Software Quality Management

Kristian Sandahl

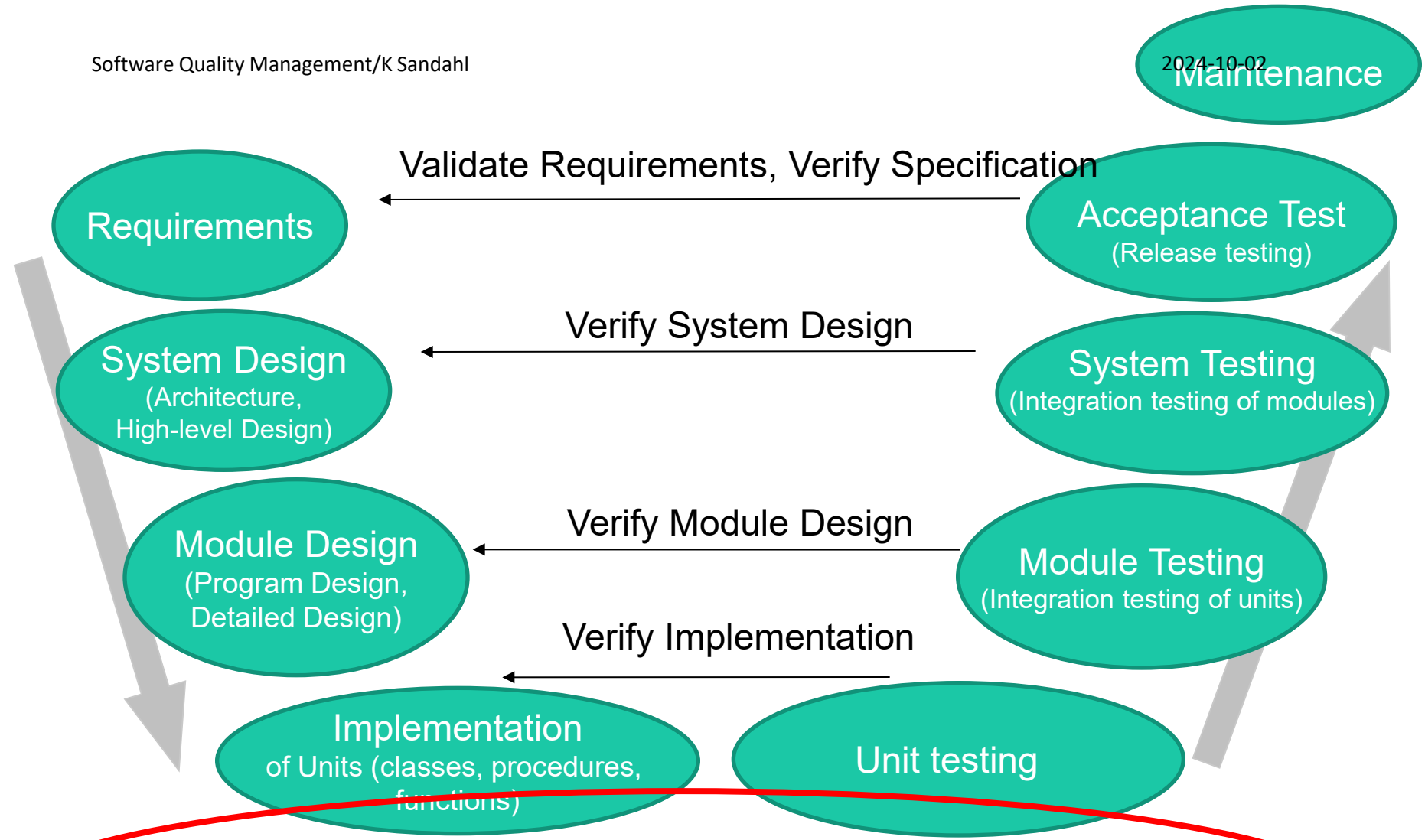
LINKÖPING UNIVERSITY

Agenda:
What is quality?
How to achieve quality
Mature organizations and CMMI
Document quality work
General quality thinking

This is my last lecture

There are some seminars and Q&A sessions left.
General quality thinking will be ceremonial 😀

LINKÖPING UNIVERSITY


Maintenance
Validate Requirements, Verify Specification
Requirements
Acceptance Test
(Release testing)
Verify System Design
System Design
(Architecture,
High-level Design)
System Testing
(Integration testing of modules)
Verify Module Design
Module Design
(Program Design,
Detailed Design)
Module Testing
(Integration testing of units)
Verify Implementation
Implementation
of Units (classes, procedures,
functions)
Unit testing
Project Management, Software Quality Assurance (SQA), Supporting Tools, Education

Swedish Quality

Views on quality

- Transcendent – something we learn to recognize
- Product-based – measurable variable
- Usage-based – in the eyes of the beholder
- Manufacturing-based – conformance to requirements
- Value-based – market sets the value

Many opinions ⇒ Statistical techniques

ISO/IEC 25010 (2023)

"The quality of a system is the degree to which the system satisfies the stated and implied needs of its various stakeholders, and thus provides value. Those stakeholders' needs (functionality, performance, security, maintainability, etc.) are precisely what is represented in the quality model, which categorizes the product quality into characteristics and sub-characteristics." (www.iso2500.com)

SOFTWARE PRODUCT QUALITY								
FUNCTIONAL SUITABILITY	PERFORMANCE EFFICIENCY	COMPATIBILITY	INTERACTION CAPABILITY	RELIABILITY	SECURITY	MAINTAINABILITY	FLEXIBILITY	SAFETY
FUNCTIONAL COMPLETENESS	TIME BEHAVIOUR	CO-EXISTENCE	APPROPRIATENESS RECOGNIZABILITY	FAULTLESSNESS	CONFIDENTIALITY	MODULARITY	ADAPTABILITY	OPERATIONAL CONSTRAINT
FUNCTIONAL CORRECTNESS	RESOURCE UTILIZATION	INTEROPERABILITY	LEARNABILITY	AVAILABILITY	INTEGRITY	REUSABILITY	SCALABILITY	RISK IDENTIFICATION
FUNCTIONAL APPROPRIATENESS	CAPACITY		OPERABILITY	FAULT TOLERANCE	NON-REPUDIATION	ANALYSABILITY	INSTALLABILITY	FAIL SAFE
			USER ERROR PROTECTION	RECOVERABILITY	ACCOUNTABILITY	MODIFIABILITY	REPLACEABILITY	HAZARD WARNING
			USER ENGAGEMENT		AUTHENTICITY	TESTABILITY		SAFE INTEGRATION
			INCLUSIVITY		RESISTANCE			
			USER ASSISTANCE					
iso25000.com			SELF-DESCRIPTIVENESS					

Functional suitability

The degree to which a product or system provides **functions that meet stated and implied needs** when used under specified conditions.

