RISK BASED TESTING

HENRIETTE HARMSE

A question that is asked regularly in testing circles is: "When should you stop testing?". With an infinite budget it may be reasonably easy to answer this question. When the budget is severely constrained, this question becomes more difficult to answer. An even more difficult situation to address is when the project starts out with a given budget, but during the project lifetime the budget gets significantly reduced (i.e. due to economic downturn). This forces us be able to provide the highest level of quality for the least amount of money.

1. How Testing Fails

A mistake that is often made is that the testing effort is distributed equally across the system – both critical and less critical portions of the system are tested equally. This results in critical parts of the system not being tested sufficiently and less critical parts being tested to the point of diminishing returns.

A further mistaken mindset of developers is that there is no such thing as a useless test, which may entice developers to add tests for the sake of adding tests. In actual fact every test (unit-, integration- or systems test) has to earn its place in the codebase. If there no good motivation for a test, the test must be deleted from the codebase. Why is that? It adds to volume of code that developers have to master to be productive members of the team. It adds to the volume of code that has to be maintained. As such tests adds to the overall cost of maintenance of a system. **The most cost effective code to maintain is the code that has never been written**.

2. Risk Based Testing

Risk based testing is a testing approach that is helpful in addressing these concerns. The main advantage of risk based testing is to enable prioritization of the testing effort. The heart of risk based testing is to provide a set of criteria for evaluating risk. Some criteria for determining risk can be found in [1]. The criteria may differ depending on the project. Ideally the set of criteria needs to be negotiated with the project owners. Another important aspect is to decide on the level at which risk will be determined. For example, risk may be determined at business process, component or use case levels.

Some general criteria are provided below:

- How frequently is this use case used?
- What will be the cost of getting this use case wrong?
- What is the complexity of this use case?
- How often will the use case be changed?

Date: 19th February 2018.

A value from 1 to 5 is assigned for each of these criteria and the product determined. This product provides a way for ranking the risk of each use case. A higher risk value will indicate a use case with a higher risk.

Ideally determining risk should not be left to developers. Developers need to be guided by the project owner, the architect and the business analyst on their project.

3. Advantages/Disadvantages of Risk Based Testing

Advantages of this approach are:

- The highest risk items can be developed and tested first which reduces the overall risk on the project.
- If testing has to be watered down, a guideline exists for deciding what to test and what not to test.
- At any given time an indication can be given of the risk of the project by considering the highest risk use cases that has not been tested.

A possible disadvantage of risk based testing is that not everything will be tested, but as stated earlier, if a testcase cannot be motivated, it may be more cost effective to not have the test as part of the codebase.

4. Conclusion

Risk based testing provides a means via which one can estimate how well, or how poorly, a system has been tested.

References

1. Kshirasagar Naik and Priyadarshi Tripathy, *Software Testing and Quality Assurance: Theory and Practice*, John Wiley and Sons, 2008.