Agile Development in the Irish Software Industry

Agile Development in the Irish Software Industry: Models for Change

By

Sineád Hayes

Edited By

Ita Richardson and Mícheál Ó hAodha

CAMBRIDGE
SCHOLARS
PUBLISHING

Agile Development in the Irish Software Industry: Models for Change,
by Sineád Hayes

This book first published 2009

Cambridge Scholars Publishing

12 Back Chapman Street, Newcastle upon Tyne, NE6 2XX, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-4438-0578-5, ISBN (13): 978-1-4438-0578-0

Acknowledgement: The research presented in this book was undertaken in part fulfilment of a M.Sc. in Software Engineering at the University of Limerick, Ireland, which was supported by the Higher Education Authority, Ireland.

“Engineering is not merely knowing and being knowledgeable, like a walking encyclopaedia; engineering is not merely analysis; engineering is not merely the possession of the capability to get elegant solutions to non-existent problems; engineering is practicing the art of the organised forcing of technological change.”
—*Dean Gordon Brown*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author wishes to thank a number of people who assisted in the completion of this book and for their support during the Master's programme:

Dr. Ita Richardson, for her assistance, guidance and encouragement in completing this project

Colleagues at "Rhythm Design Systems, Inc.", for participating in and contributing to this project

Family and friends, for their much appreciated support, advice and encouragement

Ian, for his unconditional love and support

CHAPTER ONE

INTRODUCTION

> “Change is the law of life and those who only look to the past or the present are certain to miss the future”
> —*John F. Kennedy*

1.1 Background to the Research

Rhythm[1] Design Systems Inc. (hereafter referred to as Rhythm) is one of the world’s leading Electronic Design Automation (EDA) companies. Founded in 1997, Rhythm’s headquarters are in San Jose, California but its 63 offices worldwide employ over 5,000 staff. The research described in this volume is based on a software development team located at the Dublin site, a location which is also home to the company’s corporate finance, manufacturing and electronic engineering offices.

Until September 2005, the primary mission of the Dublin software development team was the maintenance and support of a product called RHYTHMDoc, a product which consisted of the Rhythm help content viewing system. RHYTHMDoc allowed Rhythm customers to read and search the Rhythm product manuals, and display the requested content in a web browser or an Adobe PDF viewer. In 2004 a global customer satisfaction initiative revealed that customers were extremely frustrated with the RHYTHMDoc tool because various aspects of it lacked critical functionality. As a result, the senior management made a decision to cease the maintenance of RHYTHMDoc, thereby freeing up the resources and supports in order to develop a superior replacement product named Rhythm Help in Dublin.

[1] Rhythm Design Systems, Inc. is a pseudonym used in this volume in order to protect the identity of the company in question and the information provided herein.

The examination of RHYTHMDoc's maintenance projects revealed a fundamental problem in the teams' development approaches. Research demonstrated that these projects were consistently late, over-budget and - according to customer feedback - the quality of the delivered product was sub-standard. Analysis showed that the Waterfall development methodology which had been applied to projects in the past did not function well for the development team and was not suitable for the environment in which they now worked. The cumbersome nature of the Waterfall process meant that development resources were delayed and mired down in documentation reviews and updates rather than in the verification of the code required to implement newer features. To avoid a similar problem occurring in the Rhythm Help project, a more agile approach to development entitled Scrum was proposed. Scrum is a lightweight system and facilitates an iterative and incremental approach to development.

The introduction of such a dramatically different and unfamiliar development process was highlighted as a potential (critical) risk to the success of the project. In order to mitigate this risk the team followed Kotter's "Eight Steps to Transforming Your Organisation" (2005) when implementing the Scrum process. This volume describes the processes whereby the software development team at Rhythm implemented Kotter's guidelines in order to facilitate the transition from the Waterfall development methodology to agile development, using the Scrum process.

1.2 Research Objectives and Hypothesis

The aim of this study is to devise a suitable change management framework which software development companies can follow should they wish to make the transition from a traditional to an agile development methodology. The primary objectives of this study are to:

1. Investigate whether Kotter's "Eight Steps to Transforming Your Organisation" (*ibid*) is an appropriate and suitable change management technique to use within the software development industry.
2. Determine the impact agile development has on software projects.

This study sets out to prove the hypothesis that Kotter's "Eight Steps to Transforming Your Organisation" is a suitable change management technique to follow when moving from the Waterfall development methodology to the Scrum development methodology.

