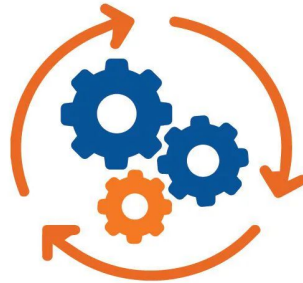


Chapter 4

Automated Testing Introduction Process



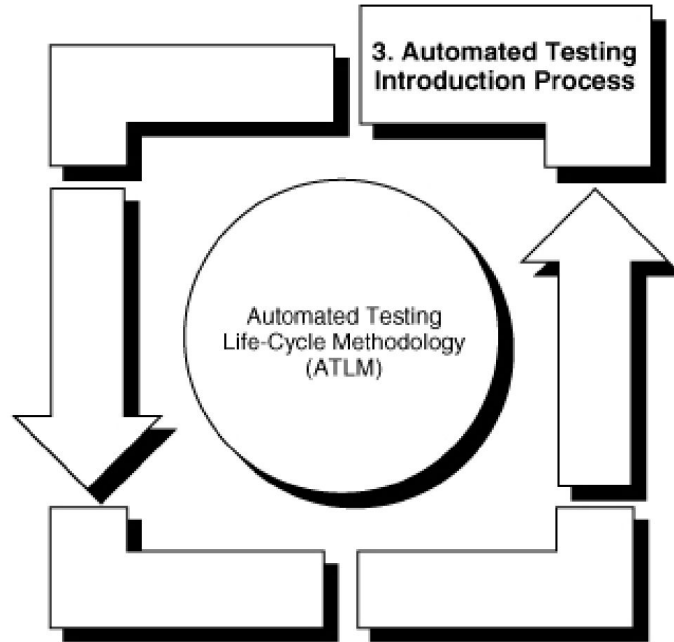
Lecturer: Nguyen Thanh Quan (MSc)
Email: tg_nguyenthanhquan_cntt@tdtu.edu.vn

Content

- 1. Test Process Analysis**
- 2. Test Tool Consideration**

Introduction (1)

A tool is only as good as the process being used to implement the tool. How a tool is implemented and used is what really matters.



Introduction (2)

- **How test teams introduce** an automated software test tool on a new project is nearly **as important as the selection** of the most appropriate test tool for the project.
- **Implementing** automated testing tools **without having a process or strategy** in place describing in detail the steps results in the development of test scripts that are **not reusable**.
- Test **efforts** that **drag out** unexpectedly tend to receive a significant amount of **criticism** and unwanted management attention.
- **Unplanned extensions** to the test schedule may have several **undesirable consequences** to the organization (loss of customer/market).

Introduction (3)

- Implement a test tool **too late** in the development life cycle, time lost while learning to work, ramping up on tool features and capabilities put the testing effort **behind schedule**.
- The team may become **frustrated**
- The test team may have had the best intentions in mind, but **unprepared** to exercise the best course of action.
- **No experience** with the tool.
- **A ripple effect throughout an organization** (reputation impact)

Introduction (3)

Test teams need to view the introduction of an automated test tool into a new project as a process, not an event.

The test tool needs to complement the process, not the reverse.

Introduction (3)

- Test teams need to view the introduction of an automated test tool into a new project as a process, not an event.
- The test tool needs to complement the process, not the reverse.

4.1 Test Process Analysis

Goals and Objectives

Test Strategies

Measurements

4.2 Test Tool Consideration

Review Requirements

Overview of
Application-Under-Test

Test Tool Compatibility

Project Schedule Review

Test Tool Demonstration

Roles and Responsibilities

Training Requirements

Continue with ATLM Phase 4

Test Process Analysis

Test Process Analysis (1)

The test process must be documented in such a way that it can be communicated to others

- If the test process is **not documented**, then it **cannot be communicated** or **executed** in a repeatable fashion.
 - If it **cannot be communicated** or is **not documented**, then often a process is **not implemented**.
 - If a process is **documented**, it can be **measured** and therefore **improved**.
- The Automated Test Life-cycle Methodology (ATLM) can be adopted to solve the issue of test process which is not yet documented, or outdated or inadequate.

Test Process Analysis (2)

The test team should become familiar with the organization's quality and process improvement objectives

- The Software Engineering Institute's Capability Maturity Model (CMM) can be applied because it is intended to create an infrastructure of people and proven practices:
 - Enable the organization to produce quality products.
 - Achieve customer satisfaction.
 - Meet project objectives.

