

Lecture 15 - recap

Kari Systä, 5.5.2014

End seminar (gala!)

- (2nd) traditional end seminar 16.05. at 1300 in SE203 (initial info)
- Participants collect one "point"
- Best project with win a prize!
- Each presentation
 - 2.5 minutes demo of the game (or video)
 - 2.5 minutes presentation "lessons learned from Software Engineering and process point of view"
- If you suspect that your computer does not co-operate with video projector, bring a memory stick

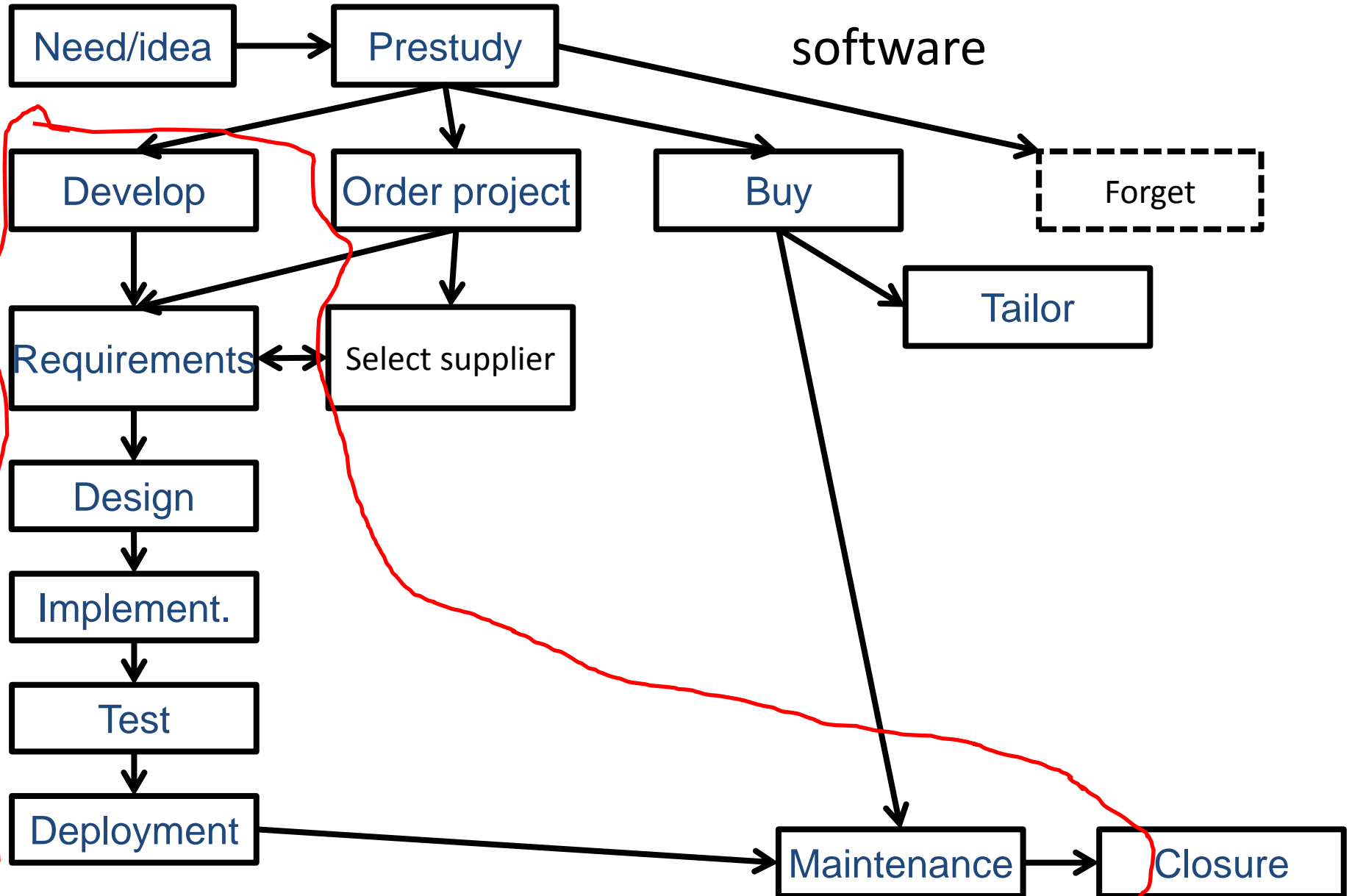
About the timetable

- 16.05.
 - Assignment deadline 1300
 - Reservations of sprint 4 review meetings in IDLE by 1600
 - End gala at 1315 in SE203
- Timing of exams
 - First exam is May 15!
 - POP has information about 23.06. for open university, but that will move towards end of July.
 - There will be “official” exams in the fall (dates not confirmed, yet).

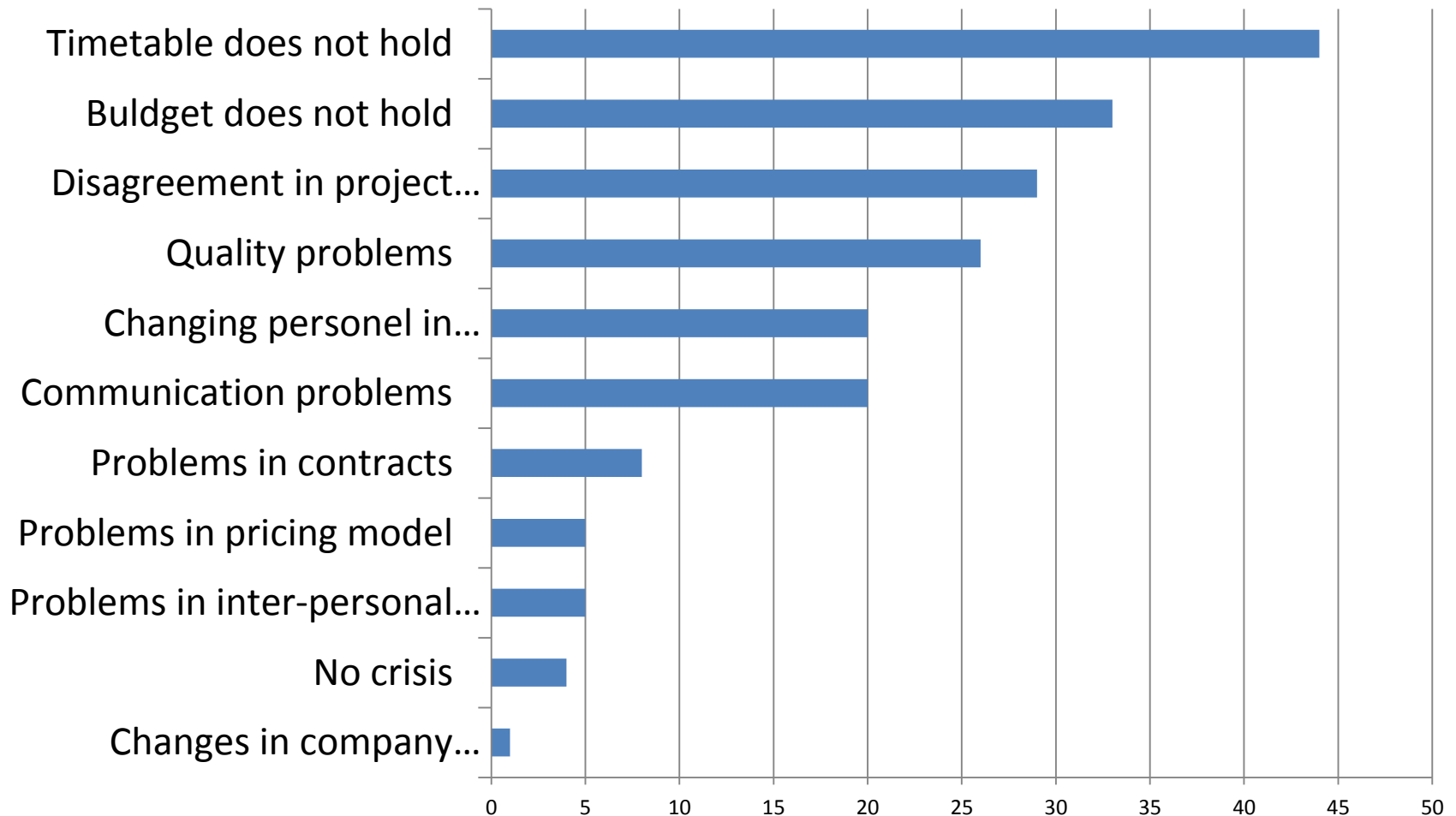
A few definitions

- “ Software engineering may be defined as the systematic design and development of software products and the management of the software process”
 - Mills, H.D. , IBM Systems Journal
Vol19 , Issue: 4, 1980
- “Software Engineering is the study and application of engineering to the design, development, and maintenance of software.”
 - http://en.wikipedia.org/wiki/Software_engineering

