Research Findings: Quality Assurance Artefacts

Date 4 August 2015 Researcher Joshua Son

1. Background

Quality assurance is a fundamental aspect of delivering a successful product. By understanding quality assurance practices, we can be sure that we are developing a product that meets our client's requirements.

2. Objectives

To research templates and common quality assurance practices in order to produce artefacts that will aid us in developing a successful product to meet our client's needs.

3. Approach

General online research.

4. Findings

- 4.1. Testing Strategies
 - 4.1.1. Unit Tests
 - 1. XP uses a test driven development where tests are created before the code. Unit tests test units of work where one unit may be an interface, class or method.
 - "Creating tests independent of code, hopefully before code, sets up checks and balances and greatly improves the chances of getting it right the first time." (Wells, 1999)
 - 3. Benefit: allows developers to find bugs earlier on, whether in the code itself or the test.
 - 4.1.2. Acceptance Tests
 - 1. These tests are created from user stories.
 - 2. One or more tests may be generated from a user story.
 - 3. User stories are not complete until they've passed their corresponding acceptance test. (Wells, 1999)
 - 4. Benefit: encourages closer collaboration between developers and users and thus being able to produce a more tailored product. (Agile Alliance and Institut Agile, 2013)
- 4.2. Flow Charts
 - 4.2.1. Activity diagram
 - 1. It is classed as part of the UML behaviour diagrams and emphasises the dynamic behaviour of the objects in a system.
 - 2. Used to show the flow of control, or 'object flow', with emphasis on the sequence and conditions of the flow. (Fakhroutdinov, 2009)
 - 3. They usually contain the following nodes and edges: activity, partition, action, object, control, activity edge.



- 4. Benefit: to diagrammatically represent the behaviour of the system and develop a more intuitive idea of the system.
- 5. An example of an activity diagram from uml-diagrams.org.

4.3. Data Models

- 4.3.1. Entity/class diagrams
 - "UML structure diagram which shows structure of the designed system at the level of classes and interfaces, shows their features, constraints and relationships - associations, generalizations, dependencies, etc." (Fakhroutdinov, 2009)



- 2. Classes are objects that share the same features, constraints and semantics.
- 3. The arrows connecting the boxes show the different relationships and how classes interact with one another. These relationships can be categorised into associations (arity, aggregation, composition, etc.), generalisations (generalisation, specialisation), dependencies (use, create, call, etc.).
- 4. Benefit: to provide a plan of, and visually represent, the structure of the system whilst minimising the outcome of a 'god class'.

5. Further Investigation

5.1. None

6. Recommendations

- 6.1. To ensure quality is built into our product, we should create unit tests before coding.
- 6.2. After the creation of user stories, create acceptance tests to determine whether the user story is complete.
- 6.3. During the planning phase of development, create both activity diagrams and class diagrams to better understand the overall system.

7. References

- Agile Alliance and Institut Agile. (2013). Guide to Agile Practices. Retrieved August 4, 2015, from http://guide.agilealliance.org/guide/acceptance.html
- Fakhroutdinov, K. (2009). Activity Diagrams. Retrieved August 5, 2015, from www.umldiagrams.org/activity-diagrams.html
- Fakhroutdinov, K. (2009). Class Diagrams. Retrieved August 5, 2015, from www.umldiagrams.org/class-diagrams.html
- Wells, D. (1999). Unit Tests. Retrieved August 4, 2015, from http:// www.extremeprogramming.org/rules/unittests.html
- Wells, D. (1999). Acceptance Tests. Retrieved August 4, 2015, from http:// www.extremeprogramming.org/rules/functionaltests.html

Images:

Fig 1: http://www.uml-diagrams.org/online-shopping-uml-activity-diagram-example.html? context=activity-examples

Fig 2: http://www.uml-diagrams.org/class-diagrams-overview.html