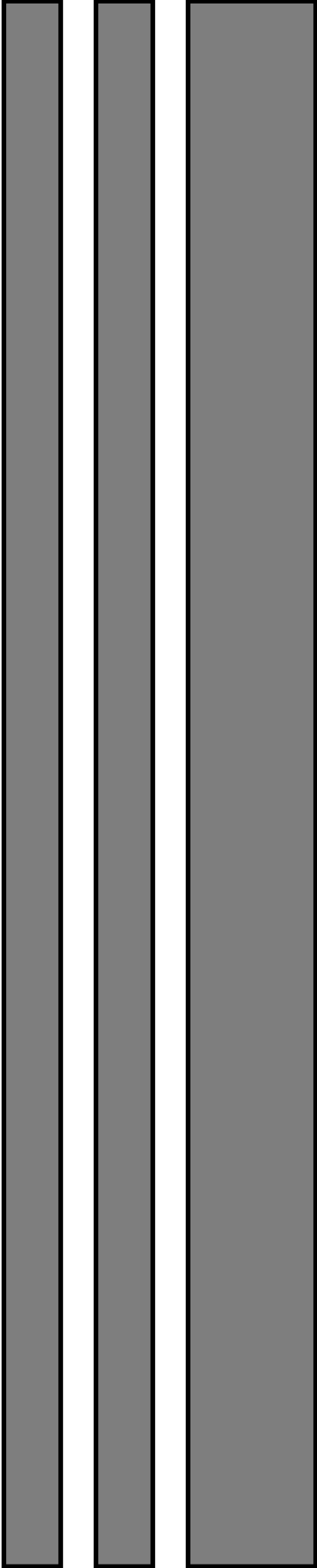




PTAA

Understanding Test Automation

Objectives

- Understand basic concepts of test automation
- Learn the role and value of test automation
- Learn some of the major issues in test automation

Synopsis

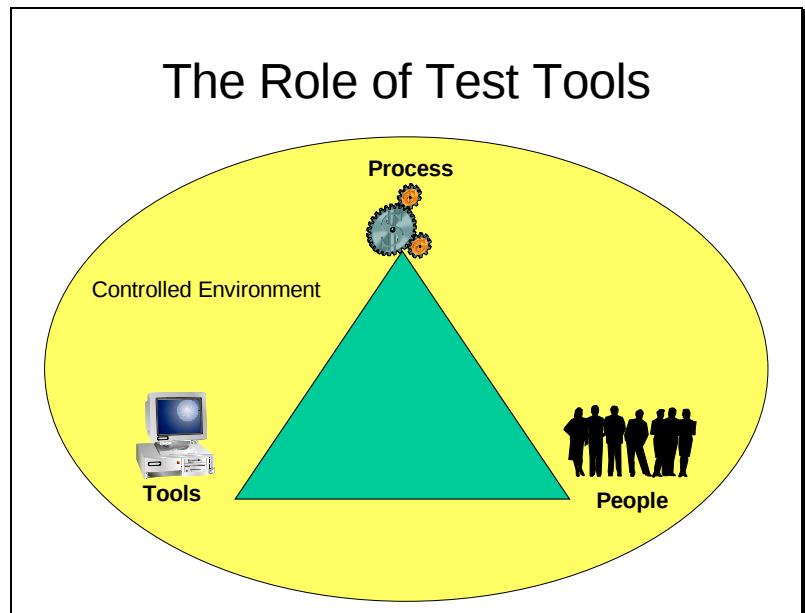
This introductory module builds a foundation of understanding about test automation.

The Relationship Between Test Tools and Test Automation

- Not every test tool is a test automation tool.
 - Defect tracking
 - Test Management
 - Test Design
- Test automation focuses on the unattended repeatable testing of software.

In this context, we need to distinguish between test automation and the larger realm of test tools. Not every test tool is a test automation tool. For example, defect tracking tools, test management tools and test design tools have an intersection with test automation tools, but they generally do not automate tests. Test automation focuses on the unattended repeatable testing of software.

