Software Engineering
Quality Management and Process Improvement

Stefan Hallerstede, Carl Schultz

Aarhus University Department of Engineering

20 November 2019

Contents

Contents

Question:

What is software quality?

Question:

What is software quality?

Simple answer: Software should meet its specification.

Question:

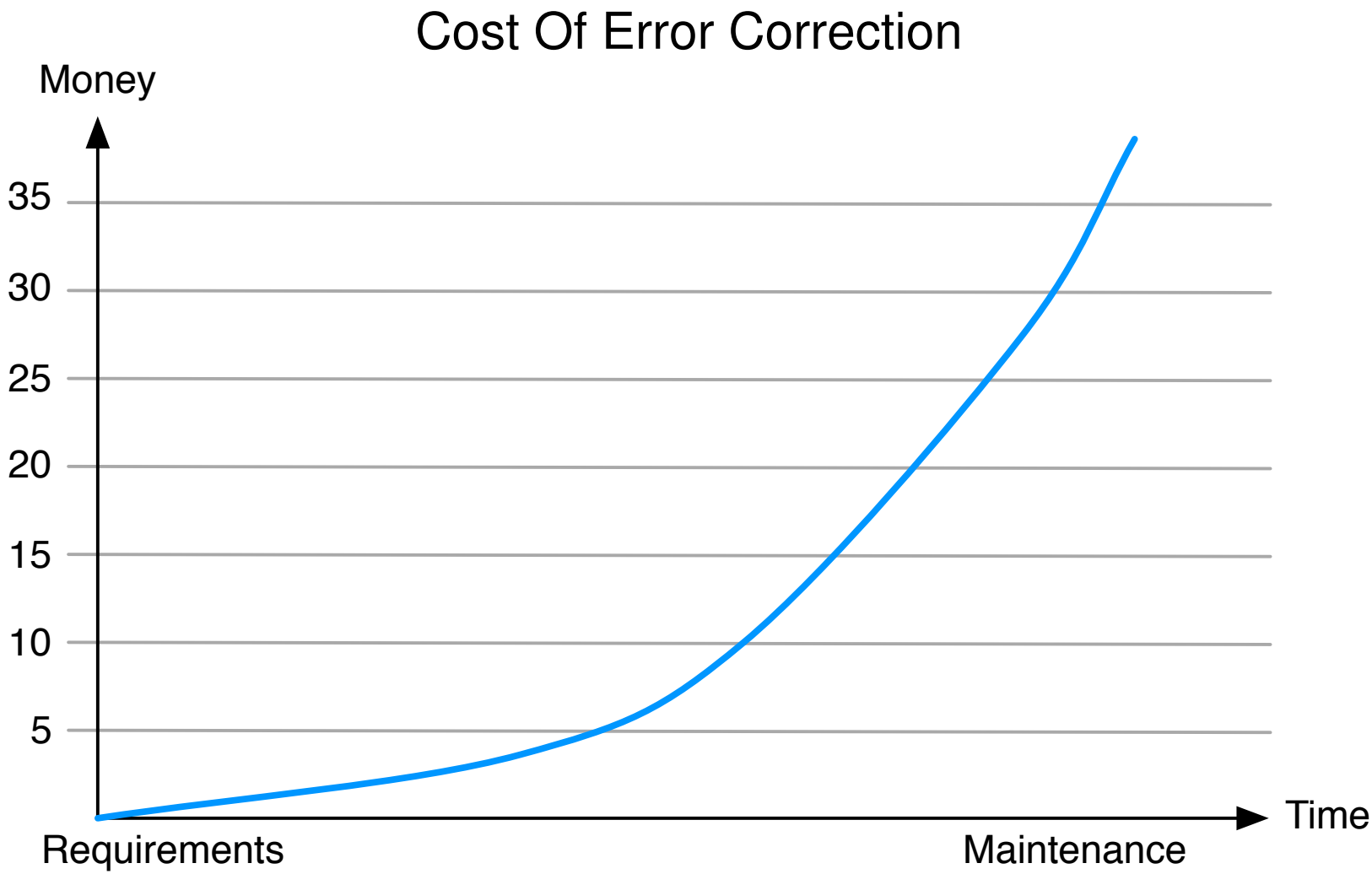
What is software quality?

Simple answer: Software should meet its specification.

