

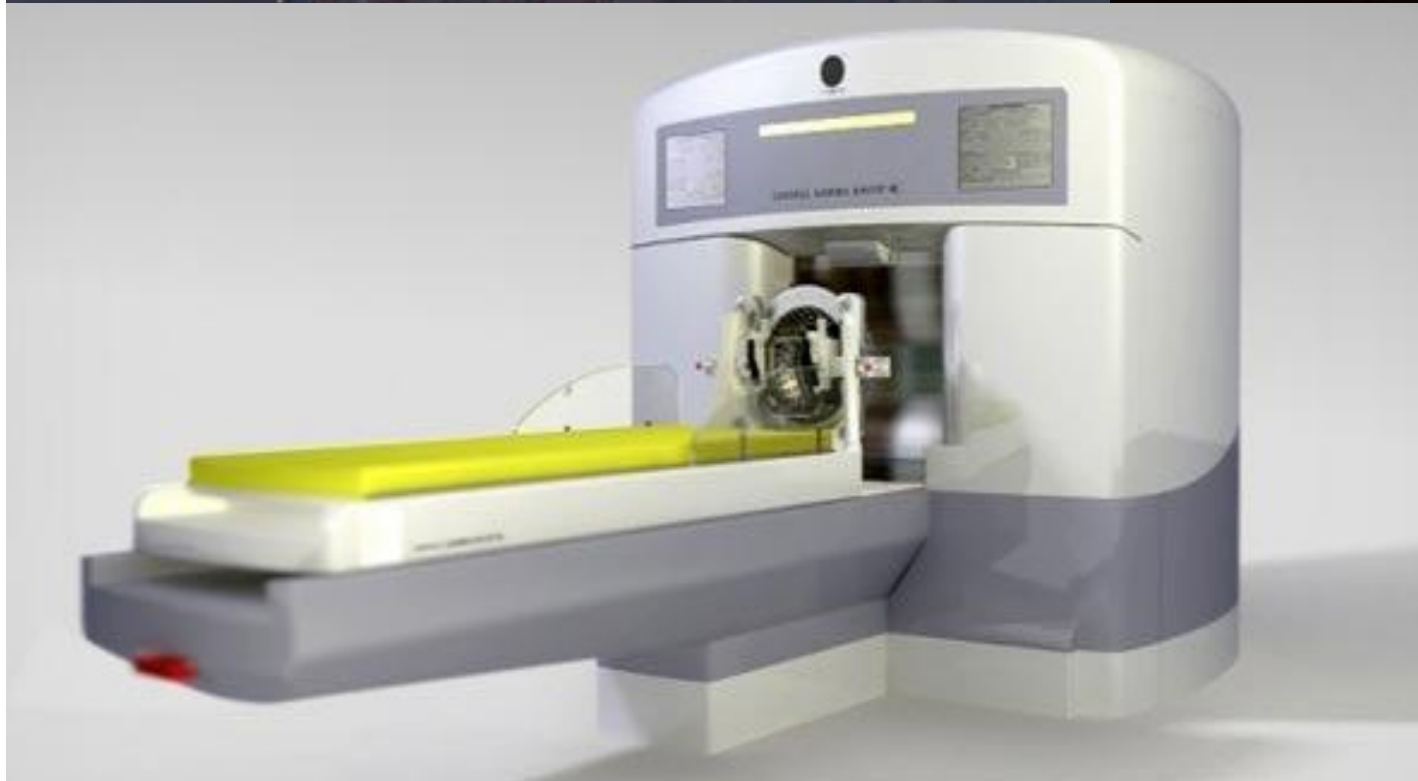
Introduction to Software Engineering for Engineers

Lecture-02: Process Models & SCRUM

Part 1: Introduction to Software Engineering

Dr.-Ing. Christoph Steup

Most Expensive Software Failures



top-10-list.org/2010/05/03/ten-costliest-software-bugs/



What is Software?

Software Systems

Software

computer programs, procedures, rules and possibly associated documentation and data pertaining to the operation of a computer system.

(IEEE Standard Glossary of Software Engineering)

Software System

A system (or parts thereof), whose components consist of software



Software Systems (2)



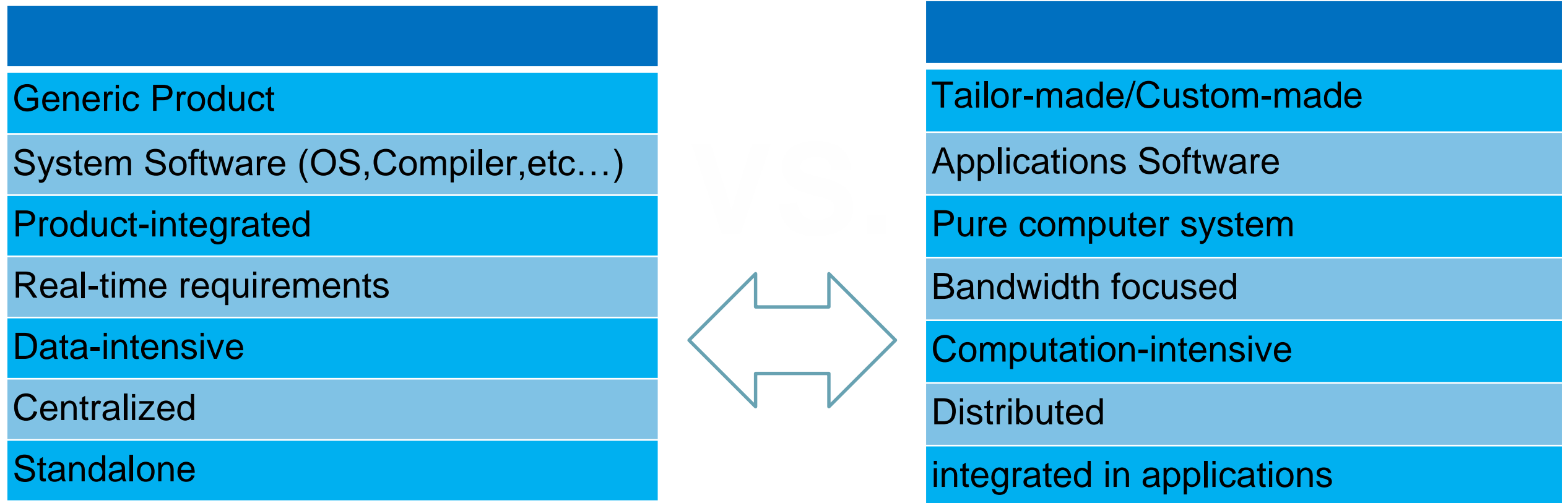
Product

A product is a self-contained, usually for a particular customer created, result of a successful accomplished project or manufacturing process. As partial product we denote a completed part of a product.

Software Product

A product, consisting of Software.

Classification of Software



No "eierlegende Wollmilchsau" (Swiss Army Knife)!

Software is characterised by application and domain



Peculiarities of Software

Software is not limited by physical boundaries.

Software is immaterial.

Software is hard to measure
(„Technical data“ of Software?).

Software is under
permanent pressure to
adapt.

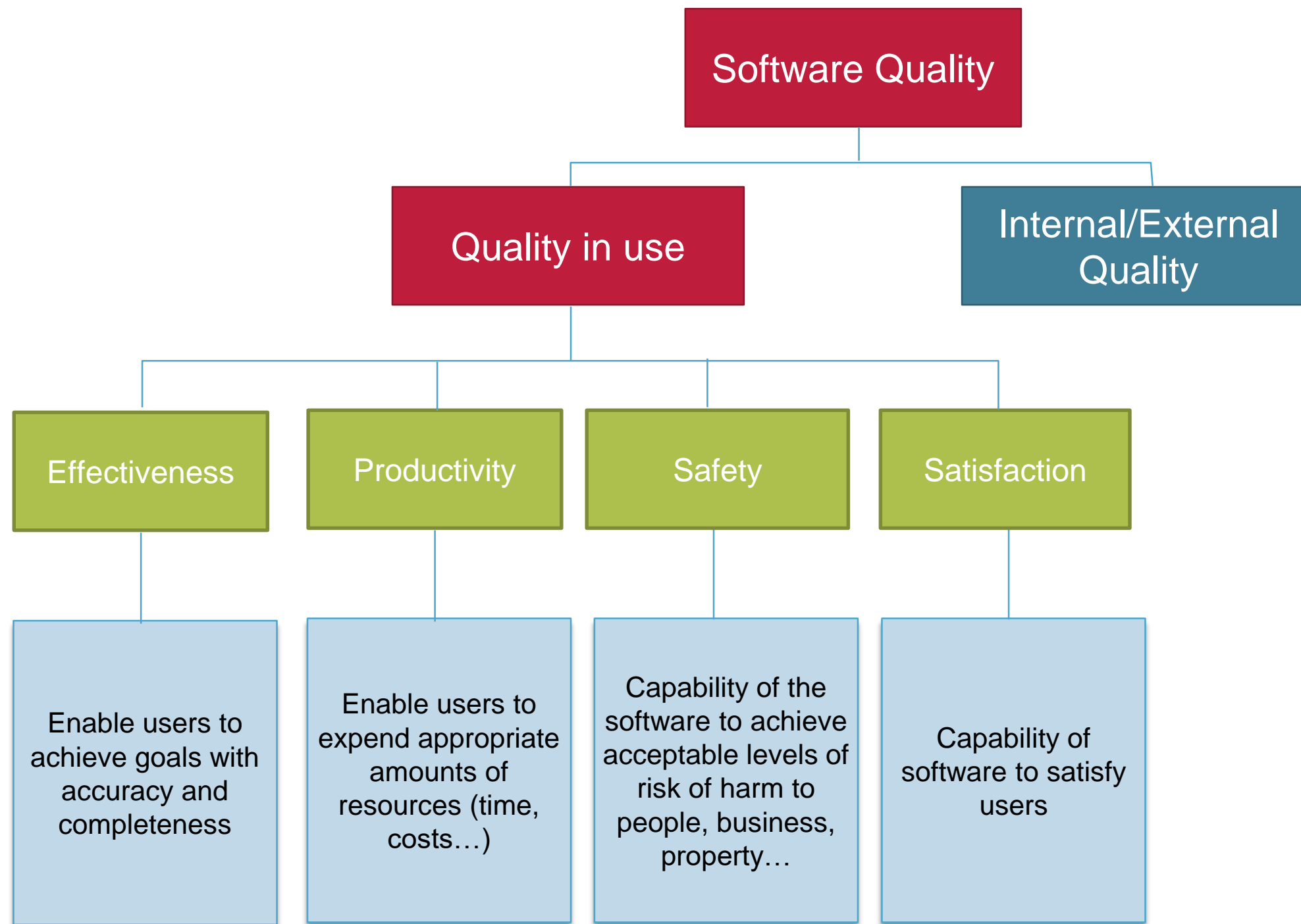
Software is relatively easy to change
(compared to material technical products).

Software is not subject to wastage.

Software becomes outdated.