From needs to
software



Problems – for customer point of view

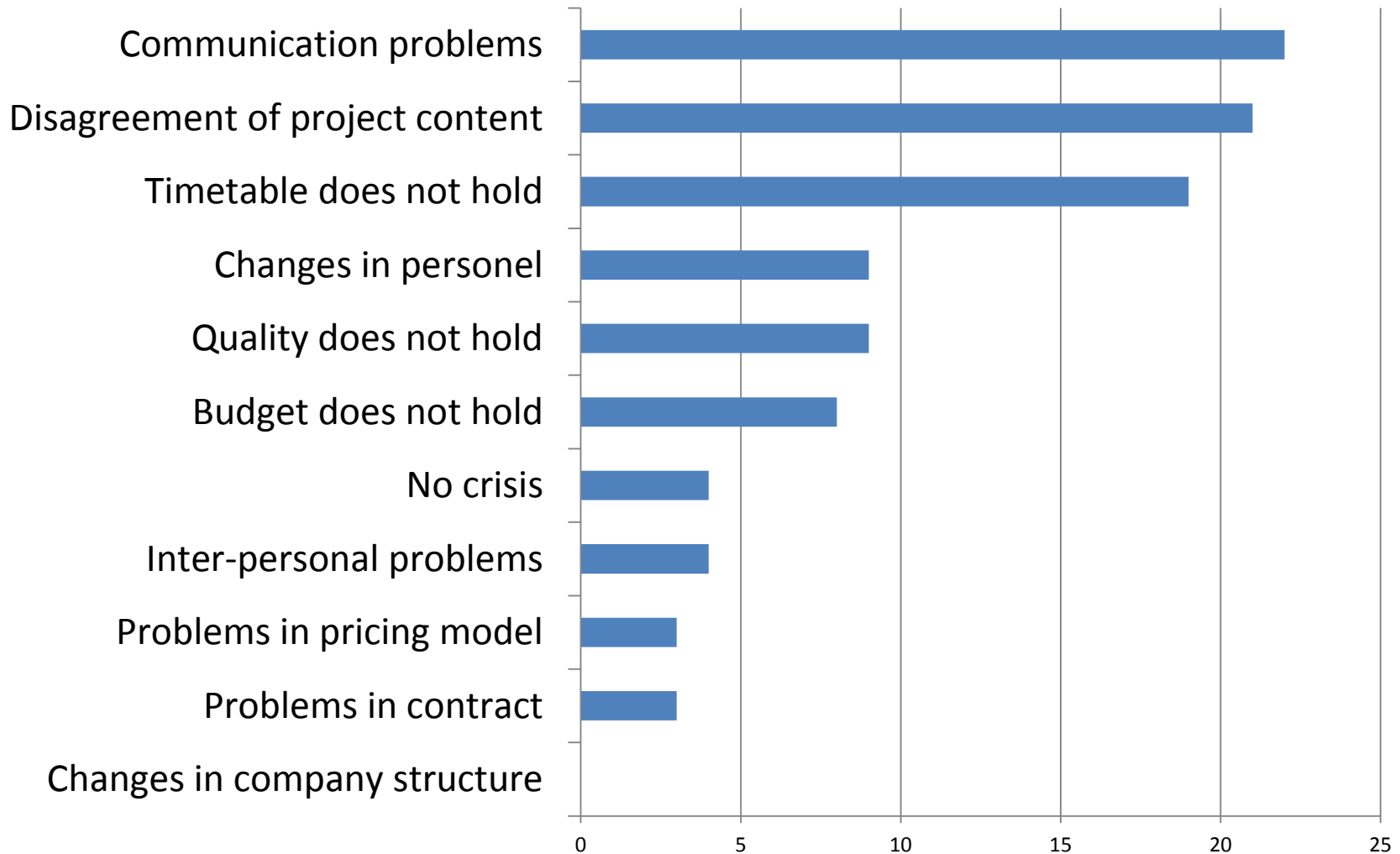


Lähde: tietotekniikan liiton, ohjelmistoyrittäjien ja Celkee OY:n tutkimus

05.05.2014

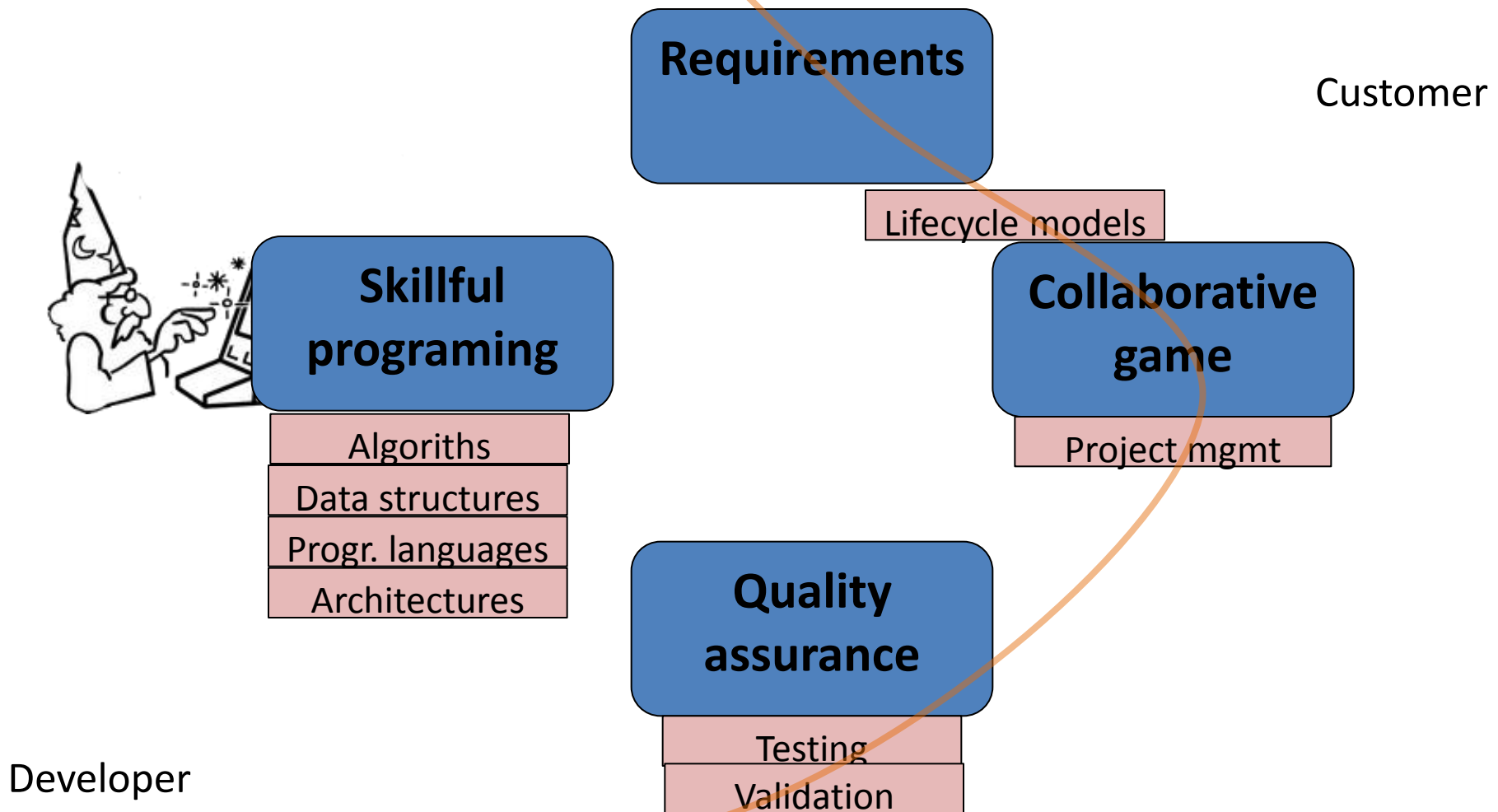
TIE-21100/6; Kari Systä

Problems – provider view

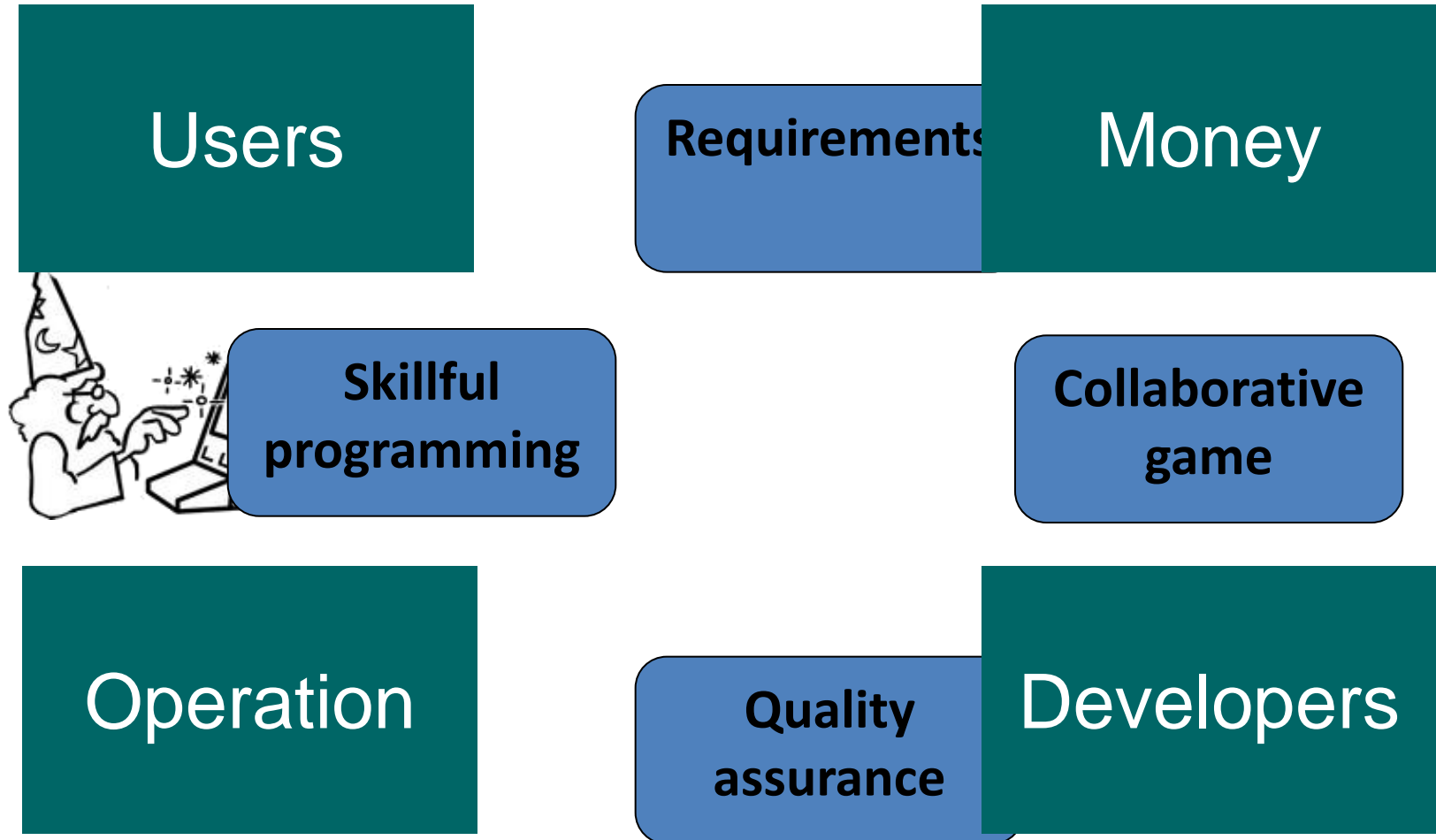


Lähde: tietotekniikan liiton, ohjelmistoyrittäjien ja Celkee OY:n tutkimus

What is software engineering



Extended view



What should be read

- Chapter 1 in both books

Life-cycle models

<http://www.ics.uci.edu/~wscacchi/Papers/SE-Encyc/Process-Models-SE-Encyc.pdf>

- A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed.
- Guideline to organize, plan, staff, budget, schedule and manage software project work over organizational time, space, and computing environments.
- Prescriptive outline for what documents to produce for delivery to client.
- Basis for determining what software engineering tools and methodologies will be most appropriate to support different life cycle activities.
- Framework for analyzing or estimating patterns of resource allocation and consumption during the software life cycle (Boehm 1981)
- Basis for conducting empirical studies to determine what affects software productivity, cost, and overall quality.

Software Process Models

- In contrast to software life cycle models, software process models often represent a networked sequence of activities, objects, transformations, and events that embody strategies for accomplishing software evolution.
- Such models can be used to develop more precise and formalized descriptions of software life cycle activities.

