Random Test Data Generation

Vision Document



Course: Bachelor of Science (Honours) in Software Development

Name: Kyle Kinsella

Date when project started: 16/01/2024

Supervisor: Chris Meudec

**Introduction**

The main aim of my project is to develop a test data generation tool to try and assist Developers and Testers to try and make testing an easier way for testing software that is made already or for new software that is being developed.

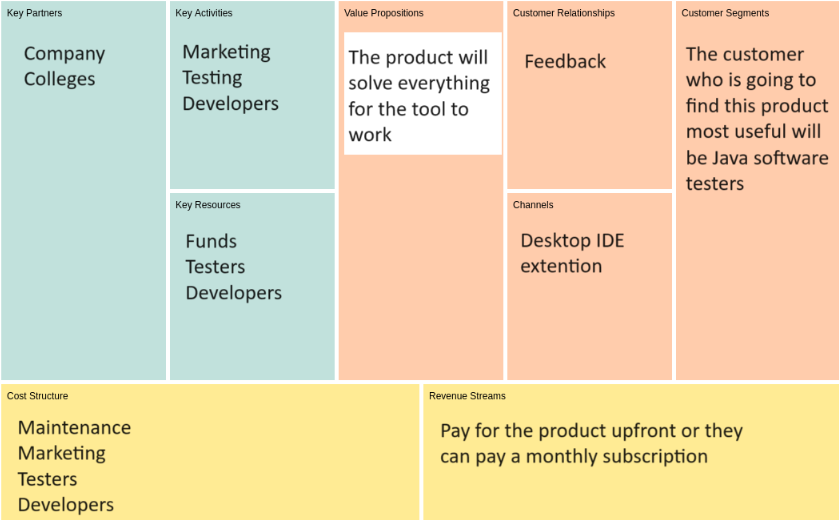
**Business Case Summary**

In today's digitally driven world, the centrality of computers is undeniable, and each machine runs a substantial amount of software. This shows the widespread trust in computer systems, emphasising the critical role software plays in our daily lives. Thoroughly testing software is essential, as any issues that arise could potentially disrupt the functionality of the software. It is crucial for developers to promptly address and resolve these problems to ensure the smooth operation of the software. Testing plays a vital role in undertaking that the software functions consistently. Neglecting the proper testing could lead to project failure, resulting in a waste of time and money [1].

During the development of the software, we need to regularly test our software as we are building it. When we test our software it is very important that we have tests that are efficient, fast and can work on certain test data for example, integers, floating-point numbers, and objects.

The tool I am hoping to develop will try to provide a solidly built solution for generating random but very effective data, this will enhance the use of this tool.

Business Model Canvas



**Stakeholders Description and Goals**

The main users that will be using this tool would be a dedicated Testing team, they will have the ability to be able to test and see if the software works like it should. The aim for the testers is to check and rigorously check to find errors and bugs in the software that the developers have constructed.

During the development of the software there are two teams:

1. Development team
2. Testing team

The Development team do the following: The developers are in charge of writing code to implement into the software, or fix some issues in the code. The development team don’t just code, they have testing responsibilities also, they normally perform unit testing to make sure that units of the code work as they should for example, A method adds two numbers together.

The Testing team does the following: The Testers are dedicated to verifying and validating the software as a whole. A tester can also be called a Quality Assurance (QA). A tester takes part in various types of testing including the following:

Functional Testing - Ensures that the software functions according to specified requirements. [2]

Integration Testing - The tester must verify that the different types of components(recognizable part of a larger program or construction) / modules(an extension to a main program dedicated to a specific function) work together seamlessly. [2]

Regression Testing - Checking that new changes to the software haven’t negatively impacted the existing functionality. [2]

Performance Testing - During this bit of testing the testers assess the softwares speed, responsiveness and scalability, all of these functions are considered as non-functional requirements because the user using the system doesn’t know that they are needed for the product, but if these functions are not implemented into the product the product won’t work as intended. [2]

User Acceptance Testing - The testers in this bit of testing confirm that the software meets the end-users(who are going to be paying for the product) expectations. [2]

**Software Overview**

The tool will automate and generate a diverse test data set. Depending on the selected data set, the tool will generate a corresponding test. Additional features may be incorporated as the project progresses. For example,

public class App {

public static int add(int num1, int num2) {

return num1 + num2;

}

}

public class AppTest {

@Test

public void addUp() {

int num1 = 1;

int num2 = 3;

int ans = App.add(num1, num2);

assertEquals(4, ans);

}

}

The target audience for this project is Java software testers aiming to enhance the robustness of their testing processes. Some of the benefits of this project is that it will include assisting unit testing through the use of automated data generation. The testers that will be using the software will be able to test specific units of the software, the system will do this randomly and automatically do this.

This project is going to be developed in Java. This is a very flexible programming language due to most computers running Java, so therefore there is a requirement which means you must have Java on your computer to run this project otherwise the project won’t run successfully.

**Summary of Software Features**

After the software for the project has been completed there will be some features, but there will not be too many features as this is only a testing tool for software. Below are the features that will be in the software:

**Automated Test Data Generation** - The tool will automate the generation of various test data sets, and make efficient software testing, and potentially boosting testers productivity.

**Flexible Test Scenarios** - The tester can select a specific data set and the tool will generate the corresponding test scenarios for targeting testing.

**Unit Testing Support** - The tool will assist in the area of unit testing by providing automated data generation for testing specific units of the software.

**Target IDE** - The tool that I will develop will specifically be optimised for Eclipse.

**Main Risks Summary**

**Cost of Development:** One of the risks for this project is that the during the development and the maintenance of the project, there could be various of different changes that might have to occur when making the project this could result in more developers to construct the software and more testers to test each and every bit of the software resulting in using more money.

**Amount of users:** As this project is only a tool that is not going to be used by very many users, it is primarily going to be used by software testers, this is one of the biggest risks with this project. For example, this tool will be very useful for software testers but on the other hand it wouldn't serve any purpose for your family members.

**Will companies buy the software:** Depending on the cost to construct the software and maintenance of the software, it is going to cost a lot of money depending on how many developers and testers are on each team within the company that is making the project. This is a risk because the more people on the project means that the more money is going to be spent and therefore the project will cost an exceptional amount of money to make some profit.

**Marketing:** In order for companies to buy our product we will have to do some form of marketing. The risk here is that we will have a challenge to associate creating awareness and convincing certain companies to purchase our product. In order to do this we will have to employ some very capable marketer’s to assist us in planning what to market and how we are going to sell our product.

**References**

1. TechTarget. (2024). Software Testing. Retrieved from:

<https://www.techtarget.com/whatis/definition/software-testing>

Accessed on 17/01/2024

1. Atlassian. (2024). Types of software testing. Retrieved from:

<https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing>

Accessed on 17/01/2024