



Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Η/Υ

Εθνικό Μετσόβιο Πολυτεχνείο

---

# Τεχνολογία Λογισμικού

7ο / 9ο Εξάμηνο 2018-19

---

Ν.Παπασπύρου, Αν.Καθ. ΣΗΜΜΥ, [nickie@softlab.ntua.gr](mailto:nickie@softlab.ntua.gr)

Β.Βεσκούκης, Αν.Καθ. ΣΑΤΜ, [v.vescoukis@cs.ntua.gr](mailto:v.vescoukis@cs.ntua.gr)

Κ.Σαΐδης, ΠΔ 407, [saiko@softlab.ntua.gr](mailto:saiko@softlab.ntua.gr)

# Έλεγχος και επαλήθευση λογισμικού

# Δύο κύρια ερωτήματα

1. Πώς εξασφαλίζουμε ότι το λογισμικό είναι σύμφωνο με τις απαιτήσεις και προσδοκίες των χρηστών (λειτουργικές απαιτήσεις);
2. Πώς διαφυλάσσουμε την ποιότητα του λογισμικού (μη λειτουργικές απαιτήσεις);

## Σε πολλούς άξονες:

- Χρηστικότητα / ευχρηστία
- Επεκτασιμότητα / βιωσιμότητα
- Ασφάλεια / ακεραιότητα δεδομένων
- Αποδοτικότητα / αποκρισιμότητα
- κ.λπ.

# Έλεγχος, επαλήθευση και επιβεβαίωση λογισμικού

## Έλεγχος

- Επιθεωρήσεις (Reviews) - στατικός έλεγχος
- Στατική ανάλυση (Static analysis) - στατικός έλεγχος
- Δοκιμασίες ελέγχου (Test cases) - δυναμικός έλεγχος

**Επαλήθευση**

Παράγουμε το λογισμικό σωστά;

**Επιβεβαίωση**

Παράγουμε το σωστό λογισμικό;

# Ελαττώματα (defects, bugs)

There are two ways to write error-free programs; only the third one works.

Alan J. Perlis

# Ελαττώματα (defects, bugs)

Σοβαρότητα (severity)

- Critical
- Major
- Minor
- Trivial



# Στατικός vs δυναμικός έλεγχος

Static testing involves verification, whereas dynamic testing also involves validation.

# Στατική ανάλυση κώδικα

Static analysis tools are generally used by developers as part of the development and component testing process. The key aspect is that the code (or other artefact) is not executed or run but the tool itself is executed, and the source code we are interested in is the input data to the tool.

# Εργαλεία για το JVM

## Γενικής χρήσης

- Checkstyle
- FindBugs
- PMD
- JDepend
- Jacoco
- ErrorProne
- SonarQube
- κ.ά

Ειδικής χρήσης εργαλεία ανάλογα με την περίπτωση

# Εκτέλεση σεναρίων / δοκιμασιών ελέγχου

Actually executing programmed code with a given set of test cases  
(the program itself is run)

# Εργαλεία για το JVM

- JUnit (unit testing)
- TestNG (unit testing)
- Mockito (mockup testing)
- Selenium (web browser automation)
- Jenkins (Continuous integration)
- Travis (Continuous integration)
- Spock (specifications)



# Κύκλος ζωής δοκιμασιών (test life-cycle)

- Επισκόπηση απαιτήσεων
- Σχεδιασμός δοκιμασιών
- Καθορισμός περιβάλλοντος δοκιμασιών
- Εκτέλεση δοκιμασιών
- Παραγωγή αναφορών

# Πλάνο δοκιμασιών (test plan)

- Έγγραφο, επιμέρους ή συνολικό (π.χ. master test plan)
- Τι θα ελεγχθεί και πώς (προσέγγιση)
- Παραδοτέα
- Εκτιμήσεις χρόνου/κόστους
- Προγραμματισμός, προσωπικό, εκπαίδευση, έλεγχοι/αποδοχή, κτλ.