Life-cycle model and process model

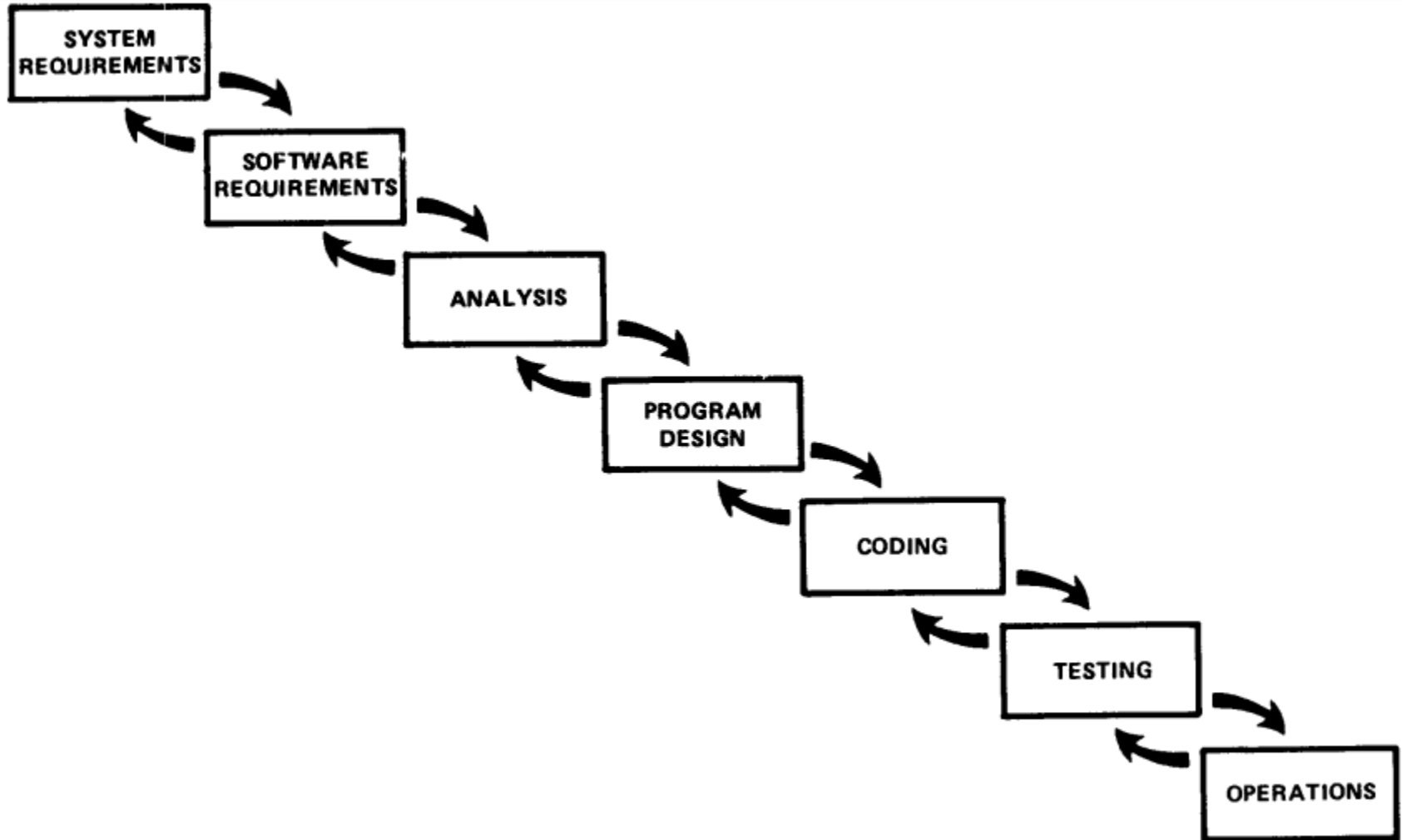
- Often seen as synonyms. For example wikipedia.org/wiki/Software_development_process writes:
 - A software development process, also known as a software development life-cycle (SDLC), is a structure imposed on the development of a software product.
- https://ece.uwaterloo.ca/~se464/06ST/lecture/02_life-cycle-models.pdf:
- Lifecycle models: Phases in the life of an artifact, e.g., a system
- Process models: Activities performed on artifacts, e.g.,
- development activities

Don't PANIC

- We mostly do not care about the possible difference in this course

Royce, 1970

(<http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>)

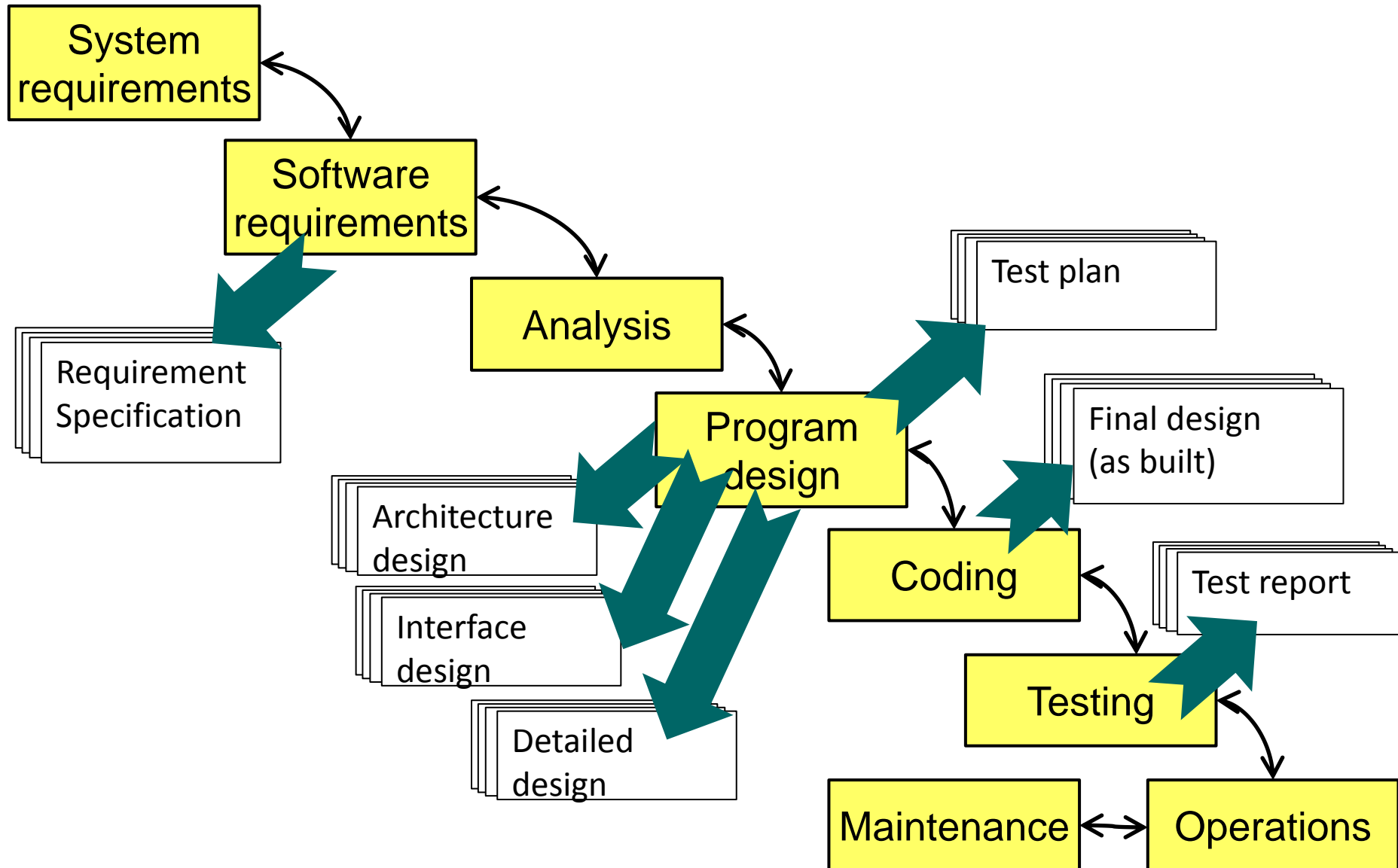


Principles of waterfall

- Waterfall is often understood as one-directional flow, but Royce considered iterations as a crucial part of the model. (mainly between consecutive steps)
- Waterfall is a **plan-driven** approach
- Move from step to next is a decision and often involves reviews, re-planing, budget decisions etc.
- Proper design and plan prevents extra work (and cost) in next steps
- The earlier the mistake is done, the more expensive it is
- Waterfal is consistent with other engineering processes

Documentation is a crucial part of waterfall

(one possible example)



Problems with waterfall

- Does not support division of the software to distinct stages
 - It is difficult to take out and use partial functionality
- Difficult to respond to changing customer requirements
- Management and motivation challenges of developers
 - Does not utilize full talent and motivation of talented and highly trained software developers
 - Does not show trust and empowerment
- Usually, waterfall is considered suitable for projects where
 - Requirements can be known in advance
 - Milestone reviews and audits are needed for example by security standards,

Precursor of interative models: Spiral Model

(picture from: <http://www.sei.cmu.edu/reports/00sr008.pdf>)

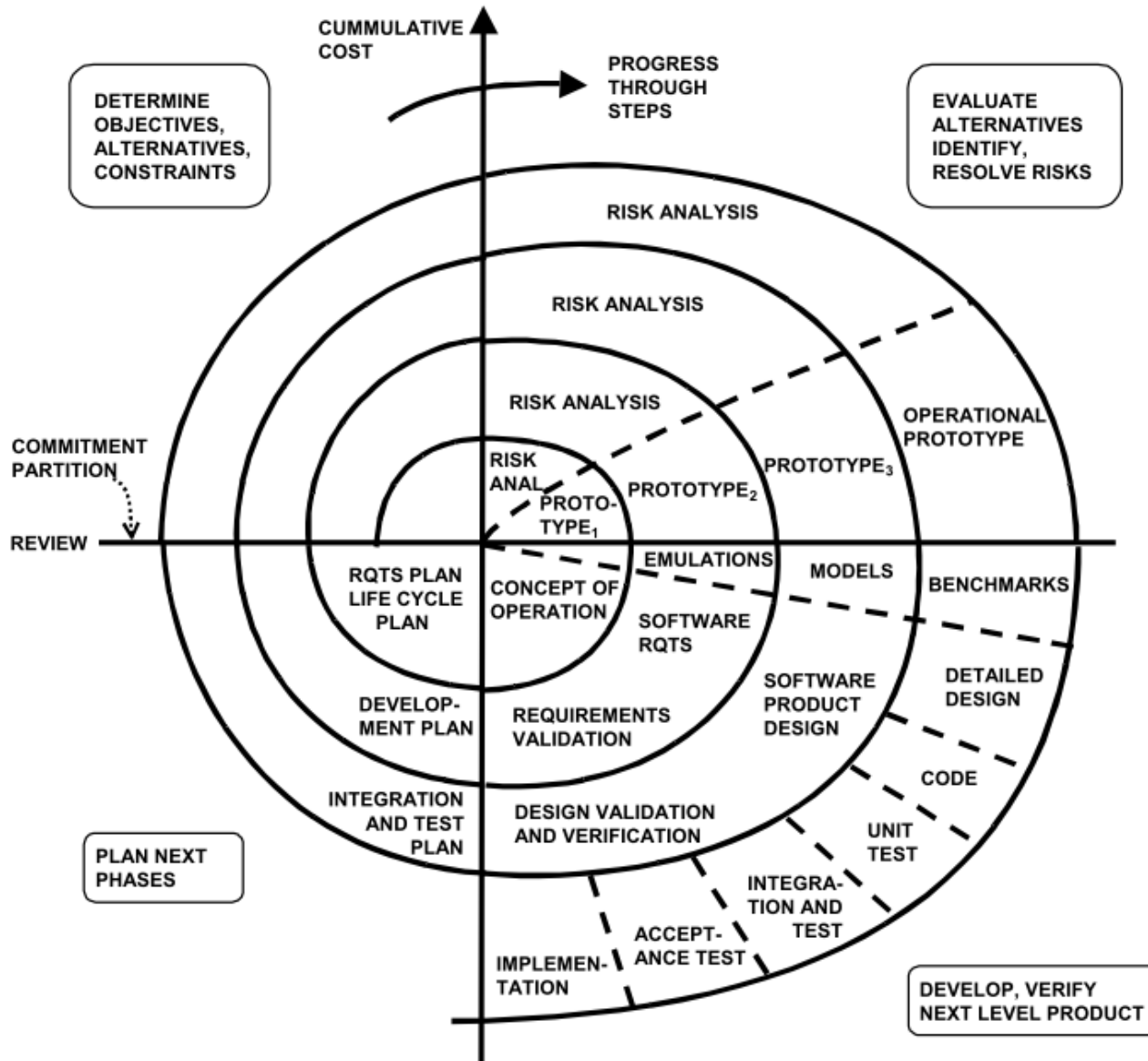
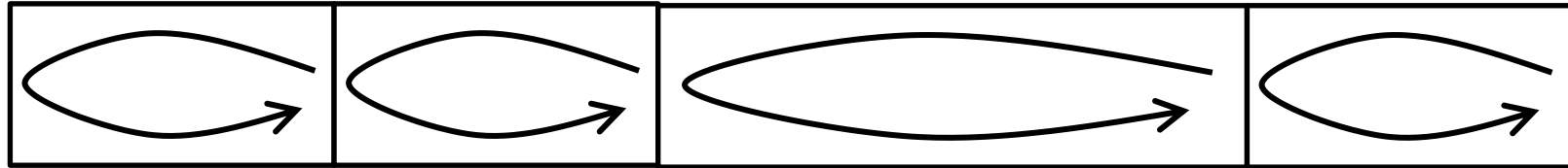


