## ALX SE

### 0x05. C - Pointers, arrays and strings (project)

### 0x6. C - More pointers, arrays and strings (project)

### 0x7. C - Even more pointers, arrays and strings (project)

### 0x8. C - Recursion (project)

### 0x9. C - Static libraries (project)

### 0x0A. C - argc, argv (project)

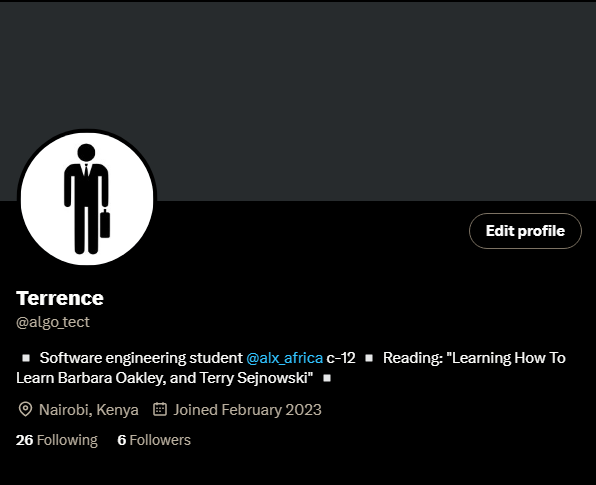
### 0x0B. C - malloc, free (project)

### ✅Professional Social Presence (project)

So, you’ve got all this information about professional social media presence, but do you know where to start?

If the answer is “no”, don’t worry, here are some tips to help build and maintain your professional social media presence!

* Find and follow one really smart person each day on Twitter
  + Give them a follow, read their work
* Each week, contribute to 3 social media conversations between Twitter and LinkedIn
  + Ask and answer questions from strangers
  + Geek out about something that excites you
* Once per week, evangelize what you’re doing and where you’re at in your journey
  + Solve a difficult problem?
  + Really work well with your partner on the last project?
* Build Connections
  + Keep the conversations going
  + Invite people to events you’re aware of
* Have Fun and Stay Engaged!
  + Each person you meet on Twitter or through LinkedIn could be someone who helps you in future professional endeavors



### 0x0C. C - More malloc, free (project)

### 0x0D. C - Preprocessor (project)

### 0x0E. C - Structures, typedef (project)

### 0x0F. C - Function pointers (project)

### 0x10. C - Variadic functions (project)

### Evaluation #1 (evaluation-quiz)

1. what does the macro TABLESIZE expand to?

#define BUFSIZE 1020

#define TABLESIZE BUFSIZE

#undef BUFSIZE

#define BUFSIZE 37

* 37
* 1020
* Nothing

1. What is the size of a pointer to an int (on a 64-bit architecture)?
2. How many bites will this statement allocate on a 64-bit machine?

malloc(sizeof(char) \*10)

1. The memory space reserved when calling malloc is on:

* The heap
* The stack

1. Given this code:

struct point {

int x;

int y;

};

struct point my\_point = { 3, 7 };

struct point \*p = &my\_point;

To set the member y of a variable my\_point to 98, I can do (select all valid answers):

* my\_point->y = 98;
* (\*p).y = 98;
* my\_point.y = 98;
* p.y = 98;
* p-> =98;

1. What command(s) can be used to list the symbols stored in a static library?

Select all valid answers

* + ar
  + ranlib
  + ld
  + nm

1. This void (\*anjula[])(int,float) is:
   * A pointer to a function that takes an array of int and float as a parameter and returns nothing
   * A pointer to a function that takes an int and a float as parameters and returns an empty array
   * An array of pointers to functions that take an int and a float as parameters and returns nothing
   * A pointer to a pointer that takes an int and a float as parameters and returns nothing
   * A pointer to an array of functions that take an int and a float as parameters and return nothing
2. What does this code print?

void print(int nb)

{

printf(“%d”, nb);

-- nb;

if (nb > 0)

{

print(nb)

}

}

int main(void)

{

print(4);

return (0);

}

1. What is wrong with the following code?

int n = 5;

int array[5];

int i = 3;

array[n] = i;

* Nothing is wrong
* The array array is not entirely initialized
* It is impossible to declare the variable array this way
* While it is possible to access array[n], we are not supposed to as this is not the array anymore

1. What is the value of n after the code is executed?

int n = 98;

int \*p = &n;

\*p++;

1. What is the size of \*p in this code on a 64-bit machine?

int \*\*p;

1. How many bytes will this statement allocate on a 64-bit machine?

malloc(sizeof(int) \* 4)

### 0x11. C - printf (project)

### 0x12. C - Singly linked lists (project)

#### [Data Structures](https://intranet.alxswe.com/concepts/120)

Data structures, as the term implies, are a way of structuring data in order to efficiently store, find, use, and create data, depending on the task at hand. If data is a plate of food, data structures are your utensils (algorithms would be how to effectively use those utensils in this analogy). Just as you will likely select a spoon to eat a bowl of soup rather than a knife, you will learn over time to select and properly use data structures which fit the nature of the data you are working with. When you begin learning basic data structures such as arrays and linked lists, you will start to conceptualize how data is stored, searched, and edited, and what these operations will contribute to the space/time complexity of your program at runtime (this will make more sense once you start to learn about Big O notation). As you continue to learn more complex data structures, you may notice that you need to think more abstractly and deliberately in order to properly implement them. But with patience and practice you will begin to see patterns emerge that will allow you to intuitively see which situation calls for which data structure, just as you know to pick up a spoon when you see a bowl of soup.

Basics (Read deep into either for explanations of common data structures):

* [Tutorials Point](https://intranet.alxswe.com/rltoken/ZjxizWAKkkJzpHs1mmZZdw)
* [Geeks for Geeks](https://intranet.alxswe.com/rltoken/j6OHt4I4GW0dOWihIwQoWA)

How to Select the Appropriate Data Structure:

* [Career Drill](https://intranet.alxswe.com/rltoken/Zf-TH854gUT9X44DXf5VJw)

Resources

* [Linked Lists](https://intranet.alxswe.com/rltoken/joxg32-tt4lUh8Afgst8tA)
* [YouTube](https://intranet.alxswe.com/rltoken/epKUCIcoA6XaN1T3Vtr_9w)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone, without the help of Google:

* When and why using linked lists vs arrays
* How to build and use linked lists

#### Quiz questions

**Question #0**

* What’s the “tail” of a linked list?
* It’s the node with the highest value
* It’s the node with the lowest value
* It’s the first node
* It’s the node with the pointer to the next equals to NULL

**Question #1**

* It’s the node with the lowest value
* It’s the node with the pointer to the next equals to NULL
* It’s the first node
* It’s the node with the highest value
* It’s the last node

**Question #2**

In a singly linked list, what are possible directions to traverse it? (select all possible answers)

* Backward
* Forward

**Question #3**

* Array can contain as value a structure
* We can add elements indefinitely to an array
* Memory is aligned for an Array - each elements are back to back in the memory
* We can easily remove an element from an Array
* Memory is aligned for a Linked list - each elements are back to back in the memory
* Linked list can contain as value a structure
* We can easily remove an element from a Linked list
* We can add elements indefinitely to a linked list

**Question #4**

* What’s a node? (Select all possible answers)
* It’s an integer
* It’s a space allocated in memory
* It’s a cell in an array
* It’s a structure with a pointer to the next node and value information
* It’s a server

### Your network is your net worth (project)

#### Create a professional LinkedIn profile

To have a professional-looking LinkedIn profile, here are some important things to do:

* **Use a professional profile photo**: Your profile photo is the first thing people see when they visit your LinkedIn profile. Make sure it’s a professional headshot that reflects your industry and personal brand. If you do not have a professional headshot (like most of us), simply put a picture of yourself. [Here is an article with examples and advice](https://intranet.alxswe.com/rltoken/AtzLsxm73YzJUEEzskhuLQ)
* **Craft a compelling headline**: Your headline should be a brief summary of your professional experience, skills, and expertise. It should make people want to learn more about you. You are allowed to use ChatGPT for this.
* **Write a summary that showcases your strengths**: Your summary should highlight your professional strengths, experience, and career goals. It should be well-written and engaging, and it should make people want to connect with you. You are allowed to use ChatGPT for this.
* **Customize your LinkedIn URL**: Customizing your LinkedIn URL can make it easier for people to find you and remember your profile. It also looks more professional than the default URL.
* **List your work experience and education**: Make sure your work experience and education are up-to-date and complete. Be sure to include any relevant certifications, awards, or achievements.
* **Use keywords**: Use keywords throughout your profile to help people find you when they search for specific skills or expertise.
* **Ask for recommendations**: Recommendations from colleagues and supervisors can help to validate your skills and experience. Be sure to ask for recommendations from people who know your work well.

Overall, a professional-looking LinkedIn profile should be well-written, complete, and engaging. It should showcase your strengths and expertise and make people want to connect with you.

#### Connect with 10 persons

Connect with 10 new persons (you can do more if you would like of course). It could be:

* your friends
* someone from the ALX SE staff
* your peers
* someone from your family

When you ask someone to connect with you, make sure you add a personalized message. For instance, if you would like to connect with [Julien Barbier](https://intranet.alxswe.com/rltoken/aHn4xc0NfjsETXb3CEUCtQ), make sure to mention that you are one of his students from ALX SE. If you would like to connect with your ALX peers, make sure to mention you are going through ALX. etc.

From this day on, anytime you meet someone (online, email, offline, etc.) get into the habit of connecting with them on LinkedIn. Remember ALX SE permits you to meet with so many incredible people. Take advantage of this. You should have at least hundreds of connections at the end of the program just by adding peers you interact with.

