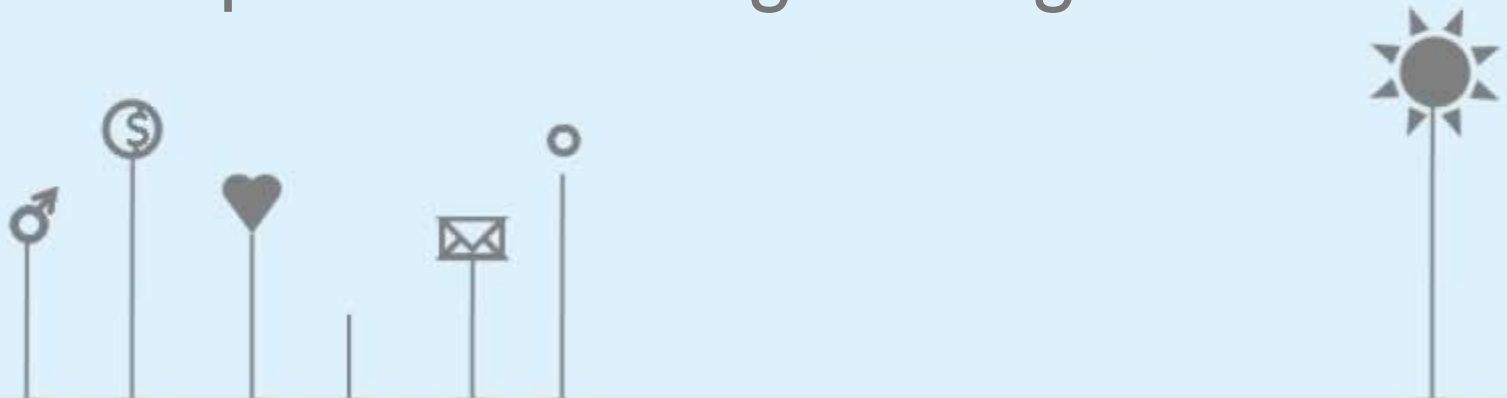


Software College Northeastern University

# Software Quality Assurance and Testing

## Chapter 11 Testing Management



[wuchenni@qq.com](mailto:wuchenni@qq.com)

# Contents



## Chapter 11

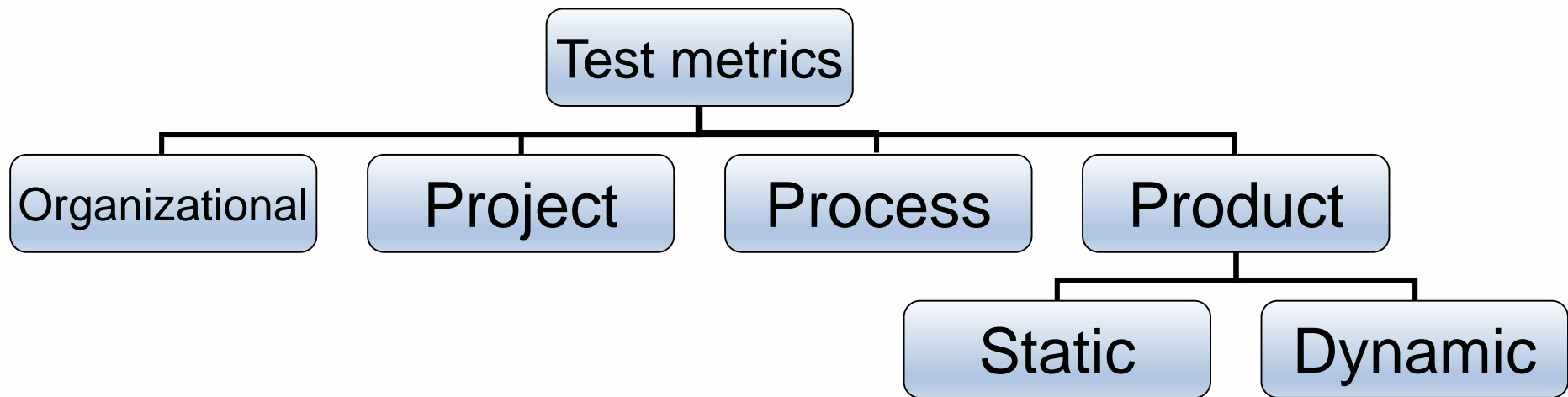
### Testing Management

---

- 11.1 Test Metrics
- 11.2 Defect Management
- 11.3 Software Testing Life Cycle
- 11.4 Basic Knowledge about Software Testing Automation
- 11.5 Automated Testing Tools

# 11.1 Test Metrics

- It refers to a standard of measurement.
- A classification of various types of metrics





## 11.1 Test Metrics

- Four general core areas that assist in the design of metrics
  - Schedule
  - Quality
  - Resource
  - Size



## 11.1 Test Metrics

- Organization metrics
  - Useful in overall project planning and management
  - Shows the quality trend across the organization
  - Allow senior management to monitor the overall strength of the organization and points to areas of weakness



## 11.1 Test Metrics

- Example
  - The average defect density across all software projects in a company is 1.73 defects per KLOC. Senior management has found that for the next generation of software products, which they plan to bid, they need to show that product density can be reduced to 0.1 defects per KLOC. The management thus sets a new goal.



## 11.1 Test Metrics

- Project metrics
  - Useful in the monitoring and control of a specific project.
  - The ratio of actual-to-planned system test effort is one project metric.
  - The ratio of the number of successful tests to the total number of tests in the system test phase is another project metric.

