SCHULICH School of Engineering

UNIVERSITY OF CALGARY

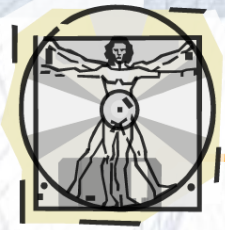
SENG 637
Dependability and Reliability of Software Systems

Chapter 12: Post-release Activities and Maintenance

Department of Electrical & Software Engineering, University of Calgary

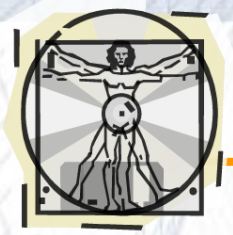
B.H. Far (far@ucalgary.ca)

http://people.ucalgary.ca/~far



Contents

- Quality in software development process
- Software Quality System (SQS); Software Quality Assurance (SQA) and Software Reliability Engineering (SRE)
- Quality, test and data plans
  - Roles and responsibilities
  - Sample quality and test plan
- Post release activities



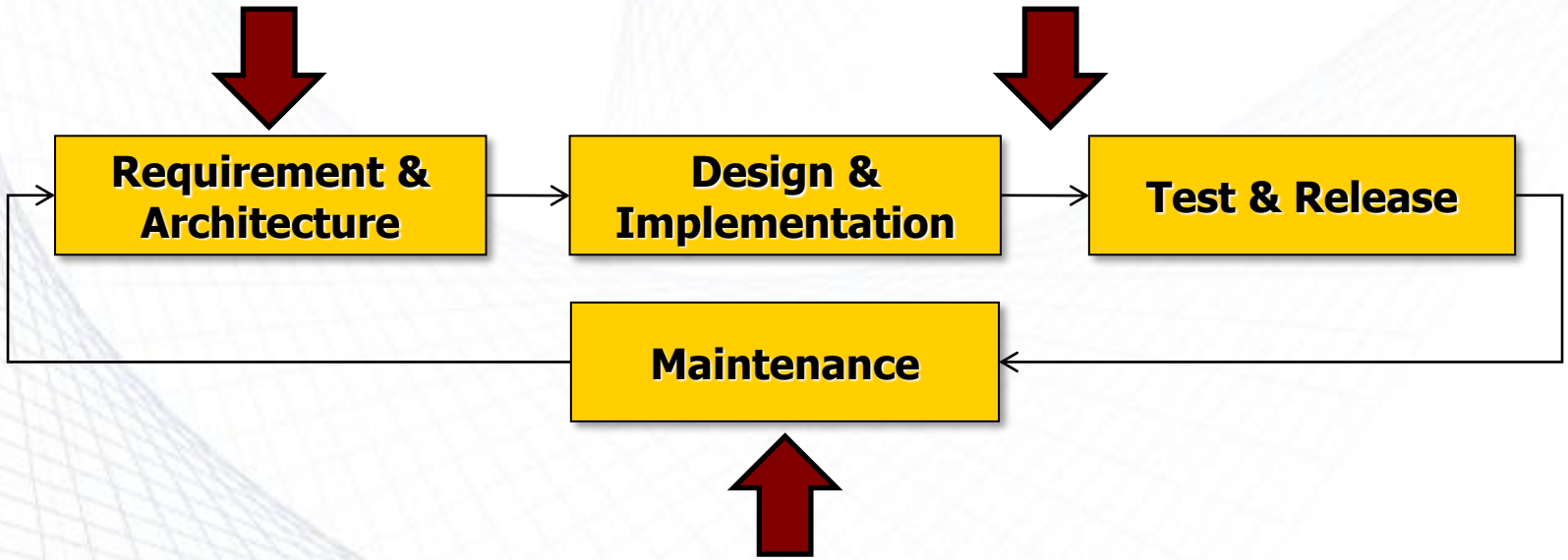
Quality in Software Development Process

Q. How to include quality concerns in the Process?

Architectural analysis
Quality attributes
Method: ATAM, CBAM, etc.

