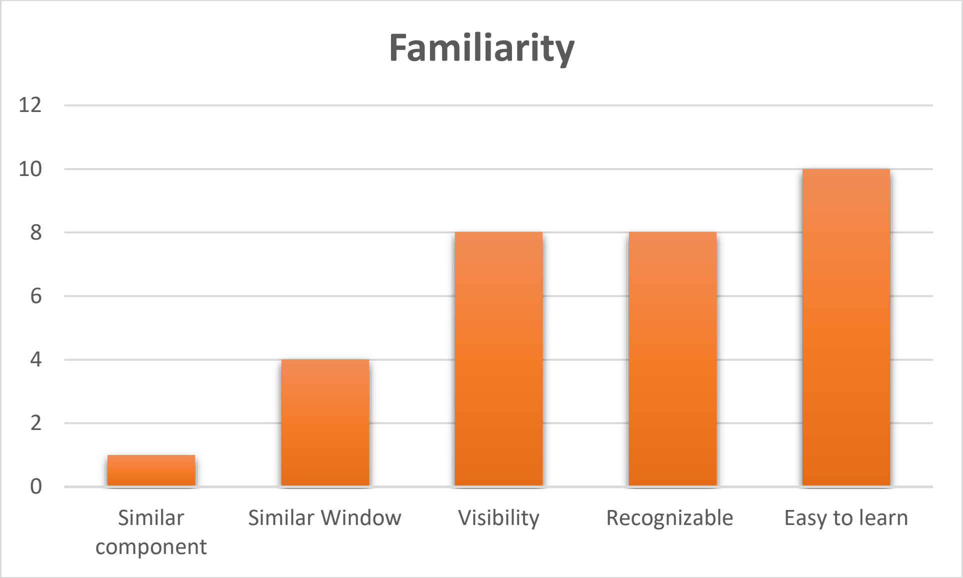


Figure 4.6.3: Code occurrence for familiarity principle

4. 6.4 Generalizability

Generalizability can be seen as the extension of research results and conclusions from a specific study. According to (*Dix at al*, 2004:264) generalizability is defined as the process of extending specific interaction knowledge to a new but similar situation.

Table 4.6.4: Results for code occurrence for generalizability

Code	Occurrence	Example phrase
Similar component	8	"Yes, In Java code blocks, File menu and connectivity objects are the same"
Similar Window	1	"In the file tab in MS word, I know I can find "new project", "open project", "save as" which also applies in C# IDE program. Also the VIEW tab u can retrieve some controls as well"
Visibility	10	*" observation: The task bar, menu bar, solution explorer, toolbox, properties"*
Recognizable	8	"I have seen a similar window in Microsoft word, excel in terms of the menu bar"

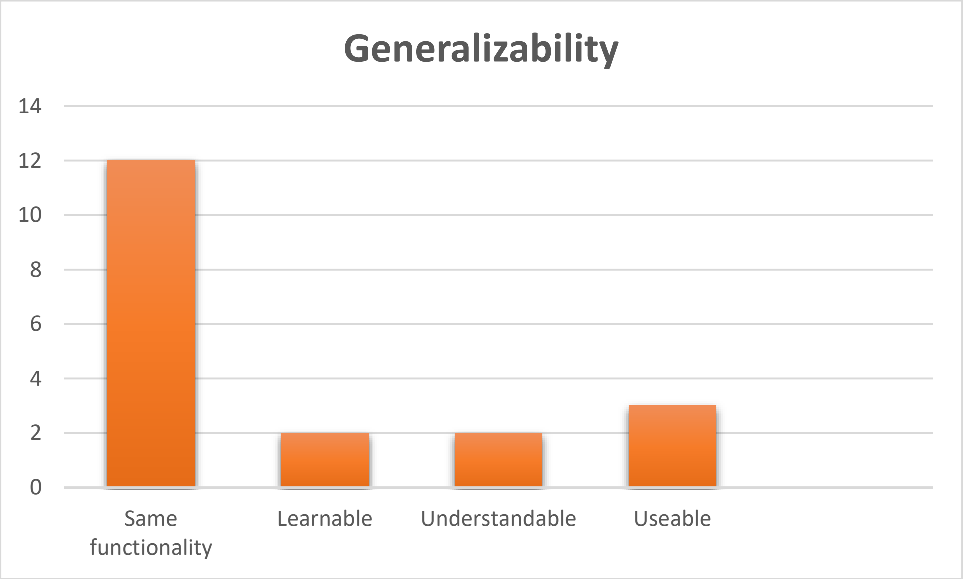


Figure 4.6.4: Code occurrence for generalizability principle

4. 6.5 Consistency

The consistency principle concentrates on the comparability in information and yield conduct rising up out of equivalent circumstances or task objectives (Hinze-Hoare, 2007:10).

