Topic Area: Project Management

- VL 2: Software Metrics
  - Properties of Metrics
  - Scales
  - Examples
- VL 3: Cost Estimation
  - "(Software) Economics in a Nutshell"
  - Expert’s Estimation
  - Algorithmic Estimation
- VL 4: Project Management
  - Project Process and Process Modelling
    - Procedure Models
    - Process Models
    - Process Metrics
    - CMMI, SPICE
- VL 5: ...

Vocabulary: Project

- Project: A 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

- (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.
  - Links people, results (intermediate/final products), and resources.
  - The organisation determines roles of and relations between people/results/resources, and the external interfaces of the project.

Lüdewig & Lichter (2013)
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:

1. Acceptable risks
   - Probabilities of occurrence are low and the consequences are not highly significant.
   - Examples: minor deviations in project schedules, small equipment malfunctions.

2. Risk management
   - Risks are identified, assessed, and controlled.
   - Strategies: prevention, mitigation, contingency planning.

3. Planning
   - Development of project plans.
   - Importance: ensures project objectives are met.

4. Assessment and Control
   - Continuous evaluation of project risks.
   - Adjustments made to plans as needed.

5. Leading and Motivation of Employees
   - Effective leadership.
   - Motivation and engagement of team members.

6. Communication
   - Clear and open communication.
   - Facilitates coordination and understanding.

7. Creation and Preservation of Beneficial Conditions
   - Favorable working environment.
   - Supportive and collaborative team dynamics.

8. Recognition and Fighting Difficulties as Early as Possible
   - Identification and mitigation of challenges.
   - Quick reactions to emerging issues.

9. Evaluation and Control
   - Continuous assessment of project performance.
   - Adjustment of strategies as necessary.

10. Leading and Motivation of Employees
    - Effective leadership.
    - Motivation and engagement of team members.

11. Communication
    - Clear and open communication.
    - Facilitates coordination and understanding.

12. Creation and Preservation of Beneficial Conditions
    - Favorable working environment.
    - Supportive and collaborative team dynamics.

13. Recognition and Fighting Difficulties as Early as Possible
    - Identification and mitigation of challenges.
    - Quick reactions to emerging issues.

14. Evaluation and Control
    - Continuous assessment of project performance.
    - Adjustment of strategies as necessary.
Planning and managing software projects involves:

- Costs and deadlines, (→ phase, milestone, deadline)
- Tasks and activities,
- People and roles.

**Phase**

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.

**Milestone, Deadline**

Planning and managing software projects involves:

- Costs and deadlines, (→ phase, milestone, deadline)
- Tasks and activities,
- People and roles.
### Procedure and Process Models

- **norm/standard supervisory committee**
  - legislator

- **systems administrator**
  - analyst
  - tester

- **cycle**, **life cycle**, **software life cycle**

- **process vs. process model**
  - multiple persons, one role
  - one person, one role
  - one person, multiple roles

### Software Development Process

- **ana**
  - **tst**
  - **mgr**

- **Common Responsibilities and Rights**

- **Roles and Responsibilities**

- **Excursion: Risk**

- **Customer Developer**

- **Project Management**

- **Software Project Planning**

- **Software Project Planning**

- **Common Activities**

- **Common Activities in Order to Develop or Adapt Software**

- **Costs and Deadlines**

- **Excursion: Risk**

- **Sanity Check**

- **Software is developed to solve a problem or satisfy a need.**

- **Towards the Implementation**

- **The Concept of Roles**
Procedure and Process Models

- cycle, life cycle, software life cycle
- Software Development Process
  - describing the relationships between the processes.

NOTE: A process description provides the description specifies, in a manner, the requirements, design, complete, precise, and verifiable behavior, or other characteristics of a process. It also may include whether these provisions have been satisfied.

Process descriptions can be found at the activity, project, or organizational level. People and Roles

- Tasks and Activities
- Costs and Deadlines

IEEE 24765 (Software) Project Content

- Goals and Activities
- Project Management
- Buildings Blocks
  - From Building Blocks to the Process Model
Product vs. Procedure Model

Before introducing a process model...
The (In)famous Waterfall Model (Rosove, 1967)

Waterfall or Document-Model

— Software development is seen as a sequence of activities coupled by (partial) results (documents). These activities can be conducted concurrently or iteratively. Apart from that, the sequence of activities is fixed as (basically) analyse, specify, design, code, test, install, maintain.

Ludewig & Lichter (2013)

system analysis
software specification
architecture
design
refined design and coding
integration and testing
installation and acceptance
operation and maintenance

The Waterfall Model: Discussion

• The waterfall model has been subject of heated discussions:
  • Original model without feedback not realistic.
  • Gives room for many interpretations; very abstract; hardly usable as a "template" for planning real projects.
  • Cycles (and the lack of milestones) makes it hard for project management to assess a project's process.

• Maybe best appreciated in the context of its time: "Dear people (of the 60's), there is more in software development than coding; and there are (obvious) dependencies. That may have been news to some software people back then. . . (cf. "software crisis").

• Everybody knows it (at least the name. . . ).

Tell Them What You’ve Told Them...