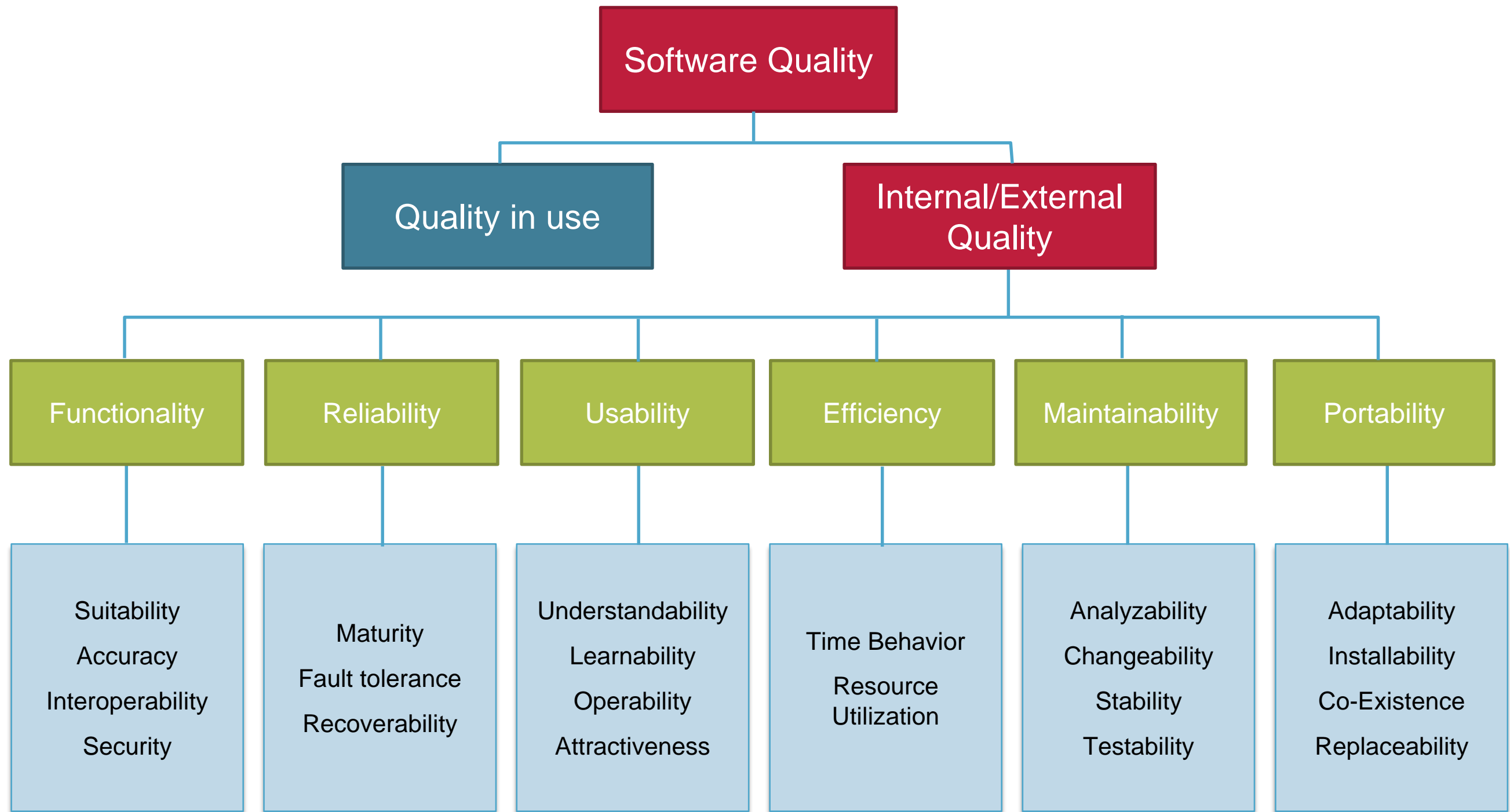
There are no software spare parts:
Defects are faults by design/construction.

Characteristics of Software Quality

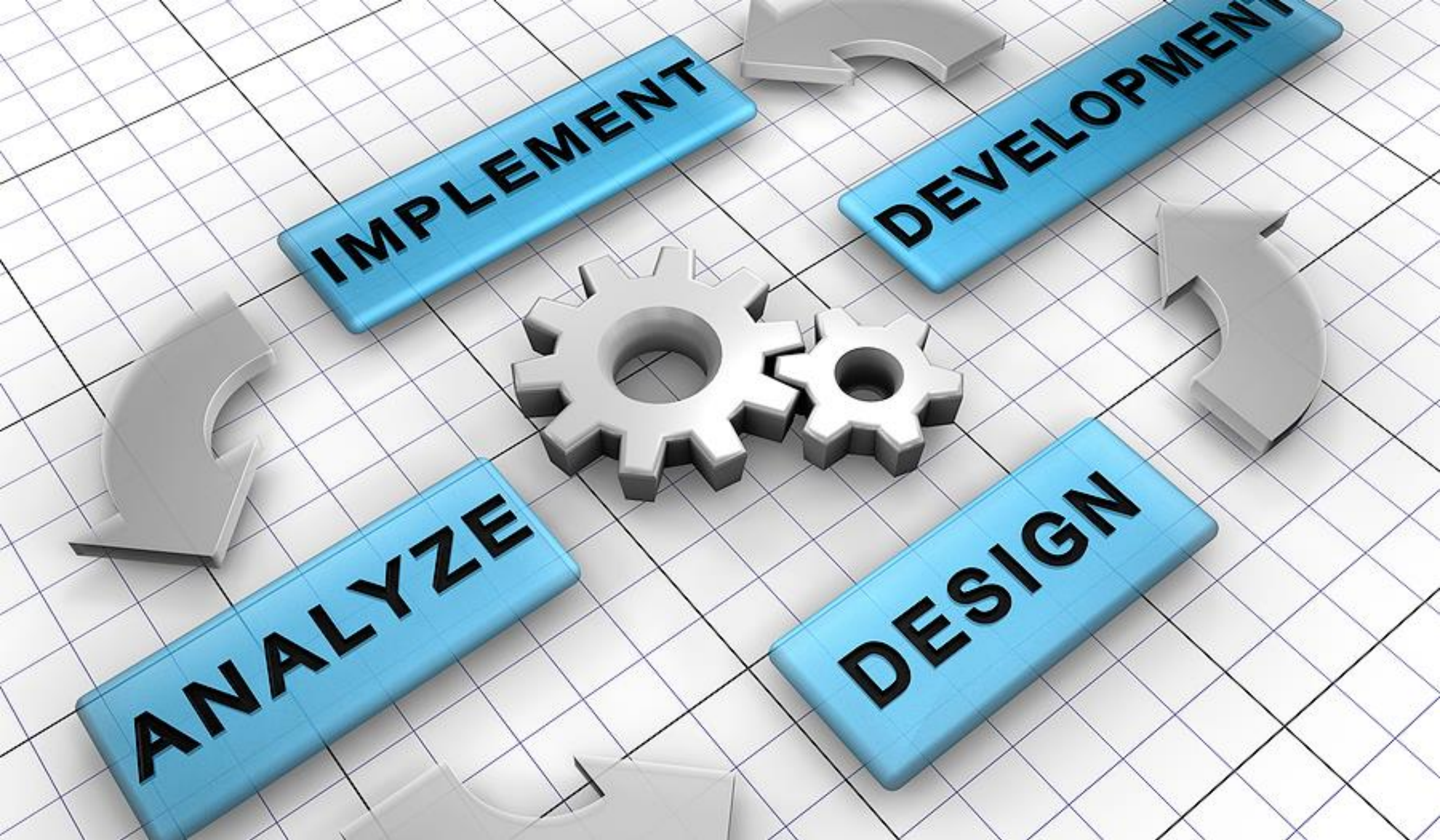


*ISO/IEC 9126: Bewerten von Softwareprodukten,
Qualitätsmerkmale und Leitfaden zu ihrer Verwendung*

Characteristics of Software (2)



*ISO/IEC 9126: Bewerten von Softwareprodukten,
Qualitätsmerkmale und Leitfaden zu ihrer Verwendung*



What is Software
Engineering?

Software Engineering

Software Engineering

The establishment and use of sound engineering principles in order to obtain economically software that is reliable and runs on real machines.

(F.L. Bauer, NATO-Konferenz Software-Engineering 1968)



Manifesto of Software Engineering (2006)

Software Engineering aims at an engineering-like development, maintenance, adaptation, and evolution of large-scale software systems. To this end, systematic processes, principles, methods, and tools should be applied.

Topics in Software Engineering

Project Management

Process Modelling

Software Development Methods

Requirements
Engineering

Software
Architecture
and Design

Software
Maintenance

Reengineering
(Sanitization)

Quality Assurance (incl. testing approaches)

Notations and Languages (e.g., UML, Java, ...)

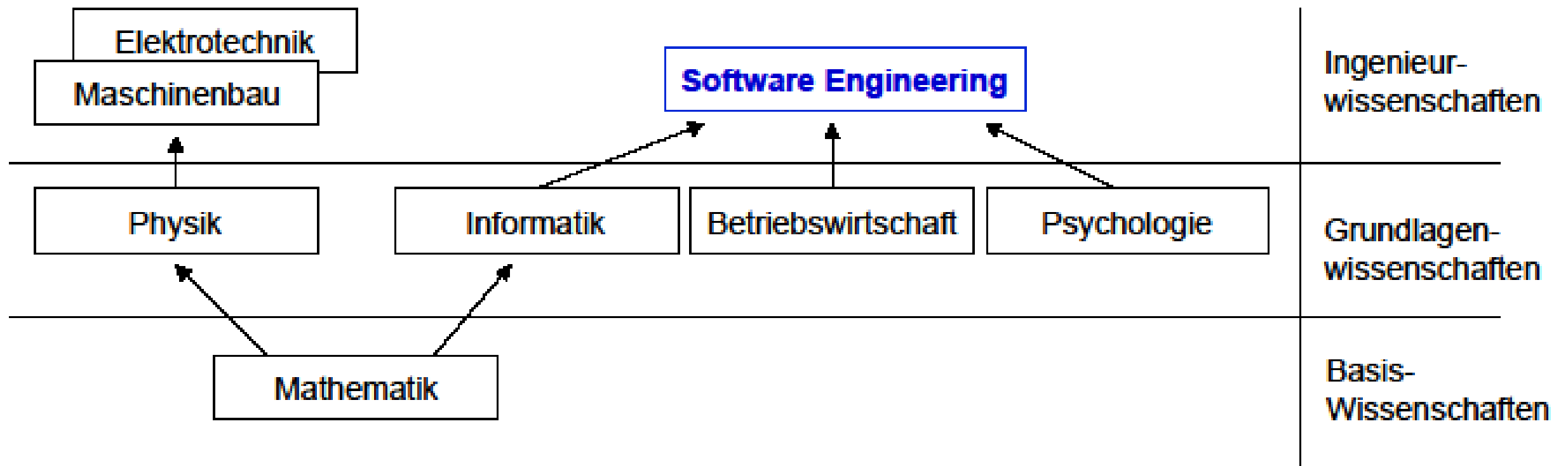
Tool Support (incl. CASE, CVS, make)

Software Engineering is much more than “just”
Programming

Constraints/Requirements of Software



Software Engineering vs. Computer Science



Software Engineering is the engineering discipline/science of computer science (similar to the relation between mechanical engineering and physics).

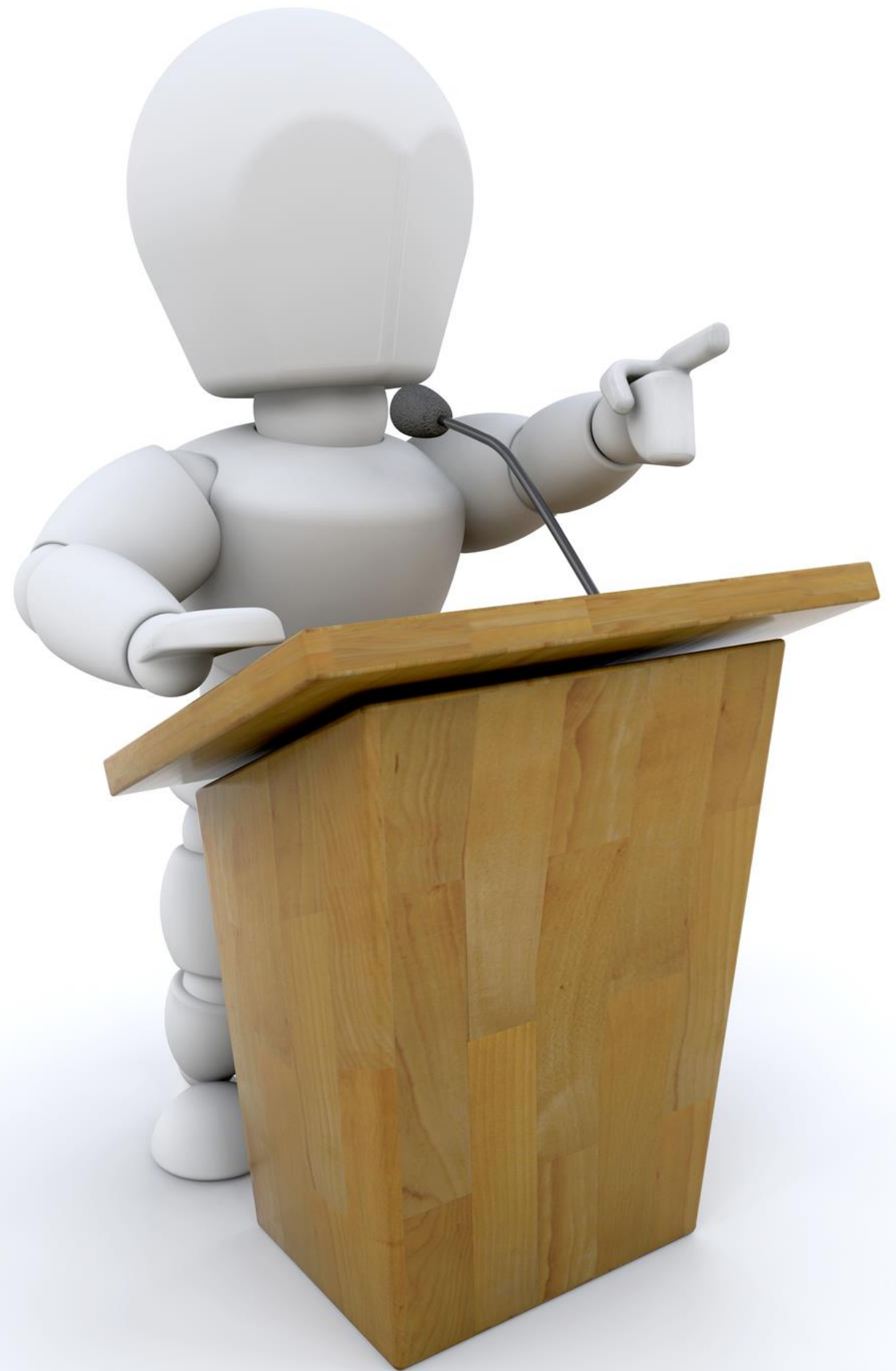
Portfolio of Software Engineering

- For each Problem the appropriate tool at hand of experts, **who can use it.**
- You don't need to be able to cope with all tools, but: **the more, the better**

