

# Software quality, processes, and standards

## Basic concepts

Jaak Tepandi, Jekaterina Tšukrejeva, Stanislav Vassiljev, Pille Haug

Tallinn University of Technology

Department of Software Science

Moodle: „Software Quality (Tarkvara kvaliteet)”

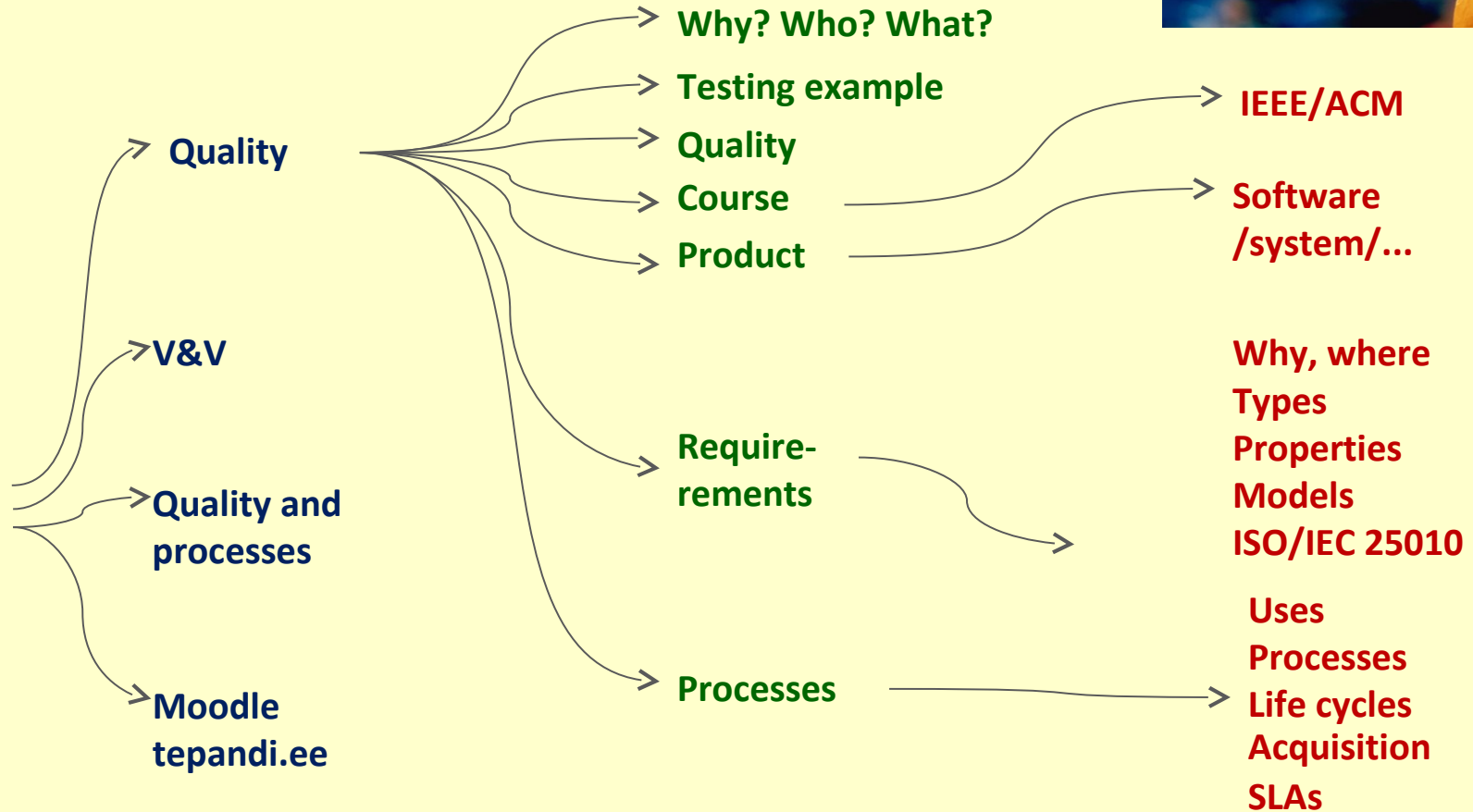
Alternate download: [tepandi.ee](http://tepandi.ee)

