Unit 2 Seminar

Title: Requirements Gathering

Requirement gathering is crucial for collecting and documenting information to understand the client's needs and define project objectives. It also guides product or service development (Sire, 2023). Smartbear (2023) states that Gherkin is a simple, plain-text language that non-programmers can easily understand. It allows concise descriptions of test scenarios and examples to illustrate business rules in real-world domains. Furthermore, Nicieja (2017) explains that Gherkin is a language used for writing requirements in Specification by Example (SBE), which helps bridge the gap between stakeholders and technical teams. It provides a clear and structured way to express requirements.

Gherkin is a descriptive language used for writing requirements under Specification by Example (SBE), as explained by Carballo (2022). It is part of the Domain Specific Language (DSL) family and is easily understood by all roles involved in the software development process. The basic syntax of a file written in Gherkin language consists of a feature, business definitions, assumptions, background, scenario, scenario outline, and **When**.

Gherkin uses the **Given**, **When**, and **Then** keywords to establish preconditions for executing requirements. The Given keyword initialises data, creates instances, and sets the database state. The **When** keyword describes the user's essential action on the system, which triggers the test. The **When** command describes interaction with a web page, another user interface element, an API, or a library function, as stated by Carballo (2022). Moreover, the **And** and **But** words allow multiple **Givens**, **When's**, and **Then** statements within the same test scenario. The **And** word replaces the repetition of **Givens**, **When’s**, and **Then’s**, while the **But** word describes an adverse action or result, making the code more understandable to readers.

Collecting project requirements can be time-consuming and costly. To avoid communication failure, use the Gherkin language to simplify the process. Gherkin is a natural language that helps business analysts and developers create scenarios for functional tests. Its specific syntax allows for easy understanding and discussion of requirements. The requirements collected in Gherkin are saved to a feature file, which can be used for automated tests in Behaviour-driven development using Cucumber (Nowacki, 2016).

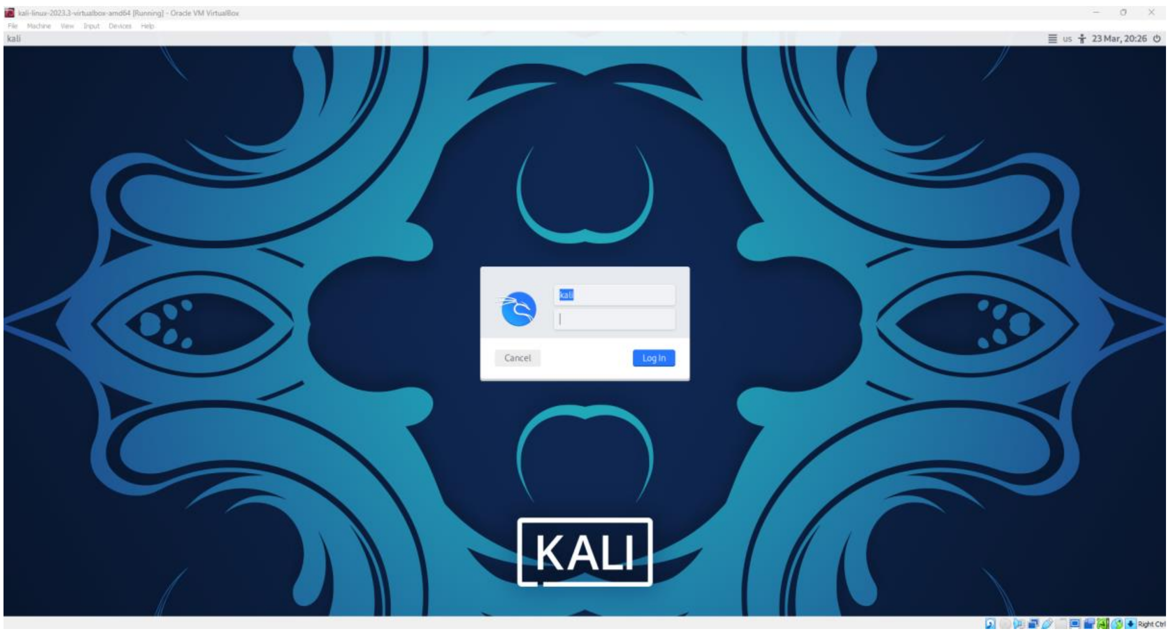
Fitzgibbons (N.D.) emphasises that Behaviour-driven development (BDD) is not just a methodology but a collaborative approach that is a crucial aspect of Agile software development. It uniquely combines the benefits of Test-Driven Development (TDD) and acceptance testing, fostering collaboration among developers, QA, and non-technical or business participants in a software project. BDD focuses on understanding software behaviour through stakeholder discussion, making it a powerful tool for aligning business and technical perspectives. BDD practices involve establishing goals, drawing out features, involving stakeholders, using examples to describe application behaviour, automating examples for quick feedback, clarifying responsibility using "should" and "ensure", using mocks, and implementing UI first (Readthedocs, 2020).