1.3 Justification for the Research

A literature review demonstrates how the change being made at Rhythm is one which is also being made by a range of other companies within the software industry. As there are no previous frameworks available, these companies could benefit greatly from having a specific and comprehensive change management model in place, one which can be followed by development teams when carrying out a similar change process. Many change management models and frameworks have been developed and published prior to this but these models relate primarily to organisational changes as opposed to software-related changes.

1.4 Methodology

After a comprehensive review of the literature, the Dublin Research and Development (R&D) team is closely observed as they introduce agile software development using the Scrum process. Kotter's "Eight Steps to Transforming Your Organisation" is used as a guideline for the change process. In addition to being observed, the 12 project members are surveyed at the beginning of the project and at various points throughout the course of the study. Interviews are conducted with both senior management teams and local management teams. An analysis of the collated information is used in conjunction with various project artefacts to form conclusions concerning the appropriateness of each of Kotter's recommendations. The final result of this analysis is the production of a suitable framework that could be utilised by other software development teams implementing a similar change process.

1.5 Outline of the Report

This chapter introduces the company involved in the study, describes the problems they are experiencing and how they intend to resolve them. Chapter 2 provides a review of the literature in relation to change management within organisations and the techniques used to perform such changes. It also provides a critique of the different development methodologies currently available to software organisations. Chapter 3 describes the research design that was used in this study, the framework that was adhered to, and the research techniques implemented throughout.

Chapter 4 is divided into eight sections, one for each of Kotter’s steps as implemented. Each section follows the actions of the software development team at Rhythm and describes the outcome and implementation of the requisite changes. Chapter 5 concludes the study by presenting a specific framework based on the experiences at Rhythm, one which other software development teams can adhere to if they are changing from a traditional development process to one which is more agile.

1.6 Definitions

The various definitions adopted by researchers are not always uniform and in many instances different words are used synonymously, all of which are not always technically correct. The key terms used in this study are therefore defined below in order to describe their intended meanings and remove any ambiguity in relation to them.

Methodology

A methodology refers to *“a recommended collection of phases, procedures, rules, techniques, tools, documentation, management and training used to develop a system” that has a “set of beliefs and assumptions underpinning it”*. (Avison and Fitzgerald, 2003)

Method

A method refers to “*a component of a methodology, alongside the techniques and tools.”* (Introna and Whitley, 1997)

Global Software Development (GSD)

Global Software Development involves *“the development of a software product by an organisation whose teams are geographically distributed”*. (Sa and Maslova, 2002)

1.7 Delimitations of Research Scope and Key Assumptions

The aim of this research is to create a framework that software development teams can follow when moving away from the Waterfall methodology to a more agile approach. Due to time limitations and a proximity to different software teams, it was not feasible to investigate every agile methodology available on the market. This research is therefore limited to a pre-existing, small and co-located team, all working on the same development project. The simplicity and flexibility of the Scrum process in addition to its suitability for small teams make it the most obvious process for implementation.

1.8 Conclusion

The research undertaken in this study aims to develop a specific step-by-step approach to implementing an agile development methodology in an organisation transferring from the more traditional or linear approach to a newer form of software development. A detailed overview of the current literature is provided in order to demonstrate how significant changes in the software industry have led to companies deciding to make such a transition. A critique of the various change management techniques available identifies Kotter's “Eight Steps to Transforming Your Organisation” (2005) as the most appropriate guideline to follow when implementing a process change in an organisation. The study described in this volume observes a software development team at Rhythm over a period of 18 months as they implemented Kotter’s guidelines and moved from the Waterfall development methodology to the more agile approach of Scrum.

CHAPTER TWO

LITERATURE REVIEW: CHANGE

> “The only thing that is constant is change”
> —*Heraclitus, Greek Philosopher*

2.1 Introduction

This chapter provides a definition of change and provides evidence for its omnipresence in society. It explores the concept of organisational change and outlines the various types of change which can exist within organisations, in addition to their potential impacts. The chapter also describes how the rate of change has dramatically increased in recent years and categorises the possible driving forces behind this. Concentrating on the software development industry, technology is identified as one of the main drivers for change. This chapter outlines how these changes have influenced product innovation and (in turn) the various processes used to develop these products in an effective manner. The necessity for software development organisations to ensure that their processes are as effective as possible is analysed and explained. By way of example, a clear differentiation is made between the traditional Waterfall development model and the more agile development that is characteristic of the Scrum process.