What about:

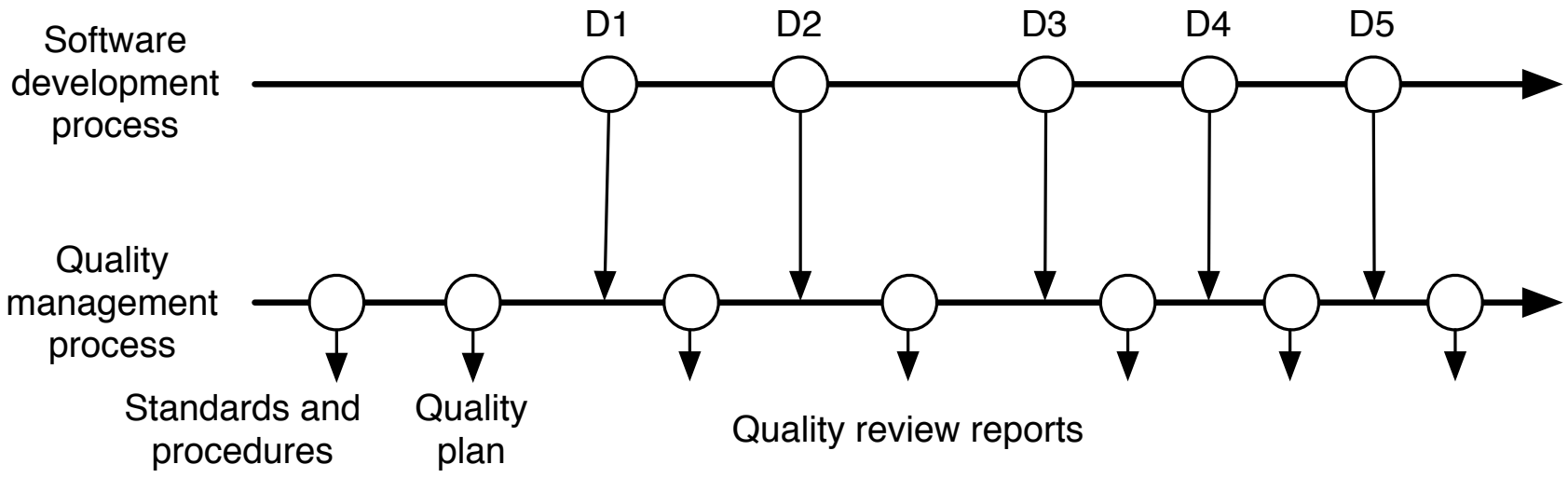
- Usability?
- Efficiency?
- Reliability?
- Maintainability?
- Portability?
- Reusability?

Why is quality important?



Contents

Quality Management And Software Development



Quality management must occur *continually* during software development.

Techniques Of Validation And Verification

Targeted At High-Quality Products

- *Testing*
- Formal specification
- Formal verification
- Test case generation
- Model-checking
- Formal proof
- Informal proof
- Animation
- Simulation
- Visualisation

Quality Management Activities

- **Quality assurance**
  The establishment of a framework of organisational procedures and standards that lead to high-quality software
- **Quality planning**
  The selection of appropriate procedures and standards from this framework, adapted for a specific software project (cf. Project Planning)
- **Quality control**
  The definition and enactment of processes that ensure the software development team have followed project quality procedures and standards

Contents

Quality Assurance And Standards

- **Product standards**
  - *Document standards*
    (e.g. structure of requirements documents)
  - *Documentation standards*
    (e.g. standard comment headers for object classes)
  - *Coding standards*
    (e.g. how some programming language is to be used)
- **Process standards**
  E.g.
  - Definition of specification, design and validation *processes*
  - Description of *documents* that should be written in the course of these processes

