# LECTURE 6

## Quality management

## Content

1 Introduction to quality management process

2 Roles of standards in quality management

3 Relationship between quality attributes and software metrics

4 Using measurement in assessing software quality

## What is quality management?

Managing the quality of the software process and products

## Software quality management

- Concerned with ensuring that the required level of quality is achieved in a software product
- Involves defining appropriate quality standards and procedures and ensuring that these are followed
- Should aim to develop a ‘quality culture’ where quality is seen as everyone’s responsibility

## What is quality?

- Quality, simplistically, means that a product should meet its specification
- This is problematical for software systems
    - Tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.)
    - Some quality requirements are difficult to specify in an unambiguous way
    - Software specifications are usually incomplete and often inconsistent

## Software quality attributes

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

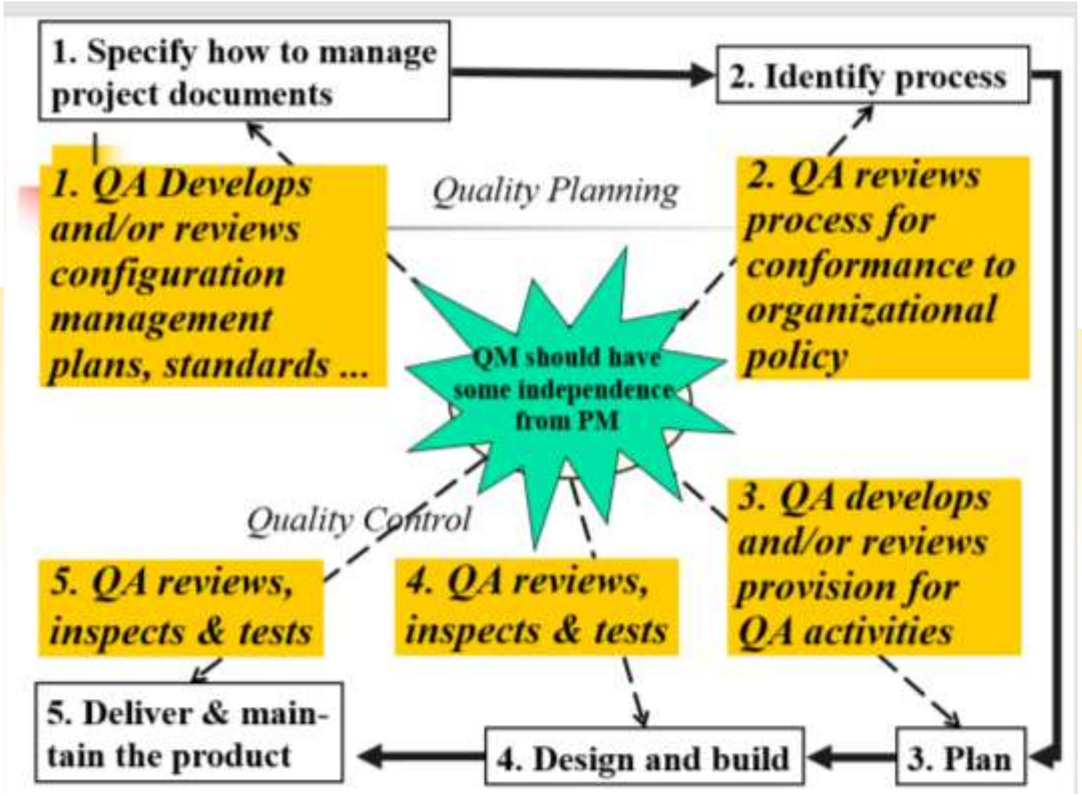
## A high quality software product

- Satisfies clearly stated requirements
- Checks its inputs and that it reacts in predictable ways to illegal inputs
- Has been inspected thoroughly by others
- Has been tested exhaustively by others
- Is thoroughly documented
- Has a known defect rate

## The quality compromise

- We cannot wait for specifications to improve before paying attention to quality management
- Must put procedures into place to improve quality in spite of imperfect specification
- Quality management is therefore not just concerned with reducing defects but also with other product qualities

