PRINCIPLES OF PROJECT MANAGEMENT AND STAFF RECRUITMENT

QUALITY ASSURANCE AND TESTING.

Lecturer: Alin G.T.

Instructor: Rakhimzhanova N.K.

Outline

- Software Quality Assurance Plan
- Definition of quality for software products
- Software Metrics
- Software Testing, types of testing

Outcomes

- Understand the key parts of the Software Testing process
- Know how to identify the metrics of software
- Be able to write a SQAP
- Have a clear understanding of what is Quality in software products

Software Quality Assurance Plan

The purpose of the Software Quality Assurance Plan (SQAP) is to define the techniques, procedures, and methodologies that will be used at project to **assure timely delivery of the software that meets specified requirements** within project resources.

Software Quality Assurance Plan

- Set common templates (standards)
- Define the sequence of actions
- Ensure that standards and processes are used
- Conduct an analysis of completed projects
- Analyze and learn, using the defect data
- Use what you have learned

What is Quality?

- How do you understand the term Quality of software product?
- Is it rather about conformance to requirements?
- Is it rather about fitness of use?

What is Quality?

- ***Verification*** – The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with *validation*.
- ***Validation*** -The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with *verification*.

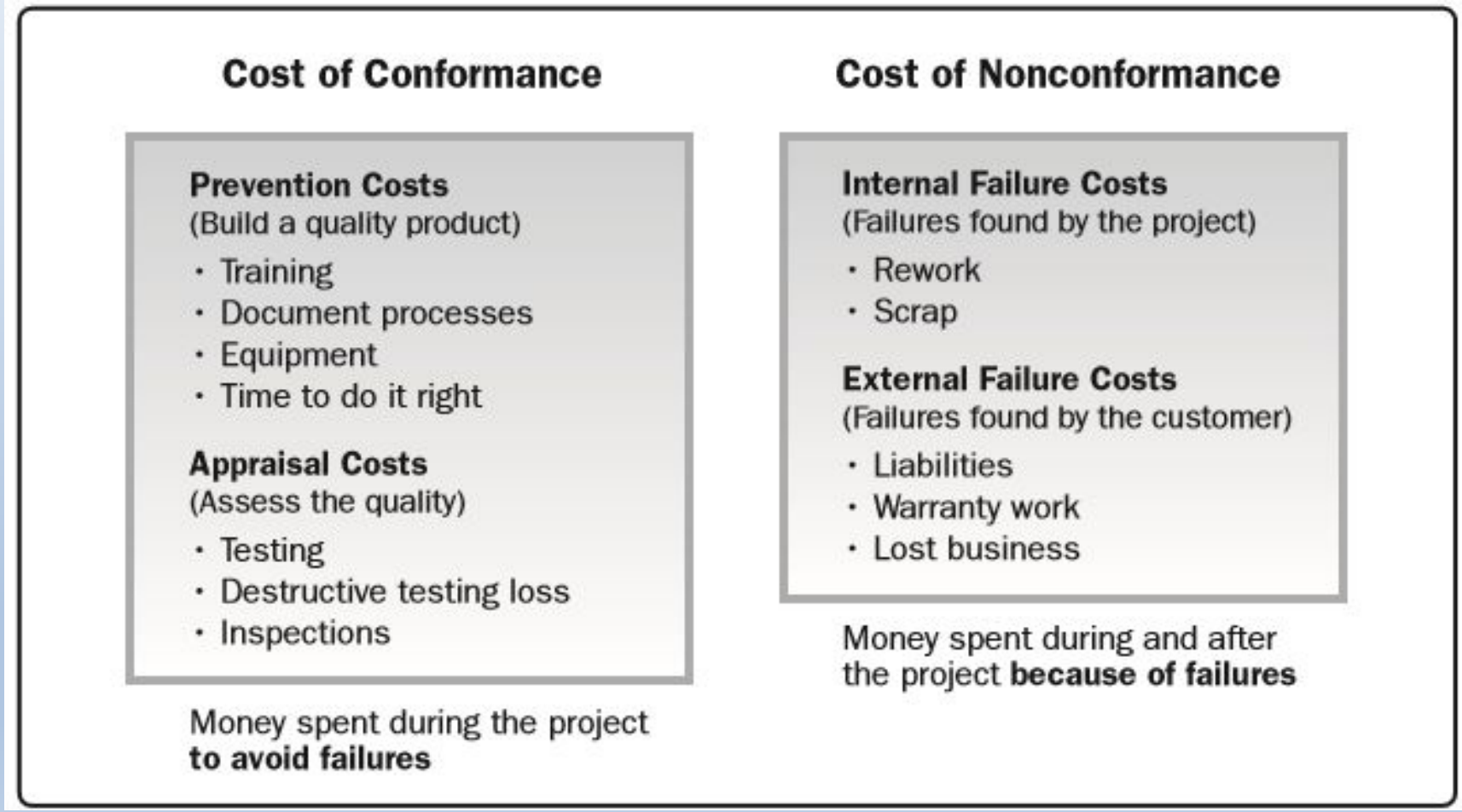
Fault, failure and error

Fault/defect – a condition that may cause a failure in a system, also called a bug.

