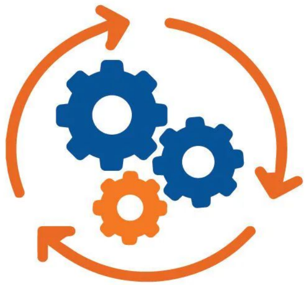
Chapter 2

Decision to Automate Test



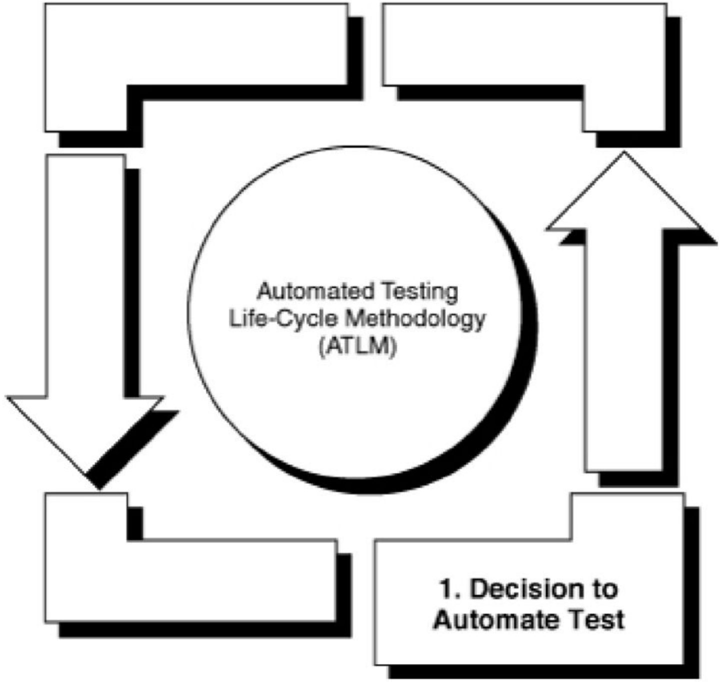
Lecturer: Nguyen Thanh Quan (MSc)
Email: tg_nguyenthanhquan_cntt@tdtu.edu.vn

Content

1. **Overcoming False Expectations for Automated Testing**
2. **Benefits of Automated Testing**
3. **Acquiring Management Support**

Introduction

“If you want a high quality software system, you must ensure each of its parts is of high quality.” - —Watts Humphrey.



Situation (1)

An organization has determined that its current testing program is **not effective**.

Analysis conducted, the outcome has shown that the current manual testing process **requires improvement**.

Looking for a more **repeatable** and **less error-prone testing approach.**

An improvement analysis determines that **automated testing** should be **introduced**.

Situation (2)

The organization’s **test lead** is **informed** about **automated testing** introduction, but a pilot project remains to be identified.

Does the **application** being developed as part of the current project lend itself to **automation**?

The test lead **gathers information** regarding automated testing.

The test lead is **not sure where to begin**.

