Cognizant
Passion for building stronger businesses

Automation – How to succeed, what to watch out for

Sragabart Mahakul
Cognizant Technology Solutions, Pune

Agenda

- What to Automate ?
- When to Automate ?
- How much to Automate ?
- Automation Methodology
- Automation Framework
- Test Automation Group's Scope & Objectives
  - Assumptions
  - Constraints
  - Critical Success Factors
  - Strategic Objectives

What to Automate ?

	Simple Things	Hard Things
Done Once	Do it Manually	Automate It
Done often	Automate It	Buy or Write Software

What to Automate ?

- What to Automate ?
  - Automate the primary functions that will be used by the End-users
  - Automate the BVTs
  - Automate End-to-End Scenarios

- What not to Automate ?
  - Do not try to automate not-so critical portions in the beginning
  - Do not automate status bars, help screens

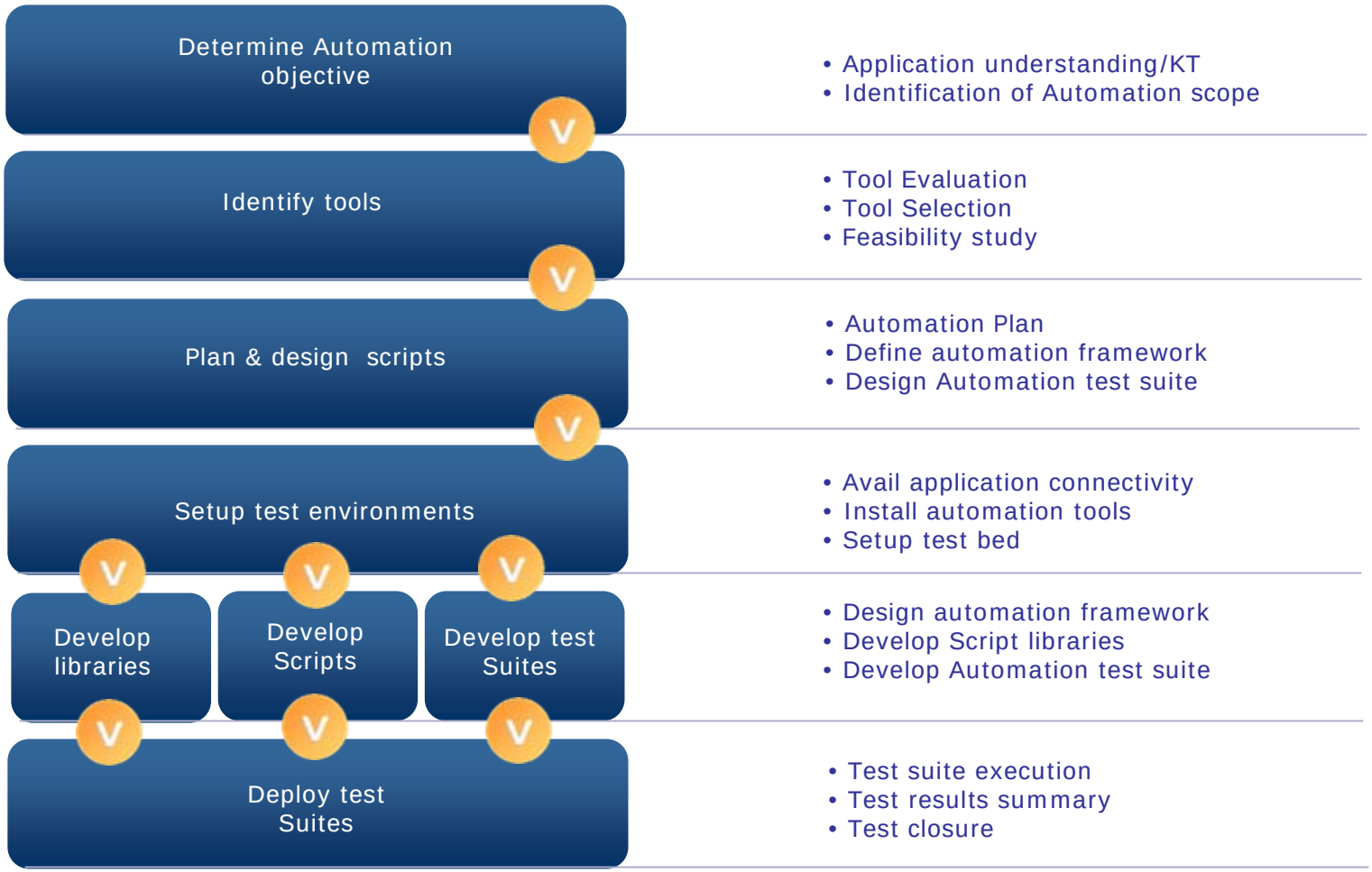
When to Automate ?