- **Functional completeness -** covers all the specified tasks and intended users' objectives.
- **Functional correctness -** provides accurate results for intended users.
- **Functional appropriateness -** facilitates task and objective accomplishment.

Performance Efficiency

The degree to which a product performs its functions **within specified time** and throughput parameters and is efficient in the use of resources under specified conditions.

- **Time behaviour -** meets response time and throughput requirements.
- **Resource utilization -** efficient use of resources when meeting requirements.
- **Capacity -** maximum limits of a product or system parameter meet requirements.

Compatibility

The degree to which a product, system or component can **exchange information with other products**, systems or components, and/or perform its required functions while **sharing the same common environment and resources**.

- **Co-existence -** Ability of a product to function efficiently in a shared environment without negatively impacting other products.
- **Interoperability -** Ability of systems or components to exchange and use information with other products.

Interaction Capability

Usability

Degree to which a product or system can be **interacted with by specified users** to exchange information via the user interface to complete specific tasks in a variety of contexts of use.

- **Appropriateness recognizability -** users can recognize if a product or system meets their needs.
- **Learnability -** ease with which specified users can learn to use a product or system within a specified time.
- **Operability -** ease of operating and controlling a product or system.
- **User error protection -** system's ability to prevent user errors.

(cont'd)

Interaction Capability (cont'd)

- **User engagement -** User interface presents functions and information in an inviting and motivating manner.
- **Inclusivity -** a product or system can be used by people of various backgrounds.
- **User assistance -** the product can be used by people with a wide range of characteristics and capabilities to achieve goals.
- **Self-descriptiveness -** Product presents information to make its capabilities and use immediately obvious without excessive interactions.

Reliability

The degree to which a system, product or component **performs specified functions** under specified conditions for a **specified period of time**.

- **Faultlessness -** Performs specified functions without fault under normal operation.
- **Availability -** Operational and accessible when needed.
- **Fault tolerance -** Operates despite hardware/software faults.
- **Recoverability -** Recovers data and re-establishes desired state after failure.

Security

The degree to which a product or system **defends against attack patterns** by malicious actors and protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.

- **Confidentiality -** Data accessible only to authorized users.
- **Integrity -** Protects system and data from unauthorized changes. (cont'd)

Security(cont'd)

- **Non-repudiation -** Actions/events cannot be denied later.
- **Accountability -** Actions of an entity are traceable to that entity
- **Authenticity -** Identity of subject/resource can be proven.
- **Resistance -** Sustains operations under attack.

Maintainability

The degree of effectiveness and efficiency with which a product or **system can be modified** to improve it, correct it or adapt it to changes in environment, and in requirements.

- **Modularity -** Changes to one component have minimal impact on others.
- **Reusability -** Can be used in multiple systems.
- **Analysability -** Assesses impact of changes and diagnoses deficiencies.
- **Modifiability -** Can be modified without introducing defects.
- **Testability -** Enables creation of test criteria and performing tests efficiently.

Flexibility

The degree to which a product can be **adapted to changes** in its requirements, contexts of use or system environment.

- **Adaptability -** Can be adapted to different environments.
- **Scalability -** Handles varying workloads.
- **Installability -** Can be installed/uninstalled efficiently.
- **Replaceability -** Can replace another product for the same purpose.

Safety

The degree to which a product under defined conditions to **avoid a state in which human life, health, property, or the environment is endangered.**

- **Operational constraint -** Constrains its operation to within safe parameters.
- **Risk identification -** Identifies events that pose risks.
- **Fail safe -** Reverts to safe mode in case of failure.
- **Hazard warning -** Provides warnings of unacceptable risks.
- **Safe integration -** Maintains safety during/after integration.

Agenda:

~~What is quality?~~

How to achieve quality

Mature organizations and CMMI

Document quality work

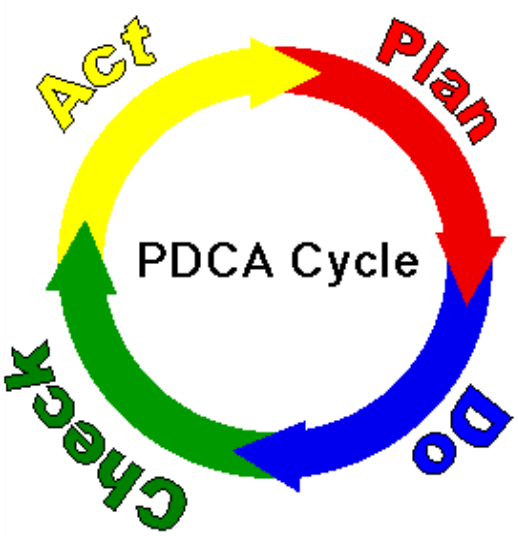
General quality thinking

The Shewhart cycle

Evaluate process

(Change the process)

Evaluate PDCA



Decide goal (the right quality)

Select process (activities)

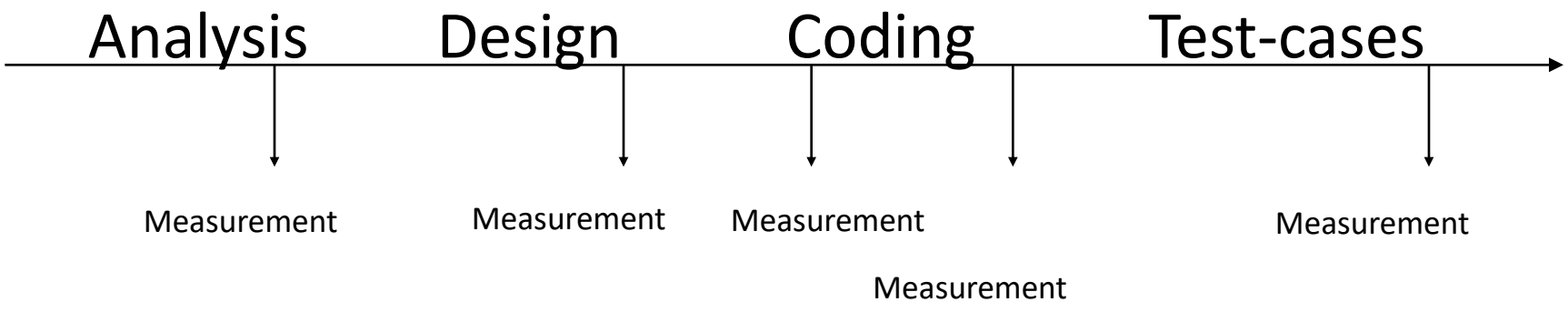
Determine present state

Formulate facts about goal fulfilment

Run the process (project)