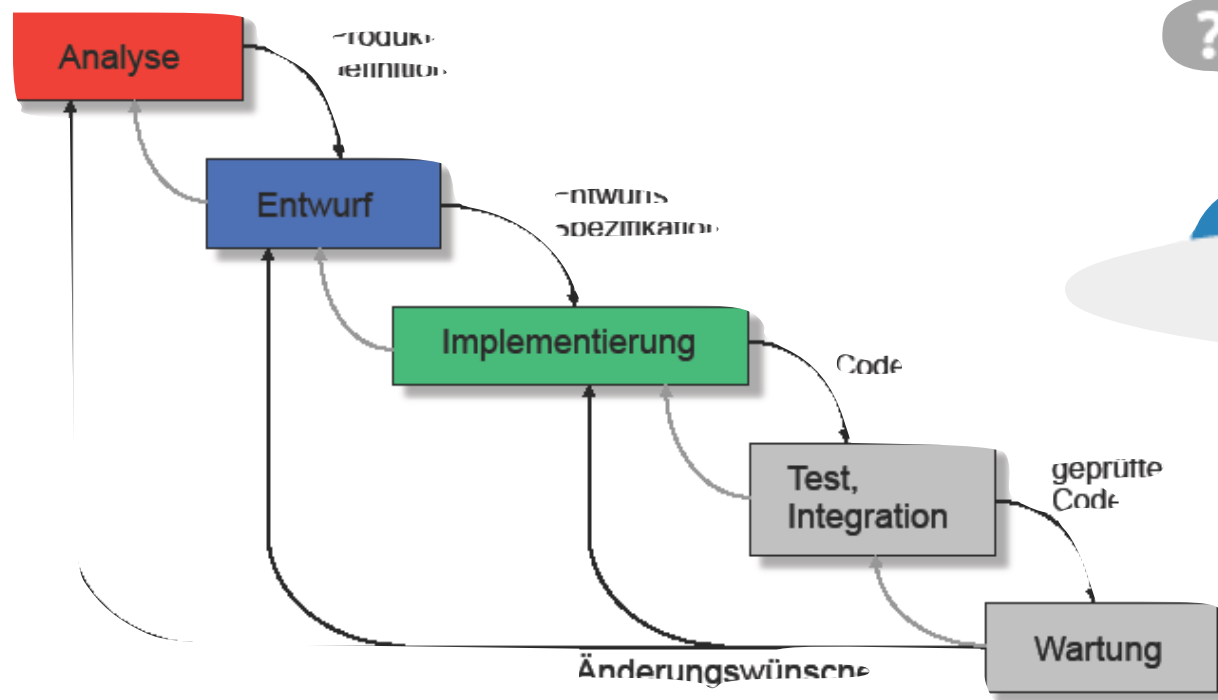


Summary: Software Systems and SE

- Software engineering is an engineering discipline that is concerned with all aspects of developing software-intensive systems.
- Diversity of application domains, size, etc. requires a Portfolio of techniques to-be applied.
- Quality is a crucial factor of the developed software.
- Software engineering is application-oriented, and thus, requires that methods and techniques are practiced frequently.



Outlook: Topics of Lecture



nach W. Royce (1970), mit Rückkopplung B. Boehm (1981)



Introduction to Software Engineering for Engineers

Lecture-02: Process Models & SCRUM Part 2: Process Models

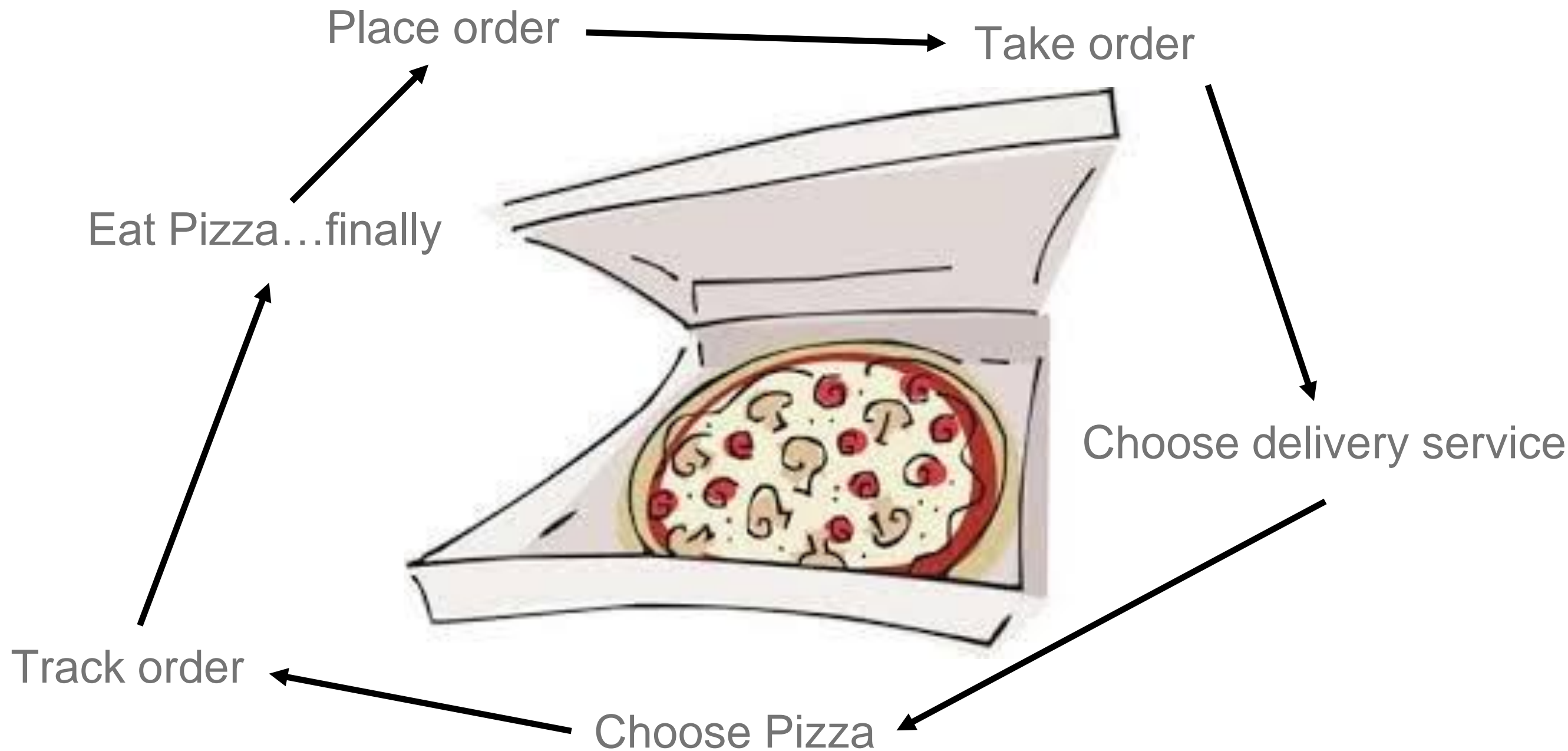
Dr.-Ing. Christoph Steup

Software Process Models

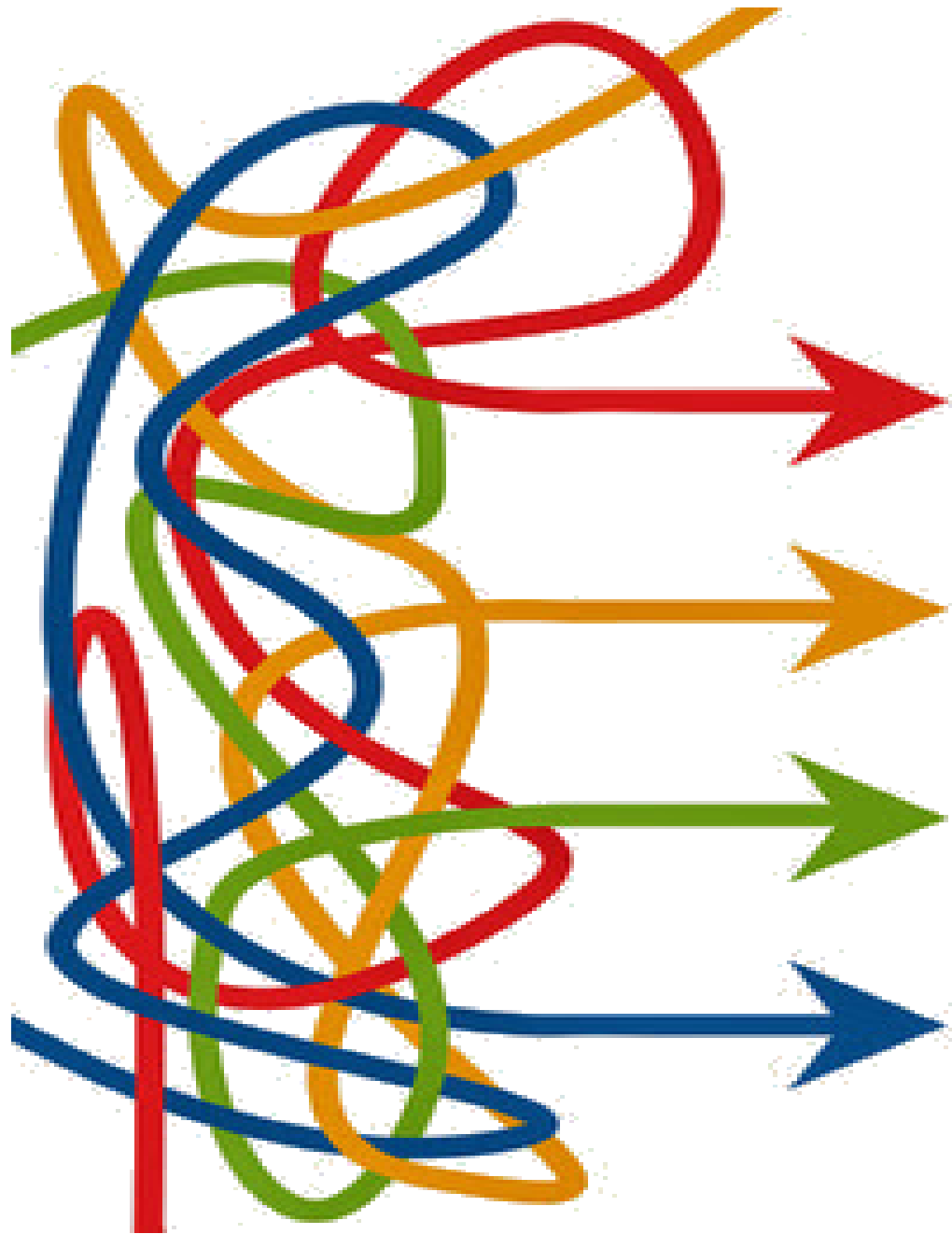
“We try to solve the problem by rushing through the design process so that enough time is left at the end of the project to uncover the errors that were made because we rushed through the design process”

–Glenford Myers

Motivation



Benefits of Process Models



- Structuring of a project
- Phases and corresponding activities
- Communication
- Responsibilities
- Completeness
- Prediction of project results
- Monitoring and Analysis of project
- Gaining experience

