

Software Engineering in Practice

Software engineering models and methods

Diomidis Spinellis
Department of Management Science and Technology
Athens University of Economics and Business

dds@aueb.gr
<http://www.dmst.aueb.gr/dds>
@CoolSWEng

2022-05-23

Assignment (Software engineering models and methods)

- Answer the following for a popular open source project:
 - Which of the software engineering models would be most suitable for modeling this project?
 - Which software engineering method was chosen for its development? You can retrieve information relevant to these tasks from the source code and the project's documentation.
 - If the system started now its development which software engineering method should be chosen?

Software engineering models and methods goals

- Systematization
- Repeatability
- Models provide:
 - An approach to problem solving
 - Notations
 - Procedures for construction and analysis
- Methods support systematic:
 - specifications
 - design
 - construction
 - tests and verification

Overview

- Modeling
- Types of models
- Analysis of models
- Software engineering methods

Modeling Principles

- Model the essentials
- Provide perspective
- Enable effective communications

“A model is an abstraction or simplification of a software component”

Model representation

- A model is represented as an aggregation of
 - Entities
 - Relations
- It can be represented
 - Graphical visualization
 - Textual languages
- Defining the representation
 - Graphical visualization: meta-model
 - For textual languages: BNF

Properties of expression of models

- Completeness
- Consistency
- Correctness

(The first two properties can be automatically checked)

Design by contract

A model can define:

- Preconditions
 - A set of conditions that must be satisfied prior to execution of the function or method.
- Postconditions
 - A set of conditions that is guaranteed to be true after the function or method has executed successfully.
- Invariants

Types of models

- Information modeling, e.g. (ER, UML)
- Structure modeling, i.e.,
 - Class diagram and relationships
 - Object diagram
 - Component diagram

- Deployment diagram
- Behavioral modeling, i.e.,
 - State-chart diagram
 - Activity diagram
 - Sequence diagram
 - Collaboration diagram

Analysis of models

- Completeness
- Consistency
- Correctness
- Traceability
- Interactions

Heuristic methods

- Structured analysis and design methods
- Data modeling methods
- Object-oriented analysis and design methods

Formal methods

- Specification languages
- Formal verification
- Logical inference

Prototyping methods

- Styles
 - Paper product
 - Executable specification
 - Throwaway code
 - Evolution of a working design
- Target
 - Requirements specifications
 - Architectural design
 - Algorithm
 - Human-machine user interface

Agile methods

- Rapid Application Development (RAD)
- Extreme programming (XP)
- Scrum

- Feature-Driven Development (FDD)

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile development principle 1

“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”

Agile development principle 2

“Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.”

Agile development principle 3

“Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”

Agile development principle 4

“Business people and developers must work together daily throughout the project.”

Agile development principle 5

“Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”

Agile development principle 6

“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”

Agile development principle 7

“Working software is the primary measure of progress.”

Agile development principle 8

“Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”

Agile development principle 9

“Continuous attention to technical excellence and good design enhances agility.”

Agile development principle 10

“Simplicity—the art of maximizing the amount of work not done—is essential.”

Agile development principle 11

“The best architectures, requirements, and designs emerge from self-organizing teams.”

Agile development principle 12

“At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

Extreme programming characteristics

- Pair programming
- Tests first
- Continuous code refactoring and integration
- Incremental system development
- Direct customer involvement in the team

Scrum’s characteristics

- A Scrum master manages the activities
- Sprint < 30 days
- Product Backlog Item
 - Cost estimation
 - Assignment
- Daily Scrum meetings

Feature-Driven Development (FDD) process

- Develop a product model to scope the breadth of the domain

- Create the list of needs or features
- Build the feature development plan
- Develop designs for iteration-specific features
- Per feature:
- Code
- Test
- Integrate

Preparation for the next lecture (1)

- Study Chapter 10 from SWEBOK v 3.0
- Assignment (Software quality)

Answer the following for a popular open source project:

- How do they perform quality assurance on the system under development?
- Which factors and quality characteristics are most important for the software quality requirements?
- How do they measure software quality?

You can retrieve relevant information from the source code and the documentation of the project. In the case that the project misses some of the aforementioned information, how do you think that it should have recorded it?

Preparation for the next lecture (2)

- Video (Software quality: Software quality assurance) <https://www.youtube.com/watch?v=hMfPCdF07hA8>

Distribution License

Unless otherwise expressly stated, all original material on this page created by Diomidis Spinellis, Marios Fragkoulis, and Antonis Gkortzis is licensed under the Creative Commons Attribution-Share Alike Greece.