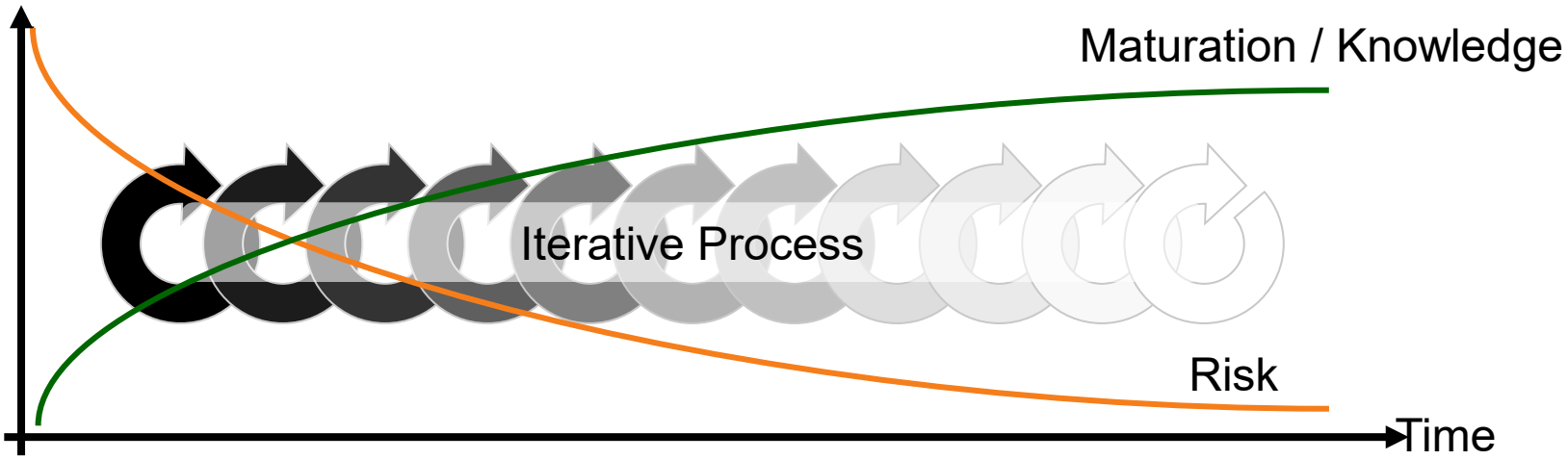
Levels of quality assurance

- Appraisal – eg. defect detection
- Assurance – eg. prediction of defects
- Control – adjust the process
- Improvement: reduce variation, increase precision



Engineering something?

Evaluate goals frequently, for instance, usability.



Agenda:

~~What is quality?~~

~~How to achieve quality~~

Mature organizations and CMMI

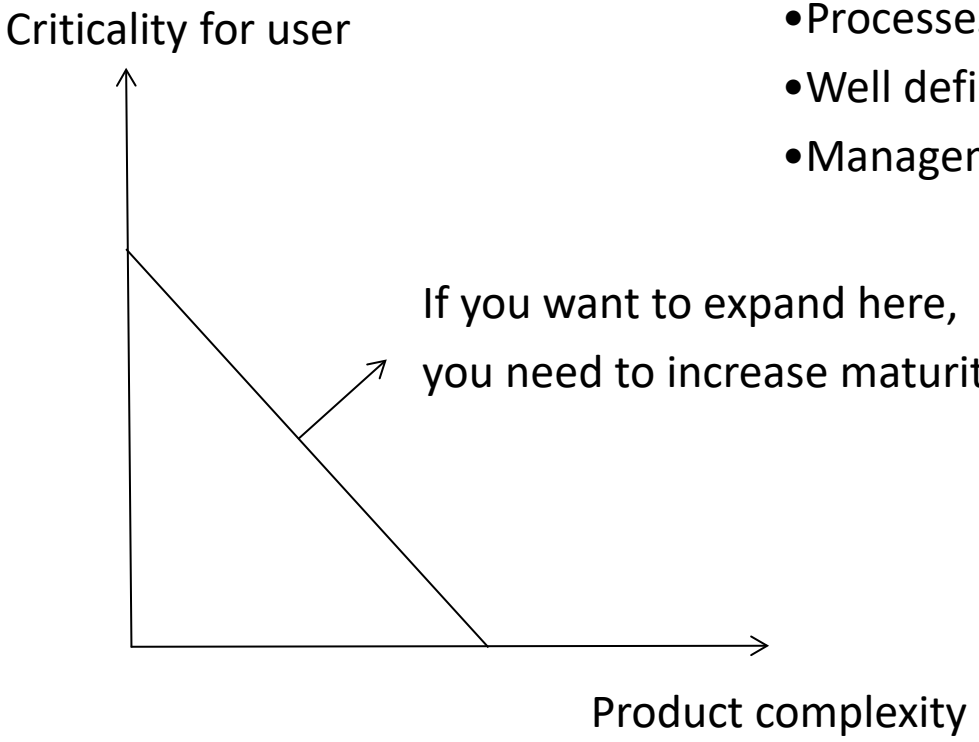
General quality thinking

Document quality work

Argument (originally Weinberg)

A mature organisation has:

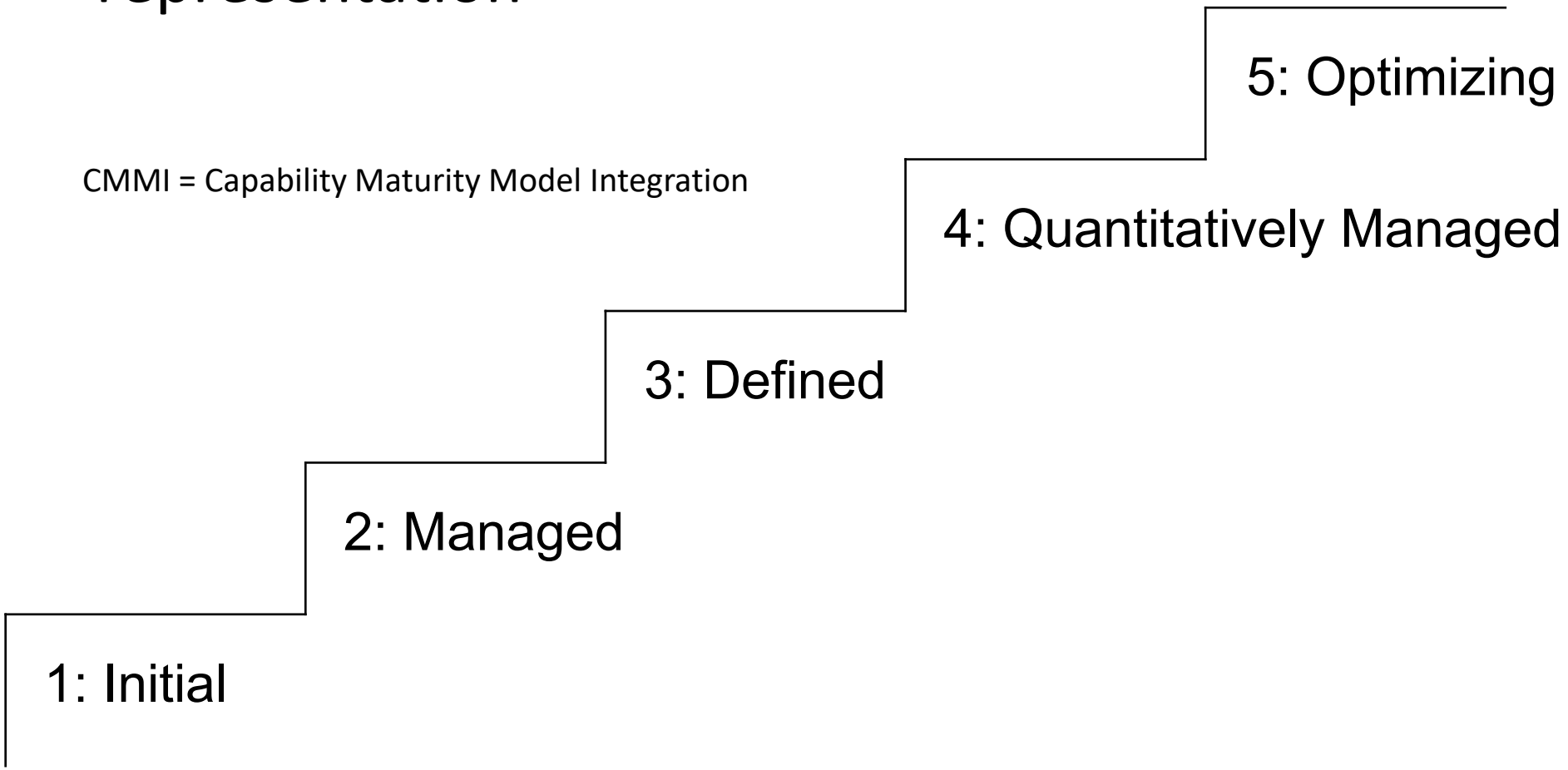
- Inter-group communication and coordination
- Work accomplished according to plan
- Practices consistent with processes
- Processes updated as necessary
- Well defined roles/responsibilities
- Management formally commits



A mature organisation do things well, which does not necessarily mean doing something good.

CMMI for development, staged representation

CMMI = Capability Maturity Model Integration



Life at level 1

The organisation is over-committed, processes are abandoned in crisis, and no repetition of success.

Success is totally dependent on heroes

Life at level 2

- Fewer surprises
- Processes are based on organizational policies
- **Process adherence** is evaluated
- Processes are established and followed even in crisis
- Projects ensure adequate competence and resources
- We know stakeholders' needs
- We can control changes
- The project is visible to managers and other stakeholders at mile-stones and toll-gates
- We can **repeat** a previous success
- Works well for individual projects

Life at level 3

- Tailoring processes from your **own standard** definitions
- Standard processes are **improved**
- Process descriptions are more complete, detailed and rigorous
- Opens for development (and creativity) of alternatives
- Works for a range of projects
- Originally the minimum level

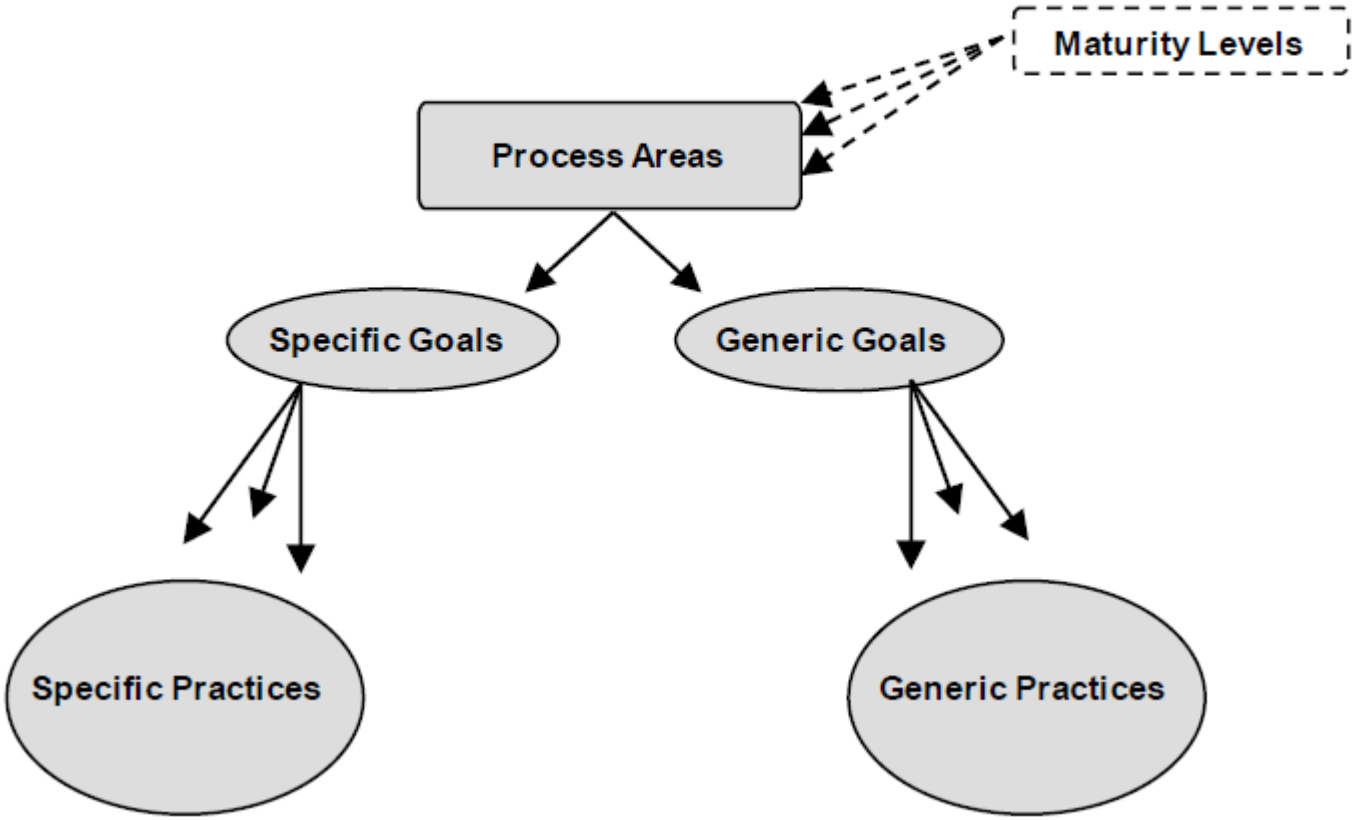
Life at level 4

- Quantitative analysis (statistics) of goals, products, processes
- Higher predictive capability
- Deviations are subject for Root Cause Analysis (RCA, 5Whys)
- Frequent measures