The discussion then explores the problems which can be encountered when implementing change in organisations and explains the reasons why change agents frequently prove unsuccessful. A critique of different change management models is provided in addition to an illustration of the various steps which managers need to take so as to ensure that any planned change is implemented effectively. The chapter concludes by describing the technique which was considered most suitable and which was adopted for the purposes of this research study.

2.2 Understanding Organisational Change

Change can be defined as - “*to make or become different*”, or more “ambivalently” - as “*dissatisfaction with the old and belief in the new*” (Harigopal, 2006). Regardless of how it is defined, it is a familiar concept that is either embraced or feared by those who anticipate it. Despite our familiarity with the concept, change has been the subject of much philosophical and sociological investigation and debate throughout the ages. From as far back as the 6th century BC (Heraclitus 534 B.C. cited in Barnes, 1982) and until the present day (Toffler, 2006) a long line of philosophers have presented their views and perceptions of the subject that is change. The underlying assumption that change is inevitable has remained consistent in the literature throughout the centuries albeit there is a distinct variation in each philosopher’s view regarding the pace at which change occur. In the past, for example, changes were distinct and predictable and generally occurred at a much slower pace as compared to the unpredictable, complex and often-turbulent manner whereby changes are experienced today.

An organisation is defined by Buchanan and Huczynski (2004) as *“a social arrangement for achieving controlled performance in pursuit of collective goals”*. Put simply, an organisation involves a group of people working together towards achieving the same objectives. In order to achieve these objectives, individuals within an organisation are supported by a common structure that comprises teams of people including managers, technology, culture, policies and procedures (Senior and Fleming, 2006). The stronger these supporting structures are, the more successful the organisation will be in achieving its intended goals. Inevitably however, there are always factors external to the organisation which influence and affect these internal structures. The most successful organisations are those which can withstand these pressures, adapt to these various outside influences and manage change effectively.

In order for organisations to manage change effectively it is necessary that they fully understand the changes which must take place. Organisational change can often cause feelings of expectation and trepidation among employees as a transformation of any kind can cause an innate sense of uncertainty within the workplace. The unrest and fear that is experienced can arise when people are removed from their “comfort zones” and forced to grow or adapt in order to survive. In this regard it is essential that managers are aware of potential problems or any resistance to change that may arise, and are capable of planning for this accordingly.

2.3 Types of Organisational Change

The speed at which organisational change takes place has increased so quickly over the last number of years that the question many organisations need to address is no longer whether or not the current trend will become obsolete, but how soon this will happen (Harigopal, 2006). This puts extreme pressure on management teams who are now required to learn how to manage these rapid developments and (simultaneously) cope with the different types of changes as they occur within an organisation. Table 1 outlines Harigopal's (*ibid*) comprehensive categorisation of these change types. Other researchers and research groups including the Harvard Business School (2003) provide a classification, which is "simpler" or more basic, classifying organisational change as either:

- Structural
- Cultural
- Procedural
- Cost cutting

Regardless of the type of change that is encountered within a particulular organisation, it has become clear that maintaining the *status quo* is no longer sufficient. Organisations are now encouraged, not only to respond to change but to anticipate it (Senior and Fleming, 2006). Being innovative and proactive, as opposed to reactive, is now recognised as the key to an organisation's survival, a proactive approach which carries fresh challenges for managers (Rooyen, 2000). Harigopal (2006) proposes that management can guarantee innovative success by observing and then aligning themselves with the changes which occur in the external environment in addition to the forces behind these changes.

Table 1: Harigopal's Classification of Change Types

Type of Change	*Description*
Total	*A drastic change of an existing system*
Planned	*A calculated change in response to a catalyst*
Happened	*A forced change due to an unpredicted development*
Transformational	*An entire organisational change required for survival*
Revolutionary	*An abrupt change in strategy*
Recreation	*A change to be different*
Strategic	*A change of all or most of the organisation's components*
Anticipatory	*A change as a result of an expected event*
Reactive	*A change in response to an event or series of events*

(Source: Adapted from Harigopal, 2006)

2.4 Forces for Organisational Change

There are a number of internal and external factors that force an organisation to change what they do and the way in which they do it. Johnson and Scholes (2002) and Goodman (1995) categorised these factors using the PEST (also known as STEP) mnemonics in an effort to demonstrate the issues that influence an organisation's operations, strategies and structures. Table 2, adopted from Senior and Fleming (2006) details a subset of the change-influencing factors as categorised using the PEST technique.

