**Portfolio**

**A How-To Guide**

**Here is an example how the title page could look like:**

**CS2S567 Professional Practice and Employability:**

**Team Based Software Development Workshop**

**Portfolio**

Detailing own role in the team effort

**By**

*Your Name here, Enrolment number here*

|  |  |
| --- | --- |
| Team Number | **1** |
| Team Task | Setup and.... |
| Own role(s) and tasks in Team | Role:   * big cheese (main bully) * agony aunt   Tasks:   * make coffee * buy pizzas * string some code together * not get in the way of the clever guys * ... |

**Here is an example of a table of content. Note that it was created automatically. Look it up if you don't know how to do this - a real time saver!**

**Table of Content**

[QA Activity Overview Log 2](#_Toc460948477)

[Weeks 1 & 2 3](#_Toc460948478)

[Weeks 2 & 3 4](#_Toc460948479)

[Appendix: Additional Evidence 5](#_Toc460948480)

**Frequently Asked Question: What is Quality Assurance (QA)?**

'Portofoilio' comprises all activities that are necessary in order to produce a product of the highest possible standard or as Wikipedia[[1]](#footnote-1) defines it:

"*...is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers; which defines as "part of* [*quality management*](https://en.wikipedia.org/wiki/Quality_management) *focused on providing confidence that quality requirements will be fulfilled*"

For the production of software this means:

Preparation work

* making sure that the program requirements are clear and well defined (PRD)
* making sure that project staff have all the required knowledge in terms of standards, processes, policies, procedures, programming know-how,
* Agreeing on backup procedures

So, in this portfolio you need to show that you have considered this, researched the 'how-to' and have taken action to fill any gaps in your knowledge, etc.

Programming work

* There are programming techniques such as 'Extreme programming' which address activities such as good communications, pair programming and continuous testing.
* Making backups (and verifying they are indeed ok)
* Verification and validation activities[[2]](#footnote-2) such as unit testing, static testing, dynamic testing, regression testing, integration testing, etc. And all of those in black box and/or white box mode. Not all of these apply of course, it depends.

So in this portfolio you need to show that you have considered this (e.g. do background research on which tests to do and why) and taken action (e.g. done testing and reacted to the outcome).

Rolling it out work

* How do you securely deploy your software
* How do you plan the software maintenance and updating
* What information does the user need (manuals, training, preparation for change-over to the software,...)

So in this portfolio you need to show that you have considered this (e.g. read-up on it) and taken action (e.g. contact users, produce a training manual).

**How to report**

Use the table structure below. Fortnightly reports, i.e. one table every 2 weeks are sufficient. The blue 'Team' items can and should be completed jointly, the green 'Own' column you obviously do on your own.

Portfolio writing start from week 3 and in a normal division of team with 4 members it will finish at week 22.

Each team member should report on 5 consecutive weeks, i.e. 3 of the tables below for each individual. Since a typical team is usually made up of 4 members the team could divide the reporting up like this: Team member 1: weeks 3 to 7, Team member 2: weeks 8 to 12, Team member 3: weeks 13 week to 17 week, Team member 4:- weeks 18 to 22, (Team with fewer members just have a few unreported weeks. Team with more members can accommodate the division including week 23, week 24)

|  |  |  |
| --- | --- | --- |
| Weeks 1 & 2 | **Project Setup** | |
|  | **Team** | **Own** |
| **Activities undertaken** | E.g.: which activities were undertaken, coding this and that, organising, preparing, and running a test, background research, setting up a unit test, creating a QA log,...seminar picture /presentation quality |  |
| **Outcome of QA activities** | E.g.: what was the result, any consequences for further development, any further tests planned? Was any document created, a test protocol developed, ... |  |
| **Activities resulting from QA process** | E.g.: entire method had to be re-coded, all code line using method x had to be changed, ... |  |
| **Personal QA development** | E.g.: researched into testing method x and applied it to code segment y, figured out how to write unit tests in python, researched how to write a test protocol, ... | |
| **Evidence of own activities** | E.g. Any communications and logs, hyperlinks to work, test documents, code snippets before vs. after, a summary or 'how-to' document on Unit Testing in python, ...  Place larger items such as longer code sequences, big screen shots, photographs, etc, that don't fit well into this table into the **Appendix** or provide links. Reference these items here, e.g.:   * *Unit test code for login class: See Appendic item 12* * *Link to wiki GIT repository with test program 5: https://....* | |

# 

# Use an overview sheet for providing a summary of ALL QA and issue Activities you undertook

The detailed tables on the previous page cover 5 weeks. The overview sheet below covers the overview of 12 issues of quality/ISSUE TRACKING in whole year of 24 weeks.