Table 4.6.5: Results for code occurrence for consistency

Code	Occurence	Example phrase
Easy to learn	4	“I felt it was actually easier having the functionality of dragging the controls and dropping”
Comparable	12	“The netbeans application and Microsoft are some the examples, they has some similar properties such as the menu bar. But most applications are built like this, mostly have a similar menu bar.”
Satisfactory	3	“Nothing really, I happy with the current outlook.”
Same functionality	12	Yes, the menu bar items do share the same functionality, for example.
Adaptable	4	“Yes, in the labs we used 2010 and has upgraded 2015. There not much adjustments from the two versions”
Subjectively pleasing	4	or a beginner I would say it is good and is very informative, in terms of clarity, simplicity and informative dialogues

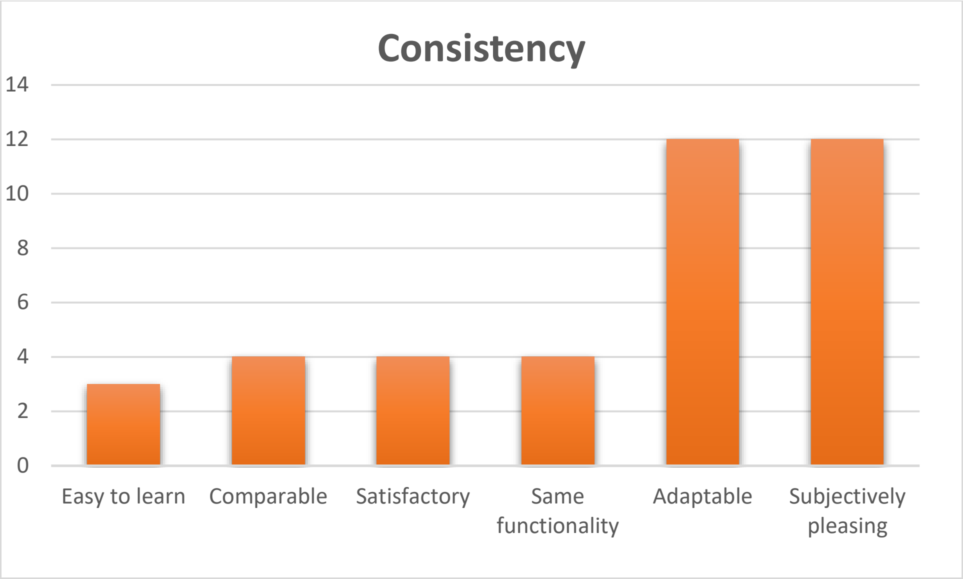


Figure 4.6.5: Code occurrence for consistency principle

4.7.1 Improvement of the C# IDE

Most of the students did feel that at some point the IDE came across as confusing as first as well as the controls. Some students further add on to say that it may be complex when there is no assistance or guidance.

4.8 Interpretation

This section discusses the findings from the participants and how they personally experience the C# IDE. The codes are ranked per HCI principle.

4.7.1 Synthesizability

The study used predictability as an HCI principle to measure learnability and based on testing code matrixes implemented by the researchers and can be discussed as follows:

Participants do demonstrate different level of skills and after the first few interaction with the Microsoft visual studio C# IDE and for the purpose of this study it was necessary to classify number of the participants per level of experience against ranking score during the interview demonstrations for each defined principle. In Section 4.6 there are 9 identified codes that influence synthesizability as an HCI principle.