Failure/problem – the inability of a system to perform a function according to its specification, result of a defect.

Error – a mistake made by software engineer or programmer

Cost of Quality (CoQ)



Software Project Metrics

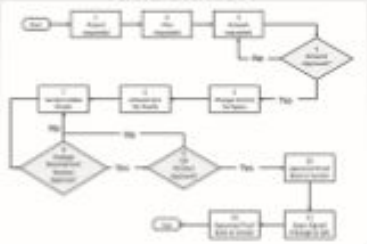
- Tools for anyone involved in software engineering to understand varying aspects of the code base, and the project progress.
- They are different from just testing for errors because they can provide a wider variety of information about the following aspects of software systems:

- Quality of the software, different metrics look at different aspects of quality
- Schedule of the software project on the whole, some metrics look at functionality and some look at documents produced.
- Cost of the software project. Includes maintenance, research and typical costs associated with a project.
- Size/Complexity of the software system. This can be either based on the code or at the macro-level of the project and it's dependency on other projects.

Seven basic Quality tools

Cause & Effect Diagram

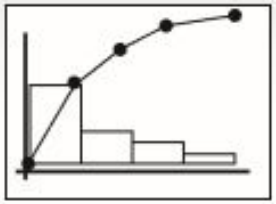
Flowcharts



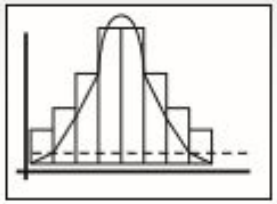
Checksheets

Category	*Strokes*	*Frequency*
Attribute 1		
Attribute 2		
Attribute ...		
Attribute n		

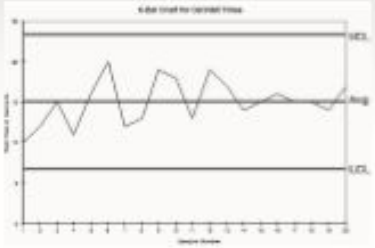
Pareto Diagrams



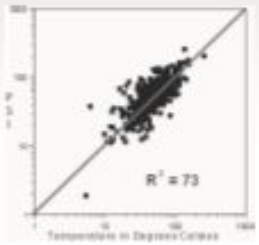
Histograms



Control Charts



Scatter Diagrams



Software Metrics: effects of proper usage

- Reduce cost by 15% - 20% by just measuring
- Create baseline of quality and productivity and compare against industry averages.
- Pinpoint opportunities for improvement.
- Ability to measure initiatives and measure ROI (return of investments).

Software Project Metrics: types

- Life Cycle Step metrics
- Costs and budget metrics
- Requirements' change metrics
- Development process metrics
- Testing metrics
- Defect metrics
- Efficiency metrics

Software Project Metrics in Agile

- An agile version of the Goal Question Metric (GQM) strategy.
- The fundamental idea behind GQM is that you first identify:
  - a goal that you would like to achieve,
  - a set of questions whose answers are pertinent to determining how well you're achieving that goal,
  - and then the metric(s) that could help you to answer each question

Software Project Metrics

General Project Metrics:

1. Completed activities budget (percentage of completed tasks)
2. Actual budget ratio of the planned budget (Budget(actual) / Budget(planned))
3. Dispersion (var) of costs (Budget(actual) - Budget(planned))
4. Schedule execution (Effort (actual) / Effort (planned))
5. Dispersion (var) of schedule (Effort (actual) - Effort (planned))
6. Schedule delays ($\sum$ delay time)
7. Coefficient of closed tasks (closed tasks / planned tasks)
8. Productivity

Software Metrics

Requirements Metrics:

- Frequency of change in the total requirements set
- Rate of introduction of new requirements
- Traceability
- Volatility of requirements
- Percentage of defects as requirement as a root cause
- Number of requirement-related change requests
- Requirement Stability Index : 1- ((No of changed + No of deleted + No of added) / Total no of Initial requirements) x100

Software Metrics

Process Metrics:

Category	Formula
Programmer productivity	LOS Produced/Persons months of effort
Module defect density	no. of defects/module size
Defect detection efficiency	no. of defects detected/total no. of defects
Requirement stability	no. of initial requirements/total no. of requirements
Test effectiveness ration	no.of items covered/total no. of items
System spoilage	effort spent fixing faults/total project effort

Software Metrics

Product Metrics:

- Testing
  - General
    - Testing time
  - Test cases metrics
    - Passed/Failed Test Cases
    - Not Run Test Cases
  - Bugs
    - Open/Closed Bugs
    - Reopened/Closed Bugs
    - Rejected/Opened Bugs
    - Bugs by Severity
    - Bugs by Priority

What is Testing of SW?