Software Reliability Engineering (SRE)

Software Quality Assurance (SQA)

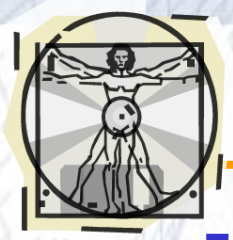


Software Quality Assessment
Method: RAM, etc.

UNIVERSITY OF CALGARY

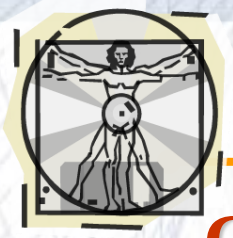
Section 1

Software Quality System (SQS) and Software Quality Assurance (SQA) programs



What is Reliable Software?

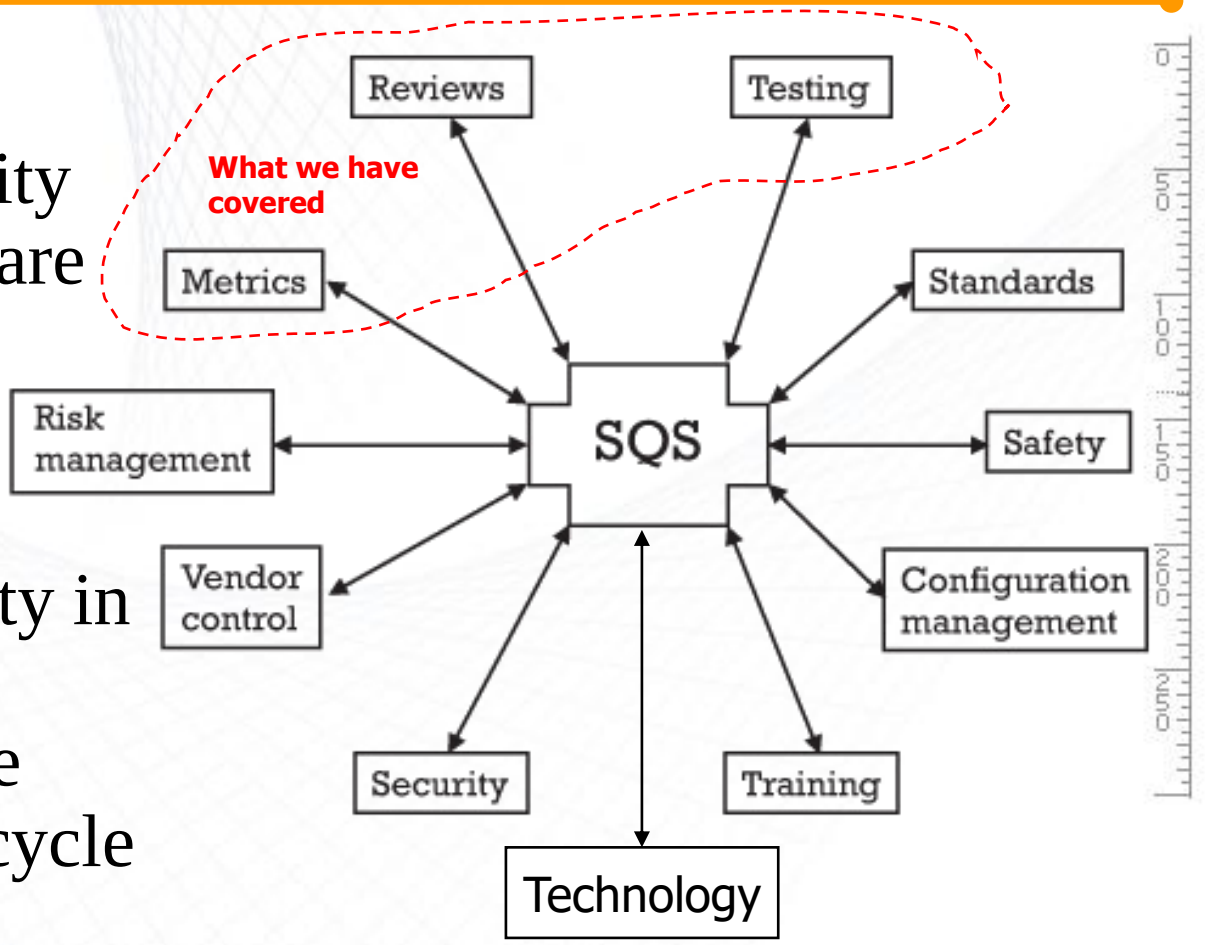
- Reliable software products are those that run correctly and consistently, have fewer remaining defects, handle abnormal situation properly, and need less installation effort
- The remaining defects should not affect the normal behaviour and the use of the software; they will not do any destructive damages to system and its hardware or software environment; and rarely are evident to the users
- Developing reliable software requires:
  - Establishing Software Quality System (SQS) and Software Quality Assurance (SQA) programs
  - Establishing Software Reliability Engineering (SRE) process



