Software Quality Assurance & Test

LECTURE # 1

INTRODUCTION

chenbo@etao.net

Course Information

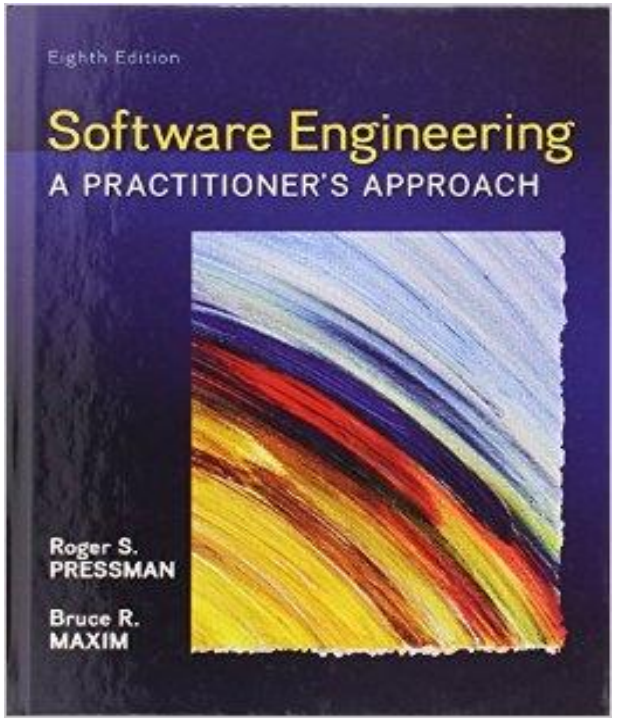
- **Introduction**

The purpose of the course is to introduce fundamental notions of software quality and the techniques used to build and check quality in software systems. A particular emphasis is placed on quantitative assessment of software quality and quality control using software testing techniques.

- **Learning Outcomes**

By the end of this course, students should have a good grasp of software quality, Software Quality Assurance, Configuration management, Static Testing techniques, White box testing and Black box testing techniques, Software metrics and models, Debugging, Quality Management, Process Improvement and Cleanroom Software Engineering.

Books and Resources



- **Software Engineering, A practitioner's approach**, 8th Edition by *Roger S. Pressman*

Reference Books

- **Recommended Text Books**

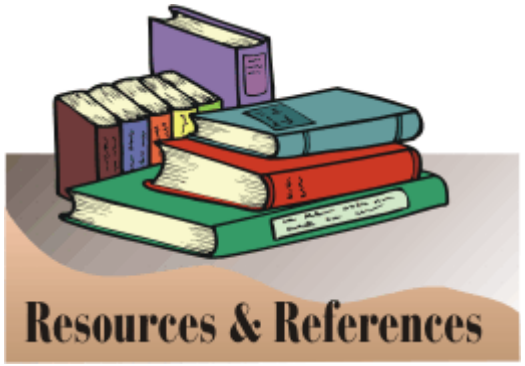
  Software Testing by Stephen Brown (2019)

- **Reference Books**

1. *Software Testing & Continuous quality improvement (2nd Edition) by William Lewis ISBN# 1420080733*
2. *The Art of Software Testing (2nd Edition) Glenford Myers ISBN# 0471043281*
3. *Software Quality Assurance by Daniel Galin （2003）*

Links and Reference Material

- http://www.onestoptesting.com/
- http://www.qastreet.com/
- http://www.computer.org
- http://www.springer.com
- http://portal.acm.org



Software Quality

- **Quality**
- **Quality Engineering**
- **Quality Types**
- **Quality Control vs Quality Assurance**
- **Different views of Quality**
- **Quality Models**
- **Cost of Quality**

What is quality?

- Quality, simplistically, means that a product should meet its specification.
- This is problematical for software systems
  - There is a 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.

Quality Engineering

Software quality assurance (SQA) consists of a **means** of monitoring the **software engineering** processes and methods used to ensure **quality**.