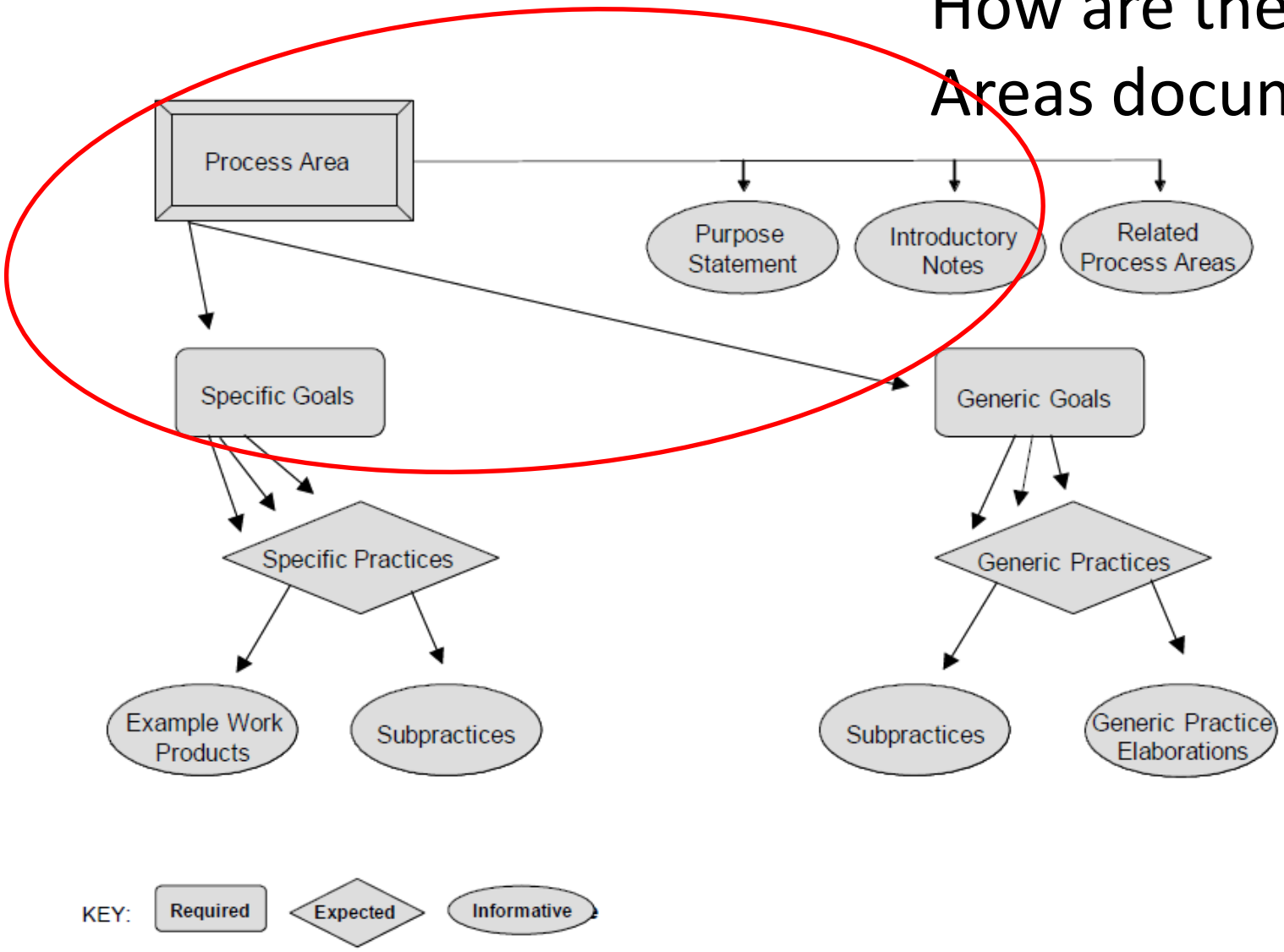
Life of level 5

- Everyone is committed to the continuous improvement of processes
- Innovation climate paired with an ability to evaluate new technology
- The outcome of improvements are evaluated at all relevant levels in the organisation
- You know your gaps in performance
- Challenge: Company culture, new markets
- Used by many sub-contractors for marketing purposes

Staged Representation

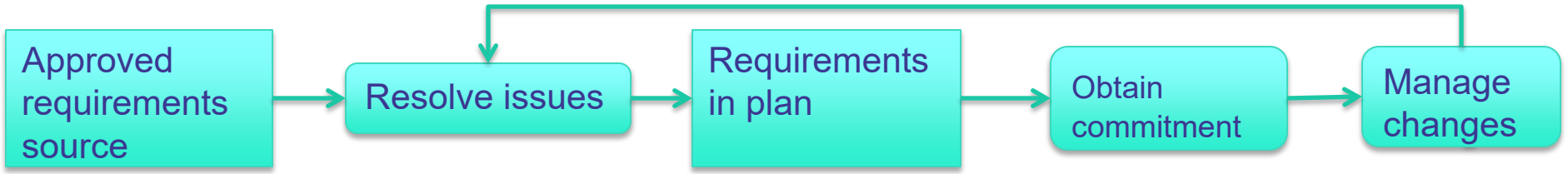


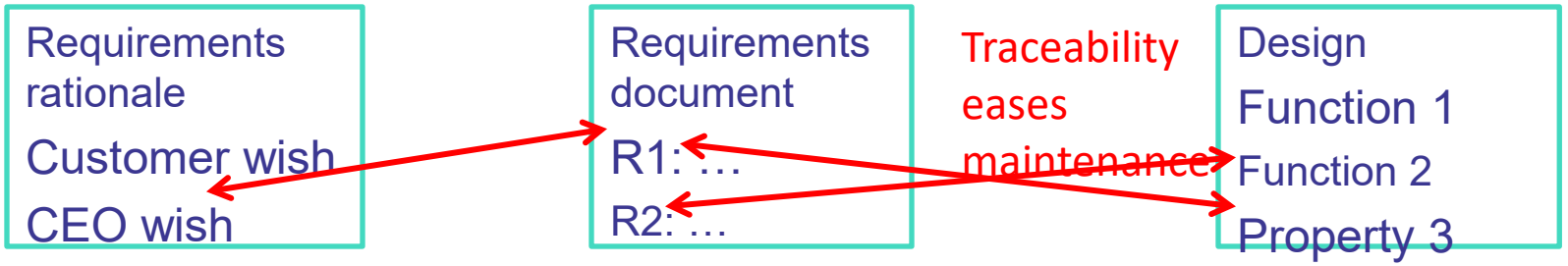
How are the Process Areas documented?