Activities in Software Development

Analysis



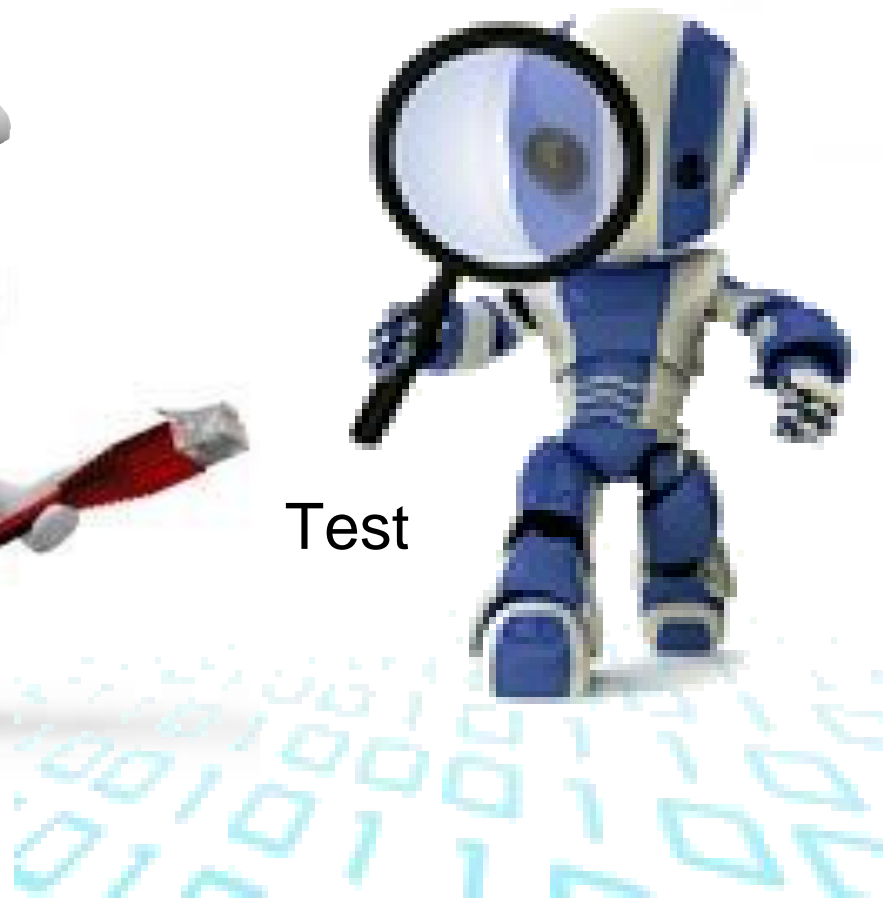
Design



Deployment



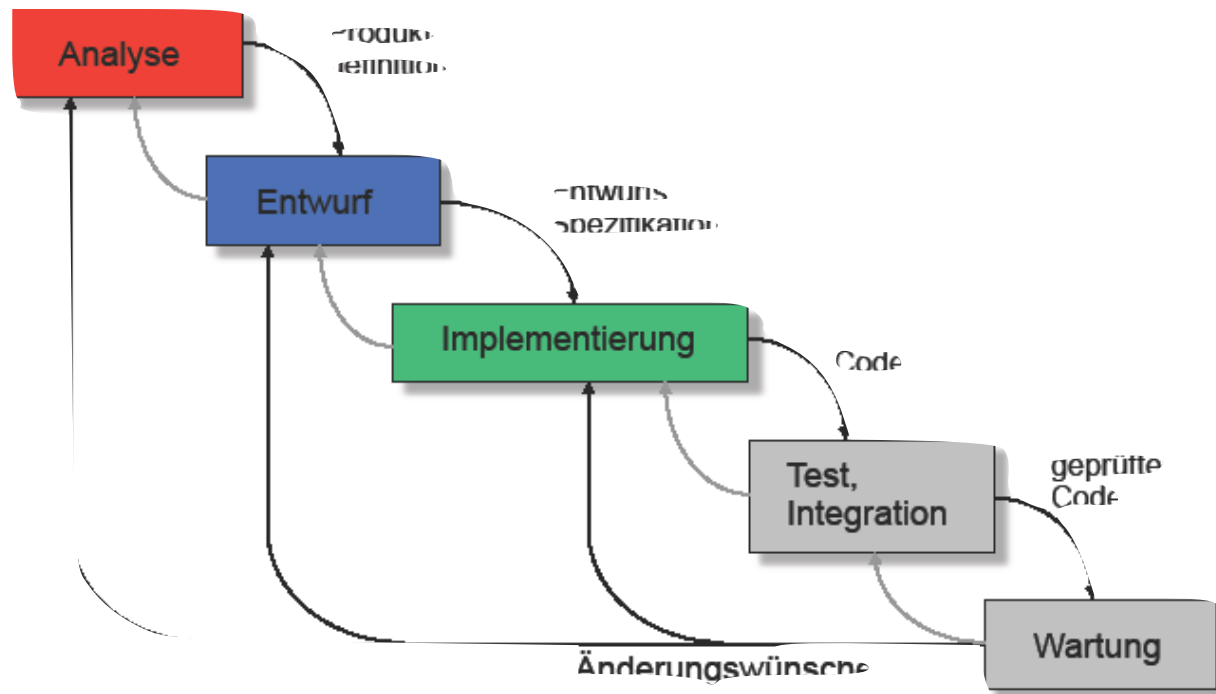
Test



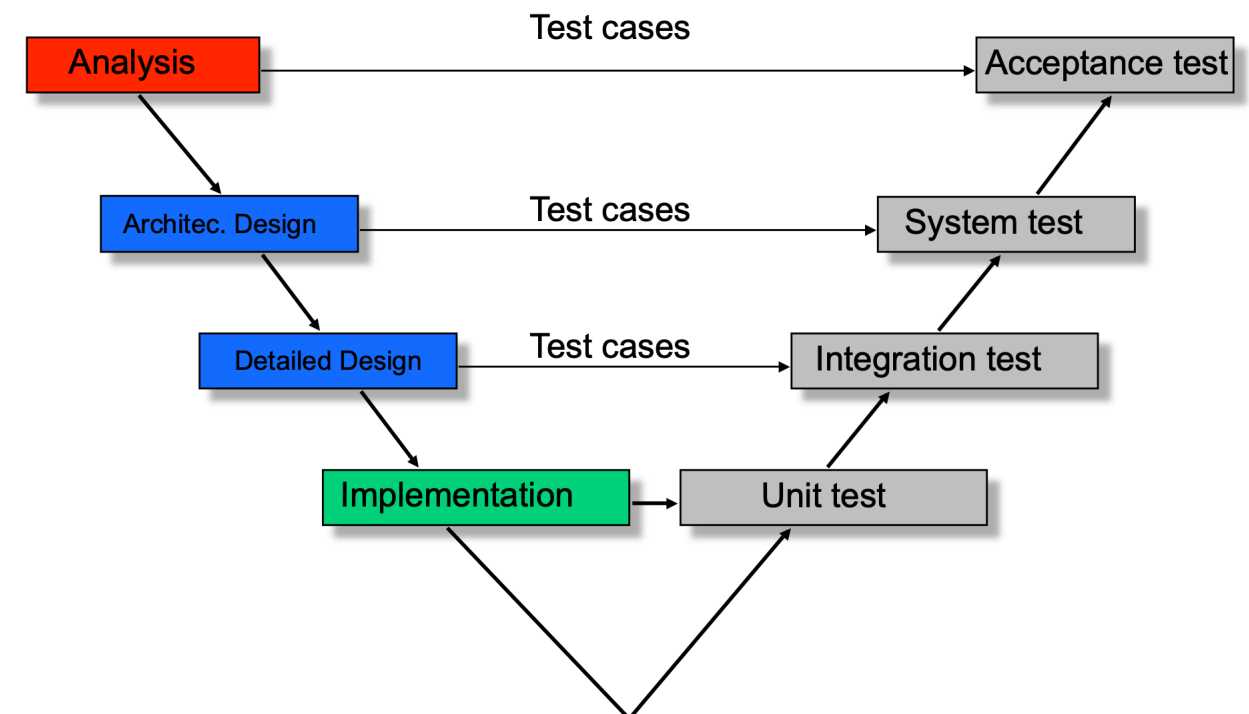
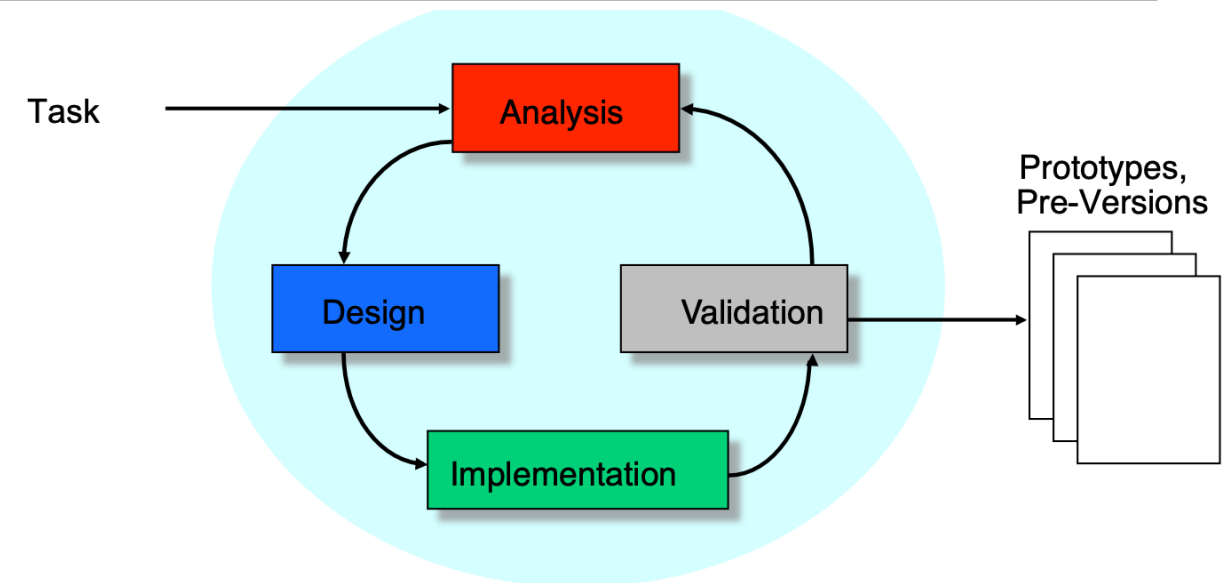
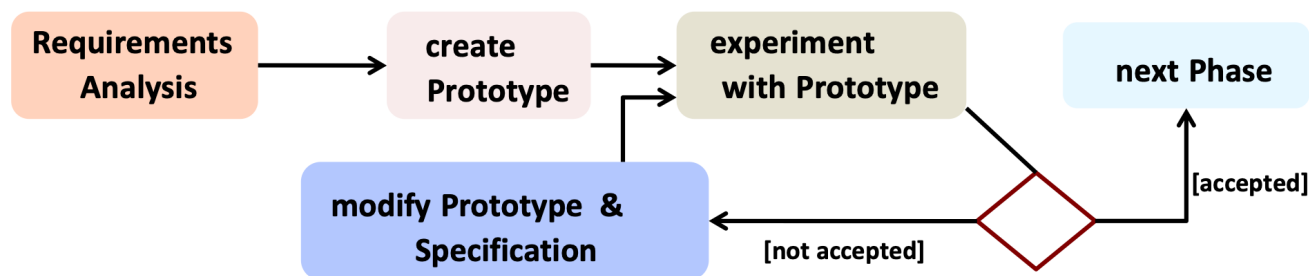
Implementation



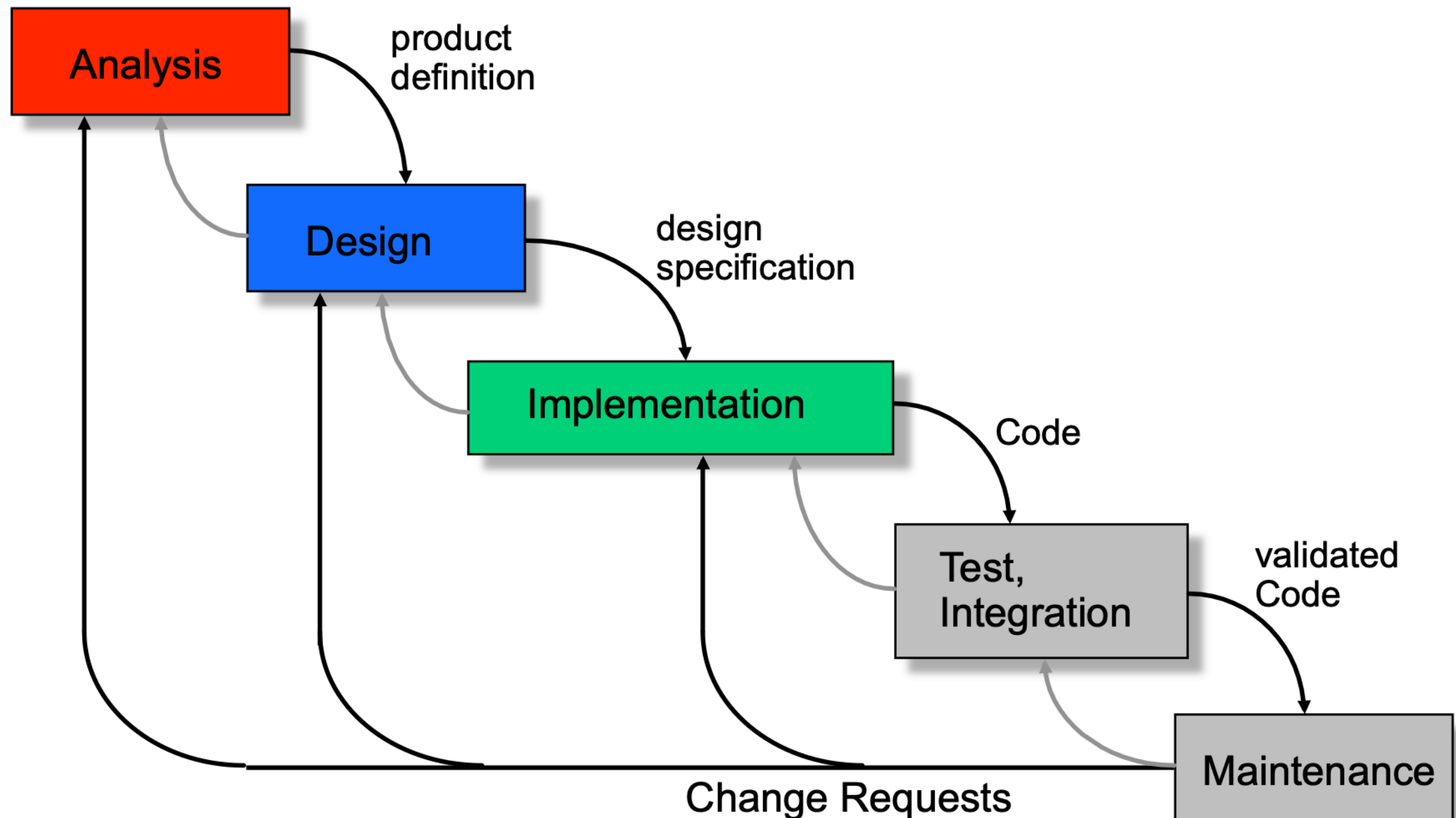
Popular Process Models



nach W. Royce (1970), mit Rückkopplung B. Boehm (1981)