# 11.1 Test Metrics



- Process metrics
  - Every project uses some test process.
  - The goal of a process metric is to assess the goodness of the process.



# 11.1 Test Metrics



- Example
  - In one software development project it was found that 15% of the total defects were reported by customers, 55% of the defects prior to shipping were found during system test, 22% during integration test, and the remaining during unit test.
  - What does it indicate?

# 11.1 Test Metrics



- Product metrics: Generic
  - Useful in making decisions related to the product.
  - Product complexity-related metrics abound.
- Product metrics: OO software
  - (1) Reliability
  - (2) Defect density
  - (3) Defect severity
  - (4) Test coverage
  - (5) Cyclomatic complexity
  - (6) Weighted methods per class
  - (7) Class coupling
  - (8) Response set
  - (9) Number of children



## 11.1 Test Metrics

- Testability
  - It is 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”. (According to IEEE)
  - High testability is a desirable goal.
  - Testability in software focuses on the verification of design and implementation.

# 11.1 Test Metrics



- Example
  - Consider an application **E** required to control the operation of an elevator. **E** allow a tester to experiment with various scheduling algorithms.
  - What should we do?( Pay attention to the testability)



## 11.2 Defect Management

- Defect management
  - It is a subprocess of the development process.
  - It entails the following
    - Defect prevention
    - Defect discovery
    - Defect recording and reporting
    - Defect classification
    - Defect resolution
    - Defect prediction



## 11.2 Defect Management

- **Status associated with a defect**
  - **New:** When a bug is found/revealed for the first time, the software tester communicates it to his/her team leader in order to confirm if that is a valid bug. After getting confirmation from the test lead the software tester logs the bug and the status of “New” is assigned to the bug.
  - **Assigned:** after the bug is reported as “New” it comes to the development team. The development team verifies if the bug is valid. If the bug is valid, development leader assigns it to a developer to fix it and a status of “assigned” is assigned to it.
  - **Open:** Once the developer starts working on the bug, he/she changes the status of the bug to “Open” to indicate that he/she is working on it to find a solution.



## 11.2 Defect Management

- **Status associated with a defect**
  - **Fixed:** Once the developer makes necessary changes in the code, he/she marks the bug as “Fixed” and passes it over to the development Lead in order to pass it to the testing team.
  - **Pending retest:** After the bug is fixed, it is passed back to the testing team to get retested and the status of “Pending Retest” is assigned to it.
  - **Retest:** The testing team leader changes the status of the bug, which is previously marked with “Pending Retest” to “Retest” and assigns it to a tester for retesting.
  - **Closed:** After the bug is assigned a status of “Retest”, it is again tested. If the problem is solved, the tester closes it and mark it with “closed” status.
  - **Reopen**
  - **Rejected**



## 11.2 Defect Management

- The following structure is recommended to report a defect:

- **Title:**

- Type the problem encountered in the application, the title needs to be understandable

- For Example:

- You can use the following categories:

- » Missing
      - » Inaccurate
      - » Incomplete
      - » Inconsistent
      - » Incorrect

- Example:

- » Missing validation in “Project” field
      - » Incorrect spelling in “status” drop down list



## 11.2 Defect Management



- **Description**
  - Type a brief description of the problem
- **Repro Steps:**
  - Type all the steps to get to the problem, all steps must be cleared
    - For example:
      - 1.- Login to FIDO
      - 2.- Click on Add Invoice
      - 3.- Type !@#\$%% in Project field
      - 4.- Click on Save

## 11.2 Defect Management



- Actual Results
  - Type the actual results of the action  
*For example: The following error message is displayed.*
  - Comments:  
Type any comments or notify to the developers of any screenshots  
(attachments)  
*For Example: This defect is reproducible in Project field. (see attached file)*
- Expected Results  
Type the expected results of the action.  
*For Example: Data should be saved successfully.*