Out of the 10 participants easy learning and helpful suggestions scored 8 ranking, user-friendly scored 6 ranking ,message warning scored 5 ranking, feedback scored 4 ranking , case sensitive feedback and design conventions scored 3 ranking and both Efficient to use & Error detection scored 1 ranking . From an observation perspective, students engaged differently based on their level of programming experience. Some skills and habits can be improved to enhance learning and overall the C# IDE is considered easy to learn by students.

4.7.2 Predictability

The study used predictability as an HCI principle to measure learnability and based on testing code matrixes implemented by the researchers and can be discussed as follows:

Participants do demonstrate different level of skills and after the first few interaction with the Microsoft visual studio C# IDE and for the purpose of this study it was necessary to classify number of the participants per level of experience against ranking score during the interview demonstrations for each defined principle. In Section 4.6 there are 4 identified codes that influence predictability as an HCI principle.

Out of the 10 participants 6 scored high in recoverability, 5 scored high on practice, 3 score high on past knowledge and only 1 scored high on memorable code. From observation perspective, student engage differently based on level of programming experience and some skills and habits can be improved to enhance learning. Overall recoverability mostly influenced the predictability principle

4.7.3 Familiarity

The study used predictability as an HCI principle to measure learnability and based on testing code matrixes implemented by the researchers and can be discussed as follows:

Participants do demonstrate different level of skills and after the first few interaction with the Microsoft visual studio C# IDE and for the purpose of this study it was necessary to classify number of the participants per level of experience against ranking score during the interview demonstrations for each defined principle. In Section 4.6 there are 4 identified codes that influence predictability as an HCI principle.

Out of the 10 participants Easy to learn scored 10, Recognizable and Visibility scored 8, similar window scored 4 and similar components scored 1. From observation perspective, student engages differently based on level of programming experience and some skills and habits can be improved to enhance learning and the above figure depicts that C# IDE is considered easy to learn by most participants.

4.7.4 Generalizability

The study used predictability as an HCI principle to measure learnability and based on testing code matrixes implemented by the researchers and can be discussed as follows:

Participants do demonstrate different level of skills and after the first few interaction with the Microsoft visual studio C# IDE and for the purpose of this study it was necessary to classify number of the participants per level of experience against ranking score during the interview demonstrations for each defined principle. In Section 4.6 there are 4 identified codes that influence predictability as an HCI principle.

Out of the 10 participants same functionality scored 12 ranking, learnable and Understandable scored 2 ranking and Usable scored 3. From observation perspective, student engages differently based on level of programming experience and some skills and habits can be improved to enhance learning and the above figure depicts that C# IDE is considered to have same functionality by most participants.

4.7.5 Consistency

The study used predictability as an HCI principle to measure learnability and based on testing code matrixes implemented by the researchers and can be discussed as follows:

Participants do demonstrate different level of skills and after the first few interaction with the Microsoft visual studio C# IDE and for the purpose of this study it was necessary to classify number of the participants per level of experience against ranking score during the interview demonstrations for each defined principle. In Section 4.6 there are 4 identified codes that influence predictability as an HCI principle. Out of the 10 participants subjectively pleasing scored 12 ranking, Adaptable scored 12 ranking, and same functionality – Satisfactory and comparable scored 4 ranking lastly Easy to learn scored 3 ranking. From observation perspective, student engages differently based on level of programming experience and some skills and habits can be improved to enhance learning and the above figure depicts that C# IDE is considered subjectively pleasing by most participants.

4.8 Conclusion

Based on the participants view it is clear that HCI principles are applied on the C# IDE and all the codes can safely validate each HCI principle. Most students can enhance their learning through these applied HCI principle and therefore can participate better knowing that the principle do indeed adhere on C# IDE.

The chapter elaborates on the interview guide compilation as well as interview guide adjustments. The adjustments were mostly influenced by the pilot interview along with application of grounded theory. The code book explains each code and what it represents in the context of this study. Themes are constructed and grouped in order to support each HCI principle. The codes are also ranked to see a particular behaviour in the principle. The codes further assist in the interpretation on how students feel that HCI principle are applied.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Primary objective

The primary objective of the research study was to understand the factors that makes it difficult for first year students to understand and be able to use C#IDE, the study focused specifically on Human Computer Interaction principles that affect learnability in an attempt to enhance the student’s learning experience by creating guidelines to support the use C# IDE as part of research recommendations.