Waterfall Model



nach W. Royce (1970), mit Rückkopplung B. Boehm (1981)

Characteristics — Waterfall Model

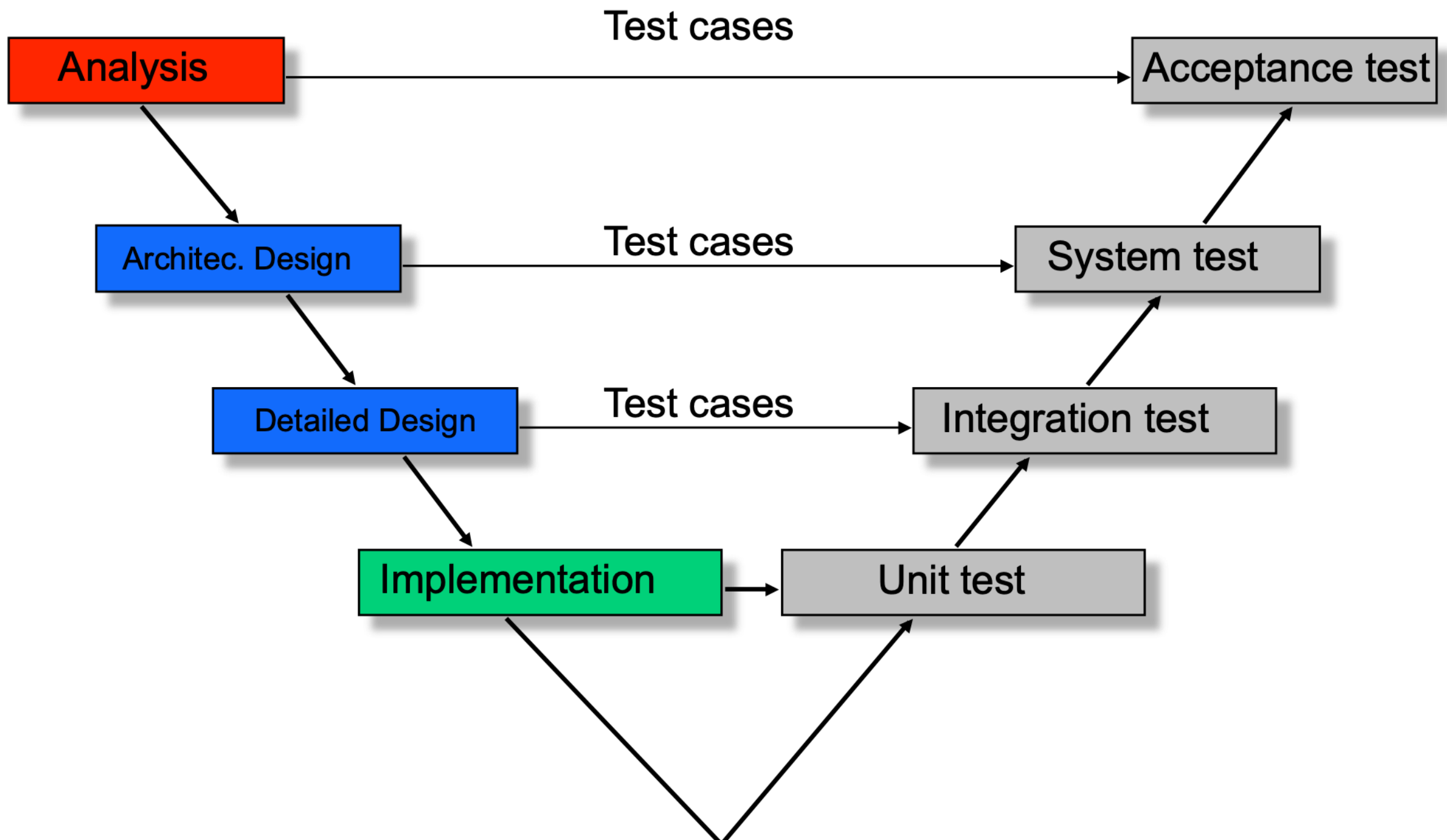
All steps are executed in sequential order

If, and only if, one step is finished, the next one is started —> result of the previous phase must be known

Assessment:

- Easy to understand
- Easy to manage
- Easy to control (defined transitions between phases)
- Problem in case of changes and delays of particular phases

V-Model



Characteristics V-Model

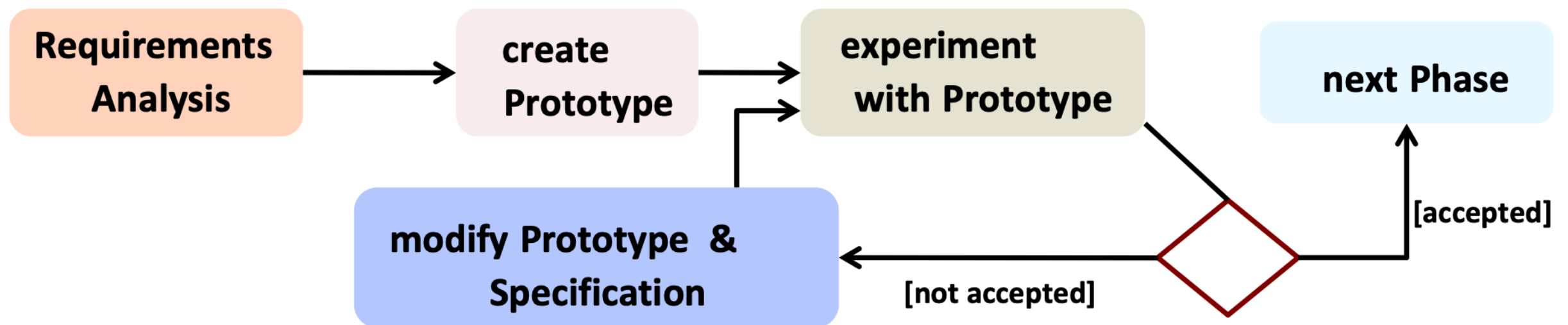
Extension of Waterfall Model

Integration of quality assurance (verification and validation)

Mandatory for German Armed Forces and Administration (V-Model XT)

Very comprehensive model; must be tailored for a concrete development task (Tailoring)

Prototype-based Process



Characteristics Prototype-based

Preliminary version (of parts) of the intended system.

Classification of Prototypes:

- One-time (throw-away) Prototype for demonstration purposes; only very basic and well-understood features are realized
- Evolutionary Prototype, i.e., development of version 0.1 as a basis for the later developed System; first of all the best understood parts/features are realized
- Paper Prototyping (e.g., for GUI development)

Assessment:

- Fast results
- Flexible adaptation
- Longer development process due to inappropriate documentation
- More expensive in the prototype phase

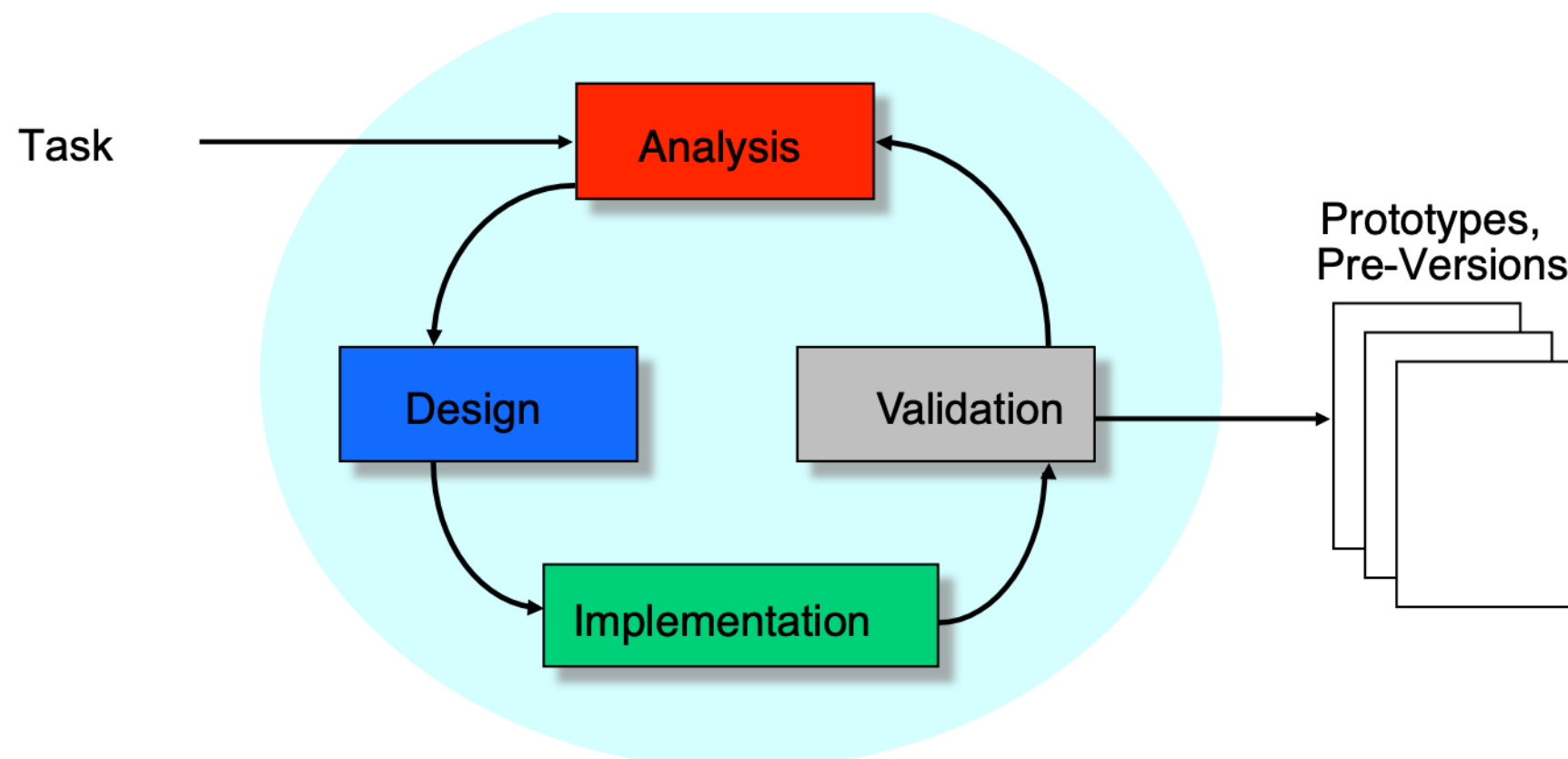
Iterative Models

Development process consists of sequence of cycles (*iterations*).

