Softwaretechnik / Software-Engineering

Lecture 4: Software Project Management

2016-05-02

Prof. Dr. Andreas Podelski, Dr. Bernd Westphal
Albert-Ludwigs-Universität Freiburg, Germany

Topic Area Project Management: Content

Vocabulary: Project

• Temporary activity that is characterized by:
  • a start date,
  • specific objectives and constraints,
  • established responsibilities,
  • a budget and schedule, and
  • a completion date.

If the objective of the project is to develop a software system, then it is sometimes called a software development project or software engineering project.

R. H. Thayer (1997)

We could refine our earlier definition as follows: a project is successful if and only if:
  • started at start date,
  • achieved objectives,
  • respected constraints,
  • adheres to budget and schedule,
  • stops at completion date.

Whether, e.g., objectives have been achieved can still be subjective (→ customer/user happy).

Vocabulary: Software Project

• Characteristics:
  • Duration is limited.
  • Has an originator (person or institution which initiated the project).
  • The project owner is the originator or its representative.
  • The project leader reports to the project owner.
  • Has a purpose, i.e. pursue a bunch of goals.
  • The most important goal is usually to create or modify software; this software is thus the result of the project, the product.
  • Other important goals are extension of know-how, preparation of building blocks for later projects, or utilisation of employees.
  • The project is called successful if the goals are reached to a high degree.
  • Has a recipient (or will have one).
  • This recipient is the customer.
  • Later users (conceptionally) belong to the customer.
  • The project links people, results (intermediate/final products), and resources.

Ludewig & Lichter (2013)

Developer

Customer

User
Goals and Activities of Project Management

• Main and general goal: a successful project, i.e., the project delivers:
  • defined results
  • in demanded quality
  • within scheduled time
  • using the assigned resources.
There may be secondary goals, e.g.,
  • build or strengthen good reputation on market,
  • acquire knowledge which is useful for later projects,
  • develop re-usable components (to save resources later),
  • be attractive to employees.

Main project management activities (and responsibilities of project manager):
  • Planning
  • Assessment and Control
  • Recognising and Fighting Difficulties as Early as Possible
  • Communication
  • Leading and Motivation of Employees
  • Creation and Preservation of Beneficial Conditions

Without plans, a project cannot be managed. Note: mistakes in planning can be hard to resolve.

Work results and project progress have to be assessed and compared to the plans; it has to be observed whether participants stick to agreements.

Unforeseen difficulties and problems in projects are not exceptional but usual. Therefore, project management needs to constantly "screen the horizon for icebergs," and, when spotting one, react timely and effectively. In other words: systematic risk management.

Distribute information between project participants (project owner, customer, developers, administration).

Leading means: going ahead, showing the way, "pulling" the group. Most developers want to achieve good results, yet need orientation and feedback (negative and positive).

Provide necessary infrastructure and working conditions for developers (against: demanding customers, imprecisely stated goals, organisational restructuring, economic measures, tight office space, other projects, etc.).

Quick Excursion: Risk and Risk Value

Risk — a problem, which did not occur yet, but on occurrence threatens important project goals or results. Whether it will occur, cannot be surely predicted.

Ludewig & Lichter (2013)

Risk value \( p \cdot K \)

\( p \): probability of problem occurrence,
\( K \): cost in case of problem occurrence.

• Acceptable risks
• Unacceptable risks
• Extreme risks

Software Project Planning

• Avionics requires: "Average Probability per Flight Hour for Catastrophic Failure Conditions of \( 10^{-9} \) or 'Extremely Improbable'" (AC 25.1309-1).

• "Problems with \( p = 0.5 \) are not risks, but environment conditions to be dealt with."
Planning and managing software projects involves:

- Costs and deadlines
- Tasks and activities
- People and roles

A phase is a continuous, i.e. not interrupted range of time in which certain works are carried out and completed. At the end of each phase, there is a milestone.

A phase is successfully completed if the criteria defined by the milestone are satisfied.

Ludewig & Lichter (2013)

- Phases (in this sense) do not overlap!
- Yet there may be different “threads of development” running in parallel, structured by different milestones.
- Splitting a project into phases makes controlling easier;
- Milestones may involve the customer (accept intermediate results) and trigger payments.
- The granularity of the phase structuring is critical:
  - Very short phases may not be tolerated by a customer,
  - Very long phases may mask significant delays longer than necessary.
- If necessary: define internal (customer not involved) and external (customer involved) milestones.

A phase is a continuous, i.e. not interrupted range of time in which certain works are carried out and completed. At the end of each phase, there is a milestone.

A phase is successfully completed if the criteria defined by the milestone are satisfied.