This study took into consideration the student’s programming background and the degree of performance on average degree modules in order to design a fair assessment matrix and methods amongst the participants.

5.2 Theoretical Objectives

The theoretical objectives of the study was to gain a better understanding on factors which influence the software design, Graphical interface design and the university ideology on selecting a specific software packages for its Information technology prescribed modules.

The second factor was to understand the challenges that student are encountering which alternatively lead to poor performance and module cancellation at first year level.

The third and last factor was to understand the measures that can be applicable to assist student understand the current prescribed programming tool from the lecture and the university teaching perspective which alternatively will help the student to perform better on module.

5.3 Coding themes

The study used the five principles affecting learnability as core themes to map out code that were formulated during data collection for analysis purpose. These codes were formulated using both quantitative and opened ended questioning method during the interview process and they were much influence by the participant’s degree of answering.

After gathering the results, the study reflected a hierarchical systematic mapping of facts build on coding assessment which reflect that some of the principles are realized as the the most influential and some being realized at the least influential principles.

5.3.1 Adaptable

The participant’s sentiment score is 4:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE as an adaptable programing environment after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE is less adaptive.

5.3.2 Case sensitive help

The participant’s sentiment score is 1:12, this results tell us that at least one participant recognize Microsoft Visual studio C# IDE as having a clear case sensitive help support after first two C# assessment encounters and the study based on this set of participants, C# IDE does not offer a clear case sensitive support to enhance user’s experience.

5.3.3 Command feedback

The participant’s sentiment score is 8:12, this results tell us that at least a larger number of participants recognize Microsoft Visual studio C# IDE as having a clear command feedback support after first two C# assessment encounters and the study based on this set{} of participants, C# IDE does offer a fairly clear command feedback support to enhance user’s experience .

5.3.4 Continuity of task sequence

The participant’s sentiment score is 12:12, this results tell us that at least all number of participants recognize Microsoft Visual studio C# IDE as having a comparable features after first two C# assessment encounters and the study based on this set{} of participants, C# IDE is comparable and this can effluence the manner in which users experience the program.

5.3.5 Continuity of task sequence

The participant’s sentiment score is 0:12, this results tell us that none of the participants recognize Microsoft Visual studio C# IDE as having a series of task continuity on its environment after their first two C# assessment encounters and the study based on this set {} of participants, C# IDE is does not have a continuity of task sequence to support user experience and usability.

5.3.6 Design conversion

The participant’s sentiment score is 1:12, this results tell us that at least one participant recognize Microsoft Visual studio C# IDE as having a design conversion functionality after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE is less design convergence.

5.3.7 East to learn

The participant’s sentiment score is 4:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE as a learnable programing environment after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE is not easy to be learnt.

5.3.8 Efficient to use

The participant’s sentiment score is 5:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE being an efficient tool to work on after their first two C# assessment encounters and the study based on this set of participants, C# IDE have les efficiency.

5.3.9 Error detection

The participant’s sentiment score is 8:12, this results tell us that at least a larger number of participants recognize Microsoft Visual studio C# IDE as having a clear error diction support after first two C# assessment encounters and the study based on this set{} of participants, C# IDE does offer a fairly error feedback support to enhance user’s experience .

5.3.10 Helpful suggestions

The participant’s sentiment score is 3:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE as having clear helpful suggestion function after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE is offer less assistance to new developers .

5.3.11 Interactive

The participant’s sentiment score is 5:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE as being less interactive tool after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE is less interactive.

5.3.12 Learnable

The participant’s sentiment score is 2:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE as a learnable programing environment after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE is not easy to be learnt.

5.3.13 Memorable

The participant’s sentiment score is 1:12, this results tell us that at least one participant recognize Microsoft Visual studio C# IDE as a memorable tool to work on after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE has a high level of complexity .

5.3.14 Message warning

The participant’s sentiment score is 6:12, this results tell us that at least a larger number of participants recognize Microsoft Visual studio C# IDE as having a clear warning messaging support after first two C# assessment encounters and the study based on this set{} of participants, C# IDE does offer a fairly warning support to enhance user’s experience .

5.3.15 Practice and past Knowledge

The participant’s sentiment score is 5:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE as being easy to learn through practice after their first two C# assessment encounters and the study based on this set of participants, C# IDE is practical and process orientated.