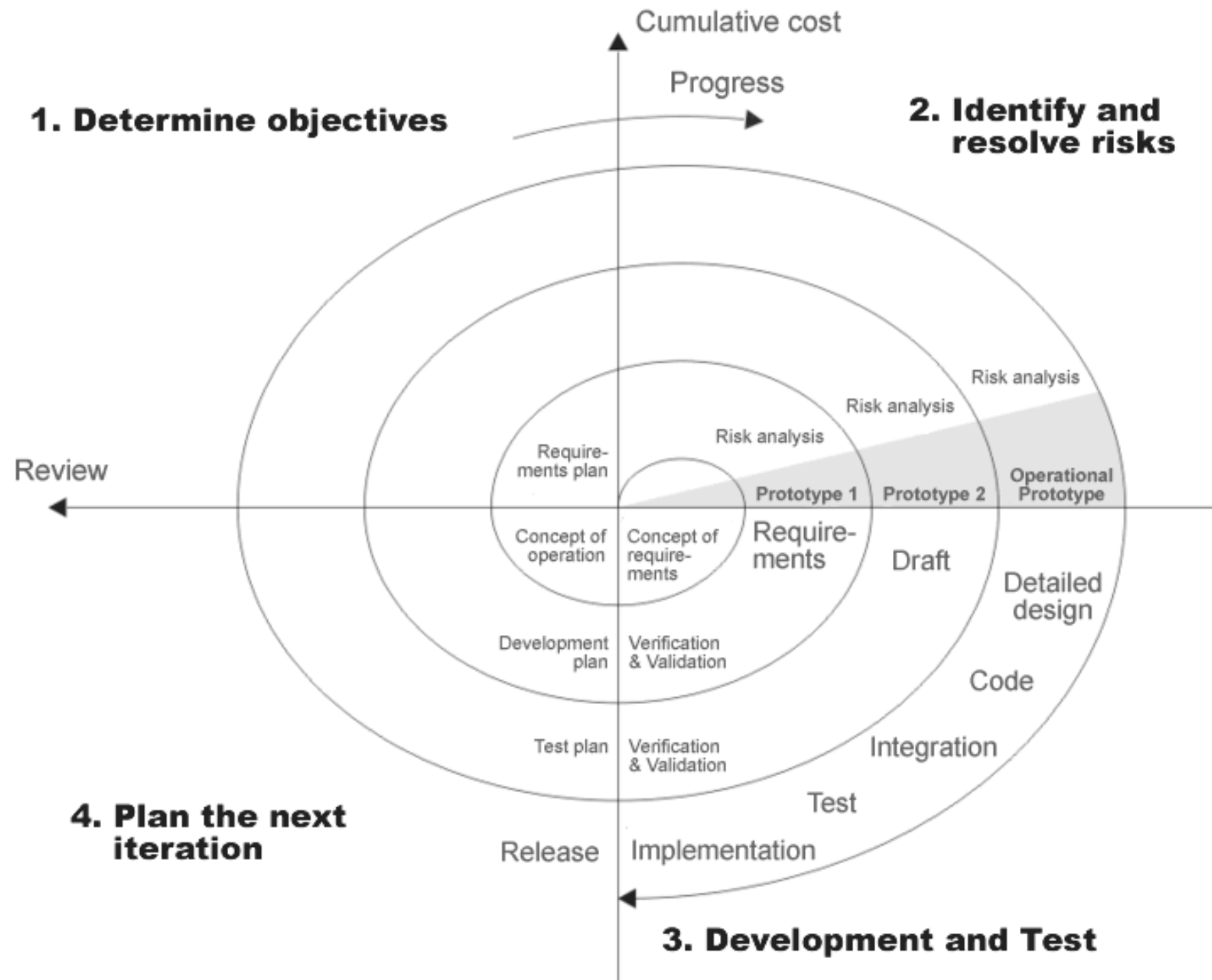
At the end of each cycle a new (executable) version of the SW product is available. Usually, this new version improves and extends the previous version.

Assessment:

- Fast results
- Flexible adaptation



Spiral Model



Characteristics Spiral Model

Iterative Method

Invented by Barry Boehm

Each cycle has the following activities in the respective quadrants:

1. Determining objectives, identifying alternatives, and description of basic conditions
2. Evaluating alternatives; detecting, estimating and reducing risk, e.g., by analysis, simulations, or prototyping
3. Realization and inspection of intermediate product
4. Planning of next cycle (of project continuation)

Process Models — Summary

- Process Models are good for
 - establishing a basic framework for SW projects
 - plan resources and major releases/milestones
 - track progress *in-the-large*
- However, they have been proven to fail due to
 - their static nature —> everything needs to be planned ahead
 - missing flexibility —> what about changing requirements or unforeseen incidents
 - their process-focused view —> technical debt is omitted as well as other aspects that are important in SW projects (humans, tools)
 - flexible models tend to be expensive and come with implementation overhead

Introduction to Software Engineering for Engineers

Lecture-02: Process Models & SCRUM

Part 3: Agile Software Engineering & SCRUM

Dr.-Ing. Christoph Steup

Standish Group CHAOS Report

- In the U.S.
 - \$250 billion per year are spent for IT application development
 - 31.1% of SE projects are cancelled before they ever get completed
 - 52.7% of projects cost 189% of their original estimates

MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Standish Group CHAOS Report (II)

- Obviously, size matters and is hard to manage even with common process models

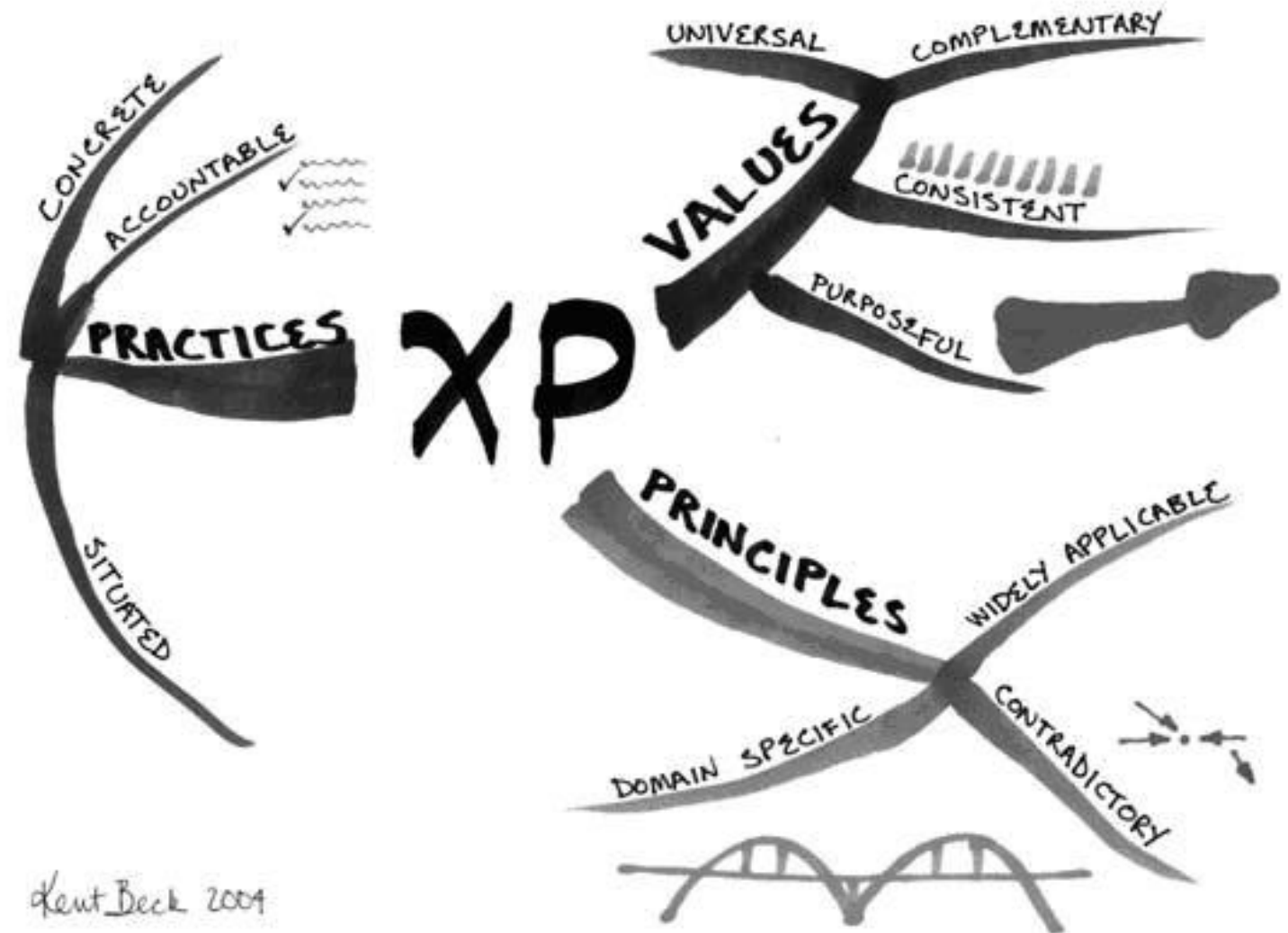
CHAOS RESOLUTION BY PROJECT SIZE			
	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

Standish Group CHAOS Report (III)

- Alternatives exist that overcome the limitations of common process models

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL				
SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Agile Models



Characteristics

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the “agile processes”