Chapter’s goal

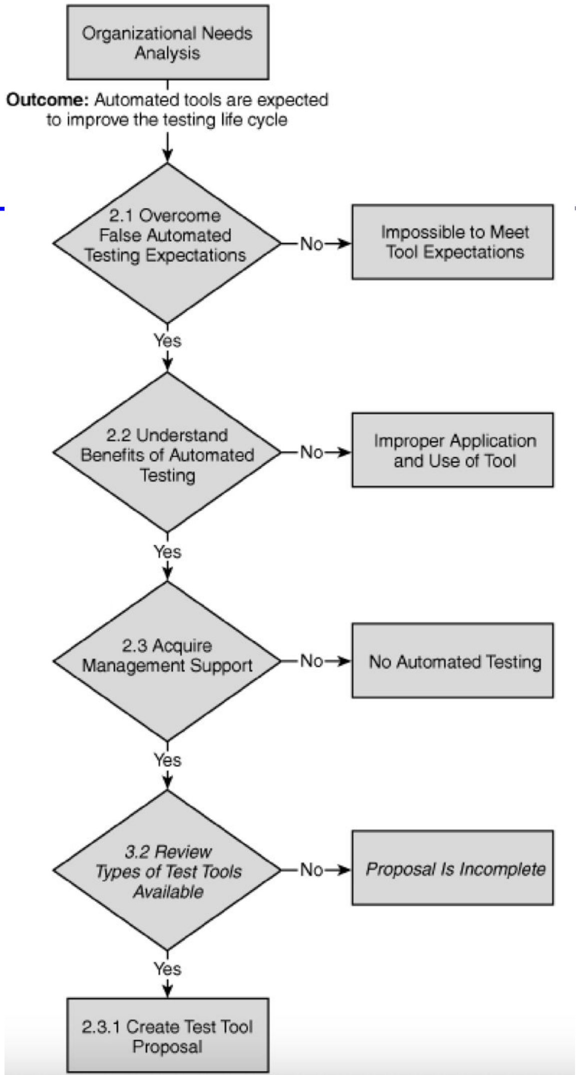
Outlines a structured way of approaching the decision to automate test.

Depicts this step-by-step methodology.

Should the process continue or should it terminate with a decision not to automate test for that particular project?

Automated Test Decision Process

- Which automated **test tool** should be used?
- How can **management** be **convinced** that automated testing is or is **not beneficial** to this project?
- How would the **test engineer** go about introducing a **new concept** such as automated testing?
- How can the **test engineer** determine whether the **application** lends itself to **automated testing**?


Organizational Needs Analysis
Outcome: Automated tools are expected to improve the testing life cycle
2.1 Overcome False Automated Testing Expectations
No
Impossible to Meet Tool Expectations
Yes
2.2 Understand Benefits of Automated Testing
No
Improper Application and Use of Tool
Yes
2.3 Acquire Management Support
No
No Automated Testing
Yes
3.2 Review Types of Test Tools Available
No
Proposal Is Incomplete
Yes
2.3.1 Create Test Tool Proposal

Overcoming False Expectations for Automated Testing

Overcoming False Expectations for Automated Testing

Act as a test lead, the project manager requests you to submit within the next week a recommendation for the specific test tool required plus a cost estimate for its procurement

Some false expectations:

1. Automated test tool should be able to **accomplish everything** from test planning to test execution, without any manual intervention.
2. A single test tool can **support all test requirements**, regardless of environment parameters, such as the operating system or programming language used.
3. An automated test tool will **immediately reduce the test effort and the test schedule.**

Automatic Test Plan Generation

Currently, no commercially available tool can automatically create a comprehensive test plan

- Test engineer will be asked to give an **overview** of test tool functionality to a senior manager or a small number of managers.
- The audience may be people who enthusiastic about automated testing, but who are **not aware** of the **complexity**.
- The managers may have obtained third-hand information and have reached the **wrong conclusions** about capabilities of automated test tools.

Automatic Test Plan Generation

Currently, no commercially available tool can automatically create a comprehensive test plan

- **Automated test** tools should be **viewed as enhancements** to manual testing
  - Not automatically develop the test plan, design and create the test procedures, and execute the test procedures.
- The term automated test tool seems to bring out a great deal of wishful thinking that is **not closely aligned with reality.**
  - Not replace the human factor necessary for testing a product.
  - A test tool can be viewed as an additional part of the machinery that supports the release of a good product.

Test Tool Fits All

Currently, not one single test tool exists that can be used to support all operating system environments

- A single test tool will **not fulfill all testing requirements** for most organizations.
    - mainframe computers and Sun workstations
    - operating systems such as MVS, UNIX, Windows 3.1, Windows 95, and Windows NT
    - programming languages such as COBOL, C, C++, MS Access, and Visual Basic
    - other client-server technologies; and Web technologies.