Why Software Standards Are Useful

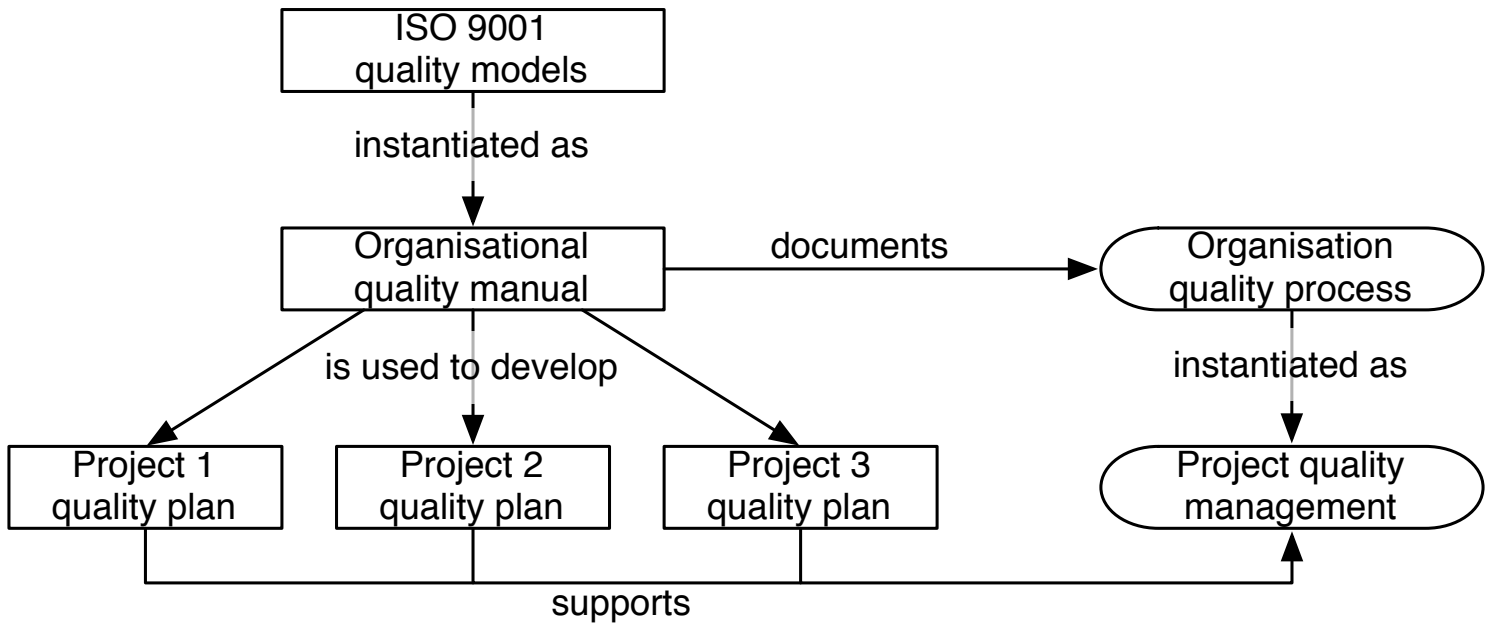
Software standards

- record *best practices*
- help to avoid repeating *past mistakes*
- provide basis for *quality assurance process*
- assist in *continuity and conformity* among staff
- reduce *learning effort* when starting new work

Example[1]

The ISO 9001 Quality Standard

- ISO 9001 Quality Systems—Model for Quality Assurance in Design, Development, Production, Installation and Servicing
- ISO 9000-3. Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software
- ISO 9004-2. Quality Management and Quality System Elements—Part 2

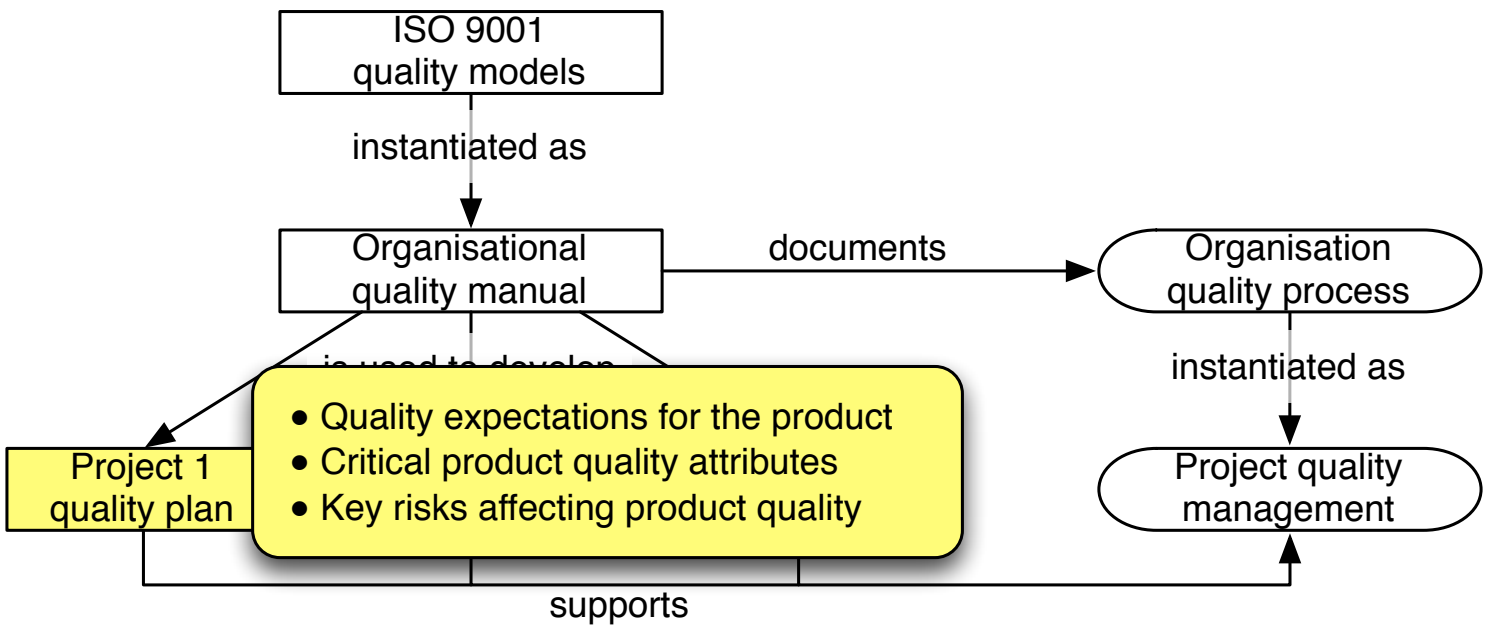


[1] R. S. Pressman (2000) *Software Engineering – A Practitioner's Approach*. McGraw-Hill