Version 27.09.2017

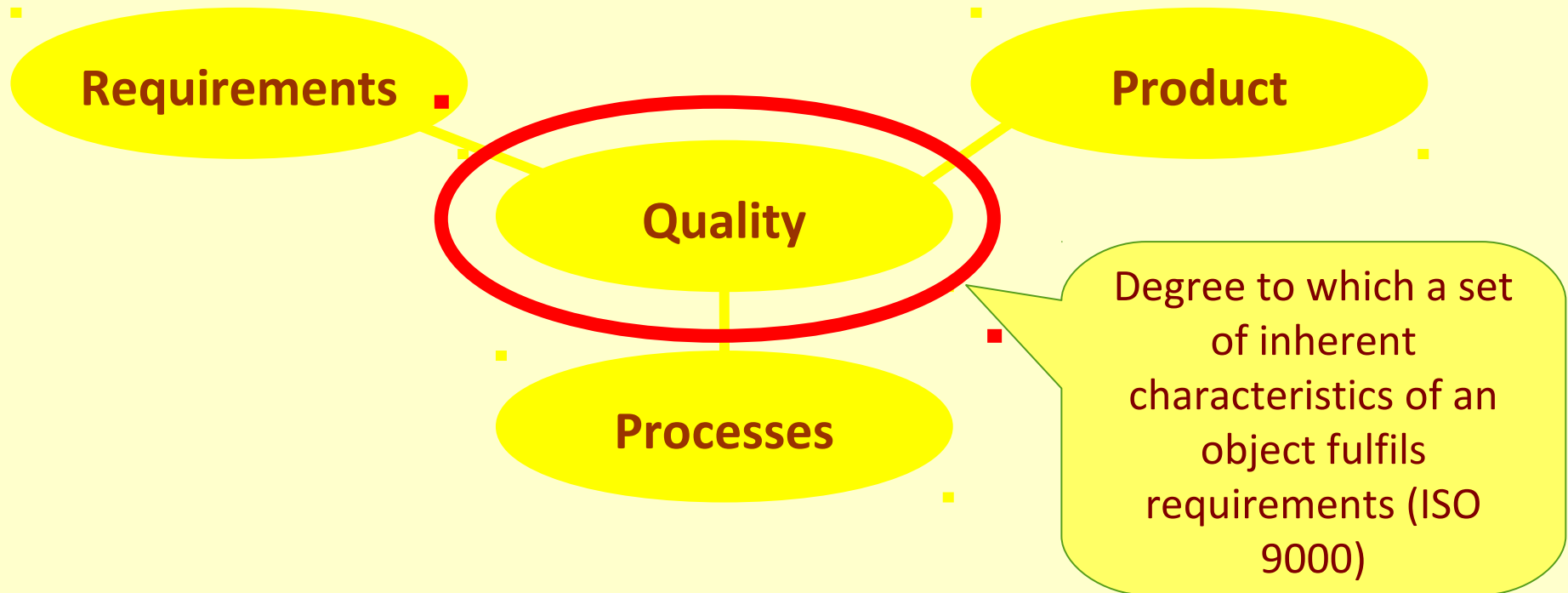
# Today and the course



Software  
quality,  
processes,  
and  
standards



# Quality: product – requirements - process



# Quality concepts

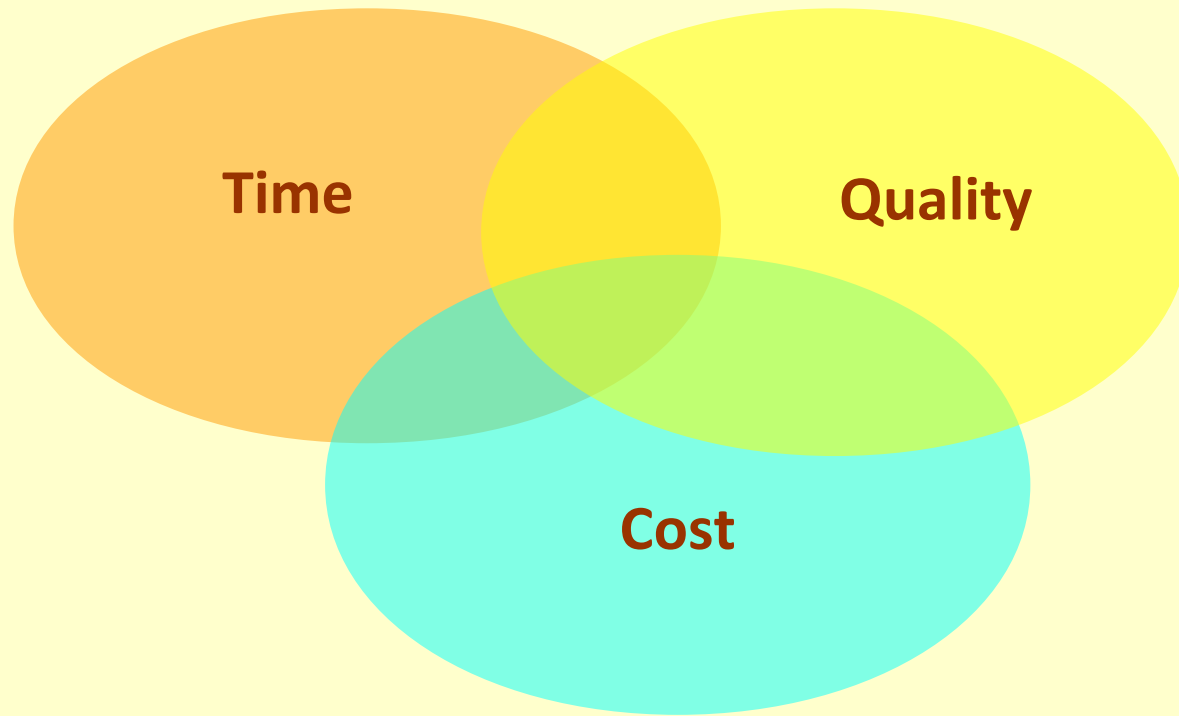


- Ideal
- Relationship  
Software product quality: capability of software product to satisfy stated and implied needs when used under specified conditions
- Measure

“There is no software quality because of hurrying and overload of programmers”: which quality concept is applied? When is it possible to have no quality?

When quality is a trade-off?

