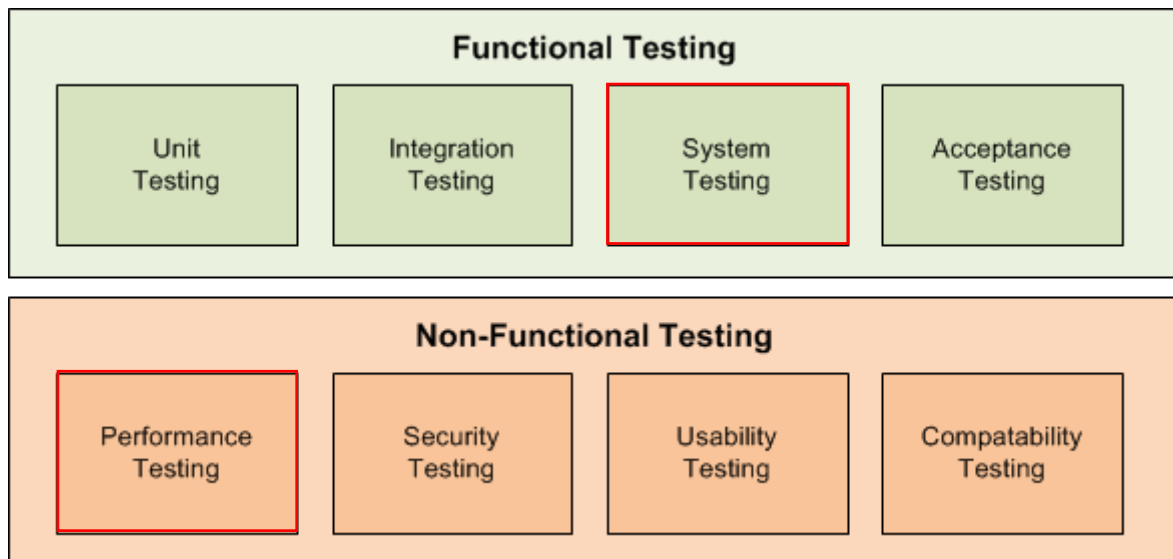
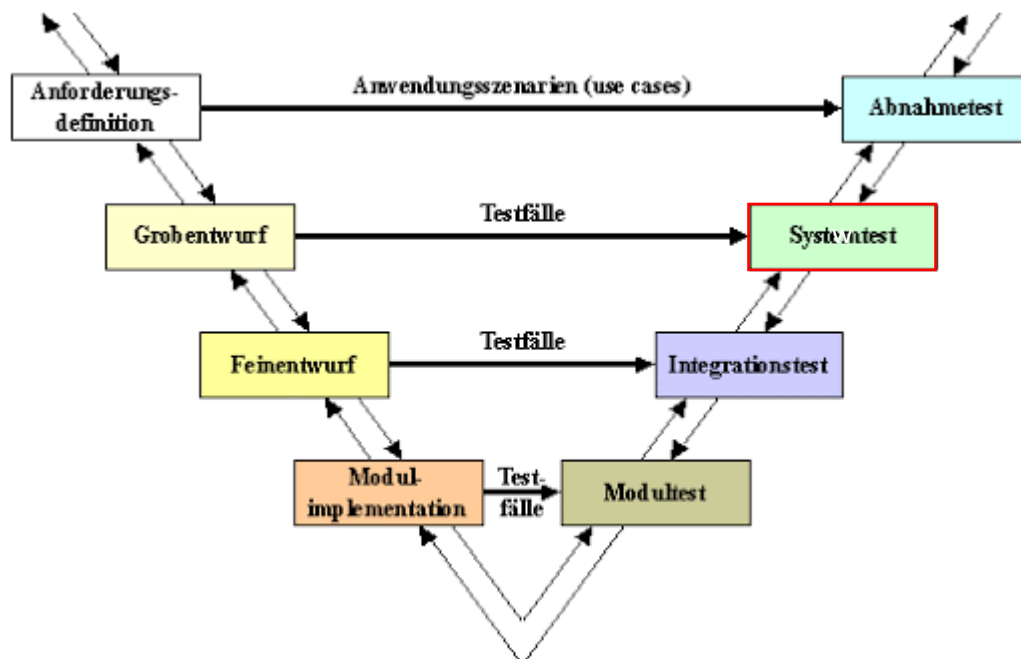


# Load & Stresstest

Load und Stress Test sind beides Teil vom Non-Functional Testing. In diesem Teil bilden sie dann gemeinsam die Gruppe der Performance Tests.



