Ponder 01: Quality Metrics

Ryan Dockstader

# Acceptance

## Description

Acceptance is generally referred to as how quickly a user accepts and begins using a piece of software. In the agile cycle, the phase that covers this is known as Acceptance Testing. The nice thing about this type of metric, is that it’s either pass or fail. Either the users accept it and then the team moves onto the next step in the development life cycle, or it fails and it’s back to rework with feedback. [1]

## Measurement

To me acceptance measuring is straightforward. You start by gathering a group of users, at least one from each group of users that are stakeholders in the software. Then you have those users use the system for x amount of time. Then, once they have completed the trial you have a set question or set of questions that they are asked, and user acceptance is measure from those yes/no questions.

# Affordability

## Description

The concept of Affordability is the concept of having software the meets the needs of the person that needs it, within the constraints of their budget. This isn’t limited to the upfront cost of developing the software but is also dealing with the cost to maintain software. On top of that (especially in this modern time) it also needs to consider any hosting, encrypting, etc. [2]

## Measurement

Affordability can be a tricky one, since it’s going to be based mostly on the time of the developers up front, and then the cost of services once the product is live. Unfortunately, humans are consistently bad at two things: prediction how long it will take to accomplish something, and prediction the future of prices. That being said, I think requirements gathering can play a large role in this, as well as getting an up-front budget from the customer. Then you must spec each requirement and make an educated guess on time. For services, you must pick the stack that will be right for the customers budget long term.

# Attractiveness

## Description

This is quite an interesting aspect of software, that is often overlooked by developers more involved in the code base then the user interface. Attractiveness in software is one of the most important aspects to sell it, as it is “The capability of the software product to be attractive to the user.” [3] Without being able to attract users, like flies to a fly strip, the software will likely not get funded, and eventually fail.

## Measurement

Attractiveness, I think, is a hard one to measure. I think what you’d have to do for this one is setup a few separate interfaces that all accomplish the same task (they could be prototypes), and take a sample of the user base, and get feedback as to which prototype they are more drawn to.

# Flexibility

## Description

Flexibility in software is the ability of the software to respond in an appropriate matter to changes. These changes may be things as simple as the passing of time, to things as complex as updated hardware, operating systems, and standards of encryption. [4]

## Measurement

This one is straightforward for the now, I think. You basically just need to install it on whatever you can and make sure that it functions as it is intended in multiple environments, across multiple devices.

# Citations

1. Agile Alliance. (2019). *What is Acceptance Testing?*. [online] Available at: [https://www.agilealliance.org/glossary/acceptance/#q=~(infinite~false~filters~(postType~(~'page~'post~'aa\_book~'aa\_event\_session~'aa\_experience\_report~'aa\_glossary~'aa\_research\_paper~'aa\_video)~tags~(~'acceptance\*20test))~searchTerm~'~sort~false~sortDirection~'asc~page~1)](https://www.agilealliance.org/glossary/acceptance/#q=~(infinite~false~filters~(postType~(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'acceptance*20test))~searchTerm~'~sort~false~sortDirection~'asc~page~1)) [Accessed 20 Sep. 2019].
2. Sebokwiki.org. (2019). *Affordability - SEBoK*. [online] Available at: <https://www.sebokwiki.org/wiki/Affordability> [Accessed 20 Sep. 2019].
3. “Attractiveness,” *Attractiveness*. [Online]. Available: <https://www.isi.edu/natural-language/mteval/html/604.html>. [Accessed: 20-Sep-2019].
4. L. Shen and S. Ren, “Analysis and measurement of software flexibility based on flexible points ,” International Journal of Software Engineering & Applications, vol. 5, no. 4, pp. 331–341, 2006. <https://pdfs.semanticscholar.org/587c/48afa6644316fb744d43d27b19cba4afa34f.pdf> [Accessed: 20-Sep-2019].