5.3.16 same function and similar window

The participant’s sentiment score is 12:12, this results tell us that all few participants are familiar with most programming tool and can identify similarities on C# IDE during their first two C# assessment encounters and the study based on this set of participants, C# IDE comparable .

5.3.17 subjectively pleasing

The participant’s sentiment score is 4:12, this results tell us that at least few number of participants were okay with the look and feel of the C# IDE their first two C# assessment encounters and the study based on this set of participants, C# IDE Is not perfect but it can be manageable.

5.3.18 Understandable

The participant’s sentiment score is 2:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE as an understandable programing environment after their first two C# assessment encounters and the study based on this set of participants, C# IDE is not easy to be understood .

5.3.19 User friendly and usable

The participant’s sentiment score is 3:12, this results tell us that at least few number of participants recognize Microsoft Visual studio C# IDE as being less user friendly tool after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE is not user friendly .

5.3.20 Visibility

The participant’s sentiment score is 10:12, this results tell us that at least a larger number of participants recognize Microsoft Visual studio C# IDE as a clear and visible interactive tool after their first two C# assessment encounters and the study based on this set{} of participants, C# IDE is clean and enhance better user experience .

5.2. Factors leading to poor performance

50% of the participants felt that the Microsoft Visual Studio C# IDE is can be complex due to the fact that they do not know the various controls and their functionalities. It thus outs a strain on them and could potentially harm the outcome of the module. The students do feel that it can take longer to understand the various behaviours of within the C# IDE without any assistance. Continuous practice is thus an obvious solution for the students

5.5 Recommendations

5.5.1 Interactive User manuals

This can serve as smaller yet more precise guide in terms of activities and what the outcomes are. It could be a version of the study guide that has list of activities can focus on the GUI at first then continue on to the next facets of the programming language.

5.5.2 Built-in Tutorials

Internal tutorials on the C# IDE can be integrated as most students felt that it can further speed up the learning process and enhance their learning process. Instead on depending on external assistance it can a good element to have included in the IDE. This is where students can be navigated through the IDE step by step from creating a new project to creating classes and so on. It can have a functionality that could also have pop-ups when each control is highlighted in order to give a brief explanation of the functionality of a certain control and how and where it is supposed to be used.

5.5.3 Supplementary Introduction Courses

The introduction course can be a seen as a prerequisite course along with the module. For example, IT component for first years is a type of course that each student should complete in order to have completed a certain module. Same applies to first IT students, where they can have an introduction to c# course that has to be completed throughout the entire year for them to successfully complete ITRW 123 module.

5.6 Overview on the study

The study proved to a successful one in establishing the application of HCI principles as well as promoting HCI overall. The problem statement as an opinion can be argued to say that ITRW 123 students possibly have other influences that impact their performance in the module negatively. Factors such as lack of external assistance and guidance from the module's perspective as a whole.

Furthermore the students do have a strong opinion in terms of an internal tutorials on the actual program that can ease them into understand the concepts, the controls and well as their particular functionalities.

Lastly HCI can enhance students learning process and should be promoted in all aspects of the module.

Appendix

Initial Interview Guide

Section 1: Visual Studio IDE Design

Question 1.1. What is your initial impression when you saw the IDE?

Question 1.2. Were you able to find and use prescribed controls?

Question 1.3. Which controls were you able to use?

Question 1.4. Which controls were you not able to use?

Section2: Windows Form Interface Usability

Question 2.1. How did you find working on windows form?

Question 2.2. How did you resize the form according to the specified instruction?

Question 2.3. Was the text clear enough on windows form to read?

Question 2.4. Were you able to recover hidden or closed controls on windows form?

Question 2.5. Was the usage of components offer flexibility based on your usage preference?*

Question 2.6. Were there any components difficult to locate?

Question 2.7. Did you find any similarities amongst components that might have caused confusion?

Question 2.8. Would you prefer to see any changes in the Windows form?

Section 3: User Support

Question 3.1. Did you get guidance on how to use difficult components?

Question 3.2. Were you satisfied with the standard of guidance provided?

Question 3.3. What did you think about the overall task?