Documented test process and **defined, collected, analyzed** metrics make effective improvements.

Test Process Analysis (3)

Performance of testing in accordance with the defined test process should be maintained and documented

Should the organization undergo an assessment of its compliance with CMM guidelines:

- It will verify that the test team's activities match the activities defined within the test process.
- It will check whether process outputs or artifacts produced comply with the defined test process.

Test Process Analysis (4)

- The **purpose of analyzing** the organization's test process is to **identify** the test **goals, objectives, and strategies** that may be inherent in the test process.
- The **purpose of documenting** the test tool introduction process is to ensure that the test team **has a clearly defined way** of implementing automated testing, then allowing the team to **fully leverage** the functionality and time-saving features of the automated test tool.

Test Process Analysis (5)

Process Review (1)

The test engineer needs to analyze the existing development and test process to determine whether the current testing process meets several prerequisites:

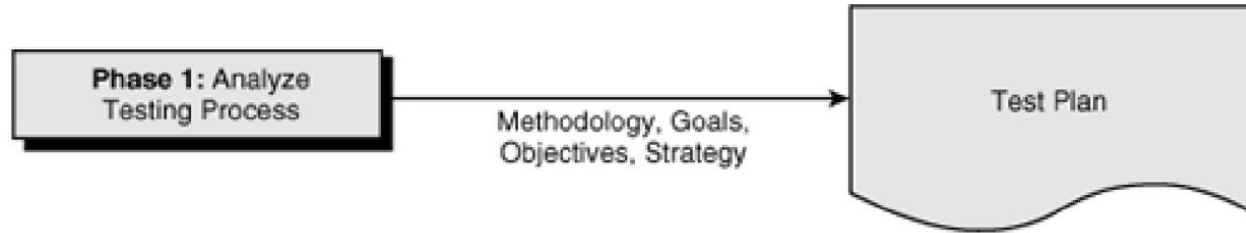
- Testing goals and objectives have been defined.
- Testing strategies have been defined.
- The tools needed are available to implement planned strategies.
- A testing methodology has been defined.
- The testing process is communicated and documented.

Test Process Analysis (6)

Process Review (2)

- The testing process is being measured.
- The testing process implementation is audited.
- Users are involved throughout the test program.
- Testing is conducted in parallel to the system development life cycle (SDLC).
- The test team is involved from the beginning of the SDLC.
- The schedule allows for process implementation.
- The budget allows for process implementation.

Test Process Analysis (7)



A successfully implemented test process will

- Minimize the schedule
- Achieve high defect detection efficiency
- Improve software quality
- Support the development of reliable systems
- **Keep the users happy**

Test Process Analysis (8)

Process Review / Early Involvement of Test Team

- The test process should be **initiated** at the **beginning** of the system development life cycle.
- The test team **augments** its **understanding** of **business rules** and **processes**.
- The test team **obtains** greater **understanding** of customer needs.
- The test team **verifies** the **testability** of requirements.
- Permits **earlier detection** of errors and **prevents** migration of **errors** from requirements specification to design, and from design into code.

Test Process Analysis (9)

Process Review / Repeatable Process (1)

- A repeatable process can be achieved by **documenting every step of the process** => testing can be controlled and implemented uniformly.
- **Automation of tests** is the most efficient means of achieving repeatability.
- A repeatable process can be further augmented through the application of **reusable test scripts**.
- The test team needs to verify that the ATLM process has been **properly implemented**.

Test Process Analysis (10)

Process Review / Repeatable Process (2)

The test team should review criteria by which the ATLM process:

- Performance of the process
- Repeatability (same results)
- Product traceability
- Stability of process
- Compliance of process

Test Process Analysis (11)

Process Review / Repeatable Process (3)

| Fitness to Execute Process | | Use of Defined Process | |
|--|---|---|---|
| Are all the requisites for successful process execution in place (for example, personnel, skills, experience, training, facilities, tools, documented procedures)? | | Is the process being executed faithfully? Are tools, methods, practices, and work plans being used? | |
| Entities | Attributes | Entities | Attributes |
| People | Skills, experience, training, quantity | People | Effort, time |
| Tool | Accessibility, adequacy, utility, skills | Tool | Use: how widely, frequently, long |
| Procedures | Coverage, sufficiency, quality, documentation | Procedures | Awareness use: how widely, frequently, long |
| Facilities | Space, computers, technical support, sufficiency | Facilities | Use: how widely, frequently, long |
| Work Plan | Targets, work breakdown structure applicability, understandable, doable | Work Plan | Awareness use: how widely, frequently, long |