Example[1]

The ISO 9001 Quality Standard

- ISO 9001 Quality Systems—Model for Quality Assurance in Design, Development, Production, Installation and Servicing
- ISO 9000-3. Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software
- ISO 9004-2. Quality Management and Quality System Elements—Part 2



[1] R. S. Pressman (2000) *Software Engineering – A Practitioner's Approach*. McGraw-Hill

Software Quality Attributes

Safety	Understandability	Portability
Security	Testability	Usability
Reliability	Adaptability	Reusability
Resilience	Modularity	Efficiency
Robustness	Complexity	Learnability

Contents

Approaches To Quality Control

See earlier lecture:

- Reviews and inspections
- Automated software assessment

Contents

Purpose Of Process Improvement

Quality of development *processes*
and quality of developed *products* are closely related.

Process improvement means:

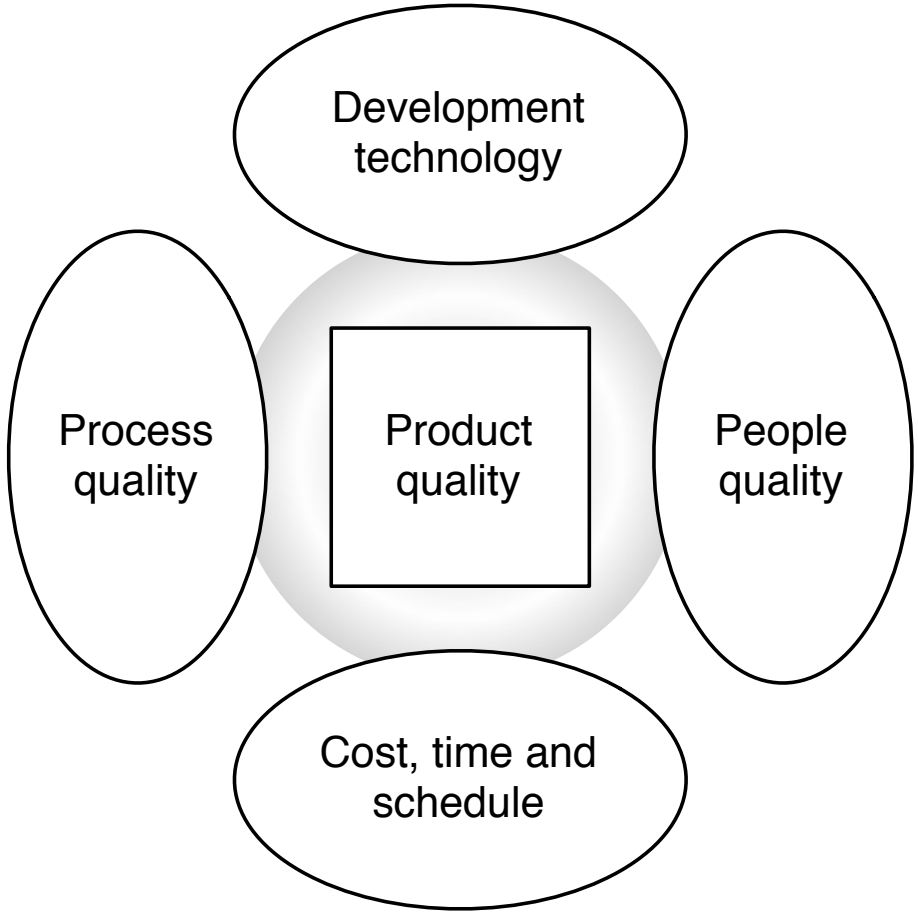
- understanding existing processes and
- changing these processes to
  - increase *product quality*
  - reduce cost and development time

Contents

Process Characteristics

Understandability	To what extent is the process defined and how easy is it to understand the process definition?
Visibility	Do the process activities culminate in clear results so that the progress of the process is externally visible?
Supportability	To what extent can CASE tools be used to line support the process activities?
Acceptability	Is the defined process acceptable to and usable by the engineers producing the software product?
Reliability	Are process errors avoided or trapped before they result in product errors?
Robustness	Can the process continue in spite of unexpected problems?
Maintainability	Can the process evolve to reflect changing organisational requirements or identified process improvements?
Rapidity	How fast can the process of delivering a system from a given specification be completed?