DEFINING QUALITY

- Perfection
- Consistency
- Eliminating waste
- Speed of delivery
- Compliance with policies & procedures
- Providing good, usable products
- Doing it right the first time
- Delighting or pleasing customers
- Total customer service & satisfaction

Quality Types

- **Functional Quality** — a measure of *what* the software does vs. what it's supposed to do

- **Non-Functional Quality** – a measure of *how well* it does it vs. how well it's supposed to do it

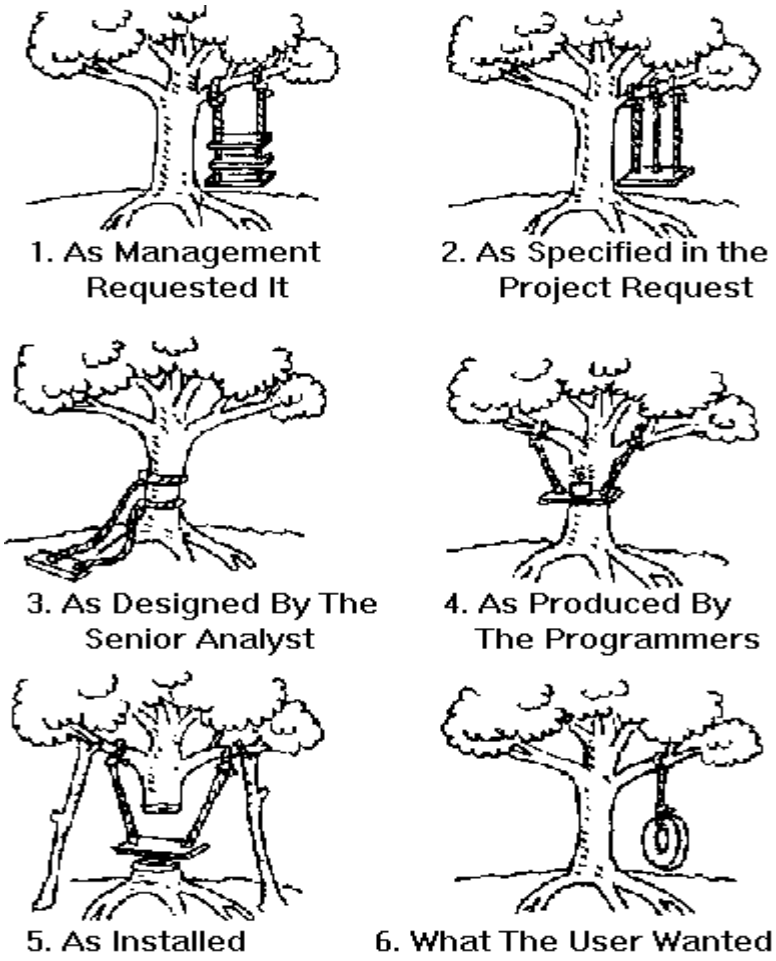
- **Latent Quality** — a measure of *how well it will continue to do it* in the future

Quality Types

- Process quality: Software processes implement best practices of software engineering in an organizational context. Process quality expresses the degree to which defined processes were followed and completed.
- Product quality: Software products are the output of software processes. Product quality is determined by the degree to which the developed software meets the defined requirements.
- Quality in use: Quality in use addresses the degree to which a product is fit for purpose when exposed to a particular context of use.

