Roadmap for Software Testing Automation

Prerequisites:

- Java core knowledge
- HTML basic knowledge
- Maven basic knowledge
- Git basic knowledge

A little bit about software testing: What? Why? How?

Software testing is the process of checking and evaluating whether a software product matches the expected requirements while also making sure it is defect free. This involves the execution of the software/ system using either manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, bugs, unexpected behaviours, gaps or missing requirements in contrast to actual requirements. All of these actions need to take place before the product is released to the end user.

Why?

As you start off small, creating applications in a company that uses software to keep track of their transactions, warehouse or employees, you're thinking "Well what's the worst that could happen?", but the reality is a little more complex. As history has taught us, a small bug or error going unnoticed can have devastating consequences in terms of money as well as human lives.
Here are a few examples of notable errors that have marked our recent existence:

- The Explosion of the Ariane 5 - On June 4, 1996, an Ariane 5 rocket launched by the ESA (European Space Agency) exploded just forty seconds after its launch from Kourou in the French Guiana. The rocket was on its inaugural voyage, after a decade of development costing $8 billion and the result of this bug was the loss of $370 million.The reason behind its failure was an integer overflow, which is a widespread bug in computer programming. In this case, an attempt was made to set a 64-bit number in 16-bit space.
- PayPal accidentally credits man $92 quadrillion - When Chris Reynolds opened his PayPal e-mail statement, the Pennsylvania PR executive's account balance was $92,233,720,368,547,800.The amount is significant in the world of 64-bit numbers suggesting a programming error. The range for 64-bit numbers is -2^63 to (2^63) i.e. -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807, but it seems to have been rounded up. The error was quickly recognized, and his account had returned to zero by the time he had logged in.
- Canada's Therac-25 radiation therapy machine malfunctioned and delivered lethal radiation doses to patients - Because of a subtle bug called a race condition, a technician could accidentally configure Therac-25 so the electron beam would fire in high-power mode without the proper patient shielding.

Not making mistakes is impossible, but luckily not all mistakes are as tragic or expensive.

There are multiple reasons why software testing is important and here are some of the benefits:

- **Product quality:** It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.
- **Cost-Effective:** Testing any IT project on time helps you to save money in the long run because paying developers to fix issues when the product is already mature and complex is pretty expensive. So if the bugs are caught in the earlier stages of testing, the costs are generally low.
- **Security:** It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
- **Customer Satisfaction:** The main aim of any product is to give satisfaction to their customers. **UI/UX Testing** ensures the best user experience.

How?

Generally software testing is classified into three categories:

- Functional Testing - Unit Testing, Integration Testing ,Smoke (done manually to check the build and that the main features still work), UAT (User Acceptance Testing), etc.
- Non-Functional Testing or Performance Testing - Performance, Endurance, Load, Volume, Scalability, etc.
- Maintenance (Regression and Maintenance) - Regression, Maintenance, etc.

There are currently around 150 types of testing, but note that not all types are suitable for all projects, we need to analyze the nature and scope of the project.

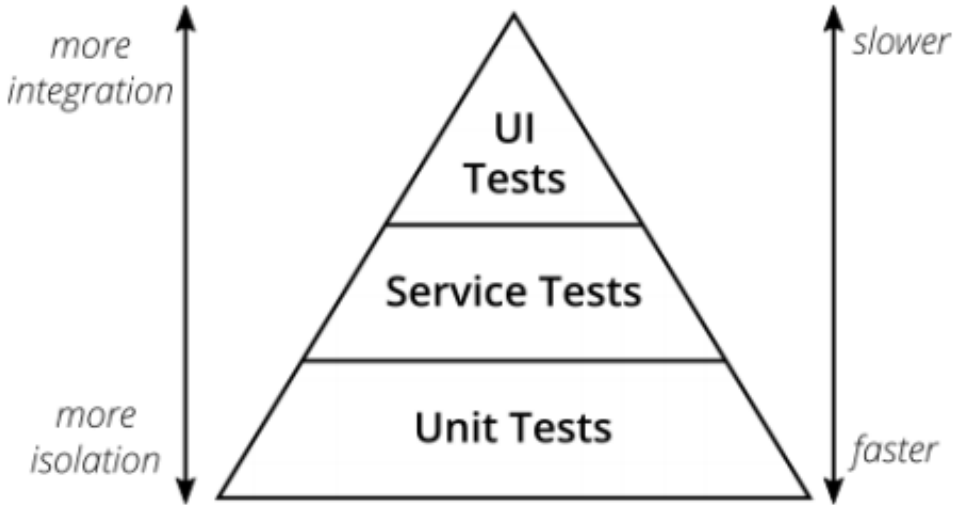
Here are some important strategies in software testing automation:

Unit Testing: This is a basic piece of test done by the programmer to test a unit of the program. It helps developers know whether the individual piece of code is working properly or not. This is applied to the business logic code.

Integration Testing/ Service Tests: It focuses on the integrated units that are working together without errors.

End to End/ UI Tests: With this method, your software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.

This pyramid represents the healthy proportion of testing you should be having in your project. This means that automated tests against the UI should mainly focus on the most important features, otherwise they can be a source of noise without adding any meaningful value.



So a good idea in learning how. This will give you a better understanding of the application itself, the business requirements and the logic. It does not involve *changing the existing code*, you'll have to write your own, which will automate the verification of the main/ most important workflows. These tests will ensure that once you start adding new functionalities to your application, the old ones will still behave the way they should.

Once you understand what is expected from the SUT (system under test) from a user's perspective, a potential involvement in developing/ maintaining the application will feel even easier, as the code will be quicker to understand.

The end goal of understanding and working with software automated testing is to have your tests run every time a new version of the application is deployed, so that you can ensure that new changes don't interfere with existing functionalities.

To cover the basic prerequisites on following this path, here are some learning essentials :

- HTML fundamentals - https://app.pluralsight.com/library/courses/html-fundamentals/table-of-contents

If you're already familiar with these topics, just watch the videos and explanations for a quick reminder

- Maven fundamentals - https://app.pluralsight.com/library/courses/maven-fundamentals/table-of-contents

Make sure to follow along with the examples and take your time to understand the explanations, this knowledge foundation will be used in all the future projects, at least in our company. Tip: if there's something that's not entirely clear you can also google that term and try to find the right explanation for you.

- Automated Web Testing with Selenium and WebDriver Using Java - https://app.pluralsight.com/library/courses/automated-web-testing-selenium-webdriver-java/table-of-contents

Again, follow along with the examples and take your time.It is important that you understand these concepts, but nobody is expecting you to become an expert overnight.