Example: Requirements Management (REQM)

- A Maturity Level 2 Process Area
- Purpose: Manage requirements, ensure alignment to project plan and work products.
- Introductory notes contain:





REQM Specific goal

Map this to your way of working

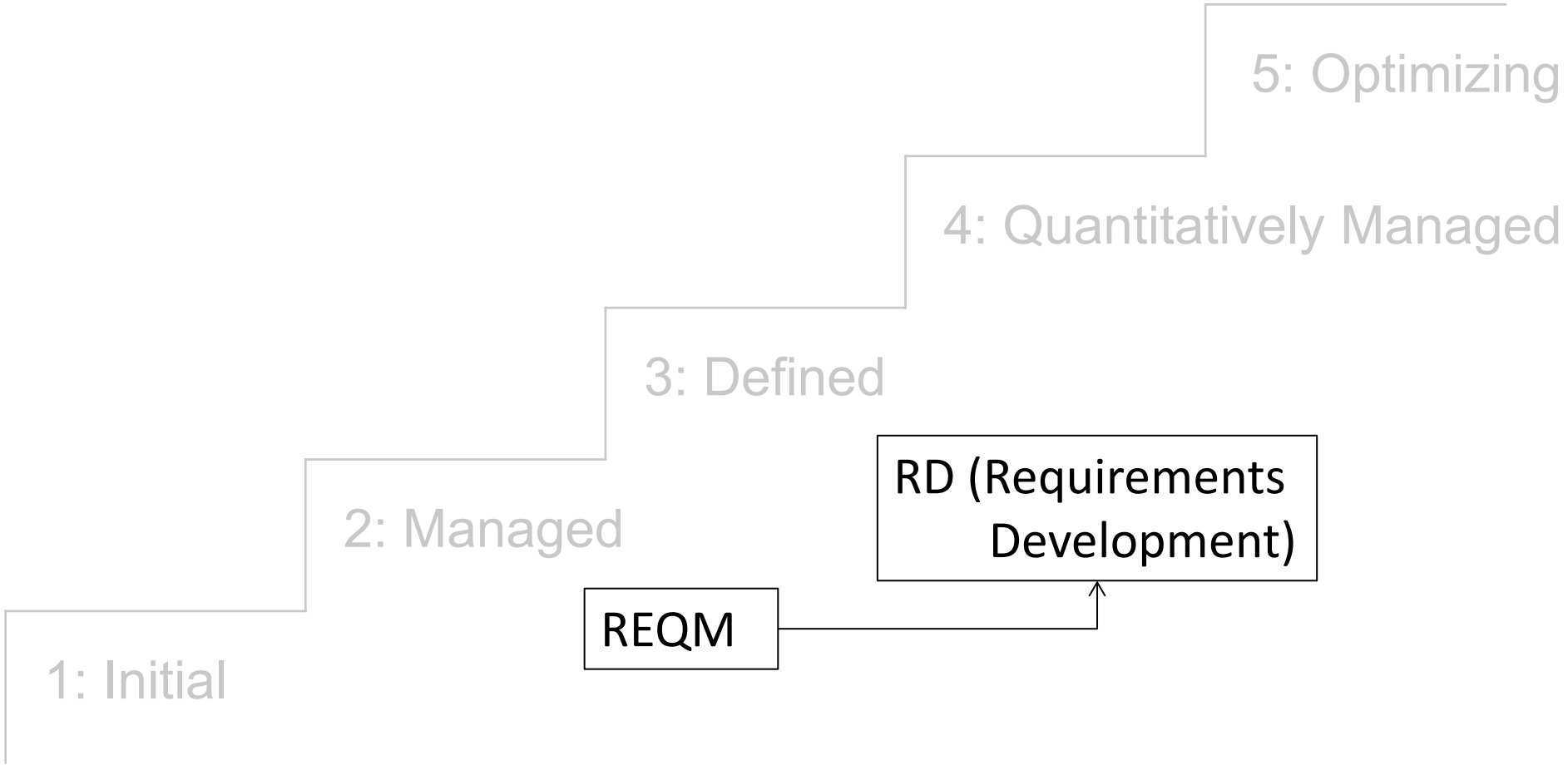
- SG1 Manage Requirements

Requirements are managed and inconsistencies with project plans and work products are identified

  - SP 1.1 Understand Requirements
  - SP 1.2 Obtain Commitment to Requirements
  - SP 1.3 Manage Requirements Changes
  - SP 1.4 Manage Bidirectional Traceability of Requirements
  - SP 1.5 Ensure Alignment Between Project Work and Requirements