➔ The **test lead** must make it **clear**. **More than one tool** is required to test the **various technologies.**

Immediate Test Effort Reduction

Introduction of automated test tools will not immediately reduce the test effort

- Test tool as part of a project is to **reduce** the test **effort**.
- Aware that test effort savings **do not necessarily come immediately**.
- The introduction of an automated test tool to a new project adds a whole **new level of complexity** to the test program.
- Automated testing also requires **careful analysis** of the target application to determine which sections of the application are amenable to automation.
- Requires that the team **pay careful attention** to automated test procedure design and development.
- **Viewed** as a **mini-development life cycle**, complete with the planning and coordination issues that come along with any development effort.

Immediate Schedule Reduction

The use of an automated testing tool on a new project will immediately minimize the test schedule

- The testing **effort** may actually **increase**, so the testing schedule will not experience the anticipated decrease at first.
- The **current testing** process must be **augmented** or an entirely new testing process must be developed and implemented.
- Become **familiar** with the new **automated testing process** (that is, ATLM) and learn to follow it.

➔ Testing process **established** and **effectively implemented** results in **productivity** and **turn-around time** that have a positive effect on schedule and cost.

Tool Ease of Use

An automated tool requires new skills, so additional training is required

- Vendors advertise the tools' **ease of use.**
- Vendors **deny** that any **learning curve** is **associated** with the **use of a new tool.**
- In fact, efficient automation is **not simple.**
- **Require** tool **scripting knowledge**, so as to make the scripts robust, reusable, and maintainable.
- The test engineer must be **trained on the tool** and the tool's built-in scripting language.
- New **training requirements** and a learning curve can be expected with the use of any new tool.

Universal Application of Test Automation (1)

It can’t be expected that all of the tests on a project can be automated

- The **performance** of **compatibility** tests is especially **important** for **GUI test tools**, because such tools have **difficulty recognizing** some custom control **features** within the application.
- Usually **written by third parties**, and most test tool manufacturers **cannot keep up with** the **hundreds of clever controls** churned out by the **various companies.**

➔ The **test engineer** must then **decide** whether to **automate** this part of the application by **finding work-around solution or** to **test** this control **manually** only.

Universal Application of Test Automation (2)

It can’t be expected that all of the tests on a project can be automated

- For example: **Printout** - the test engineer can automatically send a document to the printer, but verify the results by reading the printout.
- The test team will **not have enough time or resources** to support 100% test automation of an entire application. For example:
  - Impossible to test all inputs or all combinations and permutations of all inputs.
  - Impossible to test exhaustively all paths.

➔ **Impossible** to approach the test effort for the **entire application-under-test** with the goal of **testing 100%** of the entire software application.

Universal Application of Test Automation (3)

It can’t be expected that all of the tests on a project can be automated

- **Cost** is another limiting factor. A test that is executed only once is often not worth automating. For instance:
  - An end-of-year report for a health claim system.

➔ Must evaluate the value or payoff for investing the time in developing an automated script.

➔ Perform a careful analysis of the application.

➔ Must also weed out redundant tests.

One Hundred Percent Test Coverage

It is impossible to perform a 100% test of all possible simple inputs to a system

- The function must be **tested** with all possible data - both **valid** and **invalid**.
- Automated testing may increase the **breadth** and **depth** of test coverage.
- There still will **not be enough time** or resources to perform a 100% exhaustive test.
- For example, verification of a user password.
  - Constraint: six to eight characters long, where each character is an uppercase letter or a digit. Each must contain at least one digit.
  - **2,684,483,063,360** possible variations of passwords exist (Kenneth H. Rosen). Assume **60 test procedures per hour** ⇔ take **155 years** to prepare and execute a complete test!!

One Hundred Percent Test Coverage

It is impossible to exhaustively test every combination of a system

Consider the test of the telephone system in North America.

Constraint: a three-digit area code, a three-digit office code, and a four-digit station code.

- Let X denote a digit that can take any of the values of 0 - 9.
- Let N denote a digit that can take any of the values of 2 - 9.

