Software Development Processes (Methods)

ZhangXiang
13211144
ZSD university

Abstract

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.Introduction

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.

2.History

1970s

- Structured programming since 1969
- Cap Gemini SDM, originally from PANDATA, the first English translation was published in 1974. SDM stands for System Development Methodology

1980s

- Structured systems analysis and design method (SSADM) from 1980 onwards
- Information Requirement Analysis/Soft systems methodology

1990s

- Object-oriented programming (OOP) developed in the early 1960s, and became a dominant programming approach during the mid-1990s

- Rapid application development (RAD), since 1991
- Dynamic systems development method (DSDM), since 1994
- Scrum, since 1995
- Team software process, since 1998
- Rational Unified Process (RUP), maintained by IBM since 1998
- Extreme programming, since 1999

2000s

- Agile Unified Process (AUP) maintained since 2005 by Scott Ambler
- Disciplined agile delivery (DAD) Superseded of AUP

3. Formal methods

Formal methods are mathematical approaches to solving software (and hardware) problems at the requirements, specification, and design levels. Formal methods are most likely to be applied to safety-critical or security-critical software and systems, such as avionics software. Software safety assurance standards, such as DO-178B, DO-178C, and Common Criteria demand formal methods at the highest levels of categorization.

For sequential software, examples of formal methods include the B-Method, the specification languages used in automated theorem proving, RAISE, and the Z notation.

Formalization of software development is creeping in, in other places, with the application of Object Constraint Language (and specializations such as Java Modeling Language) and especially with model-driven architecture allowing execution of designs, if not specifications.

For concurrent software and systems, Petri nets, process algebra, and finite state machines (which are based on automata theory - see also virtual finite state machine or event driven finite state machine) allow executable software specification and can be used to build up and validate application behavior.

Another emerging trend in software development is to write a specification in some form of logic—usually a variation of first-order logic (FOL)—and then to directly execute the logic as though it were a program. The OWL language, based on Description Logic (DL), is an example. There is also work on mapping some version of English (or another natural language) automatically to and from logic, and executing the logic directly. Examples are Attempto Controlled English, and Internet Business Logic, which do not seek to control the vocabulary or syntax. A feature of systems that support bidirectional English-logic mapping and direct execution of the logic is that they can be made to explain their results, in English, at the business or scientific level.

4. Process meta-models

Some "process models" are abstract descriptions for evaluating, comparing, and improving the specific process adopted by an organization.

- ISO/IEC 12207 is the international standard describing the method to select, implement, and monitor the life cycle for software.
- The Capability Maturity Model Integration (CMMI) is one of the leading models and based on best practice. Independent assessments grade organizations on how well they follow their defined processes, not on the quality of those processes or the software produced. CMMI has replaced CMM.
- ISO 9000 describes standards for a formally organized process to manufacture a product and the methods of managing and monitoring progress. Although the standard was originally created for the manufacturing sector, ISO 9000 standards have been applied to software development as well.

Like CMMI, certification with ISO 9000 does not guarantee the quality of the end result, only that formalized business processes have been followed.

- ISO/IEC 15504 *Information technology — Process assessment* also known as Software Process Improvement Capability Determination (SPICE), is a "framework for the assessment of software processes". This standard is aimed at setting out a clear model for process comparison. SPICE is used much like CMMI. It models processes to manage, control, guide and monitor software development. This model is then used to measure what a development organization or project team actually does during software development. This information is analyzed to identify weaknesses and drive improvement. It also identifies strengths that can be continued or integrated into common practice for that organization or team.
- Soft systems methodology - a general method for improving management processes
- Method engineering - a general method for improving information system processes

2.Conclusion

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.

Acknowledgements

Thanks to my classmates helped me.

References

- Centers for Medicare & Medicaid Services (CMS) Office of Information Service (2008). *Selecting a development approach*. Webarticle. United States Department of Health and Human Services (HHS). Re-validated: March 27, 2008. Retrieved 27 Oct 2008.
- · Geoffrey Elliott (2004) *Global Business Information Technology: an integrated systems approach*. Pearson Education. p.87.
- · Wasserfallmodell > Entstehungskontext, Markus Rerych, Institut für Gestaltungs- und Wirkungsforschung, TU-Wien. Accessed on line November 28, 2007.
- ieeecomputersociety.org