## QATestLab

### Lecture 1. Introduction. What is a bug (defect)?

[Lecture #1. Fundamentals of Software Testing | Free online courses from QATestLab](https://en.training.qatestlab.com/tag/lecture-%e2%84%961-fundamentals-of-software-testing/)

[Bugs examples](https://drive.google.com/file/d/1OMa7zSl9sAewbHt9ruOZR_oZE2oxAolO/view?usp=share_link)

[Creating a bug report](https://drive.google.com/file/d/13LxQK_ohZMwOkNxVoinrGpv8RL9L_qUg/view?usp=share_link)

[Handouts 1](https://drive.google.com/file/d/1FziYGbx698StTxAakZP6nN4wy2IR8SzL/view?usp=share_link)

[Helpful materials](https://drive.google.com/file/d/19cE4aCLtX-au3i-mh92D3VXxu8SybF5-/view?usp=share_link)

[How to do Homework assignment 1](https://drive.google.com/file/d/1RIO1R3iK3XxaesG3BpAwgTFy3-7jBOyo/view?usp=share_link)

[TestLink glossary](https://drive.google.com/file/d/1u0XPi2YfQy5MaEfbzBDd2OQC9S1o9vY0/view?usp=share_link)

[Primary requirements](https://drive.google.com/file/d/1RL8fgfKfKUefPoKSj-jKc0TT1ITO8fi2/view?usp=share_link)

[Typical comments](https://drive.google.com/file/d/1UZIEIQHfaFUq-tbCqNSE6S5dpTDzuMp5/view?usp=share_link)

[Lecture test 1](https://docs.google.com/document/d/1RXw-MCjLjkJUDv4JA8XgGwhvxfaSKuacdQ\_BinW2NJk/edit?usp=sharing)

\* Bug - a defect in software that produces unexpected results.

\* QA Engineer - professional who finds and fixes bugs in a software before its launch.

\* Bug Report - a summary of bugs in software, how to reproduce them and resolutions.

\* Bug Work-flow - a sequence of steps that a bug follows from when it was found to when it was fixed.

\* Bug-tracker - is an error tracking system that allows you to create, store, review and modify bug reports.

\* Actual Result - the result of the bug

\* Expected Result - the software's desired outcome

\* Severity- how the defect affects the performance of the program

\* Priority - the order in which defects must be solved defined by developers. The higher the priority the sooner the bug should be fixed.

\* Bug Report Attributes

- Bug ID

- Severity:

- Blocker - a bug that blocks functionality

- Crash - a bug that leads to a crash of the application (or even the OS)

- Major - a bug in important but not key functionality -

- Minor - a bug in additional functionality

- Trivial - a minor but annoying bug

- Text - typo

- Tweak - a bug that causes inconvenience in using

- Priority: High Normal Low

- Summary: What? Where? When?

While describing the bug, it is very important to indicate what exactly broke in the point «What?», where in the system did it happen in the point «Where?», and also under what circumstances in the point «When?». For example:

«What?» (What happened?) – this is NOT a noun, this is what happened, a kind of clarification - VERB («What is being done?» or «What is NOT being done?»).

«The «Profile» page is opened» – WHAT?

«Where?» – place in the system (application/web-site), where the bug can be found.

«on the «Login» page» – WHERE?

«When?» (after what? Due to what action? During what period of action?) – action that led to the result different from the expected one.

«after clicking the «Sign in with Facebook» button» – When?

- Bug Description : Adds more info to the summary

- Steps to reproduce

- first step must contain the link to the main domain in it

- Actual result & Expected result:

- it's better to avoid formulations of results that differ only with negative form

- Attachment: screen-shot, video, logs

- good quality

- the URL must be shown on the video/screen-shot

- only one tab should be open: where the bug is found

- bookmark bar or other unnecessary information must be omitted

- a red rectangle highlighting the problem area and a red arrow pointing to it

- if the bug requires several steps to reproduce then all steps should be shown and numbered

-

- Additional Information: Environment: OS, browser + version, mobile device

- Rules for Bug Reporting:

- one bug = one bug report

- Statuses

- New (new ticket that still is not sent to the developer)

- Feedback (the status assigned after reopening a ticket)

- Need clarification

- Assigned (ticket processing by software developer)

- Resolved

- Closed

- Resolutions

status assigned by a developer

- open (the issue remains unresolved)

- fixed (the issue id resolved)

- reopened (the issue was resolved or closed and opened again)

- duplicate

- not fixable (the issue can't be resolved)

- no change required (changes aren't needed)

- suspended (the issue resolution is paused)

- won't fix (a developer will not solve this problem due to lack of time or because of the risk of appearing new problems after making changes)

- ::Software Testing - is a process used to identify the correctness, completeness and quality of developed computer software. It includes a set of activities conducted with the intent of finding errors in software:: so that it could be corrected before the product is released to end users.

- 7 testing principles

1. Testing shows presence of defects

1. Exhaustive testing is impossible - instead we need "optimal" amount of testing based on the "Risk assessment" of the application.

2. Early Testing

3. Defect Clustering

4. Pesticide Paradox - If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs. To overcome this, the test cases needed to be regularly reviewed & revised, adding new and different test cases to help find more defects.

5. Testing is context dependent

6. Absence of errors is a fallacy

- Bug Tracking Software

- Pivotal Tracker

- Basecamp

- Redmine

- Jira

- Azure DevOps

- targetprocess

- Assembla

- Screen-shot Tools

- Snagit

- TechSmith Capture

- Clip2net

- Bandicam

- Fraps

- Lightshot

- Shutter

### Lecture 2. Web projects testing

[Lecture #2. Fundamentals of Software Testing | Free online courses from QATestLab](https://en.training.qatestlab.com/tag/lecture-%e2%84%962-fundamentals-of-software-testing/)

[Check-list Examples layout](https://drive.google.com/file/d/1EvmvzEDkNEPbYplaMALx8X2oaJdqTa6n/view?usp=share_link)

[Checking bug reports life cycle](https://drive.google.com/file/d/19GeOYiia0aBp_s_wAdsZUlcuhLod3bvk/view?usp=share_link)

[Handouts 2](https://drive.google.com/file/d/1M8VoOTkXfG0Phqov0nrbnIi_td-6RYm3/view?usp=share_link)

[Helpful materials 2](https://drive.google.com/file/d/1lhRrT23IK8sZCrjsqokED9LVFRjMeTrh/view?usp=share_link)

[How to do Homework assignment 2](https://drive.google.com/file/d/1hj0GCg3TVISzAn1Vr5azu0rMdHlHWzMv/view?usp=share_link)

[Layout testing](https://drive.google.com/file/d/1jYbPYEAGnsW9GUxJNn77zneenaG8i7Cx/view?usp=share_link)

[Lecture test 2](https://docs.google.com/document/d/1wIcsJCOajJObHJFRy9D8hXbE9rCwGsJKz9K7-KIpE-Y/edit?usp=sharing)

[How to do Homework assignment №2.pdf](https://www.dropbox.com/s/63aoqnf1gfd1hvr/How%20to%20do%20Homework%20assignment%20%E2%84%962.pdf?dl=0)

- Layout - is a representation of the website page design.

- Checklist - a list that contains necessary checks for testing.

- Cross Browser Testing - The process of testing the web-applications and web-sites in several browsers

- Favicon (abbr. from FAVorite ICON) - The icon of a website or web-page

- Placeholder - is a hint to the text input field

- Logo - A graphic mark, emblem or symbol used by an entity for its corporate identification and recognition.

> the difference between logo and faviconis that logo is a symbol or emblem that acts as a trademark or a means of identification of an institution or other entity while favicon is an icon associated with a particular website, and typically displayed in the address bar of a browser viewing the site.

- Header - a block at the top of the page with primary information.

- Footer - a block at the bottom of the page

- Navigation menu - links to main sections of the web-page

- Content -

- Stages of web project Testing

1. Documentation Review

2. Layout Testing - comes before Functional Testing

1. Visual Part Testing

2. Availability Testing

3. Correct work after filling in the fields with the real text, layout reliability testing

4. 404 error - can be checked with tools such as : W3C Link checker, Xenu e.t.c

3. Functional Testing

4. Usability Testing - in a basic sense website usability addresses the question of how "user friendly", attractive and understandable a website is.

5. Performance Testing - testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload.

- Load Testing - the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently

- Stress Testing - is a type of performance test that checks the upper limits of your system by testing it under extreme loads

-Stress testing is a type of testing that verifies the reliability and stability of software applications. The goal of this kind of testing is to measure the error handling capabilities of the software to ensure that it does not crash under extremely heavy load conditions

- Scalability Testing

- Volume Testing

6. Security Testing

7. Cross Browser Testing

- Instruments for Cross Browser Testing include: Ghostlab, BrowserStack, SauceLabs, Browsershots

- Web project Testing Tools

- Testing in different resolutions:

- View Port Resizer

- Firefox > Tools > Web Development > Adaptive Design

- Resolution Test Plug-in or Window Resizer (Chrome Browser)

- Measuring the distances on the screen:

- Screen Calipers

- Snagit

- Photoshop

- Jing

- Checking the Mockup Accordance:

- Photoshop

- PerfectPixel

- The eternal cycle of Testing

1. Generate an idea about the perfect product

2. Find inconsistencies between a perfect product and testing product (Complete the stages of the <<Bug Life Cycle>> )

3. Give the information about the bug/conditions of reproducibility

4. After fixing the bug check the correctness

- Webpage Anatomy

- Website testing Checklist

- Lifecycle of the bug reports checking

---

### Lecture 3. Approaches to the functional testing

[Lecture #3. Fundamentals of Software Testing | Free online courses from QATestLab](https://en.training.qatestlab.com/tag/lecture-%e2%84%963-fundamentals-of-software-testing/)

[Helpful materials 3](https://drive.google.com/file/d/1fWnh5T-DXtIVj0kBwbiu0ltGXHjuRMKC/view?usp=share_link)

[How to do home assignment #3](https://drive.google.com/file/d/1FPxuH-j_pZM4sZ0MeuPoNWMyZlVC2W-X/view?usp=share_link)

[Lesson 3 materials](https://drive.google.com/file/d/19dXb27f3ATTGCie70XIvaR4WFVYsCQLP/view?usp=share_link)

(https://en.training.qatestlab.com/tag/lecture-%e2%84%963-fundamentals-of-software-testing/)

[Glossary of lecture №3](https://en.training.qatestlab.com/blog/course-materials/glossary-functional-testing/)

[Anatomy of the Webpage/Website glossary](https://en.training.qatestlab.com/blog/technical-articles/web-page-anatomy/)

~Functional Testing~ - is a quality assurance process that bases its test cases on the specifications of the software component under test.

