

# The BIG QUESTION

What “Software Quality” is all  
about

# So what is quality?

Maybe a matter of perception???

# What is Software Quality?

- Quality, simplistically, means that a product should meet its specification
- This is problematic 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

# The Questions for us

How do we create a  
**Quality Software Product**

Quality Software Development Process  
is part of that

# Process and Product quality

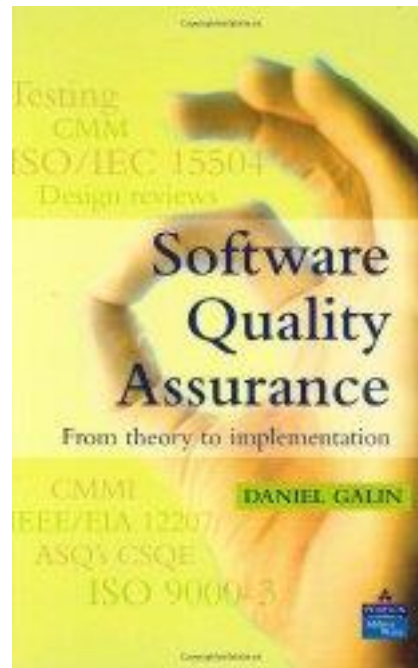
- The quality of a developed **product** is influenced by the quality of the production **process**.
- This is important in software development as some product quality attributes are hard to assess.
- However, **there is a very complex and poorly understood relationship between software processes and product quality.**

# Quality 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.

# Introduction to Software Quality

- Ref to D. Galin Ch1 – 5 ++



# Some Causes of Software Errors

- Faulty requirements definition
- Client-developer communication failures
- Deliberate deviations from software requirements
- Logical design errors
- Coding errors
- Non-compliance with documentation and coding instructions
- Shortcomings of the testing process
- User interface and procedure errors
- Documentation errors
- Many more...



# What is Software Quality ?



**Quality**

Degree of excellence or worth  
the standard of something  
superiority, high grade, or  
essential characteristic of

- **Conformance to requirements**
- Lack of bugs
- Low defect rate (# of defects/size unit)
- High reliability (number of failures per n hours of operation)
  - Measured as Mean Time To Failure (MTTF)  
probability of failure-free operation in a specified time

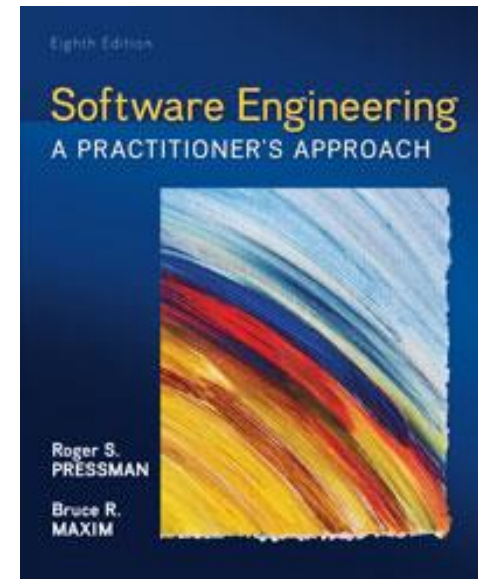
# What is Software Quality ?

- According to the IEEE Software quality is:
  - The degree to which a system, component, or process meets specified requirements.
  - The degree to which a system, component, or process meets customer or user needs or expectations.