# Σενάριο δοκιμής

## Test case

- Συνθήκες ή μεταβλητές που καθορίζουν την ορθή συμπεριφορά του χαρακτηριστικού που υπόκειται σε έλεγχο
- Συνήθως γράφεται απευθείας σε κάποιο εργαλείο

# Test scripts

- Αυτοματοποιημένοι έλεγχοι (test automation)
- Continuous integration (CI)

# Βασικά επίπεδα δοκιμασιών

Από το μικρό στο μεγάλο:

- Μοναδιαία δοκιμασία (unit test)
- Δοκιμασία ολοκλήρωσης (integration test)
- Δοκιμασία συστήματος (system test)
- Δοκιμασία αποδοχής (acceptance test)

# Μέθοδοι εκτέλεσης δοκιμασιών

- Black box testing (behavioral testing)
  - Έλεγχος της "εξωτερικής" συμπεριφοράς ενός συστατικού ως συνόλου (π.χ. εκτελώντας πειράματα στο public interface του)
- White box testing (structural testing)
  - Έλεγχος της "εσωτερικής" δομής / συμπεριφοράς ενός συστατικού (π.χ. εκτελώντας πειράματα στην εσωτερική του κατάσταση)
- Gray box testing (συνδυασμός)

# Unit testing

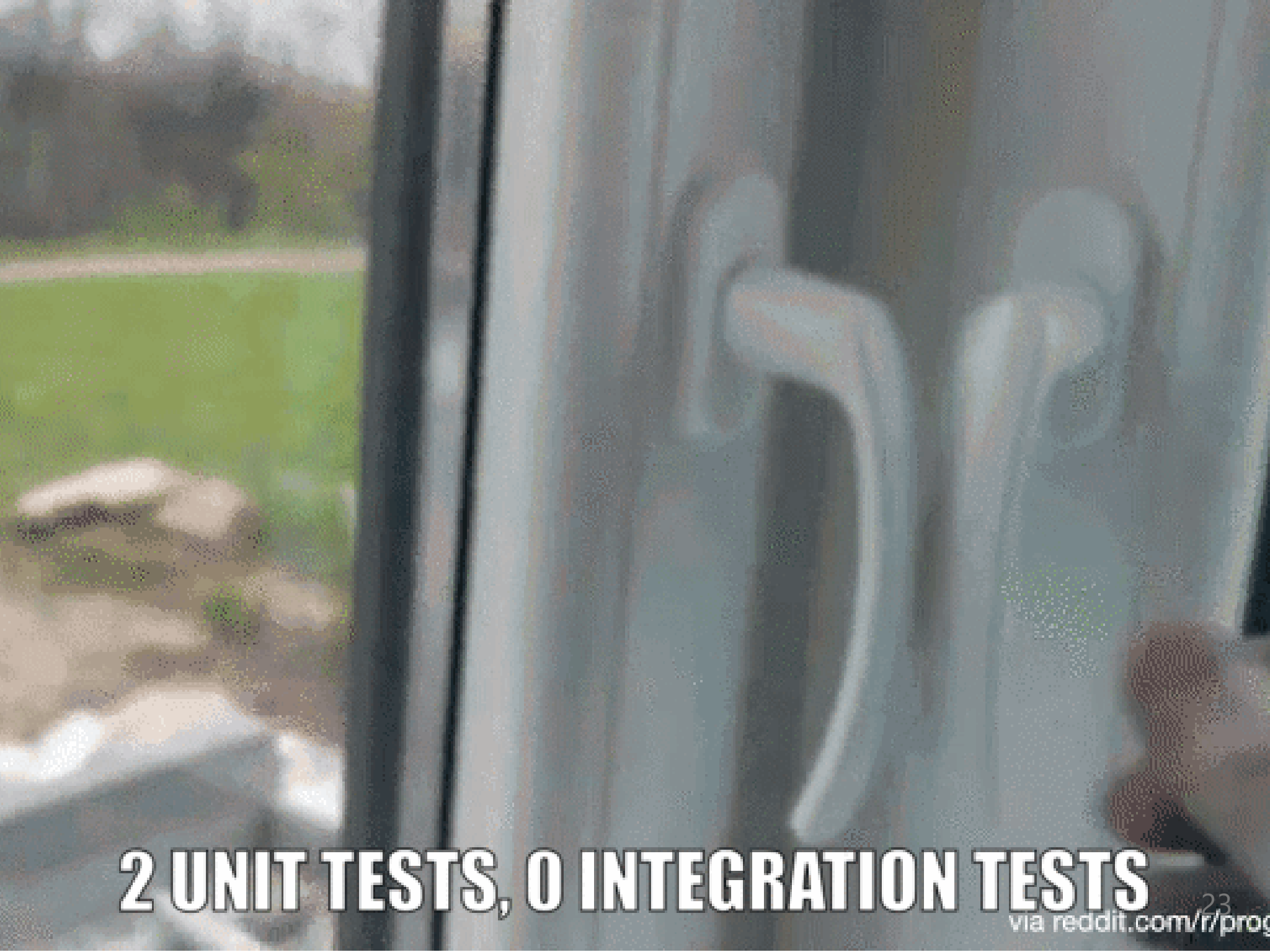
## Έλεγχος μοναδιαίων ενοτήτων του λογισμικού