Akhtar (2023) underscores BDD's significant growth trajectory, positioning it as a game-changer in the software industry. This projection highlights BDD's increasing popularity and potential to shape the future of software development, making it a compelling area for professional growth and development.

BDD is a process that involves close collaboration between business, development, and QA teams. The teams work together to create a script, written in the Gherkin language, that precisely describes the software's behaviour. This script is then transformed into automated tests that verify whether the software behaves as expected. The advantages of BDD are numerous and significant, including improved collaboration, enhanced requirement understanding, early defect detection, and increased test coverage. BDD's clear and structured process enhances the software's quality and fosters a culture of collaboration and shared understanding among the teams involved.

Behavior refers to how an individual interacts with the environment and is about the action-reaction process. BDD is an Agile software development process that combines an engineering component with a business goal, is test-centred and focuses on the product's expected behaviours. Scenarios are the cornerstone of BDD, which can be run manually or automatically (Sheremeta, 2022).

Below is the Gherkin sequence that addresses using a computer running the Linux operating system based on the Readthedocs (2020) ideas:


kali-linux-2023.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali
us 23 Mar, 20:26
kali
Cancel
Log In
KALI
Right Ctrl

```
Feature: Using a virtual instance running Linux Kali OS

Scenario: Logging into the Linux System
  Given I am at the login screen.
    When I enter my username and password
    then I should be logged into the Linux system.,

Scenario: Navigating the File System
  Given I am logged into the Linux system.
    When I open the terminal emulator
      and I type "ls" to list files in the current directory
    then I should see a list of files and directories.,

Scenario: Editing Configuration Files
  Given I am logged into the Linux system
    When I open a text editor using the command vim
      and I edit a configuration file (vim /etc/ssh/sshd_config)
    then I should save the changes and exit the text editor.
```

In BDD, stories are the fundamental unit of functionality, and acceptance criteria are an intrinsic part. They define the scope of the behaviour and provide a shared definition of "done." Conversations between stakeholders, analysts, testers, and developers create these stories. BDD is as much about the interactions between the various people in the project as it is about the development process's outputs (Dannorth, 2007).

References:

- Sire, T. (2023). How to Gather Requirements as a Business Analyst (Guide). [online] Available from: https://www.requiment.com/how-to-gather-requirements-as-a-business-analyst/ [Accessed 24 Mar. 2024].
- Smartbear (2023). Writing scenarios with Gherkin syntax | CucumberStudio Documentation. [online] Available from: https://support.smartbear.com/cucumberstudio/docs/bdd/write-gherkin-scenarios.html#:~:text=To%20create%20Gherkin%20test%20scenarios [Accessed 24 Mar. 2024].
- Nicieja, K. (2017*). Chapter 1. Introduction to specification by example and Gherkin · Writing Great Specifications: Using Specification by Example and Gherkin.* [online] Available from: https://livebook.manning.com/book/writing-great-specifications/chapter-1/52 [Accessed 25 Mar. 2024].
- Carballo, P. (2022). Writing Specification by Example Requirements with Gherkin. Available from: https://www.octobot.io/blog/gherkin-specification-by-example/ [Accessed 25 Mar. 2024].
- Nowacki, Ł. (2016). *Speak Gherkin and learn how to collect requirements for your project.* Available from: https://neoteric.eu/blog/speak-gherkin-and-learn-how-to-collect-requirements-for-your-project/ [Accessed 25 Mar. 2024].
- Fitzgibbons, L (N.D.). *What is behavior-driven development (BDD)? Definition from SearchSoftwareQuality*. [online] Available from: https://www.techtarget.com/searchsoftwarequality/definition/Behavior-driven-development-BDD [Accessed 25 Mar. 2024].
- Readthedocs (2020). *Behavior Driven Development — behave 1.2.6 documentation*. [online] Available from: https://behave.readthedocs.io/en/stable/philosophy.html#the-gherkin-language [Accessed 25 Mar. 2024].
- Akhtar, H. (2023). *What is BDD? (Behavior-Driven Development)*. Available from: https://www.browserstack.com/guide/what-is-bdd [Accessed 25 Mar. 2024].
- Sheremeta, O. (2022). *BDD & Agile Methodologies in QA*. Available from: https://testomat.io/blog/behavior-driven-development-agile-methodologies-in-quality-assurance/ [Accessed 18 Mar. 2024].
- Dannorth (2007). *What's in a Story?* [online] Available from: https://dannorth.net/whats-in-a-story/ [Accessed 25 Mar. 2024].