# What is Software Quality ?

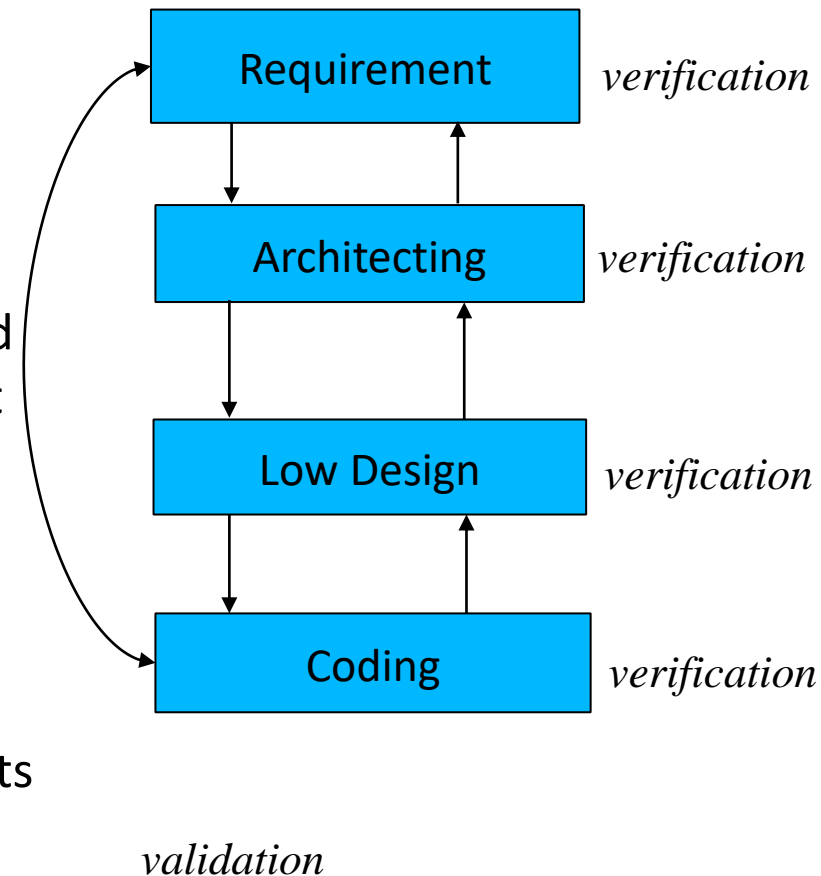
- According to Roger Pressman



Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software

# Software Quality – 2 Core Terms

- Software Quality includes
- **Verification**
  - are we building the product right ?
  - performed at the end of a phase to ensure that requirements established during previous phase have been met
- **Validation**
  - are we building the right product ?
  - performed at the end of the development process to ensure compliance with product requirements



# Software Quality in the SDLC

- **Software Quality activities should be integrated into the software development plan**
- The software development plan will implement one or more SDLC' s
- The intensity and number of quality assurance activities is affected by project and team factors...
- SDLC phases
  - Requirements
  - Specification (Analysis)
  - Design
  - Implementation
  - Integration
  - Maintenance
  - Retirement
- In waterfalls, spirals, iterations, etc.

# Factors affecting the required intensity of Software Quality activities

- **Project factors:**

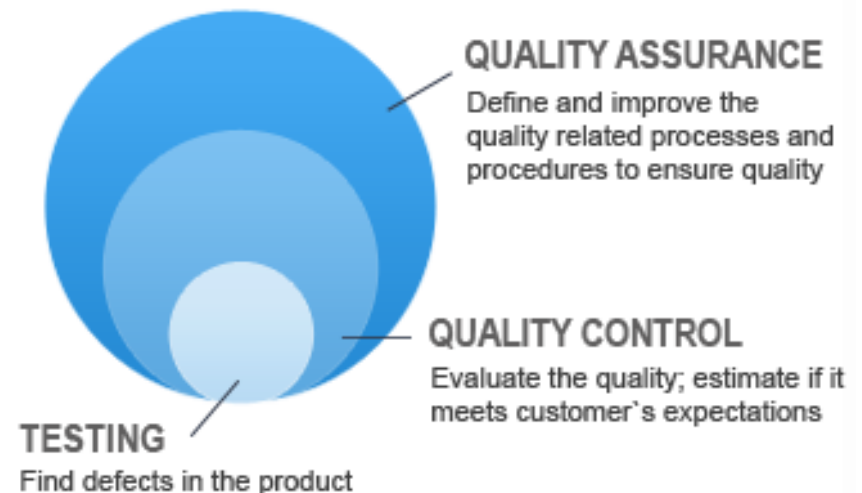
- Project's magnitude
- Project's technical complexity and difficulty
- Extent of reusable software components / OSS
- Severity of failure outcomes if the project fails

- **Team factors:**

- The professional qualification of the team members
- Team acquaintance with the project and its experience in the area
- Availability of staff members that can professionally support the team
- Familiarity with the team members, in other words, the percentage of new staff members in the team

# Software Quality Assurance?

- According to the IEEE Software quality assurance is:
  1. A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
  2. A set of activities designed to evaluate the process by which the products are developed or manufactured. Contrast with: quality control.