- Το ελάχιστο στοιχείο του λογισμικού που μπορεί να ελεγχθεί (π.χ. μέθοδος)
- White box testing (όλες οι μέθοδοι, public, private, κ.λπ)
- Δειγματοληψία των πιθανών εισόδων και δοκιμή τους αν παράγουν αστοχία (σφάλμα, λανθασμένη έξοδος)

# Integration testing

## Έλεγχος "συνδυασμού" πολλών ενοτήτων

- Διαλειτουργικότητα μεταξύ ενοτήτων (όχι αναγκαστικά μοναδιαίων)
- Είτε white, είτε black, είτε gray box testing



**2 UNIT TESTS, 0 INTEGRATION TESTS**

via [reddit.com/r/prog](https://reddit.com/r/prog)<sup>23</sup>

# System testing

## Έλεγχος του συστήματος συνολικά

- Ως ολότητα σε σχέση με το λειτουργικό του περιβάλλον
- Δικτύωση, υπηρεσίες, συστήματα αποθήκευσης, υλικό, κ.λπ.



# Acceptance testing

## Έλεγχος αποδοχής του λογισμικού

- Customer / user acceptance testing
- Black box testing
- Ad hoc testing

# Τύποι δοκιμασιών (ανεξάρτητα επιπέδου)

- Smoke testing
- Functional testing
- Usability testing
- Security testing
- Performance testing
- Regression testing
- Bebugging
- Mutation testing
- κ.ά

# Smoke testing

Επίπεδα: Integration, System, Acceptance

- Μήπως πήρε φωτιά;
- Δουλεύει τίποτα ή έχουν αποτύχει όλα;
- Κάλυψη όλων των βασικών λειτουργιών οριζόντια, αλλά όχι σε βάθος (κάθετα)
- Να προχωρήσουμε σε επιμέρους δοκιμασίες ή το σύστημα δεν είναι ώριμο ακόμα (θέλει "κι άλλη δουλειά");

# Functional testing

Επίπεδα: Unit, System, Acceptance

- Επαλήθευση συμβατότητας με προδιαγραφές και απαιτήσεις
- Καλό είναι να εκτελούνται από τους "χρήστες"
- Domain-driven design, behavior-driven development
- Automated tests/specifications

# Usability testing

Επίπεδα: System, Acceptance

- Ad-hoc testing
- A/B testing (σε ευρεία χρήση στις διαδικτυακές εφαρμογές για τη "μέτρηση" της αποτελεσματικότητας μιας UI/UX επιλογής)

# Security testing

Μια κατηγορία από μόνο του

Για διαδικτυακές εφαρμογές: [owasp.org](https://owasp.org)

# Performance / load testing

## Μια κατηγορία από μόνο του

- Load testing: stress, spike, soak testing
- Maximum usable capacity (concurrent users, throughput)
- Acceptable response time (latency)
- Παράδειγμα εργαλείου: [Apache JMeter](#)

# Stress testing

Stress testing (sometimes called torture testing) is a form of deliberately intense or thorough testing used to determine the stability of a given system or entity. It involves **testing beyond normal operational capacity**, often to a breaking point, in order to observe the results.