Die Test werden im Falle des V-Modells zum Beispiel im Zuge der Systemtests durchgeführt. Es muss nämlich das gesamte System bereits funktionsfähig sein um diese Test durchzuführen.





## LOAD & STRESS TEST

Diese Test werden hauptsächlich verwendet um folgende Punkte einer fertigen Software, eines Netzwerkes oder eines Servers zu überprüfen.

- ressource usage
- scalability
- reliability
- Performance issues
- achitecture issues

<https://www.dev-insider.de/was-ist-performance-testing-a-810708/>

<https://www.softwaretestinghelp.com/what-is-performance-testing-load-testing-stress-testing/>

## Load Test

Load-Testing gehört dabei zu den Anforderungen, denen Apps und Websites beständig ausgesetzt sein sollten, um im Nutzungsfall für jeden User verfügbar zu sein. Ein typischer Load Test fügt etwa alle 60 Sekunden 50 virtuelle Nutzer hinzu bis die Zahl von 1.000 virtuellen Usern erreicht ist und lässt diese virtuellen Nutzer weiter Anfragen senden.

### **Aufgaben:**

- Aufzeigen von Fehlern bezüglich Themen wie: buffer overflow, memory leaks und mismanagement of memory.
- Aufzeigen von load balancing Problemen
- Aufzeigen von bandwidth Problemen
- Aufzeigen wenn zu wenig Kapazitäten beim aktuellen System vorhanden sind
- Um die Obergrenze aller Komponenten der Applikation (database, hardware, network, etc.) festzustellen.

## Stress Test

Im Gegensatz dazu wird Stress Testing durchgeführt, um zu ergründen, wie sich Software unter Höchstanforderungen verhält. Load Testing prüft quasi die Breite, Stress Testing die Spitze der Anforderungen. Stress Testing ist nicht nur wichtig, um die Performance, Response Time und Erholung des Systems zu testen, sondern auch um Sicherheitslücken zu schließen.

### **Aufgaben:**

- post-crash reports zu analysieren, um das Verhalten der Software bei Überlastung zu überprüfen
- Nach einem stress Test sollte die Software genauso laufen wie davor, dann ist er erfolgreich(das Programm darf nicht abstürzen)

# Beispiel

Load und Stresstest werden in spezielle Programme ausgeführt. Sie simulieren eine gewisse Anzahl von Usern die bestimmte Handlungen durchführen. Zum Beispiel würde der Load-Test für ein Shop-Webseite sein, das zum Beispiel 1000 User simuliert werden, die nach zwei unterschiedlichen Waren suchen und eines davon in den Warenkorb verschieben. Durch diesen Test wird dann überprüft ob es für alle Benutzer möglich ist Ihre Transaktionen zu vollenden.

Erweitert man die Zahl der User jetzt auf zum Beispiel 100000 kommt man in den Bereich des Stresstest. Bei der Beispiel Seite wäre es möglich bei zu vielen Anfragen den Benutzer auf einen späteren Zeitpunkt zu verweisen. Es soll bei diesem Test aber viel Wert darauf gelegt werden das trotz der kurzzeitigen Überlastung die Webseite danach, genauso wie davor weiter funktioniert.

## Implementierung

<https://testguild.com/open-source-performance-testing-tools/>

- JMeter
- Gatling
- Taurus
- Siege
- The Grinder
- Locust
- Fiddler with BlackWidow and Watcher
- nGrinder
- k6
- Tsung
- Bees with Machine Guns

# Gatling

```
class Example extends Simulation{
  val httpProtocol = http
    .baseUrl("http://localhost:8080")

  val mySzenario = Szenario("my Szenario").exec(...)

  setUp(
    mySzenario.inject(
      constantUsersPerSec(40) during (20 minutes)
    )
  ).protocols(httpProtocol)
}
```

```
val search = exec(http("Home")
  .get("/"))
  .pause(7)
  .exec(http("Search")
    .get("/computers?f=macbook"))
  .pause(2)
  .exec(http("Select")
    .get("/computers/6"))
  .pause(3)
```

Programmiersprache => Scala

Firma => Gatling Corp

Use Case => Loadtest REST-Endpoints/Websites

Ergebnis => HTML reports

Möglichkeiten

- viele User gleichzeitig simulieren
- verschiedene User, verschiedene Dinge machen lassen
- User mit Daten aus File versorgen