
BOTTENECKS, DROP OUTS, TIMEOUTS AND MORE...
"Spot check common performance issues"
Jaziel Lopez, Experienced Software Developer
website: jlopez.mx github: jazlopez twitter: @jjlopezmx

Application Performance

- **Definition**
  - Practical Example
- **Importance:**
  - High Performance Use Case
  - Low Performance Use Case
- **Software Development Release Cycle:**
  - Plan ahead: Performance
    - Ecosystem
    - Platform
- **Conducting Testing**
  - Configure Environment
  - Spot check
  - Regression
- **Benchmark and Report**
  - Is your application scalable now?

DEFINITION

"A **continuous** development **strategy** to determine if an app **responds** to a specific usage **demand**"

- KEY FACTORS:
  - Reduce user drop outs
  - Increase traffic
  - User Experience

- KEY FACTORS
  - Optimize components
  - Reduce bottlenecks
  - Lower application costs

- KEY FACTORS:
  - Reduce down time
  - Data Integrity
  - Platform scalable

IMPORTANCE

What do they have in

COMMON?

Uber

Dropbox

PayPal

Google

amazon

stripe

[CONT.] IMPORTANCE

… and what do they have in

COMMON?

myspace

METRO FLOG

Netscape

“PERFORMANCE IS ESSENTIAL FOR HIGH BUSINESS SYSTEM operations”

AVOID LOW PERFORMANCE RELEASES

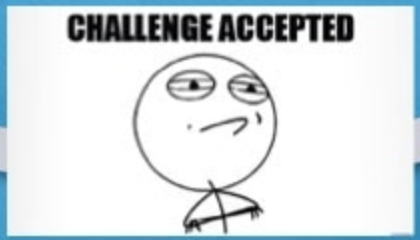
"IDENTIFY PERFORMANCE ISSUES SOONER THAN LATER"

Requirements

"Ain't nobody got time for that"

Analysis

Implementation



Testing

"It runs on my machine… with my user"

Rollout / Support

PERFORMANCE: PLAN AHEAD

PURPOSE: KNOW YOUR ECOSYSTEM AND SYSTEM BOUNDARIES

AUDIENCE

- USER LAND APPLICATIONS HAVE DIFFERENT EXPECTATION LEVELS

USER DRIVEN

- USER LAND APPLICATIONS HAVE DIFFERENT CONTENT TAXONOMY

LOAD USAGE

- USER LAND APPLICATIONS HAVE DIFFERENT SCALABILITY NEEDS

[CONT.]