~[Functional Testing & Non-functional Testing](https://drive.google.com/file/d/1--MtmogDx9aOzY26yPGu3Bxpg3Rdh8q-/view?usp=share\_link)~

~Types of Functional Testing:~

- Smoke Testing - is a software testing technique ::performed post software build to verify that the critical functionalities of software are working fine::. It is executed before any detailed functional or regression tests are executed. The main purpose of smoke testing is to reject a software application with defects so that QA team does not waste time testing broken software application.

In Smoke Testing, the test cases chose to cover the most important functionality or component of the system. The objective is not to perform exhaustive testing, but to verify that the critical functionalities of the system are working fine. For Example, a typical smoke test would be – Verify that the application launches successfully, Check that the GUI is responsive … etc.

- Regression Testing - is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases that are re-executed to ensure existing functionalities work fine.

This testing is done to ensure that new code changes do not have side effects on the existing functionalities. It ensures that the old code still works once the latest code changes are done.

- Unit Testing > Integration testing > System Testing > Acceptance Testing

- Unit Testing - is aimed at testing small seperate modules of the application which can be examined seperately from other small parts.

- Integration testing - is aimed at checking interactions between several parts of an application (given that each of them has been tested on the stage of unit testing).

- System Testing - Main goal is to check both functional and non-functional system requirments.

- Acceptance Testing - testing that is usually carried out by the customer or any other interested parties, main purpose of which is to verify that the product looks and works according to product requirements.

- Requirement - is a description of the application functions and the conditions needed for their performance during the execution of useful user tasks.

- Validation - a confirmation that the requirements were met.

- Web form - is a web-page element designed for entering information using check-boxes, radio buttons, or text fields.

- Verification - is checking whether the tasks,goals and deadlines of product development were met.

- Drop- down list - is a GUI element that allows a user to select one value from a list.

- Authorization - is confirming that a person/people have the rights to access resources and execute come predefined actions.

> Simply put, authentication is the process of verifying who someone is (a user's identity), whereas authorization is the process of verifying what specific applications, files, and data a user has access to(a user's access rights) .

~Web form Testing~

- A ~positive test~ scenario uses valid parameters. It is intended to show that the program works as it should, presupposing that the user enters the correct data and does not go beyond the scope of the provided behavior cases.

- A ~negative test~ scenario uses invalid parameters. The main goal of performing such tests is to make sure that the system does not accept incorrect data.

- Popular negative test cases for checking input fields:

- Required Data Entry

- Field Type Test

- Field Size Test

- Numeric Bounds Test

- Data Bounds Test

- Date Validity

- Rules for filling Fields

- Embedded Single Quote

~Testing Techniques (equivalence partitioning, boundary value analysis)~

- Equivalence partitioning - or equivalence class partitioning is a software testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In principle, test cases are designed to cover each partition at least once.

- Equivalence Class - a part of the input/output area where the behavior of a component or a system is considered to be the same based on the specification / a data set that is processed in the same way and produces the same result.

- Boundary value - an input or output value that is on the edge/border of the equivalent class or at the smallest distance from both sides of the edge, for example, the minimum or maximum value of the area.

\* There's a higher possibility of getting errors in the boundary areas that the other areas.

~Practical Task: Web form testing~

- Which fields of the registration form can the following negative equivalence classes apply to?

- uppercase / lowercase letters in Cyrillic / Latin

- special characters except the + sign

- less than 10, 11, 12, more than 13 characters

- empty field

Answer: These negative equivalence classes can be applied to the phone number field.

- Define the boundary values of the "Departure Date" field on the train ticket order form, provided that the ticket can be purchased 45 days before departure.

Answer: Boundary values for the "Date of departure" field will be: the day before the current date, the current date, 45th day after the current date inclusively, 46th day after the current date inclusively.

~Change-related testing~

- Smoke Testing

- Regression Testing

- Build Verification Testing - a set of tests run aimed at determining the compliance of the released version with the quality criteria to start testing. It is the analog of smoke testing aimed at accepting a new version for further testing or operation.

- Sanity Testing - Sanity testing is a kind of Software Testing performed after receiving a software build, with minor changes in code, or functionality, to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

The objective is “not” to verify thoroughly the new functionality but to determine that the developer has applied some rationality (sanity) while producing the software. For instance, if your scientific calculator gives the result of 2 + 2 =5! Then, there is no point testing the advanced functionalities like sin 30 + cos 50.

~Testing without requirements~

- The main reasons for absence of requirements

- Lack of resources to define the requirements

- The presence of mediators between the customer and the project manager.

- The unwillingness of the customer to spend money on the "formal" project description.

- Solving the issue of lack of requirements:

- ask clear & accurate questions

- perform regular product reviews with the project and customer teams;

- use ad-hoc and exploratory testing techniques; use High-Level Checklist;

- use the information collected from similar products;

- discover competitive products in the market;

- use your experience while planning.

- Ad-hoc or intuitive testing - is an informal / unstructured type of testing conducted without test documentation, test planning, test design and test cases.

> ad hoc - similar: impromptu - done without planning.

- Explanatory Testing - a more formal version of ad-hoc where the tester actively controls test design while tests are performed and uses information obtained during testing to improve tests.

- [Comparison of Ad-hoc & Exploratory Testing](https://drive.google.com/file/d/1mYR8epvniUrftCpDR0zmzdWpmPhzr9wW/view?usp=share\_link)

- Monkey Testing - is a software testing technique in which the tester enters any random inputs into the software application without predefined test cases and checks the behavior of the software application, whether it crashes or not. The purpose of Monkey testing is to find the bugs and errors in the software application using experimental techniques.

1. In Monkey Testing the tester (sometimes developer too) is considered as the ‘Monkey’

2. If a monkey uses a computer, he will randomly perform any task on the system out of his understanding

3. Just like the tester will apply random test cases on the system under test to find bugs/errors without predefining any test case

- [Ad-hoc and Monkey Testing](https://drive.google.com/file/d/1eRNVBGrdJQg9umnONriwGkwNL6oMRanF/view?usp=share\_link)

- Guides to testing without requirements:

- Checklists

- Testing tours

### Lecture 4. Software testing life-cycle.

[Lecture #4. Fundamentals of Software Testing | Free online courses from QATestLab](https://en.training.qatestlab.com/tag/lecture-%e2%84%964-fundamentals-of-software-testing/)

[L4How to do Homework №4.pdf](https://drive.google.com/file/d/1Rv2aZ9kR6WOvBJMoGXwddpXplf88AZYi/view?usp=share_link)

[Helpful materials. Lecture 4](https://drive.google.com/file/d/19HrTrPlfVao4EVTsF3gApS4U_EzggGn6/view?usp=share_link)

[Lesson 4 materials](https://drive.google.com/file/d/1JV52wmuAuTehg0sxfln05twuU7xbdWug/view?usp=share_link)

Quality Assurance (QA) - is a set of activities focused on providing the assurance that quality requirements will be fulfilled.

Quality Control (QC) - is a set of activities designed to evaluate the quality of a component or system - the process of finding bugs. The purpose of the QA is to maintain the level of product quality.

SDLC - the activities performed at each stage in software development and how they relate to one another logically and chronologically.

Software Development Models - are the set of processes or methodologies that are being selected for the software development. There are few models and each has its own processes or methodologies. Most used models nowadays are Waterfall, V-model, Incremental, Agile, Iterative and Spiral.

Software Testing - Software testing is one of the techniques of quality control and it consists of

- Test Management,

- Test Design,

- Test Execution and

- Test Analysis

Planning - is a set of actions aimed to determine main goals and tasks. It's necessary to make sure during the planning process that goals and needs were determined correctly and risks were fully analysed.

Test Plan - is a documentation describing the test objectives to be achieved and the means and the schedule for achieving them, organized to coordinate testing activities. This document describes: test items, features to be tested, features not to be tested, test deliverables, testing tasks, environmental needs, responsibilities, staffing and training needs, schedule, risks, etc.

Specification - is a complete documentation of the test design, requirements, test cases, and test scripts for a specific test item.

#### Testing as a method of QA

Software Quality - the degree to which a component or system satisfies the stated and implied needs of its various stakeholders.

#### STLC

SDLC life cycle is a process of developing software through a phased manner in the following order:

1. Requirements Gathering - Gather as much information as possible about the details & specifications of the desired software from the client
2. Design the software - Plan the programming languages like java, C, python etc depending on what is most suitable
3. Build the software - Coding
4. Test and debugging the software
5. Deployment - Installation, training users.
6. Maintenance

##### SDLC Methodologies:

* Waterfall method - is a breakdown of project activities into linear sequential phases like analysis, design, construction, testing, deployment and maintenance. Each activity is performed during each phase; there is no way back.

The Problem with the Waterfall Model is that testing in the model starts only after implementation is done.

* [V-Model](https://www.guru99.com/images/1/122118_0652_SDLCvsSTLCW1.png) - is an extension of the waterfall model. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The testing is a confirmation point to move to the next phase and in the V-Model it starts from the very beginning - requirements analysis phase.
* Iterative Model - Iterative Model is a software development model that does not require full specification of requirements, but rather a development process aimed to create a part of the project, a basis for next parts. This process is iterative.
* Spiral Model - is a risk-driven software development process model. All the software development of this model is a spiral, which starts from planning and ends with the working prototype. As a result, at the end of each circle we have tested a working prototype.
* Agile Model - is a set of various approaches to software development. Most agile development methods divide the product development work into small increments that minimize the amount of up-front planning and design. Each separate iteration is a small self-sufficient project. The main idea of Agile is an ability to apply fast changes due to both close internal and external communications.

Agile is a set of various approaches to software development. It has values and principles rather than practices.

The main ideas of Agile are:

* Individuals and interactions over processes and tools.
* Working software over comprehensive documentation.
* Customer collaboration over contract negotiation.
* Responding to change over following a plan.

##### Agile methodologies:

Scrum - is the Agile methodology. Scrum is a set of principles that makes limited in time (2-4 weeks) software development possible. As a result end users may get additional pieces of working software.

