**Manual Testing**

* A human tester checks the website or app **step-by-step**.
* Example: You open Amazon in your browser, search for “shoes,” click a product, add it to the cart, and check if it’s added correctly — all by hand.

**Automation Testing**

* A computer program (like Selenium) does the same steps **automatically**.
* Example: You write a script that opens Amazon, searches for “shoes,” clicks a product, adds it to the cart, and verifies the cart — without you touching the mouse or keyboard.

**Key difference:**

* **Manual** = slow but flexible, good for small or new features.
* **Automation** = fast, repeatable, good for large or repetitive testing.

**How testing happens in sprints**

* **Sprint 0** → No automation yet, no manual testing done.
* **Sprint 1** → You do **manual testing** for sprint 1 features, and **automation** for sprint 0 features.
* **Sprint 2** → You do **manual testing** for sprint 2 features, and **automation** for sprint 0 + sprint 1 features.

This way, automation **always tests previous sprints**, while manual testing is done for **current sprint**.

**Regression Testing**

* Means checking that new changes **don’t break old features**.
* Example: If you add “Wishlist” to a shopping site, you make sure login, cart, and payment still work.

**Why Automation helps in Regression**

* Manual regression for 50 sprints is **slow and repetitive**.
* Automation can:
  + Test faster
  + Use **multiple sets of data** easily
  + Run on **different browsers** (Chrome, Firefox, Edge, etc.)

**Automation Framework Types**

**1. Keyword Driven Framework**

* Idea: Test steps are written using keywords instead of code.
* Example:

| Step | Keyword | Test Data |
| --- | --- | --- |
| 1 | LOGIN | username, password |
| 2 | CLICK | "Add to Cart" button |
| 3 | VERIFY | "Cart has 1 item" |

* Here, LOGIN, CLICK, and VERIFY are predefined keywords in the framework.

**2. Data Driven Framework**

* Idea: Same steps are used, but test data changes for multiple runs.
* Example: Testing login with 5 sets of credentials:

| Username | Password | Expected Result |
| --- | --- | --- |
| user1 | pass1 | Success |
| user2 | pass2 | Success |
| wrong1 | pass3 | Fail |

* You don’t rewrite steps, you just run them with different data.

**3. Hybrid Framework**

* Idea: Uses both keywords and multiple data sets together.
* Example:
  + Keywords: LOGIN, CLICK, VERIFY
  + Data: Multiple usernames/passwords + different buttons to click
  + You can test login, add to cart, wishlist, etc., with many data sets — all using the same keywords.

**Testing & QA Related Tools**

**1. Test Management Tools**

**Definition:**  
Software that helps QA teams plan, execute, track, and report on testing activities, including managing test cases, test execution, requirements, and defects.  
**Examples:**

* **Jira + Zephyr** (Atlassian)
* **TestRail**
* **qTest** (Tricentis)
* **PractiTest**
* **Xray for Jira**

**2. Test Automation Tools**

**Definition:**  
Frameworks or platforms that automatically execute test scripts, compare results with expected outcomes, and generate reports—reducing manual testing effort.  
**Examples:**

* **Selenium** (Web automation)
* **Cypress** (JavaScript-based automation)
* **Playwright**
* **TestComplete** (SmartBear)
* **Appium** (Mobile automation)

**3. API Testing Tools**

**Definition:**  
Tools used to test APIs for functionality, reliability, performance, and security by sending requests and validating responses.  
**Examples:**

* **Postman**
* **SoapUI**
* **Katalon Studio**
* **REST Assured**
* **JMeter (for API load testing)**

**4. CI/CD Tools**

**Definition:**  
Continuous Integration and Continuous Deployment tools automate the build, test, and deployment pipeline, ensuring faster and reliable software delivery.  
**Examples:**

* **Jenkins**
* **GitLab CI/CD**
* **CircleCI**
* **Azure DevOps**
* **Travis CI**

**5. Performance Testing Tools**

**Definition:**  
Tools that simulate real-world load on applications to measure speed, scalability, stability, and responsiveness.  
**Examples:**

* **Apache JMeter**
* **LoadRunner** (Micro Focus)
* **Gatling**
* **BlazeMeter**
* **k6**