Different Views of Quality

- **Transcendental View**
- **User View**
- **Manufacturing View**
- **Product View**
- **Value based View**

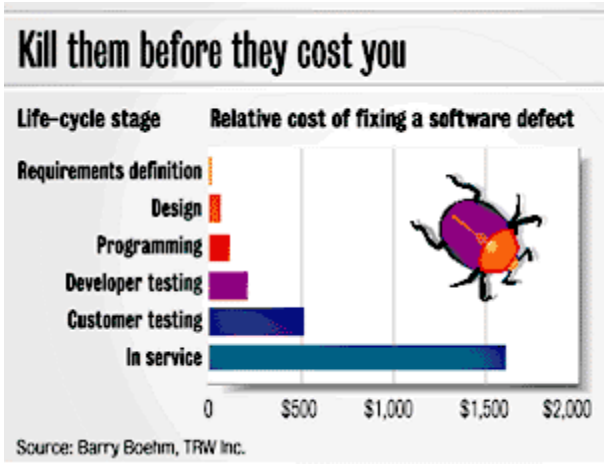


Quality Models

- **Standard quality models**
  - **McCall**
  - **ISO/IEC 9126**
  - **IEEE**
- **Application or company specific quality models**
  - **FURPS**
  - **GQM Approach**

Cost of Quality

- **Prevention cost:**
- **Appraisal cost:**
- **Failure cost:**
  - **Internal failure cost**
  - **External failure cost**



Quality Assurance

- **Quality Assurance**
- **Dealing with Pre- Post Release Defects**
- **Classification Scheme for Quality Assurance**
  - **Defect prevention**
  - **Defect Reduction**
  - **Defect Containment**

Software Quality Assurance

“All planned & systematic activities implemented within quality system, & demonstrated as needed, to provide adequate confidence that an entity will fulfill requirements for quality”.

Classification scheme for QA

- Defect Prevention
  - Education and training
  - Process conformance and standards enforcement
  - Tools/technologies and techniques
- Defect Reduction (Defect Detection and Removal)
  - Static Testing
  - Dynamic Testing
- Defect Containment
  - Software Fault Tolerance
  - Fault Containment

Software Quality Assurance

- SQA Team
- SQA Plan
- Software Review
- Formal Technical Reviews
- Review Guidelines

SQA Team

- **Testing Manager**
- **Test Team Lead**
- **Test Analyst**
- **Tester**
- **Independent Test Observer**

SQA Plan

- The software quality assurance plan is an outline of quality measures to ensure quality levels within a software development effort.
- The plan is used to compare the actual levels of quality during development with the planned levels of quality.
- If the levels of quality are not within the planned quality levels, management will respond appropriately as documented within the plan.

Software Reviews

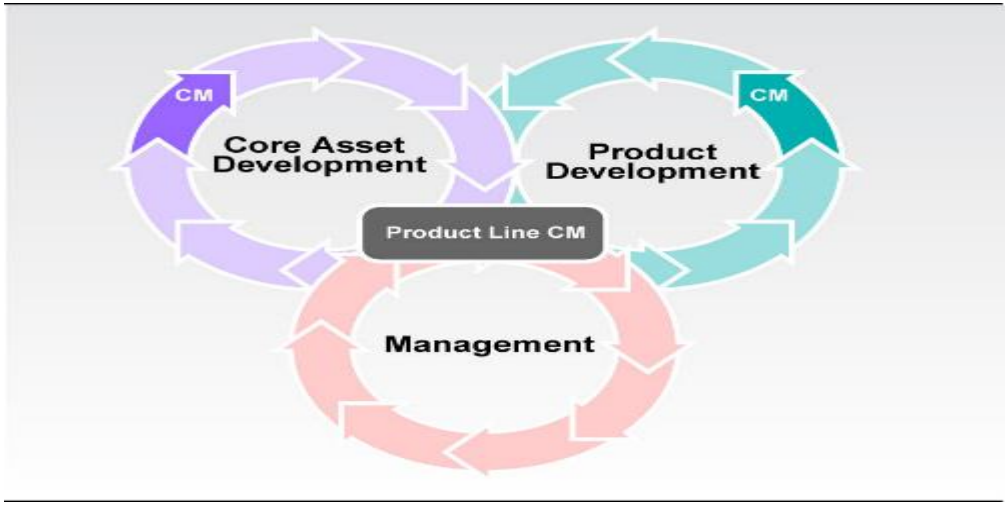
- *What is software review?*
- *Why software reviews?*
- *Objectives*
- *Review Meeting Constraints*
- *Review Meeting Results*

