

Tarkvara protsessid ja kvaliteet: põhimõisted

Software quality: basics

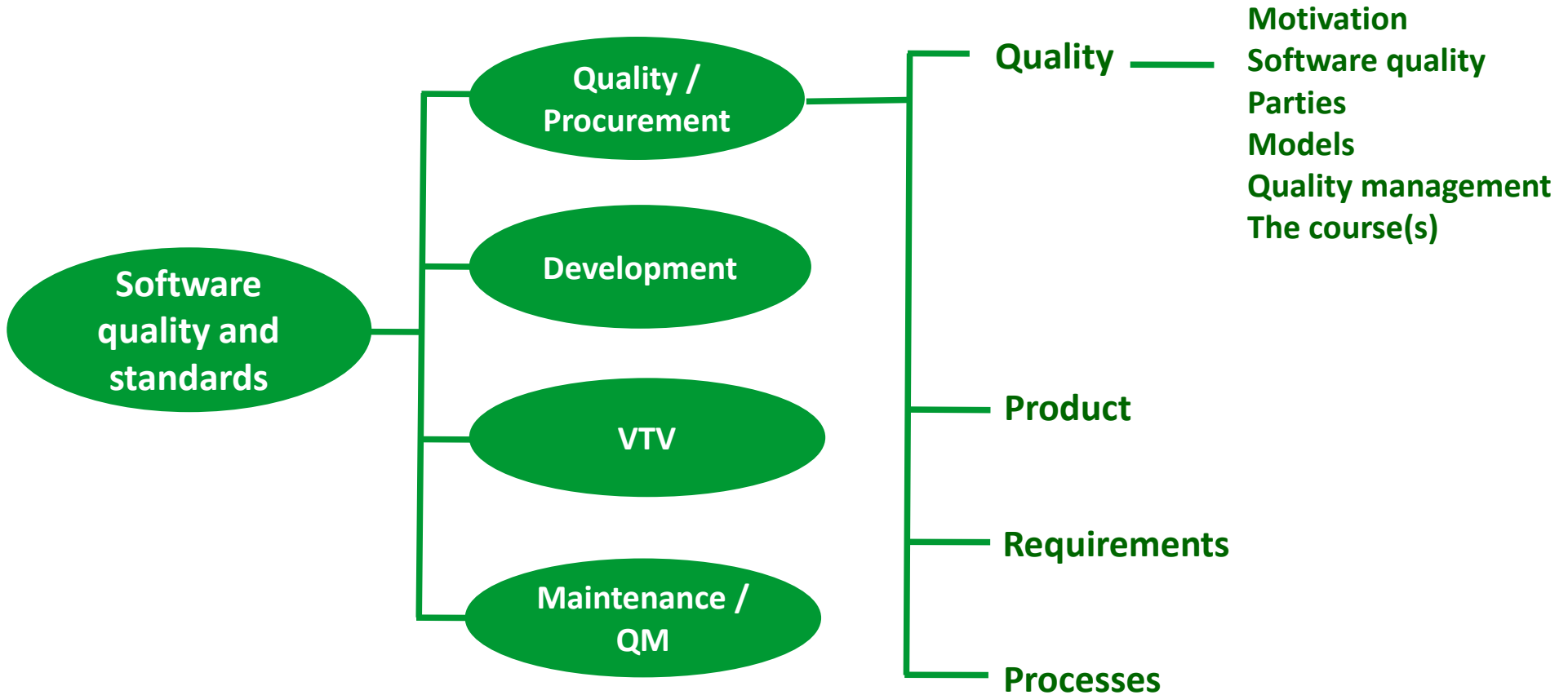
Jaak Tepandi, Jekaterina Tšukrejeva
Tallinn University of Technology
Department of Software Science

Info and slides: <https://moodle.taltech.ee/> - IDY0204, IDX1511, ITB8826

Short intro (no registration needed): <https://tepandi.ee/spqs/>

Version 9.09.2020

Context and content



Lectures, practices, and COVID-19

According to Study information system

Lectures, practices, and labs are synchronised

More info: <https://moodle.taltech.ee/> - IDY0204, IDX1511, ITB8826

Introduction (no registration needed): <https://tepandi.ee/spqs/>

Lectures are streamed and can be watched afterwards

Plan B: only distant lectures / practices

Lectures vs slides&conspectus: Not one-to-one related

Lectures

- present ideas
- are interactive
- refer to and explain slides & conspectus
- do not always fully cover slides & conspectus

Slides (English) & conspectus (Estonian)

- give detailed content
- give references

Motivation



To whom?

purchasers – to prepare the call for proposals, to formulate requirements, to choose the product, to cooperate in development and to evaluate the outcome

system developers and testers – to apply software verification, validation, and quality management methods and tools; to understand program testing principles, select and apply testing methods, and evaluate system quality

maintainers – maintenance processes and activities

users – to know what to request and expect from the software product

leaders – to know what to expect from and how to manage the parties

society – assurance against software related problems

Sorry, your luggage didn't arrive. There was a software
problem....

...meaning?

Software quality and economy

Example: Knight Capital Group

Example: Estonian National Social Insurance Board: Public Service Modernization Project (SKAIS2)

Example: Everyday experience

Example: Agilent Technologies

- the company encountered problems transferring data into a new [...] order booking and accounting system from its other computer systems
- The problem halted normal production for the equivalent of a week
- It cost the manufacturer of semiconductor testing equipment \$105 million in revenue and \$70 million in operating profits

Example: State Economy

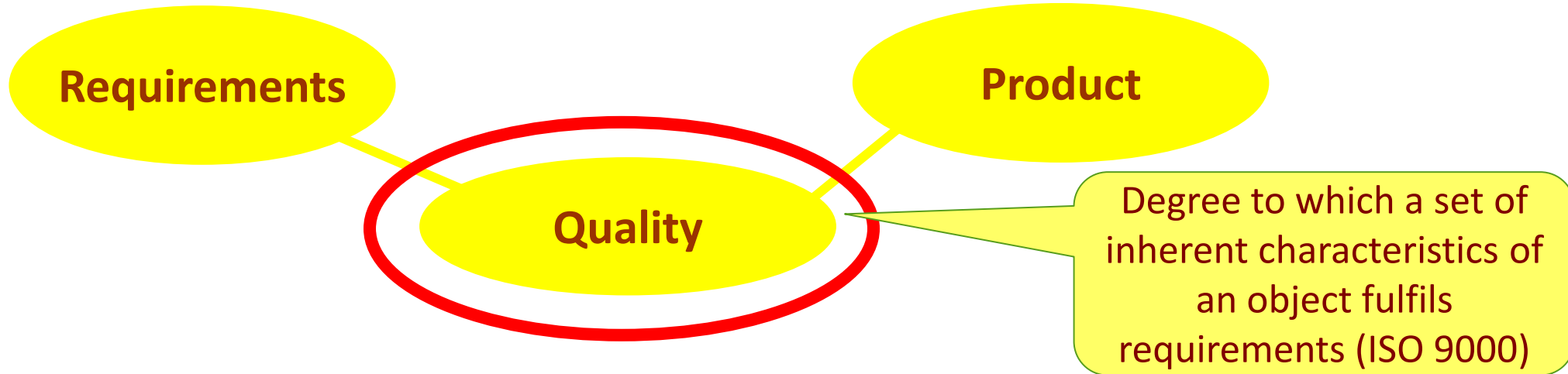
- According to a federal study, buggy software costs the US economy nearly \$60 billion a year ... 0.6% of US gross domestic product (GDP)
- Estonian GDP: about EUR 26 B => EUR 150 M for software bugs?



