**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) | All activities | Supports test management and control  Often has several capabilities  Examples: testware management, test scheduling, results logging, progress tracking, incident management, test reporting. | Testers |  |
| Defect Management | Test execution  (Primarily. As this is where most defects are found) | Recording and status tracking of defects | Various  (Testers) |  |
| Requirements Management | Test Analysis  (What is in scope?) | Supports recording of requirements, requirements attributes and annotation.  Facilitates traceability.  Some provide static analysis tools (consistency checking, requirements rules violations) | Various  (BAs) |  |
| Configuration Management | Test Implementation  (Correct software version(s) in the test environment) | Supports identification and control of configuration items.  Status over changes and versions  Release of baselines of configuration items  Configuration item: Single entity aggregate of work products for CM | Various  (Release managers, developers) |  |
| Continuous Integration | Test Implementation  (Correct software version(s) in the test environment) | Compiles and links daily builds  Keeps test environment up to date with latest changes providing consistent system. | Developers |  |
| **Tool support for static testing** | Tools that support reviews | Most  (Reviews take place in all phases)  (Test execution unusual) | Typically: review planning, review tracking support, communication support, collaborative reviews, collecting and reporting of metrics | Various |  |
| Static Analysis | Test Implementation | Static code analysis  Checks for coding standards, quality metrics, data flow anomalies | Developers |  |
| **Tool support for test design and implementation** | Test design | Test Design | Generates test inputs from a spec held in a CASE (Computer Aided Software Engineering) tool repository  Specs from sources like requirements management tool, specified test conditions, or from code | Testers |  |
| Model-based testing | Test Analysis  Test Design  Test Implementation | Creates, amends and verifies of software or system models | Developers |  |
| Test data preparation | Test Implementation  (Test data prep is a key part of TI) | Allows for test data:   * selection from a database * creation or generation * manipulation and editing | Various |  |
| Test-driven development | Test Design  (some tools can be used in TI and TE) | Develop (and automate) test cases before the software they are to test | Developers |  |
| Acceptance test-driven development (ATDD) and behaviour-driven development (BDD) | Test Design  (some tools can be used in TI and TE) | TDD tools that have been configured for use by testers | Testers |  |
| **Tool support for execution and logging** | Test Execution (to run regression tests) | Test Execution | Run tests against test items  Compare with expected results and postconditions | Testers | VS Test Explorer? |
| Test Harnesses | Test Execution | Test environment with the stubs and drivers needed to run a test | Developers |  |
| Unit test framework | Test Execution | Can support multiple test harnesses  More robust and reusable than test harness  Additional support e.g., debugging. | Developers |  |
| Coverage | Test Execution | Objective measures of coverage e.g., statements and branches | Developers | Coverlet |