# Quality as a tradeoff: project management triangle



# The course: What does it give?

- Knowledge
  - software quality management (product / requirements / development)
  - methods and arrangements of testing and control
- Skills and practical experiences
  - requirements specification, various tests and tools, walkthroughs, processes, evaluation
- ACM/IEEE Computing Curricula as a guideline
- To developers, purchasers, maintainers, users, leaders, society

# To whom?

system developers – 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

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

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

# Skills gained from the course (1)

- High-level requirements specification
- Acceptance testing
- Functional testing through equivalence classes and boundary situations
- Non- functional testing
- Program based testing using coverage criteria
- Choosing the method of the testing
- Planning the testing arrangement



# Skills gained from the course (2)

- How to start the quality management process in a company?
- How to choose appropriate standards?
- How to use the software quality attributes?
- How to formulate the task for testing or audit?

# Development of quality systems and software

## Two complementing courses

### Software quality / processes (current)

All systems

Requirements determine criticality

Refers to assurance (eg, critical systems)

Main modules:

1. Quality=product, requir-ts, processes
2. Testing, verification, validation
3. Processes

### Software assurance (ITI8610)

Cyber-physical systems

Critical systems

Refers to quality course (eg, testing)

Main modules:

1. Assurance and risk management
2. Security assurance
3. Functionality assurance

# Course materials

Moodle: „Software Quality (Tarkvara kvaliteet)”  
(+ tepandi.ee) - slides, arrangement file, references,  
labs etc

References: In Moodle, in the arrangement file, on the slides

# The course and *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),...

# Additional reading - quality (examples)

Ian Sommerville. Software Engineering. Ninth Edition. Addison-Wesley, Chapter 1

Daniel Galin. Software Quality assurance from theory to implementation. Pearson - Addison-Wesley. Chapters 1,2.

Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE. Chapter 10, Section 1.

Certified Tester Foundation Level Syllabus, ISTQB. Chapter 1.1.4.

# Aspects / models of quality: why needed?

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

Guide to the Software Engineering Body of Knowledge, SWEBOK,

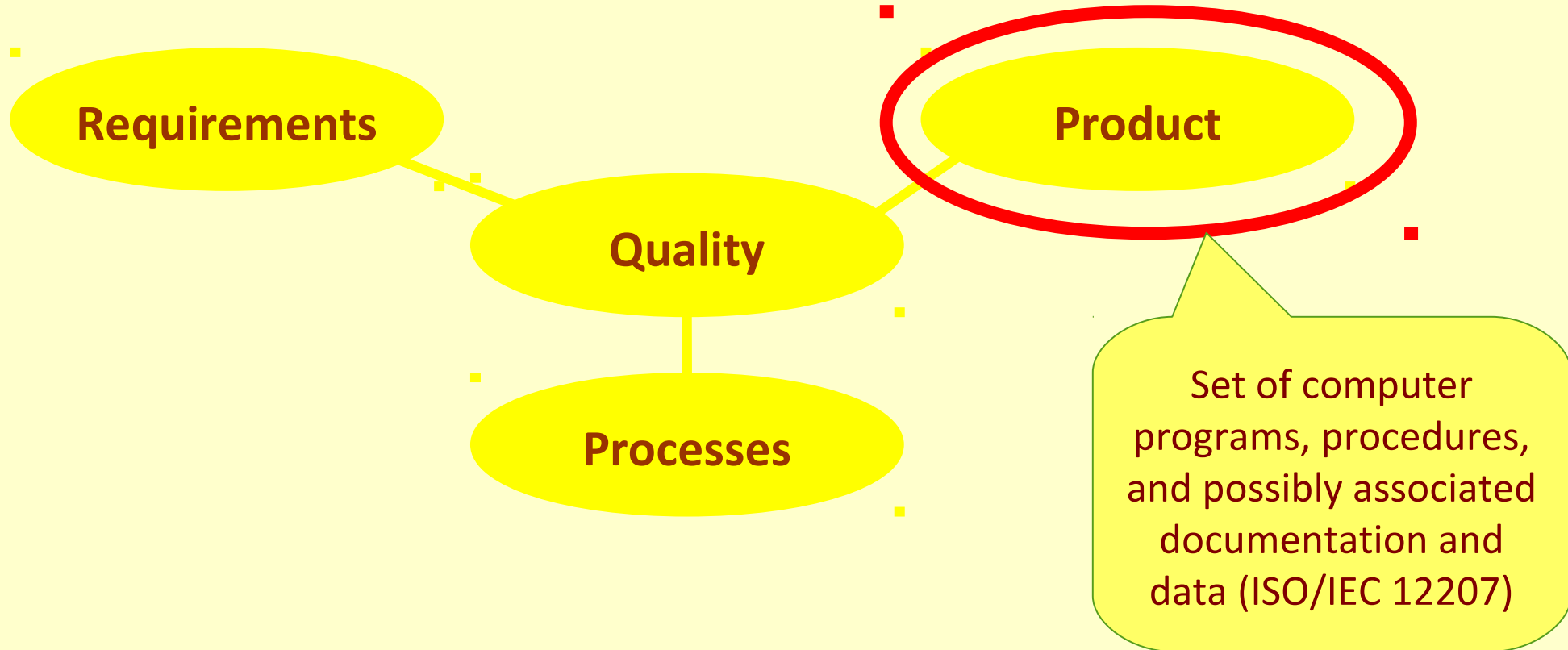
<http://www.computer.org/portal/web/swebok/v3guide>

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

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

...meaning?

