Large and Cloud-based Software Systems

Good Games

Technical Concept

Authors: Hoang Trung Hai Pham, Le Hoang Long Nguyen, Hoang Viet Chu,
Syed Sameer Ul Hasan

Date: 06/13/2019

Technology
Arts Sciences
TH Köln

Quality Attribute: Evolvability

Quality Attribute Scenario:

(Stimulus source) If a developer

(Stimulus) updates the product by adding a new feature (e.g. changing GUI)

(Artifact) to UI source codes

(Environment) while Good Games is in normal operation mode with average workload

(Response) the system would return to normal working conditions after the downtime required for updating

(Response Measure) with implementing time for the codebase less than 2 weeks and update times less than 1 hour under the condition that Google cloud is working.

Solution outline:

The final goal (updated features) as well as the design pattern of the product should be well defined prior to the development.

In order to do so, product owner and team leader should agree on which features are important and the necessary amount of time to develop and deploy them. Then working frame plays a crucial role in completing the feature on time (2 weeks in our case). An agile software development methodology therefore is needed to achieve the desired productivity. For example, Scrum framework can be used.

In addition, the codebase should be easy for other team members or newcomers to understand and be able to start working as soon as possible. It could be achieved, if we predict the complexity at the beginning. The code base should have strong cohesion and weak coupling. Also, codes should have good API documentation and comments that explains the confusing parts.

Code review is mandatory, changes must be approved by at least 3 developers before merging. It helps teammates understand the changes and keep track of the progress of the project.

New features should be tested before being deployed to the application server. Test cases should be carefully planned with concrete test cases.

Sometimes, changes are urgent and developers are more concerned about making the new feature work properly, instead of providing a stable solution for future changes. Therefore, code refactoring is considered after a while.

The developers are independent of deploying and maintaining product online. It should be done by the DevOps engineer. DevOps team ought to not only maintain the system to be accessible, but also decide which point of time is good to deploy new change, such as the time when people get access to our system minimum. All the update plans should be known in prior, and inform to users.

Measurement:

The response measure of a downtime which lasts less than 1 hour in the 0.99-percentile is tested:

DevOps and security teams must coordinate to set up the test. The test simulates repetitively the situation when the application is updated with a newly released version or a patch.

We apply a monitoring tool (e.g. Jenkins) to measure and log the downtime for each update and generate a histogram of percentiles from the logs.

To maintain availability for our users, we expect a downtime of less than 1 hour. The process may take less than that if the changes are not big, but 1 hour is the maximum amount of time.

The response measure is estimated based on the load of change applied to the system, whether it is a big change or a small change. From that, we might be able to deduce the appropriate time required for a specific task.