The Role of Test Tools



Before we explore the challenges in more detail, it is important to understand that test tools are part of the solution for effective testing, but not the entire solution. Effective testing also requires processes and people to support them.

Testing vs. Test Automation

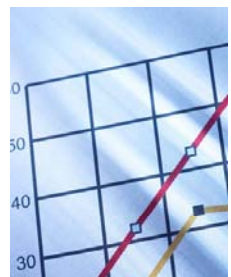
- Test automation is a different skill set than software testing.
- Not every test can or should be automated.
- Think of software testing as the broad discipline which employs test automation as one way to implement and perform tests.



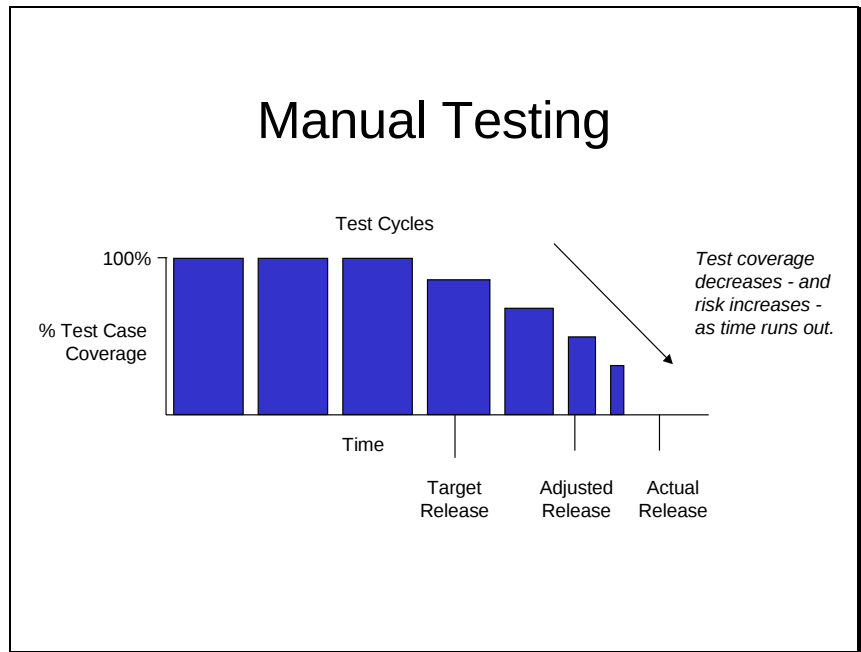
It is also important to understand that Test automation is a different skill set than software testing. Not every test can or should be automated. Those tests that can't be automated, or those chosen not to be automated, skilled manual testing is needed. Think of software testing as the broad discipline which employs test automation as one way to implement and perform tests.

The Promise and Value of Test Automation

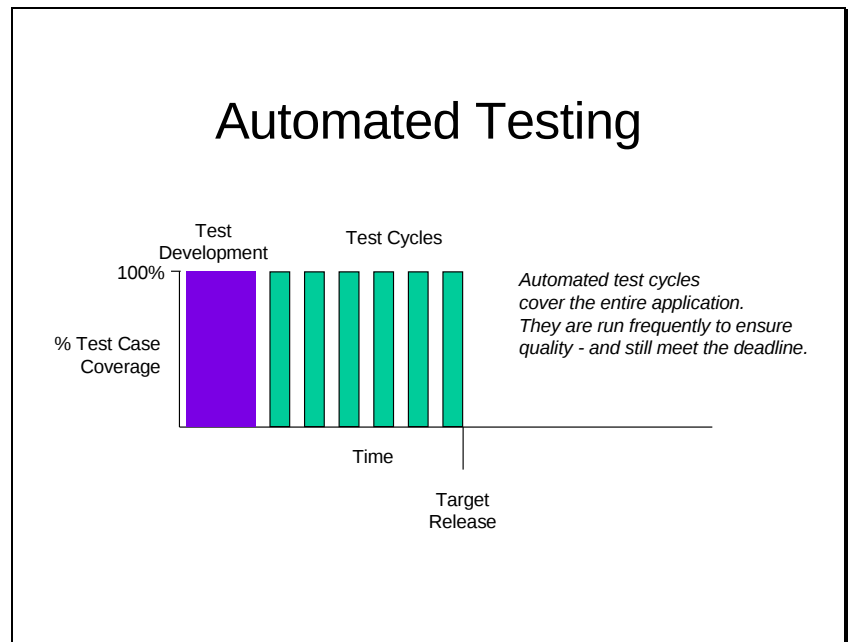
- Allow more time to be spent on tests that can only be performed manually
- Perform testing with greater precision
- Unattended testing to make the best use of testing resources.