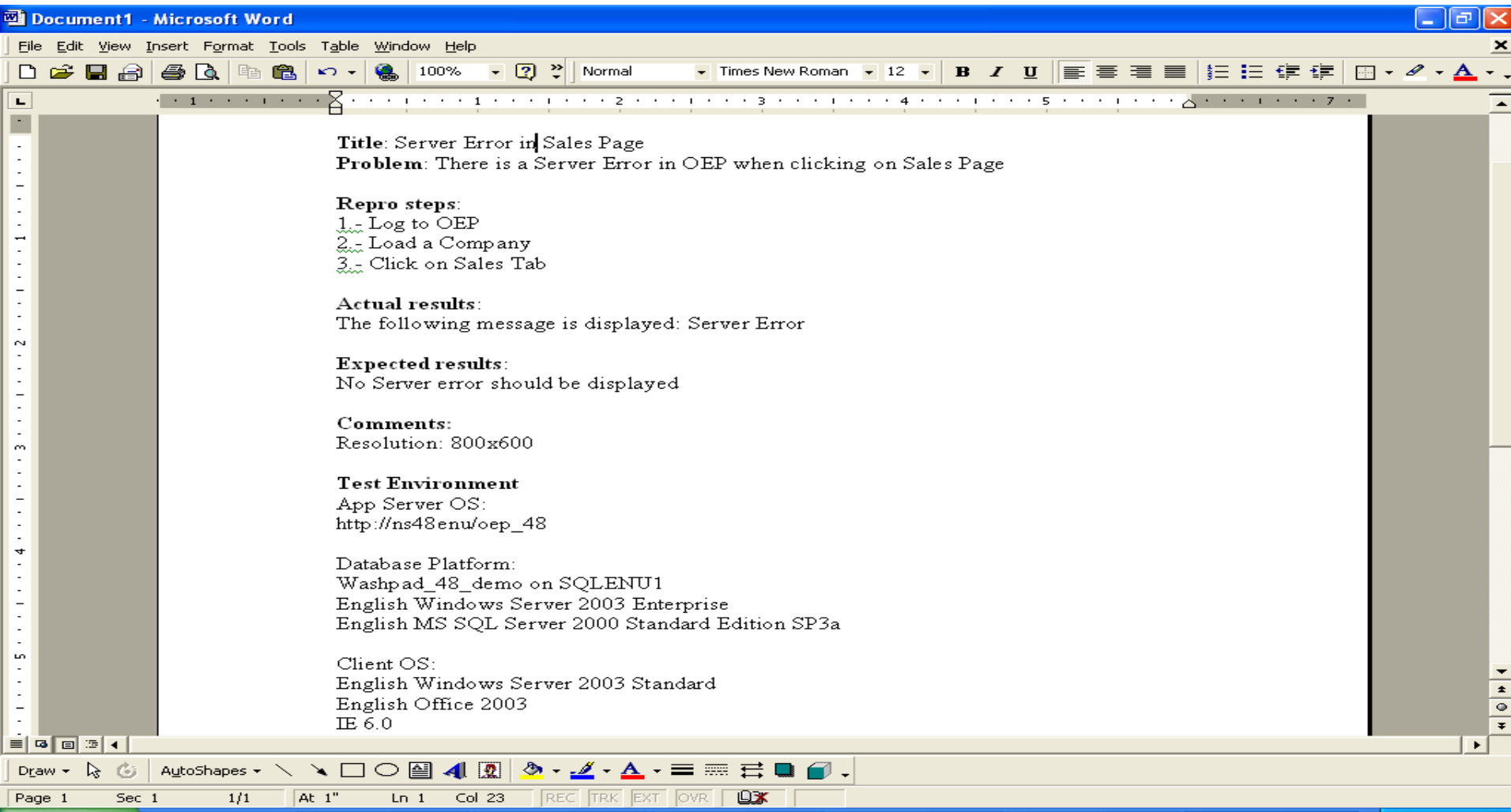
## 11.2 Defect Management



- Test Environment
  - Include details of the test environment
  - For Example:
    - Microsoft Windows 2003 Standard
    - Office 2003
- The following slide shows an example of a complete “Defect Report”

# 11.2 Defect Management

## Defect Report Example



**Title:** Server Error in Sales Page

**Problem:** There is a Server Error in OEP when clicking on Sales Page

**Repro steps:**

- 1.- Log to OEP
- 2.- Load a Company
- 3.- Click on Sales Tab

**Actual results:**  
The following message is displayed: Server Error

**Expected results:**  
No Server error should be displayed

**Comments:**  
Resolution: 800x600

**Test Environment**

App Server OS:  
[http://ns48enu/oep\\_48](http://ns48enu/oep_48)

Database Platform:  
Washpad\_48\_demo on SQLENU1  
English Windows Server 2003 Enterprise  
English MS SQL Server 2000 Standard Edition SP3a

Client OS:  
English Windows Server 2003 Standard  
English Office 2003  
IE 6.0

# 11.2 Defect Management

## Defect Tracking Example

Microsoft Excel - DefectSheetTemplate.xls

File Edit View Insert Format Tools Data Window Help

Arial 10 B I U \$ % , .00 .00

L5 =

	A	B	C	D	E	F	G	H	I	J	K	L
1	<b>FIDO_ Defect Tracking Sheet</b>											
2												
3	<b>Defect ID #</b>	<b>Problem</b>	<b>Detailed Description</b>	<b>Defect Severity</b>	<b>Defect Priority</b>	<b>Defect Status</b>	<b>Date Found</b>	<b>Date Close</b>	<b>Product</b>	<b>Product Detail</b>	<b>Assigned to</b>	<b>Responsible QA</b>
4	1	Server Error in Home Page	1.- Log to FIDO 2.- Click on Home Page Icon	Showstopper	Immediate	Open	9/11/2005		FIDO	Home Page	Mark	Gabriel
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												