Software quality



Quality: conformance to requirements



Quality challenges for software systems

„Quality = a product should meet its specification” is problematical for software systems:

- There is a tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.);
- Some quality requirements are difficult to specify in an unambiguous way;
- Software specifications are usually incomplete and often inconsistent.

The focus may be ‘fitness for purpose’ rather than specification conformance.

Process and product quality

- The quality of a developed product is influenced by the quality of the production process.
- This is important in software development as some product quality attributes are hard to assess.
- However, there is a very complex and poorly understood relationship between software processes and product quality.
 - The application of individual skills and experience is particularly important in software development;
 - External factors such as the novelty of an application or the need for an accelerated development schedule may impair product quality.

Quality: product – requirements - process - people - ...

„We want to_do that“



Quality concepts



From a forum discussion - “There is no software quality because of hurrying and overload of programmers”

- Quality as an ideal
- Quality as a relationship
- Quality as a measure

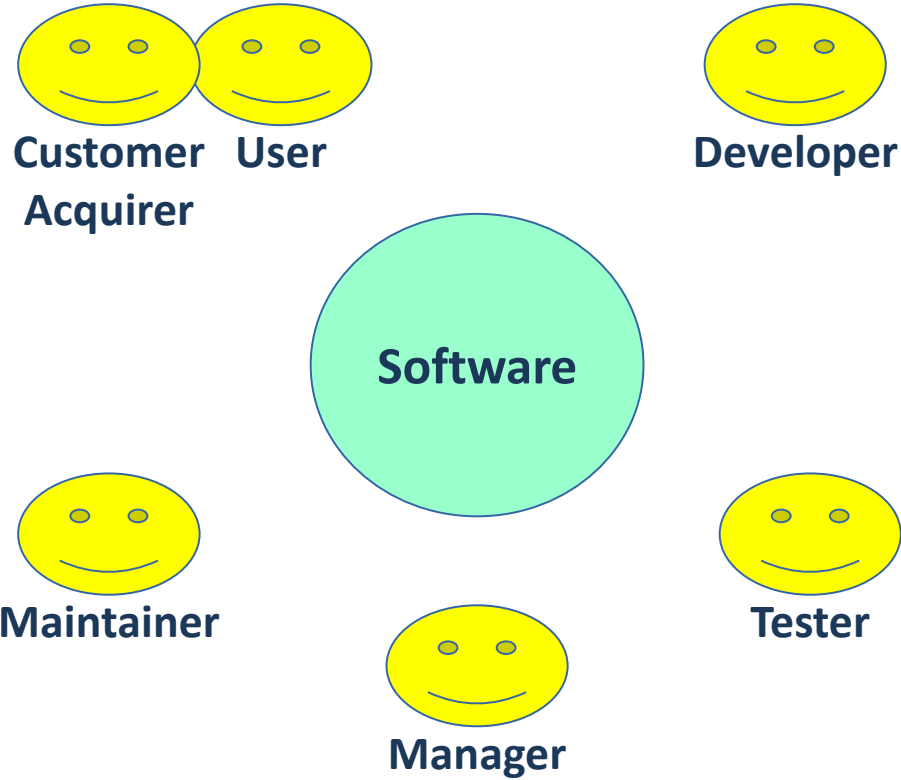
Which quality concept is applied? When is it possible to have no quality?

When quality is a trade-off?

