Improve Functional Testing via Model-Based Automated Test Design

TCE

Vinay Kumar

Honeywell

Agenda

Honeywell

- **Background**
  - **Introduction to MBT**
- **Case study**
  - **Piloting MBT**
    - Pre-requisite
    - System modeling
    - Test generation
    - Test harness & adaptor developer
    - Executable test script/Test framework
  - **Observations**
    - Quality
    - Automation advantage
    - Demands process up shift
    - Project & Tool selection
    - ROI

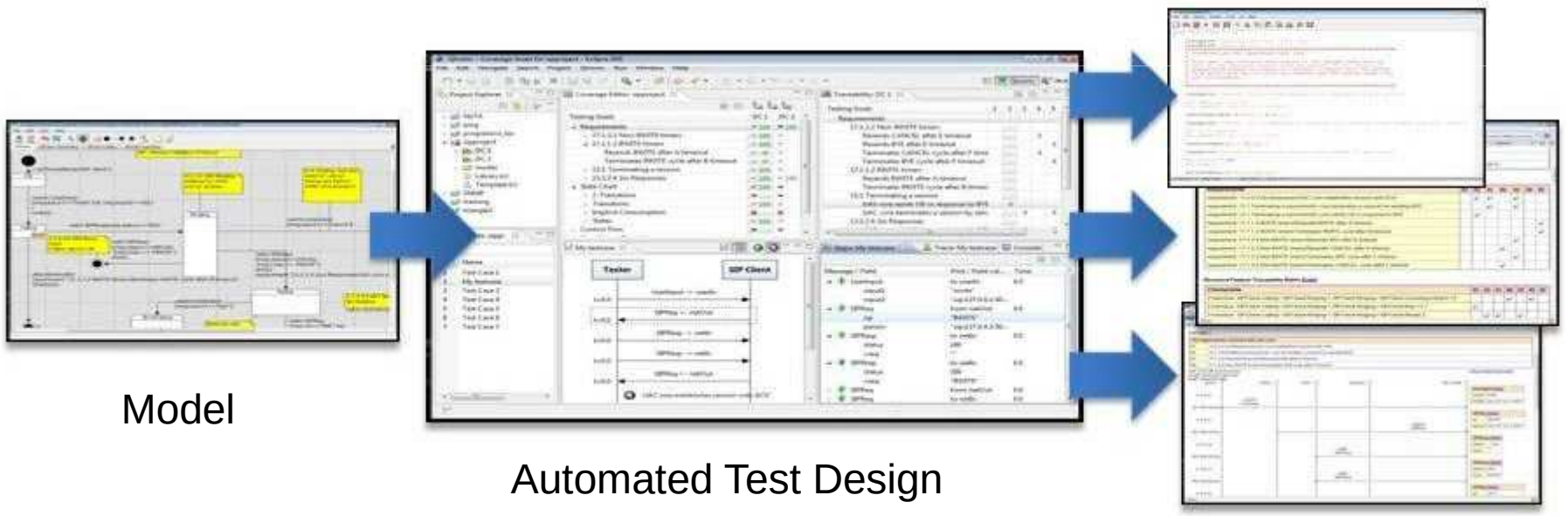
Background

Honeywell

Model Based Testing

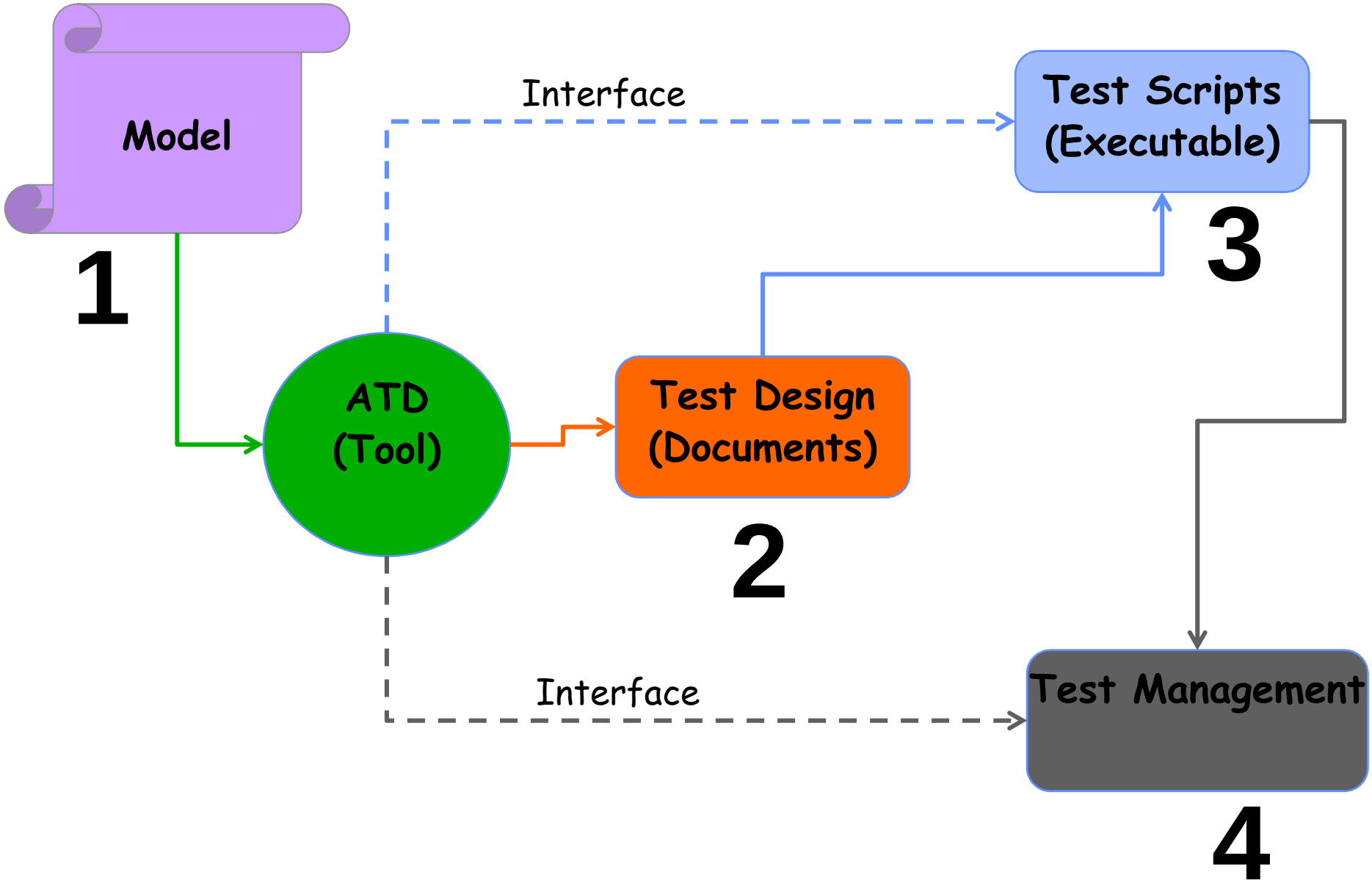
Honeywell

- **Concept**
  - Black box, Functional testing methodology
  - **“**Model-based testing is automation of test design**”**
  - A model representation of system is used to **auto generate** test design for the system



Model Based Testing

Honeywell



Case Study

Piloting MBT

Honeywell

Pre-requisite

Honeywell

- **Team**
  - **UML modeling & programming skill**
    - UML notation familiarity (State chart)
    - Not necessarily design skills
    - Should be capable of converting plain English requirement specification in UML state chart and action language Java
  - **Test execution automation expert**
    - Automation script library development
  - **Domain expert**

