Software Development Processes

Name Song Tianyu number 13211135
Zhejiang Normal University
Software Projects Management

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.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.

2 History

The software development methodology (also known as SDM) framework didn't emerge until the 1960s. According to Elliott (2004) the systems development life cycle (SDLC) can be considered to be the oldest formalized methodology framework for building information systems. The main idea of the SDLC has been "to pursue the development of information systems in a very deliberate, structured and methodical way, requiring each stage of the life cycle－－from inception of the idea to delivery of the final system－－to be carried out rigidly and sequentially"[2] within the context of the framework being applied. The main target of this methodology framework in the 1960s was "to develop large scale functional business systems in an age of large scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines".Methodologies, processes, and frameworks range from specific proscriptive steps that can be used directly by an organization in day-to-day work, to flexible frameworks that an organization uses to generate a custom set of steps tailored to the needs of a specific project or group. In some cases a "sponsor" or "maintenance" organization distributes an official set of documents that describe the process. Specific examples include:

1970s

Structured programming since 1969

Cap Gemini SDM, originally from PANDATA, the first English translation

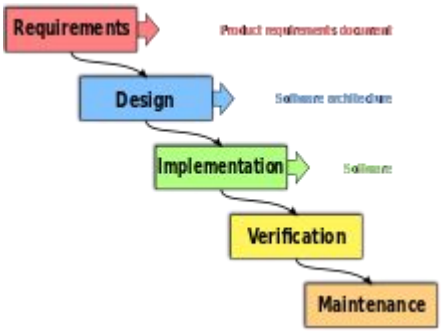
was published in 1974. SDM stands for System Development Methodology
1980s
Structured systems analysis and design method (SSADM) from 1980 onwards
Information Requirement Analysis/Soft systems methodology
1990s
Object-oriented programming (OOP) developed in the early 1960s, and became a dominant programming 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 Overview

In a software development method process workflows are defined for use cases and roles. Use cases are represented as actions having effects, and are stored in an asset repository according to a meta model. Each use case is defined as a collaboration linked to actions, all linked to an associated effect. Refinement relationships link collaborations to actions and define what stage the project life-cycle is at.

4 Methods/Techniques

Waterfall development
Waterfall model



The activities of the software development process represented in the waterfall model. There are several other models to represent this process.

The waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through several phases, typically:

Requirements analysis resulting in a software requirements specification
Software designImplementationTestingIntegration, if there are multiple subsystems

Deployment (or Installation)

Maintenance

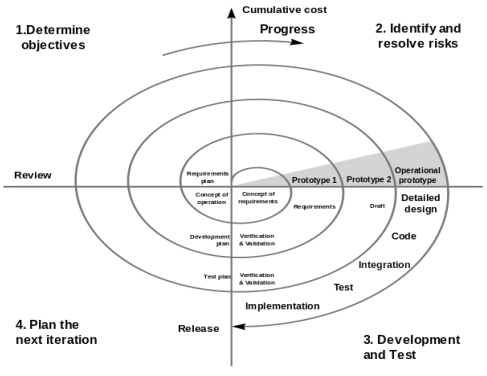
The first formal description of the method is often cited as an article published by Winston W. Royce[3] in

1970 although Royce did not use the term "waterfall" in this article. The basic principles are:[1]

Project is divided into sequential phases, with some overlap and splashback acceptable between phases.Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time.Tight control is maintained over the life of the project via extensive written documentation, formal reviews, and approval/signoff by the user and information technology management occurring at the end of most phases before beginning the next phase. Written documentation is an explicit deliverable of each phase.Iterate:- Create a prototype, then the real system. Repeat each phase using new information gained and the entire process at least once before delivering the live system.The iteration step has largely been omitted in practice due to how the methodology was taught and the perception at the time that it would be too costly for winning US DoD contracts. The waterfall model with no iteration or prototype is known as the 'traditional' engineering approach applied to software engineering, however it is a direct result of requirements for bidding for US military contracts. As a consequence, unless part of the project plan, a strict waterfall approach discourages revisiting and revising any prior phase once it is complete. This "inflexibility" has been a source of criticism by supporters of other more "flexible" models. It has been widely blamed for several large-scale government projects running over budget, over time and sometimes failing to deliver on requirements due to the Big Design Up Front approach. Except when contractually required, the waterfall model has been largely superseded by more flexible and versatile methodologies developed specifically for software development. See Criticism of Waterfall model.

The waterfall model is also commonly taught with the mnemonic A Dance in the Dark Every Monday, representing Analysis, Design, Implementation, Testing, Documentation and Execution, and Maintenance.

Spiral development



Spiral model (Boehm, 1988)

In 1988, Barry Boehm published a formal software system development "spiral model," which 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.

The basic principles are:

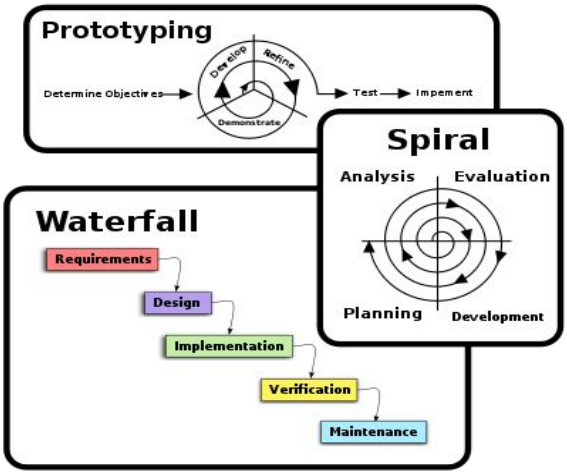
Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments and providing more ease-of-change during the development process, as well as providing the opportunity to evaluate risks and weigh consideration of project continuation throughout the life cycle.

"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."

Each trip around the spiral traverses four basic quadrants: determine objectives, alternatives, and constraints of the iteration; evaluate alternatives; Identify and resolve risks; develop and verify deliverables from the iteration; and plan the next iteration.

Begin each cycle with an identification of stakeholders and their "win conditions", and end each cycle with review and commitment.

5 Conclusion



The three basic approaches applied to software development methodology frameworks.

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.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 environmentis used, and one or more dominant programming paradigms, programming style rules, or choice of specific software libraries orsoftware frameworks. These details are generally not dictated by the choice of model or general methodology.

Acknowledgments

1. 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.

2.Geoffrey Elliott (2004) Global Business Information Technology: an integrated systems approach. Pearson Education. p.87.

3.Wasserfallmodell > Entstehungskontext, Markus Rerych, Institut für Gestaltungs- und Wirkungsforschung, TU-Wien. Accessed on line November 28, 2007.

4.ieeecomputersociety.org

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

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

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

7.Whitten, Jeffrey L.; Lonnie D. Bentley, Kevin C. Dittman. (2003). Systems Analysis and Design Methods. 6th edition. ISBN 0-256-19906-X.

8.Kent Beck, Extreme Programming, 2000.

9.McConnell, Steve. "7: Lifecycle Planning". Rapid Development. Redmond, Washington: Microsoft Press. p. 140.