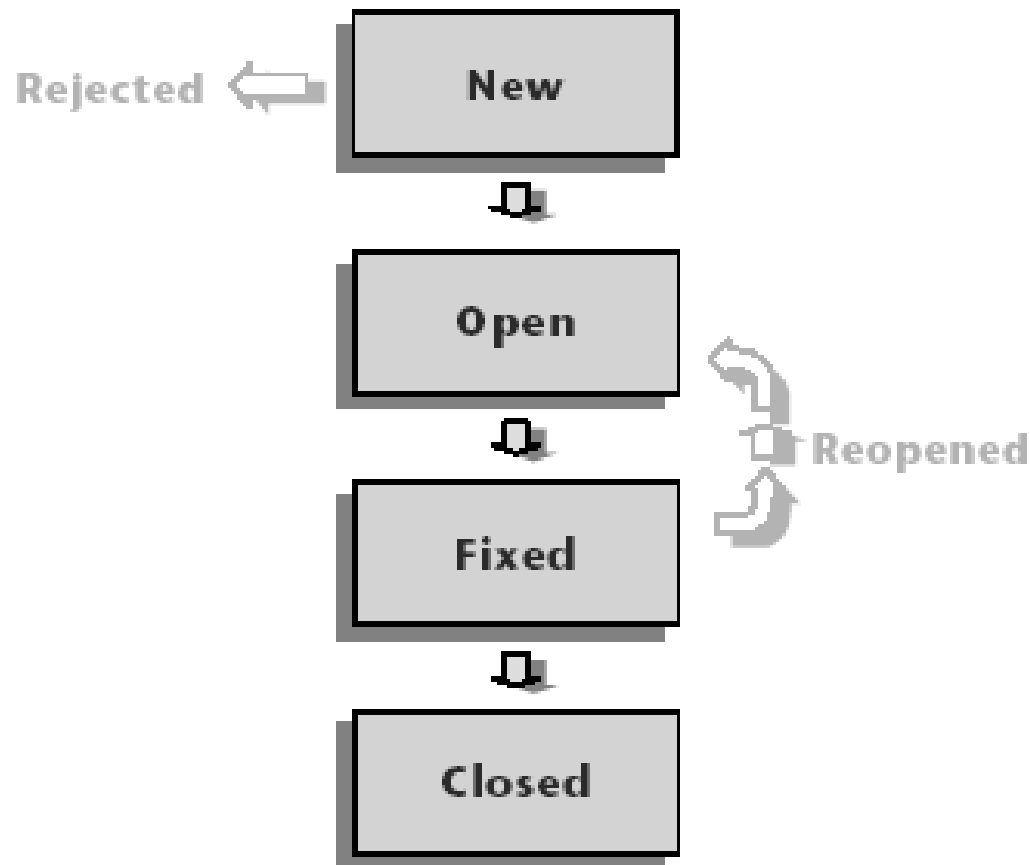
Sheet1 / Sheet2 / Sheet3 /

Draw AutoShapes

Ready

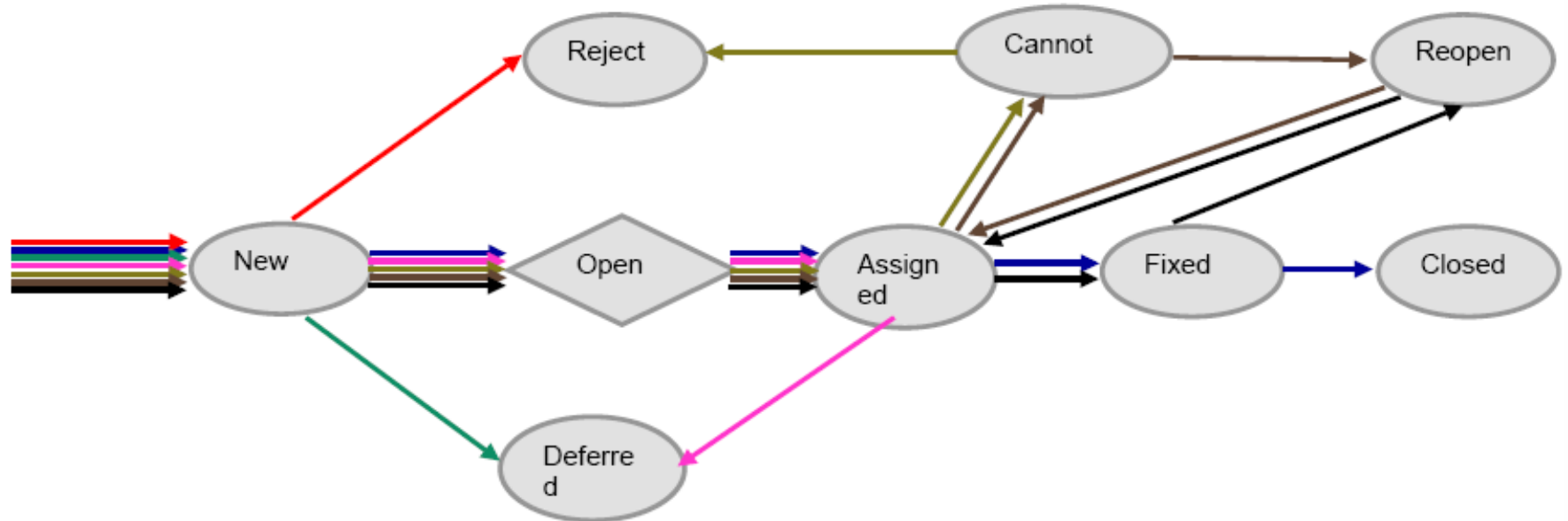
# 11.2 Defect Management

## Defect Life Cycle



# 11.2 Defect Management

## Defect Life Cycle Paths



# 11.2 Defect Management



## Defect Life Cycle Paths

1. New - Open - Assigned - Fixed - Closed

2. New - Reject

3. New - Deferred

4. New - Open - Assigned - Deferred

5. New - Open - Assigned – Cannot Reproduce - Reject

6. New - Open – Assigned – Cannot Reproduce – reopened - Assigned

7. New - Open – Assigned - Fixed – Reopen - Assigned





## 11.3 Software Testing Life Cycle

- Test planning
- Test design
- Test development
- Test execution
- Test evaluation

