Software Quality Management

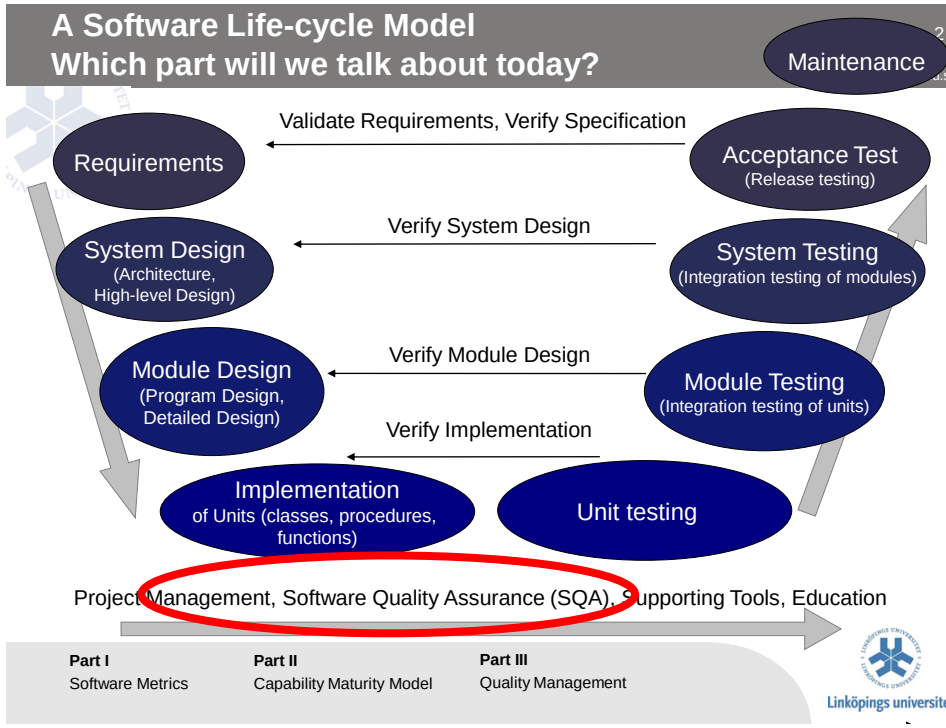
Lecture 9

Software Engineering
CUGS
Spring 2011

Kristian Sandahl
Department of Computer and Information Science
Linköping University, Sweden
krisa@ida.liu.se

Linköpings universitet

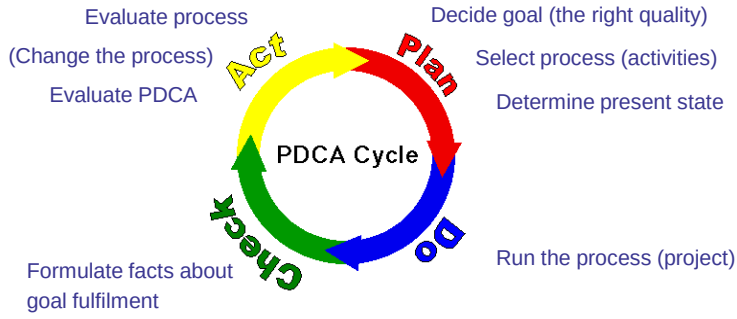
A Software Life-cycle Model
Which part will we talk about today?



Part I Software Metrics
Part II Capability Maturity Model
Part III Quality Management

Linköpings universitet

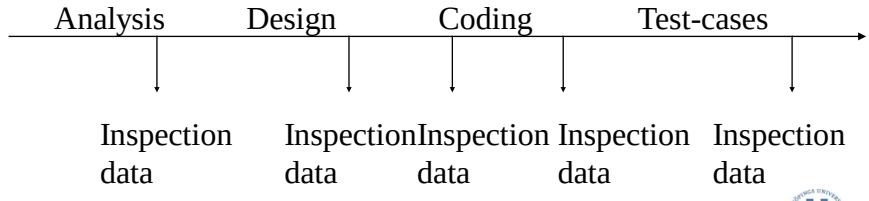
The Shewhart cycle



Inspections in quality assurance

Your default choice!