Use these to fulfill the goal

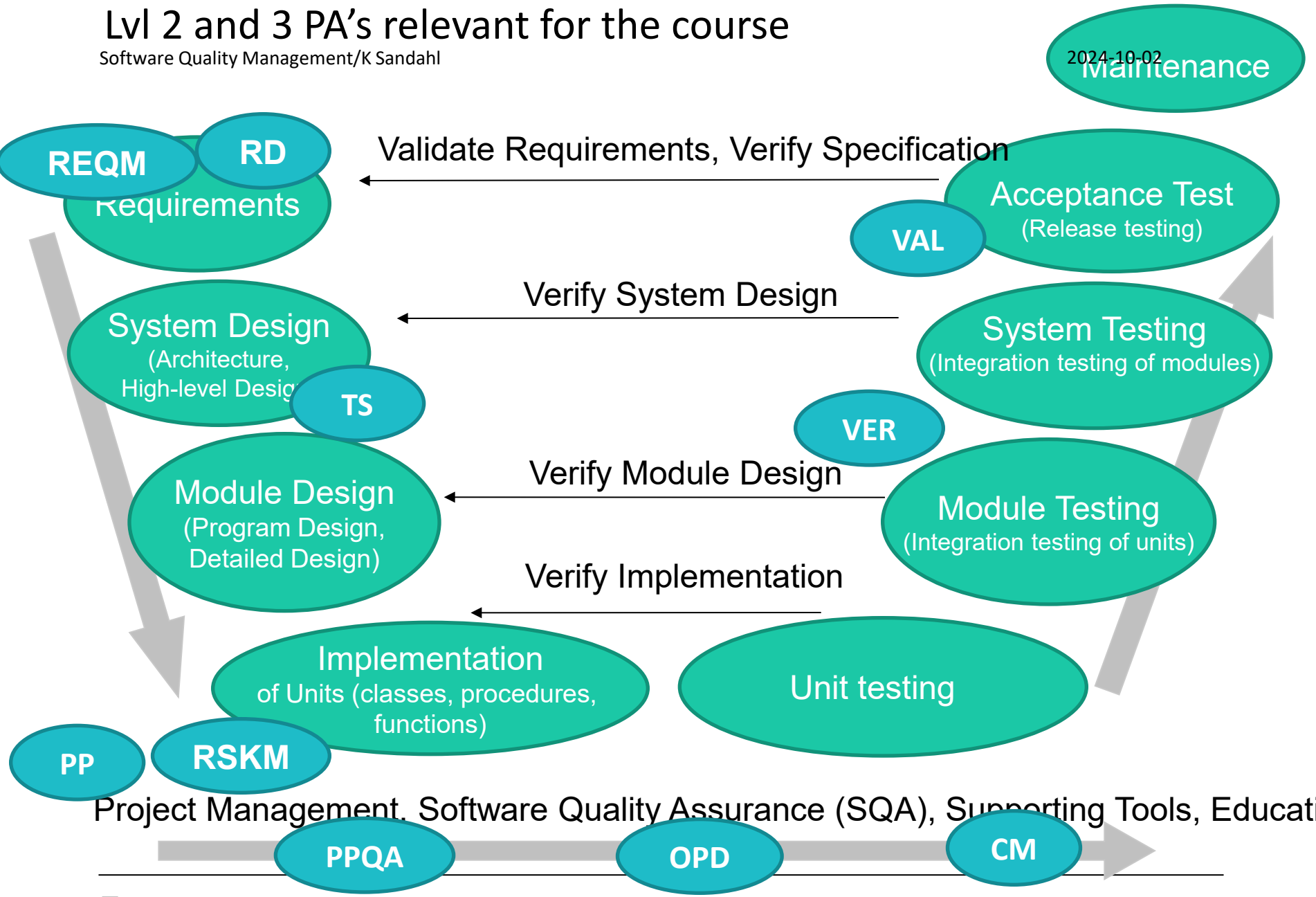
More Mature Requirements Engineering



Requirements Development

- A Maturity level 3 Process Area
- Purpose: Elicit, analyze, and establish customer, product, and product components requirements
- This means:
  - Investigate the true needs of the customer
  - Formulate functional and non-functional requirements, on relevant product levels
  - Validate requirements

Lvl 2 and 3 PA's relevant for the course



Agenda:

~~What is quality?~~

~~How to achieve quality~~

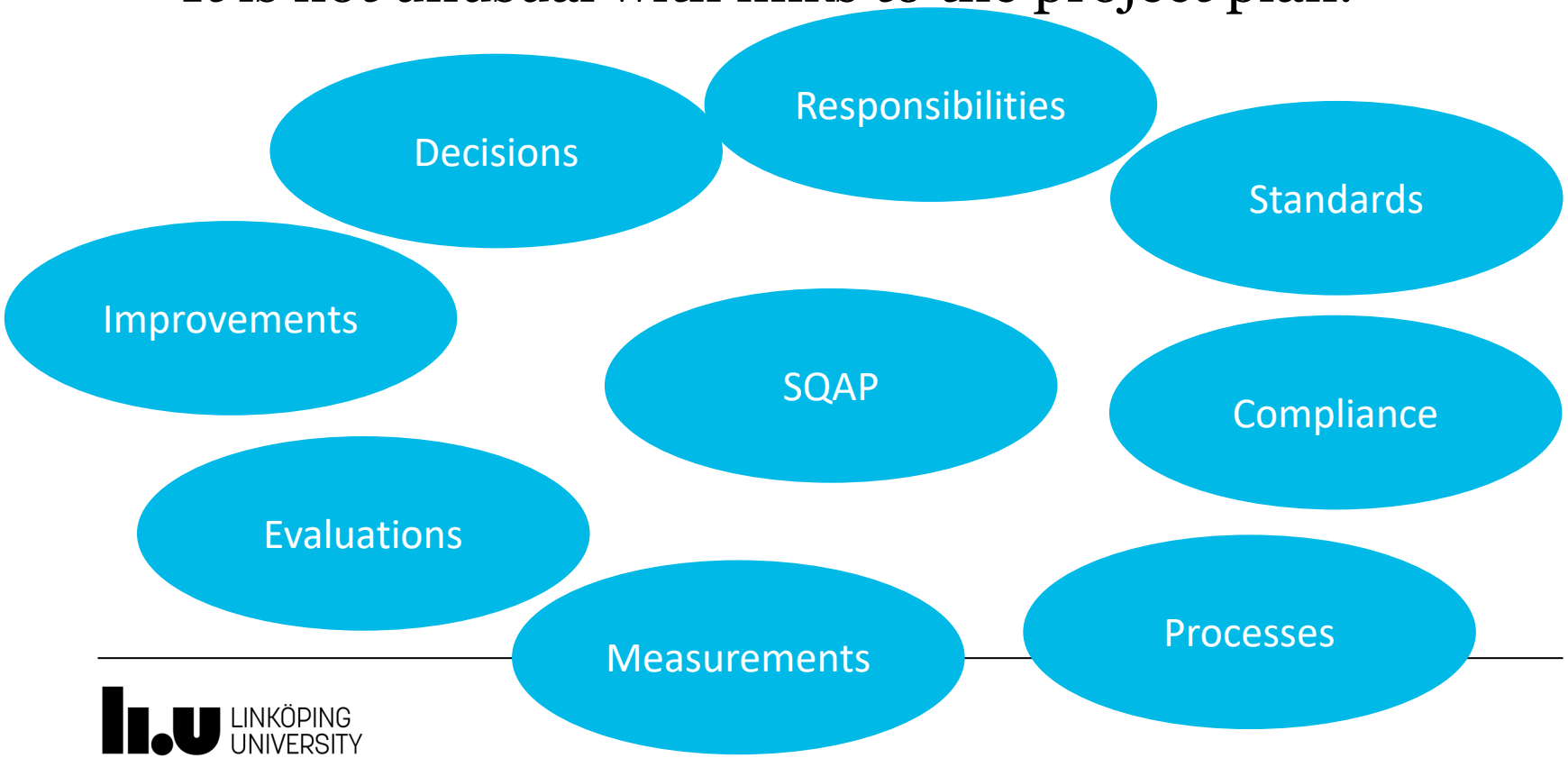
~~Mature organizations and CMMI~~

Document quality work

General quality thinking

Software Quality Assurance Plan

- Readers: Auditors, managers. Team members.
- It is not unusual with links to the project plan.



Processes (and practices)

- Processes contributing to quality
- Static properties
  - Reviews
  - Source code analysis
- Testing (link), how tests contribute to quality
- Other processes:
  - Configuration management
  - Change management
  - Risk management
  - Cooperation and communication
  - -…

Measurement program

- Goal
- Question
- Metrics

- Data sources

- Scope product, processes, and resources
- Non-functional requirements

Evaluations

- Product
- Process
- Compliance

Improvements

- Routines for handling improvements of product and process.
- Who is responsible for initiating improvements?
- Who decides to carry out improvements?
- How do you evaluate improvements?