# Product: When talking about software, what are we talking about?





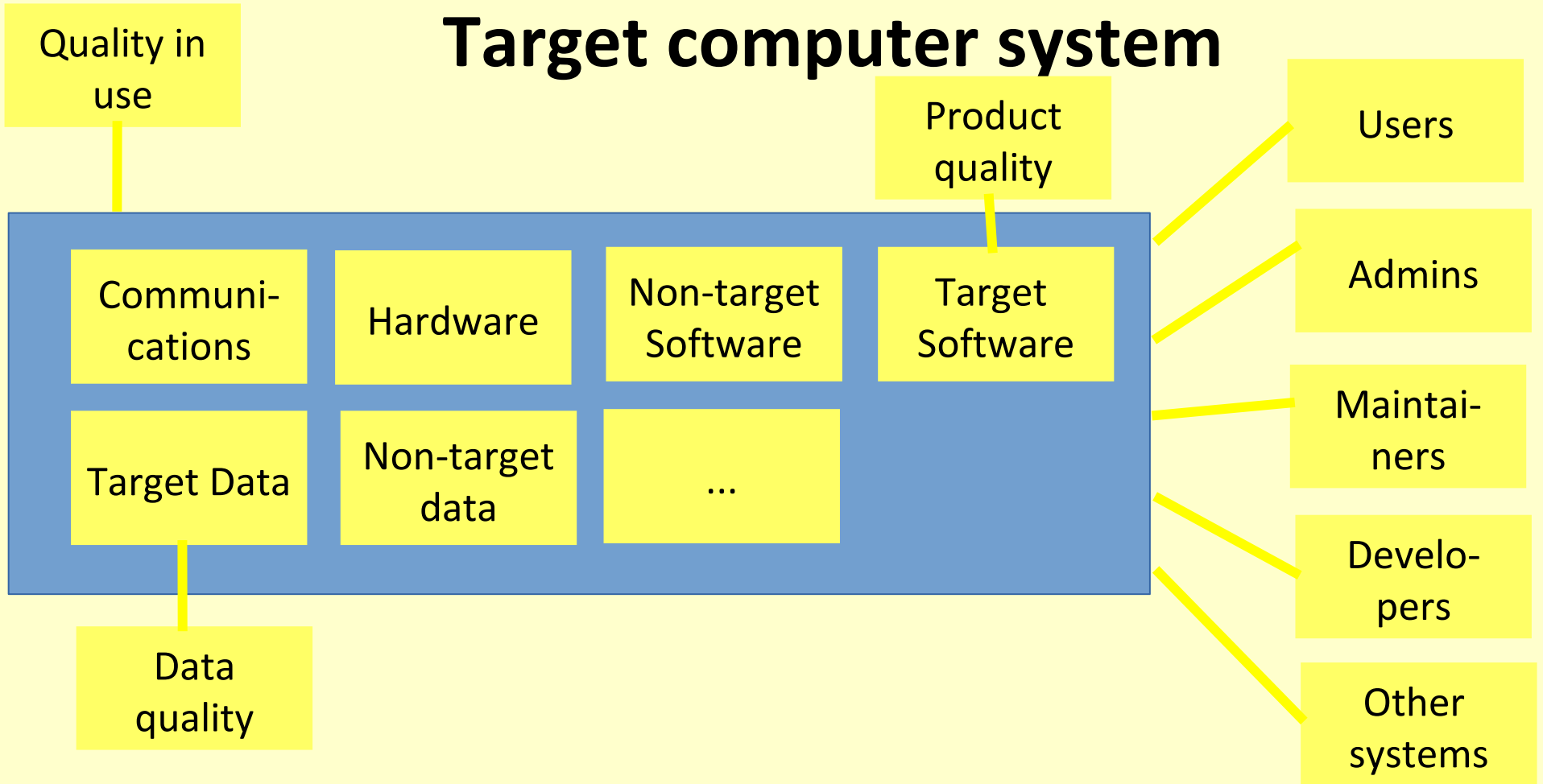
# Software (product)

- acquired IT-resources (incl source code, object code, executable code, macros, etc)
- work done in development process
- changes in customer organization, in work organization, ... ,
- project documentation
- methodology
- resources
- knowledge
- property rights: for working, for development, for distribution, ...