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

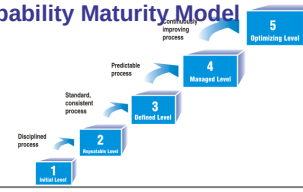


Agenda - What will you learn today?

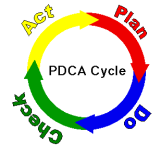
Kristian Sandal
krisa@ida.liu.se

Part I
Software Metrics

Part II
Capability Maturity Model



Part III
Quality Management



Kristian Sandal
krisa@ida.liu.se

Part I
Software Metrics

Quality factors

Kristian Sandal
krisa@ida.liu.se

- Correctness
- Reliability
- Efficiency
- Usability
- Integrity
- Maintainability
- Flexibility
- Testability
- Security
- Portability
- Reusability
- Interoperability
- Survivability
- Safety
- Manageability
- Supportability
- Replaceability
- Functionality

Measuring these requires both research, experience and imagination.

Software metrics

Kristian Sandal
krisa@ida.liu.se

- Usage-based metrics
- Verification & Validation metrics
- Volume metrics
- Structural metrics
- Effort metrics
- Direct measurement
- Indirect measurement

Note: Pedagogical model only!

Usage based metrics - example

Kristian Sandal
krisa@ida.liu.se

- Description: Number of good and bad features recalled by users.
- How to obtain data: Set up a test scenario. Let test users run the scenario. Collect number of good and bad features in a questionnaire afterwards.
- How to calculate the metric: Take the average of number of good and bad features. Two values.
- Relevant quality factor: Relevance – many good and few bad features indicates a good match with the users' mindset.

Verification and validation metrics - example

Kristian Sandal
krisa@ida.liu.se

- Description: Rate of severe defects found in inspection of design description.
- How to obtain data: Perform an inspection according to your process. Make sure that severity is in the classification scheme.
- How to calculate the metric: Divide the number of defects classified with highest severity with total number of defects in the Inspection record.
- Relevant quality factor: Safety – a high proportion of severe defects in design indicates fundamental problems with the solution and or competence.

Volume metrics - example

Kristian Sandal
krisa@ida.liu.se

- Description: Number on non-commented lines of code.
- How to obtain data: Count non-commented lines of the code with a tool.
- How to calculate the metric: See above.
- Relevant quality factor: Reliability – it is often hard to understand a large portion of code, the fault density is often higher for large modules.

Structural metrics - example

Kristian Sandal
krisa@ida.liu.se

- Description: Maximum depth of inheritance tree.
- How to obtain data: Count the depth of the inheritance tree for all classes with a tool.
- How to calculate the metric: Take the maximum value of the classes.
- Relevant quality factor: Understandability – It is hard to determine how a change in a higher class will affect inherited/overridden methods.

Effort metrics - example

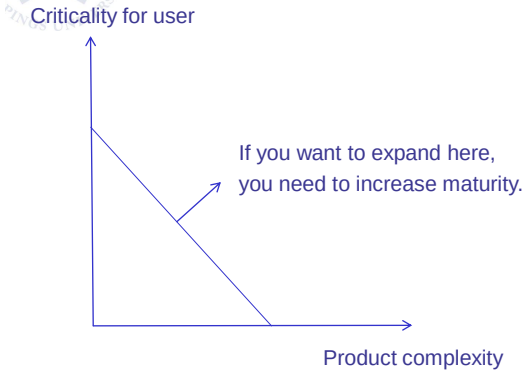
- Description: Time spent in testing.
- How to obtain data: Make sure that testing activities are distinguished in time reporting forms. Make sure that all project activities are reported.
- How to calculate the metric: Sum the number of hours for all activities in testing for all people involved.
- Relevant quality factor: Testability – a comparably long testing time indicates low testability.