The Agile Manifesto

Individuals and
interactions

over

Process and tools

Working software

over

Comprehensive
documentation

Customer
collaboration

over

Contract negotiation

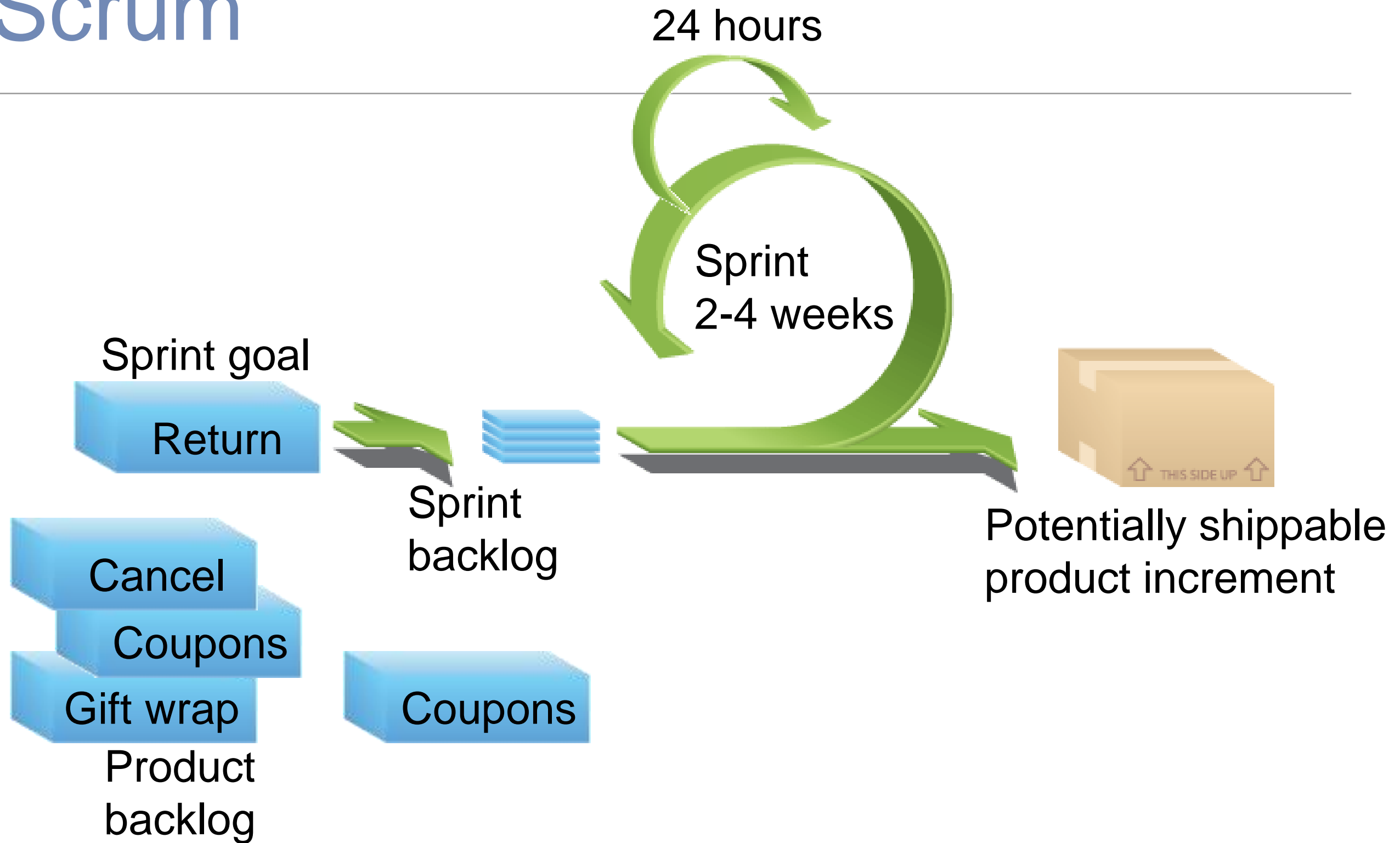
Responding to
change

over

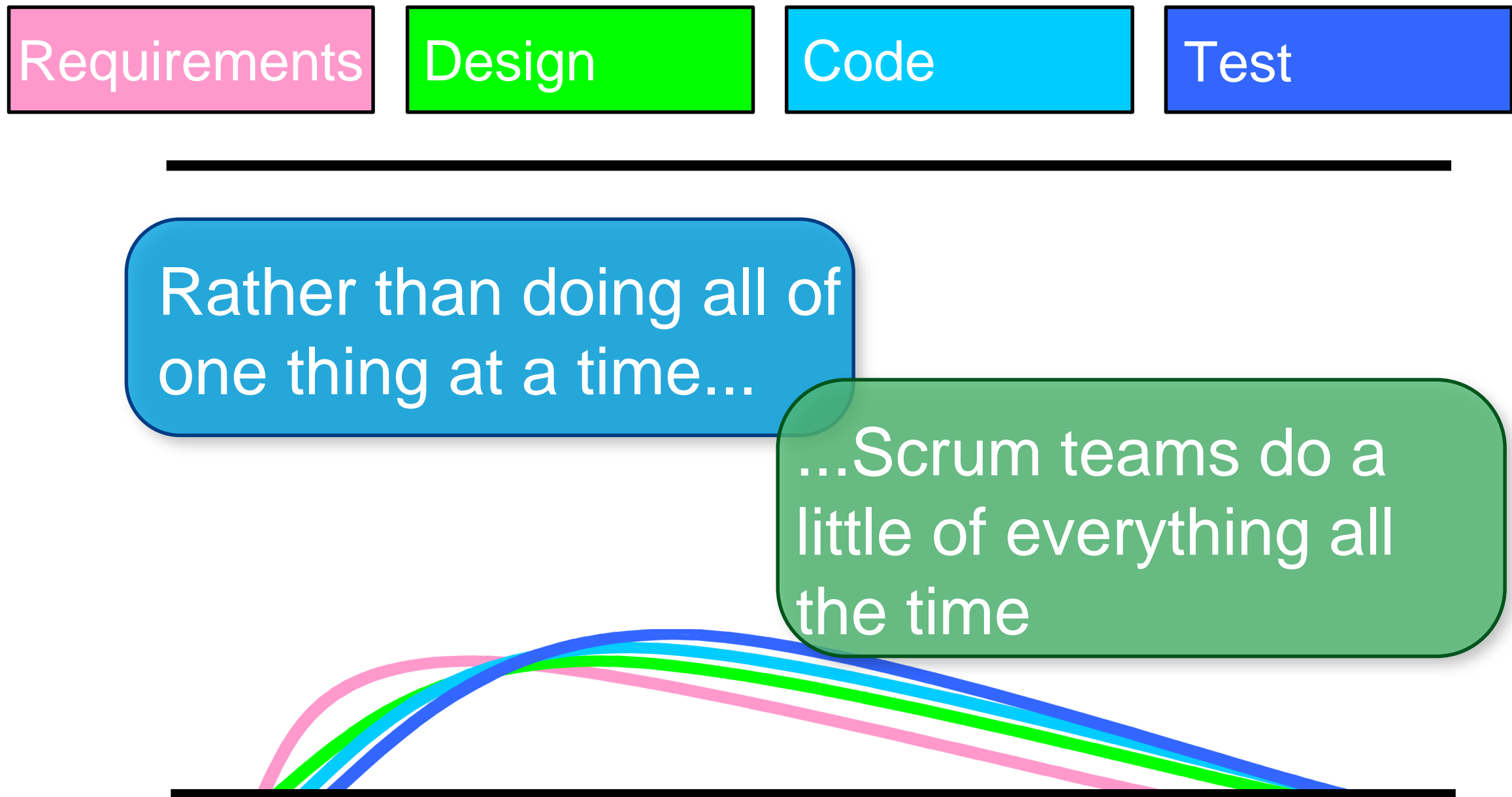
Following a plan

Source: www.agilemanifesto.org

Scrum



Sequential vs. Overlapping Development



Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

SCRUM Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

SCRUM Framework

Roles

- Product owner
- ScrumMaster
- Team

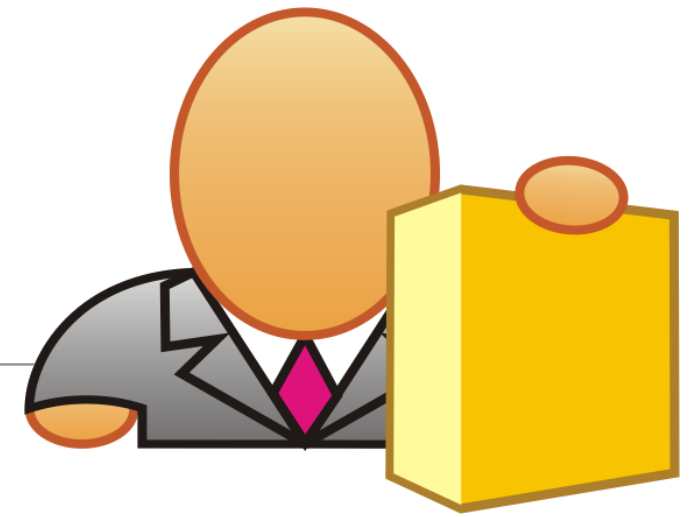
Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product Owner



- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

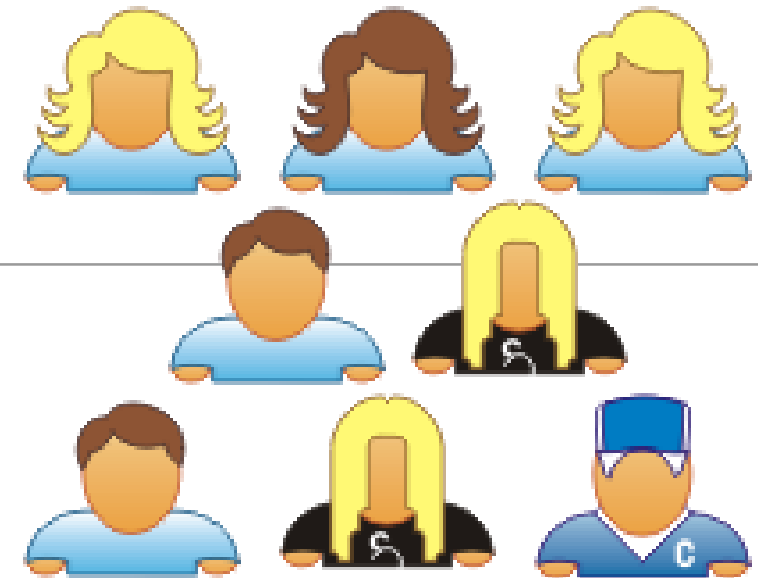
The SCRUM Master



- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

The Team

- Typically 5-9 people
- Cross-functional:
 - Programmers, Testers, User Interface Designers, etc.
- Members should be full-time
 - Exceptions possible (e.g., database administrator)
- Teams are self-organizing
 - Ideally flat hierarchy, often not possible
- Membership should change only between sprints



SCRUM Framework

Roles

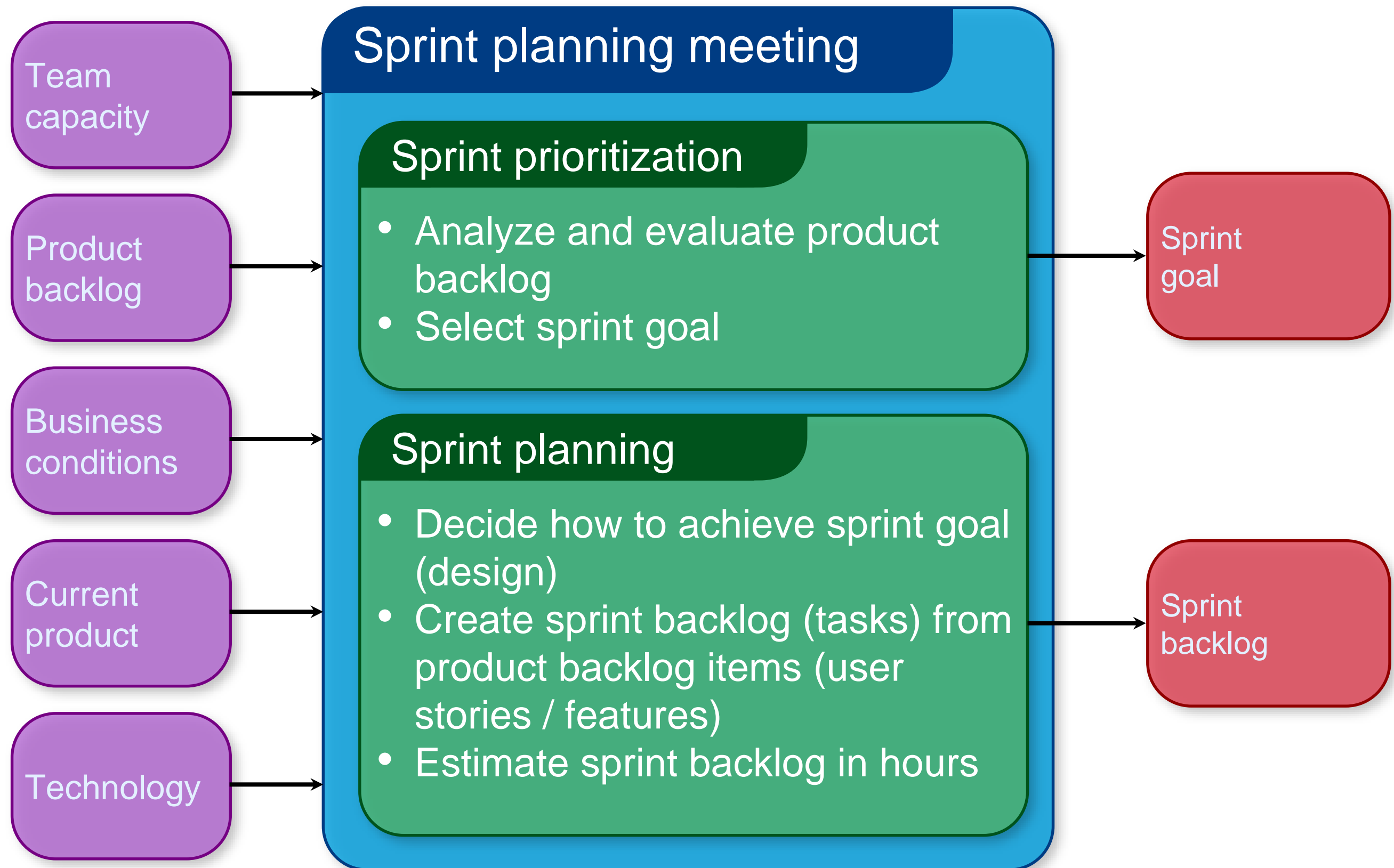
- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



Sprint Planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
- Tasks are identified and each is estimated (1-16 hours)
- Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

The Daily SCRUM

- Parameters

- Daily
- 15-minutes
- Stand-up

- Not for problem solving

- Whole world is invited
- Only team members talk; ScrumMaster, product owner, can attend

- Helps avoid other unnecessary meetings



Everyone answers THREE Questions

1

What did you do yesterday?

2

What will you do today?

3

Is anything in your way?

- These are not status for the ScrumMaster
- They are commitments in front of peers (Flat Hierarchy)

The Sprint Review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
 - 2-hour preparation time rule
 - No slides
- Whole team participates
- Invite the world



Sprint Retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
 - ScrumMaster
 - (Product owner) —> only if different from stakeholder
 - Team
- Possibly customers and others

SCRUM Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product Backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint



COPYRIGHT © 2005. MOUNTAIN GOAT SOFTWARE

This is the
product backlog

A Sample Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

The Sprint Goal

- A short statement of what the work will be focused on during the sprint

Life Sciences

Support features necessary for population genetics studies.

Database Application

Make the application run on SQL Server in addition to Oracle.

Financial services

Support more technical indicators than company ABC with real-time, streaming data.