Standards

- Documents
- Code, style, compilers
- Management, e.g. ISO900-3
- Maturity, e.g. CMMI
- Technical specifications, e.g. Bluetooth,

Compliance

- Standards
- Requirements
  - What are your main quality characteristics?
- Project plan
- Processes and practices

- How to assure compliance
  - How much resources to spend on each activity?

Agenda:

~~What is quality?~~

~~How to achieve quality~~

~~Mature organizations and CMMI~~

~~Document quality work~~

General quality thinking

Das Auto

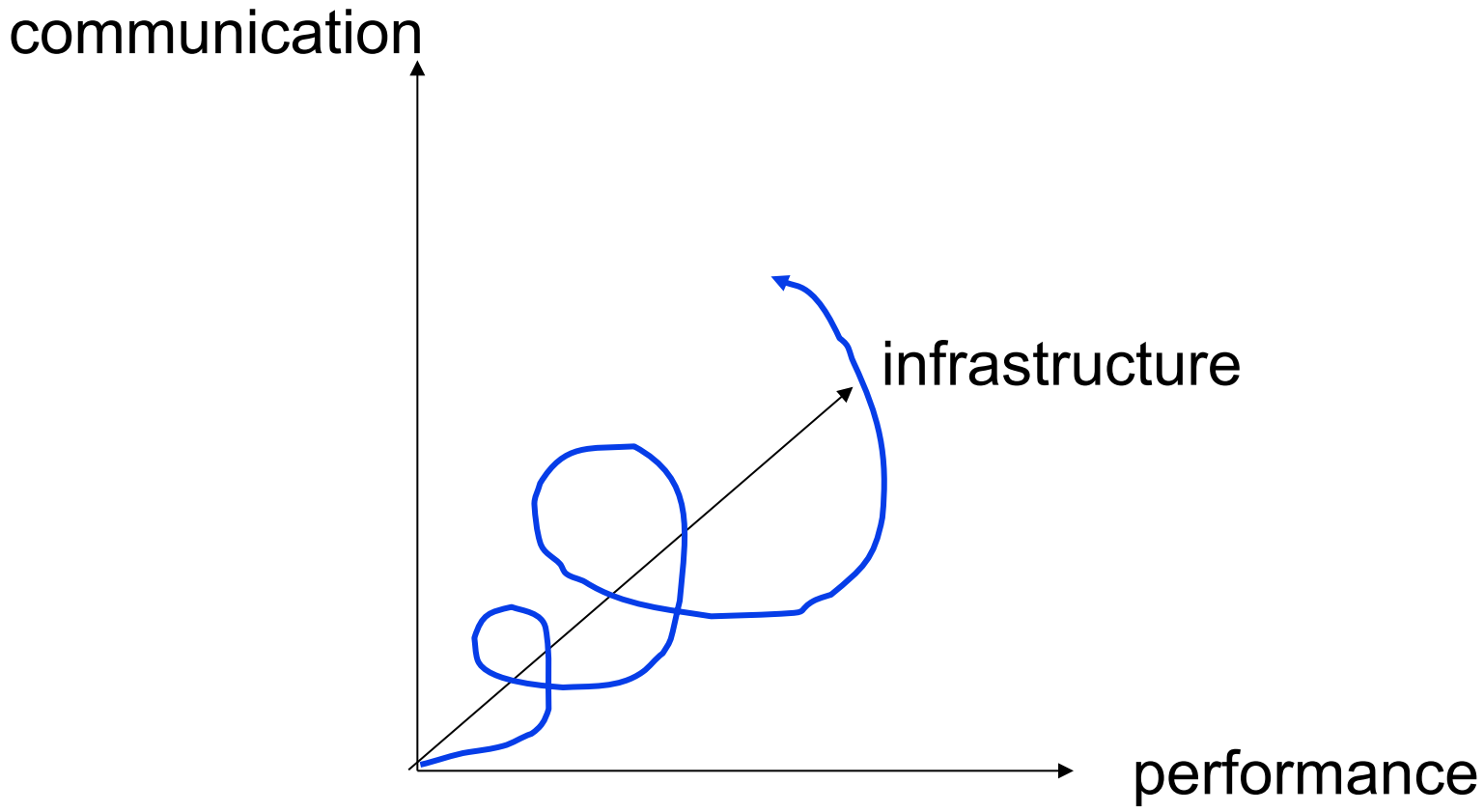

Linköpings universitet
PASSAT
Gröna mil
med LiU-bil
S
LIU
www.liu.s

ISO 9000-3

- ISO 9000-3 is guideline to apply ISO 9001 to software industry, which is built on the principles:
- *Principle 1* Customer focus
- *Principle 2* Leadership
- *Principle 3* Involvement of people
- *Principle 4* Process approach
- *Principle 5* System approach to management
- *Principle 6* Continual improvement
- *Principle 7* Factual approach to decision making
- *Principle 8* Mutually beneficial supplier relationships

- ISO = International Organization for Standardization
- The Swedish member: SIS = Swedish Standards Institute (sic!)

Wisdom



Summary

- Plan Do Check Act cycle
- ISO/IEC 25010 (2023)
- A mature organization
- CMMI (staged version)
- Software Quality Assurance Plan
- Principles of ISO 9000-3

All Unite for Software Quality

Performed by Kristian Sandahl

Music and lyrics by Dániel Varró

Lyrics

Verse 1

There are ambiguities hidden in the specs
Small inconsistencies in requirements
These are all unstructured verbose documents
Your software lacks a certain quality.

Verse 2

Why do I find circles in inheritance trees?
The architecture's broken, no redundancies
Anti-patterns, code smells everywhere I see
Your software lacks a certain quality.

Lyrics (cont.)

Refr:

Metrics reveal what's inside

Test and inspect to decide

Take a chance for software quality.

Make the process more agile

Interfaces less fragile

Please unite for software quality.

Lyrics (cont.)

Verse 3

Do you need a big bang or a sandwich test?
Covering all the branches seems like quite a quest
To control all the versions, Github is the best
Your software needs a certain quality.

Verse 4

Rarely shines a rainbow over a waterfall process
The chart burns down to highlight how your sprints progress
The peer review will walk through what you must confess
Your software needs a certain quality.

Lyrics (cont.)

Refr:

Metrics reveal what's inside

Test and inspect to decide

Take a chance for software quality.

Make the process more agile

Interfaces less fragile

Please unite for software quality.

Finale

Everybody together:

All unite for software quality

All unite for software quality

All unite for software quality

All unite for software quality

Kristian:

No ambiguities

No anti-patterns

All versions controlled

All sprints progressing

All unite for software quality

www.liu.se

LINKÖPING UNIVERSITY