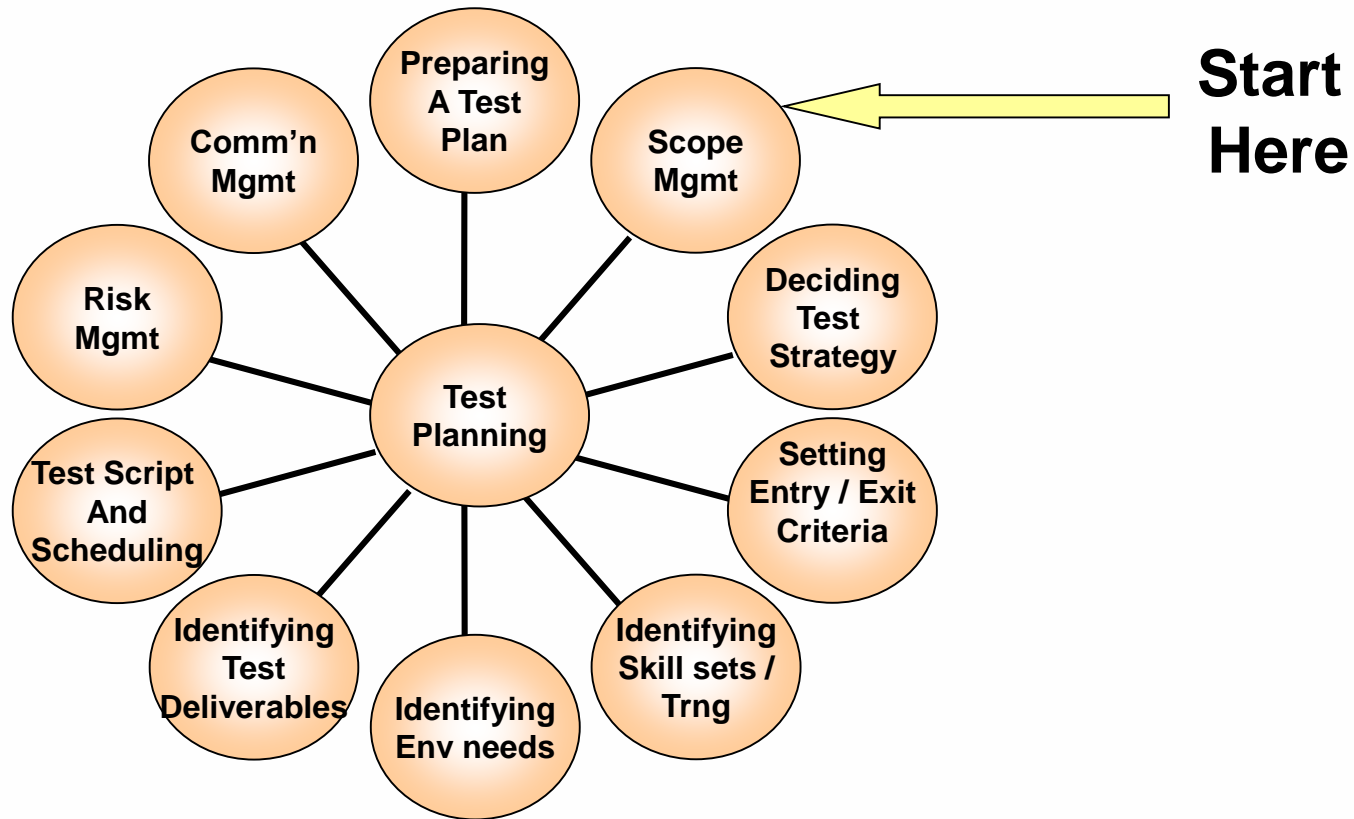


## 11.3 Software Testing Life Cycle

- Test Planning
  - *It is the process of defining a testing project such that it can be properly measured and controlled.*
  - *It includes test plan, test strategy, test requirements and testing resources.*

# 11.3 Software Testing Life Cycle

## Parts of Test Planning





## 11.3 Software Testing Life Cycle

- Test design
  - It is the process of defining test procedures and test cases that verify that the test requirements are met.
  - Specify the test procedures and test cases that are easy to implement, easy to maintain and effectively verify the test requirements.



## 11.3 Software Testing Life Cycle

- Test design techniques
  - *During test design, the test basis documentation is analyzed in order to determine what to test i.e. test condition*
  - *A test condition is defined as an item that could be verified by one or more test cases*
  - *Test case consist of a set of input values, execution preconditions, expected results and expected post conditions; are developed and described by using test design techniques*



## 11.3 Software Testing Life Cycle

- Test design techniques
  - Expected results should ideally be defined prior to test execution
  - Test cases are put in an executable order, this is test procedure specification
  - If tests are run using test execution tool, the sequence of actions is specified in a test script
  - The test procedures and automated test scripts are subsequently formed into a test execution schedule



## 11.3 Software Testing Life Cycle

- Categories of test design techniques
  - White-box Based techniques
  - Black-box Based techniques
  - Experienced Based techniques



## 11.3 Software Testing Life Cycle


- Test development
  - It is the process of creating test procedures and test cases that verify the test requirements
  - Automated testing using tools
  - Manual testing





## 11.3 Software Testing Life Cycle

- Test execution
  - It is the process of running a set of test procedures against target software build of the application under test and logging the results.
- Test evaluation
  - It is the process of reviewing the results of a test to determine if the test criteria are being met.



## 11.4 Basic Knowledge about Software Testing Automation

- What is test automation?
  - **Test automation** is the use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions.