# SQA and Risk Management

- Risk management is an integral part of the process to develop software quality
  - Quality is not free...QA activities costs time and money...trade-offs are necessary
  - Quality assurance activities are risk reduction efforts
  - A “one-size-fits-all” quality assurance plan is rarely feasible
    - QA process tailoring is necessary
    - Risk management is an approach to intelligent process tailoring
- Risk Management
- Risk Identification
  - Risk Analysis and Assessment
  - Risk Planning & Mitigation
  - Risk Tracking



# SQA and Risk Management

- **SEI Software Risk Taxonomy**

## Product Engineering

- Requirements
- Design
- Code and Unit Test
- Integration and Test
- Engineering Specialties

## Development Environment

- Development Process
- Development System
- Management Process
- Management Methods
- Work Environment

## Program Constraints

- Resources
- Contract
- Project Interfaces

- Use risk taxonomy as a checklist to identify risks and potential risk mitigation activities (i.e., QA process and products)

# Software Quality Control

- Process Enforcement
  - Independent agent (i.e., Software Quality Assurance manager/team)
  - Quantitative Metrics
  - Data Collection / Archiving
  - Review / Audit (process and products)
  - Authority to act
- Process Assessment
- Process Improvement Feedback
- Has QC changed in agile software development?

# Software Metrics

- Necessary, under-used and hard
  - Forms one of the pillars of “prove-able” quality
  - Software and its development is complex and multi-dimensional, hard to understand and measure
- Scope of software metrics – process, products or resources
  - Project management
  - Cost and level-of-effort estimation
  - Productivity measures
  - Quality attributes
  - Reliability measures
  - Performance
  - Defect tracking
  - Structural and complexity metrics
  - Software engineering process metrics

# SQ Goals

## The Bottom Line...

- Defect prevention
  - prevents defects from occurring in the first place
  - Activities: training, planning, and simulation
- Defect detection
  - finds defects in a software artifact
  - Activities: inspections, testing or measuring
- Defect removal
  - isolation, correction, verification of fixes
  - Activities: fault isolation, fault analysis, regression testing

# Many challenges exist in S/W development...

*Q. What is the right level of quality?*

*A. At any point in time there is an optimum QA/ROI proposition – but this is very difficult to achieve because our business is complex. This difficulty is amplified because change is unpredictable and (increasingly?) unavoidable.*

*Q. Manage a surplus or a deficit?*

*A. There are two types of struggle: (i) with deficient quality levels (which is soul destroying), (ii) struggle to justify SQA costs. Up to you which struggle to have – but it seems easier to manage a surplus!*

*The pursuit of appropriate levels of quality is effectively a personal philosophy. However, sub-standard work habits (be they of a technical or of an interactive nature) are eventually found out.*

# Models of Software Quality Factors

- There are many models of software quality factors
  - **ISO/IEC 9126 (now ISO/IEC 25010)**
    - Its really a framework for the evaluation of software quality
  - **McCall's software quality factors model**
    - Dated from 1977 and is still relevant today
  - Boehm's Quality Model – 1978
  - Evans and Marciniak factor model (1987)
  - Deutsch and Willis (1988)