Test Process Analysis (12)

Process Review / Continuous Improvement

- The primary goal is **continuous refinement** and **improvement** of the test process.
- It is necessary to document **lessons learned** (at the end of the **application development life cycle**) and **QA audit findings** throughout the testing life cycle and **take corrective action**.
- To identify needed improvement activities and ensure that mistakes are not repeated in the next test phase.

Test Process Analysis (13)

Process Review / Safeguarding the Integrity of the Automated Test Process (1)

- The test team needs to exercise new releases of an automated test tool in an **isolated** environment.
 - Validate that the tool performs up to **product specifications and marketing claims**.
 - Verify that the upgrades will run in the organization's current environment.
- Using a configuration management tool to **baseline the test repository** will help safeguard the integrity of the automated testing process.

Test Process Analysis (14)

Process Review / Safeguarding the Integrity of the Automated Test Process (2)

- Process definition, metric-gathering, and process improvement activities can be **expensive and time-consuming**.
- Creating and documenting standards and procedures for an automated test program may be **no more expensive** than the same activities for a manual test program.
- The test team must decide whether the process needs to be revised or whether the team can proceed with the existing test process.

Test Process Analysis (15)

Process Review / Goals and Objectives of Testing (1)

- **Meets** specific **criteria** and **satisfies** the **requirements** of the end user or customer by detecting as many defects as possible.
- Automated testing, when **implemented correctly**, promotes faster, better, and more efficient testing => reduce the **size** of the test effort, the test schedule, the production of a **reliable** system, the enhancement of a **repeatable** test process.

Test Process Analysis (16)

Process Review / Goals and Objectives of Testing (2)

A program is said to work correctly when it satisfies the following criteria:

- Given valid input, the program produces the correct output.
- Given invalid input, the program correctly and gracefully rejects the input.
- The program doesn't hang or crash, given either valid or invalid input.
- The program keeps running correctly for as long as expected.
- The program behaves as specified.

Test Process Analysis (17)

Process Review / Goals and Objectives of Testing (3)

A program is said to work correctly when it satisfies the following criteria:

- Given valid input, the program produces the correct output.
- Given invalid input, the program correctly and gracefully rejects the input.
- The program doesn't hang or crash, given either valid or invalid input.
- The program keeps running correctly for as long as expected.
- The program behaves as specified.

Test goals and objectives must be understood by each of team members.

Test Process Analysis (18)

Process Review / Goals and Objectives of Testing (4)

- When testing a commercial-off-the-shelf (COTS) tool, a primary objective would be the integration of the COTS tool with the rest of the system, and the corresponding test objective would be based on black-box testing.
 - When testing a homegrown system, the test team will be concerned with the application's internal workings, which necessitates white-box testing.
- Test objectives should be outlined early in the planning process and need to be clearly defined.

Test Process Analysis (19)

Process Review / Goals and Objectives of Testing (5)

Process Review

- **Test process.** The project will utilize the organization's standard test process, which adopts the ATLM.
- **Test tool introduction.** To support a smooth implementation of the automated test tool, the project team will verify that goals, objectives, and strategies are defined and compatible with automated testing. A test tool consideration phase will be conducted before continuing with phase 4 of the ATLM and executing each subsequent ATLM phase.

Test Process Analysis (20)

Process Review / Goals and Objectives of Testing (6)

Test Goals

- Increase the probability that the application-under-test will behave correctly under all circumstances
- Increase the probability that the application-under-test meets all defined requirements • Execute a complete test of the application within a short timeframe

Test Process Analysis (21)

Process Review / Goals and Objectives of Testing (7)

Test Objectives

- Ensure that the system complies with defined client and server response times
- Ensure that the most critical end-user paths through the system perform correctly
- Ensure that user screens perform correctly
- Ensure that database changes have not had an adverse effect on existing software modules
- Incorporate a test design that minimizes test rework following changes to the application
- Incorporate the use of automated test tools whenever feasible
- Perform test activities that support both defect prevention and defect detection
- Incorporate the use of automated test design and development standards so as to create reusable and maintainable scripts

Test Process Analysis (22)

Process Review / Goals and Objectives of Testing (8)

Example: You were assigned the responsibility for testing a database management system (DBMS) redesign project. The database was redesigned to accommodate additional data requirements related to a system upgrade. As a result of the upgrade, new tables were added and some tables were deleted. Additionally, a new version of Sybase was introduced.