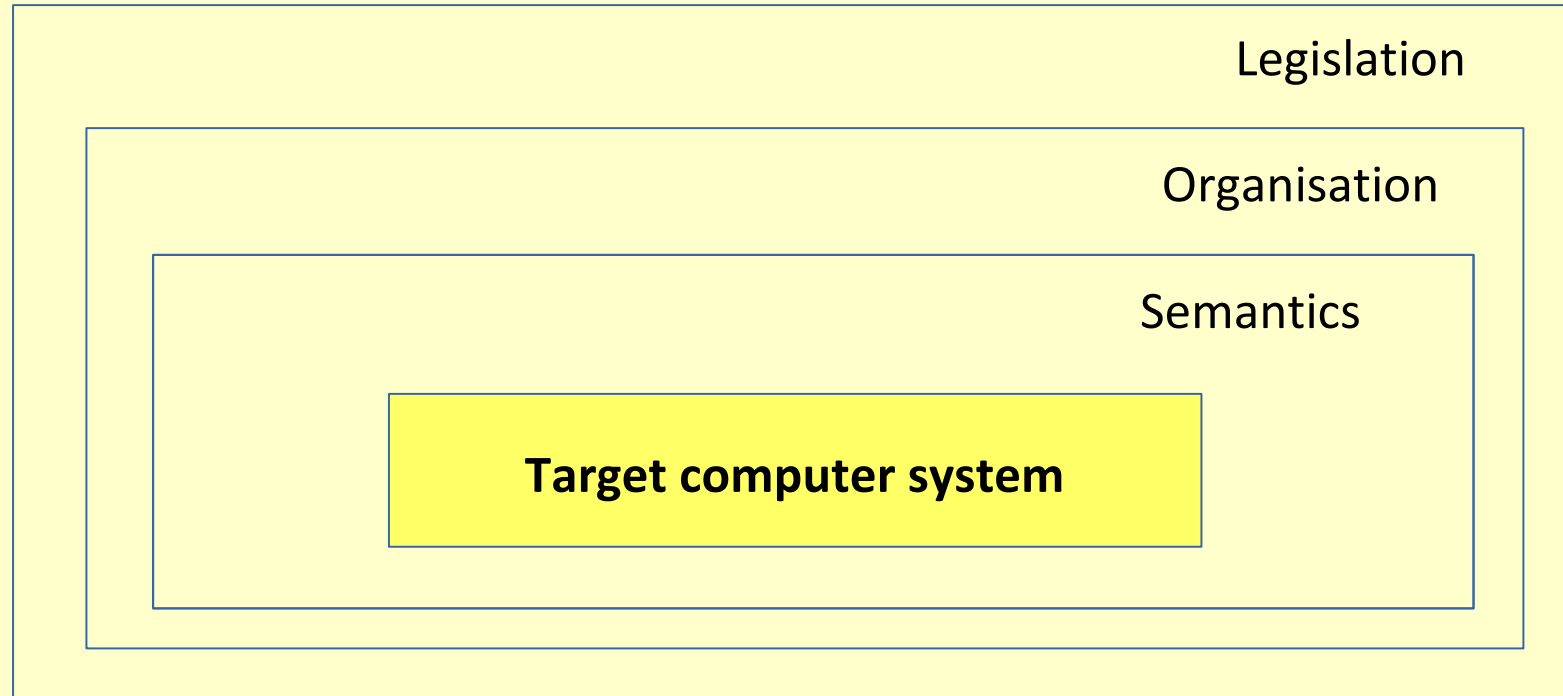
# Software and system

- (Target) software
- Target computer system: hardware, non-target software, target software, target data, non-target data, communications,...
- Target business/information system (or device with embedded software): computer system, communications, people, mechanical systems,...

