

Topic Area Project Management: Content

- VL2 • Software Metrics
 - ↳ Properties of Metrics
 - ↳ Scales
 - ↳ Examples
- VL3 • Cost Estimation
 - ↳ "Software Economics in a Nutshell"
 - ↳ Experts Estimation
 - ↳ Algorithmic Estimation
- VL4 • Project Management
 - ↳ Project
 - ↳ Process and Process Modelling
 - ↳ Procedure Models
 - ↳ Process Models
- VL5 • Process Metrics
 - ↳ CMMI Spec
 - ↳ ...

Content

- (Software) Project
 - ↳ Project Management
 - ↳ Goals and Activities
 - ↳ Common Activities
 - ↳ Excursion: Risk
- Software Project Planning
 - ↳ Costs and Deadlines
 - ↳ plan: milestone deadline
 - ↳ Tasks and Activities
 - ↳ People and Roles
 - ↳ responsibilities and rights
- Software Development Process
 - ↳ process vs. process model
 - ↳ cycle: life cycle, software the cycle
 - ↳ Procedure and Process Models

Project

Vocabulary: Project

project - A temporary activity that is characterized by having

- 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** (cf. <http://www.it-ebooks.info>)

- 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 bundle of goals
 - ↳ The most important goal is usually to create or modify software.
 - ↳ Other important goals are acquisition of know-how, preparation of building blocks for other 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(s) belong(s) to the customer
 - ↳ Later user (conceptually) belong(s) 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 interests of the project.

Ludewig & Luber (2013)



Project Management

7/17

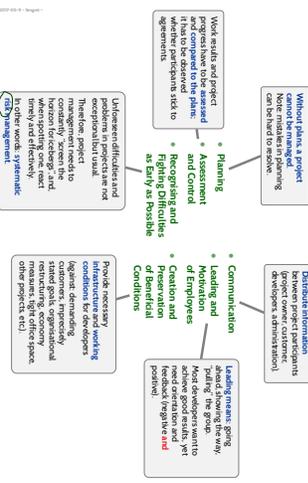
Goals and Activities of Project Management

- **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 and a wealth for later projects
 - develop new competences for later product development
 - better prepare to **surprises**
- **Main project management activities (and responsibilities of project manager):**
 1. Planning
 - Assessment and Control
 - Recognising and fighting Difficulties as Early as Possible
 2. Communication
 - Leading and Motivation of Employees
 - Creation and Preservation of Beneficial Conditions



8/17

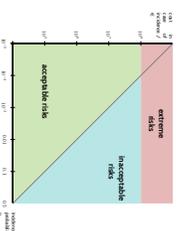
Activities of Project Management



9/17

Quick Excursion: Risk and Riskvalue

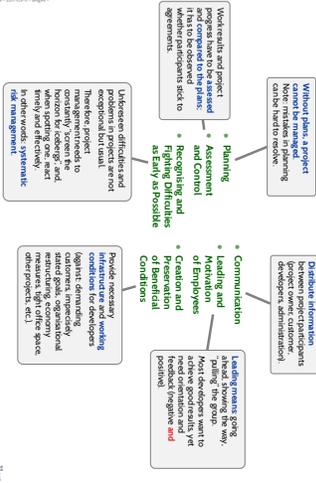
risk = a problem which did not occur yet, but an occurrence threatens important project goals or results. Whether it will occur cannot be surely predicted. (Ludwig & Aldner (2013))



- riskvalue = $p \cdot K$
- p : probability of problem occurrence.
- K : cost in case of problem occurrence.
- **Advises require:** "Average Probability per Flight Hour" for "Catastrophic Failure Conditions of 10⁻⁷" or "Extremely Improbable" (AC 25.1309-4)
- problems with $p = 0.5$ are not risks, but environment conditions to be dealt with"

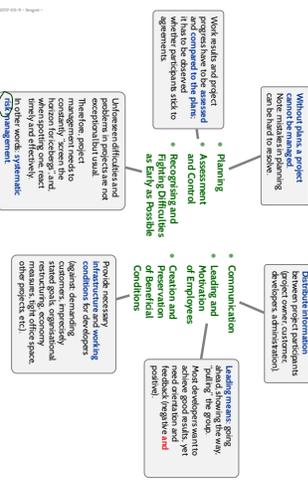
10/17

Activities of Project Management



11/17

Activities of Project Management



9/17

Project Management

Software Engineering as defensive discipline.

Analogy: safety belt.

"Our patients were working hard to avoid accidents, though soon were working to get me well again."

"Software Engineering is **proactive** and for people who do not value the difference of failures as a positive achievement!"

