

CITS5502 Software Processes

The European and ISO Initiatives - SPICE

Unit coordinator: Arran Stewart

Key concepts, cont'd

- Attributes of SPICE
 - Internal process improvement & external capability determination
 - Six level model (0 to 5)
 - Independent of any software methodologies
 - Defines the properties of a Process Assessment Tool
 - Defines standard for assessors (auditors)

ISO and ISO standards

- ISO = International Organization for Standardization
- ISO develop and publish International Standards
- Some popular standards:
 - ISO 9000 Quality management
 - ISO 14000 Environmental management
 - ISO 50001 Energy management
 - ISO 31000 Risk management
 - ISO 22000 Food safety management
 - ISO 27001 Information security management
 - ISO 45001 Occupational health and safety
 - ISO 13485 Medical devices

ISO 9000

- first published in 1987
- The ISO 9000 series are based on 8 quality management principles:
 - ① Customer focus
 - ② Leadership
 - ③ Involvement of people
 - ④ Process approach
 - ⑤ System approach to management
 - ⑥ Continual improvement
 - ⑦ Factual approach to decision making
 - ⑧ Mutually beneficial supplier relationship

ISO 9001

- Over time, standards get updated
- For example, ISO 9001 (sets out the criteria for a quality management system):
 - Developed in 1987
 - Revised in 1993
 - Revised again in 2000
 - Revised again in 2008
 - Revised yet again in September 2015. Referred to as ISO 9001:2015

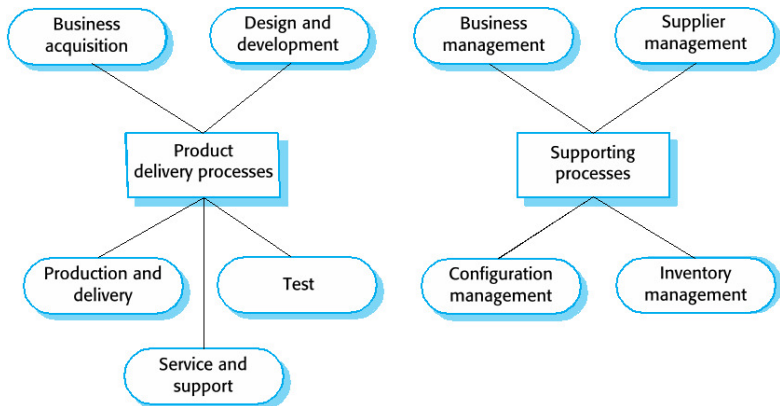
ISO 9001



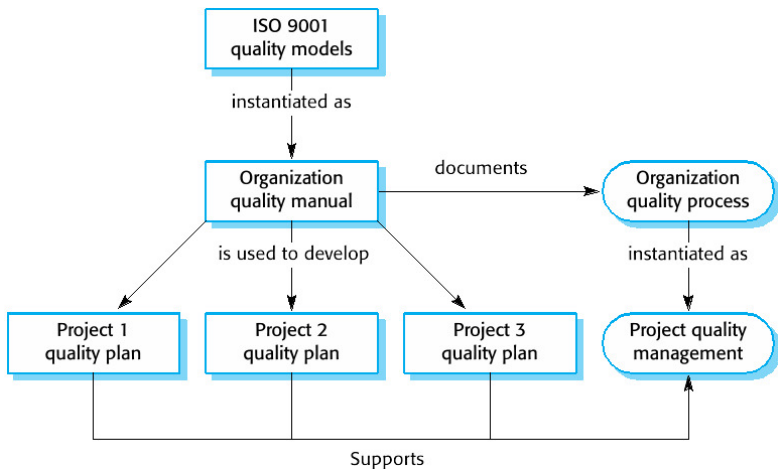
ISO 9001 standards framework

- ISO 9001 – most general of the 9000 series standards
 - applies to organizations that design, develop and maintain products, including software.
- ISO 9001 standard is a *framework for developing* standards.
 - It sets out general quality principles, describes quality processes in general and lays out the organizational standards and procedures that should be defined. These should be documented in an organizational quality manual.

ISO 9001 core processes



ISO 9001 and quality management



ISO 9001 certification

- Quality standards and procedures should be documented in an organisational quality manual.
- An external body may certify that an organisation's quality manual conforms to ISO 9000 standards.
- Some customers require suppliers to be ISO 9000 certified although the need for flexibility here is increasingly recognised.

Problems with ISO 9001

- The ISO 9001 defines quality simply to be the conformance to standards.
- It takes no account of quality as experienced by users of the software. For example, a company could define test coverage standards specifying that all methods in objects must be called at least once.
- Unfortunately, this standard can be met by incomplete software testing that does not include tests with different method parameters. So long as the defined testing procedures are followed and test records maintained, the company could be ISO 9001 certified.