Interested parties / Life cycle stakeholders

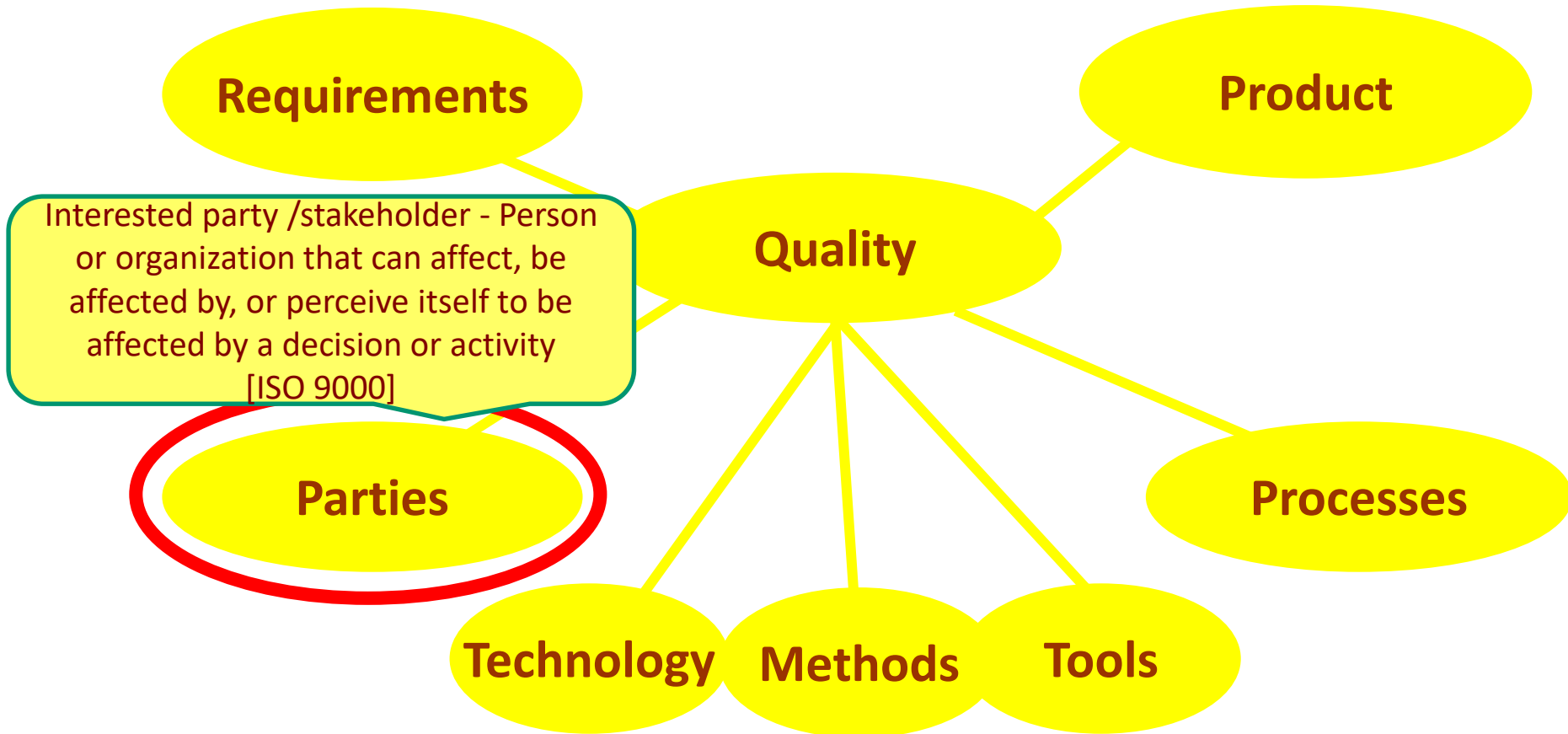


Views on software quality

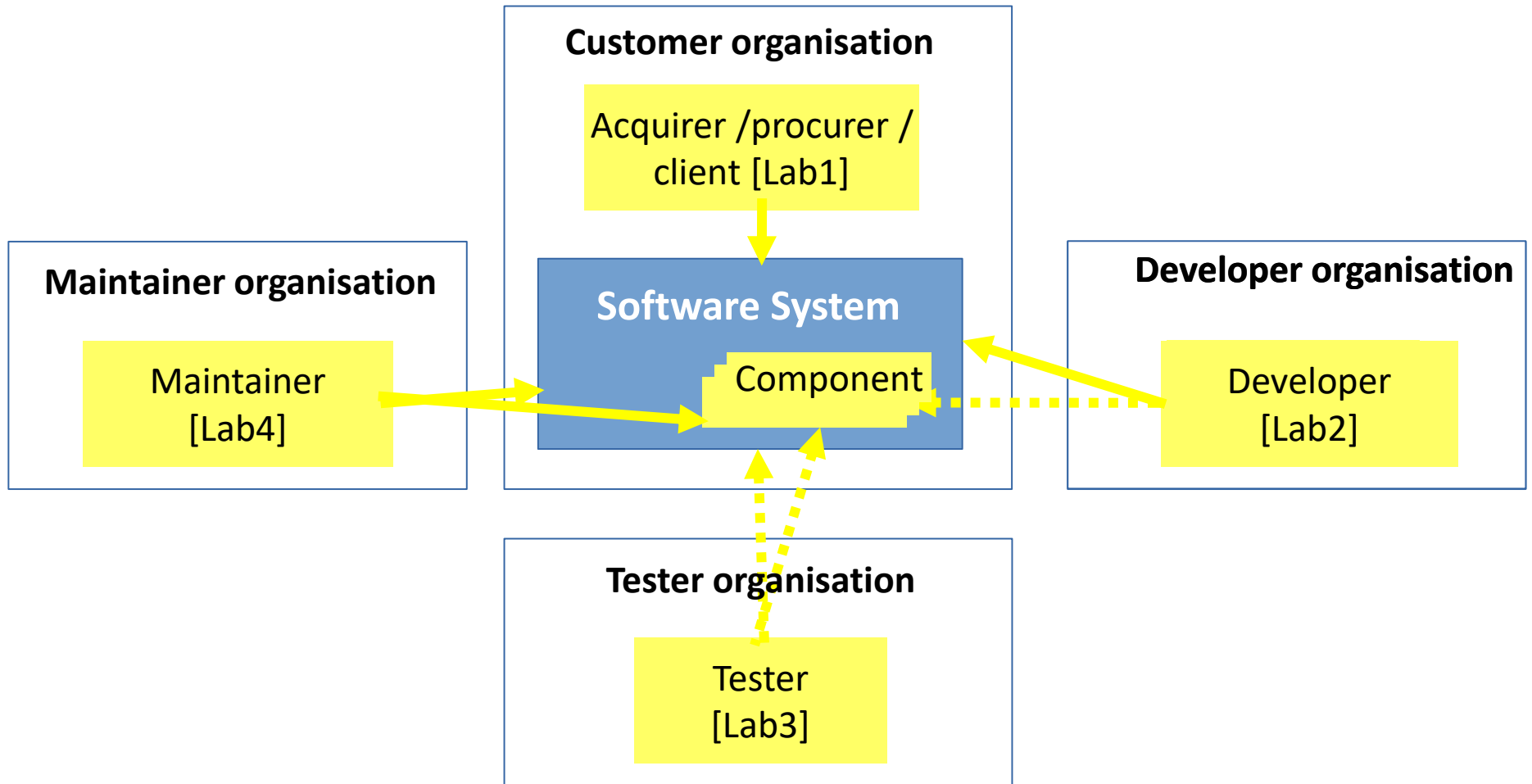


Quality: product – requirements - process - people - ...

