Test Plans

PV260 Software Quality

Matěj Karolyi, Michal Abaffy

7. 4. 2020 / 14. 4. 2020

Terms and Conditions

Before we start

Please go to the Socrative web page:
https://b.socrative.com/login/student/

Enter the following Room Name: **KAROLYIHOUCEBNA**

Test Plan

Definition

Test Plan is an asset that helps planning of the testing.

Can be just a collection of test cases. Can be extended by an information which test case is automated or manual, what is its component area, how often should the test case be executed, owner of the test and so on... Test Plan also do not have to contain any test case. Can be just a risk analysis, analysis of attributes that should be tested or anything else that helps planning of the testing.

Test Plan

Six basic questions that are important for test planning:

1. Why?
2. **What?**
3. How?
4. Who?
5. When?
6. Where?

Test Plan - What to test

These are ours **two basic principles**:

- Process and/vs result. Use not only intuition and experience, but also **test strategies** and test methods.
- Combination of different perspectives. Some things are difficult to see from one perspective, but easy to see from another.

You need **three types of knowledge**:

- Technical
- **Methodological**
- Business

Test Case

Definition

Test Case is a sequence of steps together with an expected result.

Examples of test cases:

1. Go to the registration page and try to register. You shall be able to register.
2. Go to the registration page on `http://example.com/register`. Enter Name, Email, Password and Repeat Password. Click Register button. Check that you are considered registered on http://example.com
3. Close all anonymous browser windows of Chrome version 48.0.2564.116 on Windows 7. Open a new anonymous Chrome browser. Go to the registration page on `http://example.com/register`. Enter Name = John Smith, Email = john@smith.com, Password and Repeat Password = test. Click Register button. Check `"Welcome, John Smith"` is displayed in the upper right corner.

Test Case – two perspectives

Definition

Positive scenario is a scenario of expected use of application.
Negative scenario is a scenario of unexpected use of application.

Definition

Positive test case is a test case for positive scenario. Negative test case is a test case for negative scenario.

Example: Field can contain up to 100 symbols.

1. Positive test case: Type 'ok text' and check the string is accepted.
2. Negative test case: Write string containing 101 letters and check the warning message of text is too long is displayed.

Test Case

Questions to consider asking for:

1. Are the steps defined so deterministic that everyone following them will get the same result? Is it even possible to write a test case and be 100 percent sure everyone executing it will get the same result?

2. Is it needed that test case is written in a way that everyone executing it will get the same result? How detailed should a test case be?

Test Plans based on Specification

Exercise 1

In 15 minutes, create a test plan based on the following specification. Write test cases which execution should detect as many bugs as possible. Upload your test plan draft to Socrative's current question field.

Specification

- On http://example.com/register, there is a form with fields "Name, Email, Password and Repeat Password" and "Login" button.
- @ symbol is mandatory for email and password should contain at least 1 lowercase and 1 uppercase symbol and must be at least 6 symbols long.
- Password and Repeat Password must match.
- After entering all 4 fields and clicking Login button, you are registered.

Test Plans based on Specification

Lessons learned:

1. Specification might be (and almost surely is) Incomplete.
2. Specification might be buggy (as well as its implementation).
3. Creating detailed test cases.
4. Thinking about edge/corner cases. What corner cases are worth adding to test plan and which are worth to test? Which can be omitted?
5. Positive and negative test cases.

Big application testing

Exercise 2

Exercise 2: In 15 minutes, try to define a strategy how to create a test plan for Facebook (let's assume the Facebook application with all its current functionality is just to be implemented/in the process of implementation)

"Not 'creating the test plan', but 'thinking about how to create the test plan'. "

Fill the crucial points of your strategy to the Socrative. Is there anything questionable? Write it down and we **can discuss it**.

Big application testing

Lessons learned:

1. Thinking about what is important in test planning for real (big) application.
2. Brainstorming is necessary.
3. If you feel like "not knowing how to do big application test planning" so that you more appreciate the next strategies and methods.

ACC method

1. **Attributes**. First, think of the attributes that your application should have to be successful and that you want to test.
   For example: fast, secure, extendable, testable, cross-platform supported, ...
2. **Components**. Then define Component of your application.
   For example: error logging, configuration of this, configuration of that, display of this, display of that, ...
3. **Capabilities**. For each combination of (*Attribute, Component*), try to think of Capabilities of the application which you later want to test. For example for (Secure, Registration), there are capabilities like
   - Registration is over HTTPS
   - Password should have some attributes of not being able to be easily guessed by attacker
   - There is possibility of SMS check in case of forgotten password
   - ...

ACC method

4. **Test cases**. From capabilities, define test cases if needed. For example from capability "Registration is over HTTPS", there might be test cases like
   - "Go to http://example.com/register and check you are redirected to https://example.com/register"
   - "Go to "https://example.com/register" and check the content of the page (F12 in Chrome or checking the Source code).
   - There shall be no insecure content (for example no css file served from http://)."

ACC method

Exercise 3

In 10 minutes, create a test plan for Facebook using Google's ACC method (let's assume the Facebook application with all its current functionality is just to be implemented/in the process of implementation).

