

# TEST JUNGLES IN GAME DEVELOPMENT

IVA POPOVA

**JFTW** JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

# WHO AM I?



**JFTW** JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

Test monkey  
Game enthusiast  
Timeless student



Just For The Win Game development studio - <http://justforthewin.com/>  
A group of game creators, artists, sound designers, developers and testers creating together games for casinos.

# WHAT ARE WE GOING TO TALK ABOUT?

- TESTING TYPES
- TESTING AND PROCESSES
- SURVIVAL KIT OF THE SOFTWARE TESTER

**SURVIVAL** 

# WHY DID I CHOOSE TO TALK ABOUT THIS?



**JFTW** JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

In order to take you to a virtual journey to a **game development reality from a testing perspective.**

# TESTING TYPES



## **Fuzz Testing**

Fuzz testing or fuzzing is a software testing technique that involves testing with unexpected or random inputs.

## **Gorilla Testing**

Exercise one or few functionality thoroughly or exhaustively by having multiple people test the same functionality.

**Age Testing:** Type of testing which evaluates a system's ability to perform in the future.

**Happy path testing** Also known as Golden path testing, this type of testing focuses on selective execution of tests that do not exercise the software for negative or error conditions.

# TRENDS & CHALLENGES

- MOBILE TESTING
- AUTOMATION TESTING
- USABILITY TESTING
- EXPLORATORY TESTING



# MOBILE TESTING

- DIVERSITY
- CLOUD BASED TEST LABS
- ACCESS TO REAL DEVICES
- FEATURE DIVERSITY



- Naturally with the world becoming more and more mobile centric software development also needs to support that mobile world. And it is a world of different brands of devices, with different hardware capabilities, on top of which they operate on even larger mix of operating software – android, ios, windows all the different browsers (speaking of web applications) And it doesn't get better when we sprinkle this nice mix with the software versions as well. You could only imagine the test scope that the mobile world requires. And yes, this is a challenge. A challenge which results in a lot of usage analysis, reporting, anything that can help take a decision to reduce the scope based on facts.
- Many cloud based test labs are built to address this. (e.g. browserstack, google cloud test lab, ghostlab etc. )Providing access to a lot of devices, as well as software versions. Tools to control a set of devices simultaneously also pop up lately. (e.g. browsersync)
- However it is still a need to have access to real devices as simulators are not always 100% accurate, as well as to be able to debug problems easier on site.
- The Producers of mobile devices are in a vey tough competition with each other, which results in feature diversity as well – as the force touch on the new apple phones or the multi window support of android n

This type of testing is both very challenging, but it is also very drastically changing all the time which may take away some of the challenges you have today, but also bring you new to worry about tomorrow.



# AUTOMATION TESTING

- DEV/OPS
- CONTINUOUS AUTOMATION
- CUSTOMER-DRIVEN TDD/BDD
- VISUAL VALIDATION TESTING



JFTW JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

The first interesting concept that has to do with automation is Dev/Ops. This is emerging like a movement just as agile did quite some time ago. In essence, it is about bringing development closer to operations so that they are not separate entities anymore, but rather work together as a team that delivers a product together in the end. In the context of Dev/Ops it is no longer just about functional automation anymore, but rather about focusing on the bigger picture and automating everything that can accelerate the dev process. E.g. scripts to create quicker builds of the applications or performance tests run regularly. This results in configuration management automation to become yet another part of the tester's job as well – so don't be surprised if you see a requirement for skills and knowledge about Puppet or Chef in a quality assurance engineer position.

The next aspect of automation we have nowadays is continuous automation. It is very much related to Continuous Integration and the needs for getting quick feedback on every commit. It also means automation from the start, but as always it comes with the old original challenges when automating a new product – frequent changes in the product and cost of maintenance of the tests.

BDD in short – the requirements are written in a given-when-then specific format (Gherkins) which is then also the basis for test automation and code implementation directly using those same requirements, minimizing any traceability issues. This is not an automation framework in itself as many people confuse it to be. It is rather a means to speak about your requirements as early as possible, trigger discussions

before even coding. A lot of people pick this process up claiming it is a very good way of working and structuring the work, but I have also seen cases when it has not been appropriate, e.g. working with requirements directly with the client - the format brings in too many details they don't care about.

The basic concept of visual validation testing is using image recognition in the automation testing – comparing expected and actual states of a certain application. Popular tools implementing this technique are sikuli, applitools etc.

# USABILITY TESTING



**JFTW** JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

Lately Usability has become an attribute of competitive advantage – every click matters, the more clicks the less user satisfaction -> the less likely it is to return to that application/website.

