Lecture 5: Procedure & Process Models
Ludewig and Lichter (2013) propose to distinguish:

- **Process Model** ('Prozessmodell') comprises
  - (i) **Procedure Model** ('Vorgehensmodell') e.g., "waterfall model" (70s/80s).
  - (ii) Organisational structure— comprising requirements on
    - project management and responsibilities,
    - quality assurance,
    - documentation, document structure,
    - revision control.

  e.g., V-Modell, RUP, XP (90s/00s).

- In the literature, **process model** and **procedure model** are often used as synonyms; there is not universally agreed distinction.

---

**Procedure Models** — **Waterfall** —

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.

---

**Procedure Models** — **Spiral** —

The Illustrious Spiral Model (R. DeMarco, 1978)

Procedure Models

...
The Spiral Model (Boehm, 1988)

Recall:
Quick Excursion: Risk and Riskvalue

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.

Riskvalue = \( p \cdot K \)

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

Idea of Spiral Model: do not plan ahead everything, but go step-by-step. Repeat until end of project (successful completion or failure):

1. Determine the set \( R \) of risks which are threatening the project; if \( R = \emptyset \), the project is successfully completed
2. Assign each risk \( r \in R \) a risk value \( v(r) \)
3. For the risk \( r_0 \) with the highest risk value, \( r_0 = \max\{v(r) | r \in R\} \), find a way to eliminate this risk, and go this way; if there is no way to eliminate the risk, stop with project failure

Advantages:
• We know early if the project goal is unreachable.
• Knowing that the biggest risks are eliminated gives a good feeling.

Wait, Where's the Spiral?

A concrete process using the Spiral Model could look as follows:

- investigate goals, alternatives, side conditions
- conduct risk analysis,
- develop and test the next product part,
- plan the next phase,

Procedure and Process Models
• Procedure Model Examples
  • The (in)famous Waterfall model
  • The famous Spiral model
• Procedure classification
  • linear / non-linear
  • prototyping
  • evolutionary, iterative, incremental
• From Procedure to Process Models
  • Process Model Examples
    • Phase Model
    • V-Modell XT
    • Agile
      • Extreme Programming
      • Scrum
• Process Metrics
  • CMMI, Spice

Procedure Model Classification
— Linear vs. Non-Linear —
Linear vs. Non-Linear Procedure Models

• Linear: the strict Waterfall Model (no feedback)
• Non-Linear: basically everything else (with feedback between activities)

Procedure Model Classification — By Treatment of Artefacts

• Prototyping:
  - Req. prototype
  - Prototypes: a preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system.
  - IEEE 610.12 (1990)
  - (Rapid) Prototyping
  - Prototyping: a hardware and software development technique in which a preliminary version of part or all of the hardware or software is developed to permit user feedback, determine feasibility, or investigate timing or other issues in support of the development process.
  - IEEE 610.12 (1990)

• Evolutionary Development:
  - Req. evolution
  - Evolutionary software development: an approach which includes evolutions of the developed software under the influence of practical/field testing. New and changed requirements are considered by developing the software in sequential steps of evolution.

• Iterative Development:
  - Req. plan
  - Iterative software development: software is developed in multiple iterative steps, all of them planned and controlled. Goal: each iterative step, beginning with the second, corrects and improves the existing system based on defects detected during usage. Each iterative steps includes the characteristic activities analyse, design, code, test.
  - Ludewig & Lichter (2013)
Classification By Treatment of (Software) Artefacts

• Prototyping:
  - req.
  - prototype
  - results
  - develop

• Evolutionary Development:
  - req.
  - evolution
  - I
  - . . .
  - I
  - n−1
  - I
  - n

• Iterative Development:
  - req.
  - plan
  - spec.
  - iteration
  - . . .
  - iteration
  - n−1
  - iteration
  - n

• Incremental Development:
  - req.
  - project
  - S
  - 1
  - . . .
  - req.
  - n
  - project
  - S
  - n

Incremental development — The total extension of a system under development remains open; it is realised in stages of expansion. The first stage is the core system. Each stage of expansion extends the existing system and is subject to a separate project. Providing a new stage of expansion typically includes (as with iterative development) an improvement of the old components.