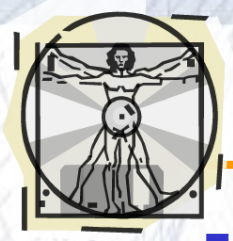
Software Quality System (SQS)

Goals:

- Building quality into the software from the beginning
- Keeping and tracking quality in the software throughout the software life cycle



John W. Horch: Practical Guide to Software Quality Management



Software Quality Assurance (SQA)

- **Software quality Assurance (SQA)** is a planned and systematic approach to ensure that both software process and software product conform to the established standards, processes, and procedures.
- The goals of SQA are to improve software quality by monitoring both software product and the development process to ensure full compliance with the established standards and procedures.
- Steps to establish an SQA program
  - Get the top management's agreement on its goal and support.
  - Identify SQA issues, write SQA plan, establish standards and SQA functions, implement the SQA plan and evaluate SQA program.

Software Quality Plans

Requirement & Architecture	Design & Implementation	Test

Quality Plan

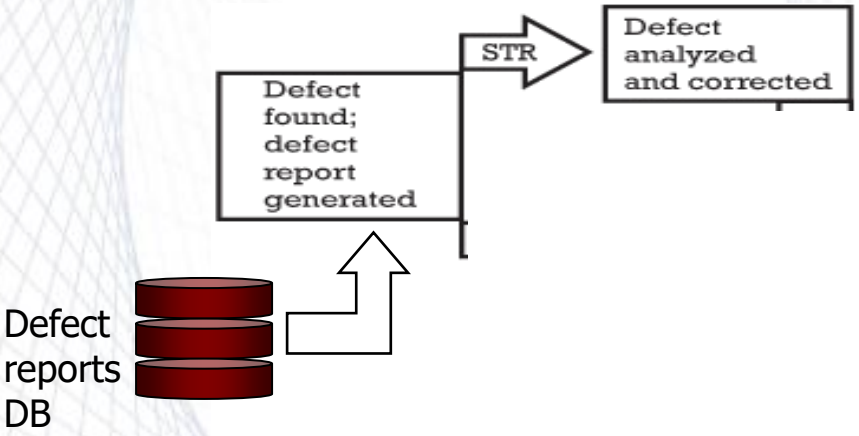
Test Plan

Data Plan

There may be many Test and Data (measurement) plans for various parts of the same project