→ **Goal:** Verify that the application will behave correctly following the database change.

Test Process Analysis (23)

Process Review / Goals and Objectives of Testing (9)

→ Objectives

- Ensure that data were successfully converted from the old to the new database.
- Ensure that the application development team can successfully upgrade to a new database management system (DBMS) release.
- Ensure that the system still complies with defined client and server response times.
- Ensure that end-user paths through the system and user screens perform correctly.

Test Process Analysis (24)

Process Review / Goals and Objectives of Testing (10)

→ **Strategies**

- Data Conversion and Data Integrity Testing.
- Database Version Upgrade.
- Performance Testing.
- Regression Testing.

Test Process Analysis (25)

Test Strategies

Defect Prevention Technologies

=> Provides the greatest cost and schedule savings, but it cannot always prohibit defects from entering into the application-under-test.

- ***Examination of constraints***
- ***Early test involvement***
- ***Use of process standards***
- ***Inspection and walkthroughs***
- ***Quality gates***

Test Process Analysis (26)

Test Strategies/Defect Prevention Strategies

- Reflect a high level of test discipline maturity.
- Represent the most cost-beneficial expenditure associated with the entire test effort.
- Early detection of defects helps to prevent the migration of errors from requirement specification to design, and from design into code.
- More recently, the test effort needs to permeate all steps of the system development life cycle (in the past, at the end of the system development life cycle).

Test Process Analysis (27)

Test Strategies/Defect Prevention Strategies

Examination of Constraints

- To verify that the product is feasible and testable.
- A careful examination of goals and constraints => an appropriate set of test strategies is selected => predictable, higher-quality outcome and support a high degree of automation.
- Potential constraints:
 - A short time-to-market schedule.
 - The limited availability of engineering resources.
- Other constraints may reflect the fact that a new design process or test tool is being introduced.

Test Process Analysis (28)

Test Strategies/Defect Prevention Strategies

Early Test Involvement

- Test team involvement is particularly critical during the requirements phase.
- The Standish Group estimates:
 - ~40% of all software projects fail.
 - ~33% of projects are completed late, over budget, or with reduced functionality.
 - Only 27% of all software projects are successful.

=> User involvement, clear business objectives, and well-organized requirements

=> factors for a successful project creation. Requirements-related management issues account for 45% of successful project development.

Test Process Analysis (29)

Test Strategies/Defect Prevention Strategies

Use of Standards

Standards provide the rules or guidance governing the interaction of the project personnel.

- Facilitates the detection of defects.
- Improves the maintainability of a software application.

=> Standards for software design, program coding, and graphical user interfaces. In some companies, the test team may take the lead in ensuring that the developers follow software design standards and be responsible for promulgating standards for the performance of testing.

Test Process Analysis (30)

Test Strategies/Defect Prevention Strategies

Inspections and Walkthroughs

- Provide for a formal evaluation of software requirements, design, code, and other software work products, such as test procedures and automated test scripts.

=> Inspections are intended to detect defects, violations of development standards, test procedure issues, and other problems.

=> Walkthroughs address the same work products as inspections, but perform a more cursory review.

Test Process Analysis (31)

Test Strategies/Defect Prevention Strategies

Inspections and Walkthroughs benefits:

- => Support the detection and removal of defects early.
- => Prevent the migration of defects to later phases of software development.
- => Improve quality and productivity.
- => Reduce cost, cycle time, and maintenance efforts.