Ludewig & Lichter (2013)

• Note: (to maximise confusion) IEEE calls our "iterative" incremental development — A software development technique in which requirements definition, design, implementation, and testing occur in an overlapping, iterative (rather than sequential) manner, resulting in incremental completion of the overall software product.

IEEE 610.12 (1990)

• One difference (in our definitions):
  - • iterative: steps towards fixed goal,
  - • incremental: goal extended for each step; next step goals may already be planned.

Examples: operating system releases, short time-to-market (→ continuous integration).
A process model may describe:

- steps to be conducted during development, their sequential arrangement, their dependencies (the procedure model)
- organisation, responsibilities, roles
- structure and properties of documents
- methods to be used, e.g., for gathering requirements or checking intermediate results
- project phases, milestones, testing criteria
- notations and languages
- tools to be used (in particular for project management).

Process models typically come with their own terminology (to maximise confusion?), e.g. what we call artefact is called product in V-Model terminology.

### Procedure and Process Models

- Procedure Model Examples
  - The (in)famous Waterfall model
  - The famous Spiral model

- Procedure classification
  - linear / non-linear
  - prototyping
  - evolutionary, iterative, incremental

### From Procedure to Process Models

- Process Model Examples
  - Phase Model
  - V-Modell XT
  - Agile
  - Extreme Programming
  - Scrum

- Process Metrics
  - CMMI, Spice
• There are different "V-shaped" process models, we discuss the (German) "V-Modell".

• "V-Modell":
  • developed by company IABG in cooperation with the Federal Office for Defence Technology and Procurement ('Bundesministerium für Verteidigung'), released 1998
  • (German) government as customer often requires usage of the V-Modell
  • 2012: "V-Modell XT" Version 1.4 (Extreme Tailoring) (V-Modell XT, 2006)
V-Modell XT: (Lots of) Disciplines and Products

V-Modell XT: Activities (as many?!)

Project Roles:
- Änderungssteuerungsgruppe (Change Control Board)
- Änderungsverantwortlicher
- Anforderungsanalytiker (AG)
- Anforderungsanalytiker (AN)
- Anwender
- Assessor
- Ausschreibungsverantwortlicher
- Datenschutzverantwortlicher
- Ergonomieverantwortlicher
- Funktionssicherheitsverantwortlicher
- HW-Architekt
- HW-Entwickler
- Informationssicherheitsverantwortlicher
- KM-Administrator
- KM-Verantwortlicher
- Lenkungsausschuss
- Logistikentwickler
- Logistikverantwortlicher
- Projektkaufmann
- Projektleiter
- Projektmanager
- Prozessingenieur
- Prüfer
- QS-Verantwortlicher
- SW-Architekt
- SW-Entwickler
- Systemarchitekt
- Systemintegrator
- Technischer Autor
- Trainer

Organisation Roles:
- Akquisiteur
- Datenschutzbeauftragter (Organisation)
- Einkäufer
- IT-Sicherheitsbeauftragter (Organisation)
- Qualitätsmanager

What About the Colours?
V-Modell XT: Project Types

- **AG**: project from the perspective of the customer (create call for bids, choose developer, accept product)
- **AN**: project from the perspective of the developer (create offer, develop system, hand over system to customer)
- **AG/AN**: customer and developer from same organisation
- **PM**: introduction or improvement of a process model

Project type variants:
- One/many customer(s)
- Development/improvement/migration
- Maintenance

**Project role**
- **customer** ('Auftraggeber')
- **developer** ('Auftragnehmer')
- **customer/developer** ('Auftragg. '/'Auftragn. ')

**Project type**
- System development project (AG)
- System development project (AN)
- System development project (AG/AN)
- Introduction and maintenance of specific process model

**Project subject**
- HW system
- SW system
- HW-SW system/embedded system
- System integration
- Introduction and maintenance of specific process model

V-Modell XT: Customer/Developer Interface

V-Modell XT: Tailoring Instance

Building Blocks

V-Modell XT: Development Strategies

V-Modell XT mainly supports three strategies, i.e. principal sequences between decision points, to develop a system:
- Incremental component based prototypical