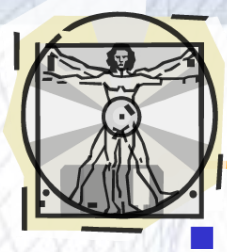
Defect Handling: Without & With SQS

- Defect reporting, tracking, and closure procedure



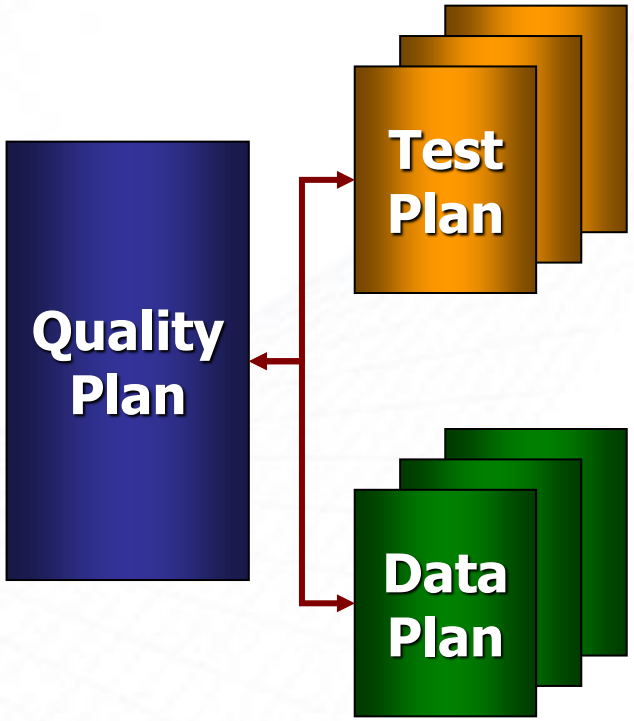
SCN: software change notice

STR: software trouble report



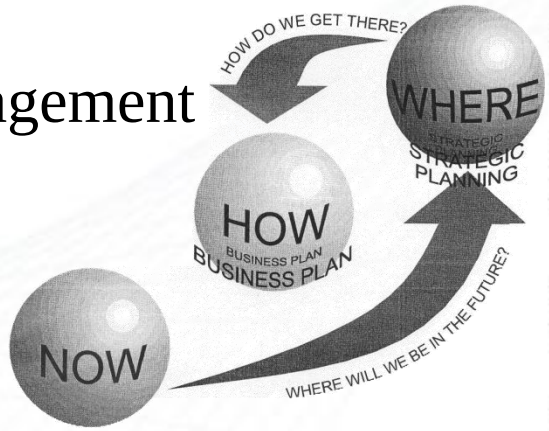
Quality Plans /1

- The most promising mechanisms for gaining and improving predictability and controllability of software qualities are **quality plan** and its subsidiary documents, including **test plans** and **data (measurement) plans**
- The creation of the **quality plan** can be instrumental in raising project effectiveness and in preventing expensive and time-consuming misunderstandings during the project, and at release/acceptance time



Quality Plan /2

- Quality plan and quality record, provide guidelines for carrying out and controlling the followings:

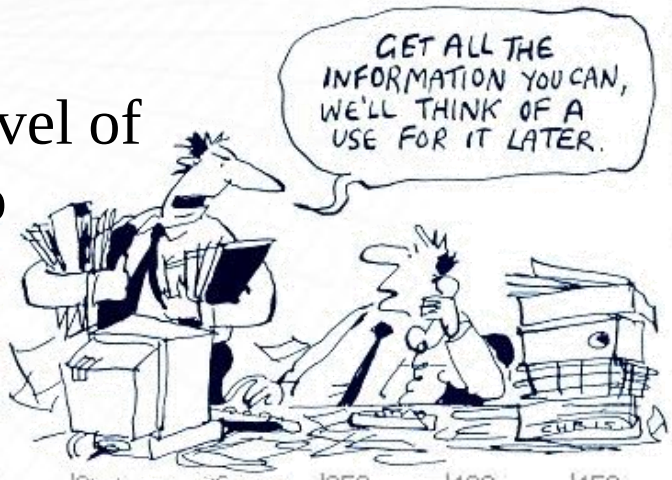


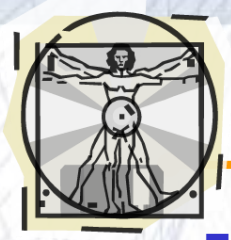
  - Requirement and specification management
  - Development processes
  - Documentation management
  - Design evaluation
  - Product testing
  - Data collection and interpretation
  - Acceptance and release processes

Testing related activities

Data (Measurement) Plan

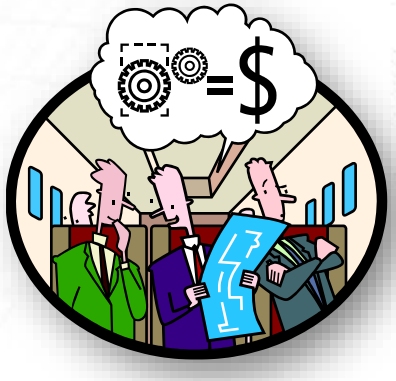
- The **data (measurement) plan** prescribes:
  - What should be measured and recorded during a project
  - How it should be checked and collated
  - How it should be interpreted and applied
- Data may be collected in several ways, within the specific project and beyond it
- Ideally, there should be a higher level of data collection and application into which project data is fed