Part II

Capability Maturity Model

Argument (originally Weinberg)

Kristian Sandal
krisa@ida.liu.se



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, staged structure

Kristian Sandal
krisa@ida.liu.se

CMMI = Capability Maturity Model Integration

5: Optimising

4: Managed

3: Defined

2: Repeatable

1: Initial

Each level has process areas.

Tutorial at: http://www.tutorialspoint.com/cmmi/

Life at level 1

Totally dependent on heroes

Over-committed, abandoned processes, no repetition of success.

Process area CMMI2

- Requirements Management
  - Analyzes requirements
  - Verify plans and products against requirements
- Project Planning
  - Plan: activities, resources, schedules
  - Use the plan to achieve commitment and approval at all levels
- Project Monitoring and Control
  - Progress control: Initiate changes to solve problems
  - Update plans

PA CMMI2 (continued)

- Supplier Agreement Management
  - Manage contracts, progress control, quality assurance.
- Measurement and Analysis
  - Develop, initiate, analyse and complete measurements to support progress control
- Process and Product Quality Assurance
  - Develop, implement and follow up application of SQA tools for processes and software products
- Configuration Management
  - Develop, implement and operate a CM system
  - Assure integrity of work products, report changes, test configuration

Life at level 2

- Fewer surprises
- Processes are established and followed even in crisis
- We know stakeholders' needs
- We can control changes
- We can **repeat** a previous success
- Works well for individual projects

PA CMMI3

Kristian Sandal
krisa@ida.liu.se

- Requirements Development
  - Elicit, specify, analyze and validate requirements.
- Technical Solution
  - Develop, analyze and **select** solutions to components or the system as a whole
  - Implement solution
- Product Integration
  - Integrate components from various sources
- Verification
  - Assure that the product and components comply with specifications
- Validation
  - Assure that the product fulfils customers actual needs

PA CMMI3 (continued)

Kristian Sandal
krisa@ida.liu.se

- Organizational Process Focus
  - Understand the organizations process and need for process
  - Strengths and weaknesses
  - Improve the process where necessary
- Organizational Process Definition
  - Develop and maintain assets: Documents and tools for processes
- Organizational Training
  - Define needed competence
  - Make competence inventory
  - Make a gap analysis
  - Schedule training

PA CMMI3 (continued)

- Integrated Project Management
  - Assure that all levels share: vision, project goals, planning and progress control process
  - Stakeholder involvement
- Risk Management – see lecture 3
- Decision Analysis and Resolution
  - Evaluate project alternatives according to criteria
  - Structured decisions selecting project implementation alternatives
- Integrated Teaming (old, but good)
  - Form teams with relevant members
  - Govern team operation and external communication
- Organizational Environment for Integration (old, but good)
  - Approach and infrastructure for team collaboration

Life at level 3

- More detailed descriptions
- Tailoring processes from your **own** definitions
- Baseline: Describe your current performance
- Opens for development (and creativity) of alternatives
- Works for a range of projects
- Originally the minimum level

Life at level 4

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

Life of level 5

- Everyone is committed to the continuous improvement of processes.
- Innovation climate paired with an ability to evaluate new technology
- Empowered co-workers
- Low variation in processes
- Reacts quickly to change
- Challenge: Company culture, new markets
- Used by many sub-contractors in marketing

Part III
Quality Management

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 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!)