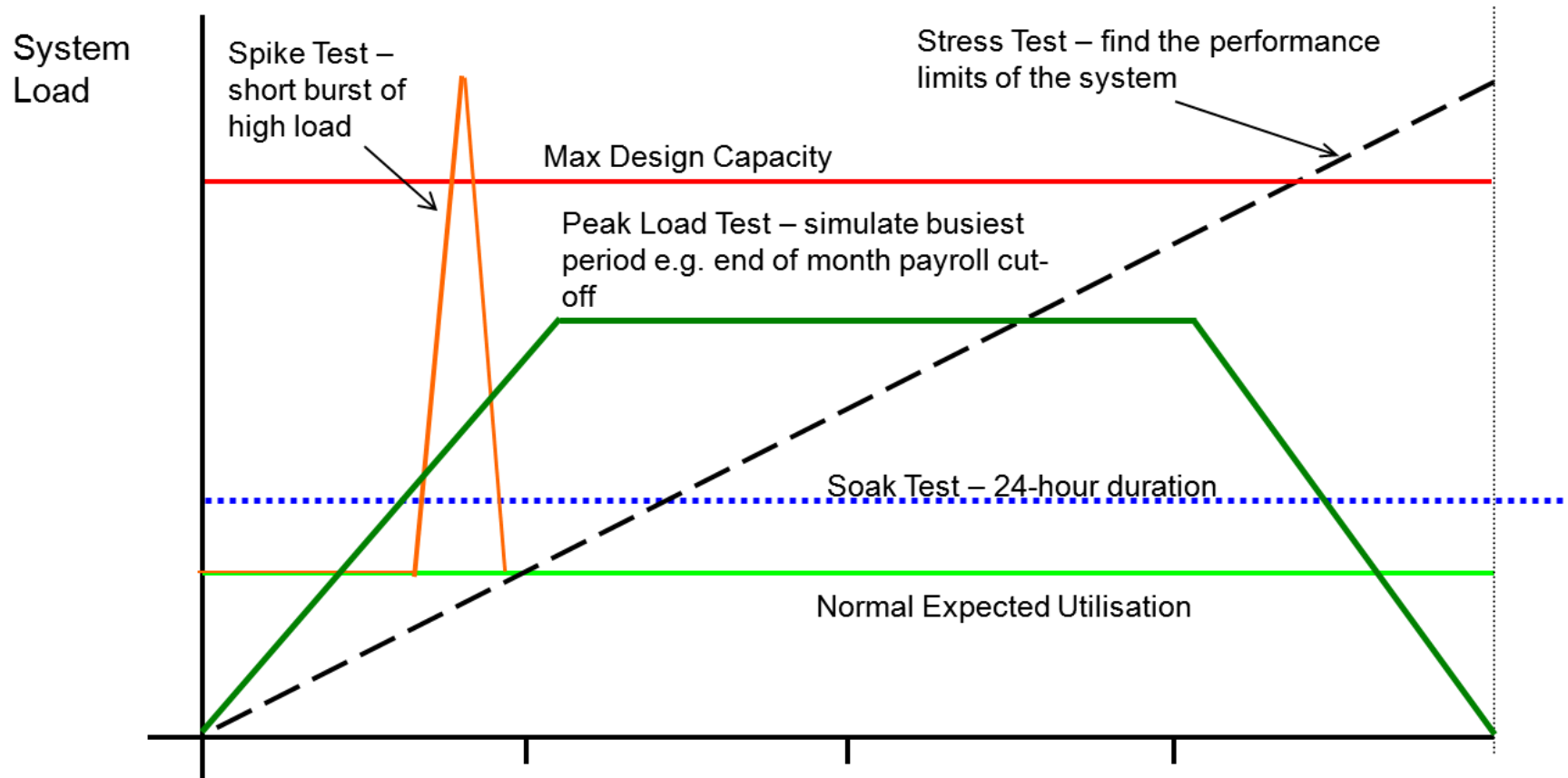


# Spike testing

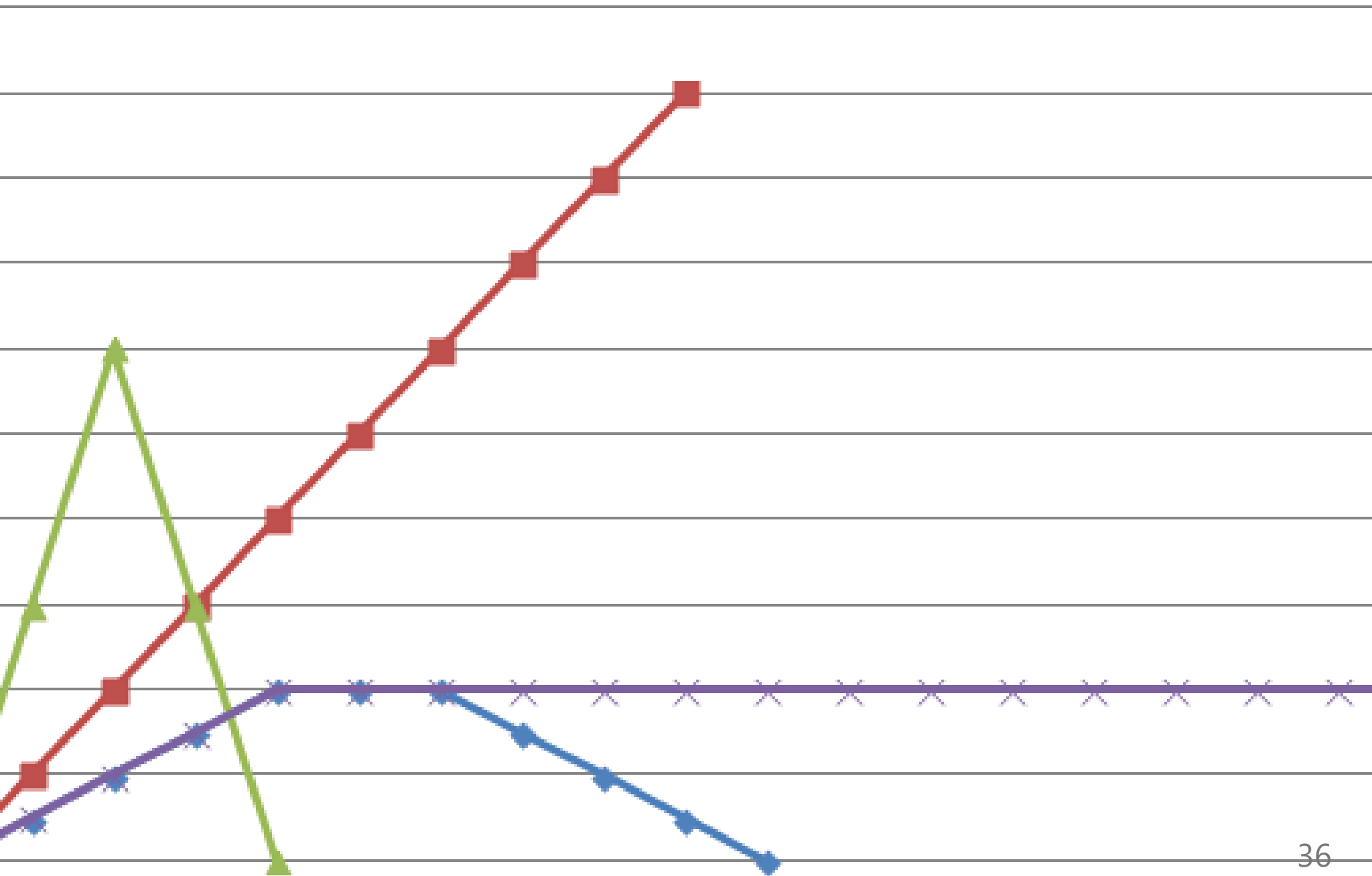
Spike testing verifies a system's stability during **bursts of concurrent user and or system activity** to varying degrees of load over varying time periods.

# Soak testing

Soak testing involves testing a system with a typical production load, over a continuous availability period, to validate system behavior under production use.



◆ Load   ■ Stress   ▲ Spike   ✕ Soak



# Παρόμοιας λογικής

- Endurance testing
- Resilience testing
- Saturation testing
- Scalability testing

# Regression testing

## Οπισθοδρόμηση

- Έλεγχος ότι μια νέα έκδοση του λογισμικού (new feature, bug fix, κτλ.) δεν προκαλεί πρόβλημα στην υπάρχουσα λειτουργικότητα
- Από τους βασικούς ελέγχους σε παραγωγικό λογισμικό

# Επίσης

## Bebugging

- Εκ προθέσεως εισαγωγή σφαλμάτων στον κώδικα με στόχο να ανακαλυφθούν από τις δοκιμασίες