Figure 1: Original Diagram of Spiral Development

Phases in RUP



Inception

Elaboration

Construction

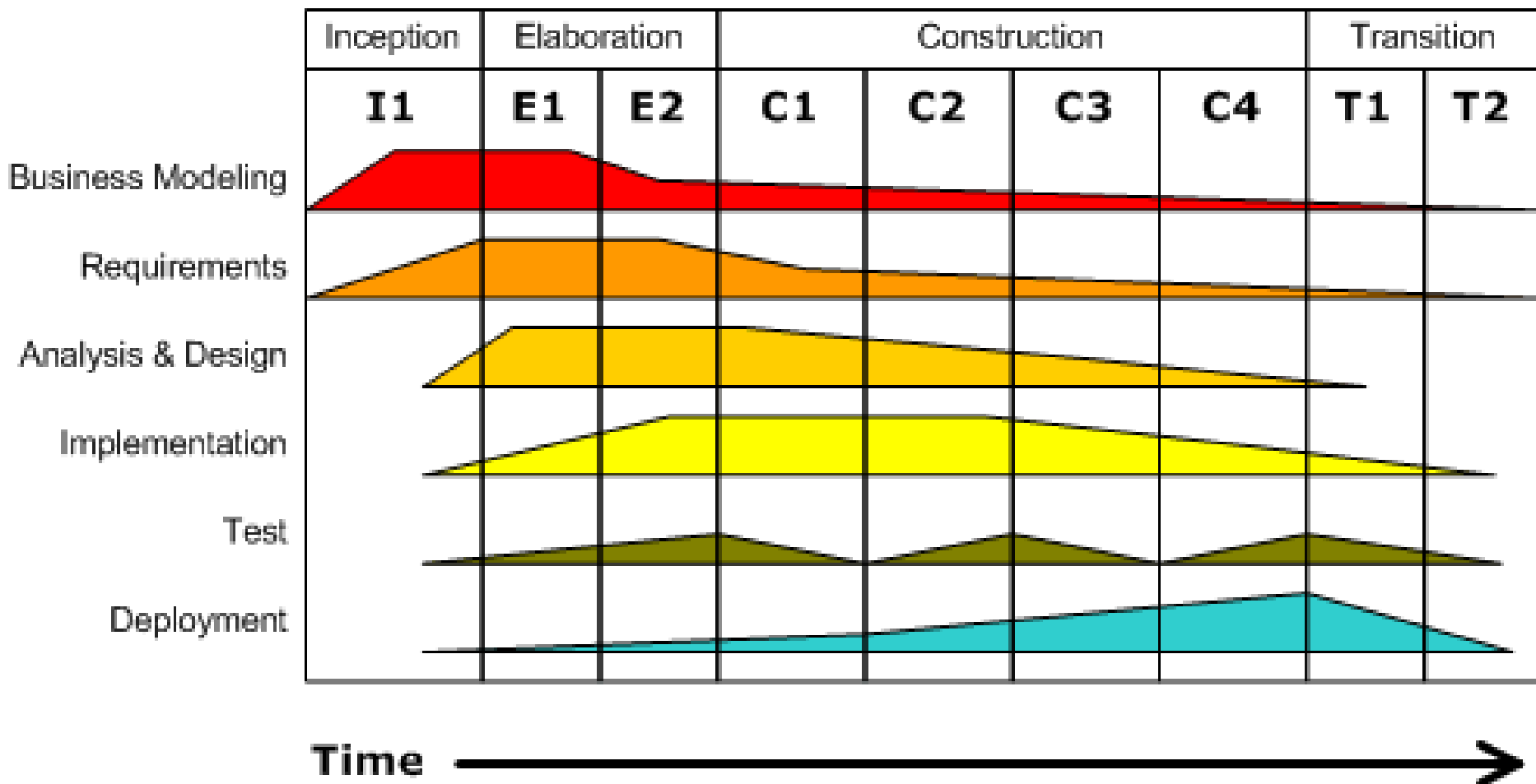
Transition

- Inception: business case and stakeholders
- Elaboration: spec, design, plan
- Construction: "the real work"
- Transition: to users

Iterative model: RUP

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Agile –manifesto

- February 2001
- 17 "inventors"
- 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.

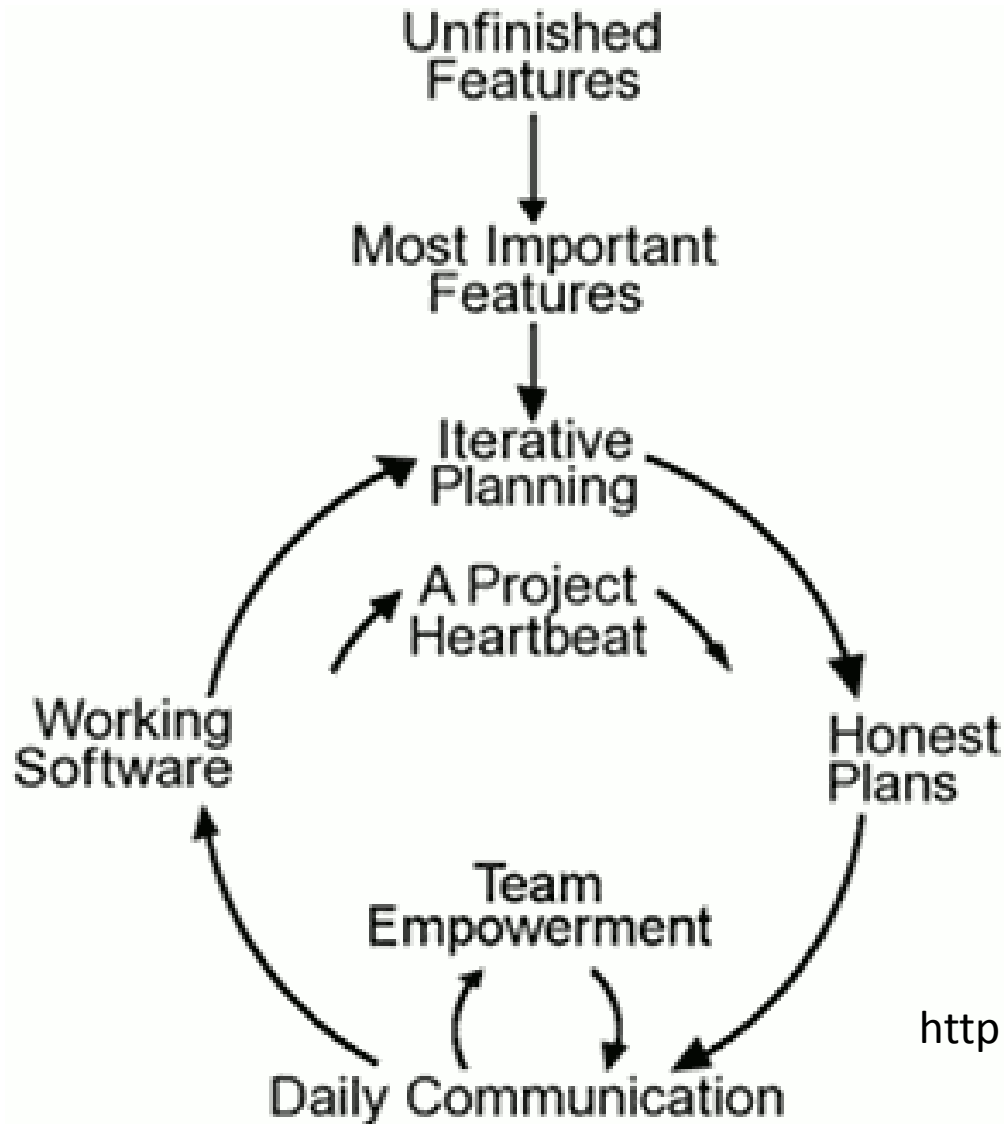
Five principles of Agile

Customer involvement	Through the project. Provide and prioritize requirements, evaluate iterations
Incremental delivery	Customer specifies the increments
People not process	Skill recognized and exploited; Team should decide on ways of working
Embrace change	Plan and design for change
Maintain simplicity	Both in process and software

Problems in realizing Agile

- Getting customer commitment and trust is difficult
- Some team members do not have suitable personalities
- Prioritizing changes is difficult – especially if there are many stakeholders
- Maintaining simplicity requires extra work
- Cultural changes through the company
(Agile training should start from management)

XP (Extreme programming)