ISO/IEC 15504 / SPICE

- ISO/IEC 15504 Information Technology – Process Assessment is another term for Software Process Improvement and Capability dEtermination (SPICE)
- SPICE is a framework (consisting of a set of technical standards documents) for the computer software development process and related business management functions
- SPICE is one of the joint ISO and IEC (International Electrotechnical Commission) standards, developed by the ISO/IEC joint technical committee (JTC)
- As of March 2015, ISO/IEC 15504 has been revised by ISO/IEC 33001:2015 Information technology – Process assessment – Concepts and terminology

Capability Levels in SPICE

- 1 Incomplete
- 2 Performed
- 3 Managed
- 4 Established
- 5 Predictable
- 6 Optimizing

Levels

0 – Incomplete

No Common Features.

There is general failure to perform the base practices in the Process. There are no easily identifiable work products, artefacts, or outputs of the process.

1 – Performed

Performed Informally.

Base practices of the Process are performed. The performance of these base practices may not be rigorously planned and tracked. Performance depends on individual knowledge and effort. Work products of the Process testify to the performance. Individuals within the organization recognize that an action should be performed, and there is general assurance that this action is performed as and when required. ☰ ↺ ↻ ↻

Capability Levels in SPICE, cont'd

2 – Managed

Planned and Tracked.

The project (and/or organization) plan, track, and verify the performance of the base practices in the Process. As appropriate, work products resulting from performing these base practices are reviewed for adequacy, corrective action is taken, and the work products are placed under version control. The primary distinction from the Performed Level is that the performance of the base practices is planned and managed.

Capability Levels in SPICE, cont'd

3 – Established

Defined, Tailored, and Using Data Retained.

Base practices are performed across a process and/or throughout the organization using approved, tailored versions of standard, documented processes.

Information is gathered and maintained relating to the use of the defined process and customer satisfaction with the results. The primary distinction from the Managed Level is that the base practices of the Performed Level are planned and managed using an organization-wide standard process.

Capability Levels in SPICE, cont'd

4 – Predictable

Measured, Analyzed, and Quantitatively Managed. Measurable process goals are established for each defined process. Detailed measures of performance are collected and analyzed. This leads to a quantitative understanding of the process and an improved ability to predict performance. The primary distinction from the Established Level is that the defined processes are quantitatively understood and controlled.

5 – Optimizing

Optimizing and Innovating. Quantitative process effectiveness and efficiency goals (targets) for performance are established, based on the business goals of the organization. The primary distinction from the Predictable Level is that the defined process and the standard process undergo

Capability levels in SPICE

Optimising

Quantitative measures used for continual improvement

Level 5 Optimizing

PA.5.1 Process Innovation
PA.5.2 Process Optimization

Predictable

Metrics make process performance and results controllable, trends

Level 4 Predictable

PA.4.1 Process Measurement
PA.4.2 Process Control

Established

A defined processes is tailored from a standard process and deployed

Level 3 Established

PA.3.1 Process Definition
PA.3.2 Process Deployment

Level 2 Managed

PA.2.1 Performance Management
PA.2.2 Work Product Management

Managed

Process and work products are managed, responsibilities identified.

Level 1 Performed

PA.1.1 Process Performance

Performed

processes are intuitively performed, input and output work products are available

Level 0 Incomplete

Incomplete

Performance and results are incomplete, chaotic processes

Han van Leen, Leistungs Consult

Process Attributes/Indicators used in SPICE

The capability of processes is measured using process attributes.
The international standard defines 9 process attributes:

Maturity Level		Process Attributes / Indicators
5	"Optimising"	PA 5.1 Process Innovation PA 5.2 Continuous Optimisation
4	"Predictable"	PA 4.1 Process Measurement PA 4.2 Process Control
3	"Established"	PA 3.1 Process Definition PA 3.2 Process Deployment
2	"Managed"	PA 2.1 Performance Management PA 2.2 Work Product Management
1	"Performed"	PA 1.1 Process Performance
0	"Incomplete"	

Rating Scale for the Process Attributes

- Each process attribute is assessed on a four-point (N-P-LF) rating scale:
 - ① Not achieved (0 - 15%)
 - ② Partially achieved (>15% - 50%)
 - ③ Largely achieved (>50%- 85%)
 - ④ Fully achieved (>85% - 100%).
- The rating is based upon evidence collected against the practice indicators

Assessment Tools

- Paper-based tools
- A limited number of computer based tools are also available
 - Tools to visualize software assessment data
 - Appraisal Assistant – a software application developed by the SEI¹, Griffith University that supports the assessment of process capability

¹SEI = Software Engineering Institute

Assessors

Must have a suitable level of the relevant skills and experience, including:

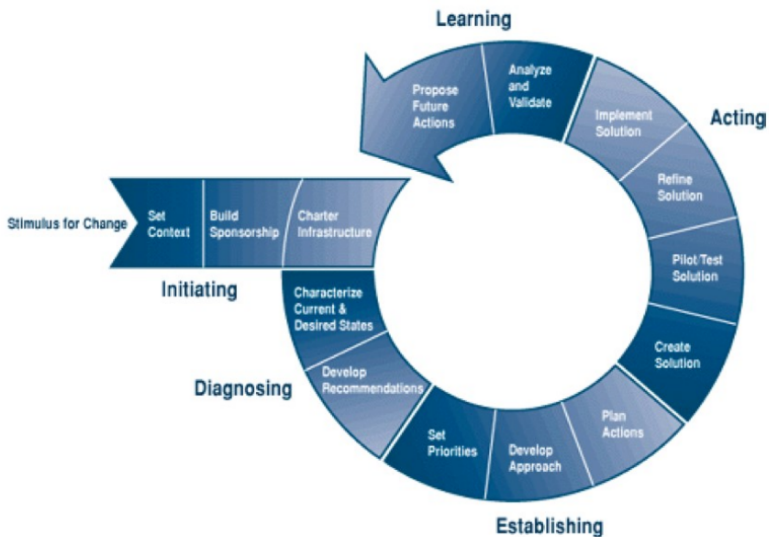
- relevant education and training and experience
- specific skills for particular categories, e.g., management skills for the management category
- ISO/IEC 15504 related training and experience in process capability assessments

IDEAL

- Apart from SPICE, there are other models available. . .
- IDEAL (Initiating, Diagnosing, Establishing, Acting & Learning) is an organizational improvement model proposed and developed by the SEI at CMU.² The model is named for the 5 phases that it describes: initiating, diagnosing, establishing, acting and learning
- It forms an infrastructure to guide organizations in planning and implementing an effective software process improvement program.

²Reference: <http://www.sei.cmu.edu/library/assets/idealmodel.pdf> 

IDEAL, cont'd



ISO and SPICE – useful references

- ISO web page: <http://www.iso.org/iso/home>
- SPICE at SEI, Griffith University: <https://www.griffith.edu.au/engineering-informationtechnology/software-quality-institute/research/software-processimprovement-and-capability-determination>
- Zahran, S. (1998) Software Process Improvement – Practical Guides for Business Success (Addison-Wesley)
- SPICE conference proceedings – <http://www.spiceconference.com>

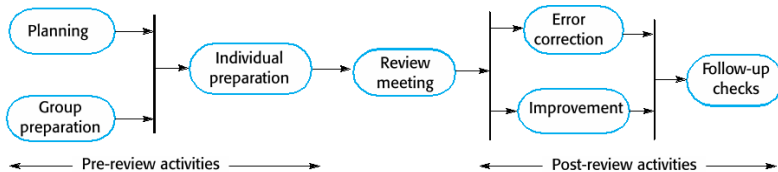
Software reviews

- A catch all term for *filters* that can be applied at various points during the software development process

Static Testing Methods

- Code Inspections
- Code Walk-throughs
- Semantic Analysis of Code
- Fagan Inspections
- Analysis of Code Metrics

Processes



Code Inspection Methods

- Inspections are a formal, efficient and economical method of finding faults in design and code [Fagan,76]
- Code inspection amounts to “executing the code in your head” or on paper
- Code inspections are very effective at finding faults³
 - detect 67% of faults [Fagan 76]
 - detect 85% of faults [Jones 77]
 - detect 93% of errors [Ackerman et al 86]

³Statistics from Pfleeger p. 291.

Code Walk-through

- A Code Walk-through is an informal meeting where the programmer leads the review team through his/her code and the reviewers try to identify faults
- You can do this with your own code but it is more effective if the reviewer is not the same person as the programmer. See egoless programming in many SE texts
- Example: perform a code walk through of the `getNumofDaysinMonth` method

getNumOfDays

```
if (year<1) {
    throw new YearOutOfBounds(year);
}

if (month==1 || month==3 || month==5 || month==7 || month==10
    || month==12) {
    numDays = 32;
} else if (month==4 || month==6 || month==9 || month==11) {
    numDays = 30;
} else if (month==2) {
    if (isLeapYear(year)) { numDays = 29;
    }
    else { numDays = 28;
    }
} else {
    throw new MonthOutOfBounds(month);
}
return numDays;
```

Fagan Inspection

- is a more formal version of a code walk-through
- Procedure:
 - 1 Overview
 - 2 Preparation
 - 3 Inspection
 - 4 Rework
 - 5 Follow up
- meetings are chaired by a team moderator rather than the programmer

Checklists

- A set of questions to stimulate critical appraisal of all aspects of the system
- Questions are usually general in nature and thus applicable to many types of system
- Checklist examples will go up on website

Semantic Analysis

Analysis based on a model of the meaning (i.e. semantics) of a program

- Formal Proofs: prove a given program (model) satisfies a required property
- Control Flow Analysis: analysis of the directed graph of the control structure of a program to identify inaccessible code, infinite loops and poor structure
- Data Flow Analysis: analysis of a diagrammatic representation of the flow of data throughout a program
- Symbolic Execution: check for agreement between code and specification using algebraic vars in place of input data. Assignment statements produce algebraic output which can be compared with expected results.

Code Metrics

- Measures of properties of code which may predict how likely the code is to contain errors
- For example,
 - graph theoretic complexity (of the program's control graph)
 - module accessibility (how many ways a module may be accessed)
 - number of entry and exit points per module
 - for some Object Oriented Metrics see <http://yunus.hun.edu.tr/~sencer/oom.html>