- When to Automate ?
  - Is the Build Stable ?
  - Are the Test Cases and Test Scenarios ready and Final ?
  - Are the Test Data ready ?
  - Is the Test Bed ready ?
  - Is the Test Automation tool installed ?
- When not to Automate ?
  - If the AUT is not Large\Complex
  - If you receive only few builds to test
  - If the feature doesn't work accurately

How much to Automate ?

- Ideally 60% Automation is expected for a Regression Suite
- First, automate the primary functions that will be performed by the targeted end-users
- Next, add the not-so-critical portions of the application as time permits.
- Develop a test coverage matrix

Automation Methodology

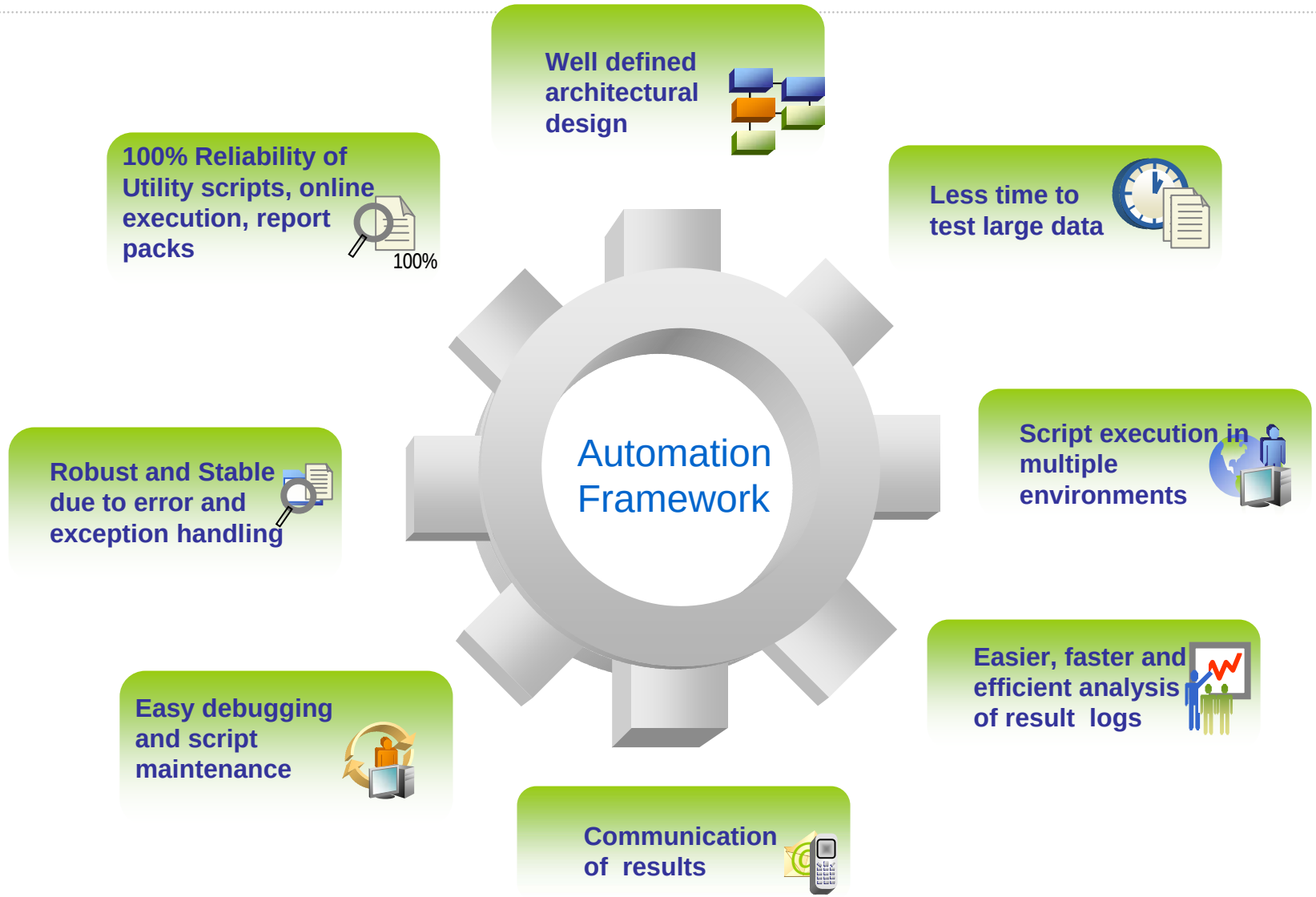


Automation Framework

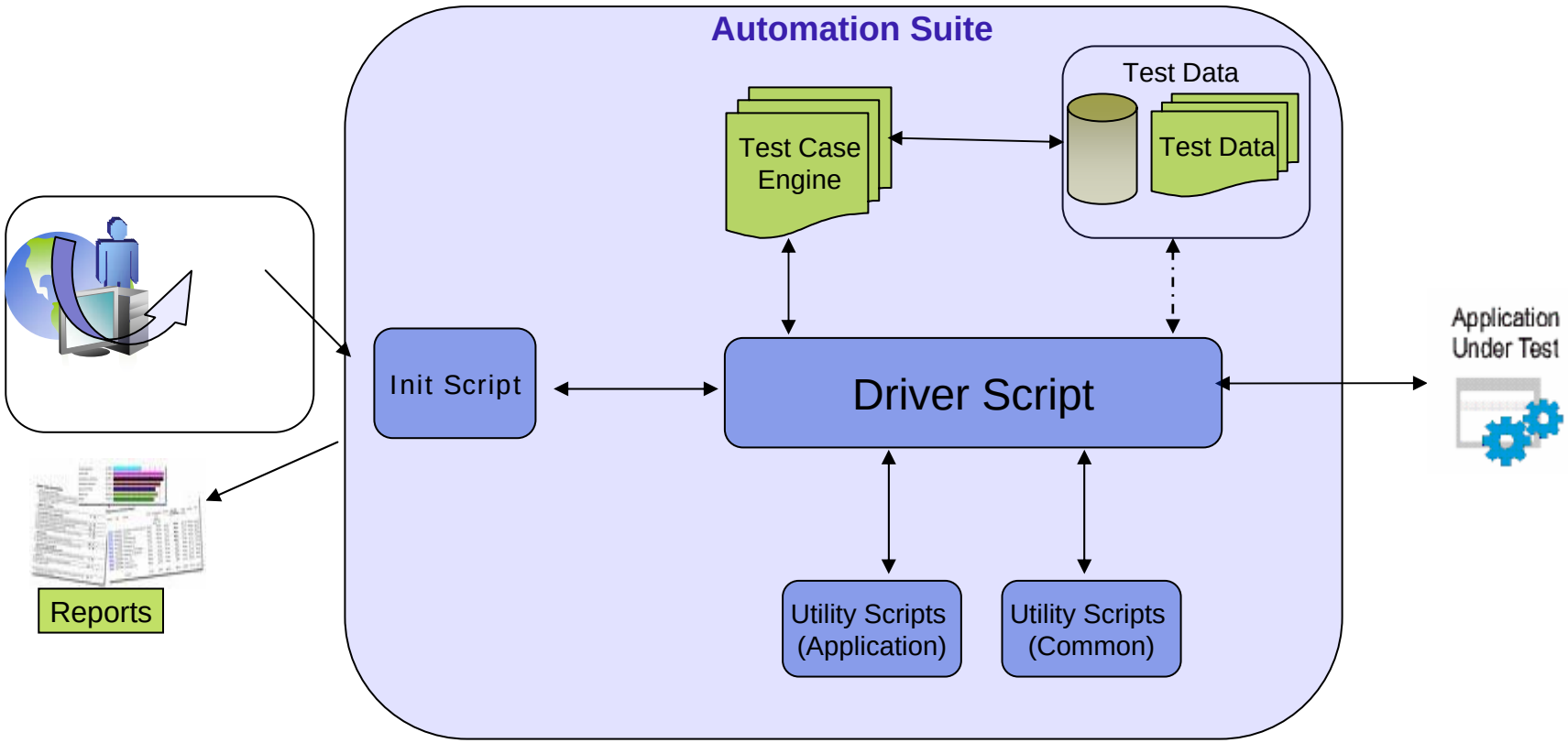
- Definition
  - An underlying skeleton upon which a test suite is built
  - A simplified description of a complex entity or process
  - A tiered organization of the function libraries