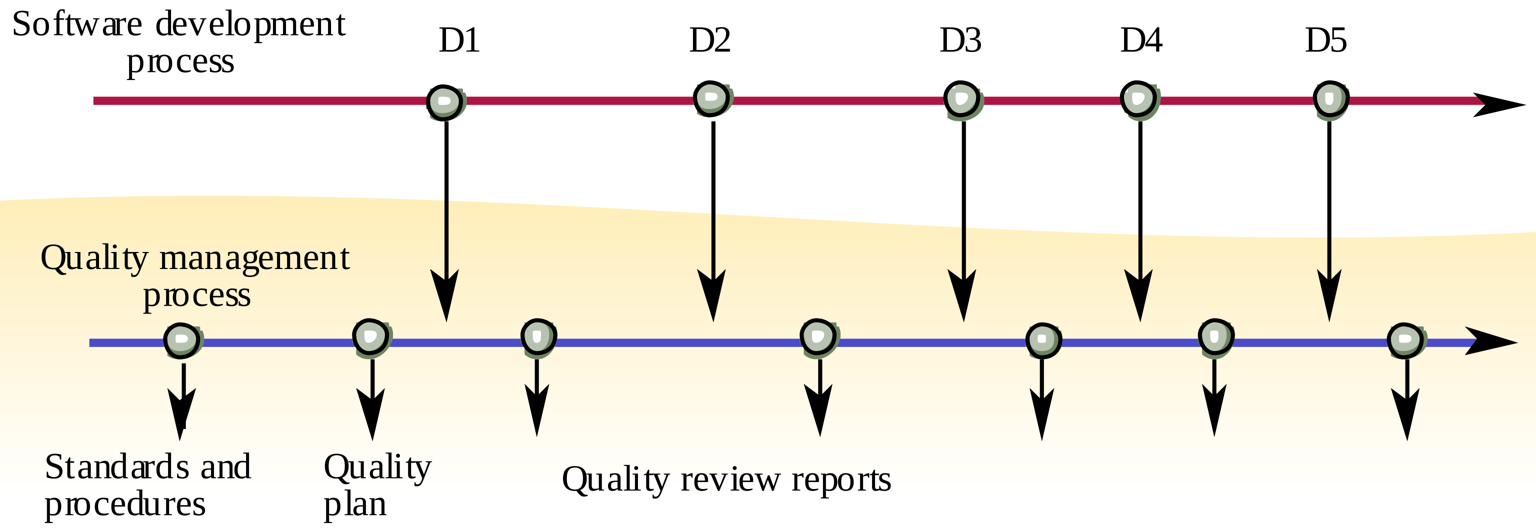
## Quality planning


1. Specify how to manage project documents
2. Identify process
Quality Planning
1. QA Develops and/or reviews configuration management plans, standards ...
2. QA reviews process for conformance to organizational policy
QM should have some independence from PM
Quality Control
3. QA develops and/or reviews provision for QA activities
5. QA reviews, inspects & tests
4. QA reviews, inspects & tests
5. Deliver & maintain the product
4. Design and build
3. Plan

## Quality management activities

- Quality assurance
  - Establish organizational procedures and standards for quality
- Quality planning
  - Select applicable procedures and standards for a particular project and modify these as required
- Quality control
  - Ensure that procedures and standards are followed by the software development team
- Quality management should be separated from project management to ensure independence

## Quality management and software development



## Quality assurance and standards

- Standards are the key to effective quality management
- They may be international, national, organizational or project standards
- Product standards define characteristics that all components should exhibit e.g. a common programming style
- Process standards define how the software process should be enacted

## Importance of standards

- Encapsulation of best practice- avoids repetition of past mistakes
- Framework for quality assurance process – it involves checking standard compliance
- Provide continuity - new staff can understand the organization by understand the standards applied

## Product and process standards

| Product standards | Process standards |
|---|---|
| Design review form | Design review conduct |
| Document naming standards | Submission of documents to CM |
| Procedure header format | Version release process |
| Ada programming style standard | Project plan approval process |
| Project plan format | Change control process |
| Change request form | Test recording process |