**6. Security Testing Tools**

**Definition:**  
Tools used to identify vulnerabilities, weaknesses, and potential security threats in applications, networks, and systems.  
**Examples:**

* **OWASP ZAP**
* **Burp Suite**
* **Nessus**
* **Acunetix**
* **Fortify**

**7. Test Reporting Tools**

**Definition:**  
Tools that generate detailed test execution reports, defect status summaries, and metrics for analysis and decision-making.  
**Examples:**

* **Allure Report**
* **ExtentReports**
* **ReportPortal.io**
* **TestRail Reporting**
* **Kibana (for visualizing test data)**

**Project & Team Management Tools**

**1. Task and Project Management**

* **Definition:** Organize, assign, and track project tasks and deadlines.
* **Tools:** Jira, Trello, Asana, ClickUp, Monday.com

**2. Communication and Collaboration**

* **Definition:** Enable real-time communication and teamwork across distributed teams.
* **Tools:** Slack, Microsoft Teams, Zoom, Google Meet, Discord

**3. Time Tracking and Reporting**

* **Definition:** Monitor working hours, productivity, and generate reports for billing or project evaluation.
* **Tools:** Toggl Track, Clockify, Harvest, Time Doctor, Hubstaff

**4. Document and File Management**

* **Definition:** Store, organize, and share documents and files securely.
* **Tools:** Google Drive, Dropbox, SharePoint, Confluence, Notion

**5. Team Performance and Analytics**

* **Definition:** Measure and analyze team performance, efficiency, and productivity trends.
* **Tools:** Jira Insights, Asana Reporting, Tableau, Power BI, Wrike Analytics

**6. Agile and Scrum Management**

* **Definition:** Manage agile workflows, sprints, and backlogs for software development teams.
* **Tools:** Jira Agile, Rally (CA Agile Central), VersionOne, Scrumwise, Taiga

**Documentation Tools**

A documentation tool is software used to create, organize, and share information or project details clearly.

**Markdown Editors**

* **Definition:** Simple text editors for writing structured documents using Markdown syntax.
* **Tools:** Typora, Obsidian, StackEdit, Dillinger, Mark Text

**Documentation Platforms**

* **Definition:** Platforms for creating, publishing, and maintaining documentation.
* **Tools:** Confluence, Notion, GitBook, Read the Docs, Slab

**Static Site Generators**

* **Definition:** Tools that convert plain text and templates into static websites.
* **Tools:** Jekyll, Hugo, Docusaurus, MkDocs, Gatsby

**Visual Documentation Tools**

* **Definition:** Tools for creating diagrams, workflows, and visual explanations.
* **Tools:** Lucidchart, Miro, Draw.io, Canva, Whimsical

**Performance Monitoring Tools**

Tools that track how well systems, applications, and networks are performing so issues can be detected and fixed quickly.

**Application Performance Monitoring (APM) Tools**

* **Tools:** New Relic, AppDynamics, Dynatrace, Datadog APM, Elastic APM

**Network Performance Monitoring Tools**

* **Tools:** SolarWinds NPM, PRTG Network Monitor, Nagios, Zabbix, Wireshark

**Infrastructure Monitoring Tools**

* **Tools:** Prometheus, Grafana, Nagios XI, Zabbix, Sensu

**End-User Monitoring (EUM) Tools**

* **Tools:** Pingdom, Catchpoint, Dynatrace RUM, New Relic Browser, Sematext Synthetics

**Risk Assessment Tools**

Tools that help identify, analyze, and manage risks so organizations can take preventive action**.**

**IT and Cybersecurity Risk Assessment Tools**

* **Tools:** Qualys, Nessus, Rapid7 InsightVM, RiskSense, OpenVAS

**Management and Assessment Tools**

* **Tools:** Resolver, RiskWatch, Archer, LogicManager, AuditBoard

**Project and Operational Risk Assessment Tools**

* **Tools:** Wrike, Risk Register (Jira plugin), nTask, ProjectManager.com, SpiraPlan

**Compliance and Regulatory Risk Assessment Tools**

* **Tools:** MetricStream, Compliance 360, NAVEX Global, VComply, Convercent