Software Configuration Management

- **Configuration**
- **Software Configuration Management**
- **Baselines**
- **Software Configuration Items**
- **SCM Process**
- **SCM Tasks**

Software Configuration Management

- Software Configuration Management is the art of identifying, organizing and controlling modifications to the software being built by a programming team. It maximizes productivity by minimizing mistakes

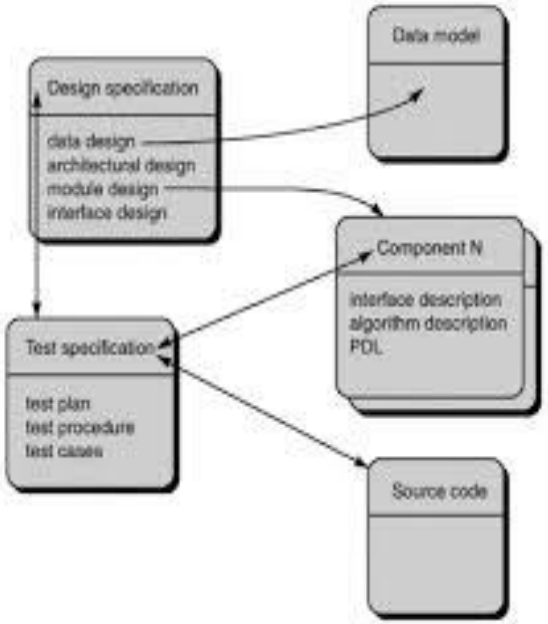


Baselines

- IEEE Std. No. 610.12-1990 defines baselines as,
  - **“ A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.”**

Software Configuration Items

- The items that comprises all information produced as a part of software process are collectively called a software configuration items.



SCM Tasks

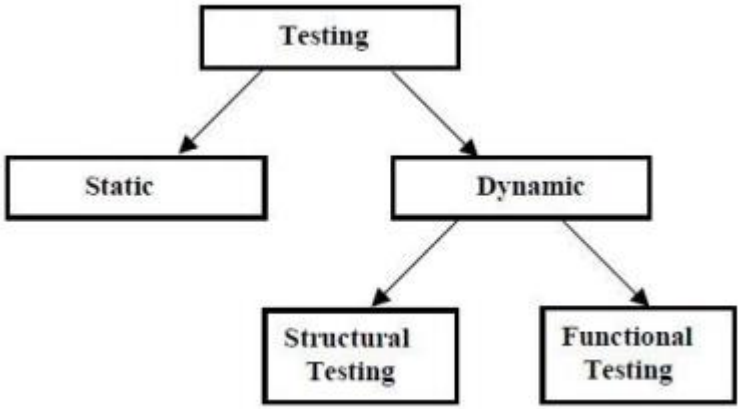
- **Identifications of objects**
- **Version Control**
- **Change Control**
- **Configuration Auditing**
- **Status Reporting**