Ludewig & Lichter (2013)

- Whether a milestone is reached (or successfully completed) must be assessable by
  - Clear, objective, and unambiguous criteria.
- The definition of a milestone often comprises:
  - A definition of the results which need to be achieved,
  - The required quality properties of these results,
  - The desired time for reaching the milestone (the deadline), and
  - The instance (person or committee) which decides whether the milestone is reached.
- Milestones can be part of the development contract; not reaching a defined milestone as planned can lead to legal claims.

Cycle and Life Cycle

- Cycle — (1) A period of time during which a set of events is completed. See also:
- System life cycle — The period of time that begins when a system is conceived and ends when it is no longer available for use.
- Software life cycle — The period of time that begins when a software product is conceived and ends when the software is no longer available for use.
- Software development cycle — The period of time that begins with the decision to develop a software product and ends when the software is delivered.

IEEE 610.12 (1990)
The Concept of Roles

Useful and Common Roles

The Concept of Roles Cont'd

Common Activities in Order to Develop or Adapt Software

Was on Time and Finished?

Dismissing and Replacement

For example

Note

Most software systems (sooner or later) become obsolete, and are often replaced by a successor system. Common reasons: existing system no longer maintainable, not adaptable to new or changed requirements.

Implementation

Design, Specification of Modules

Most software systems consist of modules or components which interact to realize the overall functionality (antonym: monolithic). Design overall structure (called software architecture). The period of time that begins with the decision to develop a software product and ends when the software is available for use.

Integration, Testing, Approval

The period of time that begins with the design phase and ends when the software is delivered. This cycle typically includes the requirements phase, development phase, implementation phase, testing phase, and installation and checkout phase. Depending upon the software development approach used, the phases listed above may overlap or be performed iteratively.

Requirements

The period of time that begins with the design phase and ends when the software is ready for integration. The problem, assess whether/in how far software can be used to solve it. Sort out, document, assess, extend, correct. . . the results of analysis. Resulting documents are basis of most other activities!

Analysis

Formal methods (1) This term is sometimes used to mean a longer period of time, either the period that ends when the phases listed above may overlap or be performed iteratively, or the period of time that begins with the design phase and ends when the software is ready for integration. Note that in a software project, the terms design phase and requirements phase sometimes overlap.

Formal methods (2) This term is sometimes used to mean a longer period of time, either the period that ends when the phases listed above may overlap or be performed iteratively, or the period of time that begins with the design phase and ends when the software is ready for integration. Note that in a software project, the terms design phase and requirements phase sometimes overlap.
Process is Product Methadology

From Building Blocks to Process (and Back)

From the Plan for Creating to the Process (and Back)
Process Description and Reference Model

- A process description is a documented expression of a set of activities performed to achieve a given purpose.
- A process description provides an operational definition of the main components of a process.
- The purpose of a process is to achieve a particular goal.
- The process description describes how the goal is achieved.
- A process description is flexible and can be adapted to meet the changing needs of the organization.
- A process description should be clear and concise.
- A process description should be reviewed and updated regularly.

Procedure Model (?!): Code and Fix

- A procedure model (procedure model) comprises:
  - Procedure model (methodology)
    - e.g., waterfall model (1970s)
  - Organizational model (organizational structure)
    - e.g., V-Model (RUP, SPICE, COCOMO)

- In the literature, process model and procedure model are often used as synonyms.
- There is no clear agreement on the relationships between the processes.

Anticipated Benefits of Process Models

- "Economy of thought"—fewer "I thought I did that already" moments.
- "Qualification, reusability"—same skills can be brought to bear.
- "Correct functionality"—a quality assurance component.
- "Feasible errors"—e.g., finding a bug is not too difficult because the "easy" methods prevent developers from being out of form.
- "Clear responsibilities"—fewer "I thought it was their job!"

Process vs. Procedure Model

- A process model (process model) comprises:
  - Procedure model (methodology)
    - e.g., waterfall model (1970s)
  - Organizational model (organizational structure)
    - e.g., V-Model (RUP, SPICE, COCOMO)

- Important
  - Maintenance
  - Operation and testing

• Resulting programs often lack expected outcome
• Tests are acceptance testing, installation and testing
• If requirements are not stated, there is no rational/explicit decision
• Corresponds to our desire to "get ahead," to solve the stated problem quickly
• Conducted activities of software development

Testing, evaluation, 2010 prepare to distinguish process model and procedure model.

- Procedure Models
  - Code and Fix—directly an approach, where coding and correction alternating, with all hierarchy and the only consistency, constructed within phases of software development.

Anticipated Benefits of Process Models

- "Economy of thought"—fewer "I thought I did that already" moments.
- "Qualification, reusability"—same skills can be brought to bear.
- "Correct functionality"—a quality assurance component.
- "Feasible errors"—e.g., finding a bug is not too difficult because the "easy" methods prevent developers from being out of form.
- "Clear responsibilities"—fewer "I thought it was their job!"

The (In)famous Waterfall Model (Roseve, 1967)