Revised Interview Guide

1.1. What is your initial impression when you saw the C# IDE?

1.2 In the first two encounters with C# IDE, were you able to identify the following controls:

- Task Bar
- Menu Bar
- Solution Explorer
- Toolbox
- Data source
- Server Explorer
- Properties

1.3 During interaction, do you feel that C# IDE gives fair responses and informative guidance (Warning signs when overwriting)? Give a practical example.

Observation (Exit project).

1.4 Based on your interactions, have you recognized any similarities in terms the C# IDE layout or design? Give an example.

1.5 Considering the items you identified on the C# IDE, would you say they share the same functionality? Give an example.

1.6 Have you used different version of Microsoft Visual Studio C#? If so, what’s your view on the general layout of the versions? Give an example.

1.7 Would you like to see anything different on C# IDE?

Participant 2

1.1. What is your initial impression when you first saw the C# IDE?

I felt it was actually easier having the functionality of dragging the controls and dropping them was easier than coding the actual object. But I hardly understood the meaning of the all the components in the toolbox therefore I had to continuously refer back to the textbook.

1.3 In the first two encounters with C# IDE, were you able to identify the following controls:

- ***Task Bar***
- ***Menu Bar***
- ***Solution Explorer***
- ***Toolbox***
- ***Data source***
- ***Server Explorer***
- ***Properties***

Observation: Participant 7 has not used a task bar, but could identify the toolbox, menu bar and properties, sometimes if their hidden I search and retrieve them in View tab.

1.3 During interaction, do you feel that C# IDE gives fair responses and informative guidance (Warning signs when overwriting)? Give a practical example.

Yes, sometimes when I exit it warns me. Helps me with errors, when I forget a method, it gives me options to choose correct words that can execute my method.

1.4 Based on your interactions, have you recognized any similarities in terms the C# IDE layout or design? Give an example.

The dialogues/message boxes are common in most programs, like when I exit on Microsoft word. I have seen a similar menu bar on Microsoft word and BlueJ.

1.8 Considering the items you identified on the C# IDE, would you say they share the same functionality? Give an example.

The menu bar works the same, File tab, there is an open a project in both c# and the identified programs.

1.9 Have you used different version of Microsoft Visual Studio C#? If so, what's your view on the general layout of the versions? Give an example.

Yes, I have used 2013 and 2015. 2013 shortcut functionality to create a project is not there like the 2015. 2013 is slightly slow, none the less it performs the same as other versions.

1.7 Would you like to see anything different on C# IDE?

I feel there should be something installed on the software installed to help explain what each object does. Also, when I drag maybe it give a pop up just to briefly explain how it works

NWU Ethics Checklist

Note:

In the Faculty Economic Sciences and Information Technology of the VTC this checklist is completed and approved with proposals by the Faculty Research Sub-Committee.

	Please answer each question by ticking the appropriate box[1]:	**YES**	**NO**
1.	Does the study involve participants who are particularly vulnerable[2] or unable to give informed consent? (e.g. children, people with learning or other mental of physical disabilities, people who are incarcerated, unemployed or otherwise compromised in responding to your questions)		✓
2.	Are you planning on making use of NWU students or direct and secondary/contracted staff members in this research?	✓	
3.	Will the study require the co-operation of a gatekeeper for initial access to the groups or individuals to be recruited? (e.g. students at school, members of self-help groups, residents of a nursing home, the Minister of Education, a tribal chief or village elder)	✓	
4.	Will it be necessary for participants to take part in the study without their knowledge and consent at the time?		✓

[1] Adapted from Economic and Social Research Council (2005). Research Ethics Framework (REF). www.esrcsocietytoday.ac.uk

[2] ***Vulnerable groups*** raise special issues of informed consent and potential risk. "Vulnerable" participants are not clearly described, but have been noted to include "…children, prisoners, pregnant women, mentally disabled persons, economically or educationally disadvantaged persons" (Common Federal Policy, 1991). Weijer and Emanuel (2000) consider participants to be vulnerable if they are not in a position to provide informed consent, due to their position (such as being in prison), or not possessing adequate intellectual faculty (such as children or the mentally ill)."Children" here are defined as participants younger than 18 years of age.