➔ Format: NXX, NXX, and XXXX.

➔ There are 8 × 8 × 10 = 640 office codes with format NNX.

➔ 8 × 10 × 10 = 800 with format NXX.

➔ 10 × 10 × 10 × 10 = 10,000 station codes with format XXXX.

➔ 800 × 800 × 10,000 = 6,400,000,000 different numbers available.

Benefits of Automated Testing

Benefits of Automated Testing

Automated testing can provide several benefits when it is implemented correctly and follows a rigorous process

1. Production of a **reliable** system.
2. **Improvement** of the **quality** of the test effort.
3. **Reduction** of the test **effort** and **minimization** of the **schedule**.

Production of a Reliable System

A strategic goal of the test effort is to

- Find defects. Minimizing errors. Little downtime.
- Ensure the system’s performance requirements meet or exceed user expectations.
- Improve all areas of testing, including test procedure development, test execution, test results analysis, error status/correction monitoring, and report creation.
- Support all test phases including unit, integration, regression, system, user acceptance, performance, and stress and configuration testing, among others.
- Help to build reliable systems.

Production of a Reliable System

1. **Improved Requirements Definition**
2. **Improved Performance Testing**
3. **Improved Load/Stress Testing**
4. **Quality Measurements and Test Optimization**
5. **Improved partnership with development team**
6. **Improved software development life cycle**

Improved Requirements Definition

- If **requirements** are **unambiguous** and consistently delineate all of the information that a **test engineer needs in a testable form**, the **requirements** are said to be t**est-ready or testable.**
- Test-ready requirements **minimize** the test **effort** and **cost**.
- Requirements that are in test-ready condition help **support** the preparation of an **efficient test design** and **requirements to test design/test procedure traceability.**

Improved Performance Testing

- The objective of performance testing is to demonstrate that a **system functions** in accordance with its performance **requirement specifications** (acceptable response times, required number of transactions, production-size database).
- Current load-testing tools are on the market that allow the test engineer to perform tests of the system functionality automatically, producing timing numbers, graphs, pinpointing the bottlenecks and thresholds of the system.
- Test engineers need **a test script** to capture the performance statistics, stay away from stopwatches.
- Use **tool-generated data**, **permit virtual user testing**, **simulate** tens, hundreds, or even thousands of users.

Improved Stress Testing

In stress testing, the system is subjected to extreme and maximum loads to find out whether and where the system breaks and to identify what breaks first

- Identify the **weak points** of the system.
- Define **thresholds** and describe the system's response to an overload.
- With an automated stress-testing tool, the test engineer can instruct the tool **when** to execute a stress test, **which** tests to run, and **how** many users to simulate all without user intervention.
- It is **expensive, difficult, inaccurate, and time-consuming** to stress test an application adequately using purely manual methods.
- Typical **types of errors** uncovered by stress testing include memory leakage, performance problems, locking problems, concurrency problems, excess consumption of system resources, and exhaustion of disk space.

Quality Measurements and Test Optimization (1)

Automated testing will produce quality metrics and allow for test optimization

- Test engineers' analysis of quality measurements supports the effort of optimizing tests, but only when tests are repeatable.
- The automated tool produces many metrics, usually creates a test log, and can accommodate automated metrics reporting.
- Automated testing also supports optimization.

Quality Measurements and Test Optimization (2)

Automated testing will produce quality metrics and allow for test optimization

A test engineer can optimize a regression test suite by performing the steps

1. Run the regression test set.
2. If cases are discovered for which the regression test set ran acceptably, but errors surface later, the test procedures that uncovered those bugs in the regression test set can be identified.
3. By repeating these steps, the regression test suite of scripts is continuously optimized using quality measurements.

Improved Partnership with Development Team

Automated testing will produce quality metrics and allow for test optimization

- Automated testing provides a convenient way for the test engineer and the application developer to work together.
- The test engineer now needs to have **similar software skills**, more opportunities will arise for **collaboration** and for mutual respect.
- Testers need to have the **same qualifications** and the same career opportunities (remuneration, appreciation) as developers.
- The relationship will be viewed more as a **partnership**.