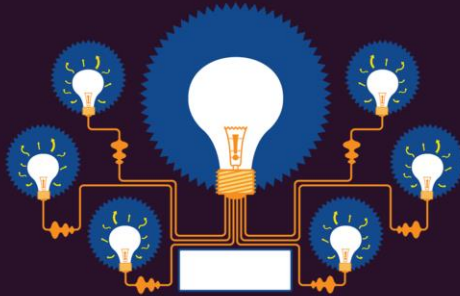
Ease of use, time to understand, time to achieve a task – those are all characteristics developers consider today. They cannot afford not to. At my previous company we organized usability studies with external people, showed them our latest game, noted down their verbal and non verbal reactions to it, interviewed them with specific questions, gave them tasks to accomplish. After which we analyzed the results and followed up with decisions how to improve our game.

Which is a very good method to evaluate the quality of the usability in your products, but the results shouldn't be followed blindly either. A proper analysis of the feedback is a requirement before taking any actions.

# EXPLORATORY TESTING



TEST CHARTERS



MIND MAPS

This is more of a somewhat trendy type of testing that has been going around for some time now. Companies move in the direction of using this method in manual testing instead of using the traditional test cases scripting and executing.

By definition:

**Exploratory testing** simultaneous learning, test design and test execution. As its name implies, exploratory testing is about exploring, finding out about the software, what it does, what it doesn't do, what works and what doesn't work.

It is usually only done by skilled testers

Session based testing – technique used in exploratory testing.

The disadvantage of using this method comes when the testers fail to keep notes of their sessions or to generate a report in the end, meaning all the effort results and knowledge is lost. If that happens there is no traceability or transparency -> no way to measure the progress or the confidence level of the executed testing so far.

# TESTING AND DEVELOPMENT PROCESSES

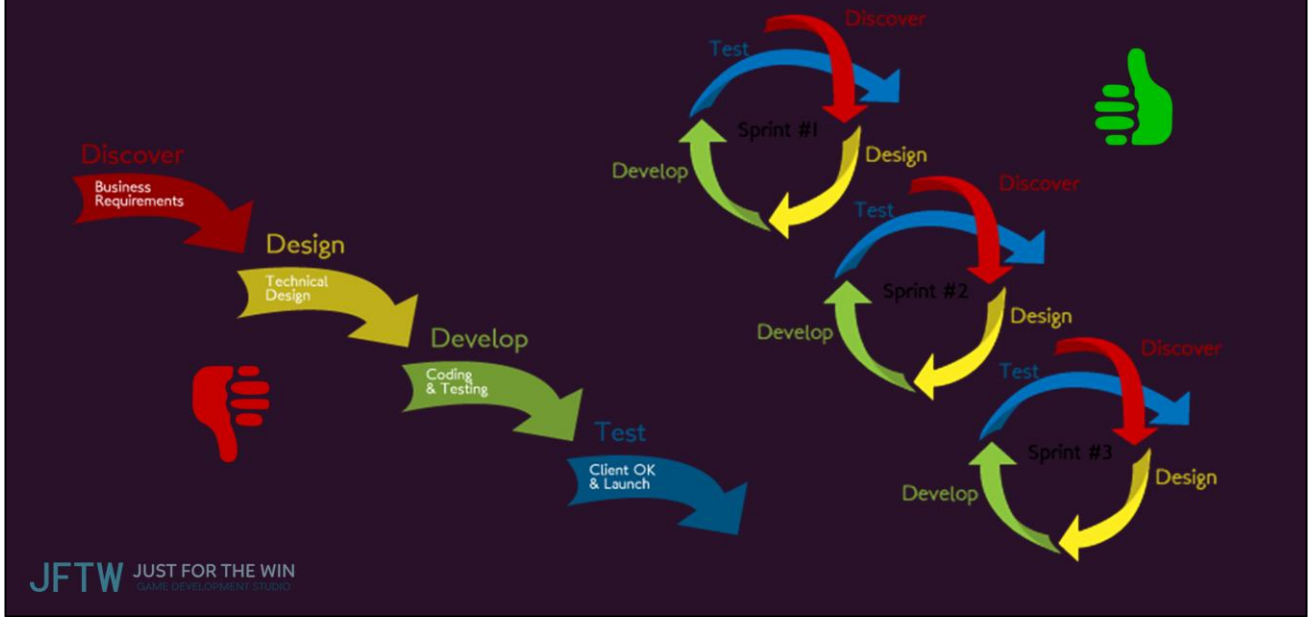


Testing is not a role – it's an activity with a deliverable.