The main Roles in Scrum:

* Scrum Master helps to ensure the team follows the agreed processes (like Scrum meetings) in the Scrum methodology, often facilitates key sessions and encourages the team to improve.
* Product Owner represents the product's stakeholders and the voice of the customer.
* Scrum Team has from 3 to 9 members and can include researchers, architects, designers, data specialists, statisticians, analysts, engineers, programmers, testers, etc.

Scrum poker is gamified technique for estimating effort needs. It is held at a meeting, before the software or its new functions development. Each member of Scrum Team has a deck. There are 2 most popular types of decks for Planning Poker: A typical deck has cards showing the Fibonacci sequence including a zero:0, 1, 2, 3, 5, 8,13, 21, 34, 55, 89, «?» and coffee cup card.

Other type of deck uses the sequence: 0,1/2,1, 2, 4, 8,16, 32, 64, coffee card and 3 empty cards

Agile Tools:

* Rally,
* Jira Agile
* HP Agile Manager
* YouTRACK
* TeamBridge
* Hp Agile Manager
* VersionOne
* Worksection
* Google Docs

#### The role of testing in STLC

If testing is done earlier in the SDLC the less costly it will be to fix an issue.

Reasons for Defects in software:

* Everyone makes mistakes
* lack of time
* code complexity
* infrastructure complexity
* changes in the technologies

#### 4 phases of testing:

1. Pick the action
2. Perform the action (discover the actual result)
3. Check the expected result
4. Compare the results

1. Requirements Analysis

* Requirements-based testing
* Functional specification testing
* Creating test cases on the early stage

1. Design

* Docs testing
* UX testing
* Creating test

1. Development

* Code testing, unit testing, integration testing, system testing
* Ul testing

1. Testing

* Full functional and non-functional testing
* Regression testing
* Acceptance testing

1. Release

* Handling users' issue feedback
* Bugs verification
* New functionality testing

There are numerous development life cycle models. Development model selected for a project depends on the aims and goals of that project. Testing is not a stand-alone activity, and it has to adapt the development model chosen for the project. In any model, testing should be performed at all levels i.e., right from requirements until maintenance.

#### Software Requirements Specification (SRS)

SRS is a description of a software system to be developed, its functions, abilities and restrictions.

The characteristics of SRS:

* Atomicity
* Completeness
* Consistency
* Traceability
* Relevance
* Testability
* Unambiguousness
* Verifiability
* Liability
* Comprehension

#### Software Testing Life Cycle:

1. Test Planning

* Specification
* Test plan
* Scope of work
* User stories

1. Test Design

* Test case
* Checklist
* Traceability matrix

1. Test Execution

* Bug report

1. Results Analysis & Reporting

* Test result report

#### Roles in the software testing process

* Test Manager
  + Planning;
  + Cost estimation;
  + Selection and training of the team;
  + Test strategies set up;
  + Decisions making and tools selection.
* Test Lead
  + Team management in the tech field;
  + Team and process control and cost reporting;
  + Specification analysis;
  + Test process improvement.
* Test Designer
  + Test cases development;
  + Test suites development;
  + Test cases improvement and update.
* QA Engineer
  + Manual test execution;
  + Bugs reporting;
  + Requesting for improvements;
  + Final results reporting;
  + Communication with the dev team.
* QA Automation Engineer
  + Auto Test architecture development;
  + Framework set up;
  + Test cases development and update;
  + Test data set up;
  + Test execution analysis.

#### Test Plan

The test plan answers the next questions:

* WHAT should be tested (object)
* HOW will you test
* WHAT are the criteria for starting and completing testing?
* WHAT will be tested
* WHEN will you test?
* WHO is in charge or will perform testing?

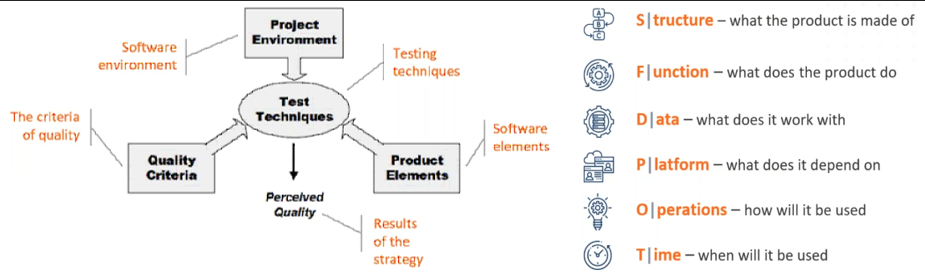
#### Test Strategy & Scope of Work

Test strategy - is a short documentation aligned with the test policy that describes the generic requirements for testing and details how to perform testing within an organization.

It is necessary for test strategy to be short. All the details of the future testing efforts are NOT included.

Test strategies are also known as high level test plan, Master Plan / Master Test Plan or Scope of Work

#### Test Strategy Development Plan



1. Project Environment - all the elements around the project: team, roadmap, resources, artefacts, reports.
2. Product Elements - are things that you intend to test. In the HTSM model we have useful SFDPOT (San Francisco Depot) mnemonic to structure the methods of project elements' discovery.
3. Quality criteria. What quality criteria are required for us? It's often case to have contradictory quality criteria. It's crucial to determine them precisely and set up here a goal (the level of quality) to achieve.
4. Test techniques. These are the techniques you may use. In this part of the model are named main and well-known techniques (functional testing, domain testing, risk- based testing).
5. Perceived Quality - is a result of testing strategy. Achieved goals as a result of testing techniques.

### Lecture 5. Test design & test cases.

[Lecture #5. Fundamentals of Software Testing | Free online courses from QATestLab](https://en.training.qatestlab.com/tag/lecture-%e2%84%965-fundamentals-of-software-testing/)

[Examples of testcases in English](https://drive.google.com/file/d/1gaBHIGoLdfrFsBSqmoY0nYoCnpUREN9p/view?usp=share_link)

[Helpful materials. Lecture 5](https://drive.google.com/file/d/1D1o484AHfwDjw3FVyHzf7CbdgklaXRCX/view?usp=share_link)

[How to do home assignment #5](https://drive.google.com/file/d/16s0MjcN2zXB94iINqjOyY2PDAxgu0TTc/view?usp=share_link)

[Lesson 5 materials](https://drive.google.com/file/d/1lsSWeFFNz1gifu2xBu79Hk43dAl3_3iC/view?usp=share_link)

[Testcase creating in TestLink](https://drive.google.com/file/d/1-hXBh8qvpOV02yp6v1TNtnBKzI8--CSb/view?usp=share_link)

* + Test Design
  + Test Suite
  + Test Case
  + Test Case Attributes
  + Main Test design Techniques - Equivalence Partitioning & Boundary Value Analysis

Test design Techniques

- Equivalence Partitioning / EP

- Boundary Value Analysis / BVA

- Error Guessing

- Cause/Effect / CE

- Decision Table

- Pairwise Testing

Test Case Attributes

- Test Case ID -

- Test Case Title

- Summary

- Preconditions

- Steps to reproduce

- Expected Result

What's the difference between a test case and a bug report?

Tools for Creating Test Cases

- TestLink

- TestRail

Test cases Quality Criteria

Test Case Lifecycle

### Lecture 6. Mobile application testing.

[Lecture #6. Fundamentals of Software Testing | Free online courses from QATestLab](https://en.training.qatestlab.com/tag/lecture-%e2%84%966-fundamentals-of-software-testing/)

[Examples of the mobile bug reports](https://drive.google.com/file/d/1uGZRm0YKUzoi2njDcHLTsp9ecenUm0dd/view?usp=share_link)

[Helpful materials 6](https://drive.google.com/file/d/1tPasAOaOCwNOe_cih6JI6LhlkU_rMizz/view?usp=share_link)

[How to do homework №6](https://drive.google.com/file/d/1qoHqUM6GiZKVcAm7VAdSIGtgri1WWebu/view?usp=share_link)

[Lesson 6 materials](https://drive.google.com/file/d/1Bn544wyeYzRAHyu6kdvsEJYPpHDlZRrS/view?usp=share_link)

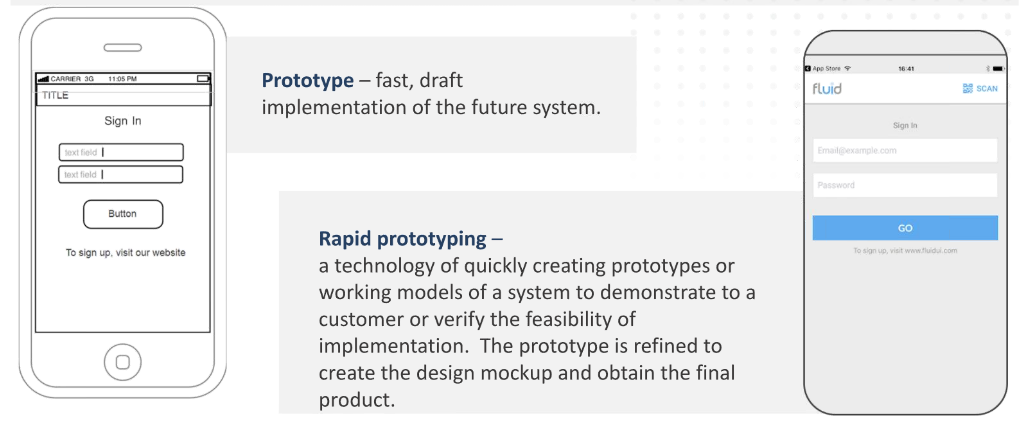
#### Glossary

* Mobile operating system (mobile OS) - is an operating system for smartphones, tablets or other mobile devices.
* Mobile application - is a special application designed to run on a specific mobile platform (iOS, Android, Windows Phone).
* Mobile website – is a site adapted to display and function on a mobile device.
* User Interface - a set of tools, methods and rules for interacting with any human-driven system.
* Mock-up - a model for the layout, stage, or other objects implementation, made in full size and copying a working instance. The model is as close as possible to the final version.
* Crash - an application unexpected exit.
* Crash Log - a file that contains all information about an efficiency error or an application crash.
* Emulator - a device, computer program, or system that accepts the same inputs and produces the same outputs as a given system.
* UDID (Unique 9 Device Identifier) - a unique device identifier, consisting of 40 characters (for iPad, iPhone or iPod Touch).
* Prototype - fast, draft implementation of the future system.
* Rapid Prototyping - a technology of quickly creating prototypes or working models of a system to demonstrate to a customer or verify the feasibility of implementation. The prototype is refined to create the design mock-up and obtain the final product.