## Mutation testing

- Mutation του κώδικα (π.χ. αλλαγή εκφράσεων σε συντακτικό επίπεδο) με αντίστοιχη λογική

# Σύνδεση με τα προηγούμενα

Κύκλος ζωής λογισμικού

Διαχείριση έργου λογισμικού

Συστατικά λογισμικού



# Software Artifacts, Versions, Releases

Κάθε release είναι ένα ξεχωριστό software artifact (που προκύπτει από διαφορετική έκδοση του κώδικα, π.χ. git commit).

Όλες οι εκδόσεις περνάνε από έλεγχο και επαλήθευση πριν διοχετευθούν στους χρήστες τους

- Pre-release (internal release)
- Early Access (EA)
  - Alpha
  - Beta
  - Release candidate
- Release (General Availability, GA)

# Online service (π.χ. Web app)

Όλες οι επαυξήσεις του λογισμικού (όχι απαραίτητα ως δημοσιευμένες εκδόσεις, αλλά σίγουρα ως διαφορετικά artifacts) περνάνε από έλεγχο και επαλήθευση πριν διοχετευθούν στους χρήστες τους.

- Developer workstation (local tests, e.g. unit tests)
- Testing server (CI, e.g. integration tests)
- Staging server (user acceptance tests)
- Production server (production use)