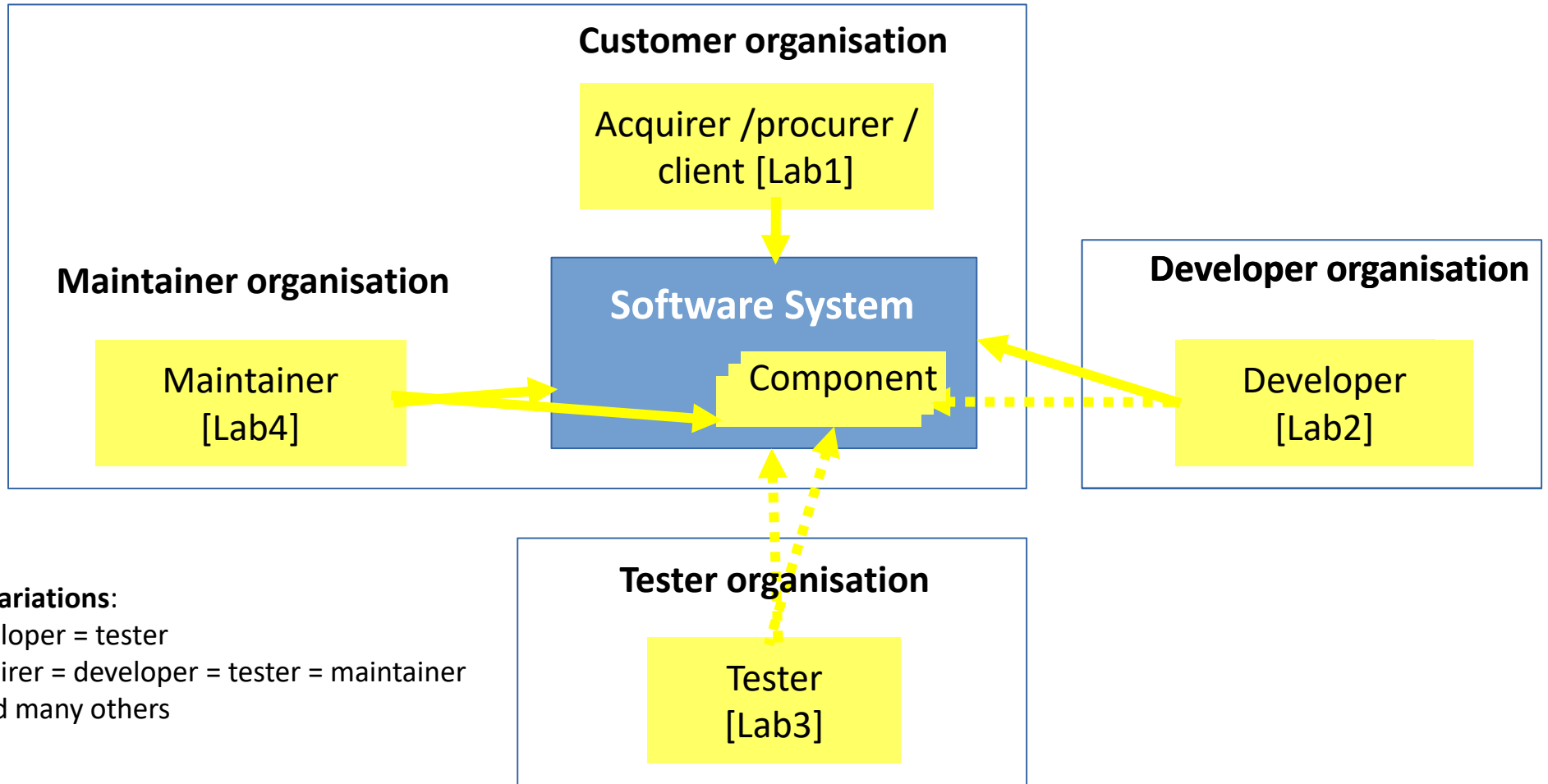
„**We** want to_do that“



Software life cycle stakeholders [cf Lab 1 ... Lab 4]



Variations: Organisations, processes, systems



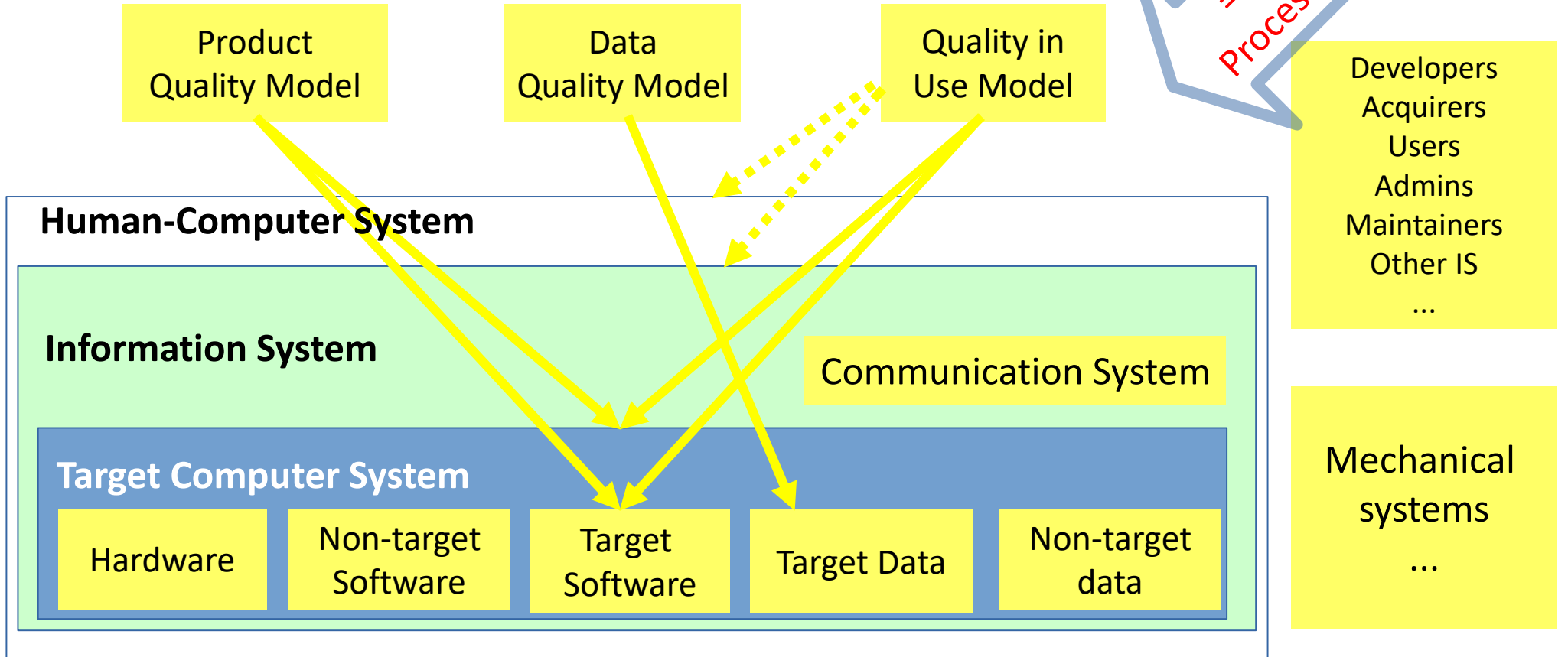
More variations:

- Developer = tester
- Acquirer = developer = tester = maintainer
- ...and many others

Quality models



Quality models (ISO/IEC 25010, SWEBOK)



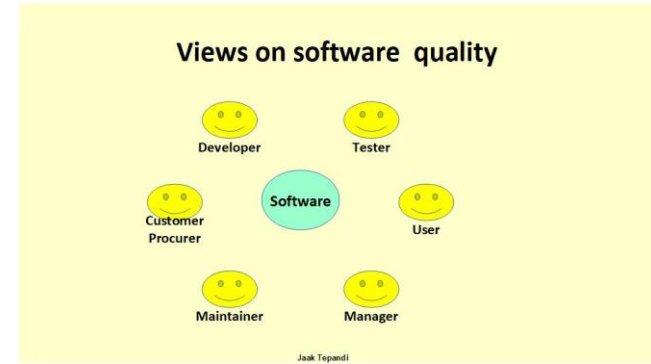
Aspects / models of quality: why needed?