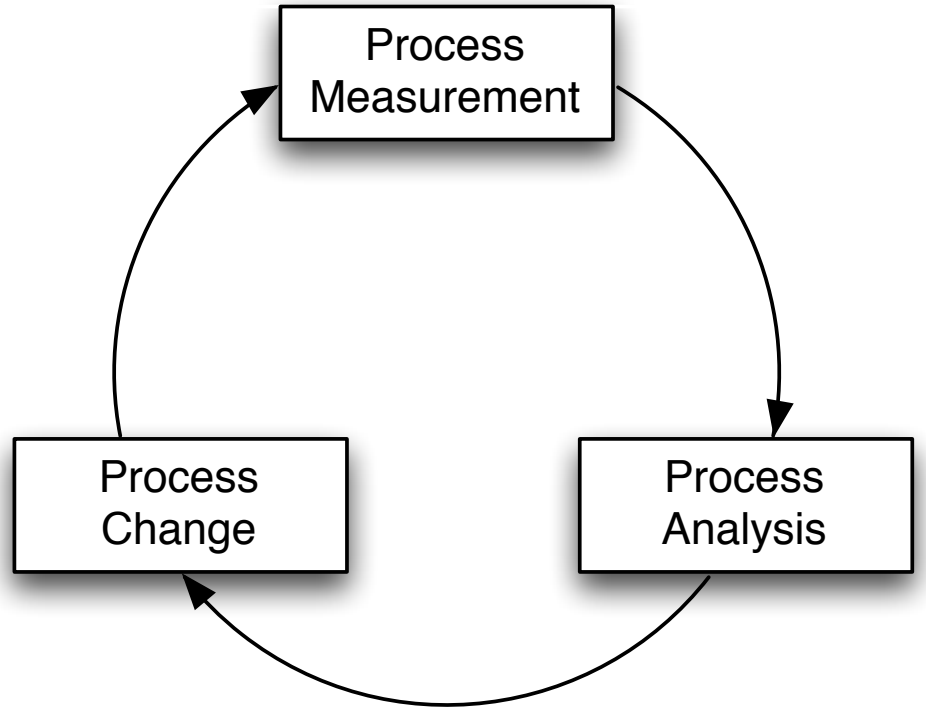
Principal Software Product Quality Factors



- Development technology refers to the basic level required (not sophisticated CASE tools)
- Schedule and available resources are essential

Contents

Process Improvement As Cyclic Activity



- Process Measurement:
  measure attributes of the current process
- Process Analysis:
  assess current process, identify weakness and bottlenecks
- Process Change:
  introduce corresponding changes to the process

Contents

Classification Of Processes

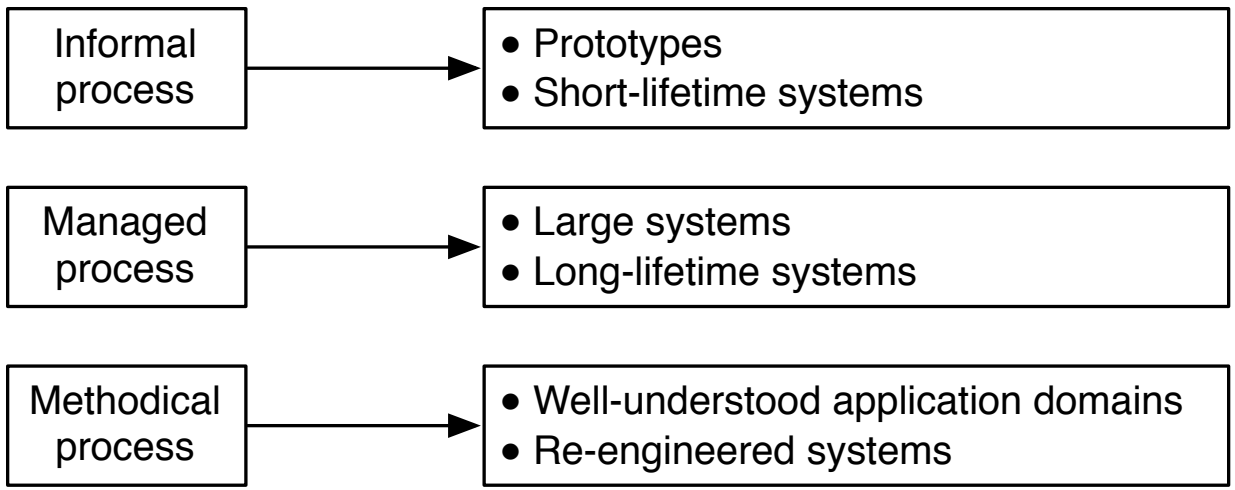
- **Informal**: development team chooses the process they will use
- **Managed**: defined process model drives the development process
- **Methodical**: defined development methods supported by CASE tools
- **Improving**: improvements considered and introduction procedures

Classification Of Processes

- **Informal**: development team chooses the process they will use
- **Managed**: defined process model drives the development process
- **Methodical**: defined development methods supported by CASE tools
- **Improving**: improvements considered and introduction procedures

---

Applicability of processes depending on classification:



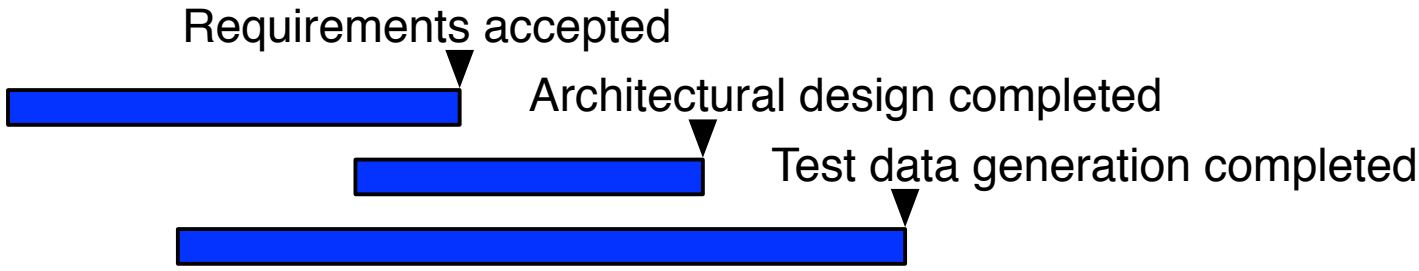
Processes may belong to more than one class (e.g. *informal and methodical*, relying on standards and associated tools)

Contents

Process Measurement

Classes Of Process Metrics

- Time taken for a particular process to be completed



- Resources required for a particular process
  E.g.
  - total effort in person-days
  - computer resources
- The number of occurrences of a particular event
  E.g.
  - defects discovered during code inspection
  - number of requested requirement changes
  - average number of lines of code modified in response to a requirement change

What To Measure?

Possible approach: *Goal-Question-Metric* (GQM) paradigm

- **Goals** What is the organisation trying to achieve?
  e.g. increased product reliability
- **Questions** Identify specific areas of uncertainty related to goals
  e.g. How can more effective reliability assessments be made?
- **Metrics** Measurements to answer questions and confirm improvements
  e.g. the number of tests required to cause product failure

Contents

Process Analysis

Understanding The Process In Use

Rely on:

- "Formal" process models
  - specifies activities, deliverables
- Questionnaires and interviews
  - question the engineers about what actually goes on
  - refine answers by subsequent interviews
- Ethnographic studies:
  - observe the working environment to gain understanding

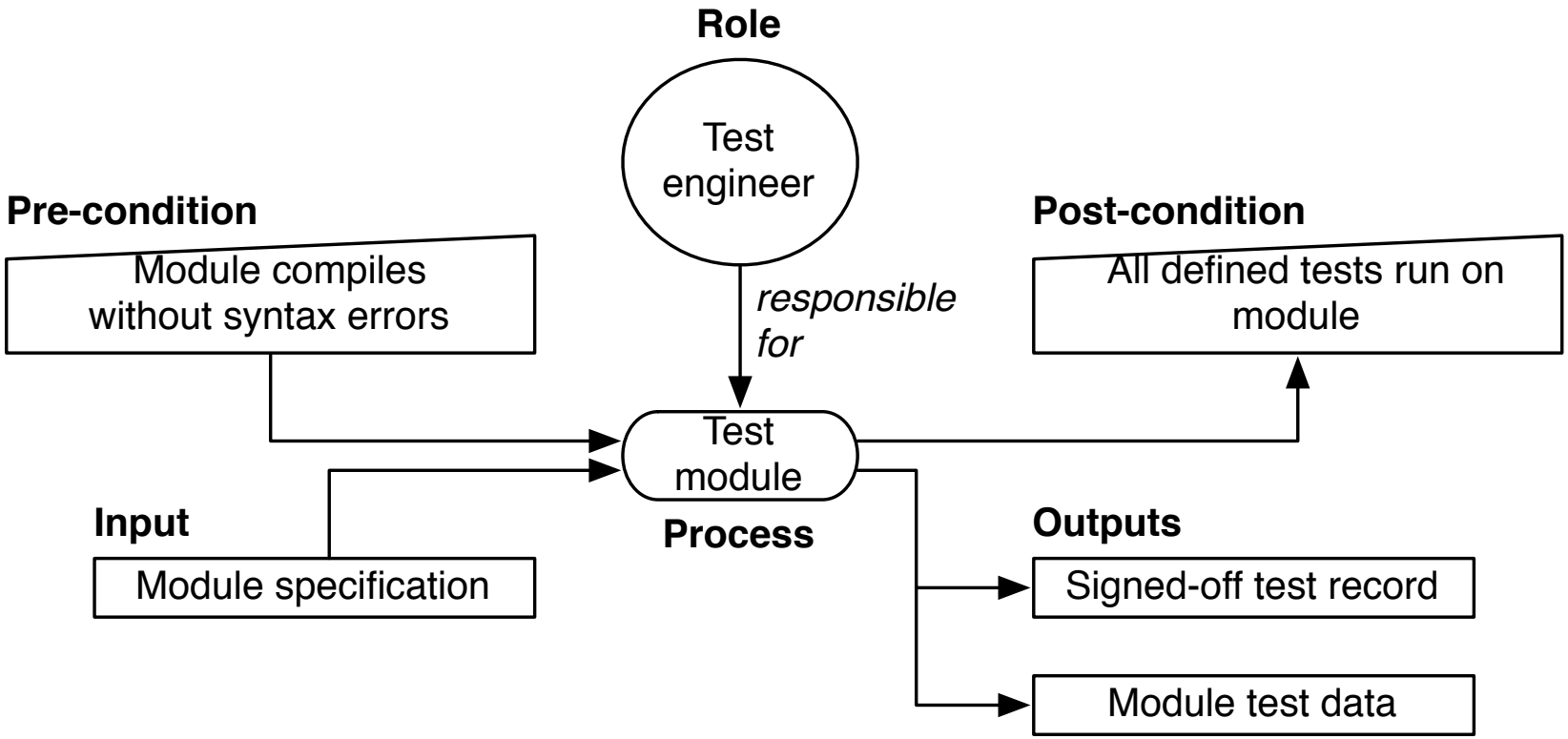
Warning:
Process models can only be approximations of real processes!