04.01.2008 17:49

12/17

- (Software) Project
- Project Management
 - Goals and Activities
 - Common Activities
 - Extension Risk
- Software Project Planning
 - Costs and Deadlines
 - phase milestone deadline
 - Tasks and Activities
 - People and Roles
 - responsibilities and rights
- Software Development Process
 - process vs. process model
 - cycle: the cycle, software life cycle
- Procedure and Process Models

13/17

Software Project Planning

- What to (Plan and) Manage?
- Planning and managing software project involves
- costs and deadlines
 - tasks and activities, (→ phase, milestone, deadline)
 - people and roles.

15/17

Phases, 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 & Ludewig (2013)

Phases (in this context) are not overlaid! They may differ in terms 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!** internal (customer not involved) and external (customer involved) milestones define internal (customer not involved) and external (customer involved) milestones.

16/17

Milestones, Deadlines

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 & Ludewig (2013)

- Whether a milestone is reached (or successfully completed) must be assessable by
 - objective criteria, 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 defined time for reaching the milestone (the deadline) and
 - the assigned 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.

17/17

What to (Plan and) Manage?

- Planning and managing software projects involves
- costs and deadlines
 - tasks and activities, (→ phase, milestone, deadline)
 - people and roles.

18/17

Software Development Process

Process –

- (1) A sequence of steps performed for a given purpose
- (2) See also: [task](#), [software development process](#)
- (3) To perform operations on data

IEEE 6012 (1970)

Software Development Process –

The process by which user needs are translated into a software product. The process involves translating user needs and software requirements into a software product. The process involves the design, coding, testing, and implementation of the software. The process involves the design, coding, testing, and implementation of the software for operational use.

IEEE 6012 (1970)

- The process of a software development project may be
 - implicit.
 - informally agreed on, or
 - explicitly prescribed (by a procedure or process model)
- Note each software development project has a process!

25/17

Process

Process –

- (1) A sequence of steps performed for a given purpose
- (2) See also: [task](#), [software development process](#)
- (3) To perform operations on data

IEEE 6012 (1970)

Software Development Process –

The process by which user needs are translated into a software product. The process involves translating user needs and software requirements into a software product. The process involves the design, coding, testing, and implementation of the software. The process involves the design, coding, testing, and implementation of the software for operational use.

IEEE 6012 (1970)

- The process of a software development project may be
 - implicit.
 - informally agreed on, or
 - explicitly prescribed (by a procedure or process model)
- Note each software development project has a process!

26/17

Describing Software Development Processes

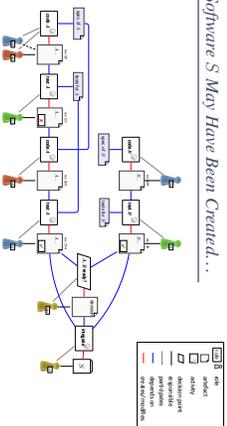
Over time, the following notions proved useful to describe and model (→ in a model) software development processes:

- **role** – has responsibilities and rights; needs skills and capabilities. In particular, responsibility for artifacts; participates in activities.
- **artifact** – all documents, evaluation protocols, software modules, etc. all products emerging during a development process. Is processed by activities; may have state.
- **activity** – any processing of artifacts, manually or automatic. Depends on artifacts; creates/modifies artifacts.
- **decisionpoint** – special case of activity; a decision is made based on **artefacts** in a certain state. Creates artifacts.
- **Delimits phases; corresponds to milestones.**



27/17

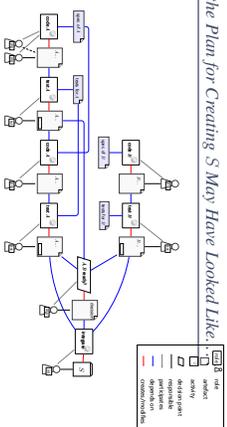
How Software S May Have Been Created...



- S consists of modules A and B.
- Assume specifications and test cases for A and B were available.
- Person coded B (according to spec), then person tested B with test cases, no errors found.
- Person coded A, with the help of person B, then person tested A, some errors found.
- Person tested A, tested again, no errors found.
- A and B ready caused positive decision, then person regraded A and B and obtained S.

28/17

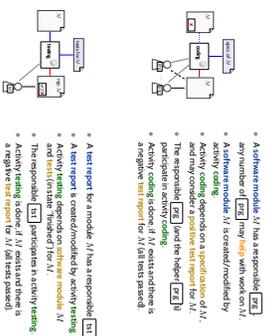
How the Plan for Creating S May Have Looked Like...



- S consisted modules A and B; specifications and test cases for A and B are available.
- Some coded B (according to spec), then some tested B with test cases, and creates test report.
- Some coded A, with the help of some B, then some tested A, and creates test report.
- If errors in A found, some retested A, some retested B, and creates test report.
- If A and B ready causes a positive decision, then some regrades A and B and obtains S.