Senior and Fleming (*ibid*) also discuss the many reasons for the dramatic speed at which change occurs nowadays and outline some of the primary forces behind such rapid change as follows:

- Technological advancement
- Increased global competition
- Governmental interventions in the form of regulations
- The availability of better-skilled employees

These forces for change have had a significant impact on all organisations but for the software development industry technological advancement has been the key factor in influencing change. The progress made in this area has given rise to the information and digital revolutions, both of which had a dramatic effect on the way in which organisations worked, what they produced, and how they produced it.

Table 2: PEST Change Factors

Political Factors	**Economic Factors**
Government Legislation	*Competitors*
Government Ideology	*Suppliers*
International Law	*Currency Exchange Rates*
Universal Rights	*Employment Rates*
Wars	*Wage Rates*
Local Regulations	*Government Economic Policies*
Taxation	*Other Countries Economic Policies*
Socio-Cultural Factors	**Technological Factors**
Demographic Trends	*Information Technology*
Lifestyle Changes	*The Internet*
Skills Availability	*New Production Processes*

Attitudes to Work / Employment	*Computerisation of Processes*
Attitudes to Minority Groups	*Changes in Transport Technology*
Gender Issues	*Research and Funding*
Willingness / Ability to Move	*Technological Communication*

(Source: Adapted from Senior and Fleming, 2006)

2.5 How Technological Advancement Changed the Software Development Industry

The underlying notion presented in the literature – i.e. that *"change is constant"* - holds true across every industry and the software development sector, in particular (Law and Learn, 2005). The technological advances made during the past sixty years or so have undoubtedly been directly responsible for the significant changes which software development organisations have encountered.

When the chairman of IBM stated in 1943 that there was a world market for (perhaps) five computers, he had no way of knowing the extent to which the industry would grow and evolve, much less the impact which such technological advances would enforce. The transformation engendered by newer information technologies would prove so radical that by the turn of the century there would be a computer in nearly every home in the Western world (Fitzgerald et al., 2002). When the IBM chairman made that premature statement in 1943, organisations were using computers purely for scientific purposes i.e. to compute mathematical calculations at a speed much greater than was then otherwise humanly possible.

Twenty years later, the use of computers in business began to exceed their use in the scientific industry as business applications were able to perform automated clerical tasks, in less time and with greater accuracy. This led to a rapid expansion of computer installations in organisations during the 1970's (*ibid*). In the 1980's the introduction of the microprocessor completely changed and revolutionised computer architectures. It gave rise to the personal computer or PC which was significantly smaller and more powerful than its predecessors.

Without a doubt, however, the most influential technological factor was the emergence of the internet or the World Wide Web. On an individual level we use the internet every day for a host of reasons; keeping in touch with our friends, managing our financial affairs, and shopping for everything from a dream home to the ideal person to share it with. At an organisational and business level, however, the internet has had a much more extensive and influential impact. The arrival of the World Wide Web completely transformed the way in which organisations worked. The power of internet-based communication increased so dramatically that, when used correctly, it ensured significant commercial rewards for organisations. “Dotcom” industries who rode the crest of the internet wave also used it innovatively to enrich their business. Companies were now in a position to provide their customers with significantly improved services in addition to better channels of communication (Senior and Fleming, 2006). As well as improved services and communications, technological advances have also revolutionised electronic devices and software products.

2.5.1 The Evolution of Software Products

As with any industry, continuous innovation is the key to a software company’s success. Software organisations must be in a position to successfully adapt what they produce and the processes they employ to develop their products if they wish to survive and maintain their market share.

This is a complex task which is becoming ever more challenging as customers put huge demands on companies to produce high-quality products within shorter time frames. Given the volatility of the current IT markets, companies have no choice but to acquiesce to these demands and they therefore aim to provide their customers with cutting-edge software ahead of their competitors - and in the most cost-effective manner possible.

This intensely competitive environment is exacerbated by the fact that little loyalty exists in the industry as fickle customers support those who provide a product which:

- Uses the latest cutting edge technology
- Is functionally superior to its predecessors
- Is the most powerful and aesthetically pleasing

An example of the type of product innovation that has emerged due to technological change factors is evident within the personal entertainment device market. In 1979, Sony released the first personal stereo, walkman®. During its 20-year shelf-life this device was modified several times becoming smaller, more powerful and providing more features and capabilities than in its previous versions. By the time the walkman was "end-of-lifed" in the 1990s the replacement discman® was ready to take its place in the market, thereby guaranteeing Sony's continued market share. The discman® soon progressed to the mini-disc® player and now Sony's latest cash-cow is their new (tiny) portable mp3 player which was launched to rival the iPod® and the Creative Zen®.