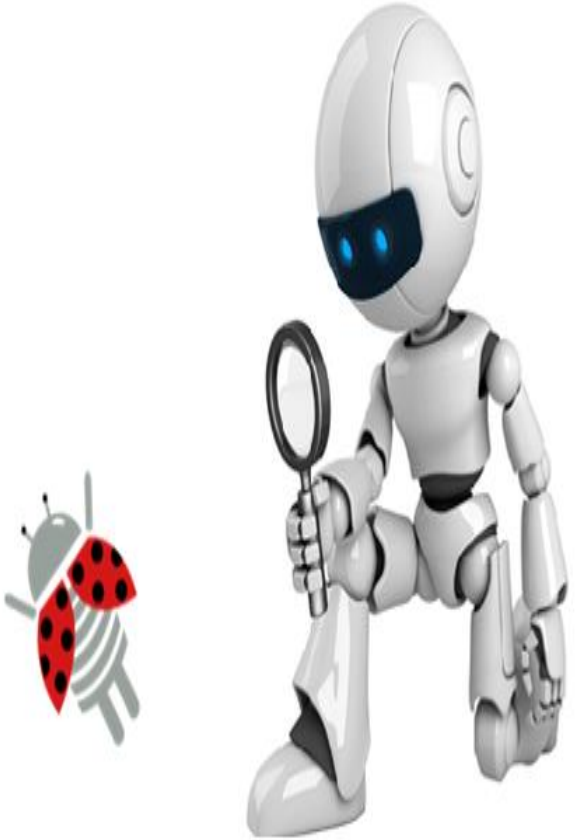
Software Testing

- What is Software Testing?
- Software Testing Activities
- Software Testing Principles
- Test Cases
- Basic Types of testing
- Methods of Testing

Software Testing

"Testing is the process of executing a program or system with the intent of finding errors."
by Myers 1979





Test Case

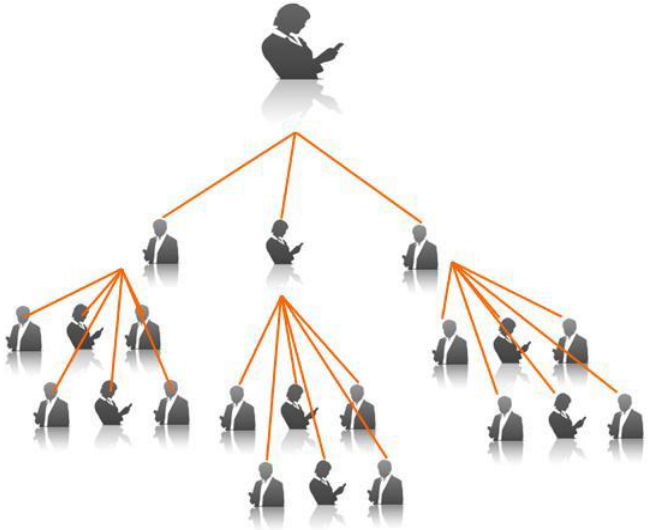
- A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system meets specifications.

Software Testing Activities

- Test Planning
- Test Case Design and Specification
- Test Set up
- Test Execution
- Test Result Analysis and Reporting
- Problem/Bug Reporting
- Test Management and Measurement
- Test Automation