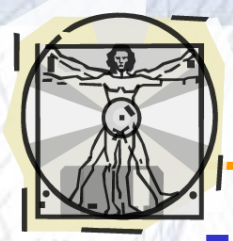




Test Plan /1

- The purpose of **test plan** is to ensure that all testing activities (including those used for controlling the process of development, and in indicating the progress of the project) are expected, are manageable and are managed
- Test plans are created as a subsection or as an associated document of the quality plan
- Test plans become progressively more detailed and expanded during a project
- Each test plan defines its own objectives and scope, and the means and methods by which the objectives are expected to be met





Test Plan /2

- For the software product, the test plan is usually restricted by the scope of the test: ***certification***, ***unit*** and ***integration*** test
- The plan predicts the resources and means required to reach the required levels of assurance about the end products, and the scheduling of all testing, measuring and demonstration activities
- Tests, measurements and demonstrations are used to establish that the software product satisfies the requirements document, and that each process during a development is carried out correctly and results in acceptable outcomes

SCHULICH School of Engineering

UNIVERSITY OF CALGARY

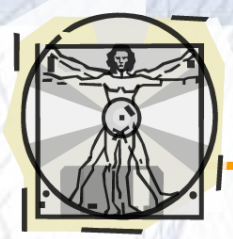
Section 2

Elements of Quality & Test Plan

Sample SQS Plan /1

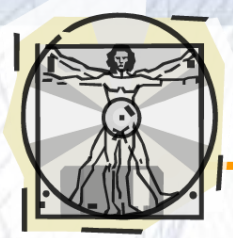
- 1 Purpose
- 2 Reference Documents
- 3 Management
  - 3.1 Organization
  - 3.2 Tasks
  - 3.3 Responsibilities

IEEE Standards
730.1 - 1989



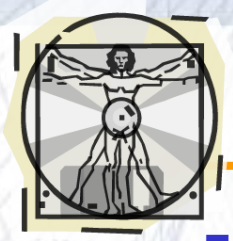
Sample SQS Plan (cont'd) /2

- 4 Documentation
  - 4.1 Purpose
  - 4.2 Minimum documentation
    - 4.2.1 Software requirements specification
    - 4.2.2 Software design description
    - 4.2.3 Software verification and validation plan
    - 4.2.4 Software verification and validation report
    - 4.2.5 User documentation
    - 4.2.6 Configuration management plan
  - 4.3 Other documentation



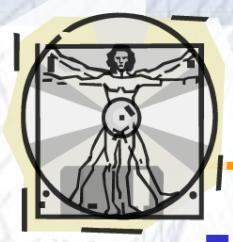
Sample SQS Plan (cont'd) /3

- 5 Standards, practices, conventions, and metrics
  - 5.1 Purpose
  - 5.2 Documentation, logic, coding, and commentary standards and conventions
  - 5.3 Testing standards, conventions, and practices
  - 5.4 Metrics



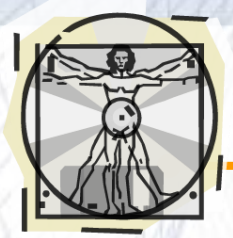
Sample SQS Plan (cont'd) /4

- 6 Review 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 Software Verification and Validation Review
    - 6.2.5 Functional Audit
    - 6.2.6 Physical Audit
    - 6.2.7 In-process Reviews
    - 6.2.8 Managerial Reviews
    - 6.2.9 Configuration Management Plan Review
    - 6.2.10 Postmortem Review
  - 6.3 Other Reviews and Audits



Sample SQS Plan (cont'd) /5

- 7 Test
- 8 Problem reporting and corrective action
  - 8.1 Practices and procedures
  - 8.2 Organizational responsibilities
- 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