# ISO/IEC 25010

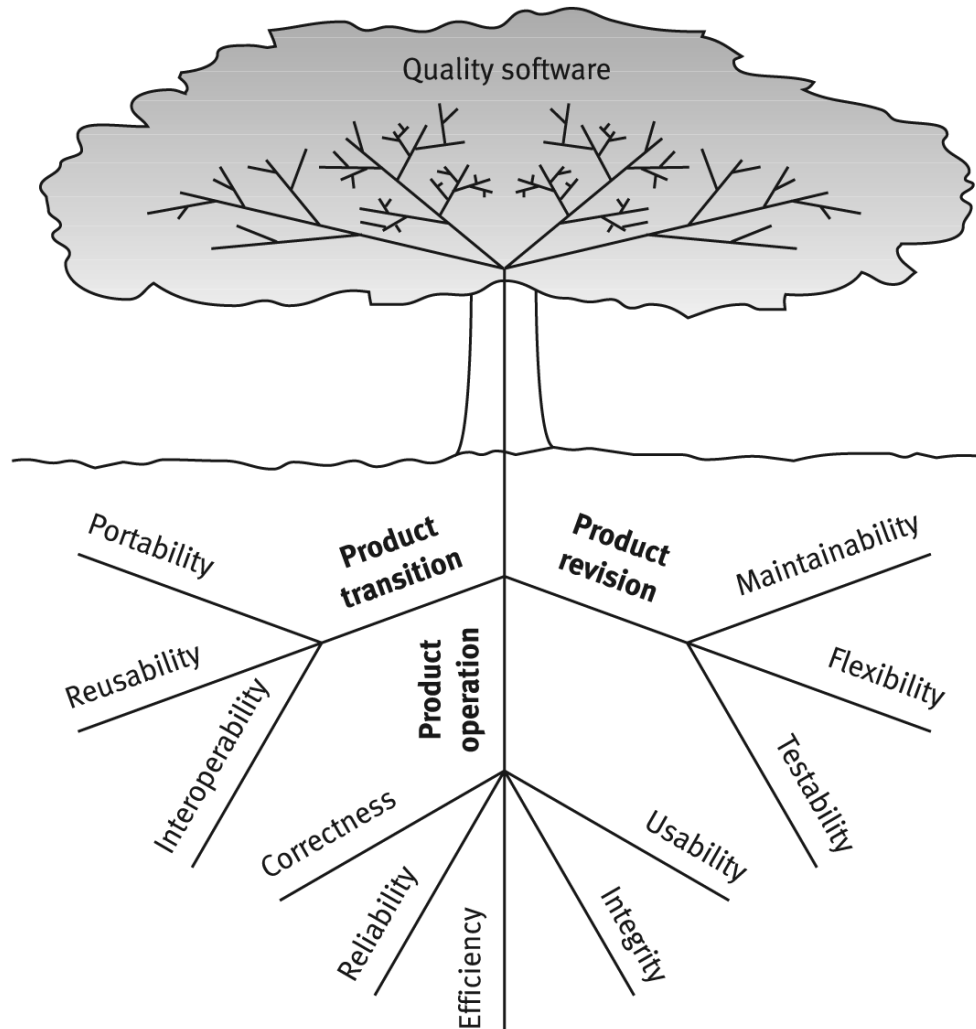
## Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models

- The quality model is the cornerstone of a product quality evaluation system.
  - The quality model determines which quality characteristics will be taken into account when evaluating the properties of a software product.
- 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.
- The product quality model defined in ISO/IEC 25010 comprises the eight quality characteristics