#### Stages of mobile application development

1. Designing technical documentation
2. Designing user interface
3. Creating a design concept
4. Rendering all screens
5. Development
6. Testing
7. Debugging
8. Regression testing
9. Creating an application icon
10. Launch on the mobile application marketplace. Publishing of the application on the marketplace includes the following stages:
    * Uploading application file;
    * Posting information materials;
    * Consideration of the application by the administration and approval it to the marketplace
11. Testing of new versions + Regression testing

#### Design Concept



#### How to find the device UDID?

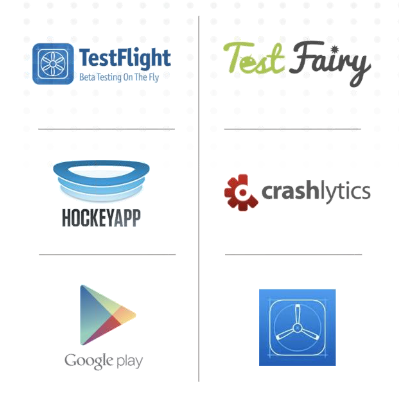
UDID (Unique Device Identifier) - a unique device identifier, consisting of 40 characters (for iPad, iPhone or iPod Touch).

In order to find serial number/identifier (UDID) you need to:

1. Launch iTunes application.
2. Connect the device.
3. Select iPhone or iPad in the «Devices» section.
4. Left-click the «Serial Number» entry in the «Summary» tab.
5. The serial number will switch to the UDID.

#### Services for beta-testing

|  |  |
| --- | --- |
| **Platforms** | **Services for beta-testing** |
| iOS | TestFlight, Test Fairy, HockeyApp, Crashlytics |
| Android | Android Alpha/Beta testing, Test Fairy, HockeyApp, Crashlytics |
| Windows | HockeyApp |



#### The main differences between mobile and desktop applications:

* Screen
* Sensors and input devices
* Phone
* Energy consumption
* Features of the platform
* The presence of cameras
* Additional devices (e.g., smartwatch)
* Narrow specialization
* Updates
* Network

#### What to test?

1. Screen size and touch-interface:
   * All the elements need to be at a such size that users can tap on them.
   * The response speed of elements needs to be high enough.
2. Support landscape & portrait mode / The correctness of displaying data in landscape & portrait mode
3. Phone resources:
   * Efficiency after updating OS
   * Memory leak
   * Saving data in the browser cache (working principle of cache in mobile browsers is analogous to desktop version)
   * Saving data in the application cache (Settings → Applications → Application of interest → Clear cache)
   * Out of space for installing or working application
   * Install to SD card
   * Battery life test
4. Different functions on devices:
   * Headset, native speaker and Bluetooth headset
   * The absence/presence of a camera: front and back camera
   * Work and quality test of a video
   * The absence/presence of GPS
5. Different screen resolutions and OS versions:
   * Application installs on the supported devices
   * Retina and common screens. The text and elements of the interface are displayed smaller on the retina-screen
   * OS versions. Application should not be installed on the unsupported devices. Supporting the necessary media-files on specific model and OS
6. Application response to external interrupts

* Incoming and outgoing SMS and MMS
* Incoming and outgoing calls
* Battery removal
* Unplug and insert SD card
* Device charging
* Work with the physical keyboard (if it is present in a list of supported models) - line breaks, moves through them and etc.
* Lock/unlock a screen. It is on / off screen by a lock button on the device
* Compatibility with other applications

1. Turn Wi-Fi on / off

Example of the incorrect message after leaving a Wi-Fi zone.

Expected result:

The informative message is displayed, for example, «Check the internet-connection».

1. Work in background mode

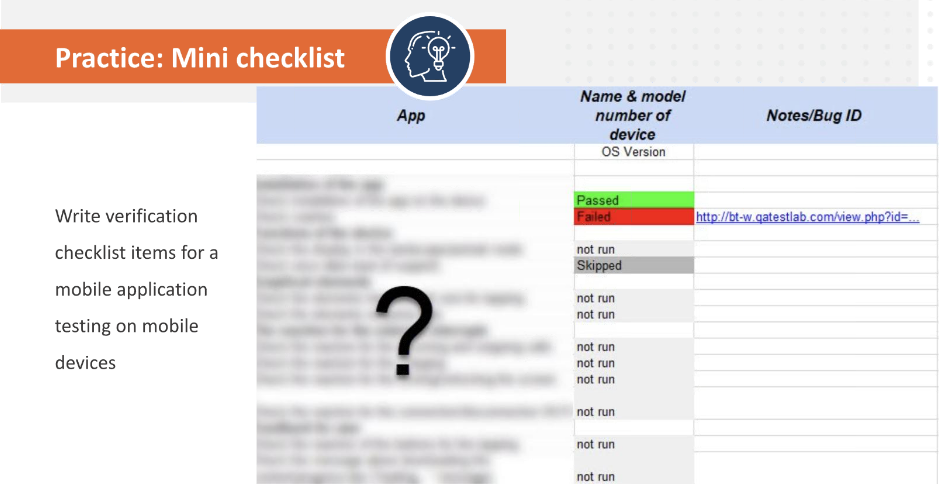
* For sending an application in background mode: first launch an application, then tap on the "Home" button

Example of the incorrect coming out of background mode.

1. Ongoing feedback with the customers:
   * Button response to tapping
   * n Messages during loading content/progress-bar
   * Messages when trying to delete the important information
   * Screen/message about the finished process
   * Availability and timing (synchrony) of the audio and vibration notifications with the notifications on a screen
2. Internationalization

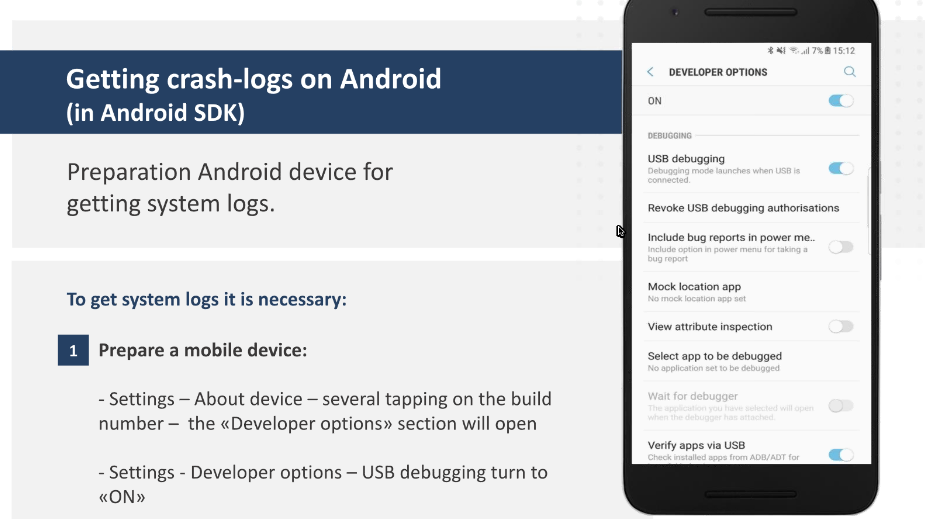
(Test in portrait and landscape modes!)

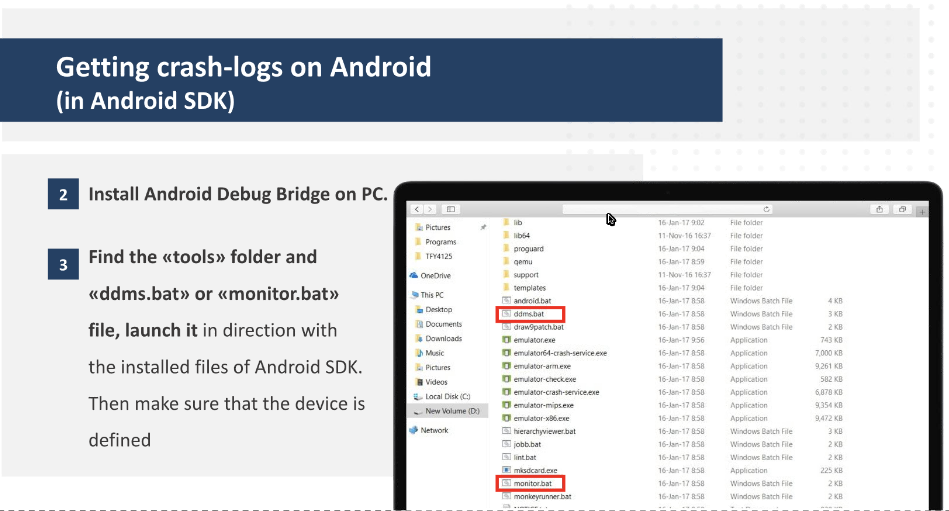
* + Translation test. Text should be meaningful and should not be mixed in more than one language in one interface.
  + Test that all labels are displayed within the corresponding forms/buttons and do not overlap the field data.
  + Test that the text is understandable and not to cut if the text contains the ellipsis marks
  + Language should be switched fast, without relaunching application.
  + After the first application launching the language should be automatically set in the application in accordance with the language of the device.
  + Date format test



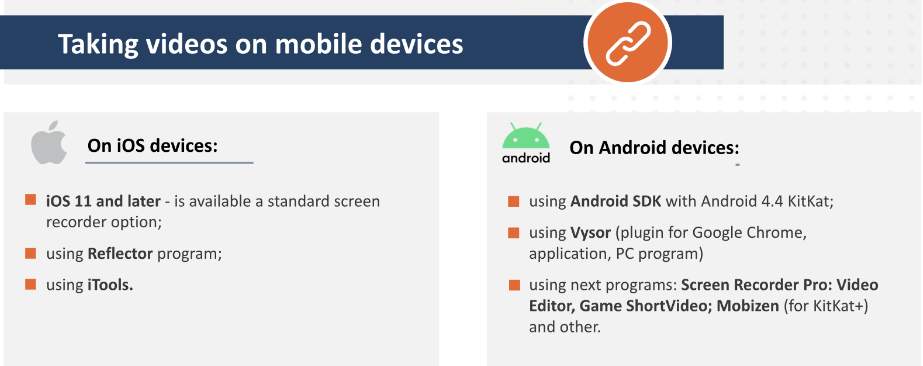
#### Getting Crash Logs







#### Taking screenshots and videos on mobile devices



#### Using iOS, Android emulators

Emulator - a device, computer program, or system that accepts the same inputs and produces the same outputs as a given system.