Maintaining a set of techniques for detecting and correcting errors in a software products

(testing process can be automated)

Testing should be applied to all artifacts of software projects development.

Testing

Test Plan - a document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

Testing

- **Master Test Plan:** A single high-level test plan for a project/product that unifies all other test plans.
- **Testing Level Specific Test Plans:** Plans for each level of testing.
  - Unit Test Plan
  - Integration Test Plan
  - System Test Plan
  - Acceptance Test Plan
- **Testing Type Specific Test Plans:** Plans for major types of testing like Performance Test Plan and Security Test Plan.

Testing of SW?

Who does the testing

-Programmers (developers)

-Testers

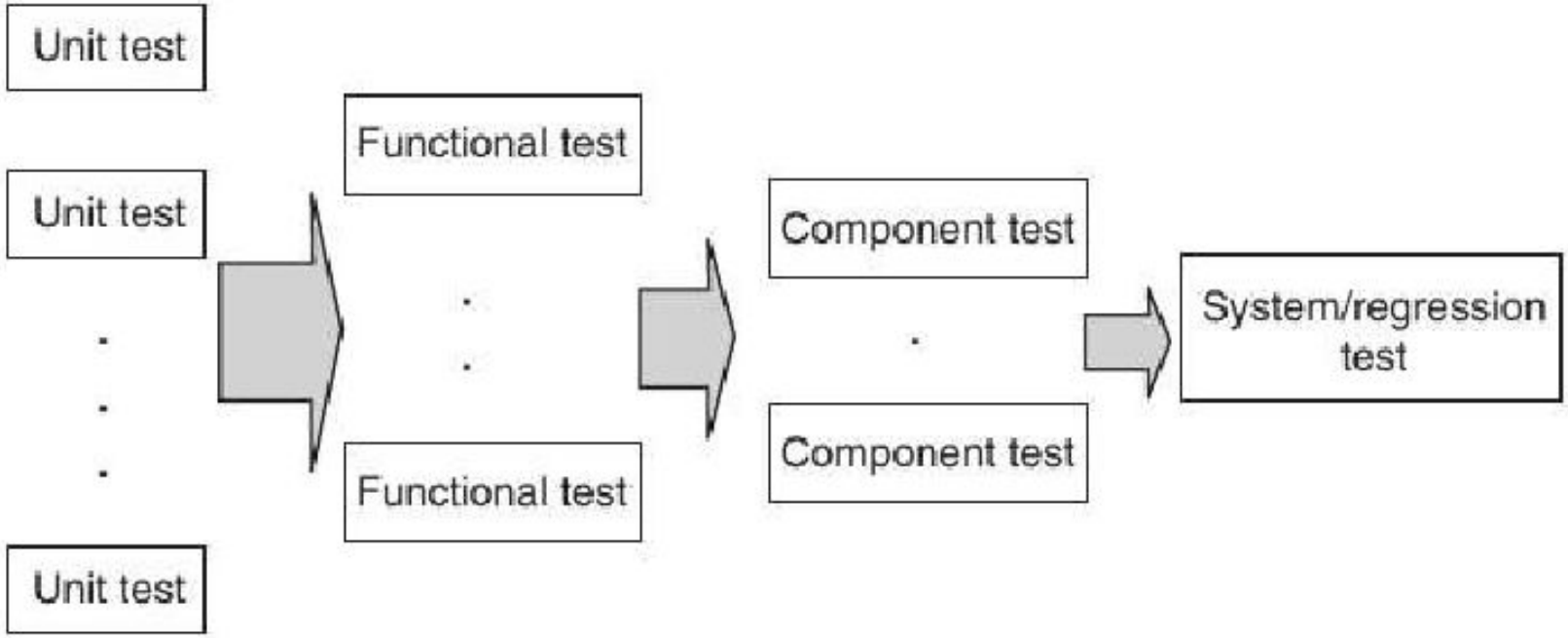
-Users (Alpha testing & Beta testing)

Testing levels:

-Unit testing;

-Functional testing;

-Integration and system testing (regression test, smoke test);


Unit test
Unit test
Unit test
Functional test
Functional test
Component test
Component test
System/regression test

Testing of SW?