While it is imperative that prototypes and products continue to advance and impress, the processes and procedures used to develop these products must also evolve accordingly. Without concrete processes and procedures in place, the end-product could be below the accepted quality level, prone to defects, and delivered late or over-budget. While the research and development teams strive to continuously change and improve the products they deliver, so too must the equivalent processes and procedures change to ensure that the most effective and efficient techniques are being applied so as to develop the highest quality products possible.

2.6 The Evolution of Software Processes

One of the critical aspects of a successful software development project is the process or methodology used by an organisation to produce and maintain their software. Like the products themselves, the processes employed must be cutting-edge and innovative if they are to achieve the desired results. The need for such processes first emerged during the 1960s when companies began to report common software development problems such as the failure to:

- deliver projects on time
- deliver projects within budget
- meet customer requirements and expectations

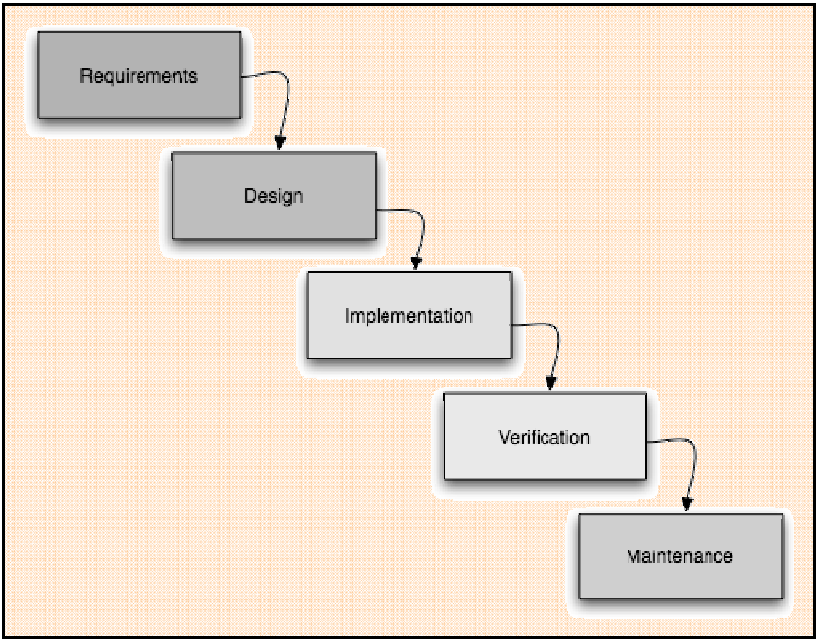
Avison and Fitzgerald (2003) define a software development methodology as *"a recommended collection of phases, procedures, rules, techniques, tools, documentation, management and training used to develop a system"* that has a *"set of beliefs and assumptions underpinning it"*. A good methodology is one which is clearly defined and documented, one which is clearly understood by those who are carrying out the process and which is improved on a regular basis.

The Waterfall development lifecycle in which *"a software product is viewed as progressing linearly from conception through requirements, design, code and test"* (Laplante and Neill, 2004) was the original approach to software development. Introduced by Royce in 1970 in an effort to overcome the *software crisis*, it was designed to improve project management by separating any project into phases which had to be carried out in a predefined order. Figure 1 displays the Waterfall sequential approach that assumed the requirements were stable and fully-evolved before proceeding onto the design phase. Each phase had to meet specific criteria before progression to the next was possible.

In recent years this methodology has been the subject of much criticism as documented experiences have since proved that, in reality, requirements cannot be set down at any point and will constantly change throughout the project lifecycle (The Chaos Report, 1994). It has also been described as a cumbersome and overly-stringent approach to development, one which has consequently failed to resolve the software crisis (Gallant et al., 1996). Advocates have been forced to continue seeking new and improved methods for developing software, methods which have resulted in a significant change as regards how software projects are managed and developed.