Total Quality Management

- What's get measured gets done
- Importance of feed-back
- Non-personal software
- Creating a passion for quality
- Live as you learn
- Incentive system
- Involve customers
- Set prioritized goals
- Quality is everybody's responsibility
- Document how you will work with quality
- Improve continuously

= Some Guidelines to TQM

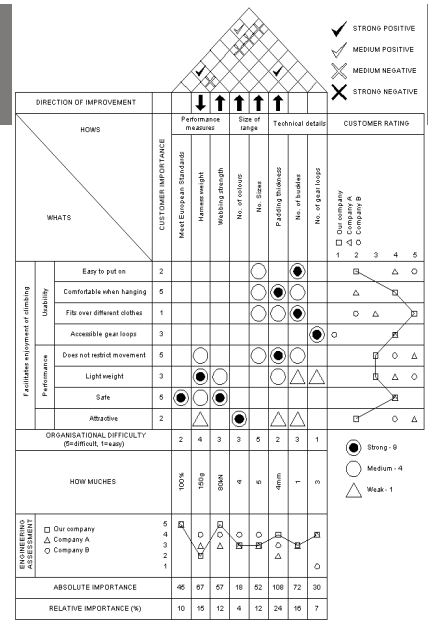
Short intro: http://managementhelp.org/quality/tqm/tqm.htm

TickIT

- An interpretation of ISO 9001 for software,
- a standard set of requirements on the competence and behavior of certification auditors,
- a standardized training course for certification auditors,
- a registration scheme for approved certification auditors,
- a system for accrediting certification bodies for conducting TickIT certifications,
- a logotype to be used on certificates to show TickIT certification.

http://www.tickit.org/

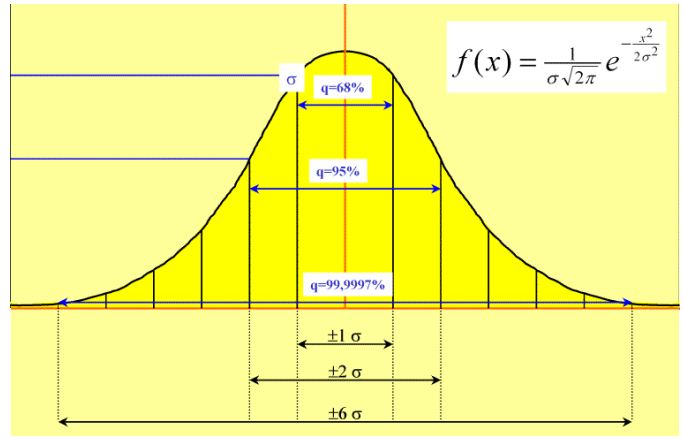
Quality Function Deployment



Six sigma

Goal: Reduce waste
Origin and application in production industry.
Applicability in software is debated.

My view:
Requires repetition
Requires sampling



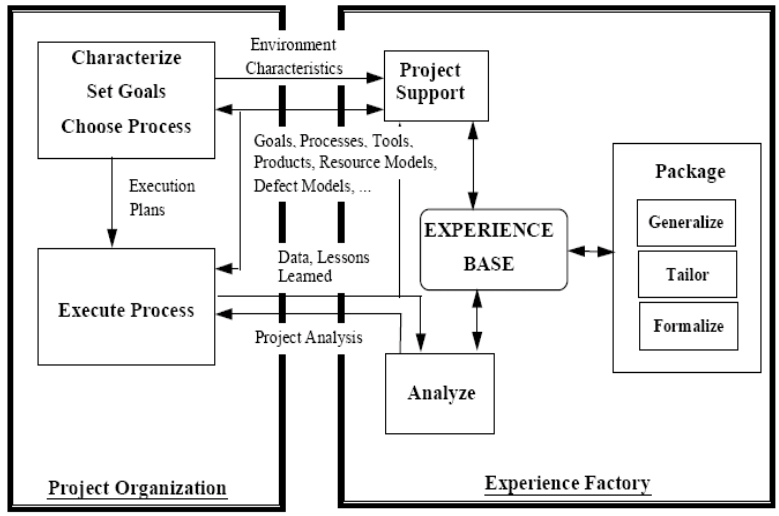
http://www.itil-itsm-world.com/sigma.htm

QIP

1. Set quantifiable goals
2. Select processes
3. Run processes
4. Measure objectives
5. Analyse measurements
6. Package experience

Experience factory

Kristian Sandal
krisa@ida.liu.se



Wisdom

Kristian Sandal
krisa@ida.liu.se