Paul Gerrard

This very much ties in with the fact that software quality is not the tester's job – it is a shared effort. As is testing – you will see that more and more the processes are shifting towards more and more team members doing various types of testing. Which is very natural - the testing needs are going up, and solving the challenge becomes more and more a shared responsibility.

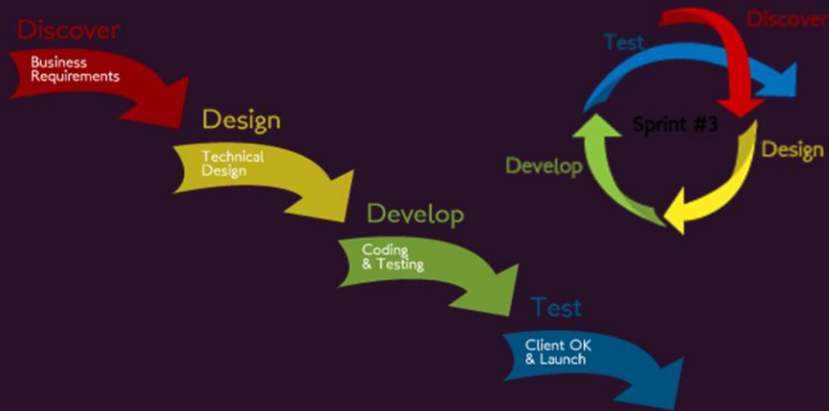
# TESTING AND DEVELOPMENT PROCESSES



1 testing is always dependent on everything else

2 implementing a particular process in an organization rarely ends as initially intended

# GAME DEVELOPMENT PROCESS



**JFTW** JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

The games that we're making are quite small projects, and it should be more or less set what the game should be by the time the development starts.

The stages very much overlap of course, but it is not until the development phase when it is really possible to do a few of those agile iterations.

So the process is more of a mixture of processes, which fits the specific production line of the specific industry.

1 testing is always dependent on everything else – if the mindset for shared testing responsibility is not there, there might be issues which resemble the standard waterfall issues

2 implementing a particular process in an organization rarely ends as initially intended – scrumfall...other mixed methods.

# IMPORTANT BUT RARELY SPOKEN ABOUT



**JFTW** JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

In the end it all comes down to the people you are working with, the relationships you have with them and the caring towards each other.

Team cooperation is crucial

The processes are only tools to create and nourish team cooperation.



# SURVIVAL KIT OF THE SOFTWARE TESTER



**JFTW** JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

Sand watch - In the testing jungle you gotta have **patience**

Magnifying glass - you will be looking for bugs in this test jungle, and so you need to have your **focus** with you

Binoculars – you will have to look out for danger in the distance as well, so that you are **future ready**

Sneakers – to be able to run fast and **adapt**

Magic crystal ball – sometimes a little **black magic** is helpful

# SURVIVAL KIT OF THE SOFTWARE TESTER

BE PATIENT

FOCUS

LOOK AHEAD

ADAPT

...AND ALWAYS CARRY A LITTLE  
BIT OF **MAGIC** 😊



**JFTW** JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO

**Be patient** – always be patient towards the others you work with, and towards yourself too. High quality takes time, that's just the way it is.

**Focus** – learn how to control your focus so that you are always working on the highest priority task for a given moment. There will always be a lot to do, and you will often have to multitask – prioritizing is crucial.

**Look ahead** – try to always keep yourself updated, be future ready. It will help you to gain time, to prioritize and to adapt easier.

**Adapt** – fast pace world requires fast pace learning. Cultivate your ability and motivation to learn and adapt quickly to changes.

**Use Magic** - consider what is worth paying attention, try to always connect the dots – even if they are not there yet

# RECAP

- TESTING TYPES

MOBILE

AUTOMATION

USABILITY

EXPLORATORY



- TESTING AND DEVELOPMENT PROCESSES

- SURVIVAL KIT OF THE SOFTWARE TESTER



KEEP CALM  
THE  
PRESENTATION  
IS OVER :)

ANY  
QUESTIONS?

JUST  
**FTW**  
FOR THE WIN  
GAME DEVELOPMENT STUDIO

THANK YOU!

[iva@justforthewin.com](mailto:iva@justforthewin.com)  
[hello@justforthewin.com](mailto:hello@justforthewin.com)

JFTW JUST FOR THE WIN  
GAME DEVELOPMENT STUDIO