## 11.4 Basic Knowledge about Software Testing Automation

- **Automated testing**
  - Machine replacement of human
    - Frequent testing
    - Execute testing use same script
  - More testing
    - reduce the work of repetition and boredom




## 11.4 Basic Knowledge about Software Testing Automation

- Which type of test can be automated
  - Big difficultly testing
    - Performance, Stress, Configure
  - Consistency testing
    - Performance Regression
  - Repeat testing
    - How many File Open drivers are there?
    - Logging, etc.
  - Stability testing




## 11.4 Basic Knowledge about Software Testing Automation

- The advantages of automated testing
  - Speed up testing to accelerate releases
  - Allow testing to happen more frequently
  - Reduce costs of testing by reducing manual labor
  - Improve test coverage
  - Ensure consistency
  - Improve the reliability of testing
  - Allow testing to be done by staff with less skill
  - Define the testing process and reduce dependence on the few who know it



## 11.4 Basic Knowledge about Software Testing Automation

- Disadvantages of automated testing
  - High investment is needed in the tools and training
  - High man power requirement for test preparations
  - A lot of testing areas left uncovered



## 11.4 Basic Knowledge about Software Testing Automation

- Which type of test can be automated
  - Unit testing
  - Integration testing
  - System testing

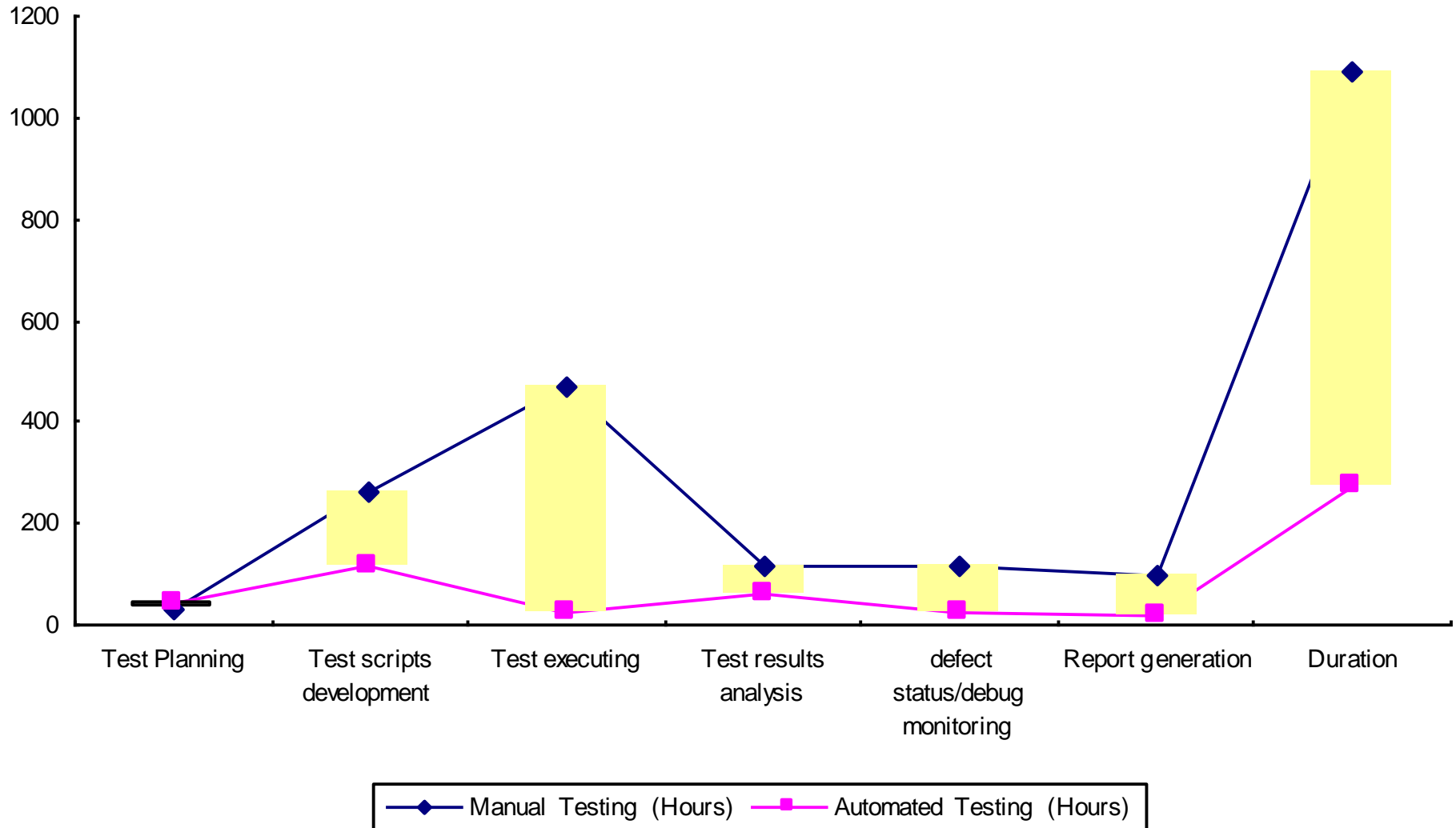
# 11.4 Basic Knowledge about Software Testing Automation

