Research Findings: Quality Assurance

Date 8 August 2015
Researcher Paul Lee

---

1. Background

For research and development of any project, Quality Assurance techniques must be in place to ensure that the work done is of high quality.

---

2. Objectives

Understand how and what we can do to ensure that quality will be maintained throughout our project.

---

3. Approach

Look at online resources, ask the supervisor, read journal articles

---

4. Findings

4.1. Extreme Programming is an agile technique that moves quality assurance from the back end of a project to the front. It features two levels of testing, Unit Testing and Functional Testing. Programmers write/develop code to tests that have been written by the customers. Automated acceptances test scripts can be run at any time, by anyone of the team, including the customers. Customers are responsible for articulating what they want and providing feedback on the results of the tests/early version releases they see.

4.1.1. Unit testing is performed by the programmers as they work. Each class that is implemented must have unit tests already developed, testing for everything that "could possibly break". The tests are run as frequently as possible during development and all unit tests in the entire system must be running at 100% before any developer releases their code.

4.1.2. The second level of testing is called functional testing. Each feature of the system, identified as a User story, must have one or more functional tests that test it. These tests are defined by the customer.

4.2. XP releases are a "Release Often" approach and the highest-priority functionality is released first, smaller chunks of code are easier to test and because testing is automated it is quick to test them.

4.2.1. Users are able to get the most important functionality more quickly, with quality assurance.

4.3. Plan, Do, Check, Act, an iterative four-step management method for control and continuous improvement of processes and products.

4.3.1. Plan – Establish the objective and processes necessary to deliver the results in accordance with the expected output. The expectations will improve the completeness and accuracy of the specifications.

4.3.2. Do – Implement the plan, execute the process and make the product. Collect data for charting and analysis.

4.3.3. Check – Study the results and compare against the expected results to ascertain any differences. Look for deviation from the plan and look for the appropriateness and completeness of the plan.

4.3.4. Act – If the CHECK shows that the PLAN was implemented in DO is an improvement to the baseline, then that becomes the new baseline for how to ACT going forward.

---

5. Further Investigation

5.1. Could look in the control frameworks and how they work, such as ITIL and COBIT. These are mainly for larger organisations though.

5.2. Take a look at a few ISO standards, a lot of these are about how to manage projects and quality.

5.3. Try and ask big companies how they ensure quality in their business. Find out their business practices.

---

6. Recommendations

6.1. We have templates for all documentation

6.2. We will be using camel-casing for all code

6.3. All code will be done using the pair-programming approach

6.4. All documents and code will be double checked by another member of the team

---

7. References

C2.com,. (2015). *Extreme Programming Quality Assurance*. Retrieved 8 August 2015, from http://c2.com/cgi/wiki?ExtremeProgrammingQualityAssurance

Ronjeffries.com,. (2015). *Quality Assurance*. Retrieved 8 August 2015, from http://ronjeffries.com/xprog/articles/qa/xp_q_and_a_qa/

Wikipedia,. (2015). *PDCA*. Retrieved 8 August 2015, from https://en.wikipedia.org/wiki/PDCA