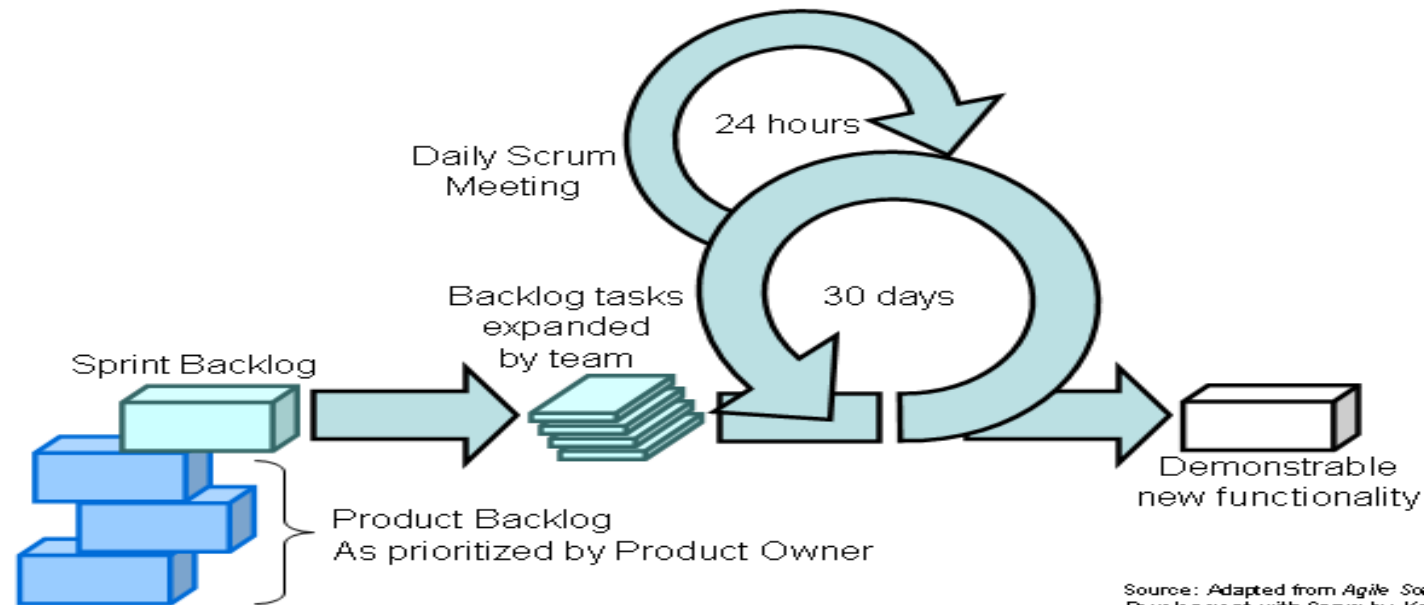
<http://www.extremeprogramming.org/>

XP Practices

Practice/principle	Description
Incremental planning	
Small releases	Minimum useful is implemented first; frequent releases
Simple design	Spend enough time in design
Test-driven development	Test written before code, automated tests
Refactoring	All team members should refactor code to keep simple and maintainable
Pair programming	Check each others' work; support
Collective ownership	No islands of responsibilities; every body can change everything
Continuous integration	Whenever something is ready it is integrated; always test
Sustainable pace	Large amounts of overtime is not sustainable
On-site customer	Continuous access to customer

Scrum

- Framework for agile and iterative development
- Jeff Sutherland, John Scumniotales, and Jeff McKenna OOPSLA 95



Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

XP vs Scrum

- XP has typically shorter iterations (1-2w instead of 2-4w)
- Scrum does not allow changes into sprints
- XP is work in strict priority order
- Scrum does not prescribe any engineering practices
- Scrum focuses more on management aspects

Principles of Lean SW Development

Wikipedia

- Eliminate waste
- Amplify learning
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build integrity in
- See the whole

www.poppendieck.com

- Eliminate Waste
- Keep getting better
- Learn first
- Deliver fast
- Energize workers
- Build quality in
- Focus on customers

Kanban

(wikipedia based on David Anderson. Kanban – Successful Evolutionary change for your Technology Business. Blue Hole Press, April 2010)

Visualise

- Visualising the flow of work and making it visible is core to understanding how work proceeds. Without understanding the workflow, making the right changes is harder.
- A common way to visualise the workflow is to use a card wall with cards and columns. The columns on the card wall representing the different states or steps in the workflow.

Limit WIP

- Limiting work-in-process implies that a pull system is implemented on parts or all of the workflow.
- The critical elements are that work-in-process at each state in the workflow is limited and that new work is “pulled” when there is available capacity within the local WIP limit.

Manage flow

- The flow of work through each state in the workflow should be monitored, measured and reported. By actively managing the flow the continuous, incremental and evolutionary changes to the system can be evaluated to have positive or negative effects on the system.

Requirements for Exam

- Know what and why we have process/life-cycle models
- Understand Kanban and Lean
- Know basic principles of Waterfall, XP, and Kanban – more detailed understanding of Scrum is assumed

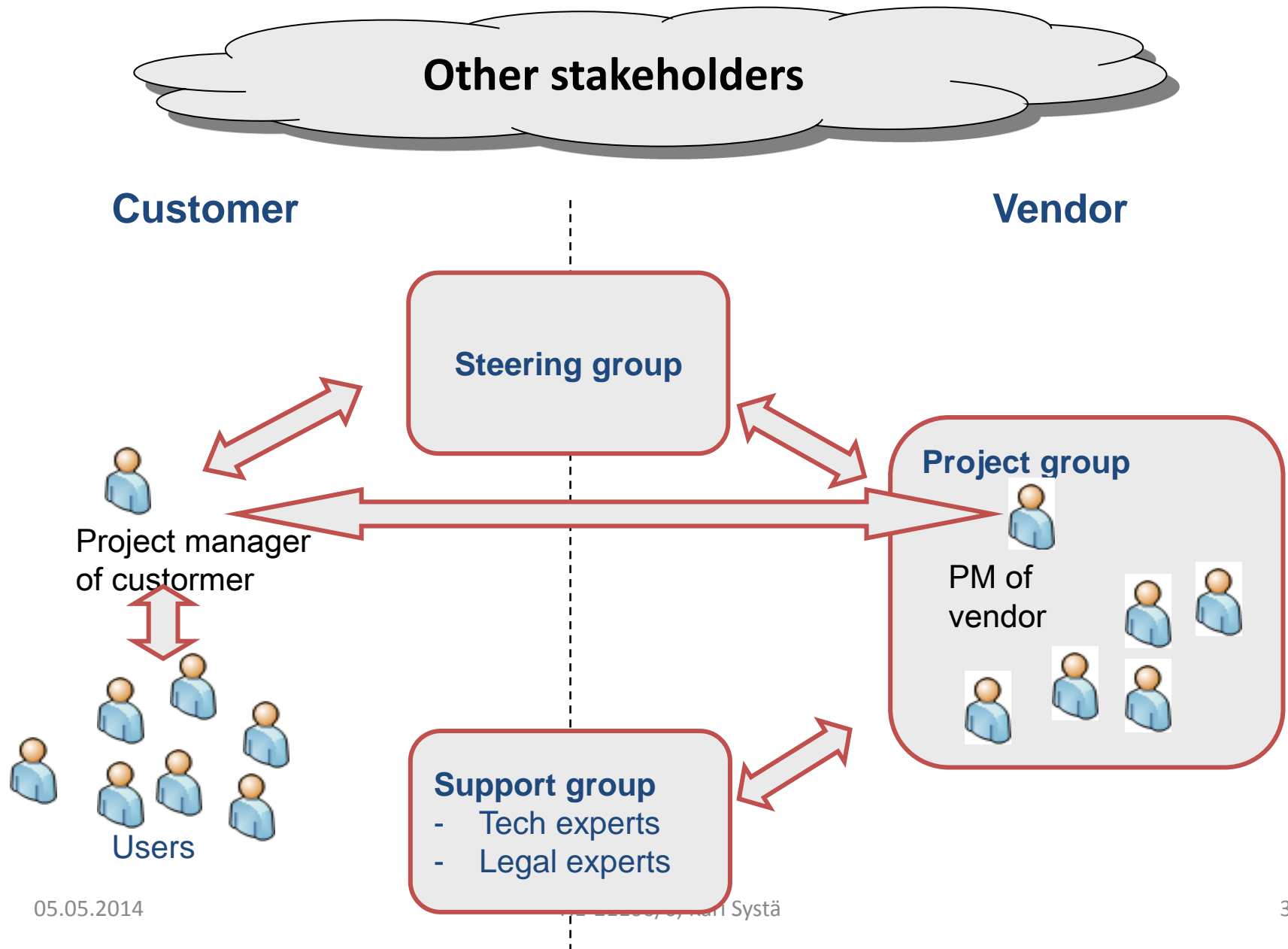
What should be read

- Chapter 1 in both books
- Chapters 2-3 in Sommerville;
Chapter 2 in Haikala&Mikkonen
Scrum Guide: <https://www.scrum.org/scrum-guide>
<http://leankit.com/kanban/what-is-kanban/>

Project management aims at

- Deliver SW to customer at the agreed time
- Keep overall costs within the budget
- Deliver software that meets customer's expectations
- Maintain spirit and performance of the team

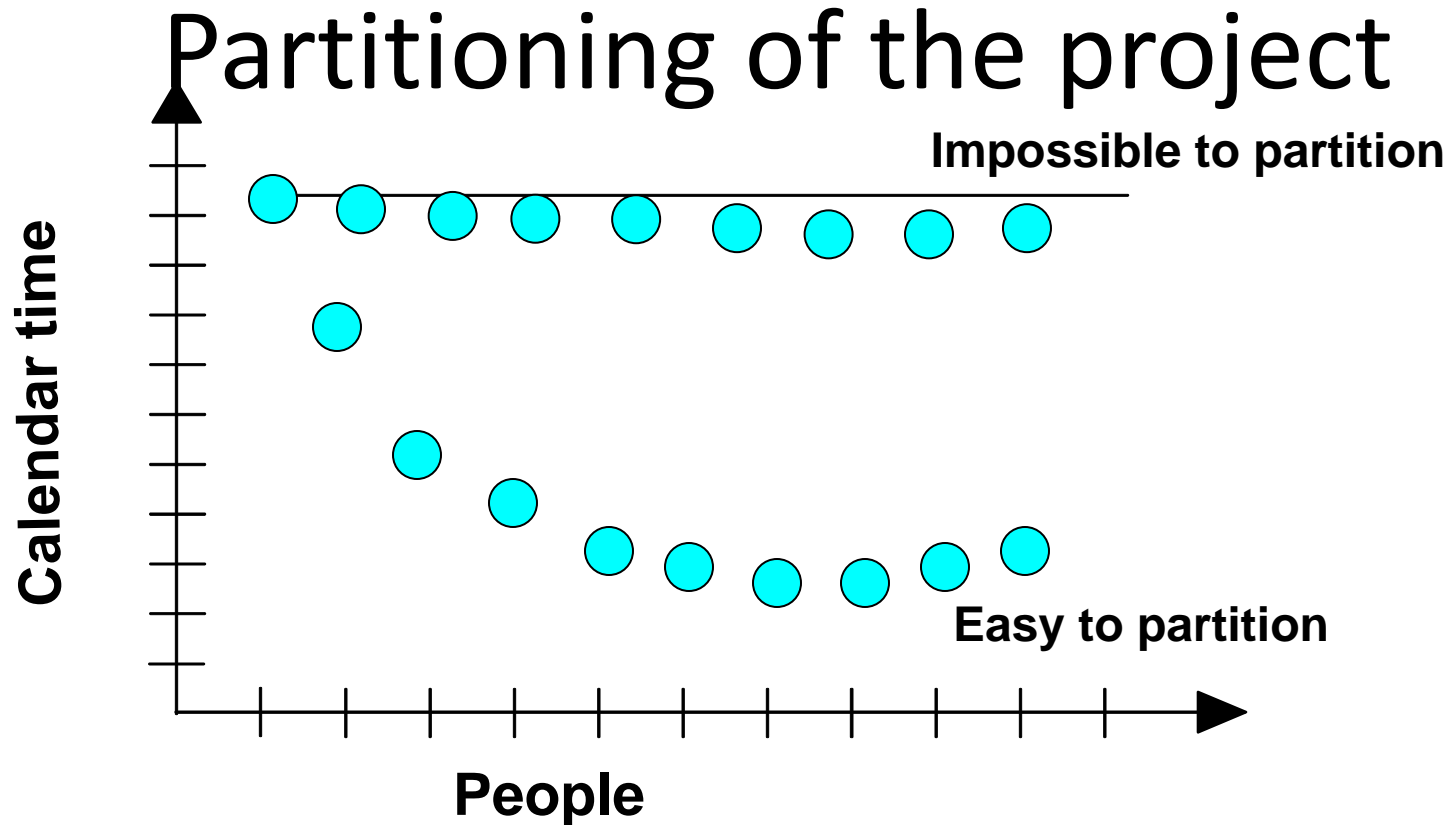
Project organization (traditional)