Improved System Development Life Cycle

Automated testing can support each phase of the system development life cycle

- Tools exist for the requirements definition phase that help produce test-ready requirements so as to minimize the test effort and cost of testing.
- Tools supporting the design phase, such as modeling tools, can record the requirements within use cases.
- Tools also exist for the programming phase.
- Tools are available that can support the test effort.

Improvement of the Quality of the Test Effort

1. Improved build verification testing (smoke testing)
2. Improved regression testing
3. Improved multi platform compatibility testing
4. Improved software compatibility testing
5. Improved execution of mundane tests
6. Improved focus on advanced test issues
7. Execution of tests that manual testing can’t accomplish
8. Ability to reproduce software defects
9. Enhancement of business expertise
10. After-hours testing

Improved Build Verification Testing (Smoke Test)

The smoke test (build verification test) focuses on test automation of the system components that make up the most important functionality

- The automated test tool supports the smoke test by allowing the test engineer to **play back the script** in software build/version verification.
- Smoke testing ensures that **no effort is wasted** in trying to test an incomplete build.
- A smoke test can **save developers, configuration management personnel, business users, and test engineers much valuable time.**

Improved Regression Testing

A regression test is a test or set of tests that is executed on a baselined system or product when a part of the total system product environment is modified

- To verify that the functions provided by the **modified system or product match the specifications** and that **no unintended change** in operational functions has been made.
- Regression testing should occur **after each release** of a previously tested application.
- Regression testing may **include test procedures** that have the highest probability of detecting the most errors.
- Should be **performed via an automated tool** because it is usually lengthy and tedious and thus prone to human error.

Improved Multi Platform Compatibility Testing

Changes in computer hardware, network versions, and operating systems can cause unexpected compatibility problems with the existing configuration

- When the test engineer creates the test scripts for an application-under-test on a Windows 95 platform, he or she can simply execute the same test scripts on the Windows NT platform, using multi-platform-compatible tools.
    - Rational’s TestStudio
    - AutoScriptor Inferno.

Improved Software Configuration Testing

The same principle that drives multi platform compatibility testing applies to software configuration testing

- Software changes (upgrade, library changes etc) can cause unexpected compatibility problems with existing software.
- The execution of automated test scripts ensures that these software changes did not adversely affect current applications and operating environments.

Improved Execution of Mundane Tests

An automated test tool will eliminate the monotony of repetitious testing

- The same steps were repeated over and over again, with the only change from one to the other being the kind of operation (add, delete, update) that was being performed.
- An end user performing acceptance testing would have become tired very quickly.
- A test script does not care whether it has to execute the same monotonous steps over and over again and can automatically validate the results.

Improved Focus on Advanced Test Issues

Automated testing allows for simple repeatability of tests

- With both manual and automated regression testing, the test teams repeatedly expend effort in redoing the same basic operability tests.
- Automated testing presents the opportunity to move on more quickly and to perform a more comprehensive test within the schedule allowed.

Execution of Tests That Manual Testing Can’t Accomplish

Software systems and products are becoming more complex, and sometimes manual testing is not capable of supporting all desired tests

- It would require many man-hours to produce the cyclomatic complexity of the code for any large application.
- Manual test methods employed to perform memory leakage tests would be nearly impossible.

Ability to Reproduce Software Defects

Automated testing eliminates the problem that a tester cannot recall the steps exercised that led to the error

- With an automated test tool, the manual steps taken to create a test are recorded and stored in a test script.
- The script will play back the exact same sequence of steps that were initially performed.
- The test engineer can inform the appropriate developer about the defect.

Enhancement of Business Expertise

- The new business user is able to play back the business test scripts that the expert had created.
- The use of these test scripts allows the test team to verify that the original functionality still behaves in the correct manner.
- It prevents the test team from worrying about the fact that the resident expert left.
- The new business user can learn the business functionality of the application-under-test by watching the script play back.

