Software Development Processes (Methods)

朱俣江*
Zhejiang Normal University
Software Project Management

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.

Keywords: software engineering, project management, planning, risk assessment

1 Introduction

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. 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.

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,

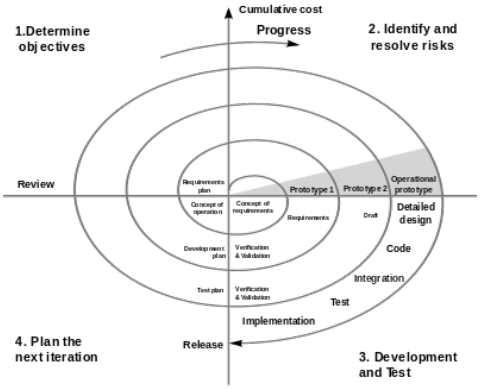

Figure 1: *spiral model - the classic spiral approach to programming.The basic principles are:Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments and providing more ease-of-change during the development process, as well as providing the opportunity to 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: (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. Begin each cycle with an identification of stakeholders and their "win conditions", and end each cycle with review and commitment.*

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".[2]

Methodologies, processes, and frameworks range from specific prescriptive 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

*e-mail:736825232@qq.com

The article is based on work to support Software Project Management.
Software Project Management (2016/17)
Author Name: 朱俣江
University: Zhejiang Normal University
Title: Software Development Processes (Methods)
Supervisor: Dr. Kenwright

Acknowledgements

I would like to thank my colleagues and reviewers for taking time out of their valuable schedule to help make this report more concise and clearer.

References

DAY, R., AND GASTEL, B. 1996. *A Spiral Model of Software Development and Enhancement*. ACM SIGSOFT Software Engineering Notes.

SAKO, Y., AND FUJIMURA, K. 2000. Shape similarity by homotropic deformation. *The Visual Computer 16*, 1, 47–61.