

MSc in Computing (Information Technology)
**Agile and Unified Process as examples of modern technologies
applied by software producers.**

Jakub Jarosz

D07112449

School of Computing,
Dublin Institute of Technology,
jakub.jarosz@student.dit.ie

Abstract

Creating software is a challenging business. Nowadays it has become impossible to develop sophisticated computer programs without applying proper methodology. Moreover, the chosen methodology should itself meet certain quality requirements and should be easy to apply within the ISO worldwide certification standards. The aim of this paper is to explain key features of two main software methodologies: the Unified Process and the Agile, and to show how these methodologies have been applied within Irish software producers.

1 Introduction

Nowadays computer software is seen as the most important technology among other engineering disciplines. Despite the fact that “software engineering is one of the youngest disciplines, its impact can be noticed almost everywhere”[1]. Over the years humans have become increasingly dependent on computer programs. Unlike in the past decades, software currently is more complicated and the state of the art software solutions has taken on a dual role. One of the roles is to fill in consumer needs, the second role is as an engine for delivering a final product. An example of a final product might be a software working on a standalone PC or more likely, on networked PC systems. As an engine for delivering the final product, software acts not only as a platform for making hardware work (e.g. operating systems), but is also used for the communication of information (computer networks) and as a platform for developing other software. In other words “software exists everywhere and a major part of our daily life depends on it” [2]. However, software is written by people and as a consequence programs are not perfect.

From early days of software engineering written programs have not been free of bugs, which are products of human mistakes. Along with an increasing demand for sophisticated software systems, “final clients, vendors and producers are now highlighting the need for better quality” [3]. Quality itself can be increased by applying appropriate development

and testing methodologies within software engineering processes.

2 Software development methodologies

Creating a computer program involves tens and sometimes even hundreds or thousands of team members. Team members play different roles and work together under tight schedules. “Decisions made by teams and interactions between team members can be taken as a key part of a software developing process” [4]. Nowadays there are two major models which give a framework for software engineering:

- Unified process
- Agile methodology

Regardless of the type of development methodology, each model consists of main activities like: assigning tasks, setting milestones, taking actions and using tools that are required to achieve final goals. The model should be capable of being adopted well by a software producer and will provide stability and control over complicated processes within a company. Nowadays, those two listed models: Unified and Agile have become extremely popular.

2.1 Unified Process

The history of remarkable success of Unified Process (UP) began in the early 1990s. The UP is tightly linked with object-oriented methods and programming languages. Along with the UML language, which has become an industry standard, the Unified process provides a framework to guide projects and developer-teams from the beginning to the end of the software life-cycle.

The Unified process consist of following phases:

- Inception
- Elaboration
- Construction
- Transition
- Production

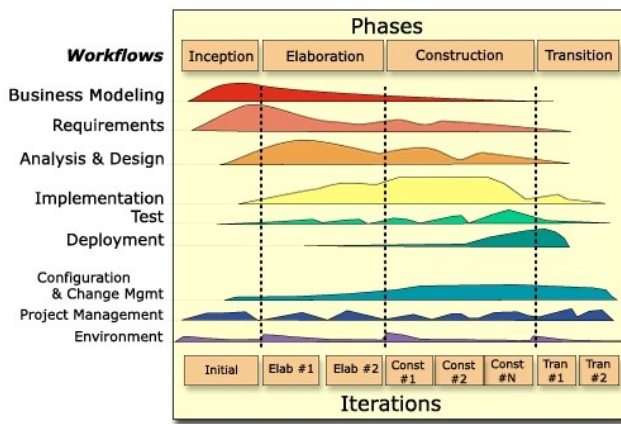


Figure 1. Unified Process phases

In the **inception** phase two activities are emphasized: communication with customers and planning activities. During the inception phase technical and business requirements are identified and transformed into use-cases. In the **elaboration** phase the process model is created and at this stage any modifications can be made. The next, **construction** phase consists of coding activities. Modeled earlier functions and software features are implemented in the source code. The **transition** phase is the phase when the software code is beta-tested and additional documentation is created. The last, **production** phase consist of developing the final product, monitoring its behavior and final requests for changes are submitted and evaluated.

The Unified Process has been successfully implemented by a Dublin based company – XSIL International Ltd. XSIL is the world leading provider of laser dicing and drilling technologies for the semiconductor industry. Products made by XSIL put together state of the art mechanical, chemical, optical, electronics and software engineering.



Figure 2. X300D+ Xsil's machine for cutting silicon wafers. [13]

The unified software developing process in XSIL has been fitted into ISO 9001:2000 framework which has in addition increased quality of developed products. The UP has been implemented within XSIL based on IBM Rational software. The four developing phases of the UP and the work flow are monitored by project managers. Team

members are assigned competencies and are responsible for the creation and the monitoring of documents assigned to each stage of the work flow. The documentation created according IEEE standards [5,6] is stored in folders mapped to the particular unified process phase.

The life-cycle time frame of described processes is usually longer than one year. The software running on the machines produced by XSIL is developed, tested and deployed under specific market circumstances. It means that software features and functionality are not defined by particular clients. Moreover, the time frame is not tightly defined. The functionality and the time frame is defined by XSIL itself and the market demand for advanced technology. The case in point might be the newest XSIL's product – machine X300D+ (Figure 2) which delivers unique laser technology for cutting thin silicon wafers.