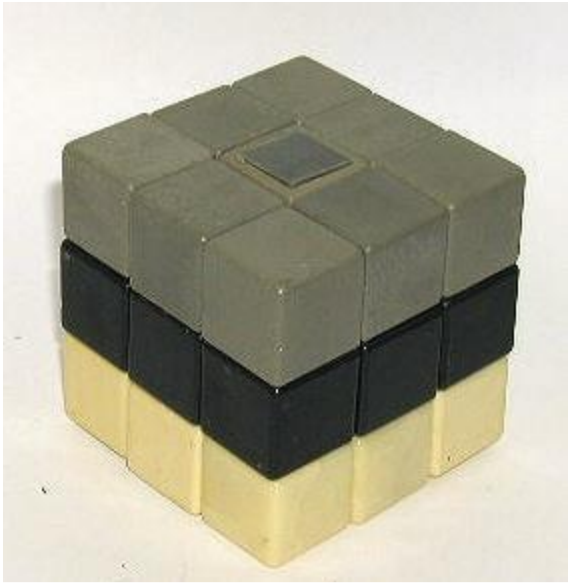
Methods of Testing

- Manual Testing
- Automated Testing



Types of Testing

- Black Box Testing
- White Box Testing
- Gray Box Testing

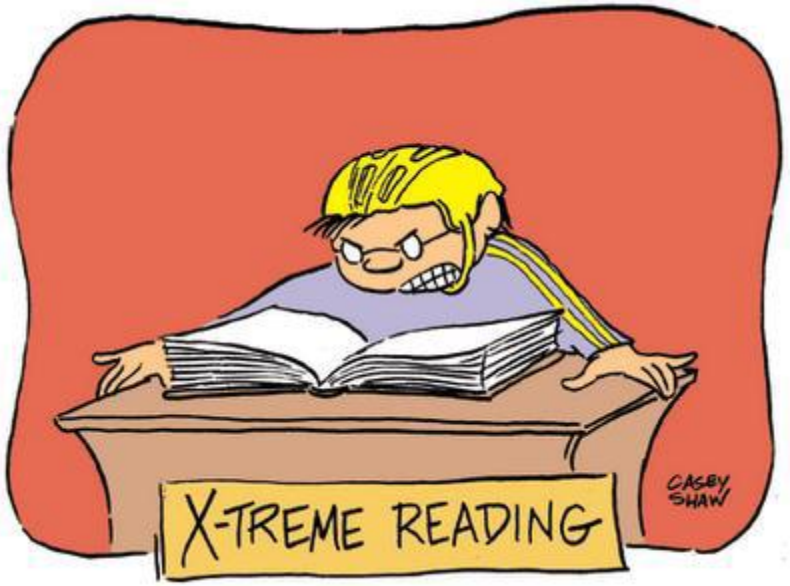


Static Testing

- Introduction of static testing
- Static Testing Approach
- Static Testing Methods
  - Inspections
  - Walkthroughs
  - Desk Checking
  - Peer Ratings

Static Testing

- In software development, static testing, also called *dry run testing*, is a form of software testing where the authors manually read their documents/code to find any errors.



Static Testing Methods

- Inspections
  - Fagan Inspection
  - Gilb Inspection
  - Two Person Inspection
  - N-Fold Inspection
  - Meetingless Inspection
- Walkthroughs
- Desk Checking
- Peer Ratings

Black Box Testing Methods

- Equivalence Class Partitioning Testing
- Boundary Value Testing
- Omission Testing
- Null Case Testing
- Volume Testing
- Load Testing
- Stress Testing
- Performance Testing
- Resource Testing
- Requirements/Specification Testing
- Button Press Testing

Black Box Testing Methods

- State Transition Testing
- Installation Testing
- Security Testing
- Integration Testing
- Compatibility Testing
- Configuration Testing
- Documentation Testing
- Smoke Testing
- Sanity Testing
- Usability Testing
- Exploratory Testing

White Box Testing Methods

- Statement Coverage
- Segment Coverage
- Branch Coverage
- Compound Condition Coverage
- Basis Path Testing
- Data Flow Testing
- Loop Testing