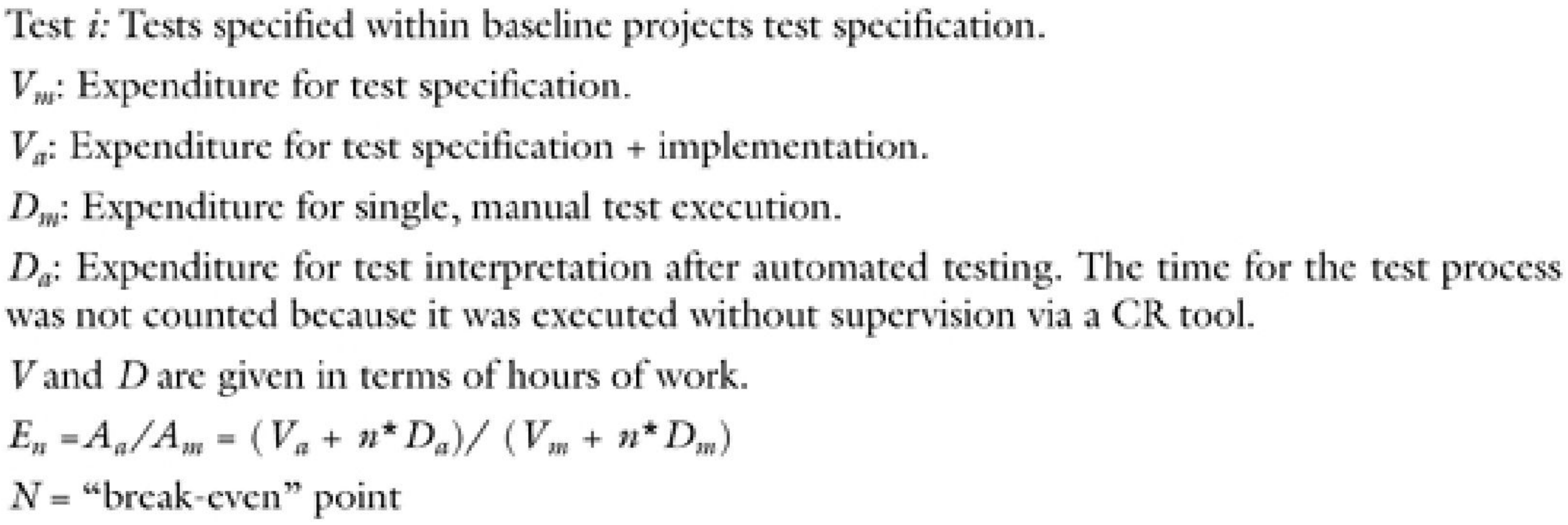
After-Hours Testing

- Automated testing allows for simple repeatability of tests.
- Automated testing allows for after-hours testing without any user interaction.
- Initiating tests at silent/leisure/meeting times makes maximum use of the test lab and time.

Reduction of Test Effort and Minimization of Schedule

Test steps (Manual Versus Automated Testing conducted by the Quality Assurance Institute in November 1995.)	Manual testing (hrs)	Automated testing (hrs)	Percentage improvement with tools
Test plan development	32	40	-25%
Test procedure development	262	117	55%
Test execution	466	23	95%
Test result analysis	117	58	50%
Error status/correction monitoring	117	23	80%
Report creation	96	16	83%
Total duration	1090	277	75%

Reduction of Test Effort and Minimization of Schedule



Test *i*: Tests specified within baseline projects test specification.

V_m: Expenditure for test specification.

V_a: Expenditure for test specification + implementation.

D_m: Expenditure for single, manual test execution.

D_a: Expenditure for test interpretation after automated testing. The time for the test process was not counted because it was executed without supervision via a CR tool.

V and D are given in terms of hours of work.