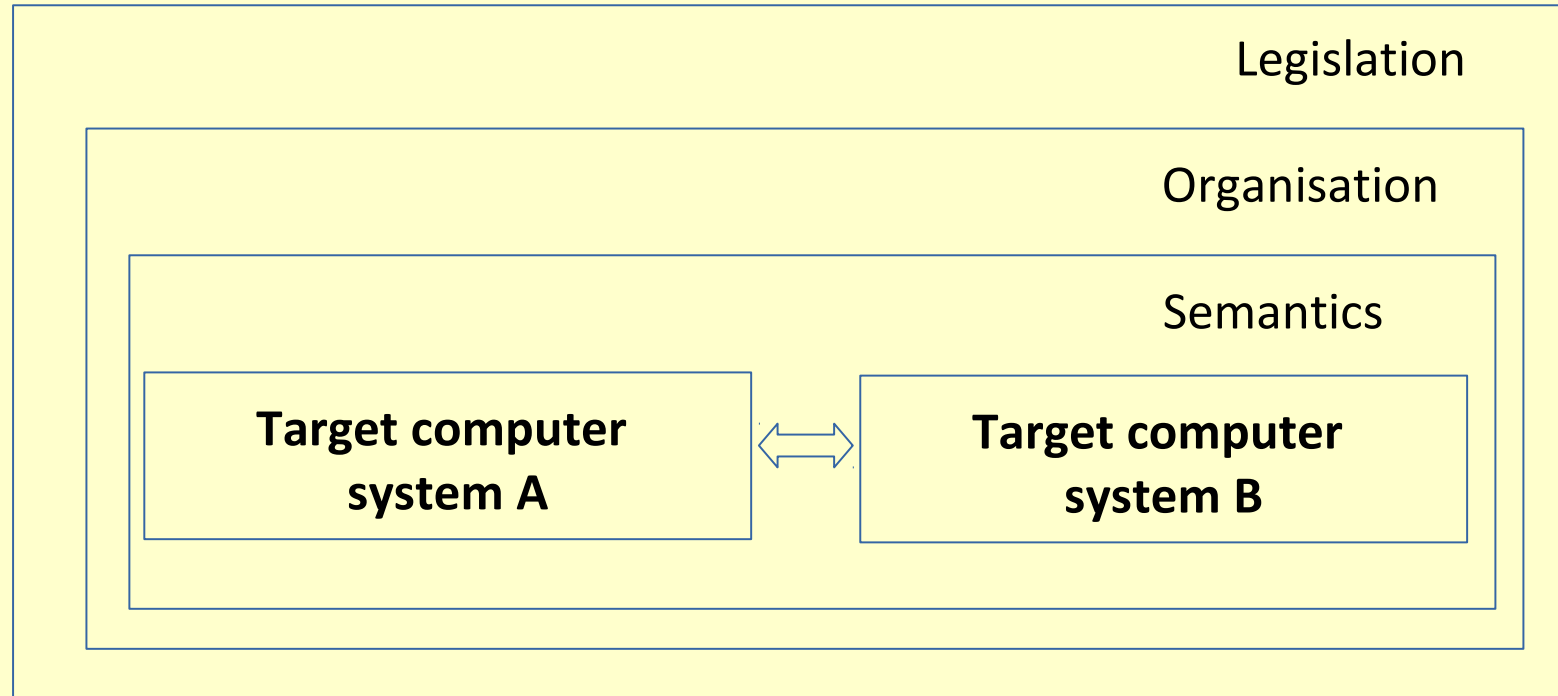
# Target computer system



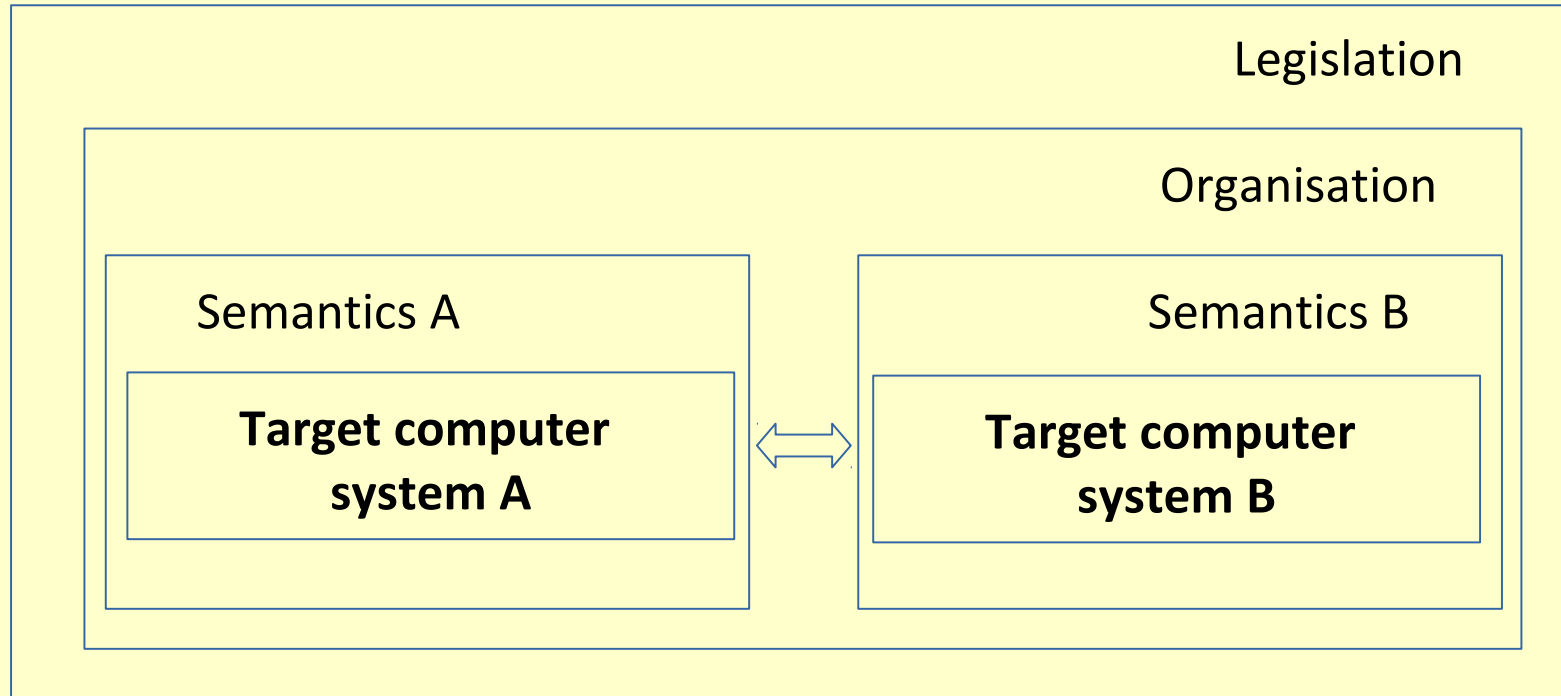
# Layers of complexity / interoperability