V-Modell XT: Discussion

Advantages:
- Certain management related building block are part of each project, thus they may receive increased attention
- Publicly available, can be used free of license costs
- Very generic, support for tailoring
- Comprehensive, low risk of forgetting things

Disadvantages:
- Comprehensive, tries to cover everything; tailoring is supported, but may need high effort
- Tailoring is necessary, otherwise a huge amount of useless documents is created
- Description/presentation leaves room for improvement

Needs to prove in practice, in particular in small/medium sized enterprises (SME).
"Agile — denoting 'the quality of being agile; readiness for motion; nimbleness, activity, dexterity in motion' — software development methods are attempting to offer an answer to the eager business community asking for lighter weight along with faster and nimbler software development processes. This is especially the case with the rapidly growing and volatile Internet software industry as well as for the emerging mobile application environment." (Abrahamsson et al., 2002)

The Agile Manifesto (2001):

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**

That is, while there is value in the items on the right, we value the items on the left more.

**Agile Principles**

- Continuous / sustainable delivery
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Simplicity — the art of maximizing the amount of work not done — is essential.
- Working software is the primary measure of progress.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

**Similarities of Agiles Process Models**

- Iterative: cycles of a few weeks, at most three months.
- Work in small groups (6–8 people) proposed.
- Dislike the idea of large, comprehensive documentation (radical or with restrictions).
- Consider the customer important; recommend or request customer's presence in the project.
- Dislike dogmatic rules. (Ludewig and Lichter, 2013)

**Extreme Programming (XP)** (Beck, 1999)

**XP values**

- simplicity,
- feedback,
- communication,
- courage,
- respect.

**XP practices**

- management
  - integral team (including customer)
  - planning game (→ Delphi method)
- short release cycles
- stand-up meetings
- assess in hindsight
  - team
    - joint responsibility for the code
    - coding conventions
    - acceptable workload
    - central metaphor
    - continuous integration
- programming
  - test driven development
  - refactoring
  - simple design
  - pair programming

...
CMMI, SPICE
- Assess how well the scrum process was implemented;
- Identify actions for improvement (if necessary)

Scrum
- Sprint retrospective

Process Metrics
- Assess amount and quality of realisations;
- Product owner accepts results

Agile
- Sprint

V-Modell XT
- Phase model
  - Team members, scrum master, product owner (if possible)
- Can (as other process models) be combined with techniques from XP.

Model Examples
- Procedure and Process Models
  - Necessary for success.
  - Team members are responsible for planning, and for adhering to process and rules, thus
  - Competent product owner
  - Success depends on motivation, competence, and communication skills of team members.

Scrum: Discussion
- Scrum Process
  - The famous Spiral model
  - The (in)famous Waterfall model

Procedure and Process Models
- Procedure classification
  - Linear / non-linear
  - Evolutionary, iterative, incremental
- From Procedure to Process Models
  - Team size bigger 7–10 may need

Product Increment
- Sprint Report
  - Sprint Burndown
- Sprint Review
- Sprint Burndown report
- Sprint-burndown report
- Sprints, which major requirements in which sprint
- Based on initial version of product backlog
- Assesses results of sprints
- Up-to-date
- Environment needs to support communication and cooperation, e.g. by spatial locality

Product Backlog
- Product Backlog planning
- Release Plan
- Product Backlog
  - Requirements
  - Requirements to be realised in next sprint, taken from product backlog
  - More precise estimations
  - Members capable of developing autonomously

Sprint Backlog
- Sprint Backlog
  - Requirements to be realised in next sprint, taken from product backlog
  - Priority and effort estimation for requirements
  - Reports
  - Completed/open tasks from sprint backlog

Sprint Report
- Sprint Report
  - More precise estimations
  - Maintains requirements in the product owner
  - Representative of customer
  - Scrum team
  - Responsibilities:
    - Scrum master
      - Ensures that the team is not disturbed from outside
      - Moderates daily scrum
      - Responsible for keeping completed/open tasks from sprint backlog
      - Plans and decides which requirement(s) to realise in next sprint
      - Plans and decides which techniques and approaches
      - Environment needs to support communication and cooperation, e.g. by spatial locality
    - Scrum of scrums
      - Three roles (passive) participant of
      - Sprint-burndown report
      - Based on initial version of product backlog
      - Product owner
      - Scrum master
      - Daily scrum
        - Moderates
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Sprint-burndown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-burndown report
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
      - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Based on initial version of product backlog
      - Release plan
        - Maintained by product owner
        - Sprint-backdown report
      - Release-burndown report
      - Daily scrum
        - Daily update (tasks done, new tasks, new estimations)
### Process Metrics