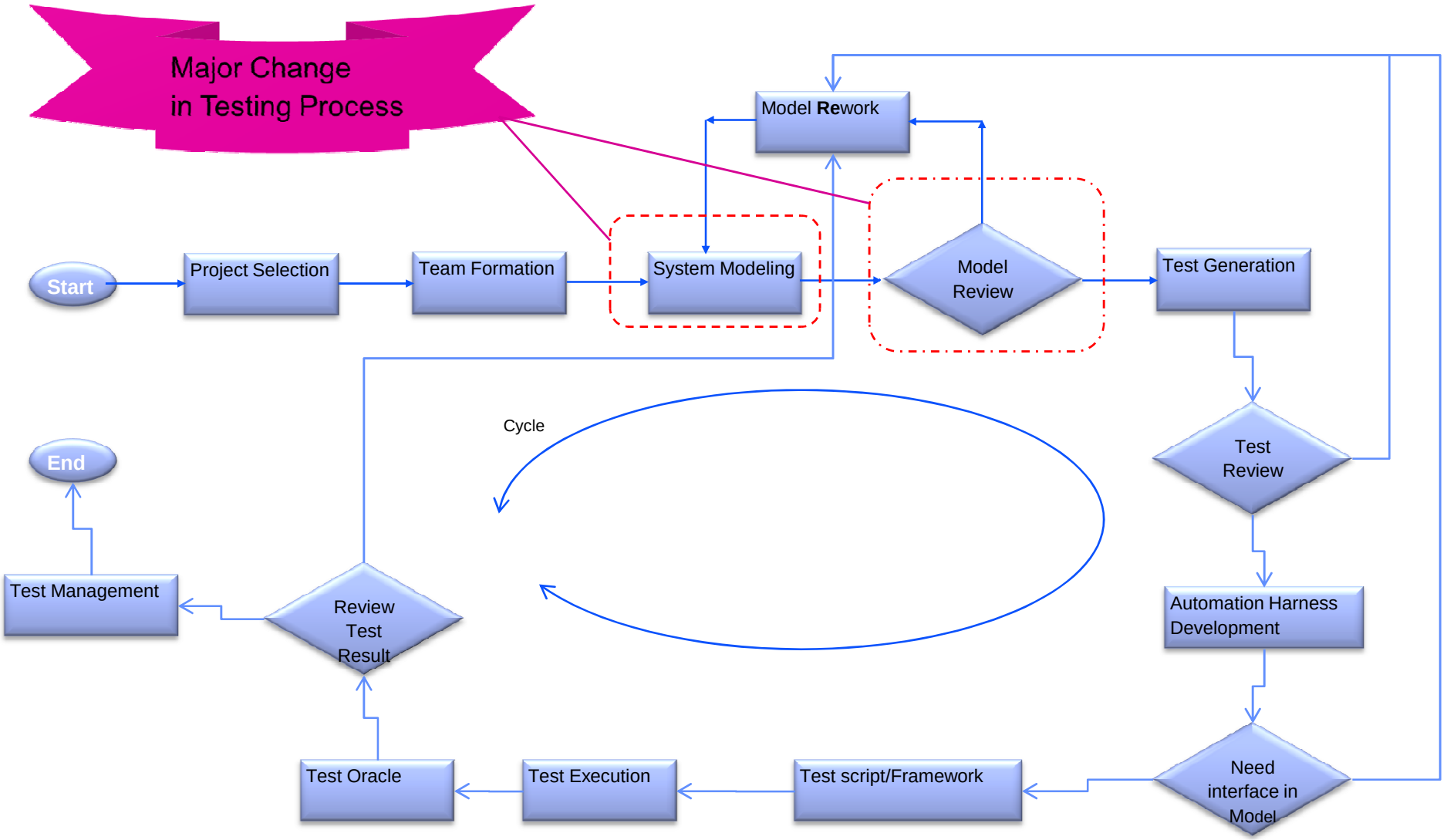
- **Infrastructure**
  - MBT tool (Qtronic)

- **Automatic test execution**
  - Feasibility of Automatic test execution should be nearing to 100%