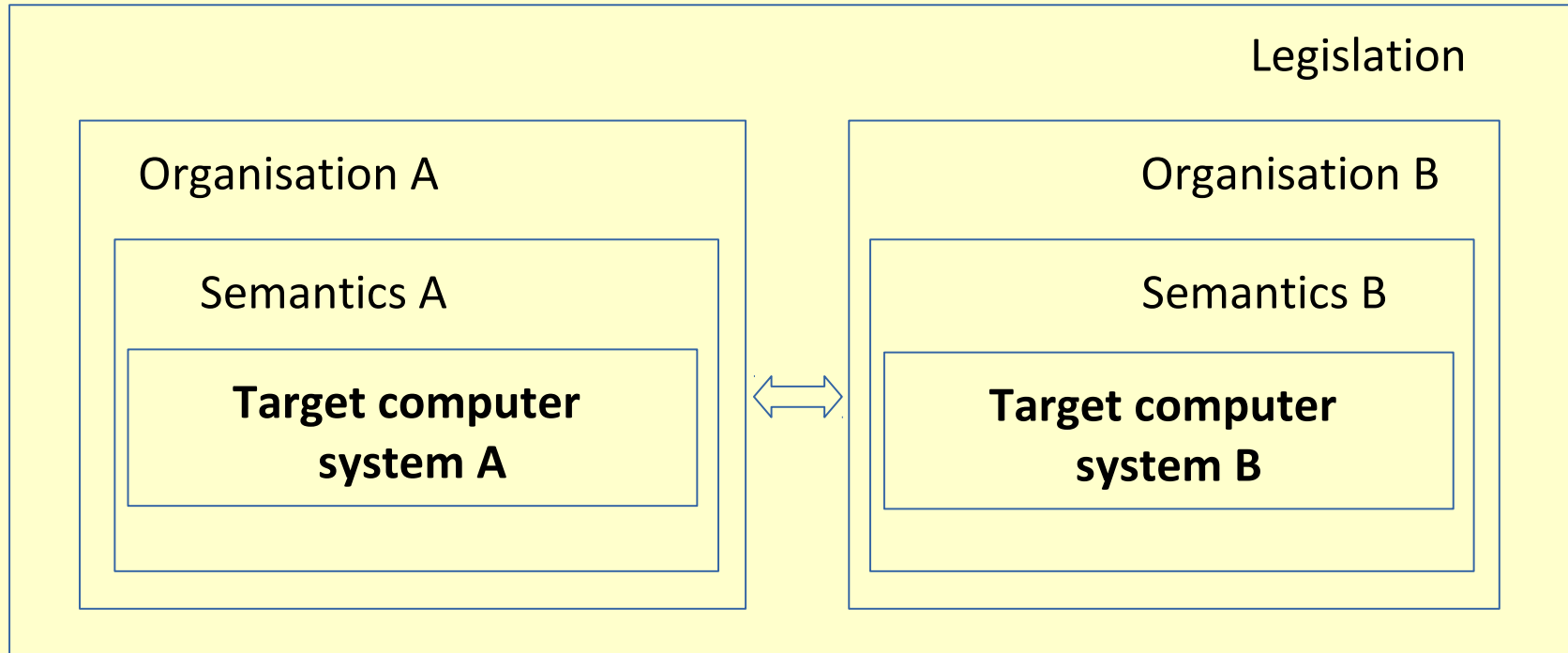
# Example: subsystems within one system



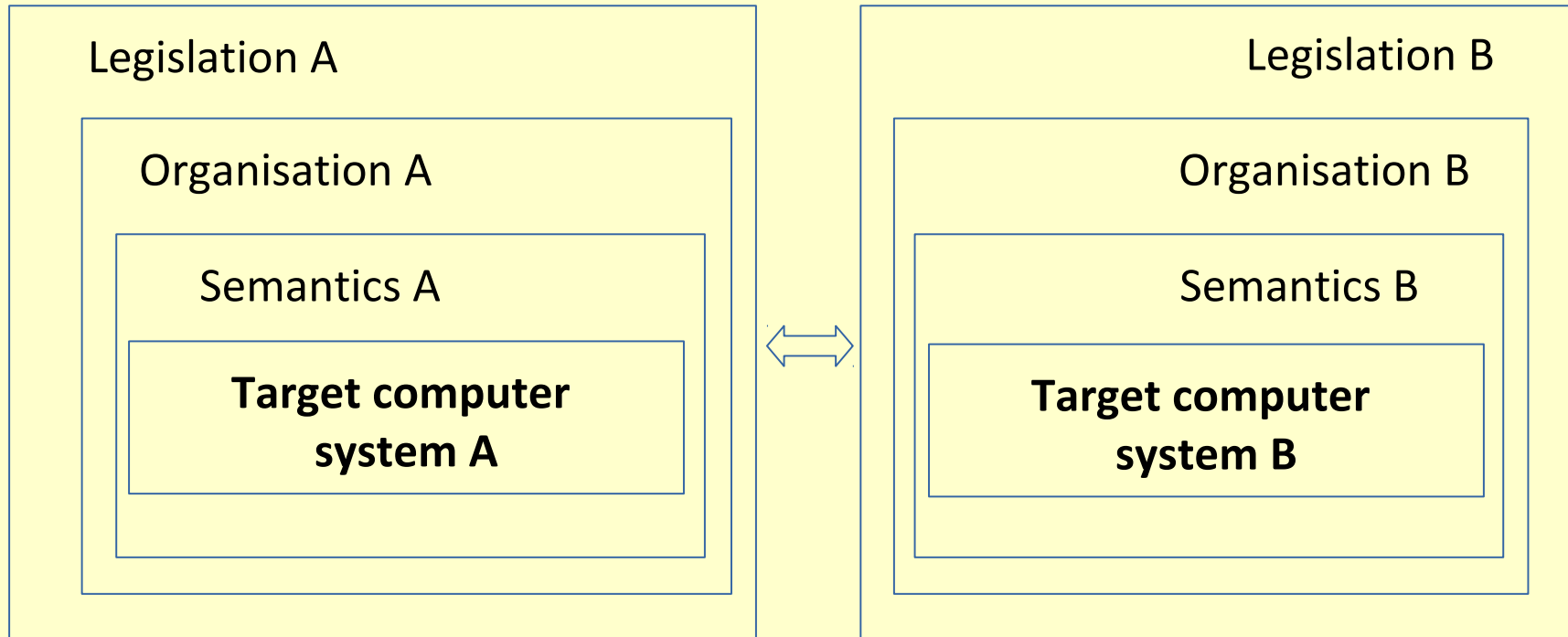
# Example: systems within one organisation