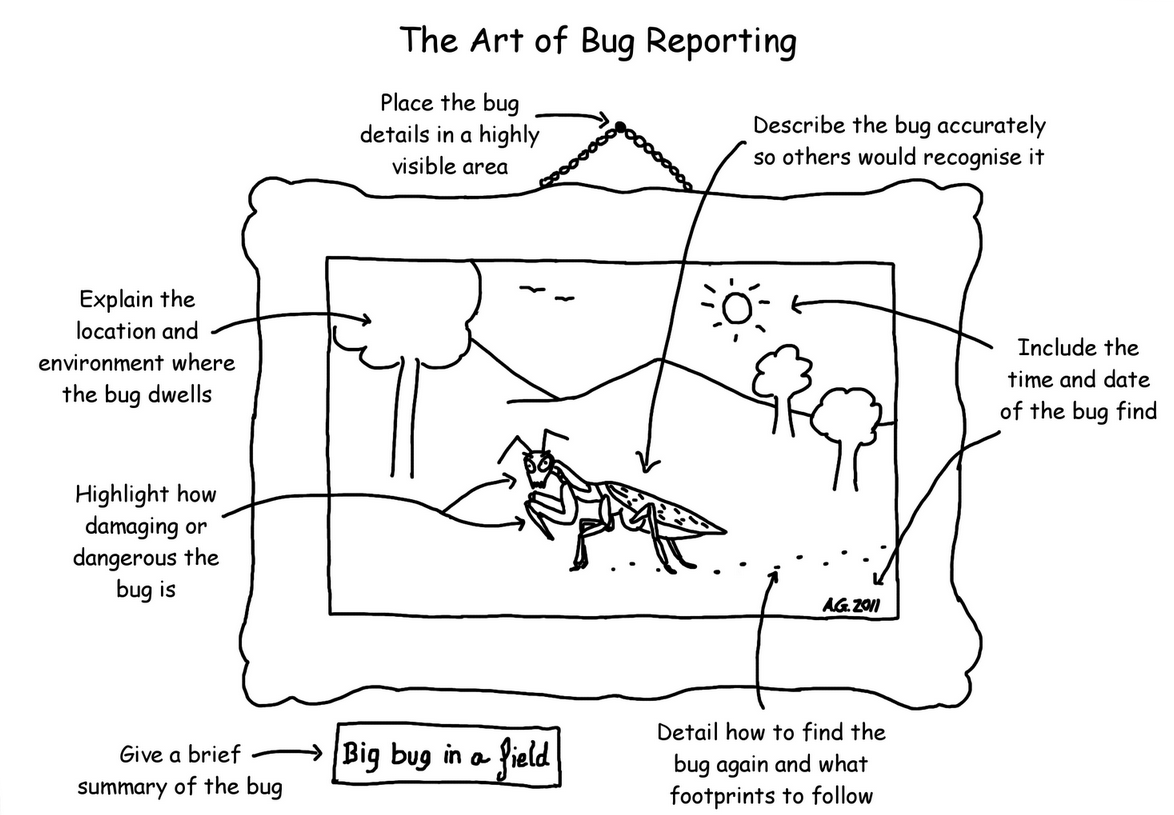
Provide evidence for your activities and/or outcomes in the Appendix (see next page)

# Personal QA/ISSUE TRACKING Activity Overview Log

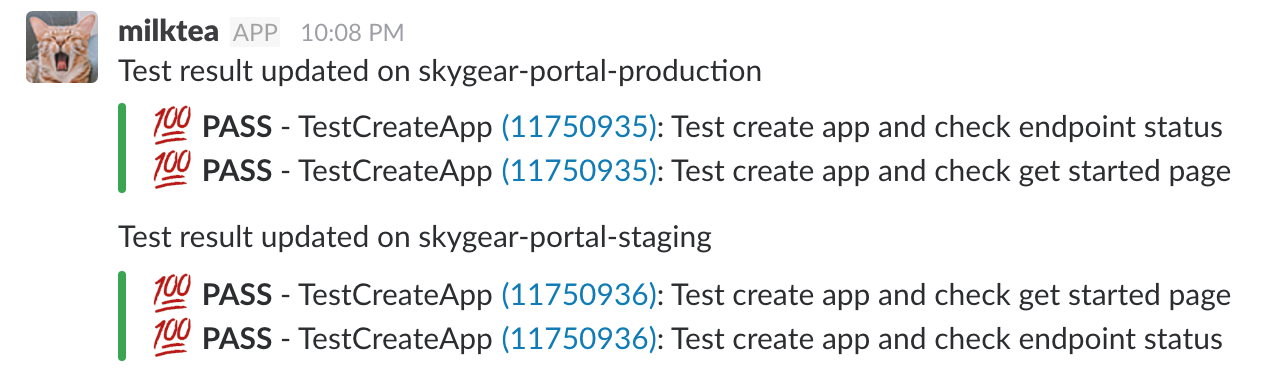
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Issue** | **Week**  **date** | **Activities undertaken** | **Outcome of QA Activities** | **Activities resulting from QA Process** | **Personal QA development** |
| *1* | *2* | *What did you do?*  *Also: reference to* ***Evidence of you actions****(Appendix or www hyperlink)* | *What happened as a result of the action? (e.g. error corrected, referred to team mate .,..)*  *Also: reference to* ***Evidence of the outcome*** *(Appendix or www hyperlink)* | *What actions or consequences happened as a result of the Activities and outcomes to the left?* | *E.g:. new knowledge you have acquired, things that you would do differently next time* |
| 2 | 2 |  |  |  |  |
| 3 | 2 |  |  |  |  |
| 4 | 3 |  |  |  |  |
| 5 | 3 |  |  |  |  |
| 6 | 3 |  |  |  |  |
| . | . |  |  |  |  |
| . | . |  |  |  |  |
| . | . |  |  |  |  |
| . | . |  |  |  |  |
| . | . |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Any larger pieces of information that don't conveniently fit into the tables is placed in the Appendix.

