Software Development Processes

Abang Teboh Tatiana

Zhejiang Normal University
Software Project Management

Abstract

This report gives a brief and concise explanation of software development process and some of the software process models. It goes further to outline some of the advantages and disadvantages of each model. Given that this is a short report and there many software process models, we chose to discuss three of them; waterfall model, prototyping model and phased development model. Thereby we suggest further reading in case the reader wants to know more about other models like the V Model, Spiral Model, Fish Model and so on.

1 Introduction

Software development process is a set of activities, methods, and practices that are used in the production and evolution of software. It is sometimes called software life cycle because it describes the life of a software product from its conception to its implementation, delivery, use and maintenance.Each process can be described in a variety of ways using text, pictures, or a combination. Software engineering reseachers have suggested a variety of formats for such descriptions, usually organized as a model that contains key process features.[1]

2 Software Process Models

A software process model is an abstract representation of a process.Many process models are described in the software engineering literature. Some are prescriptions for the way software development should progress, and others are descriptions of the way software development is done in actuality. In theory the two kinds of models should be similar or the same but in practice they are not. Building a process model and discussing its subprocesses help the team understand the gap between what should be and what is.[1] Three kinds of models will be discussed in this report. They are: Waterfall, Prototyping and Phased development models. Other models include: the V model, Spiral model, Operational Specification Model, Transformational Model, Fish model etc.

2.1 Waterfall Model

The waterfall model, as illustrated in figure 1, is the classical model of software. As the figure implies, one development stage should be completed before the next begins.

It is **advantageous** in that it is simple to understand which makes it easy to explain to customers who are not familiar with software development. It is a good model to use when requirements are clearly defined and developers know exactly what to do. One of its **drawbacks** is that it does not provide for iterations or loops as such in case of any discrepancy during development, developers will have to go back to the start to be able to understand the problem and propose solutions to address it.[2]

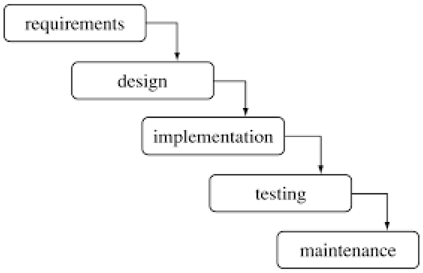

Figure 1: Waterfall Model

2.2 Prototyping Model

Its goal is to meet (some) user requirements at an early stage. Requirements or design require repeated investigation to ensure that the developer, user and customer have a common understanding both of what is needed and what is proposed. Prototype has three classifications:

- Throwback: After users agree the requirements of the system, the prototype will be discarded.
- Evolutionary: Modifications are based on the existing prototype.
- Incremental: Functions will be arranged and built accordingly.

Prototyping is **advantageous** in that it demonstrates the consistency and completeness of a specification. But sometimes users misunderstand the role of the prototype.

2.3 Phased Development: Increments and iterations

One way to reduce cycle time is to use phased development. The system is designed so that it can be delivered in pieces enabling users to have some functionality while the rest is being developed. Thus, there are usually two systems functioning in parallel: the production and the development system. The production system is the one currently being used by the customer and user; the development system is the next version that is being prepared to replace the current production system.

Two most popular approaches are incremental development and iterative development. In **incremental devlopment**, the system as specified in the requirements document is partitioned into subsystems by functionality. The releases are defined by beginning with one small, functional subsystem and then adding functionality with each new release. However, **iterative development** delivers a full system at the very beginning and then changes the functionality with each release. Phased development model is **advantageous** in that it allows for early feedback and there is less possibility of requirements changing. As

drawbacks it might cause software breakage and reduced productivity.

3 Conclusion

There is no best software process model for software development activities. As such the suitability of a model will depend on requirements and the understanding of developers.

References

[1] Shari Lawrence Pfleeger and Joanne M. Atlee *Software Engineering Theory and Practice Fourth edition*

[2] Nabil Mohammed Ali Munassar and A. Govardhan *A Comparison Between Five Models Of Software Engineering http://www.ijcsi.org*