Project planning

- Organization
- Target setting
- Risk analysis
- Selection of technologies, methods, practices, tools
 - For instance if agile is selected
- Support (documentation, quality assurance, product management)
- Splitting and phasing
(WBS = Work Breakdown Structure)
- Worktime estimation
- Resource availability and time table
 - Developers, external experts
 - Special tools
- Budget, financing, funding
- Success criteria



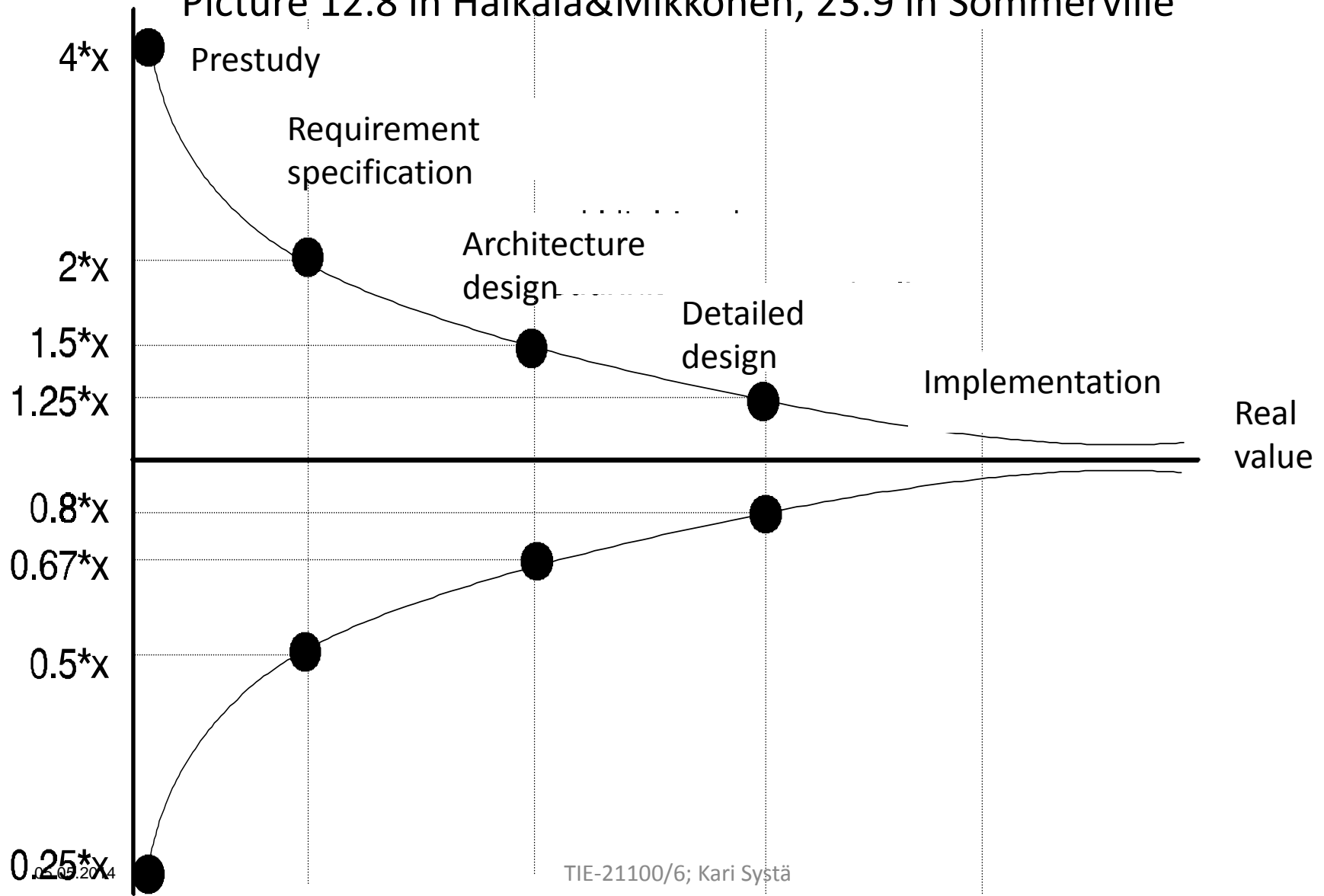


What one programmer can do in a day,
two programmers can also do in day.

A man's got to know his limitations. -- Dirty Harry.

Estimates improve as the project progresses

Picture 12.8 in Haikala&Mikkonen, 23.9 in Sommerville



Plan-driven scheduling

- Start with constraints.
- Make work breakdown structure
 - Including dependencies
- Estimate efforts for tasks (e.g. days) – best to take input from several people
- Check availability of resources
- Put to calendar
 - Who, when
- Project planning tools help in mechanical work
 - The challenge is the “guessing” in advance



Methods for effort estimation

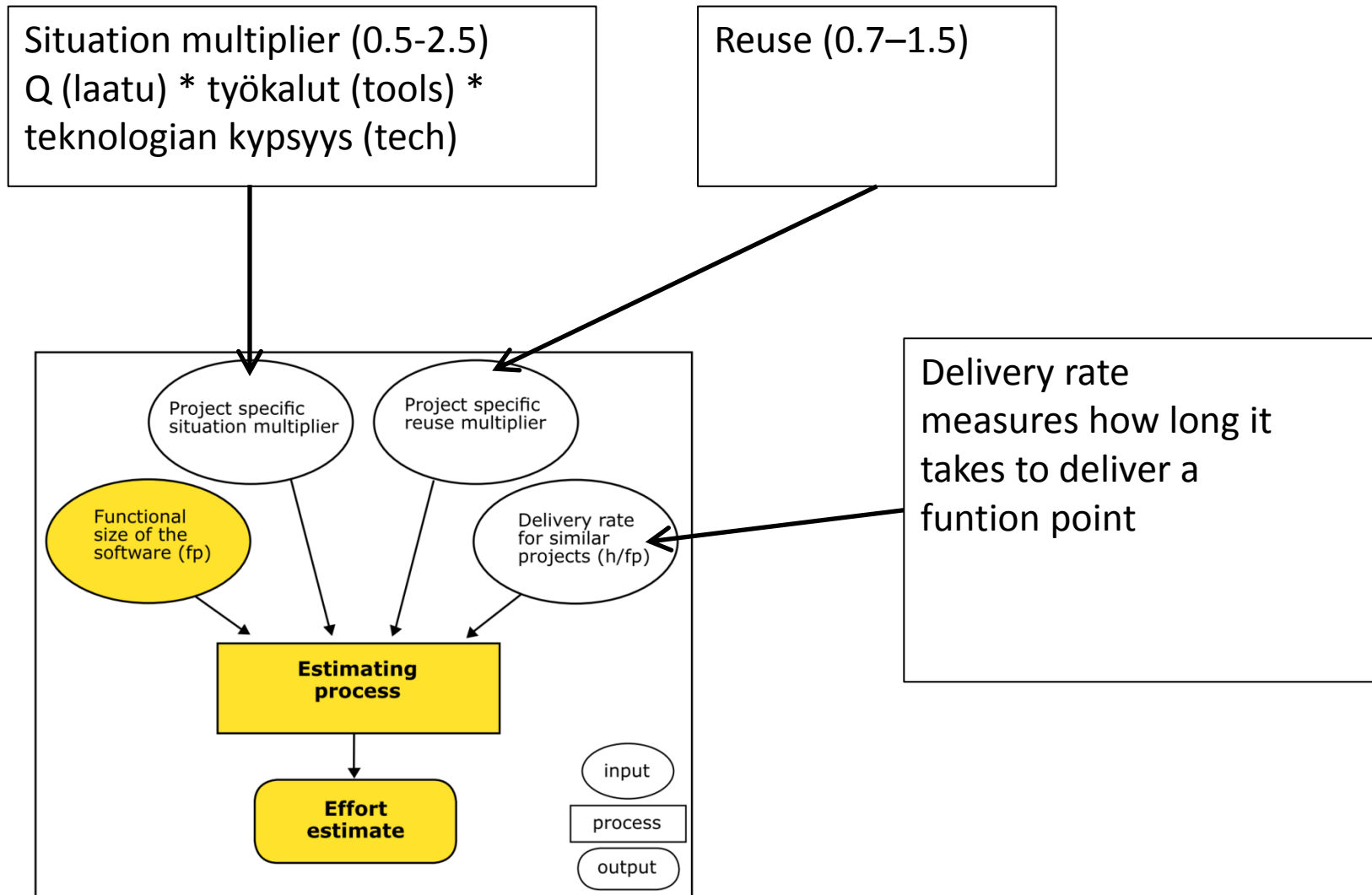
- Planning poker
- Cocomo (constructive cost model) – will be described today
- FPA (function point analysis)
- Use your experience and historical data
 - Collect information for next projects



Planning poker - how

- Participants get a deck of cards with numbers
 - Often Fibonacci series 0, 1, 2, 3, 5, 8, 13...
- Somebody present the task to be estimated
- Everybody shows a card that describes his or her opinion about the effort
 - The cards are shown synchronously (at the same time)
- Those who are different from common opinion defend their view
- As long as there are different opinions repeat

Estimation based of Function points (FiSMA)



Requirements for Exam

- Role of project planning
 - difference to specification & design
 - relation to different life-cycle models
- Principles of effort estimation