# McCalls factor model tree

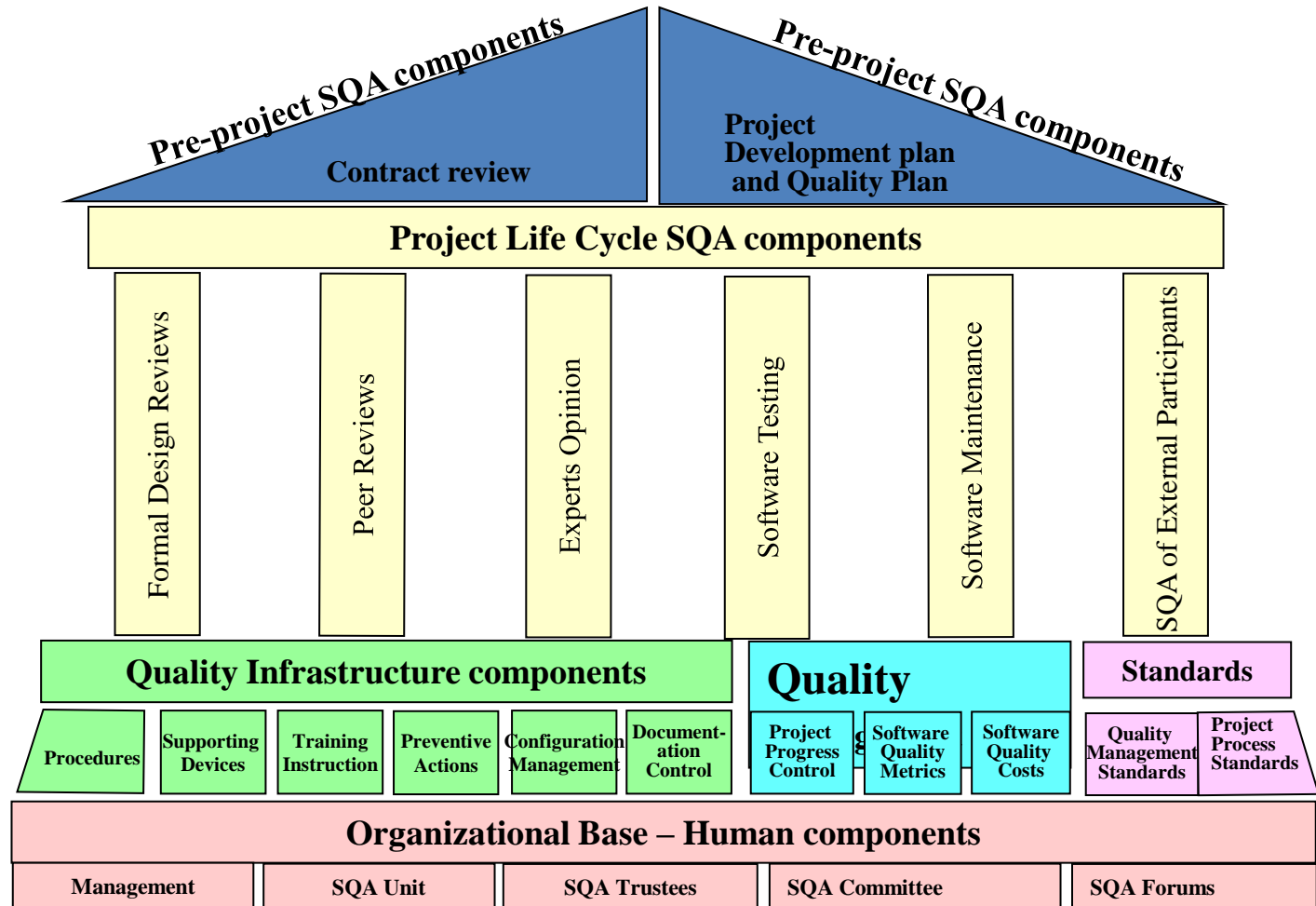
*Galin Chapter 3 pp. 38*





# The Software Quality Shrine

*Galin Chapter 4 pp. 59*



# The SQA system

- Goal is to minimize the number of software errors and to achieve an acceptable level of software quality
- Can be divided into to six classes:
  - Pre-project components
  - Components of project life cycle activities assessment
  - Components of infrastructure error prevention and improvement
  - Components of software quality management
  - Components of standardization, certification, and SQA system assessment
  - Organizing for SQA-the human components

# Summary

- Software Quality is important
  - Software controls life-and-death decisions, has enormous economic consequences, affects reputations, etc.
- An independent Quality Management System is vital component of an effort to produce quality software
  - Quality needs a “stakeholder”
- Think about quality requirements and a quality process.
- Quality assurance is a product of effective risk management
  - Tailor project management and software engineering practices to mitigate quality-related risks
- Measurement program is necessary
  - Demonstrate progress toward quality objectives using metrics, testing and other measurements

# Key points

- Software quality management is concerned with ensuring that software has a low number of defects and that it reaches the required standards of maintainability, reliability, portability etc.
  - Software standards are important for quality assurance as they represent an identification of ‘best practice’.
  - When developing software, standards provide a solid foundation for building good quality software.
- Reviews of the software process deliverables involve a team of people who check that quality standards are being followed. Reviews are the most widely used technique for assessing quality.
- Agile quality management relies on establishing a quality culture where the development team works together to improve software quality.