Sample Test Plan /1

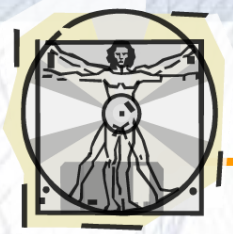
- 1 Test Plan identifier
- 2 Introduction
  - 2.1 Objectives
  - 2.2 Background
  - 2.3 Scope
  - 2.4 References

IEEE Standards
829 - 1983

Sample Test Plan (cont'd) /2

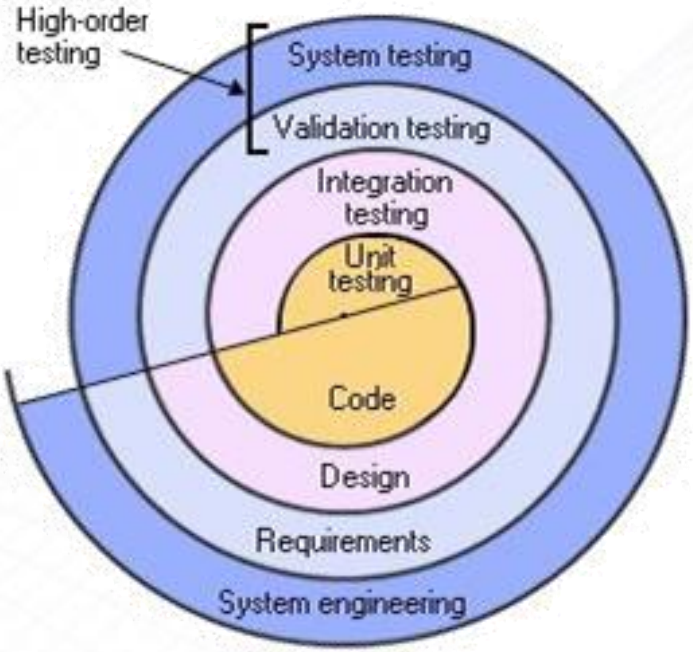
- 3 Test Items
  - 3.1 Program modules
  - 3.2 Job control procedures
  - 3.3 User procedures
  - 3.4 Operator procedures
- 4 Features to be tested
- 5 Feature not to be tested

Based on IEEE Standard 829-1983



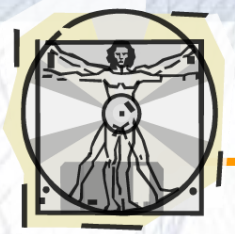
Sample Test Plan (cont'd) /3

- 6 Approach
  - 6.1 Conversion testing
  - 6.2 Job stream testing
  - 6.3 Interface testing
  - 6.4 Security testing
  - 6.5 Recovery testing
  - 6.6 Performance testing
  - 6.7 Regression
  - 6.8 Comprehensiveness
  - 6.9 Constraints



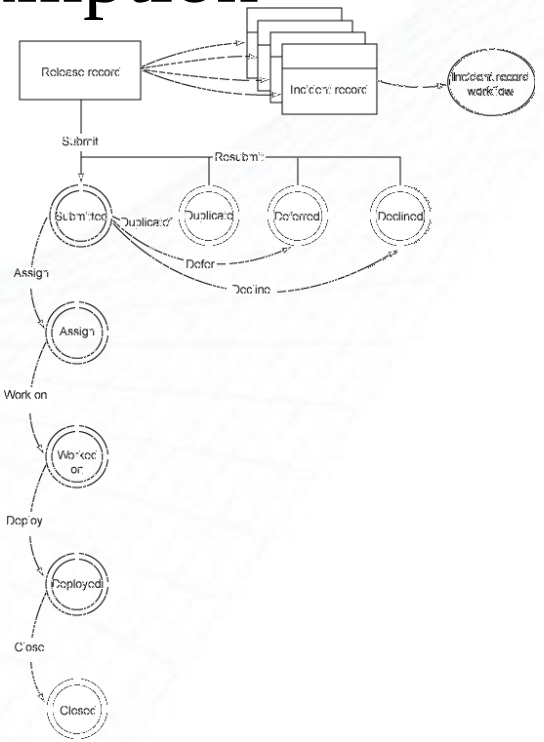
The testing strategy

Based on IEEE Standard 829-1983

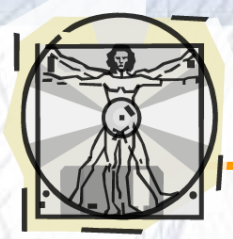


Sample Test Plan (cont'd) /4

- 7 Item pass/fail criteria
- 8 Suspension criteria and resumption requirements
  - 8.1 Suspension criteria
  - 8.2 Resumption requirements
- 9 Test deliverables
- 10 Testing tasks