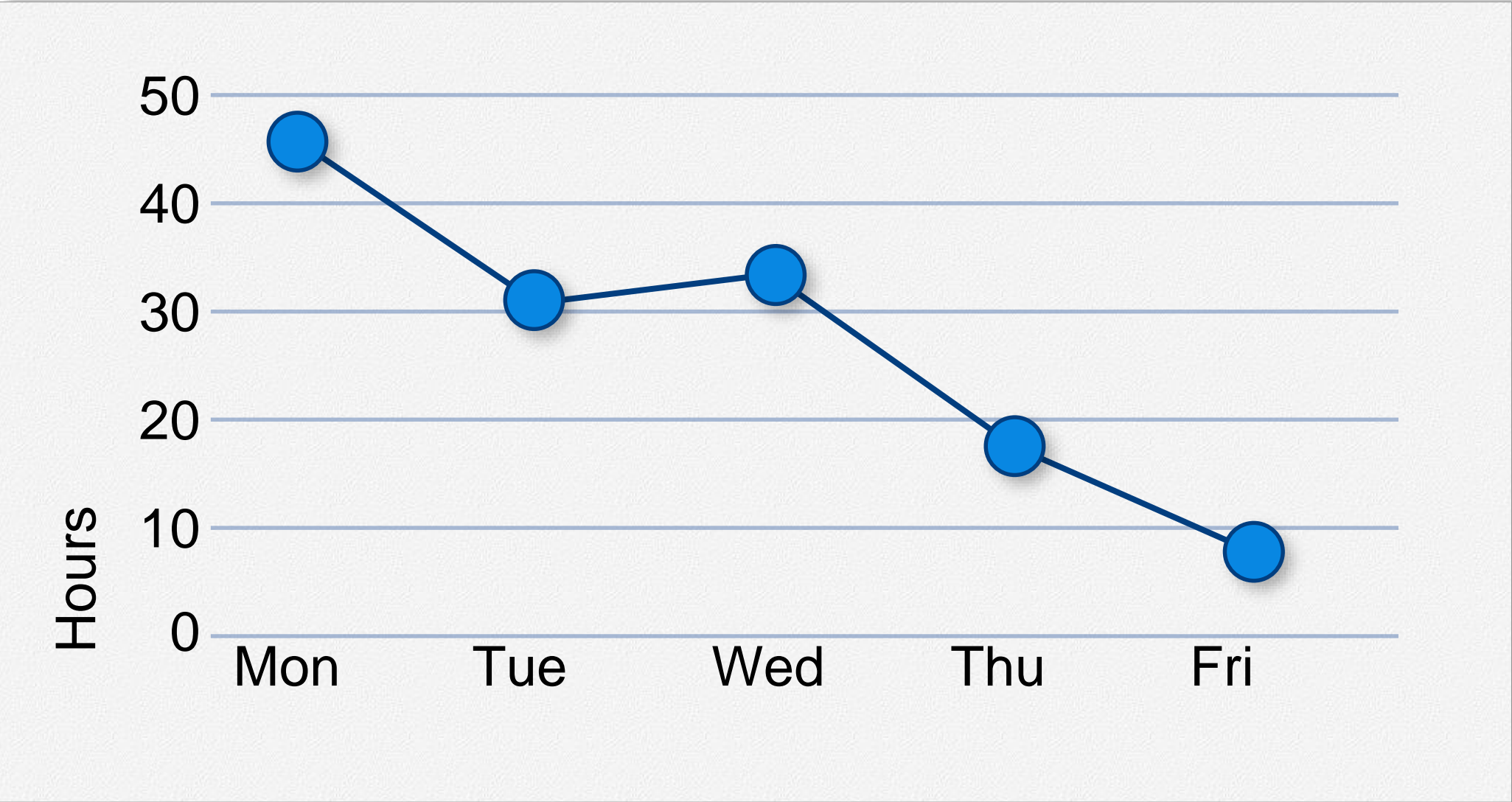
Managing the Sprint Backlog

- Individuals sign up for work of their own choosing
 - Work is never assigned
- Estimated work remaining is updated daily
- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known

A Sprint Backlog

Tasks	Mon	Tue	Wed	Thu	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



Introduction to Software Engineering for Engineers

Lecture-02: Process Models & SCRUM

Part 4: User Stories

Dr.-Ing. Christoph Steup

Motivation

- Predicting a schedule for weeks or even months is impossible
- Thus
 - make decision on information available, but do it often
 - Instead of making all-encompassing decisions spread decision-making across the project

Samples

As a user, I want to reserve a hotel room.

As a vacation traveler, I want to see photos of the hotels.

As a user, I want to cancel a reservation.

As a frequent flyer, I want to rebook a past trip so that I save time booking trips I take often.

What about Details?

- *As a user, I can cancel a reservation*
 - Does the user get a full or partial refund?
 - Refund to credit card or site credit?
 - How far ahead a reservation must be cancelled?
 - Is it the same for all hotels?
 - For all site visitors? Can frequent travellers cancel later?
- *Is a confirmation provided to the user?*
 - How?

Details as conditions of satisfaction

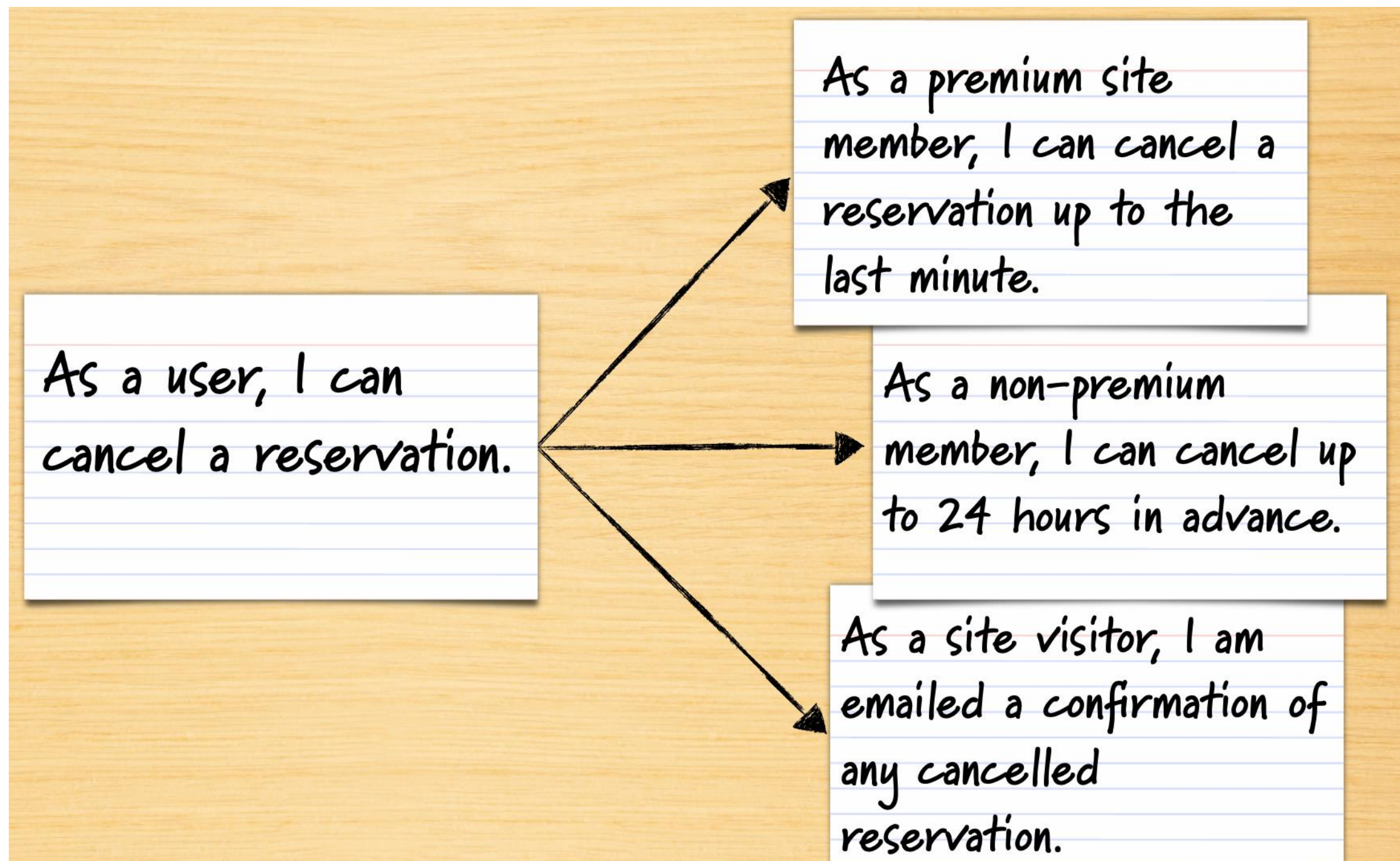
As a user, I can cancel a reservation

- Product owner's conditions of satisfaction can be added to a story
- These are essentially tests

Also called
Acceptance Criteria

- Verify that premium member can cancel the same day without a fee
- verify that an email confirmation is sent
-

Alternative: Details added in smaller sub-stories



Techniques can be combined

- Approaches are not mutually exclusive
- Write stories at an appropriate level
- By time of implementation/realization, each story will have conditions of satisfaction associated with it

Useful Terminology

Theme

A collection of related user stories

Hence, several themes may form an epic

Epic

A large user story (usually to be decomposed)

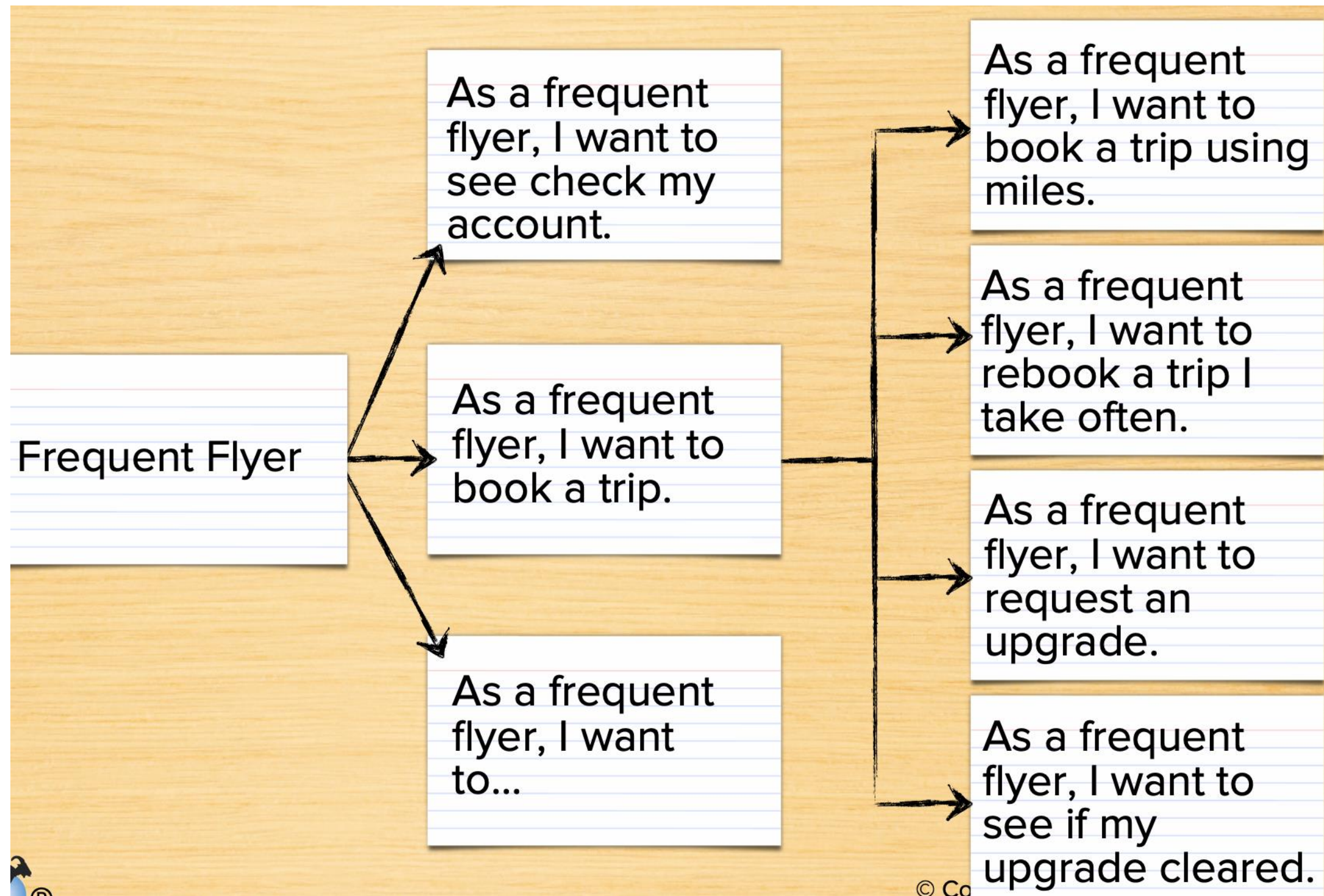
Writing User Stories

“As a <user role>,
I <want/need/can/etc> <goal>
so that <reason>”

Story-Writing Workshops

- Includes the whole team (external stakeholders possible)
- Not done every sprint
- Instead, try to write as many stories as possible at such a workshop
 - some will be “implementation-ready”
 - Others will be epics
- No prioritization at this point

Start with Epics and iterate



Summary

- **Software Engineering**
 - Motivation & Terminology
- **Process Models**
 - Waterfall
 - V-Model
 - Prototype-based
 - Iterative Models
- **Agile Models**
- **Blog Articles**
 - Team Presentation