$E_n = A_a / A_m = (V_a + n * D_a) / (V_m + n * D_m)$

N = "break-even" point

Acquiring Management Support

Acquiring Management Support (1)

Whenever an organization tries to adopt a new technology, it faces a significant effort to determine how to apply the technology to its needs

- The challenge is how to make the best case for implementation of a new test automation technology to the management team.
- Test engineers need to influence management's expectations for the use of automated testing on projects.
- Test personnel need to convince management about the potential return on investment by conducting cost-benefit analysis.
- The test team will need to point out the risks involved and may need to reconsider a recommendation to automate testing.

Acquiring Management Support (2)

Whenever an organization tries to adopt a new technology, it faces a significant effort to determine how to apply the technology to its needs

- What does the test team intend to accomplish, and which need will be met, by using automated testing tools?
- Management needs to be made aware of the additional cost:
  - the tool purchase itself.
  - the initial schedule/cost increase.
  - additional training costs.
  - costs for enhancing an existing testing process or new testing process implementation.

Acquiring Management Support (3)

The potential obstacles that organizations must overcome when adopting automated test systems include the following:

- Finding and hiring test tool experts.
- Using the correct tool for the task at hand.
- Developing and implementing an automated testing process, which includes developing automated test design and development standards.
- Analyzing various applications to determine which are best suited for automation.
- Analyzing the test requirements to determine which are suitable for automation.
- Training the test team on the automated testing process, including automated test design, development, and execution.
- Dealing with the initial increase in schedule and cost.

Test Tool Proposal

- The test tool proposal needs to **convince management** that a positive cost benefit is associated with the purchase of an automated test tool.
- A test tool proposal helps to **outline in detail the cost** of test tool procurement and training requirements.
- The proposal helps **document plan phases.**
- This helps document the need for a **phased buildup** of a test engineering staff as well as the desired skills sought for the bolstered test team.
- The test team's responsibility to **define the test tool requirements** of the organization and provide an associated cost estimate.
- The associated project teams have the r**equisite skills** to successfully utilize the automated test tool.

Test Tool Proposal

1. Estimated Improvement Opportunities
2. Criteria for Selecting the Correct Tool
3. Tool Cost Estimate
4. Additional to introduce tool
5. Tool expertise
6. Too training cost
7. Tool evaluation domain
8. Tool rollout process

Estimated Improvement Opportunities

- The organization may require one or more automated test tools, with each test tool having its own features and strengths.
- Management needs to be well aware of the functionality and value of each test tool.
- A list of tool benefits needs to be provided within the proposal.

Criteria for Selecting the Correct Tool

The return on investment obtained by using an automated test tool largely depends on the appropriate selection of a test tool

- The challenge for the test engineer is to select the best test tool for the organization and/or the particular test effort and to understand what types of tools are available that could meet these needs.
- Automated test tools can vary from ones having simple functionality to those offering complex functionality, and their performance can vary from mediocre to excellent.
- A test tool with only minimal functionality will often cost less than a tool with extensive functionality.

Criteria for Selecting the Correct Tool

The criteria definition guidelines are given below

- Gather third-party input from management, staff, and customers regarding tool needs.
- Select tool criteria to reflect the organization’s system engineering environment.
- Specify tool criteria based on long-term investment assumptions.
- Ensure that the tool will be usable in many testing phases.

➔ researching and evaluating the various tools will require personnel resources, time, and money.

Tool Cost Estimate (1)

- It may be necessary to outline a phased implementation of the test tools so that costs can be spread over a period of time.
- Whether this cost is in line with management’s expectations.
- A cost-benefit analysis should be conducted, with the test team ensuring that funding is available to support this analysis.
- Costs associated with the implementation of the test tool may include
  - costs necessary to upgrade hardware
  - software maintenance agreements.
  - hotline support.
  - requirements for tool training.