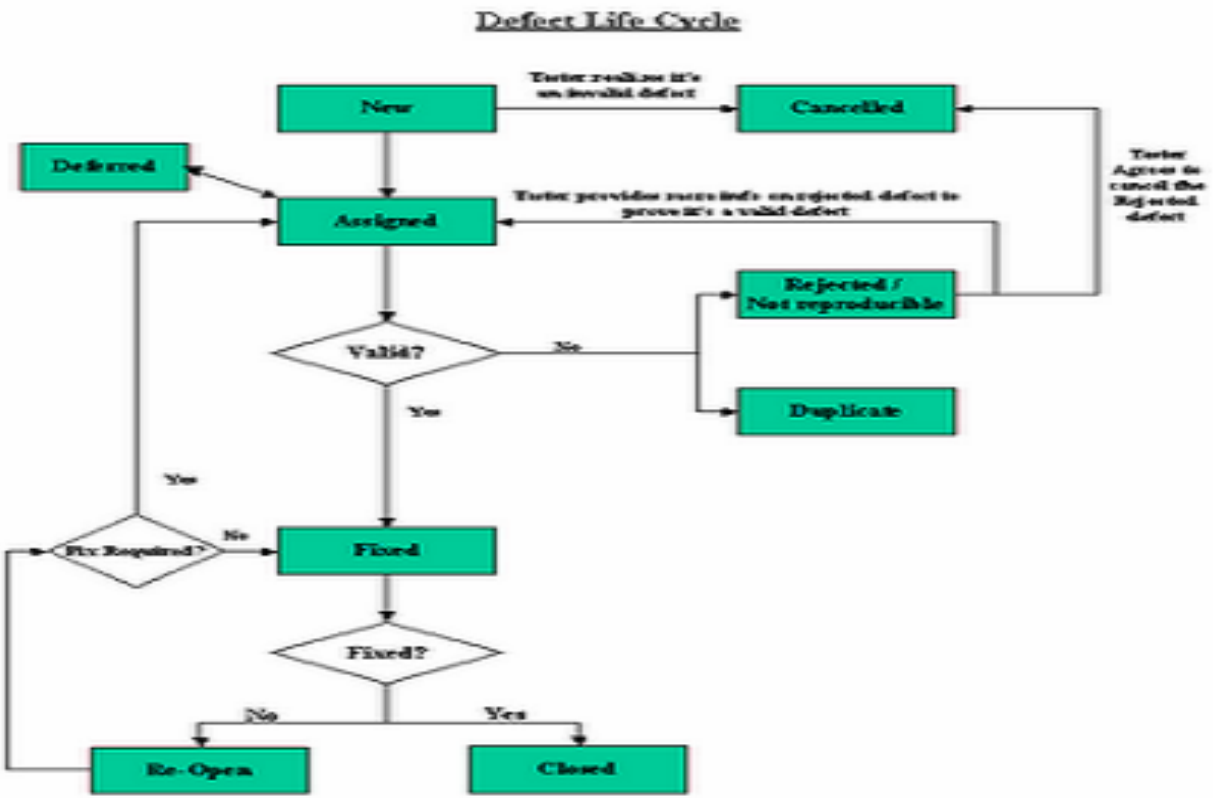
Debugging

- Debugging
- Bug Life Cycle
- Reporting
- Bug/Defect Types
- Defect Report
- Methods of Debugging

Debugging

- **Debugging** is the activity which is performed after executing a successful test case.
- Debugging consists of determining the exact nature and location of the suspected error and fixing the error.

Bug Lifecycle



Reporting

- Daily
- Weekly
- End of Cycle

Bug Types

- Showstopper
- Critical
- Non-Critical (Minor)

Methods of Debugging

- **Debugging by Brute Force Attack**
- **Debugging by Induction**
- **Debugging by Deduction**
- **Debugging by Backtracking**
- **Debugging by Testing**

Quality Management

- Quality Management Overview
- Quality Assurance
- Quality Planning
- Quality Control
- Software Measurement and Metrics

Quality Management

- The term **Quality management** has a specific meaning within many business sectors. This specific definition can be considered to have the following main components:
  - quality assurance
  - quality planning
  - quality control

Software Metrics

- A software metric is a quantitative measure of degree to which a system, component or process possesses a given attribute.

  - **Lines of code in a program, number of person-days required to develop a component, KLOC/defects etc.**