PERFORMANCE: PLAN AHEAD

PURPOSE: WHAT PLATFORM SHOULD I USE

DATA CONCURRENCY

- YOUR PLATFORM SHOULD SUPPORT DATABASE ACID CAPABILITIES

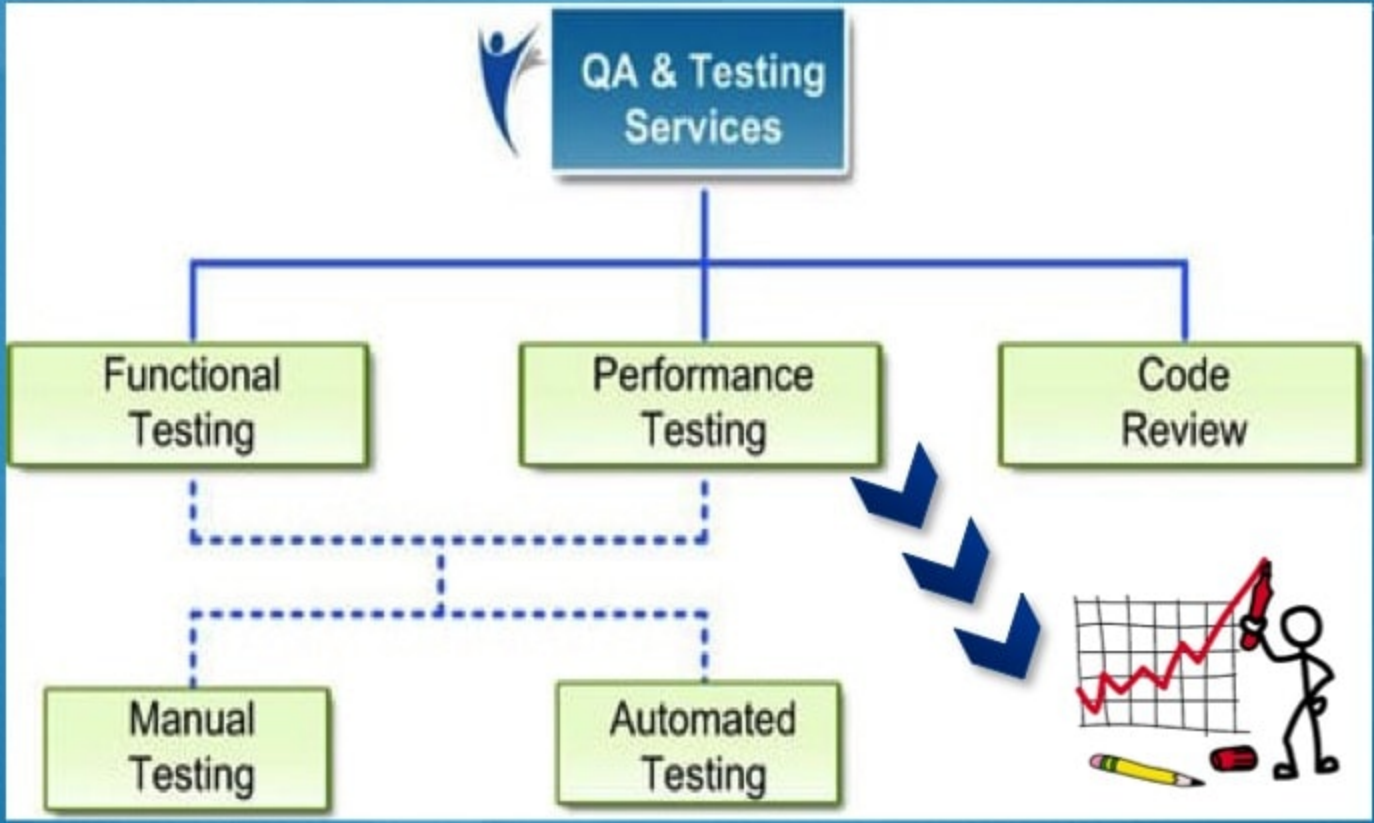
DATA REPLICATION

- YOUR PLATFORM SHOULD SUPPORT MASTER/SLAVE DISK ARRAY

SPECIALIZED SYSTEMS

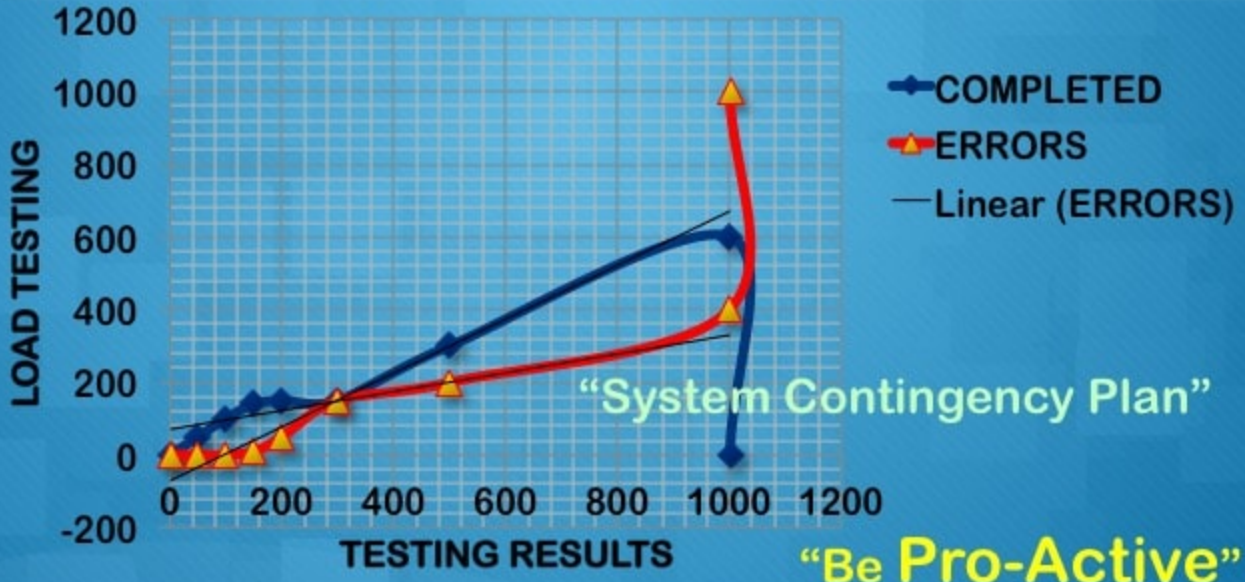
- YOUR PLATFORM SHOULD SUPPORT RUNTIME COMPILATION (SPECIFIC KERNEL HEADERS)

CONDUCTING TESTING



PERFORMANCE TESTING: SPOTCHECK

"Analyze **PERFORMANCE TESTING** results:
SPOT CHECK where the application **CRASHES"**



"System Contingency Plan"

"Be **Pro-Active**"

CONCLUSION

QUESTIONS