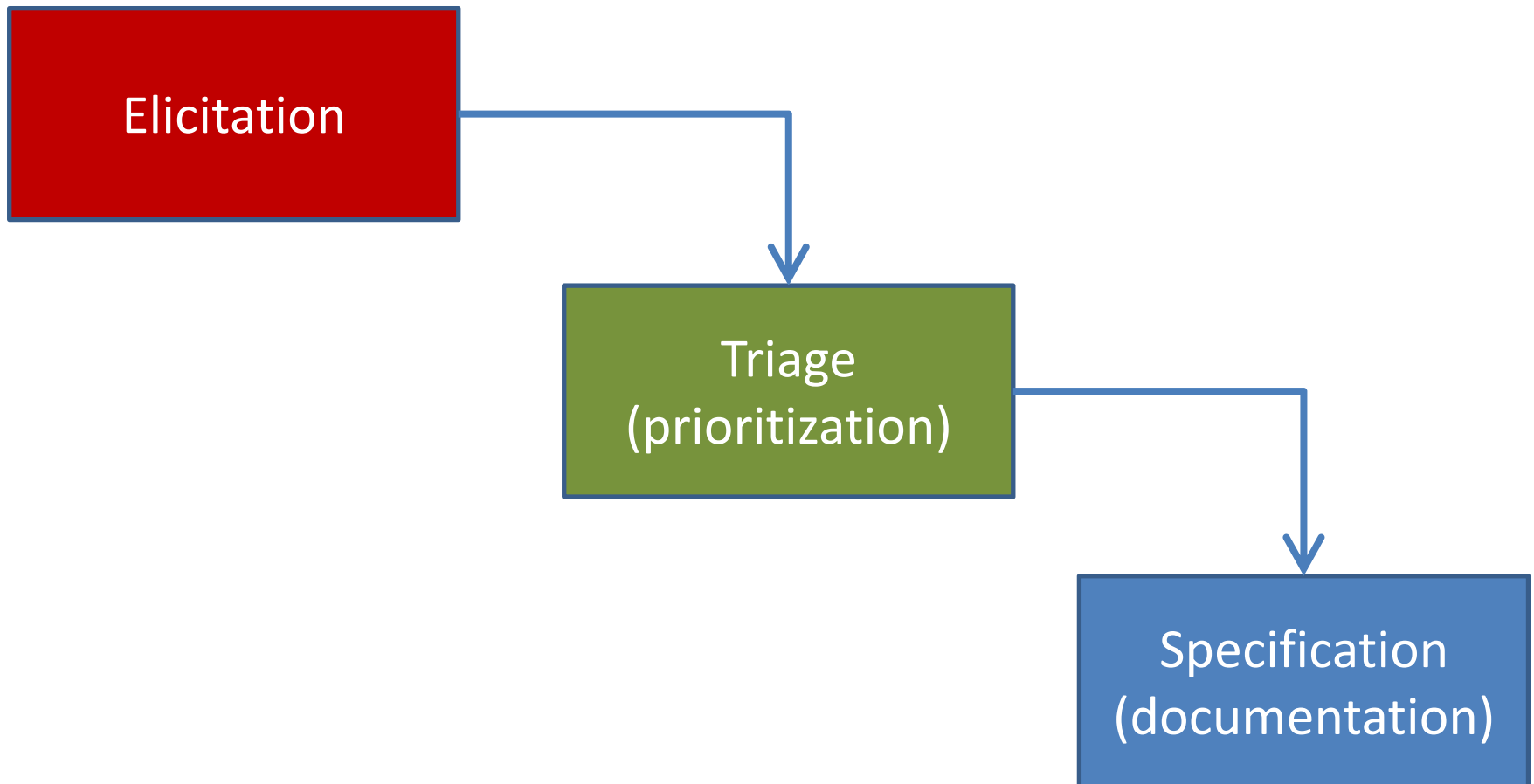
What should be read

- Chapter 1 in both books
- Chapters 2-3 in Sommerville;
Chapter 2 in Haikala&Mikkonen
Scrum Guide: <https://www.scrum.org/scrum-guide>
<http://leankit.com/kanban/what-is-kanban/>
- Chapters 22-23 in Sommerville;
Chapter 12 in Haikala&Mikkonen

Requirements vs constraints

- **Functional requirement**, for example the software has support for spell checking.
- **Non-functional requirement**, for example User interface follows the UI-guideline of XXX or installation can use at most 5MB disk space.
- **Constraints**, for example the software has to implemented in Windows-operating system in C++-language.

Requirements management



What should be read

- Chapter 1 in both books
- Chapters 2-3 in Sommerville;
Chapter 2 in Haikala&Mikkonen
Scrum Guide: <https://www.scrum.org/scrum-guide>
<http://leankit.com/kanban/what-is-kanban/>
- Chapters 22-23 in Sommerville;
Chapter 12 in Haikala&Mikkonen
- Chapter 4 in Sommerville;
Chapters 3-4 in Haikala&Mikkonen;
<http://www.softwareengineering-9.com/> (CH 4)
-

Definitions

- IEEE 610.12-1990:

inspection = a static analysis technique that relies on visual examination of development products to detect errors, violations of development standards, and other problems. Types include code inspection; design inspection.

review = a process or meeting during which a work product, or set of work products, is presented to project personnel, managers, users, customers, or other interested parties for comment or approval. Types include code review, design review, formal qualification review, requirements review, test readiness review.

walk-through = a static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a segment of documentation or code, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems.

Testing

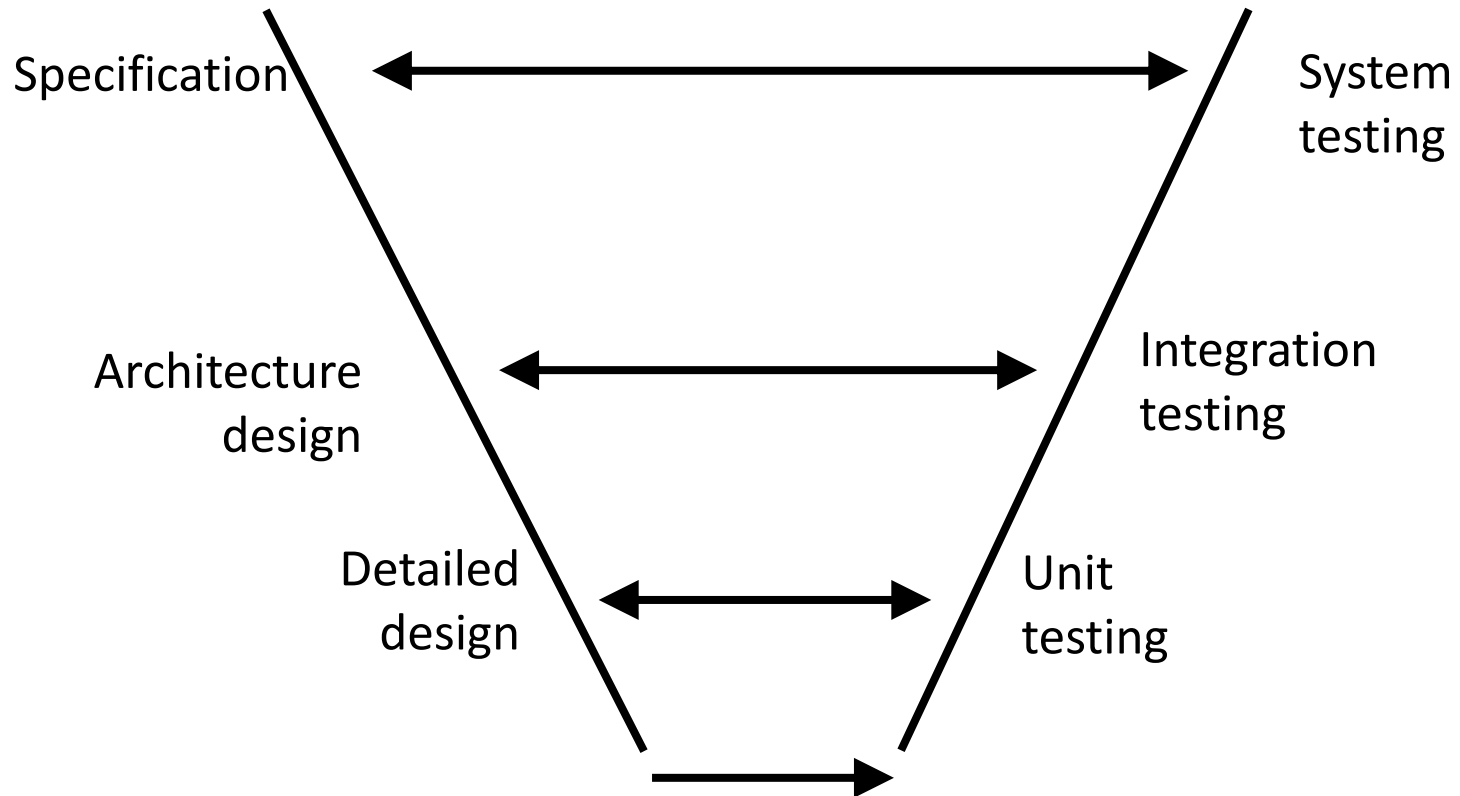
Disjkstra:

- "Testing can only show presence of errors, not their absence"

Inspections and testing both have their roles:

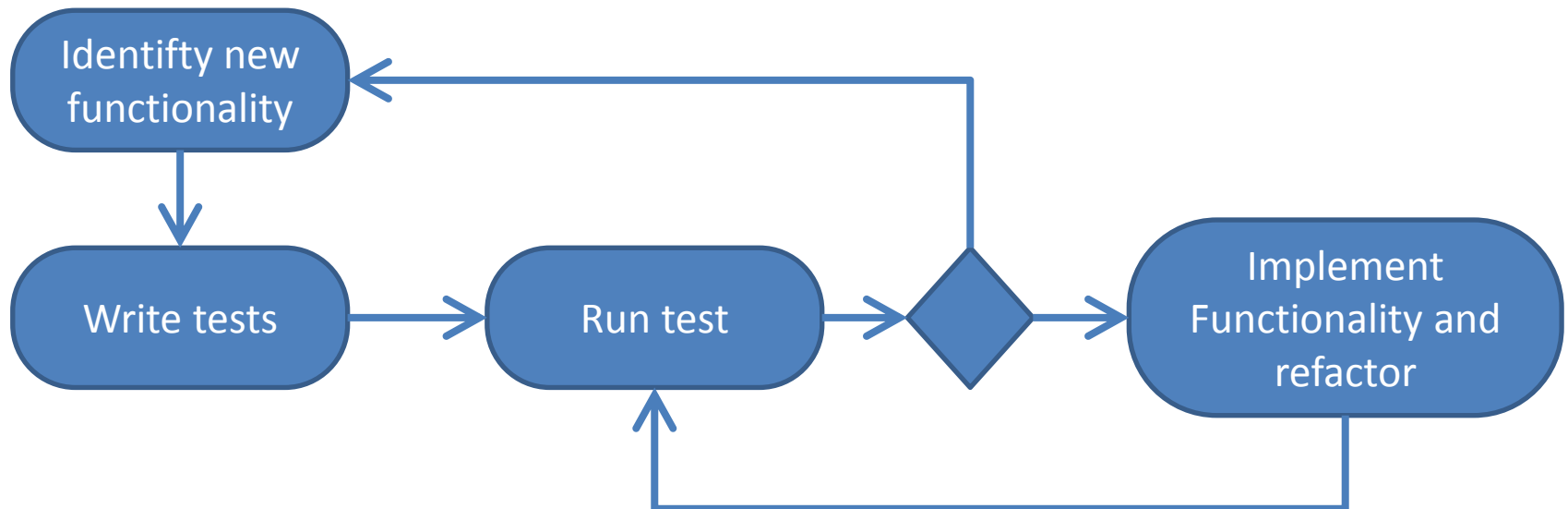
- In testing error can mask other errors
- Incomplete versions can inspected
- Inspections can consider broader set of quality attributes
- Tests are easy to repeat
- Testing can discover issues that relate timing, interactions between different parts of software, ...