#### Assessment and Improvement of the Process

- **Idea** (for material goods): The quality of the (production) process influences product quality.
- **Plan**: Specify abstract criteria (metrics) to determine good production processes (e.g., to choose manufacturer).

**Industry in general** (production):
- ISO 9001, ISO/TS 16949 (automotive), ...

**Software industry** (development):
- CMM(I), SPICE

**Note**: A good process does not stop us from creating bad products; (the hope is, that) bad products are less likely when using a good process, i.e., there is a correlation:

\[
\begin{array}{c|c|c}
\text{Process Quality} & \text{Product Quality} \\
\hline
\text{Low} & \text{High} & \text{false positive} \\
\text{High} & \text{Low} & \text{false negative} \\
\text{High} & \text{High} & \text{true positive} \\
\text{Low} & \text{Low} & \text{true negative} \\
\end{array}
\]

### Improving processes for developing better products and services

- **1991**: Capability Maturity Model (CMM), DoD/SEI/CMU; superseded by
- **1997**: Capability Maturity Model Integration (CMMI) (Team, 2010);

**Constellations**:
- CMMI-DEV (development), CMMI-ACQ (acquisition), CMMI-SRV (service)

**Goals**:
- Applicable to all organisations which develop software,
- Make strengths and weaknesses of the real process visible, to point out ways for improvement,
- Neutral wrt. technology employed in project,
- Levels: Higher levels have lower levels as premise,
- Be consistent with ISO 15504 (SPICE)

**Assumptions**:
- Better defined, described, and planned processes have higher maturity,
- Higher maturity levels require statistical control to support continuous improvement,
- Higher maturity level yields:
  - Better time/cost/quality prediction;
  - Lower risk to miss project goals;
  - Higher quality of products.

### CMMI Levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Level Name</th>
<th>Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Managed</td>
<td>REQM, PP, PMC, MA, PPQA, CM, SAM</td>
</tr>
<tr>
<td>3</td>
<td>Defined</td>
<td>+ RD, TS, PI, VER, VAL, OPF, OPD, OT, IPM, RSKM, DAR</td>
</tr>
<tr>
<td>4</td>
<td>Quantitatively Managed</td>
<td>+ OPP, QPM</td>
</tr>
<tr>
<td>5</td>
<td>Optimising</td>
<td>+ OID, CAR</td>
</tr>
</tbody>
</table>

---

**References**

- Team, (2010). [Improving processes for developing better products and services](#).
- ISO 9001, ISO/TS 16949 (automotive), SPICE.
Spice, CMMI: focuses on management aspects
Scrum: proposes methods and approaches
XP: ...-N and not already in level N
CMMI levels are chosen somewhat arbitrarily: “why is an area in level N changed processes may require new (expensive) appraisal, in this sense CMMI certification may hinder innovation, can be tailored in various ways... quite comprehensive... CMMI assumptions are based on experience in specific projects; may not be present for all kinds of software, slightly different vocabulary:... Criticism
V-Model XT for important aspects of process models wrt. product quality
incremental, iterative, evolutionary... inspiration CMMI can serve as an... SPICE / ISO 15504 provides “process reference models” (in particular specific ones for automotive, aerospace, etc.)... may guide selection of sub-contractors (a certificate at least proves that they think about their process)... by certain (U.S) government customers; required CMMI certificate is... CMMI as such very well-known, very abstract, of limited practical use. 

CMMI: Discussion

<table>
<thead>
<tr>
<th>Area</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD</td>
<td>GG.1</td>
<td>GG.1,</td>
<td>GG.1,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GG.2</td>
<td>GG.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GG.3</td>
<td>GG.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Termination e D Improvement and I Process P Software S T ell Them What Y ou’ve T old Them...


