



Narzędzia do testowania:

- **JUnitPerf** – kolekcja klas JUnit służących do dekorowania klas Javy, których wydajność lub skalowalność chcemy oszacować
- **JMeter** – zaawansowane narzędzie rozwijane przez Fundację Apache służące do wykonywania testów obciążeniowych i wydajnościowych różnych typów obiektów (plików, servletów, klas Javy, serwerów HTTP i FTP, ...), pozwalające na symulowanie działania wirtualnych użytkowników oraz rejestrowanie czasu ich reakcji
- **Cactus** – Apache Jakarta framework przeznaczony do testowania logiki biznesowej oraz do wykonywania testów integracyjnych i funkcjonalnych serwerowego kodu Javy
- **FIT/FitNesse** – framework do przeprowadzania testów akceptacyjnych; testy tworzone są automatycznie na podstawie opisu dostarczanego przez użytkownika

Narzędzia do testowania:

- **xUnit** – rodzina frameworków do przeprowadzania testów jednostkowych, oparta na wspólnej architekturze zaprojektowanej przez Kenta Becka i Ericha Gamma, oferująca wsparcie dla wielu języków programowania: **SUnit** (Smalltalk), **JUnit** (Java), **NUnit** (C#), **CPPUnit** (C++), **PyUnit** (Python), **fUnit** (Fortran), **JSUnit** (JavaScript), **XMLUnit** (zawartości plików XML), **SQLUnit** (procedury składowane SQL), **HTTPUnit** (zawartości stron webowych), ...
- **Canoo** – framework do testowania aplikacji webowych, umożliwiający testowanie wybranych (opisanych w XML-u) elementów stron
- **Selenium** – framework do testowania aplikacji webowych, pozwalający na nagrywanie/odtwarzanie testów lub ich przygotowywanie/modyfikowanie w dedykowanym języku

Narzędzia do testowania:

- **Abbot** – framework przeznaczony do testowania GUI Javy pisanego w AWT lub w Swingu
- **SwingUnit, JFCUnit, Marathon** – frameworki przeznaczone do testowania GUI Javy pisanego w Swingu
- **Cucumber** – framework do zautomatyzowanych testów akceptacyjnych pisanych zgodnych z ideą BDD (*Behavior-Driven Development*); scenariusze testowe pisane w języku naturalnym (format wykorzystujący klauzule *Given-When-Then*) są interpretowane zgodnie z przygotowanymi definicjami w języku

```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

Scenario: Add two numbers
  Given I have entered 50 into the calculator
  And I have entered 70 into the calculator
  When I press add
  Then the result should be 120 on the screen
```

Koszty automatyzacji procesu testowania:

- **Koszt implementacji** – jednorazowy koszt związany z przygotowaniem środowiska testowego oraz skryptów testowych
- **Koszt eksploatacji środowiska testowego** – koszt związany z ingerencją człowieka bieżącą obsługą procesu testowania (utrzymanie niezawodności testów, analizę wyników)
- **Koszt dostosowania testów** – koszt związany z koniecznością dostosowania testów do zmian testowanego oprogramowania (np. implementacja nowych testów w przypadku dodania do oprogramowania nowych funkcjonalności)

Dobór danych do testowania:

- wykorzystanie wiedzy dziedzinowej
- wykorzystanie doświadczenia testera
- zastosowanie generatora losowego