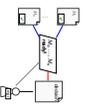
29/17

How the Plan for Creating S May Have Been Created...

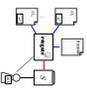


- A software module A has a responsible [A1] any number of [A2] may work on it.
- A software module A is created/modified by activity coding.
- Activity coding depends on a specification of A.
- The responsibility [A1] and the help of [A2] B participate in activity coding.
- Activity coding is done if A exists and there is a negative test report for A (all tests passed).
- A test report for a module A has a responsible [A1].
- A test report is created/modified by activity testing.
- Activity testing depends on software module A.
- The responsibility [A1] participate in activity testing.
- Activity testing is done if A exists and there is a negative test report for A (all tests passed).

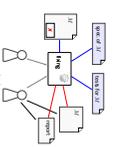
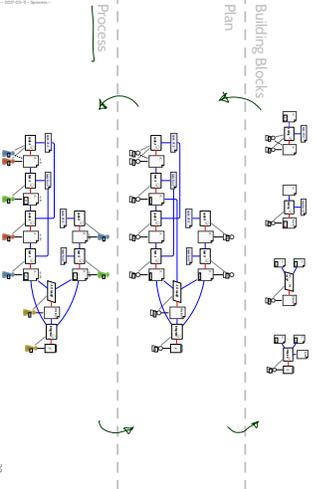
30/17



- A **key decision** for a module M_1, \dots, M_n , has a responsible person
- A **key decision** is created/modified (decision point ready)?
- Decision point ready depends on negative **test report** for M_1, \dots, M_n .
- The responsible [] participates in decision point ready?
- Decision point ready is done if a positive decision exists



- A software S has a responsible []
- S is created by integrating modules M_1, \dots, M_n .
- A software is created/modified by **activity integration**.
- **Activity integration** depends on **software modules**
- The responsible [] participates activity integrate
- **Activity integration** is done, if S exists



- Example: Distinguish coding and fixing software.
- If there is a negative test result for M_i .
- A [] responsible for fixing M_i .
- The responsible [] who was responsible for the fixing depends on the test cases in addition to the specification of M_i .
- a report (analysis of the error documentation) that will be written.

- (Software) Project
- Project Management
 - Goals and Activities
 - Common Activities
 - Excursion: Risk
- Software Project Planning
 - phase: milestone, deadline
 - Tasks and Activities
 - People and Roles
 - responsibilities and rights
- Software Development Process
 - process vs. process model
 - cycle, life cycle, software life cycle
- Procedure and Process Models

Process vs. Procedure Models

Process Description and Reference Model

process description – documented expression of a set of activities performed to achieve a given purpose.
 NOTE: A process description provides an operational definition of the major components of a process.
 process description provides a template, goals, and verifiable evidence the requirements, design behavior, or other characteristics of a process.
 It also may include procedures for determining whether these provisions have been satisfied.
 Process descriptions can be found at the activity, project, or organizational level.
IEEE 7195 (2019)

process reference model – a model comprising definitions of processes in a life cycle described in terms of process purpose and outcomes, together with an architecture describing the relationships between the processes.
IEEE 7195 (2019)

cycle - (I) A period of time during which a set of events is completed []
IEEE 6021(1999)

system life cycle - The period of time that begins when a system is conceived and ends when it is no longer available for use.
IEEE 6021(1999)

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. []
IEEE 6021(1999)

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 6021(1999)

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.
The software life cycle typically includes

- a concept phase,
- a development and design phase,
- a testing phase,
- an operation and maintenance phase, and
- an implementation phase.

Note: These phases may overlap or be performed iteratively.
IEEE 6021(1999)

software development cycle - The period of time that begins with the decision to develop a software product and ends when the software is delivered.
This cycle typically includes

- a requirements phase,
- a test phase, and
- a development phase.

Note: (I) The phases listed above may overlap or be performed iteratively, depending upon the software development approach used.
(2) This term is sometimes used to mean a longer period of time, either the period that ends when the software is no longer being enhanced by the developer, or the entire product life cycle.
IEEE 6021(1999)

(Ludwig and Lichten, 2013) propose to distinguish: process model and procedure model.

• A Process model (Prozessmodell) comprises

(I) Procedure model (Vorgangmodell)

e.g., waterfall model (V08/80).

(II) Organisational structure = comprising requirements on

- project management and responsibilities,
- quality assurance,
- documentation, document structure,
- revision control.

e.g., V-Model, RUP, SP 900/001.

• In the iterative process model and procedure model are often used as synonyms: there is no universally agreed definition.