Test automation is an exciting prospect because we know how time-consuming and tedious manual testing can be. Test automation also allows more time to be spent on tests that can only be performed manually and performs testing with greater precision. Unattended testing can be applied to make the best use of testing resources.



In this diagram, manual testing is seen as successive large blocks of time. As the deadline gets closer, the temptation is to reduce the level of testing, which increases the risk of missing a defect. Eventually only the changes are tested, giving the illusion of a rigorous test, but the reality is that only a percentage of the test cases are being tested.



In contrast, in automated testing, an initial large block of time indicates the investment in creating automated test scripts. The subsequent tests take much less time, but still achieve the same levels of test coverage. The benefit is that you can save time without sacrificing the scope of testing.

Common Measures and Metrics for Test Automation

- Percent of tests automated
- Percentage of defects found by automated tests
- Time saved by test automation
- Cost savings of test automation
- Component reuse



This slide shows some major test automation metrics. We will explore these in more detail later in the course.

Examples of Implementing Test Automation

- A large company creates a regression test library for one of their legacy applications.
- A software company creates a robust automated test suite that runs after each build.



These are just two examples of how test automation can be applied.

Your Test Automation Approach

- Success or failure depends on your approach, which includes:
 - Management and administration
 - Technical approach including test automation frameworks
 - Design approach
 - Testing processes
 - Organizational involvement



Your test automation approach will determine your success. Dorothy Graham and Mark Fewster, authors of the book “Software Test Automation” refer to the approach as an “automation regime.” In the following slides we will explore each of the major elements of the approach.

Management and Administration

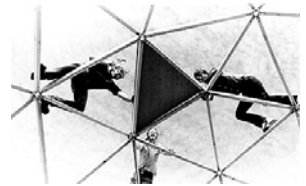
- This includes:
 - Configuration management
 - Testware management
 - Oversight of test automation tasks
 - Project management



Test automation management and administration includes configuration management, testware management, oversight of test automation tasks and project management. These management functions are important to control and mature the test automation effort. Without these functions, test automation becomes chaotic and failure-prone.

Technical Approach

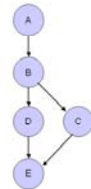
- This includes defining and applying approaches that fit the technical environments for applications to be tested.



The technical approach includes defining and applying approaches that fit the technical environments for applications to be tested.

Design Approach

- This includes approaches such as:
 - Data-driven testing
 - Keyword or Action word testing



The design approach includes approaches such as data-driven testing and keyword or action word testing

Testing Processes

- This includes the full range of testing processes such as:
 - Test planning
 - Test standards (including naming conventions)
 - Defect reporting and tracking
 - Metrics and measurement



These include the full range of testing processes such as test planning, test standards (including naming conventions), defect reporting and tracking, metrics and measurement.

Organizational Concerns

- Team structure for test automation
- Training
- Motivation to apply test automation
- Keeping expectations conservative



These are human issues for the organization and culture. They include team structure for test automation, training, motivation to apply test automation, and keeping expectations conservative.

Issues in Test Automation

- Time needed to achieve ROI
- Understanding what to test
- Selecting tools
- Getting trained
- Building a framework and infrastructure



Some of the major issues in test automation include the time needed to achieve return on investment, understanding what to test, selecting tools, getting trained, and building a framework and infrastructure.

What is Testware?

- “Artifacts produced during the test process required to plan, design, and execute tests, such as documentation, scripts, inputs, expected results, set-up and clear-up procedures, files, databases, environment, and any additional software or utilities used in testing.”
 - ISTQB Glossary [After Fewster and Graham]

Testware is basically all the items needed for testing.