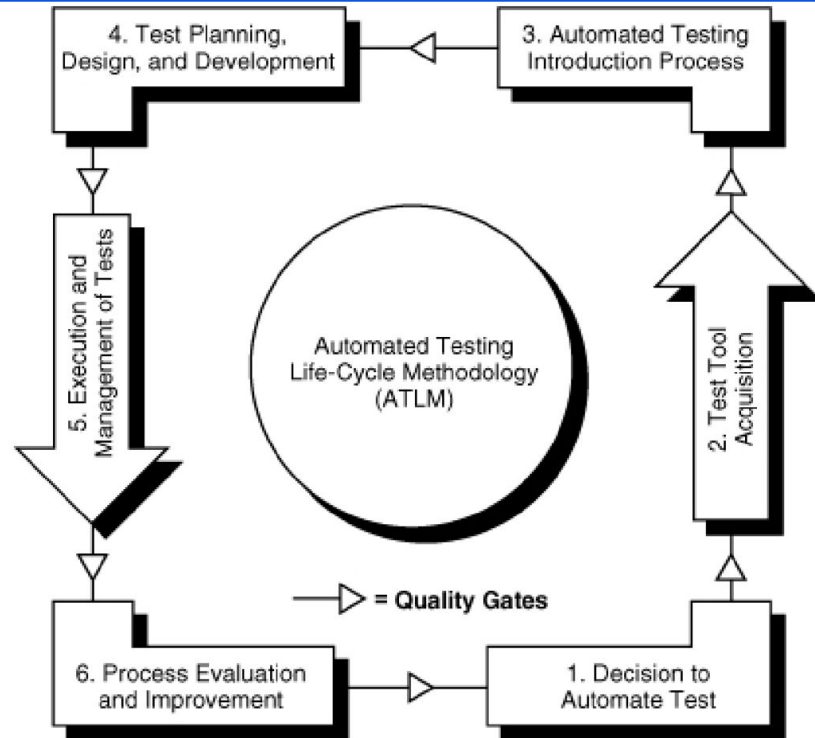
Test Process Analysis (31)

Test Strategies/Defect Prevention Strategies

Quality Gates

Successful completion of the activities prescribed by the test process (such as walkthroughs and inspections) should be the only approved gateway to the next phase of software development.

=>The test team needs to verify that the output of any one stage.



Test Process Analysis (32)

Test Strategies

Defect Detection Technologies

=> Best applied in combination with defect prevention technologies.

- *Inspection and walkthroughs*
- *Quality gates*
- *Testing of product deliverables*
- *Designing testability into the application*
- *Use of automated test tools*

Test Process Analysis (33)

Test Strategies

Defect Detection Technologies

- **Unit test phase:** Condition coverage, path coverage, fault insertion, memory leak, error handling, string test, statement coverage, decision coverage, cyclomatic complexity, data flow coverage.
- **Integration test phase:** Integration testing
- **System test phase:** Stress, regression, replication, data integrity, configuration, performance, functionality, security, alpha/beta, acceptance, compatibility/conversion, benchmark, usability, error guessing, backup and recoverability, operational readiness, random.

Test Process Analysis (34)

Test Strategies

Defect Detection Technologies

- *Acceptance test phase*
- *Following a test process*
- *Risk assessment*
- *Strategic manual and automated test design*
- *Execution and management of automated test*
- *Test verification method*
- *User involvement*

Test Process Analysis (35)

Test Strategies

Defect Detection Technologies

- *Acceptance test phase*
- *Following a test process*
- *Risk assessment*
- *Strategic manual and automated test design*
- *Execution and management of automated test*
- *Test verification method*
- *User involvement*

Test Process Analysis (36)

Test Strategies/Defect Detection Technologies

Inspections and Walkthroughs

- Concentrate on eliminating errors after program coding has begun.
- A **code walkthrough** consists of a top-level or cursory review of program code that the code complies with relevant standards (checklist).
- A **code inspection** examines the code in more detail.
- A **test procedure walkthrough** is performed at a cursory level.
- A **test procedure inspection** involves a more detailed examination of test procedures and test scripts.

Test Process Analysis (37)

Test Strategies/Defect Detection Technologies

Inspections and Walkthroughs

The defined roles for the code inspection are as follows:

- **Moderator:** The moderator distributes the program code before the actual meeting, schedules the meeting, leads the meeting, records the meeting minutes, and follows up on any resulting actions.
- **Programmer:** Each programmer is responsible for narrating his or her code from beginning to end.
- **Test Lead:** The test lead should come prepared with several test procedures to talk through with the group.
- **Peers:** Programmers help review code and provide objective feedback.

