

Group assignment - Testing

Why

Maintaining quality in software engineering is of paramount importance. In order to manage quality levels, we need methods of checking that the materials (or artefacts) that we produce are sufficiently good.

In order to measure quality, we need to have ways to test what we do - something which is covered in more detail in the Software Quality and Testing module. This assignment is designed to get you to consider the ways you can drive quality standards up.

Quality is not only seen in the outputs of a project, however. In the talk by Matt Bowers, from Sony, we heard how using techniques such as Kanban help the team focus on producing a quality product, through helping to maintain the quality of the process as well. Simple tools to help you keep focus on what is important on a project can make a significant difference to the quality of the outputs.

Using collaborative tools, such as CSGitLab also provide a stable platform for working together - keeping the work that is being done in one place, with the added benefit of providing a mechanism for discussion and alleviating the need to "get together" - facilitating remote working and providing an audit trail so you can reflect on what went well during a project.

Key skills for employment - according to research done at Google, as well as in academia more generally - include the soft skills. See an article on Project Aristotle for more information.

Sometimes we also have to work to short deadlines. Part of this assessment looks at how well you can organise yourselves in your team, get to work, and produce results. In many cases, a long deadline appears to leave people procrastinating, so this is an opportunity to see how well you work together with a short deadline to get something done.

How

Using CSGitLab as your development tool, your group will design a basic test plan, and some tests to support the development of a game. The game is based on the same idea as in the first coursework - but do read the specification carefully.

You will produce your work in markdown format (not, e.g. in Word) and you can either do this online on CSGitLab or on your local machine and use git to sync it with the repository on CSGitLab.

You will use Issues on CSGitLab to communicate with one another, alongside creating a group Trello Kanban. You will include at least 2 screenshots of your Trello board in your final submission, to demonstrate its use.

The tests you develop will be written in the Gherkin language.

What

Produce a markdown document which gives a brief quality plan for the testing of a game software development project. This plan should cover 5 key areas (hint: use these as section headings):

- Project introduction (describe briefly what the project aims to achieve)
- Product plans (a brief explanation of the order in which elements will be developed, especially where this might impact the test design and implementation)
- Process descriptions (what processes will be used for testing - who (which role) is responsible, and how is quality measured?)
- Quality goals (what are you aiming for? Specific, measurable, achievable, realistic, timely goals to be met)
- What risks, especially to the quality of the project, are there, and how can you manage them?)

In the Process descriptions, you will include your Gherkin code for three (3) specific tests (you do not need to worry about how these tests will be implemented in executable code).

1. Tests for a valid/invalid login attempts
2. Tests for adding a new high score entry
3. Tests for sanitising user entered text

You can assume that login IDs will be held in a database, along with hashed passwords, and accessed through an API call `attemptLogin(username, hashedpassword)`

You can assume that scores are stored in a database table, with username and score as elements of the record, and can be read using `getHighscores()` and new entries can be added using `addHighscore(username, score)`

You can use the information on the big list of naughty strings to help identify suitable tests - a few examples will be adequate, do not try and write tests for all of these unless you are going to parameterize it!

Your submission should be quite short - if quality plans are too long, no-one will read them. Consider a 250 word limit on the introduction, plans, goals and risks (consider using a table to present risks). The processes section may be up to about 1000 words.

You must submit the link to the file in your repository and the SHA of the last commit on Blackboard (only one submission needed per team)

The game

The game will be web-based (played in a browser). University students will be expected to log in, play the game and solve puzzles in a graphical adventure, earning points for their solutions. The solution to the puzzles will require them to have read and understood various historical sources. Their scores will be stored in a high score table. Solutions to puzzles will include short text entries, and they will be able to use an in-line chat/forum facility to collaborate with other students on their course. The system will run on a server, with a separate web-server used to deliver the pages across

the internet to whatever devices the students choose to use. The material served will all be in HTML format.

Marking

- If you do not contribute significantly, using your own login ID, on CSGitLab you will receive a mark of 0.
- The Trello snapshots are worth 10% of the mark for the assessment, if they are correctly embedded in the final document, and demonstrate that you have used the Kanban to help keep track of the work.
- The Project Introduction is worth 5%
- The Product Plans are worth 5%
- The Process Descriptions are worth 10%
- The example tests are worth 15% each (and each should consist of at least 3 examples in Gherkin)
- The Quality goals are worth 10%
- The Risks and risk management are worth 10%
- 5% of the marks will be for the overall quality of the work (well formatted, referenced, etc.)

Remember this piece of work is worth 25% of the module. It is reasonable to expect it to take at least 15 hours - you have the resources, make sure you ask any questions to clarify anything you don't understand as soon as possible.

There are groups set up for you already on CSGitLb (using the UK phonetic call sign alphabet), and on Blackboard. You can - and should - use CSGitLab as the main communication tool, but can also email each other using the groups on Blackboard.