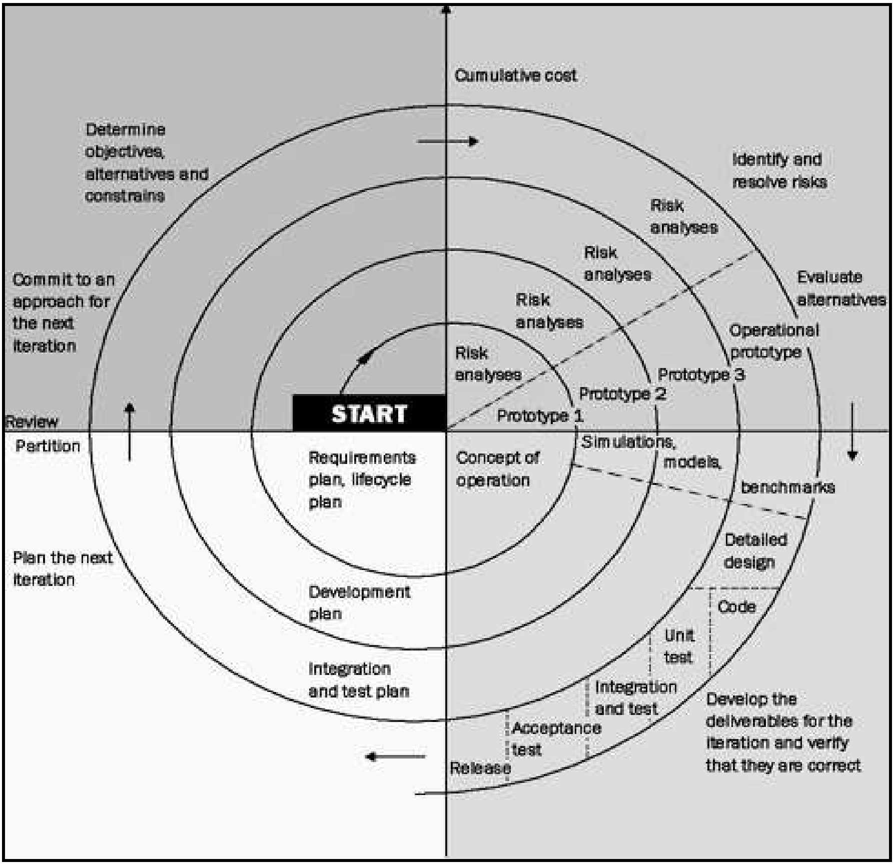
Barry Boehm (1988) was responsible for the introduction of the new Spiral model which was a marked improvement on Royce's Waterfall model (1970) and overcame some of the pitfalls that were associated with it (Galin, 2004). Figure 2 shows how the Spiral model continued to use the Waterfall model as a base but enhanced it with the use of prototyping. This process allows software to be developed on a cyclical basis using a subset of requirements at particular stages of the process. The end result of each development cycle is a working prototype that demonstrates the functionality of a product and allows for the refinement and modification of the necessary requirements as the product evolves (Ramamorthy, 1996).

Figure 1: The Waterfall Model



(Source: Adapted from Creative Attribution, 2005)

Figure 2: The Spiral Model



(Source: Adapted from Barry Boehm, 1988 © 1988, IEEE)

The 1990s witnessed a further transformation and evolution of software development processes. The software industry experienced a complete shift away from the traditional waterfall methodology to more iterative and incremental approaches to development. This new phenomenon, called agile development, saw the introduction of concepts such as Scrum, Test Driven Development (TDD) and Extreme Programming (XP). Larman (2004) describes the focus of agile development as encompassing:

- Close collaboration between the development teams and stakeholders
- Less documentation
- Frequent delivery of code (less features, more often)
- Self-organizing teams
- Ways to craft the code and the project teams so that all inevitable changes to requirements do not become an issue.

The success of the new approach was instantaneous (Schatz and Abdelshafi, 2005) and as the very positive feedback and empirical evidence for this new approach grew, so too did the range and diversity of agile methodologies (Conboy and Fitzgerald, 2004). Key benefits reported as an element of these new success stories included the faster delivery of higher-quality products and the better matching of customer requirements as a consequence of their close involvement throughout the project process. It has also been argued that the transition to agile methodologies was initiated as a way of achieving a positive return on investment in quality, and at an early stage in the development life cycle (Leszak et al., 2000).

All reports relating to such revolutionary agile development techniques did not describe entirely-positive experiences, however (Law and Learn, 2005). Despite promoting the fact that they are *“simple”* and *“quick”* (Beck, 2000), most processes are very difficult to get right and require extensive training, employee discipline and managerial support. The ever -increasing number of agile methods that are available also present a problem – i.e. not every technique is suitable for each type of project. This factor must be given serious consideration before any specific development methodology is chosen for a project.

The appropriate process or methodology for a project should be chosen at an early stage of the process because it has the potential to improve the quality of the product being developed. While all development methodologies strive to serve the same purpose - i.e. assist in the production of high-quality and low-cost software which is delivered on-time; they each approach development in a different way.