- **Well documented requirement specification**

Pilot Flow

Honeywell

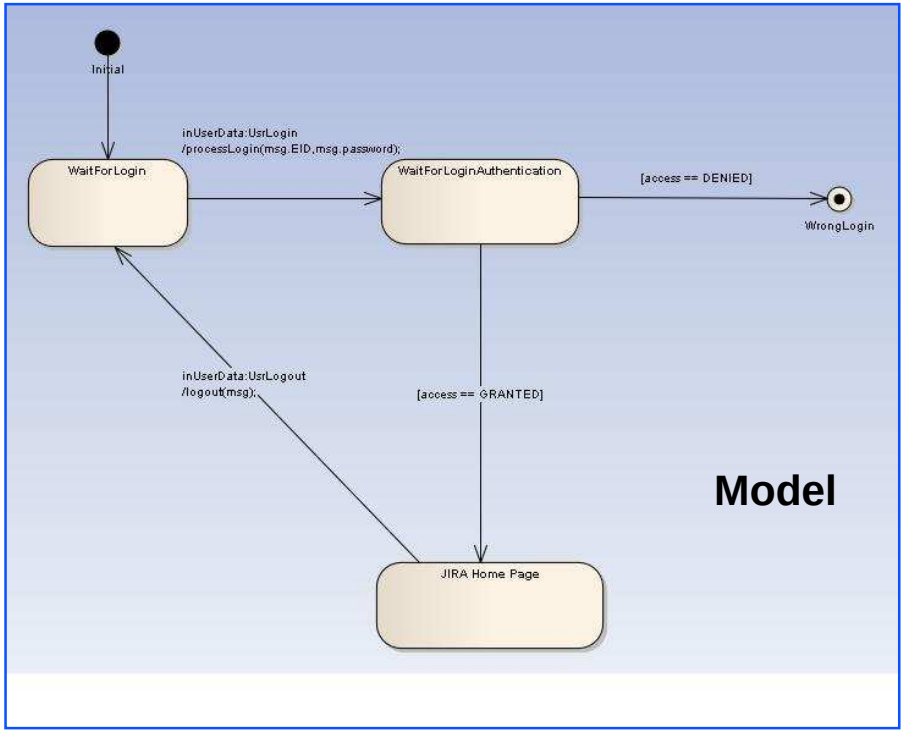


System Modeling

Honeywell

- Black box
- Functional aspects
- The Model must represent exact system behavior as described in system specification and requirement or the documentation

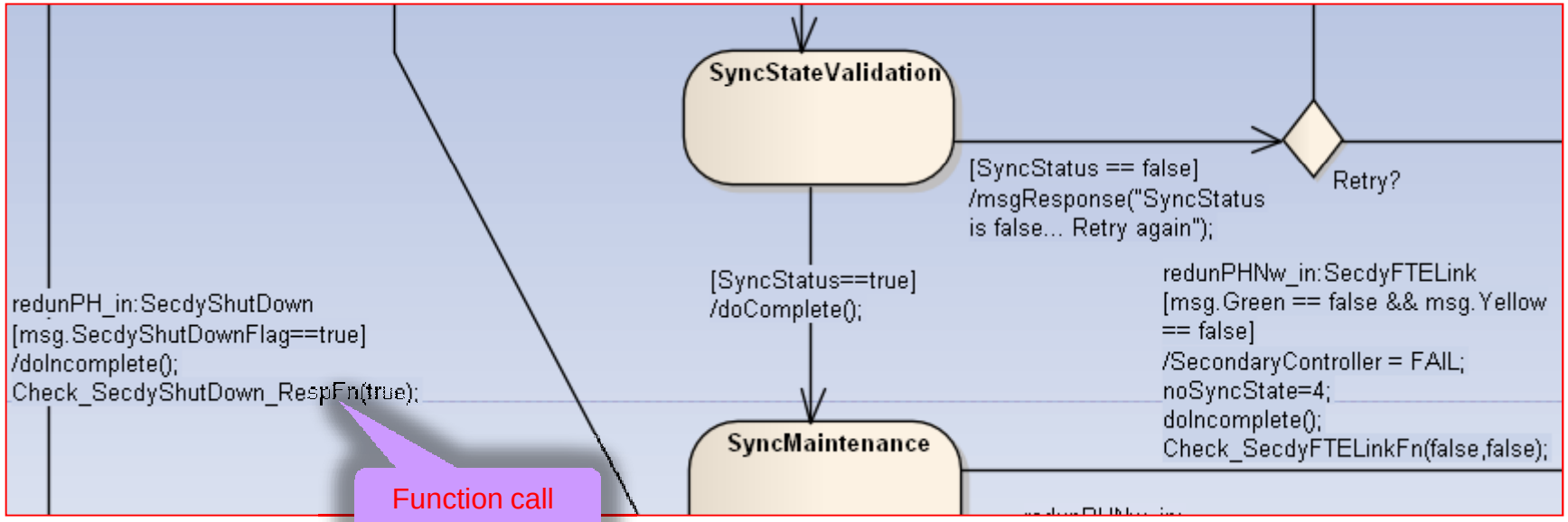
UML + Java =



Model Review

Honeywell

- Think model as a computer program
- Needs deep review
- Coding errors should not alter functional behavior of the system under test



