

6.1. Tool Support for Testing

Test tools support and facilitate many test activities. Examples include, but are not limited to:

- Management tools – increase the test process efficiency by facilitating management of the SDLC, requirements, tests, defects, configuration
- Static testing tools – support the tester in performing reviews and static analysis
- Test design and implementation tools – facilitate generation of test cases, test data and test procedures
- Test execution and coverage tools – facilitate automated test execution and coverage measurement
- Non-functional testing tools – allow the tester to perform non-functional testing that is difficult or impossible to perform manually
- DevOps tools – support the DevOps delivery pipeline, workflow tracking, automated build process(es), CI/CD
- Collaboration tools – facilitate communication
- Tools supporting scalability and deployment standardization (e.g., virtual machines, containerization tools)
- Any other tool that assists in testing (e.g., a spreadsheet is a test tool in the context of testing)

6.2. Benefits and Risks of Test Automation

Simply acquiring a tool does not guarantee success. Each new tool will require effort to achieve real and lasting benefits (e.g., for tool introduction, maintenance and training). There are also some risks, which need analysis and mitigation.

Potential benefits of using test automation include:

- Time saved by reducing repetitive manual work (e.g., execute regression tests, re-enter the same test data, compare expected results vs actual results, and check against coding standards)
- Prevention of simple human errors through greater consistency and repeatability (e.g., tests are consistently derived from requirements, test data is created in a systematic manner, and tests are executed by a tool in the same order with the same frequency)
- More objective assessment (e.g., coverage) and providing measures that are too complicated for humans to derive
- Easier access to information about testing to support test management and test reporting (e.g., statistics, graphs, and aggregated data about test progress, defect rates, and test execution duration)
- Reduced test execution times to provide earlier defect detection, faster feedback and faster time to market
- More time for testers to design new, deeper and more effective tests

Potential risks of using test automation include:

- Unrealistic expectations about the benefits of a tool (including functionality and ease of use).
- Inaccurate estimations of time, costs, effort required to introduce a tool, maintain test scripts and change the existing manual test process.
- Using a test tool when manual testing is more appropriate.
- Relying on a tool too much, e.g., ignoring the need of human critical thinking.
- The dependency on the tool vendor which may go out of business, retire the tool, sell the tool to a different vendor or provide poor support (e.g., responses to queries, upgrades, and defect fixes).
- Using an open-source software which may be abandoned, meaning that no further updates are available, or its internal components may require quite frequent updates as a further development.
- The automation tool is not compatible with the development platform.
- Choosing an unsuitable tool that did not comply with the regulatory requirements and/or safety standards.