**B. Effective use of tools**

## **i. The main principles of tool selection**

* Organisational maturity
* Clear requirements and objective criteria
* Vendor support
* Coaching and mentoring internal requirements
* Training needs
* Cost-benefit ratio
* A proof-of-concept

## **ii. Pilot projects for Introducing a tool into an organisation**

* Learn more detail about the tool
* Evaluate how the tool fits with existing processes
* Decide on standard ways of using the tool
* Asset cost vs benefits

## **iii. Success factors for tools:**

* Rolling out tool incrementally
* Adapting and improving processes
* Providing training, coaching and mentoring
* Defining usage guidelines
* Gather usage information
* Monitoring tool use and benefits
* Providing support

**6. TOOL SUPPORT FOR TESTING**

**A. Test Tools**

A test tool is software product that supports one or more test activities, such as planning and control, specification, building initial files and data, test execution and test

## **i. Benefits**

* Improve the efficiency of test activities
* Automate activities that require significant resources when done manually
* Automate activities that cannot be executed manually.

## **ii. Risks**

* Unrealistic expectations for the tool
* Underestimating effort for the initial introduction
* Underestimating effort to achieve significant benefits
* Effort to maintain the test assets
* Over-reliance on the tool
* Neglecting interoperability between critical tools
* Poor vendor support
* Neglecting version control
* Risk of suspension of open-source / free tool project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ISTQB Classification** | **Test Tool Type** | **Test process where it is most beneficial** | **Brief Description** | **Use by who?** | **Example** |
| **Tool support for management of testing and testware** | Test management and application life cycle management (ALM) | Throughout ? |  |  |  |
| Defect Management | Test Execution (?) | Used to store defect reports and maintain the details about the defect as it progresses through its life cycle. Database structure to manage details of defects. | Testers, Test Managers |  |
| Requirements Management | Test Design | Used to record, manage, and prioritise the requirements of a system. Can also be used to manage changes to requirements. Including traceability function, which allows to identify which items need to change if requirements change. | Testers (retrieve requirements)/Bas (record, manage, prioritise requirements) |  |
| Configuration Management |  | Tools designed primarily for managing the version of different software (and hardware) components. | ??? |  |
| Continuous Integration | Execution ? | Ensures that correct versions of different programs are integrated into the Daily Build that is deployed into the test environment. |  |  |
| **Tool support for static testing** | Tools that support reviews |  | Provide a framework for reviews or inspections; maintaining information about the process such as rules and checklists, the ability to record communicate and retain comments and defects, traceability functions to enable changes to deliverables that may be affected by the change |  |  |
| Static Analysis | Test implementation () | Analyse code before it’s executed to identify defects early. Can be used to enforce programming standards. | Developers |  |
| **Tool support for test design and implementation** | Test design | Test Implementation | The level of automation can vary and depends on the characteristics of the tool itself and the way in which the test basis is recorded in the tool.  For example, some tools allow specifications or requirements to be specified in a formal language. This can allow test cases with inputs and expected results to be generated. Some tools (sometimes known as test frames) merely generate a partly filled template from the requirement specification held in narrative form. The tester will then need to add to the template and copy and edit as necessary to create the test cases required. | Testers |  |
| Model-based testing | Test analysis/design | ??? | Developers |  |
| Test data preparation | Test Execution? | Used to manipulate data so that the environment is in the correct state for a test to be run. Change field values in databases, data files, etc. | Testers/Developers |  |
| Test-driven development | Test Design to Execution |  | Developers/Testers |  |
| Acceptance test-driven development (ATDD) and behaviour-driven development (BDD) |  |  |  |  |
| **Tool support for execution and logging** | Test Execution (to run regression tests) | Test Execution | Allow for test scripts to be run automatically, may provide functionality to allow scrips to identify particular GUI objects, etc. | Testers |  |
| Test Harness | Test Implementation | Simulates a small section of the environment in which the software will operate, stubs and divers etc. | Developers |  |
| Unit test framework | Test Implementation/Execution | Generally, a more robust and re-usable version of a test harness, which is usually a standalone device built in-house. Generally able to support multiple test harnesses and may provide additional dev support such as debugging capabilities. | Developers |  |
| Coverage | Test Completion(?) | Measure the percentage of the code covered across white box measurement techniques. |  |  |