Advantages:

* Rapid testing of the application, when the target mobile phone is unavailable (or turns out in short supply);
* Testing of difficult or dangerous scenarios, which is impossible or not recommended to test on real mobile phone (e.g., tests that can break a phone in any way or violates terms of the agreement with cellular operator).

Disadvantages:

* Often emulators are very demanding to resources, the most quality of them emulate the application work from the lowest level;
* Testing on an emulator can not repeat the behaviour of a real device on 100% and guarantee the identical result; It is impossible to work with Bluetooth and calls or to test GPS work and mobile network correctly on emulators.

#### Recommended emulators:

1. iOS and Android browser emulator Appetize.io.

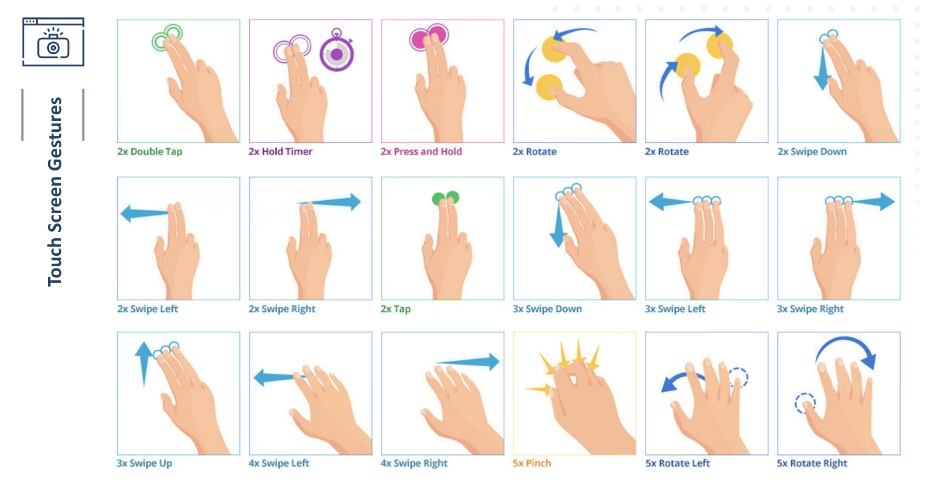
Appetize.io - browser emulator for iOS and Android applications. Also has the ability to use demo iOS.

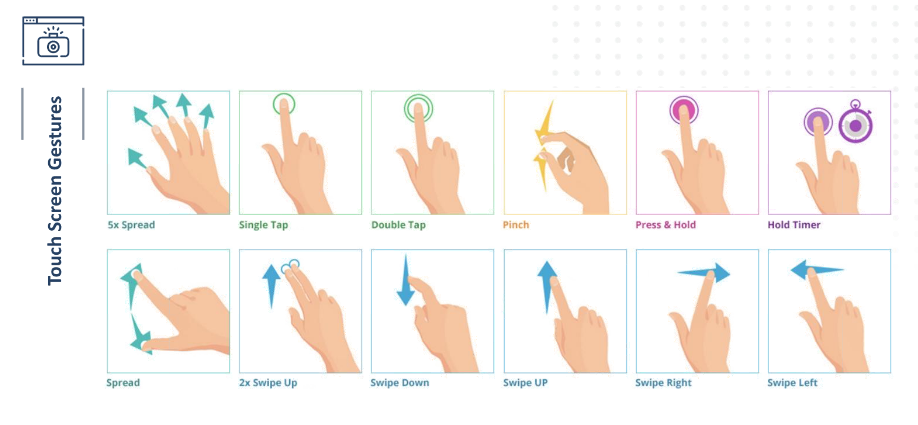
1. Android emulator Blue Stacks for Windows PC and Mac OS X.

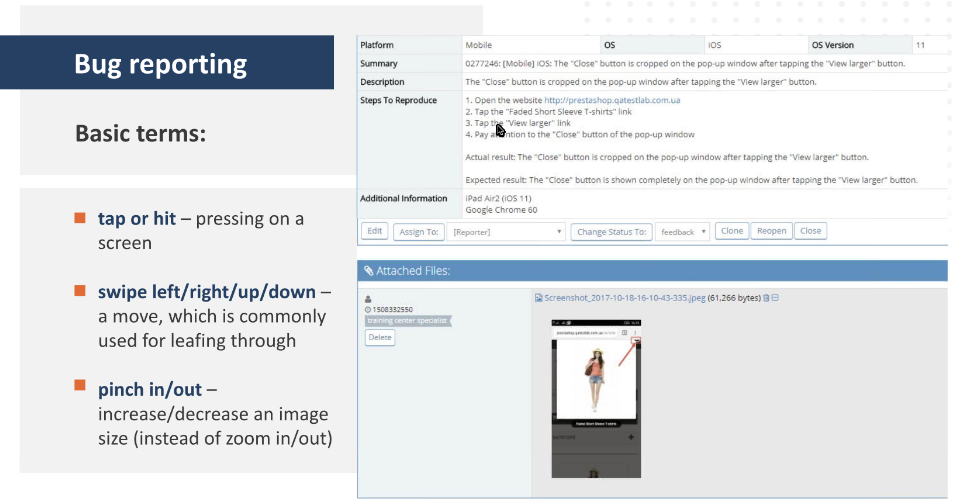
Blue Stacks - one of the most popular program to launch Android games and applications not only in Windows, but in Mac OS X.

1. Android emulator Nox for Windows PC and MacOS.

Nox - good emulator for testing Android games and applications on PC.







### Lecture 7. Game testing.

[Lecture #7. Fundamentals of Software Testing | Free online courses from QATestLab](https://en.training.qatestlab.com/tag/lecture-%e2%84%967-fundamentals-of-software-testing/)

[Examples of game bug reports](https://drive.google.com/file/d/1-zN9AD_3FgiwyDSJX-xrCLvkkHc2SLo0/view?usp=share_link)

[Helpful materials 7](https://drive.google.com/file/d/1oTgqrRC1VkrnhcO5Gwgs30dTROE-QLyT/view?usp=share_link)

[How to do Homework assignment 7](https://drive.google.com/file/d/1xwRd01c58ny2vWexz8jJdQG6nKD5ZQYy/view?usp=share_link)

[Lesson 7 materials](https://drive.google.com/file/d/1n3TYtIfnlMcKuSRFrq7LC5Mvuf0q37tP/view?usp=share_link)

* 3D Model - a geometric three-dimensional object in the game, consisting of points in space and polygons (faces) that form these points.
* Texture - an image that is imposed on the surface of the model, giving it a color, coloring or illusion of relief. Using of textures can be easily imagined as a pattern on the surface of a sculptural image.
* Animation - is a frame-by-frame change of the image to create a motion effect. In 3D games, it is a frame-by-frame change in the coordinates of objects or their individual parts (possibly points) to create a motion effect.
* Camera - an object that is the user's point of view in games. In 2D games, the camera is also present, but it is orthogonal (without perspective distortion) and its direction is fixed.



#### First-Person View (FPV)

* The camera position allows the player to feel like a game character.
* The player sees what the character sees.

The most popular type of first-person video game today:

Shooter games: S.T.A.L.K.E.R., Metro, Far Cry

Platform games: Mirror's Edge, Portal

RPG: Skyrim, Kingdom Come Deliverance



#### Third-Person View

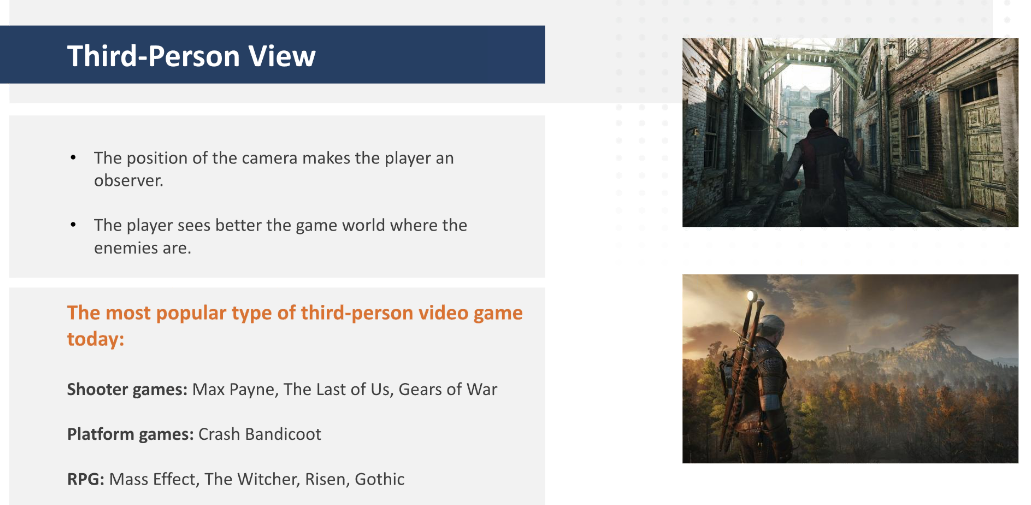
* The position of the camera makes the player an observer.
* The player sees better the game world where the enemies are.