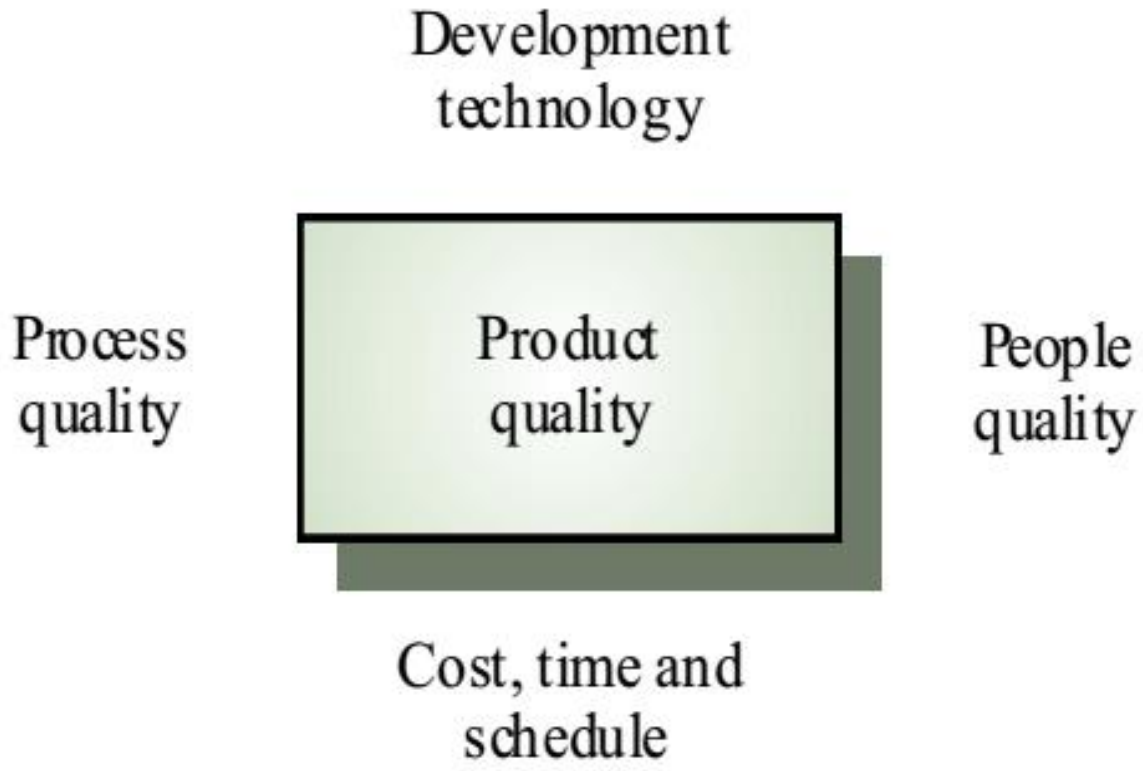
Process Improvement

- Process Improvement
- Process and product quality
- Process analysis
- Process change
- CMMI Model
- Six Sigma

Process Improvement

- "Process improvement" means making things better or more formally Understanding existing processes and introducing process changes to improve product quality, reduce costs or accelerate schedules.

Process & Product Quality



Process Analysis

- Questionnaires and Interviews

- Ethnographic Analysis

Process Change

- Involves making modifications to existing processes.

CMMI [9]

- CMMI is a process improvement approach that provides organizations with the essential elements of effective processes that ultimately improve their performance.

- The CMMI model has 5 levels:
  - **Initial**
  - **Managed**
  - **Defined**
  - **Quantitatively Managed**
  - **Optimizing**

Six Sigma [8]

- **Six Sigma** is a business management strategy originally developed by Motorola, USA in 1986. As of 2010, it is widely used in many sectors of industry.
- Six Sigma seeks to improve the quality of process outputs by identifying and removing the causes of defects (errors) and minimizing variability in manufacturing and business processes.

Cleanroom Software Engineering[10]

- Cleanroom Software Engineering
- Cleanroom Functions
- Cleanroom Strategy

References

1. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement: JEFF TIAN, 2005.
2. The Art of Software Testing by Glenford Myers
3. Software Engineering by Ian Sommerville
4. Software Engineering by Roger Pressman
5. WhiteBox and BlackBox Testing methods used in LMKR Software house
6. http://www.mosaicinc.com/mosaicinc/rmThisMonth.asp
7. http://en.wikipedia.org/wiki/Quality_management
8. http://en.wikipedia.org/wiki/Six_sigma
9. http://en.wikipedia.org/wiki/CMMI
10. http://en.wikipedia.org/wiki/Cleanroom_software_engineering