- Product quality
- Data quality
- Quality in use
- Process quality

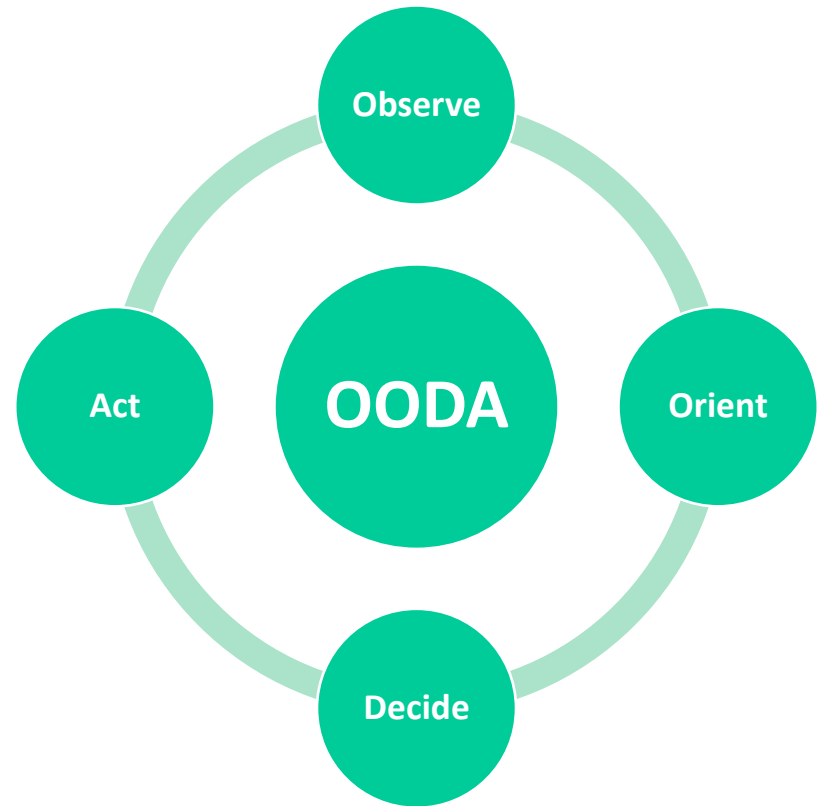
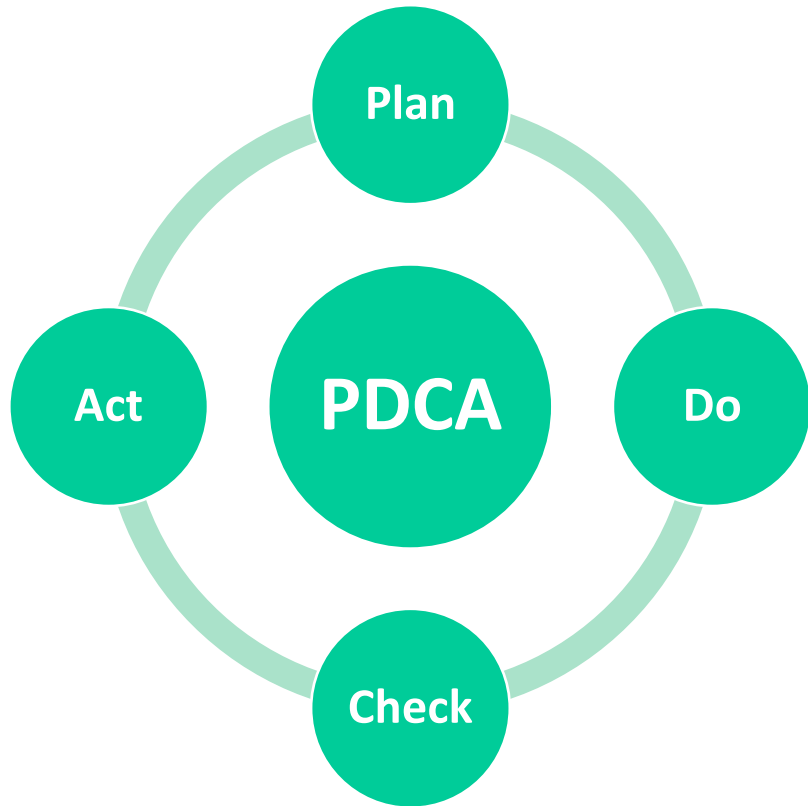
Guide to the Software Engineering Body of Knowledge, SWEBOK,

<http://www.computer.org/portal/web/swebok/v3guide>

ISO/IEC 25010 Software engineering: Software product Quality Requirements and Evaluation (SQuaRE) — Quality model



Useful process models



Quality management (a lot about parties)