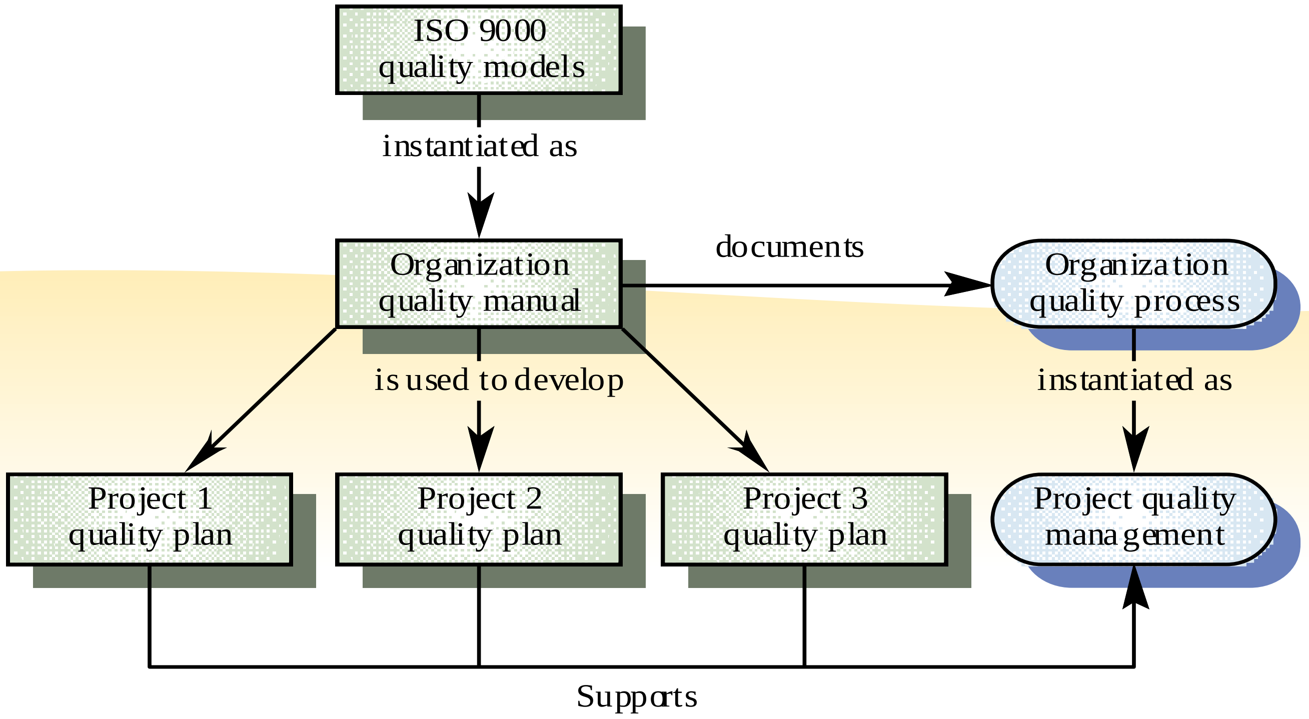
## ISO 9000 certification

- International set of standards for quality management
- Applicable to a range of organizations from manufacturing to service industries
- ISO 9001 applicable to organizations which design, develop and maintain products
- ISO 9001 is a generic model of the quality process that must be instantiated for each organization

## ISO 9000 certification

- Quality standards and procedures should be documented in an organizational quality manual
- External body may certify that an organization's quality manual conforms to ISO 9000 standards
- Customers are, increasingly, demanding that suppliers are ISO 9000 certified

# ISO 9000 and quality management



## Problems with standards

- Not seen as relevant and up-to-date by software engineers
- Involve too much bureaucratic form filling
- Unsupported by software tools so tedious manual work is involved to maintain standards

## Overcoming the problems

- Involve practitioners in development. Engineers should understand the rationale underlying a standard
- Review standards and their usage regularly. Standards can quickly become outdated and this reduces their credibility amongst practitioners
- Detailed standards should have associated tool support. Excessive clerical work is the most significant complaint against standards