- Advantages and Benefits
  - Rapid Script Development
  - Reusability
  - Smart Exception-handling
  - Well Organized Code
  - Test Data Management
  - Ease of maintenance

Automation Framework Features



Architecture



Test Automation Group's Scope

- Assumptions
  - An integrated tool suite must be the primary test management, planning, development, and implementation vehicle.
  - The tool suite must be used to direct and control test execution, to store and retrieve test artifacts, and to capture/analyze/report test results.
  - Testing standards must be documented and followed.

Assumptions…

- The tool suite must include
  - a tool of choice for defect tracking and resolution.
  - a component for test requirements management.
  - a configuration management tool of choice
- All of the tools described above must be integrated with desktop tools such as MS Office.
- The proper automated testing workspaces must be created on test servers that are separate from development servers.

Constraints

- These constraints limit the success of the automation effort if they are not heeded.
  - The automated tools group resources must remain independent of any manual testing group.
  - There may not be a large enough number of available staff on the automation team.
  - The level of cooperation of the software development group and their management with respect to automated tool use may be too low.

Constraints...

- There may be a lack of cooperation and information exchange with developers in creating testable applications.
- The release schedules for major versions of the AUT and for customer-specific releases of the AUT can be too tight.
- There is uncertainty associated with the GUI updates in AUT.
- There may be corporate mandates on what tools must be used.

Critical Success Factors

- Following critical success factors can be treated as a set of test automation guidelines
  - » Test automation must be implemented as a full-time effort, not a sideline.
  - » The test design process and the test automation framework must be developed as separate entities.
  - » The test framework must be application independent.

Critical Success Factors...

- The test framework must be easy to expand, maintain, and enhance.
- The test strategy/design vocabulary must be framework independent.
- The test strategy/design must hide the complexities of the test framework from testers.

Strategic Objectives

- These objectives are based on the critical success factors listed above.
  - » Implement a strategy that will allow tests to be developed and executed both manually (initial test cycle) and via an automation framework (regression test cycles).
  - » Separate test design and test implementation to allow test designers to concentrate on developing test requirements, test planning, and test case design while test implementers build and execute test scripts.

Strategic Objectives…

- Implement a testing framework that both technical and non-technical testers can use.
- Employ a test strategy that assures that test cases include the navigation and execution steps to perform, the input data to use, and the expected results all in one row or record of the input data source.
- Realize an integrated approach that applies the best features of keyword-driven testing, data-driven testing, and functional decomposition testing.

Strategic Objectives...

- Implement an application-independent test automation framework.
- Document and publish the framework.
- Develop automated build validation (smoke) tests for each release of the application.
- Develop automated environmental setup utility scripts for each release of the application.

Strategic Objectives...

- Develop automated regression tests for
  - GUI objects and events
  - Application functions
  - Application special features
  - Application performance and scalability
  - Application reliability
  - Application compatibility
  - Application performance
  - Database verification

Cognizant
Passion for building stronger businesses

Thank you

Testing Services