	(e.g. covert observation of people)		
5.	Will the study involve discussion of or questions about a sensitive topic? (e.g. sexual activity, drug use, crime, harassment, violence)		✓
6.	Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants or will the study involves invasive, intrusive or potentially harmful procedures of any kind or any physical, psychological or socio-economic intervention?		✓
7.	Will blood or tissue samples be obtained from participants?		✓
8.	Could the study induce physical, psychological or social stress or anxiety or cause harm or negative consequences beyond the risks[3] encountered in normal life?		✓
9.	Will the study require the identification of individuals for follow-up evaluation?		✓
10.	Will financial inducements (other than reasonable expenses and compensation for time) or inducements of any other kind be offered to participants?		✓
11.	Could the image of the NWU, the relevant academic department, your employer, or any other institution however affected by/involved in the project be negatively		✓

[3] ***Risk***: These possible risks are described as an "…invasion of privacy, loss of confidentiality, psychological trauma, indirect physical harm, embarrassment, stigma, and group stereotyping" (Oakes, 2002: 449), and also risks posed to . "…a subject's personal standing, privacy, personal values and beliefs, their links to family and the wider community, and their position within occupational settings, as well as the adverse effects of revealing information that relates to illegal, sexual or deviant behaviour" (Economic and Social Research Council (ESRC), 2005: 21). Minimal risk may be defined as where "…the probability and magnitude of harm or discomfort anticipated in the proposed research are not greater, in and of themselves, than those ordinarily encountered in daily life" (Code of Federal Regulations, 2005).

	affected by this research or put in a bad light?		

Please note that it is the responsibility of the student to follow NWU's Guidelines for Ethical Research as set out in the *Manual for Postgraduate studies* and any relevant academic or professional guidelines in the conduct of your study. **This includes providing appropriate information sheets and consent forms, and ensuring the confidentiality in the storage and use of data.** Any significant change in the question, design or conduct over the course of the research should be notified to the Study Leader and may require a new application for ethics approval.

	Please tick the appropriate box below:	**YES**	**NO**
12.	I have read the NWU's Manual for Postgraduate Studies and am familiar with the Guidelines for Research Ethics contained therein.	✓	

If you answered **no** to **questions 1-11**, submit the completed and signed form with your title registration. You should attach relevant documentation such as: applications and/or permission required to conduct the study, consent forms for participants (if applicable) and instruments you plan to use (if applicable). Students should retain a copy of the form and the decision taken by the relevant sub-committee and submit it with their dissertation/thesis.

If you answered **yes** to any of the **questions from 1-11**, you will need to describe more fully how you plan to deal with the ethical issues raised by your proposal. **This does not mean that you cannot do the research, only that your proposal will need to be approved by the Research Ethics Committee.** You will need to submit your plans for addressing the ethical issues raised by your proposal using the NWU Ethics Approval Application Form.

The NWU Ethics approval form may be obtained from: http://www.nwu.ac.za/library/documents/manualpostgrad.pdF

Alternatively, you may attach a fuller description of the specific issue to this declaration, for discussion by the panel at the Proposal Meeting.

CANDIDATE

Name and Surname: Bosele MI & Monyake TM

Date: 12 August 2016

SUPERVISOR

Name and Surname: Marlie Zeeman and Imelda Smit

Date: 12 August 2016

CHAIR: FACULTY RESEARCH SUB- COMMITTEE

Name and Surname: Prof B Surujlal

Date:

EVALUATING THE INTEGRATED DEVELOPMENT ENVIRONMENT IN C# USING HUMAN-COMPUTER INTERACTION PRINCIPLES.

First year students find it difficult to interact with the C# IDE and therefore fail to progress at required in the development of programming skills. Therefore we need to collect data on how the implementation of HCI principle in C# IDE helps to enhance student user experience for better learn.

We will also develop a selection test which will be used to classify and measure ITRW 123 student's understanding of the C# IDE.

The study will help determine if HCI principles were applied to the c# IDE. With the assistance of our supervisor, she has granted us opportunity to use a few students in her ITRW 124 class.

REFERENCES

Boote, D.N. & Beile, P. 2005. Scholars before researchers: On the centrality of the dissertation literature review in research preparation. *Educational researcher*, 34(6):3-15.