Capture/Playback vs. Scripting

- At first, Capture/Playback seems easier to create automated test scripts.
- However, once captured (recorded), the scripts must be modified to run reliably in the future.
 - Still has high maintenance.
- Scripting starts from a blank slate and is designed with maintenance in mind.



At first, Capture/Playback seems easier to create automated test scripts. However, once captured (recorded), the scripts must be modified to run reliably in the future. Capture/playback has high maintenance. Scripting, on the other hand, starts from a blank slate and is designed with maintenance in mind. However, there are effective and ineffective ways to create automated test scripts.

Skills Needed for Test Automation

- This varies, depending on the tasks at hand.
- You don't need technical skills to use an automation tool or framework.
- However, to create and implement test automation, technical skills are needed.



The skills needed for test automation varies depending on the tasks at hand. You don't need technical skills to use an automation tool or framework. However, to create and implement test automation, technical skills are needed.

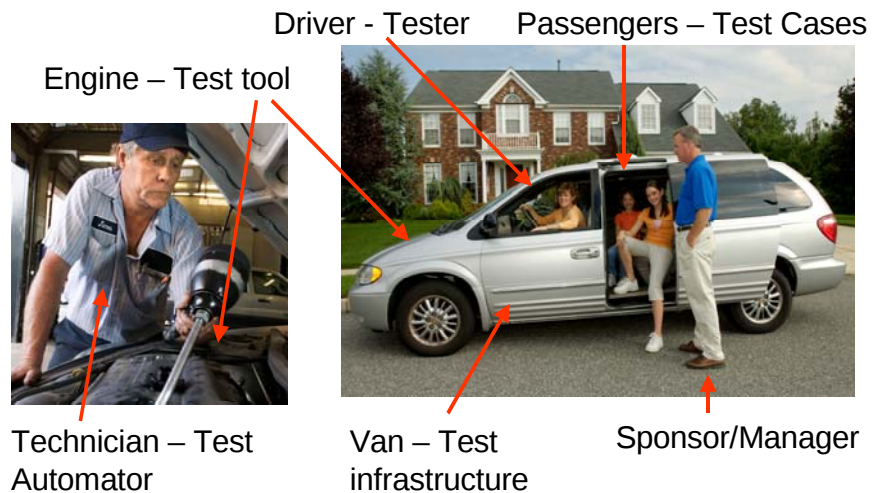
Technical Skills for Test Automation

- Coding skills
- Database skills
- Specific technology skills and knowledge
 - Web
 - Client/server
 - Mainframe
- Logical thinking



The technical skills needed for test automation include coding skills, database skills, specific technology skills and knowledge (such as web, client/server, mainframe) and logical thinking.

The Car Analogy



This is based on analogy from Dorothy Graham. In this analogy, the vehicle is the test infrastructure. The engine is the test tool and the auto tech is the test automator. In the driver's seat is the tester, who has ways to interact with the test infrastructure and engine. The passengers are the test cases. These are transported by the test infrastructure. The sponsor or manager is the owner of test automation, but does not directly use or apply the automation.

Test Automation is a Multi-Faceted Project

- With objectives, schedules, roles and costs
- Requires effective management
- Containing a variety of other projects
- An ongoing effort



Test automation is a project in its own right, with objectives, schedules, roles and costs. This project requires effective management and often contains a variety of other projects. Test automation is an ongoing effort.

About Specific Tools

- In this course, we do not assume a particular tool or vendor.
- These concepts can be applied with many different tools.
- For the exercises, we will use Macro Scheduler which has a robust scripting language.



In this course, we do not assume a particular tool or vendor. These concepts can be applied with many different tools. For the exercises, we will use Macro Scheduler which has a robust scripting language.

Critical Success Factors for Test Automation

- Senior management support
- Keeping expectations realistic
- Having a good proof-of-concept for the tools selected
- Reuse of automated testware
- Training
- Grassroots support and realization of value



These are major critical success factors for test automation.