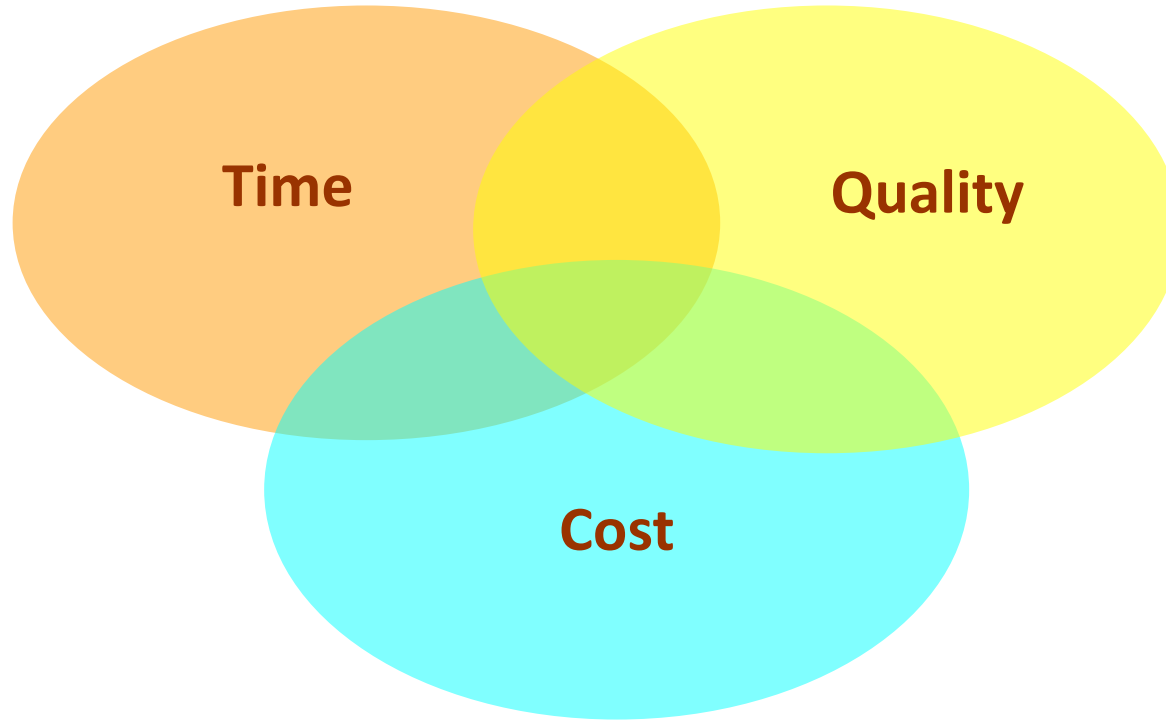
Quality management etc [...and minimum level]

- Quality management (QM) - management with regard to quality, can include establishing quality policies and quality objectives, as well as processes to achieve these quality objectives [... at least, management has a position on quality]
- Quality control - part of quality management focused on fulfilling quality requirements [... at least, some static analysis and testing is performed] ... related to product
- Quality assurance - part of quality management focused on providing confidence that quality requirements will be fulfilled [...at least, there is feedback on how the policies are implemented] ... related to process

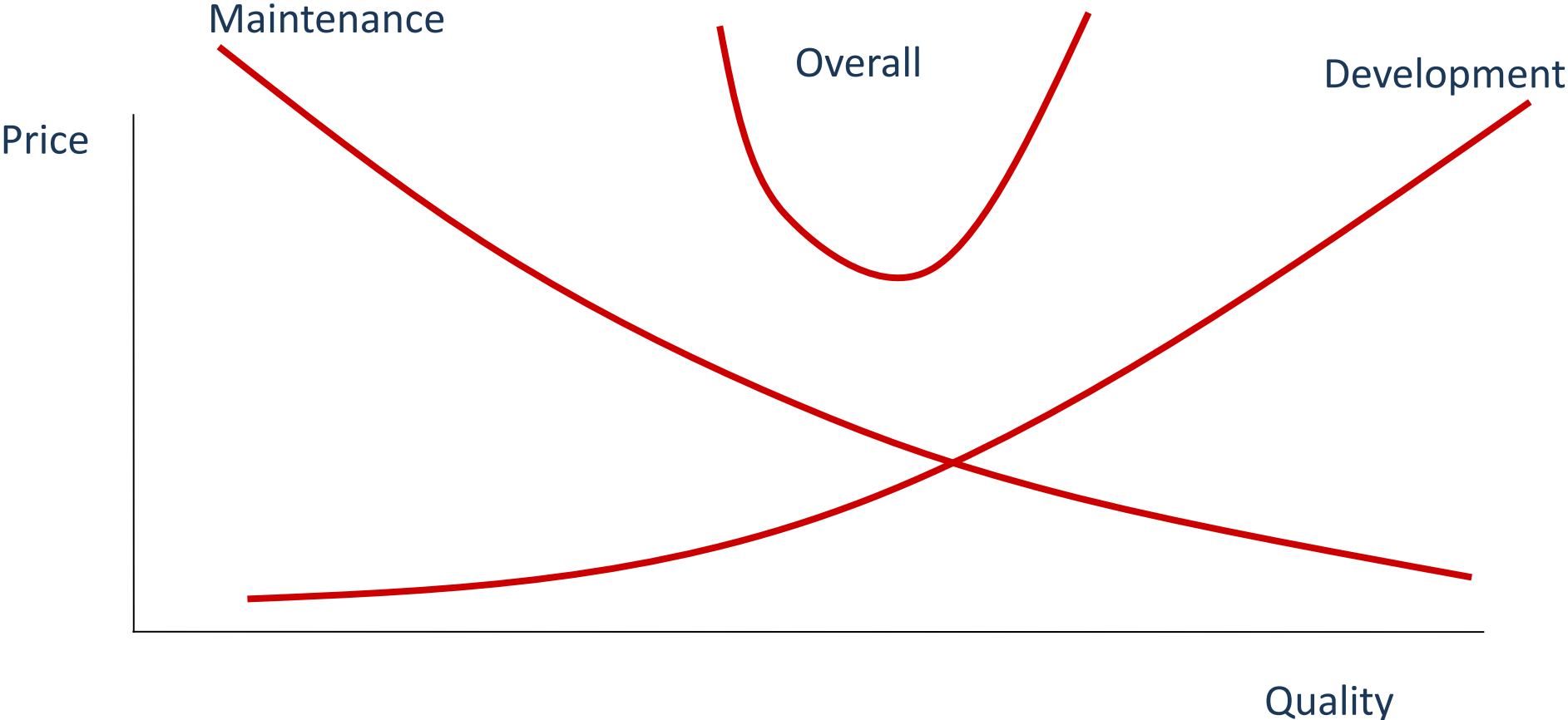
(Based on ISO 9000:2015)

Is quality always a tradeoff?