Test Process Analysis (38)

Test Strategies/Defect Detection Technologies

Testing Product Deliverables

- Product deliverables may include requirement specifications, design documentation, test plans, training material, user manuals, system help manuals, system administration manuals, and system implementation plans.
- Product deliverables may consist of hard-copy documents or on-line system documentation.

=> The test team should review product deliverables and document suspected errors in isolation or in comparison with other product deliverables.

Test Process Analysis (39)

Test Strategies/Defect Detection Technologies

Designing Testability into the Application

The ***IEEE Standard Glossary of Software Engineering Terminology*** defines “testability” as:

1. The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.
2. The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met.

Test Process Analysis (40)

Test Strategies/Defect Detection Technologies

Designing Testability into the Application

1. Testability must be factored into development strategies as well.
2. Application partitioning is another development strategy that incorporates testability => applications are separated into several layers.
3. Another testability issue to consider is test machines and end-user machines to be identical.

Test Process Analysis (41)

Test Strategies/Defect Detection Technologies

Use of Automated Test Tools

Automated test tools can be applied to support the defect detection process.

- If the organization hopes to meet today's challenging schedules and adequately test its increasingly complex applications.
- Test tools are being applied to generate test data, catalog the tests in an organized fashion, execute tests, store test results, and analyze data.
- Project schedule pressures are another reason why the use of automated test tools should be viewed.

Test Process Analysis (42)

Test Strategies/Defect Detection Technologies

Traditional Testing Phases

In the book, ***Software System Testing and Quality Assurance***, Boris Beizer identifies three phases of testing:

- **Unit testing:** testing of the smallest unit or block of program code.
- **Integration testing:** is to verify that each software unit interfaces correctly with other software units.
- **System testing:** seeks to test all implementation aspects of the system design.

Test Process Analysis (43)

Test Strategies/Defect Detection Technologies

Adherence to a Test Process

- Ensure that the requisite activities of an effective test program are properly exercised.
- Possible to identify and refine the defined process, thereby permitting continual process improvement.

Test Process Analysis (44)

Test Strategies/Defect Detection Technologies

Risk Assessment

- The test engineer identifies the parts of a project that pose the greatest risk and the functionality that is most likely to cause problems.
- The test team therefore needs to weigh the risk that system requirements cannot be successfully supported.
- Should include a determination of the probability that a defined risk will happen and an estimate of the impact.
- Risk mitigation strategies should be defined.

Test Process Analysis (45)

Test Strategies/Defect Detection Technologies

Strategic Manual and Automated Test Design

- Tests are designed to incorporate reusability and maintainability, defects can be repeatedly identified.
- Eliminating defects within the most critical component of the software application.

=> In either case, the most critical component should be as close to error-free as possible.

Test Process Analysis (46)

Test Strategies/Defect Detection Technologies

Development of Automated Tests

If tests are to be uniform, repeatable, maintainable, and effective, the test team must follow its test development guidelines.

Test Process Analysis (47)

Test Strategies/Defect Detection Technologies

Execution and Management of Automated Tests

- Often the project schedule is tight and limited project resources.
- The end result should be a software application product that operates correctly and performs in accordance with defined requirements.

Test Process Analysis (48)

Test Strategies/Defect Detection Technologies

Test Verification Method

- A test qualification method is employed to verify that the application satisfies all system requirements.
- This method involves the creation of a test verification summary matrix.
- Verification methods include demonstration, analysis, inspection, and test.

