Comparison of Software Process Model under the Software Engineering

Taranpreet Singh
Department of Information Technology Engineering
S.B.B.S.I.E.T, Jalandhar, Punjab, India

Paramjot Kaur
Department of Information Technology Engineering
S.B.B.S.I.E.T, Jalandhar, Punjab, India

Abstract- A software development process, also known as a software development life-cycle (SDLC), is a structure imposed on the development of a software product. Similar terms include software life cycle and software process. It is often considered a subset of systems development life cycle. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Some people consider a life-cycle model a more general term and a software development process a more specific term. For example, there are many specific software development processes that 'fit' the spiral life-cycle model. ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

I. INTRODUCTION

Software Engineering

The seminal definition: Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

The IEEE definition: Software Engineering: The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

- More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.
- It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

Software Process Models

- A software process model is an abstract representation of a process. It presents a description of a process.
- When we describe and discuss processes, we usually talk about the activities in these processes such as specifying a data model, designing a user interface, etc. and the ordering of these activities.
- Process descriptions may also include:
  - Products, which are the outcomes of a process activity;
  - Roles, which reflect the responsibilities of the people involved in the process;
  - Pre- and post-conditions, which are statements that are true before and after a process activity has been enacted or a product produced.

II. MAJOR SOFTWARE PROCESSES

Major Software Processes

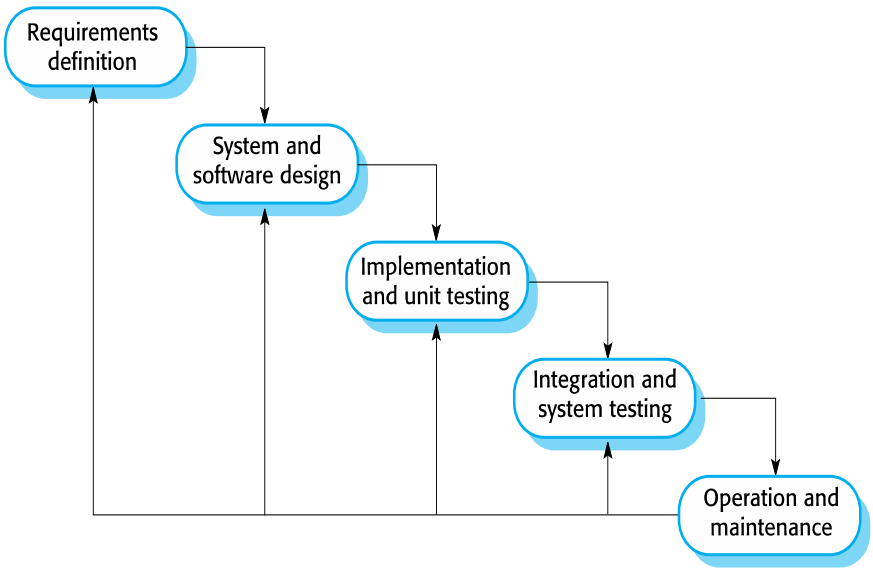
Plan-driven and agile processes

- Plan-driven processes are processes where all of the process activities are planned in advance and progress is measured against this plan.
- In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements.
- In practice, most practical processes include elements of both plan-driven and agile approaches.
- There are no right or wrong software processes.

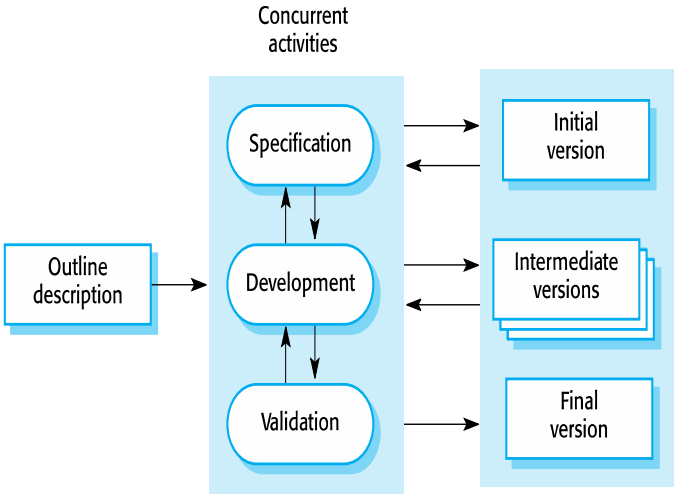
Software Process Models

- The waterfall model
  - Plan-driven model. Separate and distinct phases of specification and development.
- Incremental development
  - Specification, development and validation are interleaved. May be plan-driven or agile.
- Reuse-oriented software engineering
  - The system is assembled from existing components. May be plan-driven or agile.

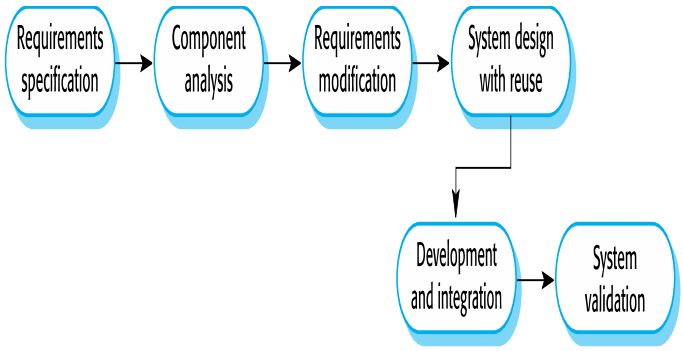
- The waterfall model



- Incremental development



- Reuse-oriented software engineering



IV.CONCLUSION

There is no best software development model. Depending on the size of the program definition and the time constraints on the project a certain model may be better than another. I prefer the reuse-oriented software engineering model.

IV.ACKNOWLEDGEMENT

This report has been prepared by Taranpreet Singh and Paramjot Kaur from S.B.B.S.I.E.T. Valuable comments have been offered by a number of persons including Assistant Professor Tajinder Kaur from S.B.B.S.I.E.T. Their comments have been incorporated where applicable.

REFERENCES

[1] Stellman, Andrew; Greene, Jennifer (2005). Applied Software Project Management. O'Reilly Media. ISBN 978-0-596-00948-9.
[2] IEEE magazine article "Why Software Fails" John C. Reynolds, Some thoughts on teaching programming and programming languages, SIGPLAN Notices, Volume 43, Issue 11, November 2008, p.108:
[3] Jalote, Pankaj (2002). Software project management in practice. Addison-Wesley. ISBN 0-201-73721-3.

[4] Murali Chemuturi, Thomas M. Cagley Jr. & (2010). Software Project Management: Best Practices, Tools and Techniques. J.Ross Publishing. ISBN 978-1-60427-034-1
[5] Resources on Software Project Management from Steve McConnell: http://www.construx.com/Page.aspx?nid=22
[6] Resources on Software Project Management from Dan Galorath: http://www.galorath.com/wp/category/project-management