Process improvement (among other things) can be based on:

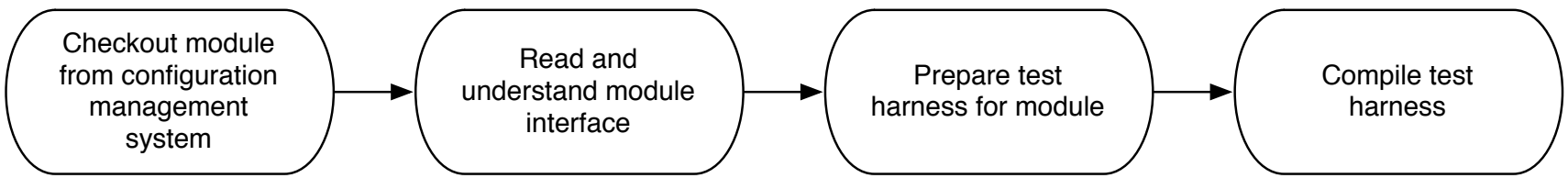
- activities
- deliverables
- people
- communications
- schedules

Example: A Process Fragment Concerning Testing

Testing process:



Activity fragment *Module test harness preparation* of the testing process:

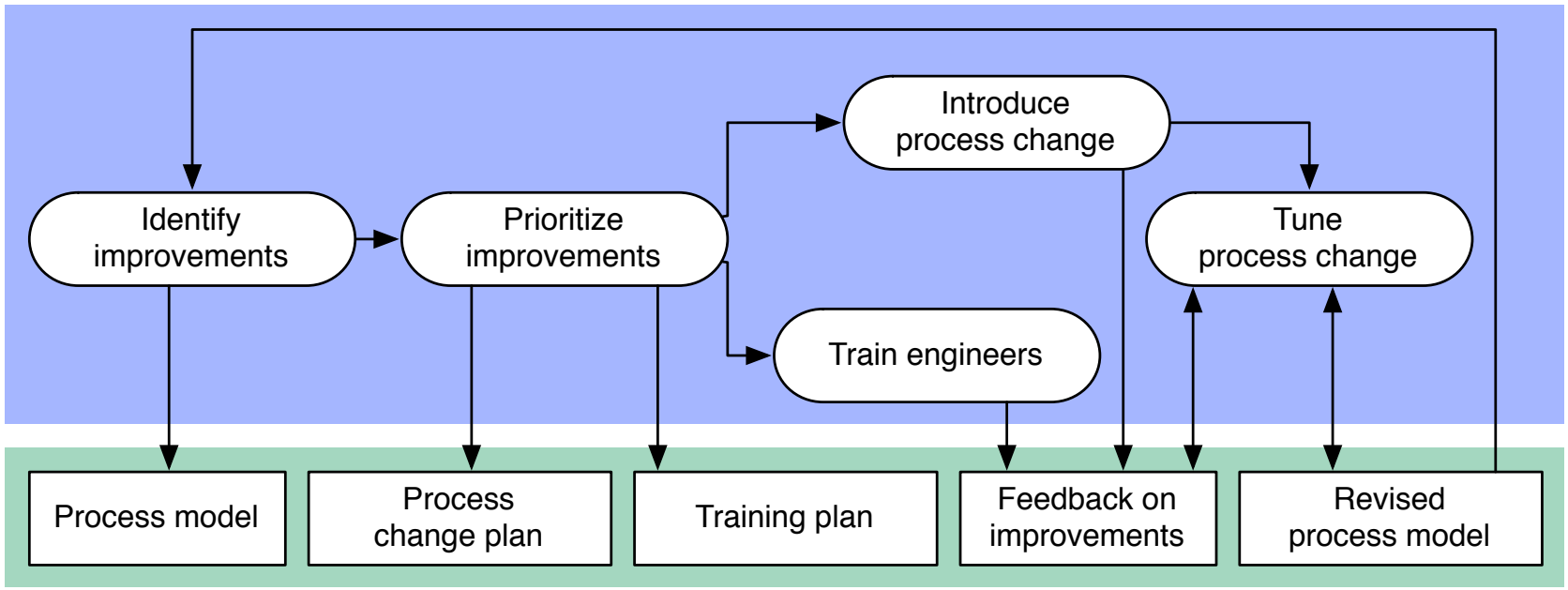


Contents

Changing processes

Phases:

- Improvement identification
- Improvement prioritisation
- Process change introduction
- Process change training
- Change tuning



Set measurable goals such that progress can be measured (e.g. “reduce number of defects during integration testing by 25%”)

Contents

The CMMI Process Improvement Framework

- CMMI: Integrated capability maturity model
- Very complex (more than 1,000 pages of description)
- Simplified structure:
  - **Process areas:**
    identifies 24 process areas relevant to software process capability and improvement, *e.g.*
    Requirements management, Requirements development
  - **Goals:**
    abstract descriptions of desirable states to be attained by organisations, *e.g.*
    The requirements are analysed and validated, and a definition of the required functionality is developed
  - **Practices:**
    descriptions of recommended ways to achieving a goal, *e.g.*
    Analyse derived requirements to ensure that they are necessary and sufficient

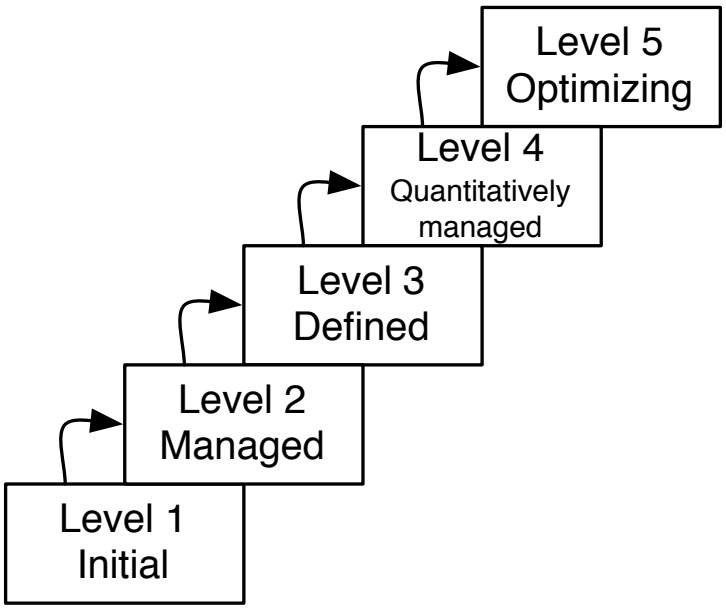
CMMI process assessment scale

- **0. Not performed:**
  Some goals associated with the process area are *not satisfied*
- **1. Performed:**
  All goals associated with the process area are *satisfied*
- **2. Managed:**
  All goals satisfied and *organisational policies* define use of processes
- **3. Defined:**
  Each project has a managed process from a defined set of organisational processes (*organisational standards*)
- **4. Quantitatively managed;**
  Use of statistical and other quantitative methods to *control processes*
- **5. Optimizing:**
  Use of process and product measurement to drive *process improvement*

Two Variants Of CMMI Model

The staged model

Fixed set of goals at each stage

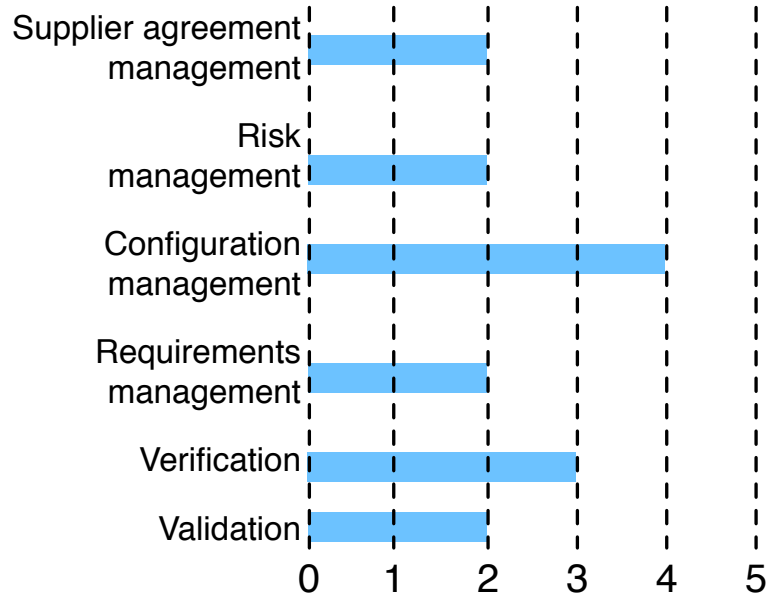


Disadvantage:

Difficult to fit organisations into such a rigid scheme

The continuous model

Differentiated rating for processes or process groups



Main advantage:

Can easily be tailored to an organisation's needs

Contents

Concluding Remarks

- Process improvement measures how well a process is performing
- It can be targeted at improved quality or improved efficiency
- Models used for process improvement are imprecise
- Using the right metrics is essential