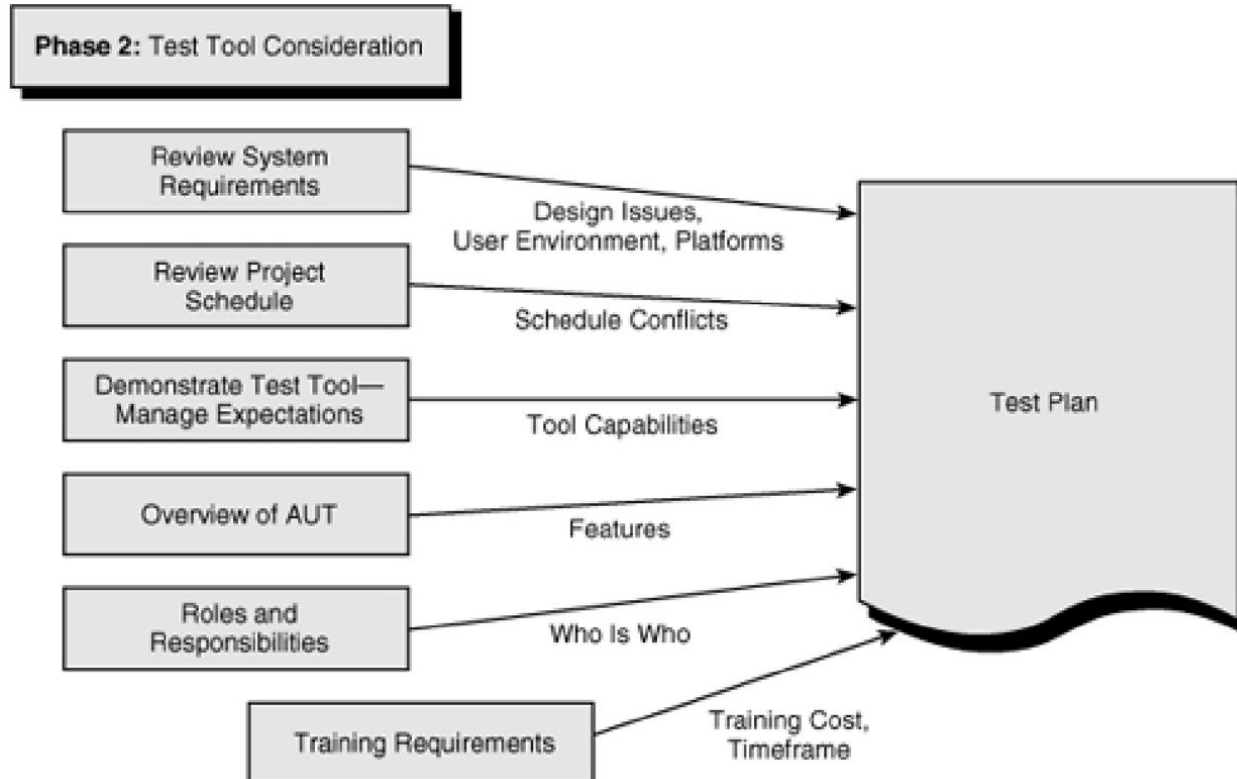
Test Process Analysis (48)

Test Strategies/Defect Detection Technologies

User Involvement

- Test team interaction with prospective end users => ensuring that defect prevention or defect detection becomes incorporated into the test process.
- The test team needs to obtain end-user or customer buy-in for the test plan
=> satisfy the end user's system functionality and performance concerns.

Test Tool Consideration (1)



Test Tool Consideration (2)

Review of Project-Specific System Requirements

- The test engineer needs to understand the requirements for the AUT.
- The test team must understand the user environment, computing platforms, and product features of the application-under-test, collectively referred to as the system architecture (DBMS, network...).
- Analyze and provide a preliminary determination (test tool compatibility...).
- Understand of the primary business tasks addressed by the application.
- Determine the types of database activity.
- Should identify the operational profile and user characteristics of the application-under-test.

Test Tool Consideration (3)

Application-Under-Test Overview

- This overview may consist of a system prototype when only parts of the system application are available.
- Technical aspects of the application now need to be explored.
- Determine which section or part can be supported with an automated test tool (divide and conquer the system requirements).
- Determine which automated test tool can be used for which system requirement or section of the application.

Test Tool Consideration (4)

Review of Project Schedule

- Early introduction of automated test assures that lead time is adequate for the test team to become familiar with the tool and its advanced features.
- A decision to incorporate automated testing at the end of the development life cycle carries risks.
- A majority of the test effort will involve test planning, design, and development.
- Whether the project schedule permits the utilization of an automated test tool and the performance of an automated testing process?

Test Tool Consideration (5)

Test Tool Compatibility Check

Verify that the resulting tool is still compatible with the application by conducting the following tests:

1. Presentation layer calling the database server
2. Presentation layer calling the functionality server
3. End-to-end testing (presentation calling functionality server, calling database server)
4. Functionality server calling another functionality server
5. Functionality server calling the database server

=> access to the application's internal workings, such as hidden APIs and protocols?

Test Tool Consideration (6)

Demonstration of the Tool to the Project Team

- Provide project personnel with a mental framework.
- Avoid false expectations.
- Avoid wrong assumption for managers.

=> It is important that members of the project team understand both the capabilities and the limitations of the test tool.