Tool Cost Estimate (2)

What if test team members are unsure of tools:

1. The test team could select a less expensive tool that supports test requirements adequately for the near term.
2. It could outline the cost savings or performance enhancing benefits in a way that convinces management that the tool is worth the upfront investment.
3. It could scale down the implementation of the test tool and plan for additional implementation during the next budget period.

Additional Time to Introduce Tool

A major concern when selecting a test tool focuses on its impact and fit with the project schedule

- Will there be enough time for the necessary people to learn the tool within the constraints of the schedule?
- If there isn't sufficient time to support implementation of a sophisticated tool, can the team deploy an easy-to-use tool?
- A less expensive, easy-to-use tool offers minimal functionality may put at risk the test team's ability to follow up later.
- If no one in the organization uses the tool, the effort to obtain and incorporate it will have been wasted.

Tool Expertise

Many people incorrectly believe that the test team skill set does not need to include technical skills for automated testing tools

- Need to include personnel with technical expertise on the operating system, database management system, network software, hardware device drivers, and development support software, such as configuration and requirements management tools.
- Should have a technical or software development background, ensuring that the features of the automated tool will be exercised sufficiently.
- The test team needs to maintain its manual testing expertise.
- The introduction of a new test tool to the project or organization adds short-term complexity and overhead.

Tool Training Cost

Some test engineers may take the initiative on their own to obtain additional technical training with automated testing tools

- Need to identify each individual who requires training and specify the kind of training necessary.
- Training may be required at different levels.
- The test team should identify organizations that offer the desired training.
- The test team should not depend entirely upon consultants for the execution of the test program.

The automated test tool proposal should list the costs associated with various sources of training and mentoring that will be required for a specific project or organization.

Tool Evaluation Domain

When first implementing a test tool on a high-visibility project, the benefits of success are great - but so is the downside of failure

- As part of the test tool proposal, consideration should be given to the method with which the test tool or tools will be evaluated.
- It may require advanced coordination and approvals from several different managers.

Tool Rollout Process

Once a test tool has successfully passed through the decision the test team needs to execute a plan to roll out the tool to the target project or projects

- Review the test tool, develop simplified implementation procedures, and then teach or mentor members of the test team on the tool's use.
- Experience has shown that the best rollout strategy involves the use of a separate test team to implement the tool.
- The test team might consider organizing a test tool user group within the organization so as to transfer knowledge about the tool.
- If the test tool proposal is accepted and funded by management, the test
- team then needs to obtain permission to proceed with test automation.

Chapter summary (1)

- Provide a structured way of approaching the decision to automate testing.
- Make sure that management understands the appropriate application of automated testing for the specific need.
- How much of the test effort can be supported using an automated test tool.
- The benefits of automated testing (when implemented correctly) such as minimizing test effort, test schedule reduction etc.
- The total test effort required with automated testing and the test effort required with manual methods comparison.
- An automated test tool cannot be expected to support 100% of the test requirements of any given test effort.
- Automated testing may increase the breadth and depth of test coverage.

Chapter summary (2)

- The optimal value of test automation is obtained through the proper match of a test tool with the technical environment and the successful application of the Automated Test Life-cycle Methodology (ATLM).
- The test tool proposal may be especially helpful in persuading management to set aside future budget dollars for test tool support.

References

1. Rosen, K.H. *Discrete Mathematics and Its Application*, 2nd ed. New York: McGraw-Hill, 1991.
2. Poston, R. *A Guided Tour of Software Testing Tools*. San Francisco: Aonix, 1988. www.aonix.com.
3. Myers, G.J. *The Art of Software Testing*. New York: John Wiley and Sons, 1979.
4. RTCA. “Software Considerations in Airborne Systems and Equipment Certification.” Document No. RTCA/DO-178B, prepared by: SC-167. December 1, 1992.
5. Quality Assurance Institute. *QA Quest*. November 1995. See http://www.qaiusa.com/journal.html
6. Linz, T., Daigl, M. *GUI Testing Made Painless. Implementation and Results of the ESSI Project Number* 24306. 1998. www.imbus.de.