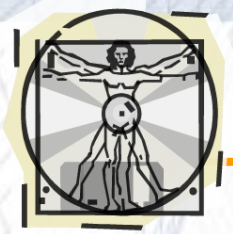


Based on IEEE Standard 829-1983



Sample Test Plan (cont'd) /5

- 11 Environmental needs
  - 11.1 Hardware
  - 11.2 Software
  - 11.3 Security
  - 11.4 Tools
  - 11.5 Publications
- 12 Responsibilities
  - 12.1 Test group
  - 12.2 User department
  - 12.3 Development project group



Sample Test Plan (cont'd) /6

- 13 Staffing and training needs
  - 13.1 Staffing
  - 13.2 Training
- 14 Schedule
- 15 Risks and contingencies
- 16 Approvals

Based on IEEE Standard 829-1983

SCHULICH School of Engineering

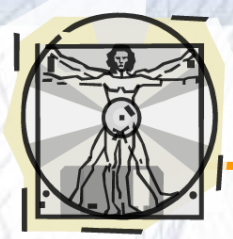
UNIVERSITY OF CALGARY

Section 3

Post Release Practice

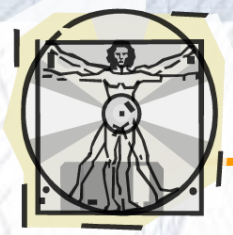
Post Release Activities

- **Post delivery and maintenance:**
  - Project post-release staff needs
  - Monitor field reliability vs. objectives
  - Track customer satisfaction with reliability
  - Time new feature introduction by monitoring reliability
  - Guide product and process improvement with reliability measures



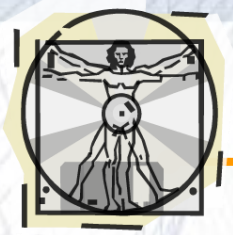
Post Release Activities /1

- **Activity 1: Project post-release staff needs**
  - Customer's staff for system recovery; supplier's staff to handle customer-reported failures and to remove faults
  - **Benefits:** Reduce post-release costs with better planning
- **Activity 2: Monitor field reliability vs. objectives**
  - Collect post release failure data systematically
  - **Benefits:** Maximize likelihood of pleasing customer with reliability



Post Release Activities /1

- **Activity 3: Track customer satisfaction with reliability**
  - Survey product features with a sample customer set
  - **Benefits:** Maximize likelihood of pleasing customer with reliability
- **Activity 4: Time new feature introduction by monitoring reliability**
  - New features bring new defects. Add new features desired by the customers if they can be managed without sacrificing reliability of the whole system
  - **Benefits:** Ensure that software continues to meet customer reliability needs in the field



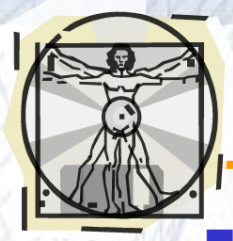
Post Release Activities /1

- **Activity 5: Guide product and process improvement with reliability measures**
  - Root-cause analysis for the faults
  - Why the fault was not detected earlier in the development phase and what should be done to reduce the probability of introducing similar faults
  - **Benefits:** Maximize cost-effectiveness of product and process improvements selected

What is Next?

- Testing practices and tools need to evolve to address the challenges of achieving "Quality at Speed" amid the increasing complexity of systems, environments, and data.

- DevOps



Conclusions …

- Practical implementation of an effective software quality program is a non-trivial task
- Mechanisms for collection and analysis of data on software product and process quality must be in place
- Fault identification and elimination techniques must be in place
- Other organizational abilities such as the use of reviews and inspections, reliability based testing, and software process improvement are also necessary
- Quality oriented mindset and training are necessary!