Test Tool Consideration (7)

Test Tool Support Profile

- Who have sufficient experience with the tool to plan, prepare, and execute testing?
- Whether any individual on the test team has enough experience with the tool to leverage the tool's more advanced features?
- The test team manager should map out a staffing profile of each of the team members to ascertain the team's strengths and weaknesses.

Test Tool Consideration (8)

Test Tool Support Profile

| | Test Tool Experience | Advanced Tool Training | Advanced Tool Experience | |
|--|----------------------------|------------------------------|--------------------------------|---|
| Test Team Manager | | | | |
| Skills include: Rational's TestStudio, Purify, WinRunner, MS Project, MS Access, UNIX, C, SQL, Oracle | ✓ | | | |
| Test Team Lead | | | | |
| Skills include: QA Partner, Rational's Performance Studio, Visual Basic, MS Access, UNIX, C, SQL, Sybase | ✓ | ✓ | ✓ | |
| Test Engineer 1 | | | | |
| Skills include: C, C++, Visual Basic, MS Access, MVS, COBOL, Fortran | | | | |
| Test Engineer 2 | | | | |
| Skills include: C, UNIX, HTML, Java, Powerbuilder, Novell Netware | | | | |
| Test Engineer 3 | | | | |
| Skills include: Visual Test, C, C++, PVCS, Visual Basic, MS Access, SQL, Oracle | | | | ✓ |
| Test Engineer 4 | | | | |
| Skills include: Visual Test, C, C++, Visual Basic, MS Access | | | | ✓ |

Test Tool Consideration (9)

Review of Training Requirements

- The need to incorporate training into the project schedule.
- It is important to develop test process expertise among those individuals involved in software test.
- Test team members should develop expertise in one or more automated test tools through training, knowledge transfer.

Chapter summary (1)

- How test teams introduce an automated software test tool into a new project is nearly as important as the selection of the appropriate test tool for the project.
- The purpose of analyzing the organization's test process, which is required as part of the test analysis phase, is to ascertain the test goals, objectives, and strategies that may be inherent in the test process.
- Test process analysis documentation is generated through the test team's process review and its analysis of test goals and objectives.

Chapter summary (2)

- Test strategies can be classified into two different categories: defect prevention technologies and defect detection technologies.
- Defect prevention methodologies cannot always prevent defects from entering into the application-under-test, because applications are very complex and it is impossible to catch all errors. Defect detection techniques complement defect prevention efforts, and the two methodologies work hand in hand to increase the probability that the test team will meet its defined test goals and objectives.

Chapter summary (3)

- Unit testing is often performed by the developer of the unit or module. This approach may be problematic because the developer may not have an objective viewpoint when testing his or her own product.
- The test strategies that apply to a particular project depend on the application development environment as well as the test objectives and requirements.
- The test team manager must determine whether the test team has sufficient skills to support the adoption of the automated test tool.

Chapter summary (4)

- One of the most significant steps in the test tool consideration phase requires that the test team decide whether the project schedule will permit the appropriate utilization of an automated test tool and whether an automated testing process can offer value under the particular circumstances.
- The compatibility check is intended to ensure that the application will work with the automated testing tool and, where problems exist, to investigate work-around solutions.

References (1)

- [1] Kuhn, T. *The Structure of Scientific Revolutions, Foundations of the Unity of Science*, Vol. II. Chicago: University of Chicago, 1970.
- [2] Florac, W.A., Park, R.E., Carleton, A.D. *Practical Software Measurement: Measuring for Process Management and Improvement*. Guidebook CMU/SEI-97-HB-003. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, April 1997.
- [3] Burnstein, I., Suwanassart, T., Carlson, C.R. *Developing a Testing Maturity Model*, Part II. Chicago: Illinois Institute of Technology, 1996.
- [4] Standish Group. <http://www.standishgroup.com/chaos.html>.

References (2)

- [5] Voas, J., and Miller, K. “*Software Testability: The New Verification*,” IEEE Software, page 3. May 1995. <http://www.rstcorp.com/papers/chrono-1995.html>
- [6] Corporate Computing Inc. *Corporate Computing’s Top Ten Performance Modeling Tips*. Monroe, LA, 1994.
- [7] Corporate Computing Inc. *Corporate Computing’s Test to Evaluate Client/Server Expertise*. Monroe, LA, 1994.