Questions before start

* Why do you want me to do automation testing?
* What tools are available and which I can use?
* How can I get the data for tests?
* Can you explain somth…?

Performance testing – type of testing where you verify how the system perform in terms of stability and responsiveness.

Reason for performance testing – bad stability – less users.

Responsiveness – how quick the system response when asked to do something.

* Users feeling negative about app and leave

Stability – how consistent(цілісна) or reliable the system is.

* No response
* Incomplete responses
* Inconsistent experience
* Users may never come back

Workload – how we load or stress system to check it.

Load testing – measure the overall impact on system.

* Basic load – same number of parallel users for certain time
* RampUpDown – from 0 users to some number and then decrease again to 0 one by one
* Multi-step ramp – rump by steps 0 10, 20 to some number and then decrease to 0 by same steps`
* random
* linear – from 0 to some number one by one
* exponential 0 1 2 4 8 and so on

Stress testing – we want to understand the point where the app will break in case of a big number of users.

Endurance testing – big number of users will interact with app for a long time.

Scalability testing – test resource scalability, we want to understand how system will decrease or increase number of servers and do it as expected.

Spike testing – analyze results on spikes conditions, sudden increases during load for example 10 10 70 10 10, double spike 10 10 70 30 70 10 10 to understand system scalability and resources free after users log out, prevent DDoS attacks, and how system will react on that behavior

Non-Code performance issues:

* Latency (затримка) time for request-response result
* Network speed (User part)

Performance measurements:

What questions do you want to answer?

Does it help you to understand user experience?

* Timing (different types) – first byte, overall load time, render time
* Stability – errors (assertions between different types in same situations), server-side problems (memory usage, thread utilization)
* Analytics – discover popular user actions, understand performance impact that users care about

Bottlenecks

Understand where the weak side is – UI, server-side or side-services?

Potential bottlenecks

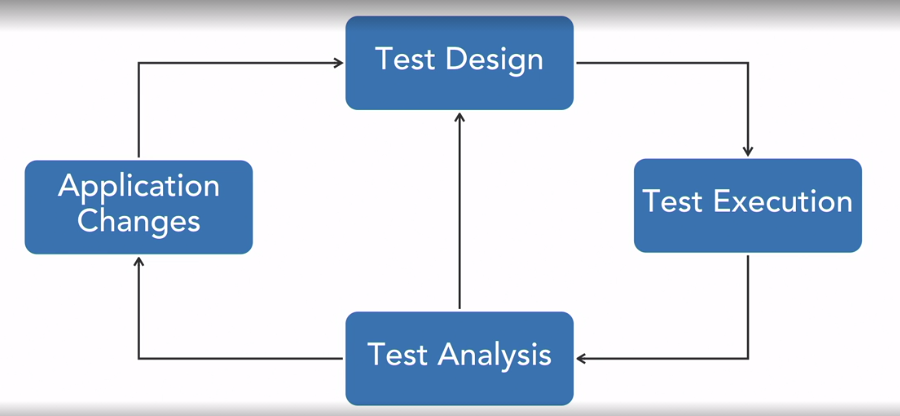
* File IO
* Third-party apps
* Network latency
* Design issues

Challenges

* Real issue or slowdown?
* A lot of variations

How to solve – assert with average results from last 5 runs and pay attention on different alerts, notifications and indicators

Test design lifecycle



Tools for testing

WebPageTest – simple tool to run test by one user, helps to understand time for loading different parts of site on waterfall chart with histograms for all components from page. You can set network speed, device and location or browser.

PageSpeed Extension for Chrome browser – page load speed for different devices using chrome browser and can determine the weak side of page.

Loader

SoapUI

Gatling

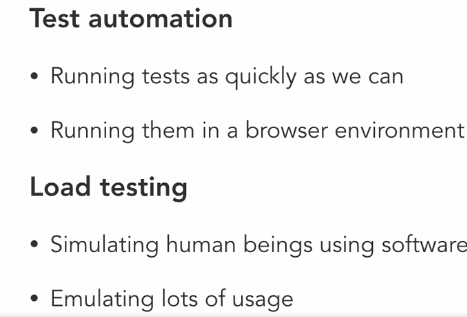
Locust

Telerik Fiddler

DevTools

Test construction





Gatling test component:

* Protocol
* Scenario
* Simulation

Gatling

