Software development process

Author Name (ZhengXize)
StudentNumber （13211247）*
Zhejiang Normal University
Software Project Management

Abstract

The production of software is a labour intensive.Given the scale of software projects related to current and future HEP experiments,it is worth trying to improve the knowledge of the PEOPLE involved , the organization of the software development PROCESS and the TECHNOLOGY used in the various aspect of this activity. The goal is better systems at lower cost and happier users of the software.

Keywords: Software develop process,methodology,model,framework

1 Introduction

In software engineering, a software development methodology (also known as a system development methodology, software development life cycle, software development process, software process) is a splitting of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application. [1]

2 In practice

A variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One software development methodology framework is not necessarily suitable for use by all projects. Each of the available methodology frameworks are best suited to specific kinds of projects, based on various technical, organizational, project and team considerations.[1]

Software development organizations implement process methodologies to ease the process of development. Sometimes, contractors may require methodologies employed, an example is the U.S. defense industry, which requires a rating based on process models to obtain contracts. The international standard for describing the method of selecting, implementing and monitoring the life cycle for software is ISO/IEC 12207.

A decades-long goal has been to find repeatable, predictable processes that improve productivity and quality. Some try to systematize or formalize the seemingly unruly task of designing software. Others apply project management techniques to designing software. Without effective project management, software projects can easily be delivered late or over budget. With large numbers of software projects not meeting their expectations in terms of functionality, cost, or delivery schedule,[citation needed] it is effective project management that appears to be lacking.

Organizations may create a Software Engineering Process Group (SEPG), which is the focal point for process improvement. Composed of line practitioners who have varied skills, the group is at the center of the collaborative effort of everyone in the organization who is involved with software engineering process improvement.

A particular development team may also agree to programming environment details, such as which integrated development environment is used, and one or more dominant programming paradigms, programming style rules, or choice of specific software libraries or software frameworks. These details are generally not dictated by the choice of model or general methodology.

3 Overview

3.1 Process Activities/Steps

- Requirements Analysis Extracting the requirements of a desired software product is the first task in creating it. While customers probably believe they know what the software is to do, it may require skill and experience in software engineering to recognize incomplete, ambiguous or contradictory requirements.
- Specification Specification is the task of precisely describing the software to be written, in a mathematically rigorous way. In practice, most successful specifications are written to understand and fine-tune applications that were already well-developed, although safety-critical software systems are often carefully specified prior to application development. Specifications are most important for external interfaces that must remain stable.
- Software architecture The architecture of a software system refers to an abstract representation of that system. Architecture is concerned with making sure the software system will meet the requirements of the product, as well as ensuring that future requirements can be addressed.
- Implementation Reducing a design to code may be the most obvious part of the software engineering job, but it is not necessarily the largest portion.
- Testing Testing of parts of software, especially where code by two different engineers must work together, falls to the software engineer.
- Documentation An important task is documenting the internal design of software for the purpose of future maintenance and enhancement.
- Training and Support A large percentage of software projects fail because the developers fail to realize that it doesn't matter how much time and planning a development team puts into creating software if nobody in an organization ends up using it. People are occasionally resistant to change and avoid venturing into an unfamiliar area, so as a part of the deployment phase, its very important to have training classes for the most enthusiastic software users (build excitement and confidence), shifting the training towards the neutral users intermixed with the avid supporters, and finally incorporate the rest of the organization into adopting the new software. Users will have lots of questions and software problems which leads to the next phase of software.

*e-mail:Authors Email

- Maintenance Maintaining and enhancing software to cope with newly discovered problems or new requirements can take far more time than the initial development of the software. Not only may it be necessary to add code that does not fit the original design but just determining how software works at some point after it is completed may require significant effort by a software engineer.

3.2 Process Models

A decades-long goal has been to find repeatable, predictable processes or methodologies that improve productivity and quality. Some try to systematize or formalize the seemingly unruly task of writing software. Others apply project management techniques to writing software. Without project management, software projects can easily be delivered late or over budget. With large numbers of software projects not meeting their expectations in terms of functionality, cost, or delivery schedule, effective project management is proving difficult.

3.3 Why modelling?

Because model can provide a common understanding,use to locate any inconsistencies,redundancies and omissions,use to reflect the development goals and provide early evaluation and use to assist the development team to understand any special situation.

Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and various types of agile methodology. Some people consider a life-cycle ”model” a more general term for a category of methodologies and a software development ”process” a more specific term to refer to a specific process chosen by a specific organization. For example, there are many specific software development processes that fit the spiral life-cycle model.

3.4 Approaches

Several software development approaches have been used since the origin of information technology, in two main categories. Typically an approach or a combination of approaches is chosen by management or a development team.”Traditional” methodologies such as waterfall that have distinct phases are sometimes known as software development life cycle (SDLC) methodologies, though this term could also be used more generally to refer to any methodology. A ”life cycle” approach with distinct phases is in contrast to Agile approaches which define a process of iteration, but where design, construction, and deployment of different pieces can occur simultaneously.