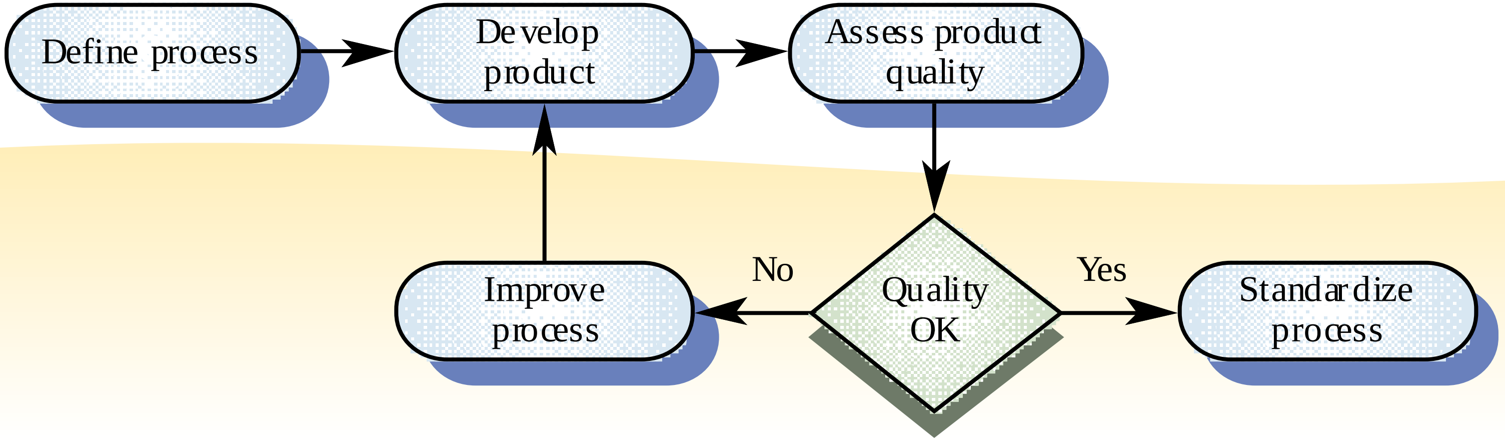
## Process and product quality

- The quality of a developed product is influenced by the quality of the production process
- Form (product) follows function (process)
- Particularly important in software development as some product quality attributes are hard to assess
- However, there is a very complex and poorly understood relationship between software processes and product quality

## Process-based quality

- Straightforward link between process and product in manufactured goods
- More complex for software because:
  - The application of individual skills and experience is particularly important in software development
  - External factors such as the novelty of an application or the need for an accelerated development schedule may impair product quality
- Care must be taken not to impose inappropriate process standards

## Process-based quality



## Practical process quality

- Define process standards such as how reviews should be conducted, configuration management, etc.
- Monitor the development process to ensure that standards are being followed
- Report on the process to project management and software procurer

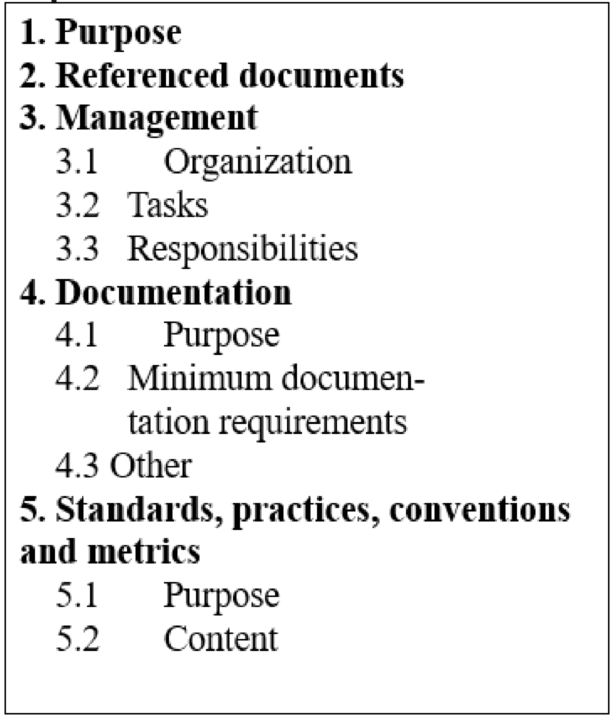
## Quality planning

- A quality plan sets out the desired product qualities and how these are assessed and define the most significant quality attributes
- It should define the quality assessment process
- It should set out which organizational standards should be applied and, if necessary, define new standards