The need to create documentation and to fit the production process into ISO framework makes the UP the best choice for XSIL.

2. Agile development

“Agile development methodology became popular in the early 2000s”. [7] The Agile software engineering combines a set of developing paths and the philosophy of the customer and provider working in close collaboration. Agile methodology encourages client satisfaction as the most important feature(what) within the business process and consequently has direct and indirect impact on quality.

The main key features of the agile methodologies can be found in the document called “Manifesto for Agile Software Development”:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan* “ [8]

The Agile methods represent an alternative to conventional methods. Agile is seen as the method which does not require as many documents as the UP. “The main framework activities like communication with a customer, planning, modeling, construction, delivery and evaluation still remain but are limited to minimal number of tasks” [8].

Among many companies worldwide which have adopted the agile methodologies is an Irish company - NewBay Software Ltd. NewBay “provides state of the art software for mobile operators” [9] and specializes in developing “Web 2.0” solutions. The adopted methodology works extremely well and allows NewBay to meet customers needs in an efficient way.

Every team within NewBay consists of different specialists: project managers, developers, testers and operation engineers. The software life cycle activities like developing,

testing, deploying are performed together with engineers from the customer's team.

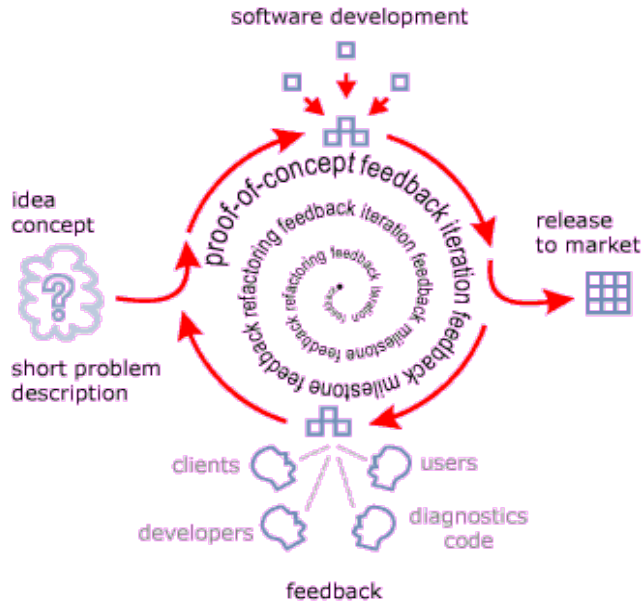


Figure 3. The Agile concept

Many activities require access to the customer's laboratories. A case in point might be the software developed for a mobile operator - "T-Mobile" which opened access to its laboratory in Bonn in Germany. (NewBay's engineers were responsible for setting up servers while German engineers carried on setting up routers. Both group of specialists were given access to the customers laboratory .It can be seen as a remarkable example of working with customers within the agile methodology framework and can be accurately described as tight collaboration.

From the perspective of NewBay Software Ltd, which copes with fast changing web technologies and hardware (mobile handsets), it is easy to conclude that there were no other ways to achieve great success other than to chose the Agile software development model.

3 Conclusion

“At the beginning of many projects, clients rarely can define system (product) requirements precisely” [10]. The non-agile approach, which is the UP gives engineers only one shot at getting the analysis and design right. Discovering requirements late in the project can significantly increase its overall cost. The UP requires to perform analysis and develop many written documents prior to any modeling and coding activities. From this point of view the UP fits better than other methodologies into companies like XSIL.

Agile methodology focuses on building software rather than on creating documentation. The “agile lifecycle consist of small but frequently make iterations” [11]. As a consequence the developed product (computer program) is continuously being modified. Frequently changed

requirements do not have negative impact on the entire developing process. This asset makes agile methodology a perfect choice for companies like NewBay Software Ltd.

In conclusion, “to chose and adopt the right software development methodology is the key decision for the company's future” [12]. Many technical and economical aspects should be taken into consideration in order to make the right decision, to achieve and sustain success in the market.

References

- [1] Lindsey Vereen, *Improving Software Quality*. Software Development Times, June 2005
- [2] Ron Patton, *Software Testing*, Sams, 2006
- [3] Shep Hyken, *The Dangerous Customer*. The Quality Assurance Institute Journal, January 2007
- [4] Kaner, Falk, Nguyen , *Testing Computer Software*, Wiley; 2004.
- [5] *IEEE Standard for Software Test Documentation*. The Institute of Electrical and Electronics Engineers, Inc.(USA) September 1998
- [6] *IEEE Recommended practice for Software Requirements Specifications* The Institute of Electrical and Electronics Engineers, Inc. (USA) June 1998
- [7] Brian Maric. *Agile Addendums*, Better Software, July 2007
- [8] Roger S. Pressman, *Software Engineering – A Practitioner's Approach*, McGraw-Hill Int., 2005.
- [9] www.newbay.com
- [10] Allan Shalloway. *Jack be Agile Jack be Lean. Sustainable Product Development. The Lean Agile Connection*. Better Software June 2007
- [11] Hambling, Morgan, Samaroo, Thompson, Williams, *Software Testing. An ISEB Foundation*, British Computer Society Press, 2007
- [12] Dr. Gerard M. Weinberg, *Destroying Communication and Control in Software Development*. The Quality Assurance Institute Journal, April 2007
- [13] www.xsil.com