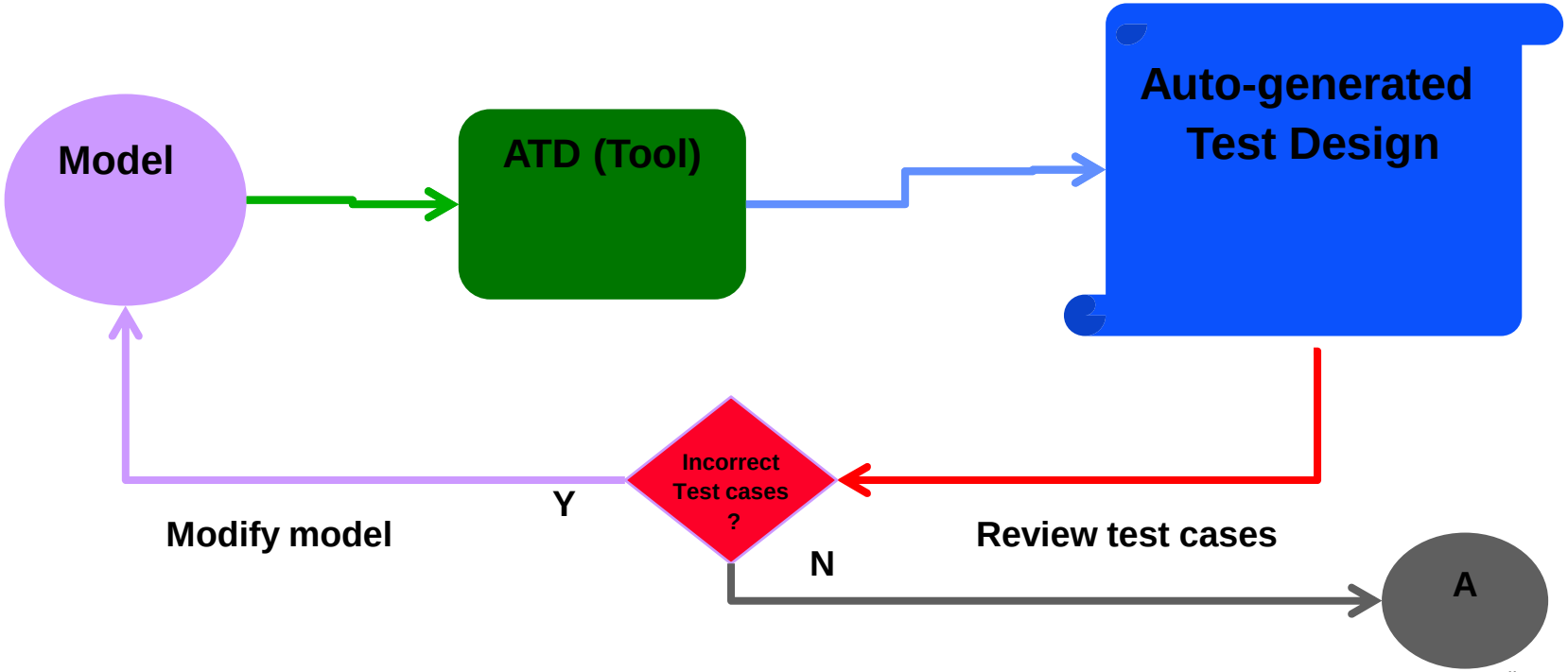

```
public void Check_SecdyShutDown_RespFn(boolean state)
{
      Check_SecdyShutDown resp;
      resp.Check_SecdyShutDown_Resp = state;
      respMsg.send(resp);
}
```

