**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 | Capabilities include testware management, scheduling of tests, logging of results, progress tracking, incident management and test reporting | Testers |  |
| Defect Management | Test execution | Facilitates recording and status tracking of defects | Testers |  |
| Requirements Management | Test analysis | Supports recording of requirements | Business Analysts |  |
| Configuration Management | Test implementation | Supports the identification and control of configuration items | Various |  |
| Continuous Integration | Test implementation | Provides regular and frequent builds | Developers |  |
| **Tool support for static testing** | Tools that support reviews | Various, except execution | Provides support to review process | Various |  |
| Static Analysis | Test implementation | Carries out static code analysis | Developers |  |
| **Tool support for test design and implementation** | Test design | Test design | Generates test inputs from a specification held in a CASE tool repository | Testers |  |
| Model-based testing | Test analysis, Test design, Test implementation | Supports the creation, amendment and verification of models | Developers |  |
| Test data preparation | Test implementation | Enables data to be selected from existing databases | Various |  |
| Test-driven development | Test design | A way of developing tests before software is developed to run | Developers |  |
| Acceptance test-driven development (ATDD) and behaviour-driven development (BDD) | Test design | An enhancement to TDD tools configured so they can be used by testers | Testers |  |
| **Tool support for execution and logging** | Test Execution (to run regression tests) | Test execution | Executes tests and evaluates outcomes | Testers |  |
| Test Herness | Test execution | Composed of stubs and drivers needed to conduct a test | Developers |  |
| Unit test framework | Test execution | Supports multiple test harnesses | Developers |  |
| Coverage | Test execution | Provides objective measures of what structured elements have been exercised by a test suite | Developers |  |