- Manual Testing vs. Automated Testing

Benefit	Automated Testing	Manual Testing
Productivity	Automation's greatest talent.	Very time consuming. Mistakes can be made.
Repeatability and consistency	The tests can be exactly reproduced and repeated.	It's difficult to reproduce the exact circumstances of a failure.
Speed of Execution	Fast	Slower
Maximizing Machine Resources	Tests can be run out of hours at no additional cost.	It is expensive to employ a team of testers out of hours.
Results Recording	Automatically recorded for all outcomes.	Only record failures.
Ad hoc and exploratory Testing	Not suitable	Ideally suited
Dynamic Applications	Automated scripts need a high level of maintenance	Easy to adapt to dynamic applications

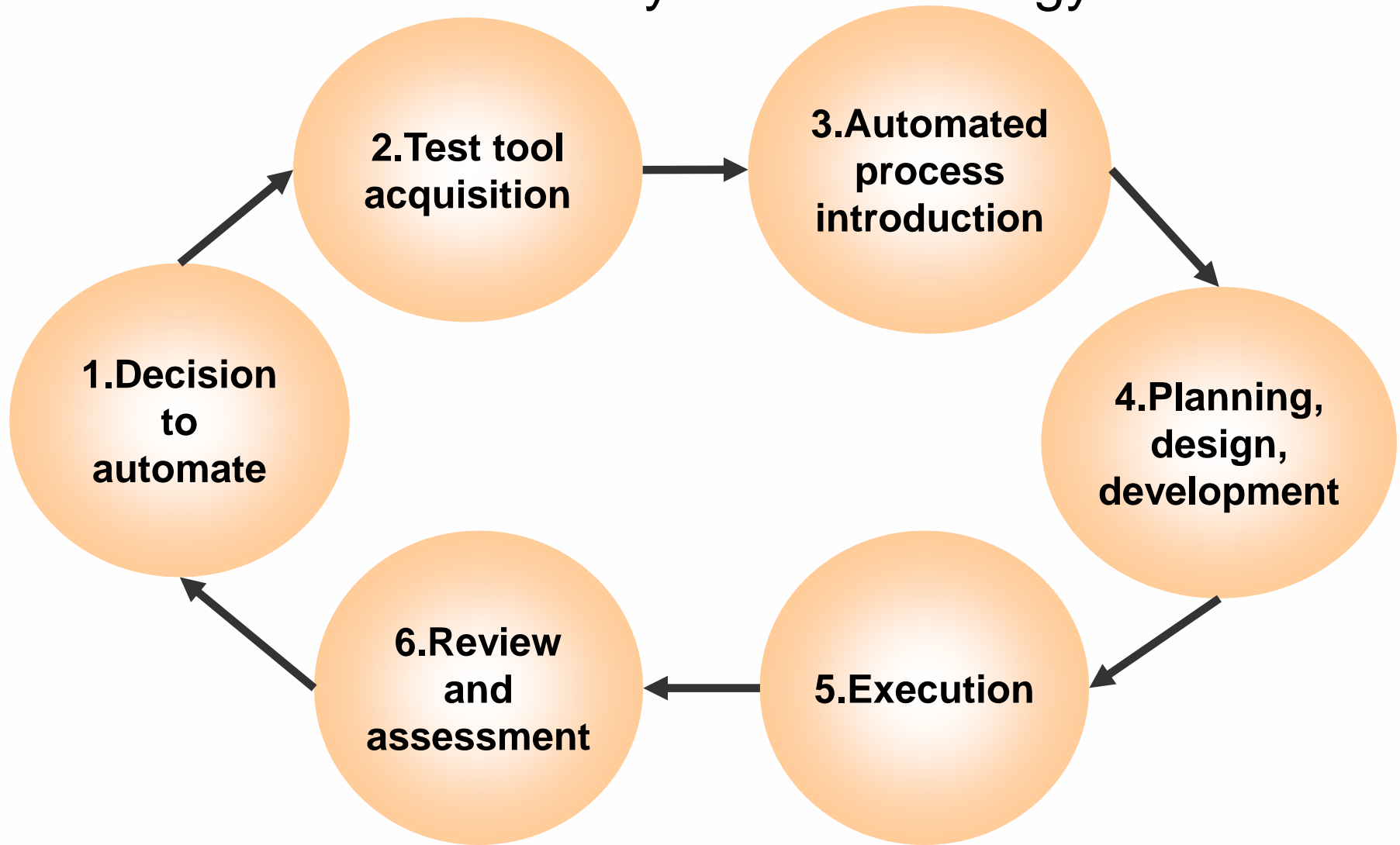


# 11.4 Basic Knowledge about Software Testing Automation



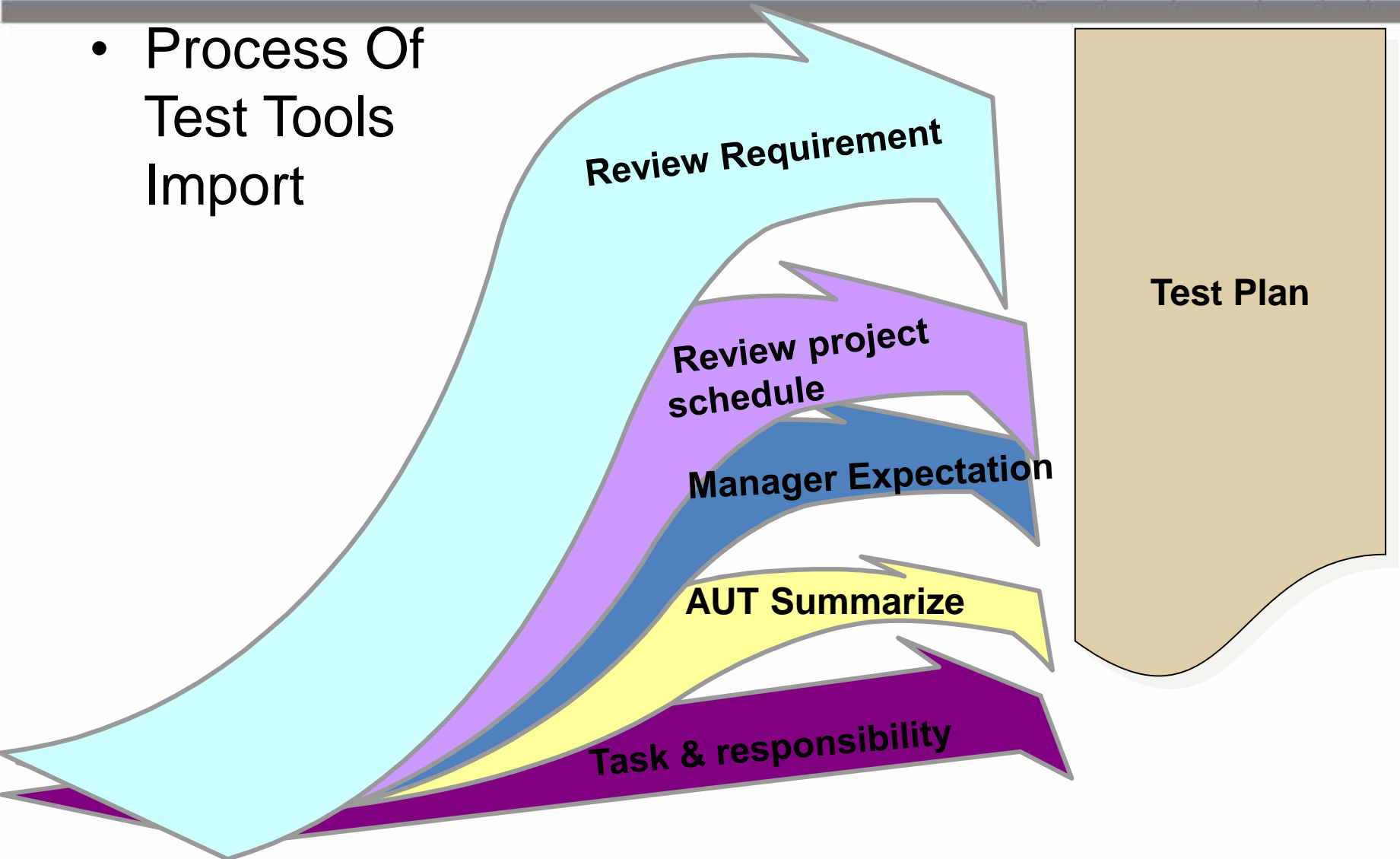
# 11.4 Basic Knowledge about Software Testing Automation

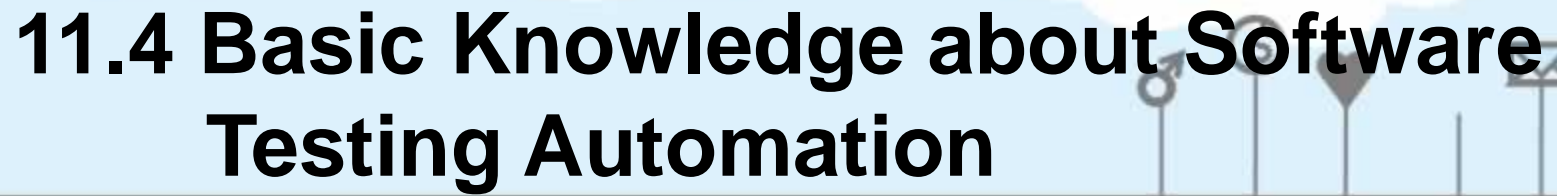
- Automated Test Lifecycle Methodology



# 11.4 Basic Knowledge about Software Testing Automation

- Process Of Test Tools Import





## 11.4 Basic Knowledge about Software Testing Automation

- Misconceptions
  - Our project schedules are too tight. Let's use automated testing.
  - Automation will eliminate the need for all manual testing.
  - We do not need any training.

# 11.5 Automated Testing Tools

Order number	Tool type	Description
1	Test Data generator	according to requirement/design/object model create testing process
2	Code (test) coverage analyzer code management	White-box testing
3	Memory leak check	Confirm whether application correctly manage memory resources
4	Measurement report tools	Analysis complexity of Data flow\construct\control flow, provide code size measurement according module\operator\code line
5	Available measure tools	User Configure, task analysis, develop prototype system....

# 11.5 Automated Testing Tools



Order number	Tool type	Description
6	test management tools	Provide some function of manage test in order to manage test document /defect status , etc
7	web test tools	Monitor ,measurement, test and conclude network performance
8	GUI test tool (record/replay)	Recode the dialog between user & interactive system
9	load/performance/stress test tools	Used in load/performance/stress testing
10	special test tool	Some test tools specific to special technology & form , e.g.. embedded systems testing



## 11.5 Automated Testing Tools

- Choosing Automation Tools
  - Ease of integration
  - Compatibility
  - Performance
  - Types of tests
  - Maintainability
  - Affordability (负担能力)

**Thank you !**

**Good Bye!**