V-moded



Testing and Agile

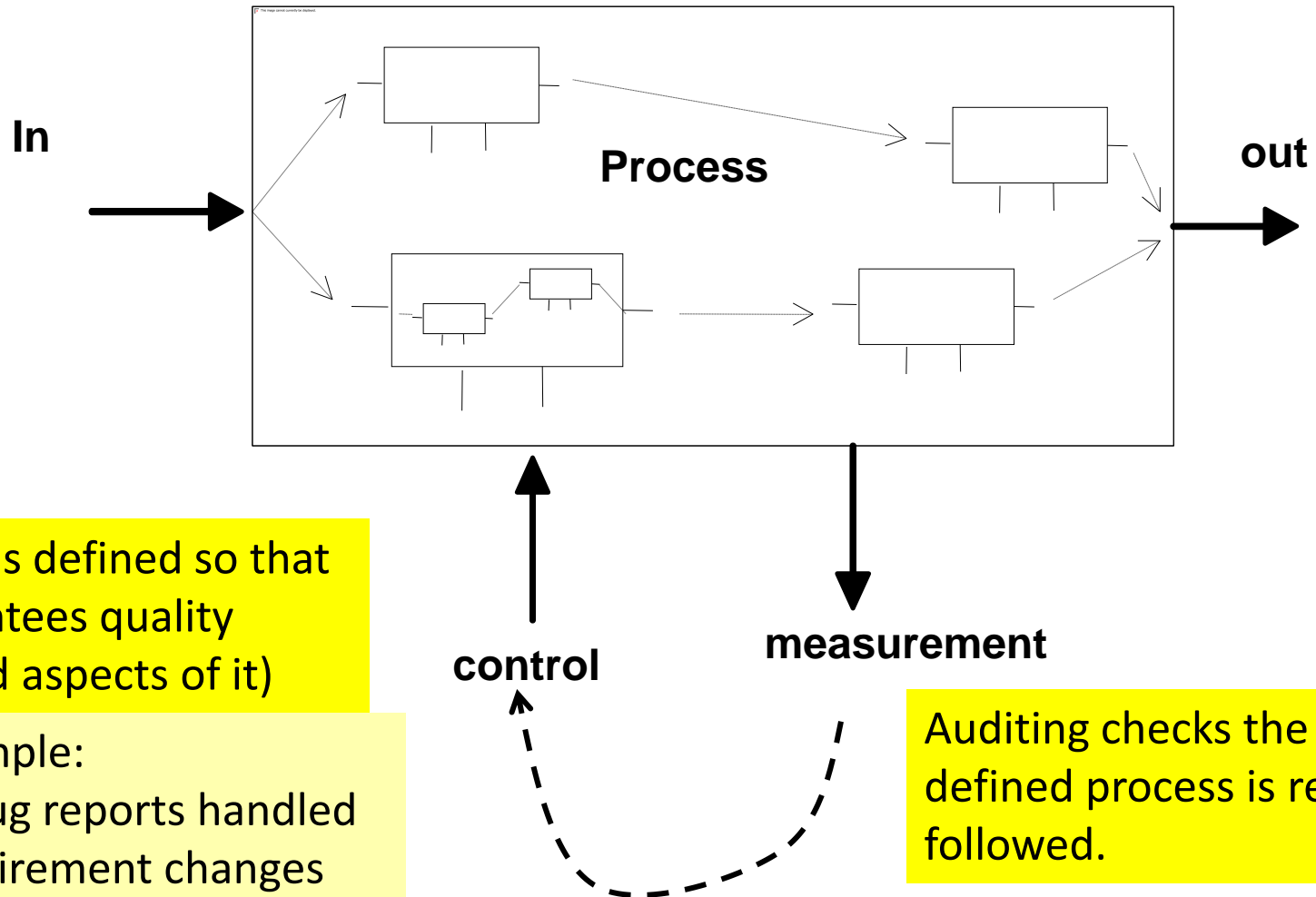
- Testing is a crucial part of agile processes
- The iterative nature means that already changed code need to be tested again
 - => A lot of **regression testing** needed
 - => Test automation is an important tool in Agile
- Test driven development is often used with Agile



What should be read

- Chapter 1 in both books
- Chapters 2-3 in Sommerville;
Chapter 2 in Haikala&Mikkonen
Scrum Guide: <https://www.scrum.org/scrum-guide>
<http://leankit.com/kanban/what-is-kanban/>
- Chapters 22-23 in Sommerville;
Chapter 12 in Haikala&Mikkonen
- Chapter 4 in Sommerville;
Chapters 3-4 in Haikala&Mikkonen;
<http://www.softwareengineering-9.com/> (CH 4)
-

Process view to SW development



Quality management systems (Laatujärjestelmät)

- Documented description of the development process
- Aim is to provide predictable and sustainable quality
- Customer may be interested in this
- Can be audited by a neutral body

An example from [http qmp_2012.pdf](http://qmp_2012.pdf) (process approach)

Key benefits

- Lower costs and shorter cycle times
- through effective use of resources
- Improved, consistent and predictable results
- Focused and prioritized improvement opportunities.

Means:

- Systematically defining the activities necessary to obtain a desired result
- Establishing clear responsibility and accountability for managing key activities
- Analysing and measuring of the capability of key activities
- Identifying the interfaces of key activities within and between the functions of the organization
- Focusing on the factors – such as resources, methods, and materials – that will improve key activities of the organization
- Evaluating risks, consequences and impacts of activities on customers, suppliers and other interested parties.

<http://cmmiinstitute.com/>

- A framework developed by Software Engineering Institute (SEI) of Carnegie-Mellon University (CMU)
- CMMI (or Capability Maturity Model Integration) is a proven approach to performance management with decades of results showing it works.
- Organizations using CMMI have predictable cost, schedule, and quality—business results that serve as discriminators among their competitors.
- CMMI is built with practices and goals seen in thousands of real organizations worldwide. Use these practices and goals to evaluate your own performance and decide what to improve for your own business reasons.

CMMI and ISO9000-series

- CMMI is a set of best practices
ISO9000 set of standards
- CMMI is about software
ISO9000 is for all industries

Requirements for Exam

- Purpose and meaning of quality assurance practices and process approaches
- Basic idea of ISO9000 and CMMI
- Dependability issues (Lecture 9)

What should be read

- Chapter 1 in both books
- Chapters 2-3 in Sommerville;
Chapter 2 in Haikala&Mikkonen
Scrum Guide: <https://www.scrum.org/scrum-guide>
<http://leankit.com/kanban/what-is-kanban/>
- Chapters 22-23 in Sommerville;
Chapter 12 in Haikala&Mikkonen
- Chapter 4 in Sommerville;
Chapters 3-4 in Haikala&Mikkonen;
<http://www.softwareengineering-9.com/> (CH 4)
- Chapters 8, 24, 26 in Sommerville
Chapters 15,16,11 in Haikala&Mikkonen
-

Motivation – becomes difficult very fast

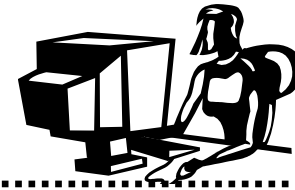
Leenu



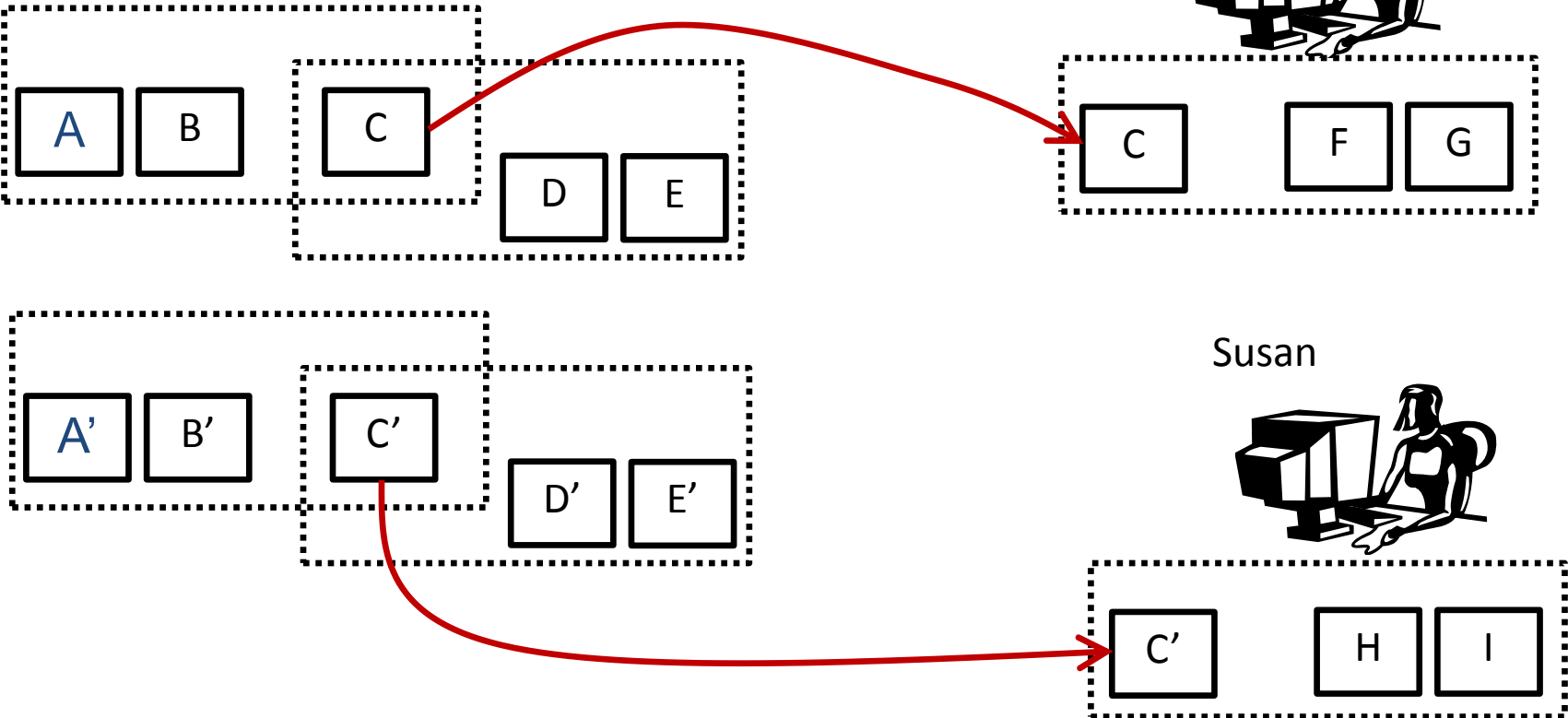
Liinu



Ann



Susan



So, what is configuration management?

- **Change management:** managed way to decide which change ideas to implement and when.
- **Version management:** keep track of multiple versions of components and ensure that changes by different developers do not disturb each other.
- **System building:** collect and assemble correct versions of required components and then compile.
- **Release management:** prepare for external releases and keep track of external releases.

What should be read

- Chapter 1 in both books
- Chapters 2-3 in Sommerville;
Chapter 2 in Haikala&Mikkonen
Scrum Guide: <https://www.scrum.org/scrum-guide>
<http://leankit.com/kanban/what-is-kanban/>
- Chapters 22-23 in Sommerville;
Chapter 12 in Haikala&Mikkonen
- Chapter 4 in Sommerville;
Chapters 3-4 in Haikala&Mikkonen;
<http://www.softwareengineering-9.com/> (CH 4)
- Chapters 8, 24, 26 in Sommerville
Chapters 15,16,11 in Haikala&Mikkonen
- Chapter 25 in Sommerville
Chapter 13 in Haikala&Mikkonen

About the exams

- First exam is May 15!
- POP has information about 23.6 for open university, but that will move towards end of July.
- There will be “official” exams in the fall (dates not confirmed, yet).