Test Generation

Honeywell

- Automatic test case generation from abstract system model

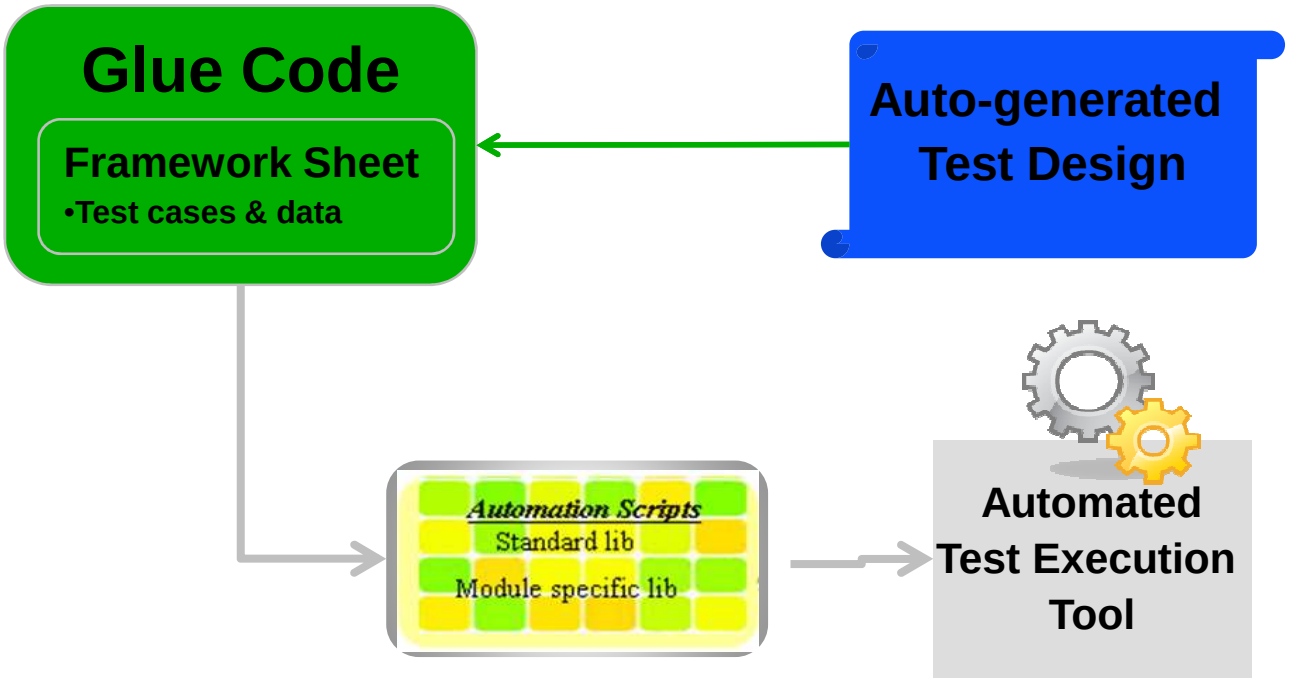
- Auto generate test document for manual test execution in given template