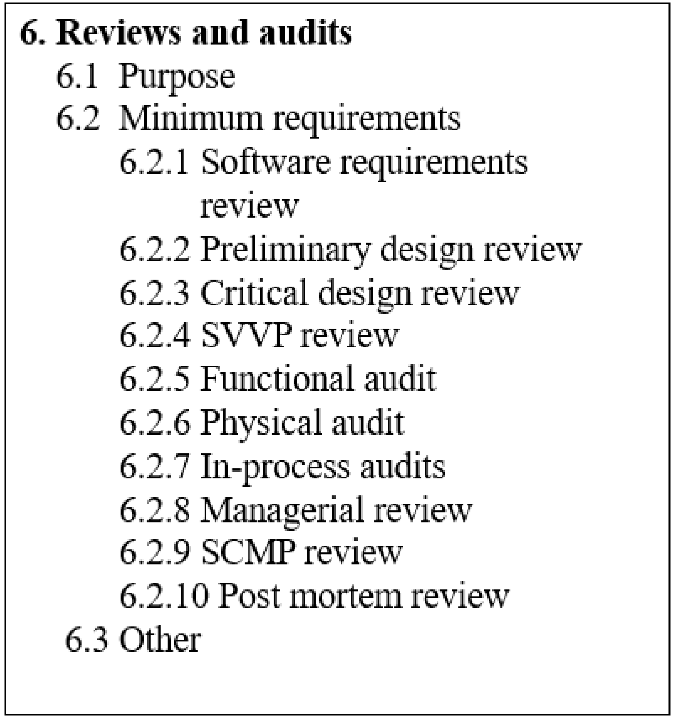
## Quality plan structure

- Product introduction
- Product plans
- Process descriptions
- Quality goals
- Risks and risk management
- Quality plans should be short, succinct documents
    - If they are too long, no-one will read them

## IEEE 730-1989 software quality assurance plans – Tables of contents

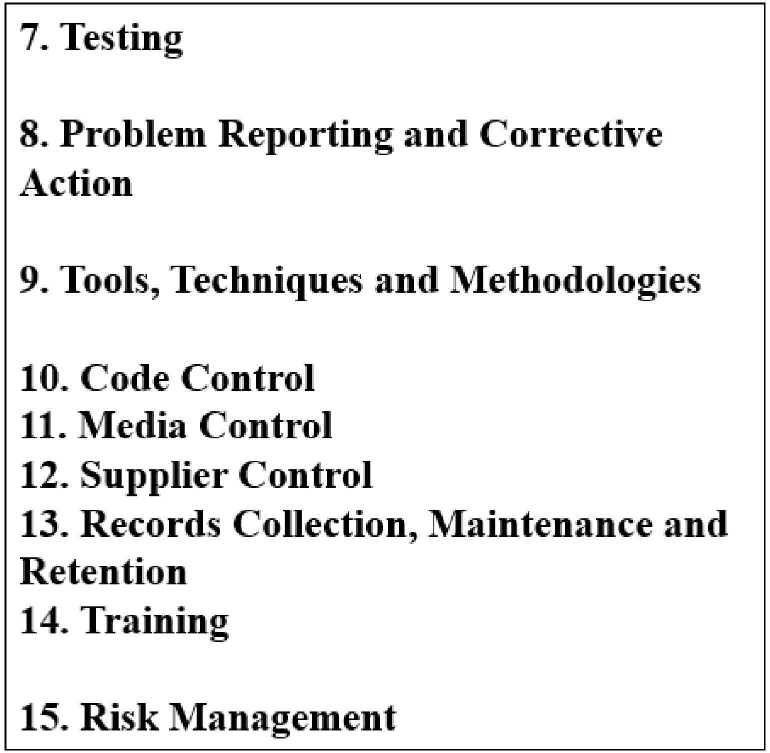


**1. Purpose**
**2. Referenced documents**
**3. Management**
3.1 Organization
3.2 Tasks
3.3 Responsibilities
**4. Documentation**
4.1 Purpose
4.2 Minimum documentation requirements
4.3 Other
**5. Standards, practices, conventions and metrics**
5.1 Purpose
5.2 Content