# Appendix: Additional Evidence

****

**Item 1:** Methodology applied for bug reporting to team, Andy Glover ©, Blogspot.com

**Item 2:** Example of GitLab CI testing report for the 'start and stop' modules

A portfolio structure based on 4 team members per team .Quality assurance will start from 3rd week.

**Table of Content**

[Issue Activity Overview Log 2](#_Toc460945716)

[Weeks 1 & 2 3](#_Toc460945717)

[Weeks 3 & 4 4](#_Toc460945718)

[Weeks 5 & 6 5](#_Toc460945719)

[Appendix: Additional Evidence 6](#_Toc460945720)

**Frequently Asked Question: What is an 'issue'?**

An 'issue' is everything that is out of the normal and 'smooth running' range of events. Example of issues are:

Personal issues

* problems with one or more people in the team on a personal level
* as above but with other teams, the boss, the client or any other people
* private life getting in the way (consider your privacy here and describe these only very generally, e.g. not: "I have been declared bankrupt" but "financial problems". Not "I had syphilis" but "health issues"
* forgot to attend a meeting, screwed-up, made a mistake,...

Organisational issues

* tricky to get the team together for a meeting
* hard to agree on a way forward
* unable to get resources (knowledge, code snippets, help, ...) in time or not at all
* not enough time to finish this subroutine

Team issues

* certain teams member to pulling their weight
* insufficient knowledge inside team
* low and/or infrequent meeting attendance
* bad communications inside team or with other teams, the boss, ...

Technical issues

* software bugs that need to be fixed
* it just won't bl@@dy work!
* can't figure out how to ....

Ethical and/or legal issues

* is it OK to use this image I found on the internet in our program?
* should I use my time to help other team members rather than doing my own work? (answer: yes. Document it, you will get points for it - but don't forget )
* this part of the program I am asked to write spies on people / collects too many data. What now?

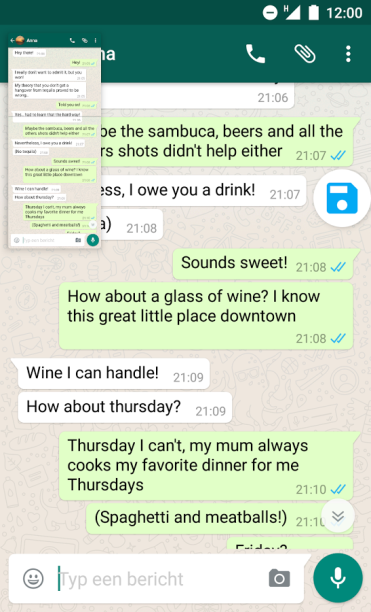
Professional issues

* I can't do this. No idea how. What training do I need? How shall I tell the boss?
* our communications with other teams is complicated. What could be done to improve matters?

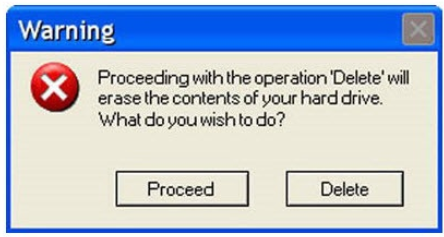
Note that the above list is by no means complete and the issues shown are just an example to give you an idea. Some issues can also span several categories.

**Finally, on legal, social, ethical and professional issues you gain points when you can evidence background research into the respective issues.**

  
**Item 1:** Photographs of author's cat eating the only printout of the team's server login details and result of the team's actions to avoid the issue in the future



**Item 2:** Screen shot of the author (green boxed text) arranging the next team meeting

**Item 3:** The error messages created by the author to address the hard drive issues encountered

A portfolio structure based on 4 team members per team .issue tracking will start from 1st week.

1. https://en.wikipedia.org/wiki/Quality\_assurance [↑](#footnote-ref-1)
2. Verification = Are we building the product right? Validation=Are we building the right product? [↑](#footnote-ref-2)