The most popular type of third-person video game today:

Shooter games: Max Payne, The Last of Us, Gears of War

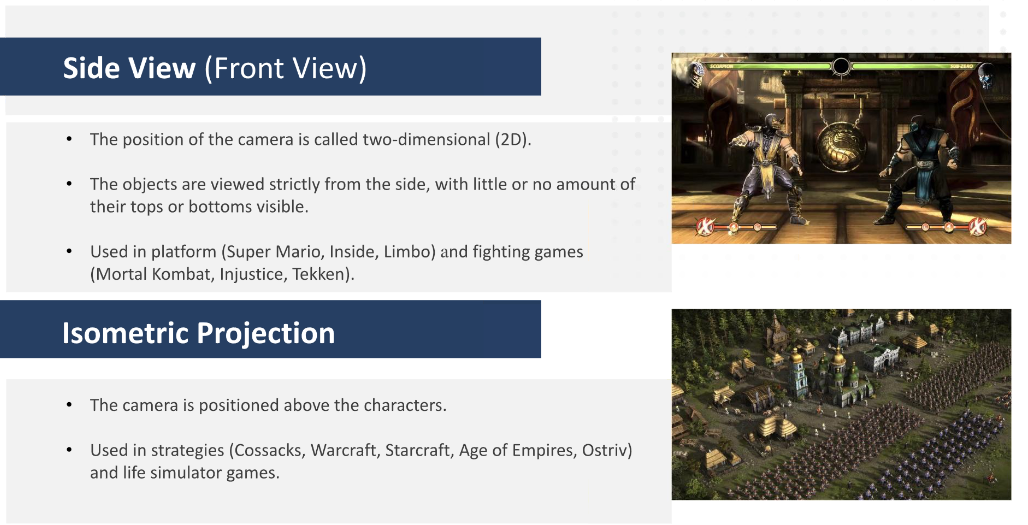
Platform games: Crash Bandicoot

RPG: Mass Effect, The Witcher, Risen, Gothic



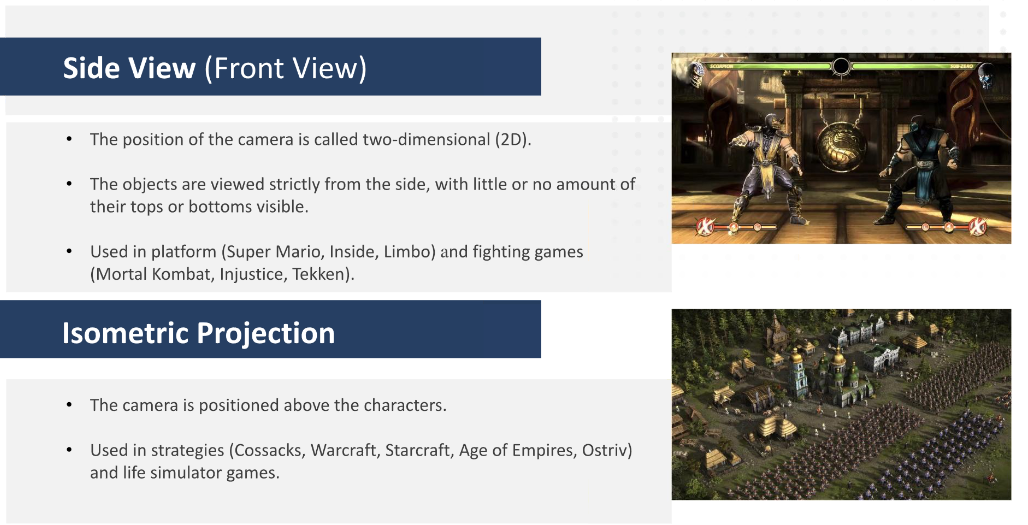
#### Side View (Front View)

* The position of the camera is called two-dimensional (2D).
* The objects are viewed strictly from the side, with little or no amount of their tops or bottoms visible.
* Used in platform (Super Mario, Inside, Limbo) and fighting games (Mortal Kombat, Injustice, Tekken).

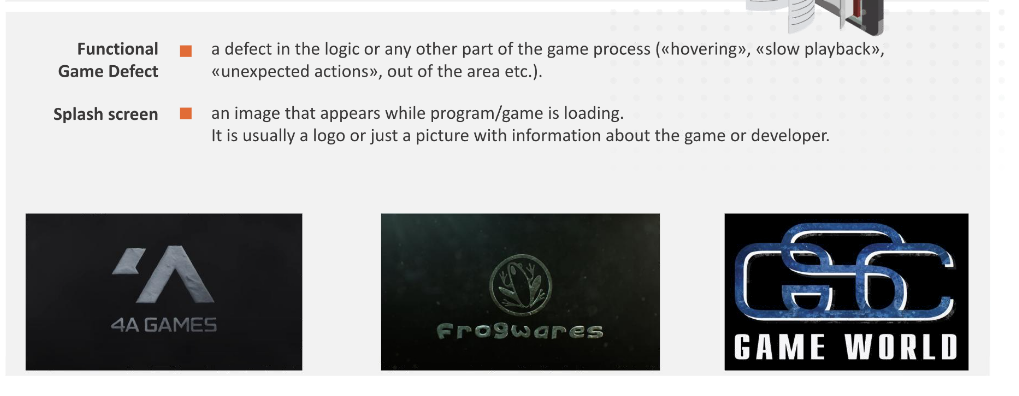


#### Isometric Projection

* The camera is positioned above the characters.
* Used in strategies (Cossacks, Warcraft, Starcraft, Age of Empires, Ostriv) and life simulator games.



* Functional Game Defect - a defect in the logic or any other part of the game process («hovering», «slow playback»,
* Game Defect - «unexpected actions», out of the area etc.).
* Splash screen - an image that appears while program/game is loading. It is usually a logo or just a picture with information about the game or developer.



* Cheat Code - an ability to test separate parts of games (levels, locations, scenes), go to them without going through the most part of the game and also add unlimited resources to characters (money, weapon etc.).
* ADB (Android Debug Bridge) - «Debug Bridge for Android» or «Android debugging interface».
* Stuck - a stuck of a game character that can stop gameplay. Mainly because of poor design or game-level geometry.
* Lag - a delay in the game in the form of a temporary suspension of the game process.
* FPS - Frames Per Second.
* Freeze - a temporary freeze in the game.
* Gameplay - this is what differentiates a computer game from non-interactive forms of entertainment such as books or movies.
* Game Mechanics - sets of rules and feedback loops that are designed to create fun gameplay
* Localization Testing - a process of checking the correct display of application texts in supported languages.
* Lockit - a document provided by the developer, which lists all the application texts in all supported languages.
* Umlaut - in linguistics - different superscript, subscript and inline characters, which can be used in alphabetic and syllabic writing systems not as independent designations of sounds, but to change or clarify the meaning of other signs.
* Landing page - an advertising insertion that allows the user to see other released projects of the company, subscribe to newsletters.

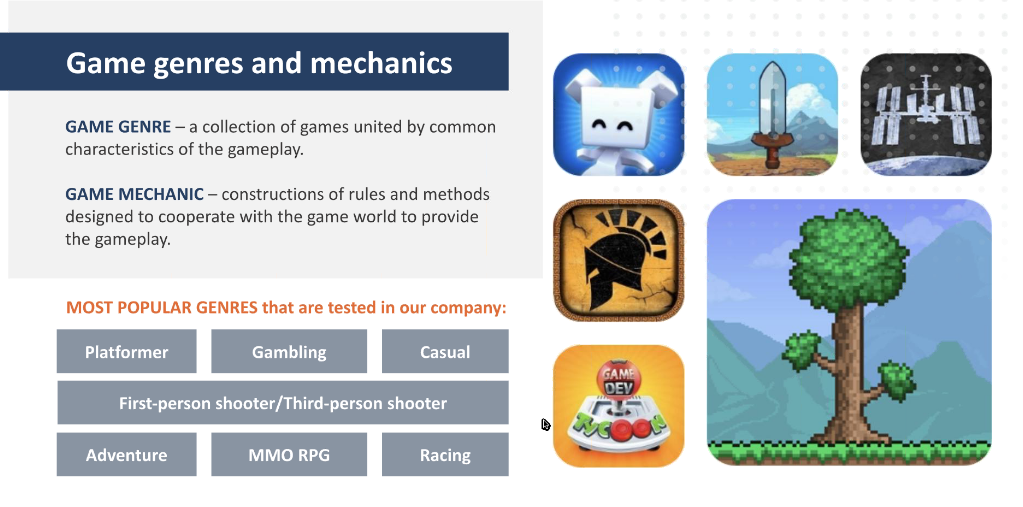
#### Game genres and mechanics

Game genre - a collection of games united by common characteristics of the gameplay.

Game mechanic - constructions of rules and methods designed to cooperate with the game world to provide the gameplay.

Most popular genres that are tested in our company:

Platformer Gambling Casua| First-person shooter/Third-person shooter Adventure MMO RPG Racing



#### Performance measurement in games

Game Performance Testing:

* Measurements of the maximum, minimum and average FPS.
* Loading speed of game and levels.
* PC CPU load measurements.

To measure the frame rate you can use such tools:

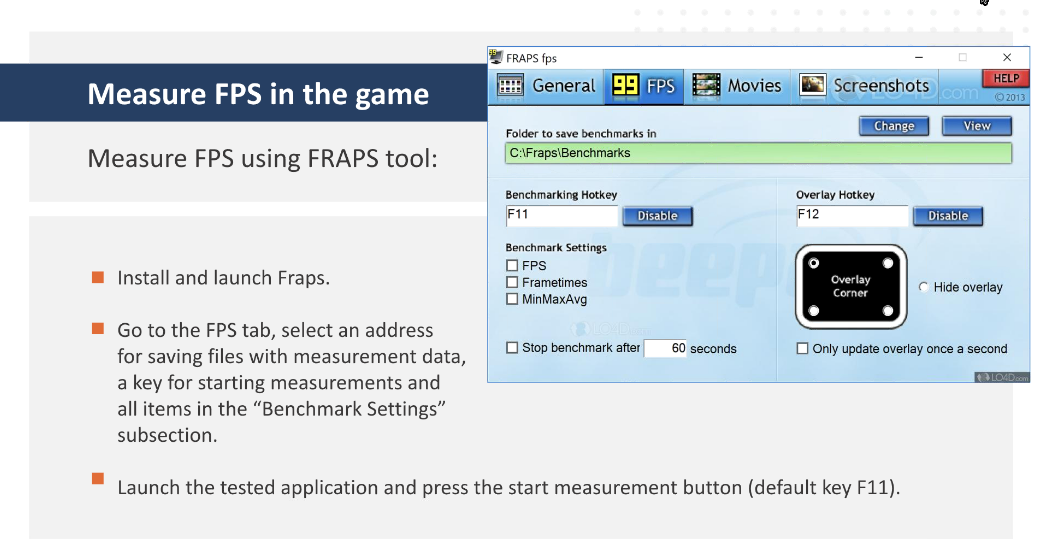
MSI Afterburner, Fraps, FPS Meter (Android), GameBench (Android and iOS ).