Anticipated Benefits of Process Models

- "economy of thought"
 - > deontic-ethical principles
 - > quantification, reproducibility
 - > one can assess the quality of how products are created (-> CMM)
- Identify weaknesses, learn from bad experience, improve the process
- Fewer errors
 - > e.g., testing module cannot be forgotten because the ready-decision point depends on module with test passed/flagged.
 - > clear responsibilities
 - > fewer "thought" proof for the module!"
- Process modelling is easily overdone - the best process model is worthless if your software people don't use it.
 - Before introducing a process model
 - understand what you have, understand what you need
 - process model as much as needed, not more (-> warning)
 - process model from the perspective of process model makes things better or worse (-> metrics)
 - Note: customer may require a certain process model.

Procedure Models

Warning: The following text is a warning or note.

Procedure Model (?): Code and Fix

Code and Fix - denotes an approach, where coding and correction alternating with ad-hoc tests or the only conditionally conducted activities of software development.
Ludwig & Lichten (2013)

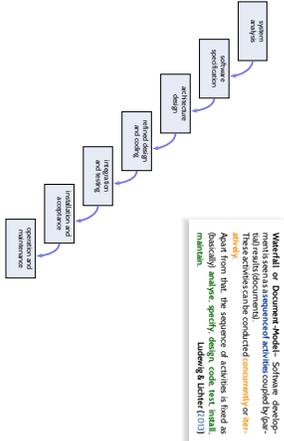
Advantages:

- Corresponds to our desire to "get ahead" to solve the said problem quickly.
- The conducted activities (coding and ad-hoc testing) are easy.

Disadvantages:

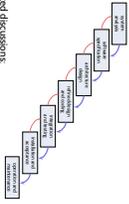
- It's hard to plan the project, there are no rational system decisions.
- It's hard to distribute work or to do performance reports.
- It's hard to do retrospectives (e.g., meeting requirements).
- Feature backlog expected outcome (outcome, e.g., derived from requirements).
- Realizing programs often hard to maintain.
- Effort for maintenance high, most errors are easily detected in operation.
- Important concepts and decisions are not documented, but only in the heads of the developers, thus hard to transfer.

...



Waterfall or Document Model - Software development is a sequential sequence of activities coupled by gates. 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*. Ludwig & Ullmer (2013)

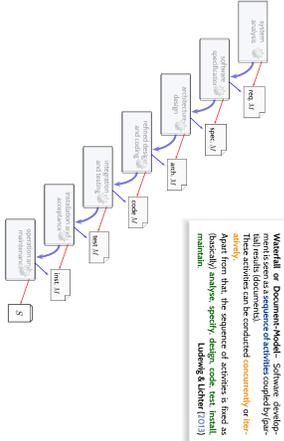
The Waterfall Model: Discussion



- The waterfall model has been subject of heated discussions
- Original model without feedback not realistic
- Great source for many interpretations, very abstract
- Can be used to describe the software development process, but not the actual development process
- Conducted the lack of milestones/intermediate hard for project management to assess a project's progress

Maybe best appreciated in the context of its time and there are other (optional) dependencies and there are other (optional) dependencies

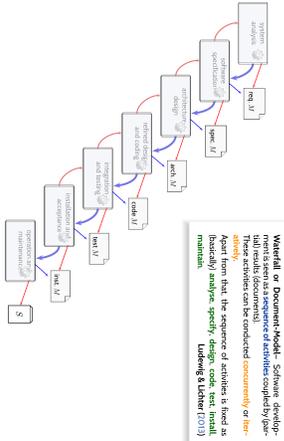
Everybody knows it (at least the name...)



Waterfall or Document Model - Software development is a sequential sequence of activities coupled by gates. 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*. Ludwig & Ullmer (2013)

Tell Them What You've Told Them...

- **Project** has (among others)
 - project owner, project leader
 - goals (Escursion Risk)
 - process - each project has one
- **processes** can be modelled
 - descriptive ('we did like that', or 'prescriptive' (please to it like that))
 - A process model includes
 - roles, artifacts, activities, decision points
 - relations: *responsibility, dependency, creation/modification*
 - A process model can allow us to (→ exercises)
 - derive a schedule (who does what, when)
 - estimate and control phases and deadlines.
- Distinguish procedure model and process model.
- Example: The Waterfall procedure model.



Waterfall or Document Model - Software development is a sequential sequence of activities coupled by gates. 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*. Ludwig & Ullmer (2013)

References

References

- IEEE (1990). *IEEE Standard Glossary of Software Engineering Terminology*. Std 610.12-1990.
ISO/IEC/IEEE (2016). *Systems and software engineering - Vocabulary*. 24745.001/EI.
Ludewig, J. and Licker, H. (2013). *Software Engineering*. dpunkt.verlag, 3. edition.
Rouse, P. E. (1987). *Developing Computer-based Information Systems*. John Wiley and Sons.
Thayer, R. H. (1997). *Inteval - Software Engineering Project Management*. IEEE Society Press, revised edition.