Think mostly about attributes and components.

Fill your result into Socrative.

ACC method

Lessons learned:

1. Importance of attributes in test planning / testing (and know about attributes).
2. Importance of components in test plan (defining component have lots of usage also later – definition of teams; easy grouping of issues); and in the design creation as well. Warning: there might be sub-components of components.
3. Knowing method which enables you to come up with a test plan for any product quite quickly and containing most of the important things.

ACC method

Links:

- http://googletesting.blogspot.cz/2011/09/10-minute-test-plan.html
- https://code.google.com/p/test-analytics/
- https://en.wikipedia.org/wiki/List_of_system_quality_attributes
- http://www.istqb.org/downloads/send/10-advanced-level-syllabus-2012/55-advanced-level-syllabus-2012-technical-test-analyst.html (Chapter 4)

Clustering

- Class of problems: Solution is a long list of items.
- Very hard to think of all the items without giving some structure to the thinking.
- Solution: Put similar items in the clusters.
- Examples where it would be very hard without clustering:
  1. Things to take on a vacation without putting them in clusters 'Clothes', 'Cosmetics', ...
  2. Things you should learn at Informatics university without putting the knowledge into subjects.
  3. Test plans with test cases without using keywords like 'attributes', 'components' and/or others.
- ACC method is an example of 2-dimensional clustering (capabilities from combination of attribute and component). Clustering can be used even within components or attributes.
- Similar approach is 'divide-and-conquer'.

Risk Analysis

- What is a 'risk'?
- Anything - attribute, component, capability, impact, use case, stakeholder, ...
- Product risks vs Project risks

Phases of risk analysis

1. Risk Identification
2. Risk Assessment
3. Risk Mitigation

Risk Identification of Product Risk by Impact

1. Your product will destroy the world.
2. Your product will make you being killed/murdered.
3. Your product will kill or harm others.
4. Your product will make you/others being arrested.
5. Your product will suddenly need to be shut down (1. lawsuits from customers - not behaving due to Terms and Conditions, 2. from competitors - incorrect patent usage, 3. against the law - missing Cookie Policy on web site; 4. terrible problem get it publicly and related lose of reputation, ...)

Risk Identification of Product Risk by Impact

6. Your product will not have ambition to be successful:
   - People will not want to use it (not easy to use, competitors are better, missing some core functionality, ...).
   - Bad business plan (not enough money for one user, not enough users, ...).
7. Your product will suddenly lose many users (security bug leaked to public, some critical bug injected in the application + potential inability to fix it quickly, some public speech from representatives that will annoy the users, ...)
8. Your product will annoy its users (obvious bug too long not being fixed, some component too difficult to be used, ...)
9. Your product will not be sustainable (too many bugs, badly written code, bad processes, ...).

Risk Matrix

Is the possible impact the only criterion for definition how big the risk is?

No.

Probability of the problem occurring is another very important measure for how risky the problem/component is.

Links:

- https://en.wikipedia.org/wiki/Risk_Matrix

People risk

- People bottleneck during development - lack of knowledge sharing, one person being the only expert on some area. What if such person is on PTO, sick or leaves the company?
- Unfriendly working environment (a lot of stress, low salary, micro-management, lack of team-building, boring work without any option to escape it, too crowded offices, ...). People losing motivation; not willing to do something more than necessary; or even leaving the company
- Employee intentionally do some harm to the company. Employee steals production code and sells to competitor; steals and misuse customers' private data; Technical Support tells something harmful to customer; Developer intentionally makes a bug in the code; businessman steals customers...

People risk

- Not good motivating system.
- Human error.
- Someone steals some employee's access and do harmful things with this stolen access ("trust your employee, but do not trust employee's computer")
- Reputation of bad employer and inability to get new employees.
- ...

Other Risks

- Weather (hurricane, flood, fire, ...)
- Change of laws in the country (some new taxation; some new "employee-protection" law, ...) or even complete change of regime
- Terrorist attack
- ...

Risk Analysis

Exercise 4

In 15 minutes, try to think of the most risk places of Google Chrome that should be included it the test plan.

Hint 1: Working with keywords such as attribute, capability, component, use case or impact might help you. **Hint 2**: Clustering technique might help you too.

Fill your result into Socrative.

Lessons learned

1. Given you the feeling of what might go wrong (surely not a complete list). Given you feeling that those risks should be addressed and some ideas how to address them.
2. Risk-based test planning.

Links:

- https://www.istqb.org/downloads/send/12-expert-level-documents/76-expert-level-syllabus-test-management-2011.html

Test Approaches

1. Specification/Requirements-based testing
2. Testing based on the ACC method
3. Risk-based testing

Other test approaches not covered in the seminar:

4. Cause-effect graphing (visualization helps to understand the problem)
5. Checklist-based testing (suitable for small changes)
6. Experience-based testing
7. Reactive approach (not much planning before test execution)
8. Coverage-based testing
9. Testing based on standards (health sector, cars, spaceships,...)
10. ...