Charmaz, K. & Smith, J. 2003a. Grounded theory. *Qualitative psychology: A practical guide to research methods*:81-110.

Charmaz, K. & Smith, J. 2003b. Grounded theory.

Chen, E. & Schoeneman, C. 2004. On-demand presentation graphical user interface (pp. 1-10): Google Patents.

Cohen, D.J. & Crabtree, B.F. 2008. Evaluative criteria for qualitative research in health care: controversies and recommendations. *The Annals of Family Medicine*, 6(4):331-339.

Cutcliffe, J.R. 2000. Methodological issues in grounded theory. http://onlinelibrary.wiley.com/doi/10.1046/j.1365-2648.2000.01430.x/full Date of access.

DiCicco-Bloom, B. & Crabtree, B.F. 2006. The qualitative research interview. *Medical education*, 40(4):314-321.

Dix, A., Finlay, J. & Abowd, G.B. 2004. Human-computer interaction. Vol. 3.

Golafshani, N. 2003. Understanding reliability and validity in qualitative research. *The qualitative report*, 8(4):597-606.

Hinze-Hoare, V. 2007. The review and analysis of human computer interaction (HCI) principles. *arXiv preprint arXiv:0707.3638*:1-12.

Huberman, A.M. & Miles, M.B. 1994. Data management and analysis methods.

Johari, J. 2009. Interpretivism in information system (is) research. *Integration & Dissemination*, 4:25-27.

Kendall, J. 1999. Axial coding and the grounded theory controversy. *Western journal of nursing research*, 21(6):743-757.

Klein, H.K. & Myers, M.D. 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*:67-93.

Konsynski, B.R., Kottemann, J.E., Nunamaker Jr, J.F. & Stott, J.W. 1984. PLEXSYS-84: An integrated development environment for information systems. *Journal of Management Information Systems*, 1(3):64-104.

Kothari, C.R. 2004. Research methodology: Methods and techniques: New Age International.

Kvale, S. 2006. Dominance through interviews and dialogues. *Qualitative inquiry*, 12(3):480-500.

Luciano, T. 2011. Doing a literature review: releasing the social science research imagination. *Evaluation & Research in Education*, 24(4):303-304.

MacKenzie, I.S. 2012. Human-computer interaction: An empirical research perspective: Newnes.

Miskon, S., Bandara, W. & Fielt, E. 2015. Applying the principles of interpretive field research: An example of an IS case study on shared services.1-7.

Myers, B.A. 1998. A brief history of human-computer interaction technology. *interactions*, 5(2):44-54.

Ngwenyama, O.K. 1991. The critical social theory approach to information systems: problems and challenges (pp. 267-280): North-Holland, Amsterdam.

Nicola K Gale, G.H., Elaine Cameron, Sabina Rashid, Sabi Redwood. 2013. Using the framework method for the analysis of qualitative data in multi-disciplinary health research. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3848812/ Date of access.

Rajasekar, S., Philominathan, P. & Chinnathambi, V. 2006. Research methodology. *arXiv preprint physics/0601009*.

REIMER, J. 2005. A Histroy of the GUI. http://arstechnica.com/features/2005/05/gui/ Date of access.

Sharp, J. 2013. Microsoft Visual C# 2013 Step by Step: Pearson Education.

Terblanche, J.T. 2014a. Design of an e-Registration prototype using HCI principles: with specific reference to tax registration: IBIMA.

Terblanche, J.T. 2014b. Design of an e-Registration prototype using HCI principles: with specific reference to tax registration.

Tracy, S.J. 2012. Qualitative research methods: Collecting evidence, crafting analysis, communicating impact: John Wiley & Sons.

Vaishnavi, V.K. & Kuechler, W. 2015. Design science research methods and patterns: innovating information and communication technology: Crc Press.

Wilson, S. 2008. Research is ceremony: Indigenous research methods.

Yuan, Y. 2001. An inquiry into empirical pragmatics data-gathering methods: Written DCTs, oral DCTs, field notes, and natural conversations. *Journal of pragmatics*, 33(2):271-292.

Zak, D. 2015. Programming with Microsoft Visual Basic 2015: Cengage Learning.

APPENDIX A – Ethical clearance form

APPENDIX B – Questionnaire