Testing purposes:

-Acceptance testing

-Conformance testing

-Configuration testing;

-Performance testing;

-Stress testing;

-User interface testing

Test cases based on:

- Intuition
- Specification (known as black-box testing)
- Code (white-box testing)
- Existing test cases
- Faults

Requirements Traceability Matrix

Requirement Traceability Matrix – Parameters include

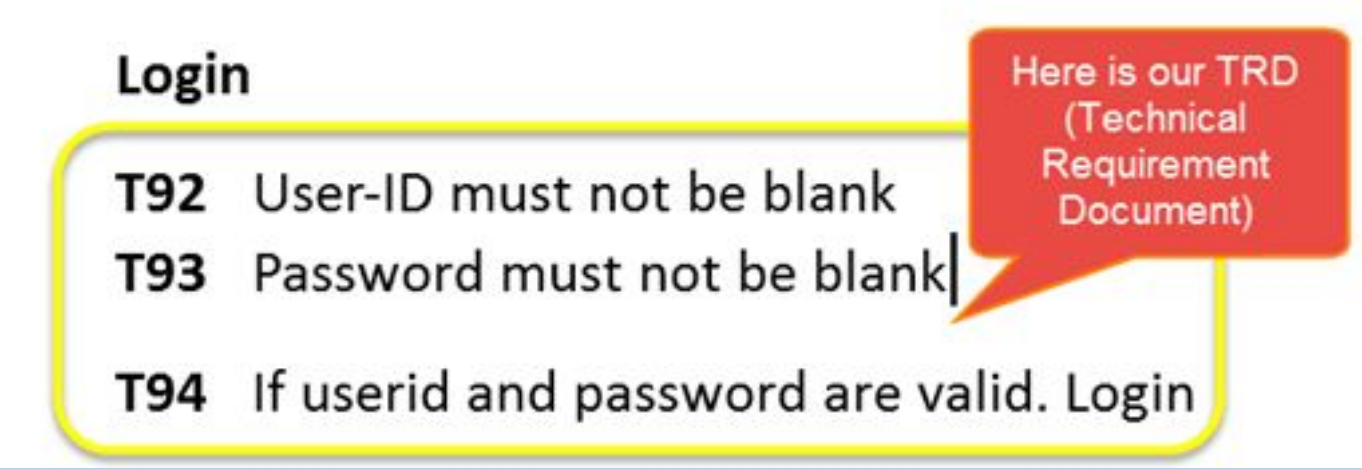
- Requirement ID
- Risks
- Requirement Type and Description
- Trace to design specification
- Unit test cases
- Integration test cases
- System test cases
- User acceptance test cases
- Trace to test script

Requirements Traceability Matrix

BR#	Module Name	Applicable Roles	Description
B1	Login and Logout	Manager Customer	**Customer:** A customer can login using the login page **Manager:** A manager can login using the login page f customer. Post Login homepage will show ifferent links based on role
B2	Enquiry	Customer	**ustomer:** A customer can have multiple bank accounts. He can view balance of his accounts only **Manager:** A manager can view balance of all the customers who come under his supervision
B3	Fund Transfer	Manager Customer	**Customer:** A customer can have transfer funds from his "own" account to any destination account. **Manager:** A manager can transfer funds from any

Business Requirement # for Guru99 banking project

Requirements Traceability Matrix



Requirements Traceability Matrix

TestCase #	Test Case	Test Steps	Test Data	Expected Result
1	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

When correct password and id entered, it should login successfully

Requirements Traceability Matrix

Requirements Traceability Matrix

TestCase #	TR #		Test Steps	Test Data	Expected
1	T94	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

Note the Technical Requirement in the test case

Requirements Traceability Matrix

TestCase #	BR #	TR #	Test Case	Test Steps	Test Data	Expe
1	B1	T94	Verify Login	1) Go to Login Page 2) Enter UserID 3) Enter Password 4) Click Login	id= Guru99 pass= 1234	Login Successful

Requirements Traceability Matrix

Business Requirement #	Technical Requirement #	Test Case ID
B1	T94	1
B2	T95	3
B3	T96	3
B4	T97	4

Requirement Traceability Matrix

Product Complexity Metrics

1. Source lines of code.
2. Cyclomatic complexity, is used to measure code complexity.
3. Function point analysis (FPA), is used to measure the size (functions) of software.
4. Bugs per lines of code.
5. Bang Metric

Reading assignments

1. Software Project Survival Guide By Steve McConnell Microsoft Press, Chapter 9
2. Essentials of Software Engineering By Frank Tsui , Orlando Karam Jones & Bartlett Learning, Chapter 10
3. A Guide to the Project Management Body of Knowledge. Chapter 6