**6. Reviews and audits**
6.1 Purpose
6.2 Minimum requirements
6.2.1 Software requirements review
6.2.2 Preliminary design review
6.2.3 Critical design review
6.2.4 SVVP review
6.2.5 Functional audit
6.2.6 Physical audit
6.2.7 In-process audits
6.2.8 Managerial review
6.2.9 SCMP review
6.2.10 Post mortem review
6.3 Other

## IEEE 730-1989 software quality assurance plans – Tables of contents



7. Testing

8. Problem Reporting and Corrective Action

9. Tools, Techniques and Methodologies

10. Code Control
11. Media Control
12. Supplier Control
13. Records Collection, Maintenance and Retention
14. Training

15. Risk Management

## Quality control

- Checking the software development process to ensure that procedures and standards are being followed
- Two approaches to quality control
  - Quality reviews
  - Assessment via software metrics

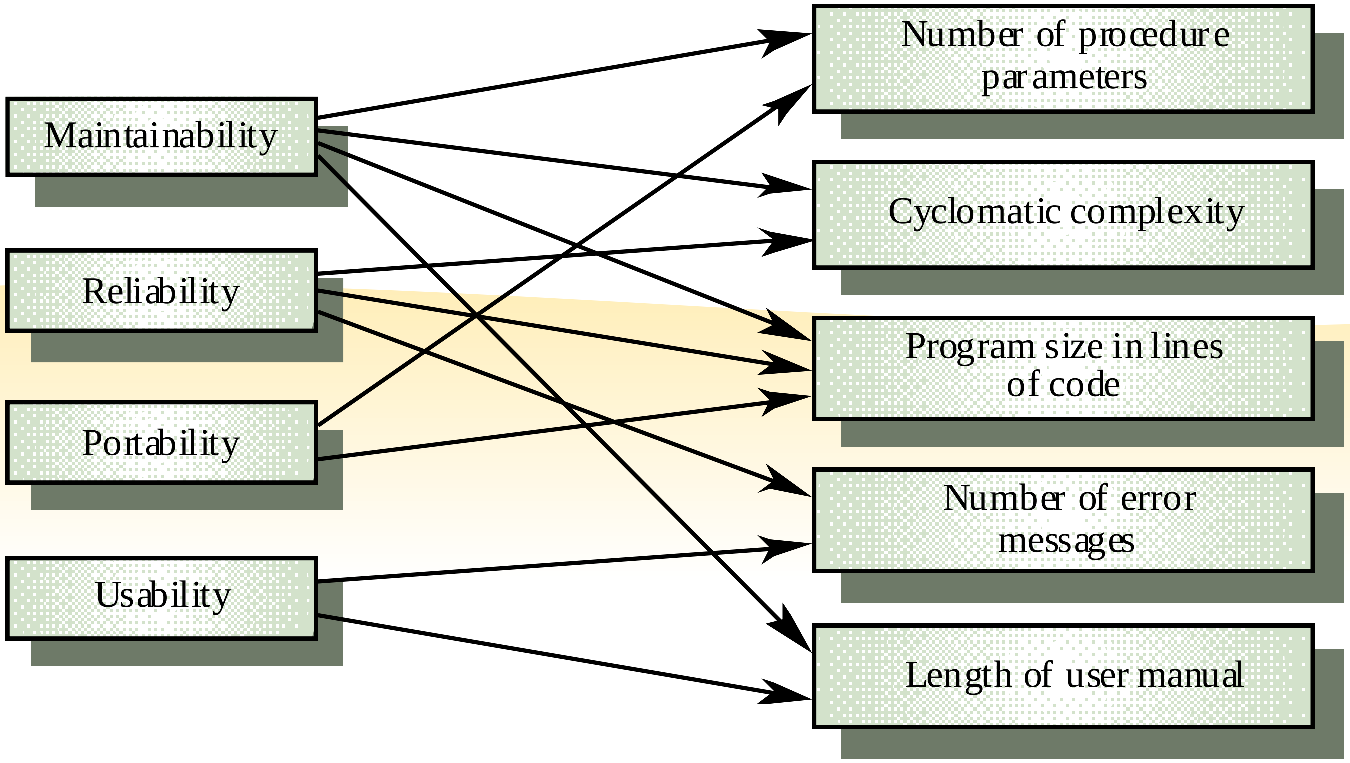
## Quality review

- The principal method of validating the quality of a process or of a product
- Group examines part or all of a process or system and its documentation to find potential problems
- There are different types of review with different objectives
  - Inspections for defect removal (product)
  - Reviews for progress assessment (product and process)
  - Quality reviews (product and standards)

## Quality attributes and software metrics

- Software measurement is concerned with deriving a numeric value for an attribute of a software product or process
- Software metric is any type of measurement which relates to a software system, process or related documentation
- This allows for objective comparisons between techniques and processes
- There are few standards, no systematic use

# Quality attributes and software metrics



## Important software metric assumptions

- A software property can be measured
- A relationship exists between what we can measure and a quality attribute
- This relationship has been formalized and validated

## The measurement process

- A software measurement process may be part of a quality control process
- Data collected during this process should be maintained as an organizational resource
- Once a measurement database has been established, comparisons across projects become possible

## Product metrics

- A quality metric should be a predictor of product quality
- Classes of product metric:
  - Dynamic metrics which are collected by measurements made of a program in execution
  - Static metrics which are collected by measurements made of the system representations
  - Dynamic metrics help assess efficiency and reliability; static metrics help assess complexity, understandability and maintainability

## Dynamic and static metrics

- Dynamic metrics are closely related to software quality attributes
  - Collected by a program in execution (response time, number of failures)
  - Help assess efficiency, effectiveness, availability and reliability
- Static metrics have an indirect relationship with quality attributes
  - Collected from system representation (lines of code)
  - Help assess complexity, understandability and maintainability