Project management triangle



Cost of quality

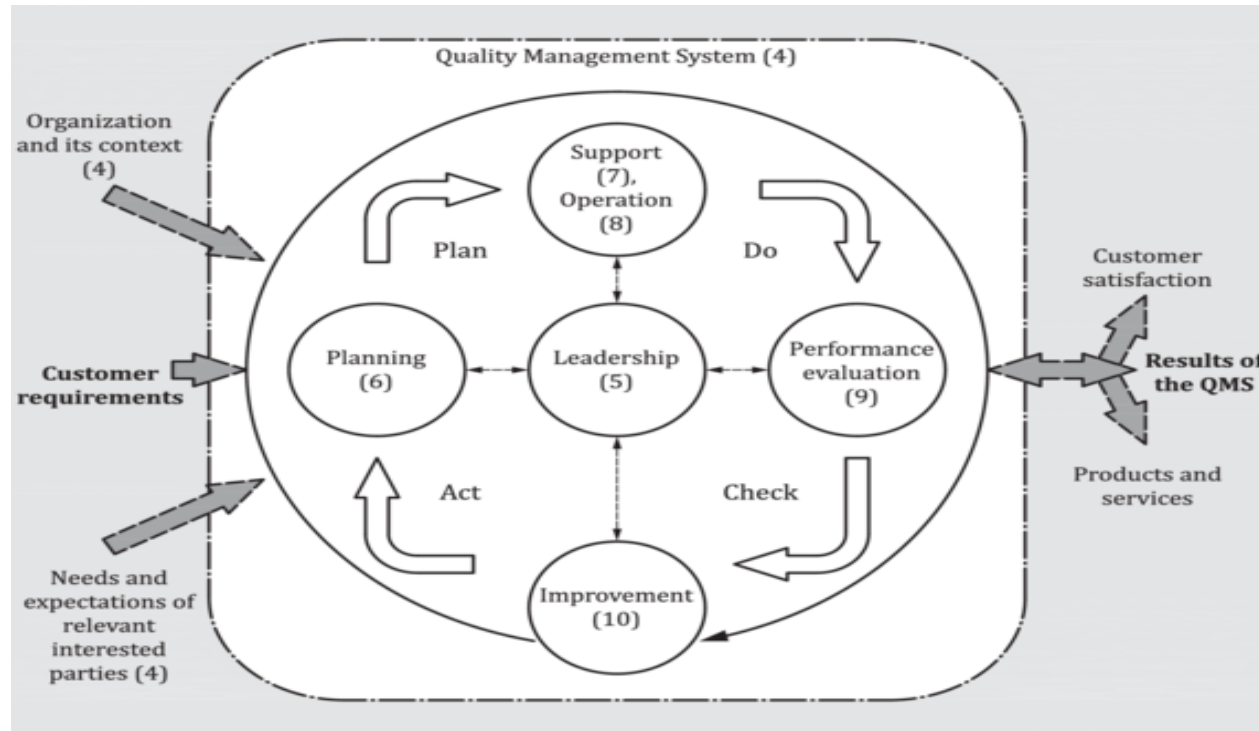


ISO 9000 family of standards: history and present situation

First version 1987 => renewed 1994 (~ 20 standards) => renewed several times
=> current:

- ISO 9000:2015, Quality management systems. Fundamentals and Vocabulary
- ISO 9001: 2015, Quality management systems. Requirements
- ISO 9004:2018, Quality management. Quality of an organization. Guidance to achieve sustained success
- + more, e.g. ISO/IEC 90003:2018, Software engineering — Guidelines for the application of ISO 9001:2008 to computer software

ISO 9001:2015 and the PDCA cycle



NOTE Numbers in brackets refer to the clauses in ISO 9001:2015

Source: <https://www.iso.org/obp/ui/#iso:std:iso:9001:ed-5:v1:en>

agilemanifesto.org and quality components

... we have come to value:

Individuals and interactions over processes and tools [people]

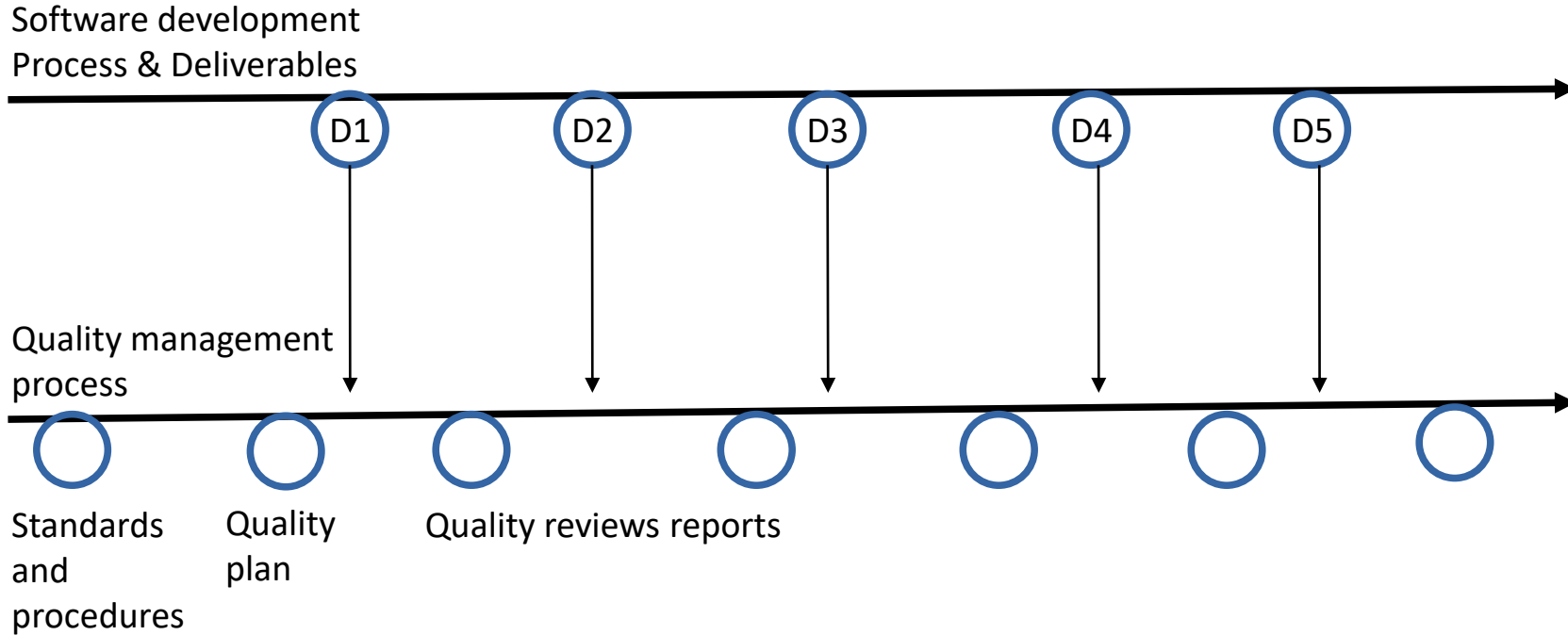
Working software over comprehensive documentation [product]

Customer collaboration over contract negotiation [requirements]

Responding to change over following a plan [processes]