FRAPS - a program for recording videos, creating screenshots. Also, it has the functionality of displaying FPS overlay and benchmarking with saving to a file (helps during determining maximum, minimum and average FPS).

#### Measure FPS in the game

Measure FPS using FRAPS tool:

* Install and launch Fraps.
* Go to the FPS tab, select an address for saving files with measurement data, a key for starting measurements and all items in the "Benchmark Settings" subsection.
* Launch the tested application and press the start measurement button (default key F11).



* Game genres and mechanics
* Performance measurement in games
* Game functional testing
* Compatibility testing
* Localization testing and peculiarities of bug reporting

### Final Exam

1. What is the summary of the bug on the [video](https://www.dropbox.com/s/3t80vwp0sozeeaa/4.%20Droid%20Jump_Gameplay.mp4?dl=0)? (Several answers are possible)

* The main game character is passed through platforms on the gameplay screen.
* There is no sound while playing the game.
* The platforms are displayed differently in size while playing.
* The one platform isn't displayed on the gameplay screen.

1. Which test design technique is not possible because of the huge number of input values? (Select one answer)

* Error guessing
* Cause/Effect
* Exhaustive testing
* Pairwise testing

1. Arrange the stages of the mobile application development process in a chronological order:

* Development of technical documentation
* User interface development
* Development
* Rendering all screens
* Debugging
* Launching in a mobile app store
* Testing
* Regression testing

1. The main reasons for the lack of requirements are: (Several answers are possible)

* Maintaining several projects simultaneously
* The unwillingness of the customer to spend money on a "formal" description of a small project
* Excessive confidence in the simplicity of application logic and developer competence and user expertise
* Lack of resources to define the requirements

1. Which of the presented mobile operating systems are being developed and manufactured only for devices manufactured by the same company? (Select one answer)

* iOS
* Windows Phone
* Android

1. What symbols are better to use to test input field that takes values from 2 to 15 characters for the least amount of time? (Select one answer)

* 2,4,5,...,15
* 0, 1, 2, 10, 15, 16
* 2, 10, 15

1. Formulate the summary of the bug in the [screenshot](https://eng-clients.qatestlab.com/api/common/picture/c0jjl1kosk4mse4yqxuqqy817) (Select one answer)

* The confirmation pop-up window is displayed on the user's personal information page after filling the valid data
* The invalid date of birth is set on the user's personal information page
* The name of the month is displayed cut off on the right side on the user's personal information page

1. What bug report status does not exist in the bug life cycle? (Select one answer)

* Reopened
* New
* Open
* Deleted

1. Fast, draft implementation of the future system is called a... (Select one answer)

* Picture
* Prototype
* Sample
* Draft

1. What of the listed does NOT apply to the functional testing of an eraser? (Select one answer)

* Check if pencil drawings can be erased
* Check that a pencil can draw on the places where writings were just erased
* Inspect the appearance of the eraser (size, color, inscription...)
* Check that eraser doesn't smear the writing when you erase it

1. The key idea of the eternal test circle sounds like: (Select one answer)

* The more testing, the better the end result
* All answers are correct
* It is necessary to formulate an idea of the ideal product
* The aspiration for excellence – engine of the testing process

1. Which of the following are NOT the priority of a bug? (Several answers are possible)

* Immediate
* Normal
* Minor
* Low
* Urgent
* High
* Critical

1. The message input field is required. Maximum message length is 1088 characters. Which minimum dataset is best for testing this field? (Select one answer)

* 1 and 1088 characters
* 1 and 1088 characters, 0 (empty value), 1089 characters
* 0 (empty value), 1089 characters
* 1, 50, 999, 1088 characters, 0 (empty value), 1089 characters

1. Which of the following are not included in checklist for PC? (Several answers are possible)

* Check design
* Check login in Facebook
* Checking device rotation
* Checking device charge
* Text display

1. Security testing is conducted for (Several answers are possible)

* determine the degree of usability, learnability, comprehensibility and attractiveness of the product for users
* analysis of risks associated with providing a holistic approach to application protection, hacker attacks, viruses, unauthorized access to sensitive data
* system security audit
* analysis of reaction speed of the application to external influences by different in nature and intensity loads

1. Good bug report should answer the following questions:

* Why did the error happen?
* What steps led to the error?
* What did you really see?
* What did you expect to see?
* Can the mistake be corrected?

1. Select the best description for the bug on the [video](https://www.dropbox.com/s/3ou5jqkx1eo0l4z/app_crashing.mp4?dl=0) (Several answers are possible)

* The app crashes in the "me" video group after tapping the "Take Video" button.
* The additional free space is shown above the "Take Video" button in the "me" video group.
* The camera is not opened in the "me" video group after tapping the "Take Video" button.

1. What information can be specified in the "Environment" field when creating a defect? (Several answers are possible)

* Employee, for whom the bug report is being assigned
* Operating system
* Stage of the software development, where the defect was detected
* Mobile device model
* Version of the tested software
* Browser version

1. What programs can you use to get crash logs from Android devices? (Several answers are possible)

* Android SDK
* X-Code
* Xenu
* Logcat
* Catch Notes

1. What is the definition of a logo? (Select one answer)

* A graphic mark, emblem or symbol used to aid and promote public identification and recognition
* The sign of the web page, displayed in the browser tab
* Graphic representation of the future website

1. Select the best description for the bug on the [video](https://www.dropbox.com/s/mgtkj5svpdafed9/Call_Me-Link.mp4?dl=0) (Select one answer)

* Generated tag contains incorrect picture link in the left part of the screen after selecting first two pictures
* Picture link is not changed in the tag after selecting the third link
* It is not possible to select the generated link completely in the "My CallMe-Link" form

1. What is equivalence class? (Select one answer)

* It's a class where all tests lead to the same result
* It's a class that combines all boundary values
* It's a class where all the tests lead to different results

1. Select the best description for the bug on the [screenshot](https://eng-clients.qatestlab.com/api/common/picture/ep567yti4seik8y2lmasnqj1t) (Select one answer)



* The model is partially displayed without hair after choosing the outfit
* The model textures have low detailization after selecting the equipment
* Helmet texture is not displayed on the model after selecting the equipment

1. What is the summary of the bug on the [screenshot](https://eng-clients.qatestlab.com/api/common/picture/9d0mfyl7dwrs6pbl1qgifk2i5)? (Select one answer)

* The "Quote ID" inscription is not horizontally centered on the "Quote Detail" form
* An extra colon is displayed on the "Quote Detail" form after "Quote ID"
* The "Quote ID" is not displayed on the "Quote Detail" form

1. Select the official beta testing service for Windows (Select one answer)

* Crashlytics
* TestFlight
* HockeyApp
* Android Alpha/Beta testing

1. Select the best description for the bug on the [video](https://www.dropbox.com/s/pkdsa06ox3hxed9/Battlefield.mp4?dl=0) (Select one answer)

* The armoured vehicles flew by without helping the player
* There is no place for armoured vehicles to land on the game location
* Armoured vehicles are moved through the air on the game location

1. What browser plugins are used for websites and web applications testing (Several answers are possible)

* Pushbullet
* Asana
* Window Resizer
* Perfect Pixel
* Snagіt
* LiveReload

1. Which test case attribute are NOT required? (Several answers are possible)

* Attachment
* Title
* Expected result
* Steps to reproduce
* Precondition

1. What size should the iPhone game icons be (in pixels)? (Several answers are possible)

* 180х180, 120х120, 87х87
* 144х144, 120х120, 57х57
* 120х120, 80х80, 58х58

1. Select the best description for the bug on the [video](https://www.dropbox.com/s/nuuqlv7s46iht97/Connection.mp4?dl=0) (Several answers are possible)

* Wi-Fi is not connected to the app after activating Wi-Fi.
* The "Not connected" message is shown in the app after activating Wi-Fi.
* Wi-Fi is connected to the app after tapping the "Connection" button.
* The camera is not opened in the app after turning Wi-Fi off.

1. Emulator - is a... (Select one answer)

* specific playing or learning mechanisms by which certain processes are managed
* model of the original software that implements the logic of the operation of this software (partially or completely), simulates the behavior of the software, copies its interface
* program copying functionality and behaviour of another program

1. Which are the methods for solving the problem of the lack of testing requirements? (Several answers are possible)

* Use of high level check-lists
* Use of information collected from similar products
* Ad-hoc testing and exploratory testing
* Automated testing
* Planning with experience from past projects
* Regular product discussions with the project team and the customer team
* Load testing

1. Which of the listed are the features of describing game defects? (Several answers are possible)

* The time spent by the tester to describe the defect must be specified in the description
* The tester added the defect should be specified in the summary
* In the description it is necessary to specify the platform on which the test was performed with the OS version (iPad 3 retina iOS 6.1.3, iPhone 5S iOS 7.0.4)
* In the description it is necessary to specify the build number on which the testing was performed
* In the summary it is necessary to specify the platform on which the testing was performed (iPad, iPhone, Android tablet...)
* During localization testing it is necessary to specify the localization language (abbreviation) in which the testing was performed (not necessary for functional testing)
* In the summary it is necessary to specify the location (level, section, coordinates) of the defect in the game

1. Formulate the summary of the bug in the [screenshot](https://eng-clients.qatestlab.com/api/common/picture/4sz9529xwl4oult75ut6x93sk) (Select one answer)

* The hieroglyph is displayed on the "Language" button on the site header
* The "Language" button is not aligned with other elements on the site header
* The "Language" button is displayed in English on the website header after changing the interface language of the page to Russian