## Software product metrics

| Software metric | Description |
|---|---|
| Fan in/Fan-out | Fan-in is a measure of the number of functions that call some other function (say X). Fan-out is the number of functions which are called by function X. A high value for fan-in means that X is tightly coupled to the rest of the design and changes to X will have extensive knock-on effects. A high value for fan-out suggests that the overall complexity of X may be high because of the complexity of the control logic needed to coordinate the called components. |
| Length of code | This is a measure of the size of a program. Generally, the larger the size of the code of a program's components, the more complex and error-prone that component is likely to be. |
| Cyclomatic complexity | This is a measure of the control complexity of a program. This control complexity may be related to program understandability. The computation of cyclomatic complexity is covered in Chapter 20. |
| Length of identifiers | This is a measure of the average length of distinct identifiers in a program. The longer the identifiers, the more likely they are to be meaningful and hence the more understandable the program. |
| Depth of conditional nesting | This is a measure of the depth of nesting of if-statements in a program. Deeply nested if statements are hard to understand and are potentially error-prone. |
| Fog index | This is a measure of the average length of words and sentences in documents. The higher the value for the Fog index, the more difficult the document may be to understand. |

# Object-oriented metrics

| **Object-oriented metric** | **Description** |
|---|---|
| Depth of inheritance tree | This represents the number of discrete levels in the inheritance tree where sub-classes inherit attributes and operations (methods) from super-classes. The deeper the inheritance tree, the more complex the design as, potentially, many different object classes have to be understood to understand the object classes at the leaves of the tree. |
| Method fan-in/fan-out | This is directly related to fan-in and fan-out as described above and means essentially the same thing. However, it may be appropriate to make a distinction between calls from other methods within the object and calls from external methods. |
| Weighted methods per class | This is the number of methods included in a class weighted by the complexity of each method. Therefore, a simple method may have a complexity of 1 and a large and complex method a much higher value. The larger the value for this metric, the more complex the object class. Complex objects are more likely to be more difficult to understand. They may not be logically cohesive so cannot be reused effectively as super-classes in an inheritance tree. |
| Number of overriding operations | These are the number of operations in a super-class which are over-ridden in a sub-class. A high value for this metric indicates that the super-class used may not be an appropriate parent for the sub-class. |

THANK YOU !