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

Quality management and software development



(More) software quality challenges

Software complexity

Unclear / not understood requirements

Unrealistic deadlines

Insufficient resources

Lack of personnel

Software is part of organisation, and organisations are difficult to change

Software quality is not considered from beginning to end

Complex standards

Inadequate processes, methods, tools

Overcoming QM for the sake of QM (or certification, fashion, etc)



Quality: value and smells

QM needed:

- Understanding organisation goals
- Identifying and strengthen critical success factors
- Support reaching the goals in deadline and budget
- Example: Ensuring that critical errors are removed before the software is deployed to millions of customers

QM not needed (“smells”):

- Bureaucracy
- Unnecessary rules
- Inflexible processes
- Example: products and documentation not needed in the short or long run

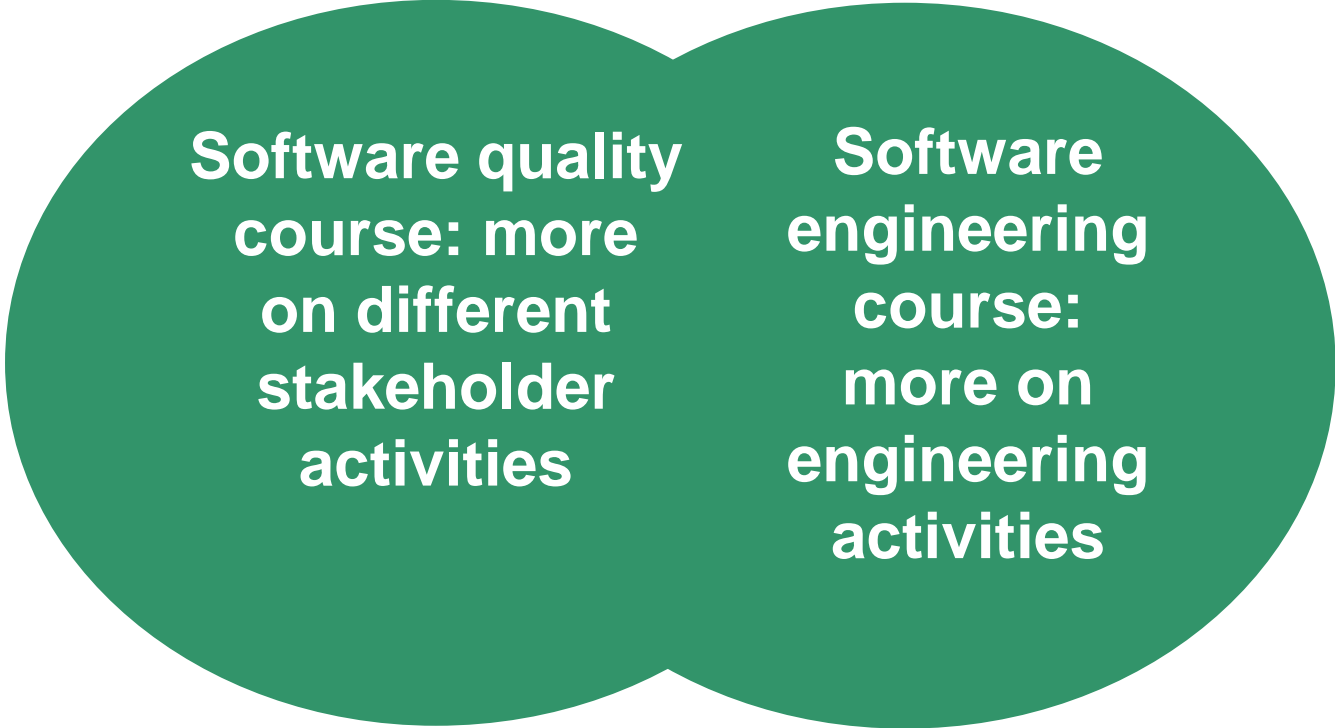
The course(s)



The course

- Software quality related methods and techniques
- Independent of concrete programming languages, tools, environments...
- ... but using and teaching these in order to present course content
- Changing, developing, responding to needs
- Using large collections of requirements (standards etc)
- Presenting several roles besides developer + related processes
- Not a VTV course (but VTV is a component)
- Knowledge: SW QM (requirements / product / testing / maintenance)
- Skills and practical experiences: requirements specification, code quality, tests and tools, static methods, processes etc

Courses: quality management and software engineering



**Software quality
course: more
on different
stakeholder
activities**

**Software
engineering
course:
more on
engineering
activities**

The course and ACM/IEEE Computing Curricula

- ACM/IEEE Computing Curricula, Software Engineering Volume, <http://www.acm.org/education/curricula-recommendations>
 - The current course integrates basic concepts of "Software Quality", "Software Verification and Validation", "Software testing", and (to less extent) "Software Engineering Process"
- + Guide to the Software Engineering Body of Knowledge, SWEBOK Ch 4 (Software testing), Ch 10 (Software quality),...,

Lecture participation vs recordings

Impact of lecture capture availability on student attainment – summary:

- Lecture capture viewing shows no significant relationship with attainment
- Capture viewing also fails to compensate for the impact that low attendance has on attainment
- Thus, the net effect of lecture capture introduction ... is generally negative
- The study serves as a useful example (that can be communicated students) of the pitfalls of an over-reliance on lecture capture as a replacement for lecture attendance

<https://link.springer.com/article/10.1007/s10734-018-0275-9>

Make it as simple as possible, but no simpler

- Quality – degree of fulfilment of requirements
- Management – creating processes, directing, controlling
- Quality management - management with regard to quality
- Quality control – are quality requirements fulfilled?
- Quality assurance – will quality requirements be fulfilled?
- Components of quality - parties, product, requirements, processes,...
- Procurer, acquirer, customer – receives the value from the product
- User – uses the product
- PDCA – Plan Do Check Act
- „Quality is free“

Quality: people - requirements – product - process



Learning outcomes

- Give examples of the need for software quality and explain that quality is context specific
- Analyse different concepts and components of quality
- Characterise stakeholders related to quality and main quality models
- Explain quality management and challenges, the cost of quality, quality related components in software technologies
- Outline the ISO 9000 and ISO 25000 standards series, PDCA, OODA, ACM/IEEE computing curricula

NB! Additional reading (examples)

Ian Sommerville. Software Engineering. Ninth Edition (or later). Addison-Wesley, Ch 1,24.

Daniel Galin, Software Quality assurance from theory to implementation, Pearson - Addison-Wesley. Chapter 2.

Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE. Chapter 13, Section 8.

Certified Tester Foundation Level Syllabus, ISTQB. Chapter 1.1.

moodle.taltech.ee - slides, arrangement, references, labs etc (short summary: tepandi.ee)