Test Harness

Honeywell

- Generates executable script
- Need one time effort to develop glue code



Case Study

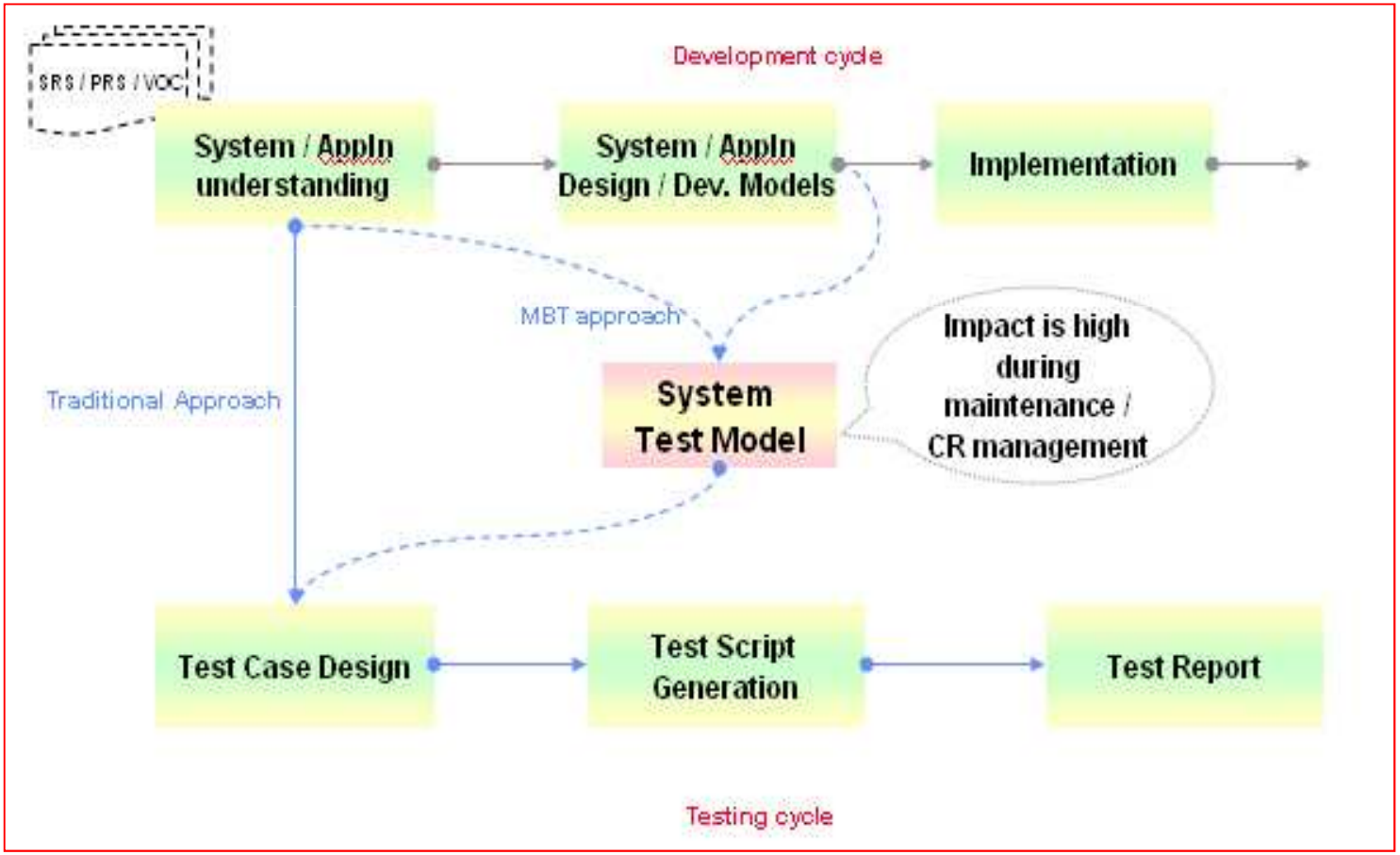
Observations

Honeywell

Conclusion

Honeywell

- **Key benefits**
  - **High productivity during maintenance and change requests**



Conclusion

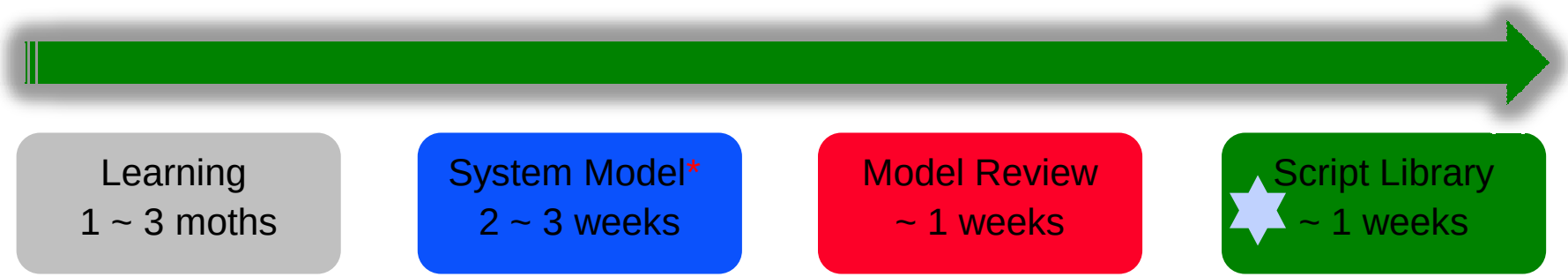
Honeywell

- **Test Quality**
  - Test case quality is a stamp of the Model
- **Automation advantage**
  - Modeling of the system necessitates a deep thinking about the system that aids the project and architect to deliver a highly mature product. This “extra” thinking process finds specification ambiguities, omissions, and conflicts.
  - The support of various test execution environments through a customizable interface gives end to end automation from test design to test execution and management
- **Demands process change**
  - It introduces formal modeling and modifies the test development processes. Competency building within the conventional tester could be an issue because of the modeling and programming skill demand of MBT tools.
- **Project & Tool Selection**
  - Model based testing is suitable for functional testing (as a black box) and brings more value for projects having an automated test execution environment

Contd..

Honeywell

- **ROI**
  - More/Same/less cycle time in 1st iteration as compare to manual testing
    - Depends on modeling & programming skill
  - Benefit is more in subsequent iterations in case CRs and maintenance phase
  - Better coverage



* 10+ requirements,
moderate complexity
3 resources

Questions?

Thank You!!

Honeywell