# Example: systems within one country



# Example: cross-border systems





# Product, User, Data quality

Software quality: capability of software product to satisfy stated and implied needs when used under specified conditions

Quality in use: the extent to which a product used by specific users meets their needs to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use

Data quality: degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions

(SWEBOK, ISO/IEC 25000, COBIT)

# Quiz

Search for “X system architecture” (pictures), where “X” is usually referred to as a software system.

Where is software on this picture?

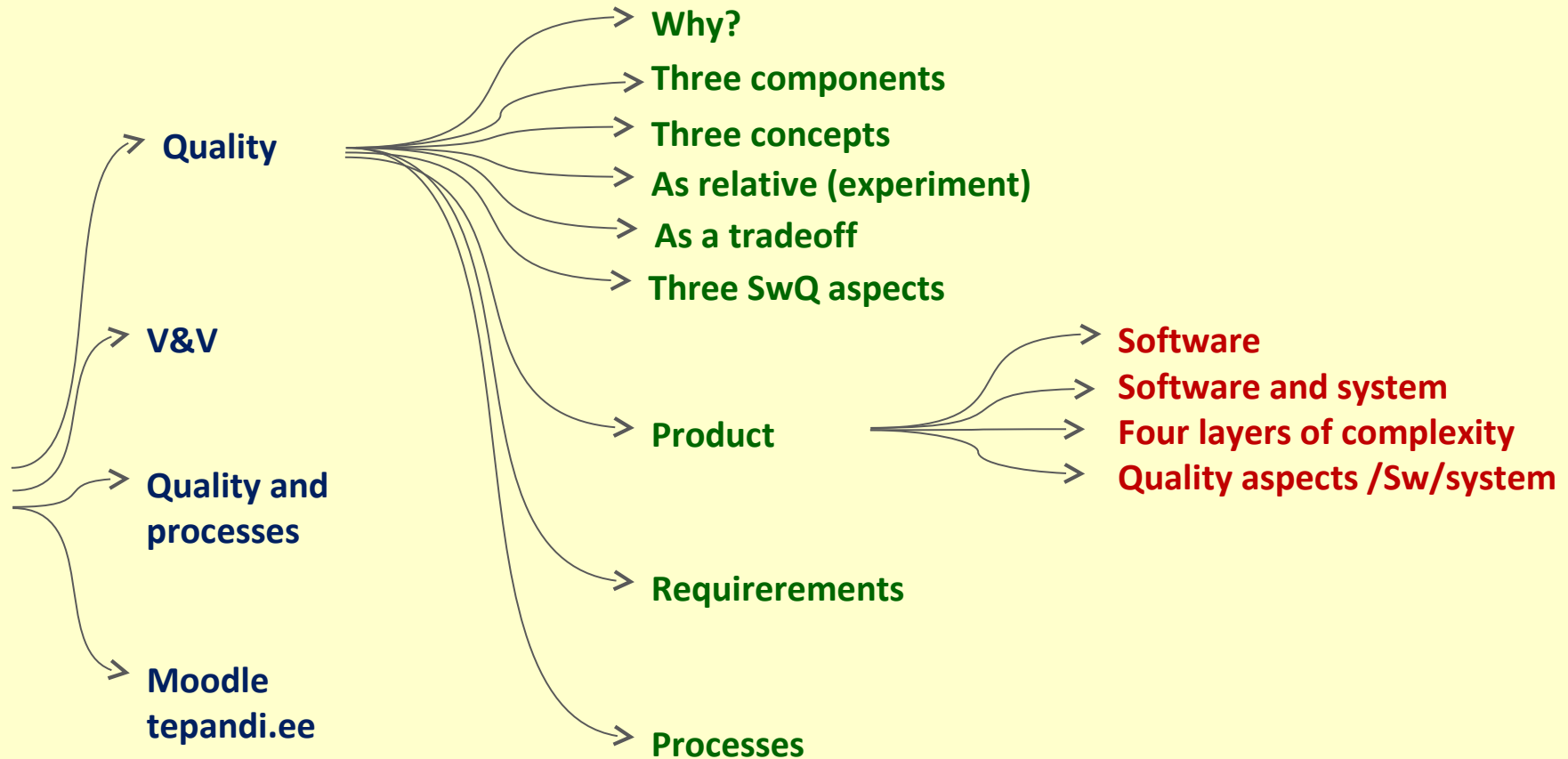
Where is the developed software?

Which components could be influenced by developers? testers? quality management?

# Takeaway: explore the SwQ landscape



Software  
quality,  
processes,  
and  
standards



# Key points to know

- Quality, its components
- Three quality concepts
- Quality as relative
- Quality as a trade-off
- Four quality models
- Software as a product
- Software and system
- Target and non-target software, data, hardware, communications
- Layers of complexity / interoperability

# Additional reading (examples)

Ian Sommerville. Software Engineering. Ninth Edition. 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.