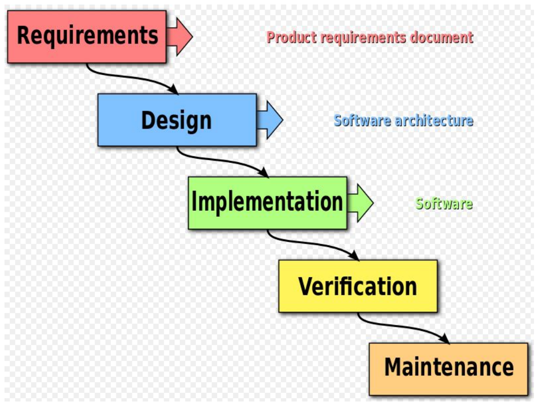

Figure 1: *The waterfall model*

- The first formal description of the method is often cited as an article published by Winston W. Royce[2] in 1970 although Royce did not use the term ”waterfall” in this article. The waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through several phases.It is a classical model use to software development.Although it can effective control software develop but it limited scope of iteration and need long cycle time not suitable for system of high uncertainty .So it is no longer used.

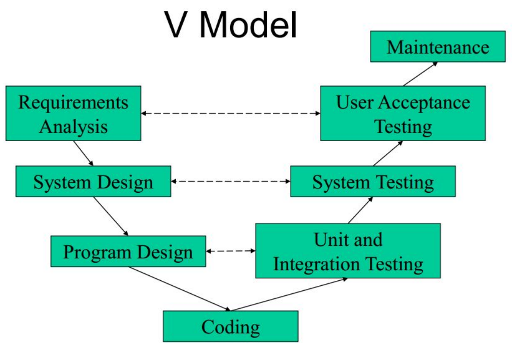

Figure 2: *V Model*

- The V model have some additional validation process introduced.Relate testing to analysis and design and loop back in case of discrepancy.

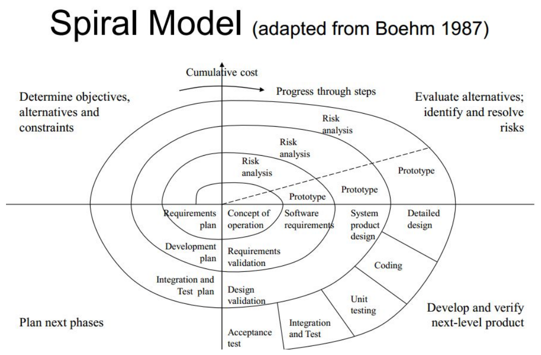

Figure 3: *Spiral Model*

- The Spiral is an evolutionary approach.It combines some key aspect of the waterfall model and rapid prototyping methodologies, in an effort to combine advantages of top-down and bottom-up concepts. It provided emphasis in a key area many felt had been neglected by other methodologies: deliberate iterative risk analysis, particularly suited to large-scale complex systems.Each cycle involves a progression through the same sequence of steps, for each part of the product and for each of its levels of elaboration, from an overall concept-of-operation document down to the coding of each individual program.”[3]Each trip around the spiral traverses four basic quadrants: (1) determine objectives, alternatives, and constraints of the iteration; (2) evaluate alternatives; Identify and resolve risks; (3) develop and verify deliverables from the iteration; and (4) plan the next iteration.[4]Begin each cycle with an identification of stakeholders and their ”win conditions”, and end each cycle with review and commitment.[5]
- The prototyping model the goals are meet (some) user requirements at an early stage,reduce risk and uncertainty,verify a design or implementation approach.It benefit learning by doing ,improved communication with user,improved user involvement and clarification of partially-known requirements.
- other The are many other methods to those listed above, and you can find out more by visiting the websites below.

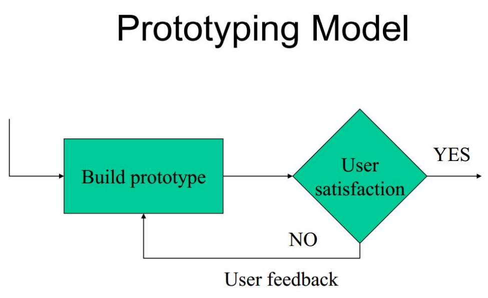

Figure 4: *Prototyping Model*

4 Conclusion

In the rapid development of software development and design,Programmers should also learn to master different models to cope with different kinds of project requirements.

5 References

1.Centers for Medicare and Medicaid Services (CMS) Office of Information Service (2008). Selecting a development approach. Webarticle. United States Department of Health and Human Services (HHS). Revalidated: March 27, 2008. Retrieved 27 Oct 2008.

2. Wasserfallmodell ¿ Entstehungskontext, Markus Rerych, Institut f眉r Gestaltungs- und Wirkungsforschung, TU-Wien. Accessed on line November 28, 2007

3. Barry Boehm (1996., ”A Spiral Model of Software Development and Enhancement”. In: ACM SIGSOFT Software Engineering Notes (ACM) 11(4):14-24, August 1986

4.Richard H. Thayer, Barry W. Boehm (1986). Tutorial: software engineering project management. Computer Society Press of the IEEE. p.130

5.Barry W. Boehm (2000). Software cost estimation with Cocomo II: Volume 1.