Guiding Quality through Business Value

Most software is produced to give a return in terms of business value. The value of software quality is often difficult to quantify but contributes in no small part to the projects business value.

Defining quality

Software quality and testing teams often struggle to demonstrate the business value that they bring to projects. This is a source of frustration for both quality managers and other stakeholders who understand that quality is necessary but very difficult to have a conversation about.

The Oxford English dictionary defines quality as:

> *"The standard of something as measured against other things of a similar kind; the degree of excellence of something"*

Much confusion is derived from people's perceptions of what the "thing" is that we are trying to make of better quality. Even some seasoned test managers think it is just "the software" but the Product Owner in all likelihood views it as "customer satisfaction".

Customer satisfaction hinges on both Product and Project quality. If you want to excel in software quality, your "thing" needs to become "customer satisfaction".

Improving software quality management

The good news is that once you start focusing on overall quality rather than purely the product, you may find that you are investing too much, not too little! To do this though, you need to start doing the following three things:

1. Understand acceptable quality and risk
2. Invest your quality resources in the right areas at the right time
3. Communicate early so that better decisions can be made

Acceptable quality and business value

A major reason software quality is difficult to quantify is that the desired quality level differs between firms (and even internal teams) because of factors such as those below:

- Stakeholder expectation,
- Direct business penalties for failure which can include lost subscribers, brand damage, downtime and legal costs for bad analysis etc,
- The time required to correct an issue,
- And finally, the resources required to correct an issue

In all cases, if any of these factors is higher in a certain area of your software, you will want to invest more resource in understanding the quality level for that area.

It is apparent that these factors also help define business value, whether that is inherent value to the customer or potential costs to the producer. Quality and business value are very closely correlated.

In order to define acceptable quality, often teams will make a broad qualitative judgement. In some regulated industries attempts are made to enforce a minimum quality baseline through formal procedures and regulations.

However, to maximise your return, you need to quantify each area of your software in terms of the above factors so you can invest resource appropriately. This could be a complicated calculation involving the factors above, or something as simple as a Product Owner's educated guess may suffice.

Acceptable risk

Simplistically, software quality activities are inherently about ascertaining whether we are at the desired level of acceptable quality. However, given that testing and other quality activities are both expensive and infinite; whether we are at an acceptable level of quality should always be tempered with a probability score.

Managing quality really comes down to understanding acceptable quality and reducing the probability of loss; in other words, managing acceptable risk. Risk can be defined as:

Risk = Likelihood x Potential Loss

We can use the value for acceptable quality in place of potential loss. While not perfect, it will still enable you to compare the risks in the different areas of your project sensibly. This just leaves you to assess likelihood through your quality activities.

The business value of quality

Our analysis so far shows that the business value of quality processes in a company comes predominantly through reduced risk.

We have also shown that we can begin to build a quality value calculation engine around two values:

1. The value of the feature being implemented.
2. A probability rating

More value can be attributed to the efforts of your QA team for significantly reducing the likelihood of a negative event with a high value feature than a small reduction for a low value one.

However, improving quality, for example through more testing, comes at a significant price. So the next question becomes 'how do I maximise the value of quality in my project?'

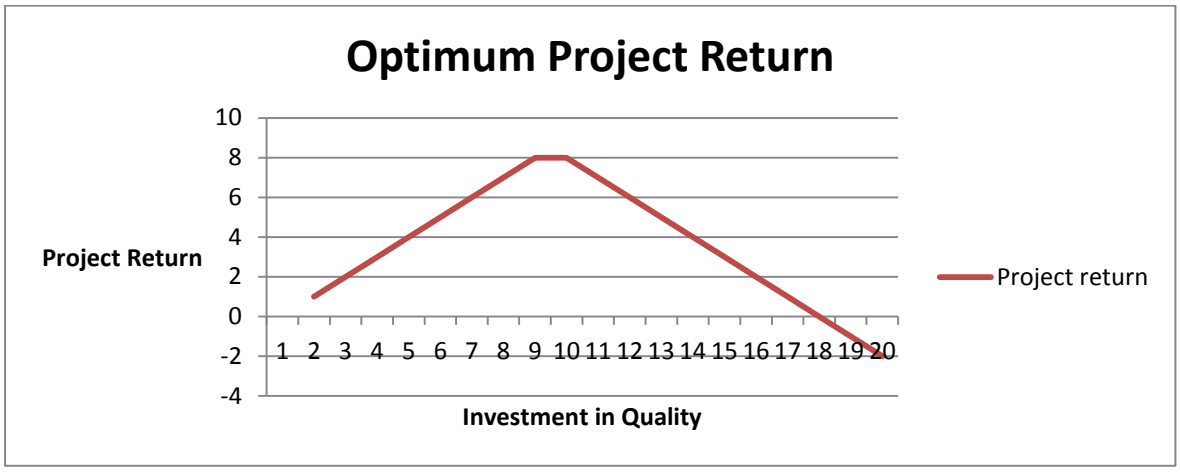
Maximising business value with quality

The business value of new features in a perfectly run Agile software project will show diminishing returns over time as the team deals with high-value features first and the lower value ones later.

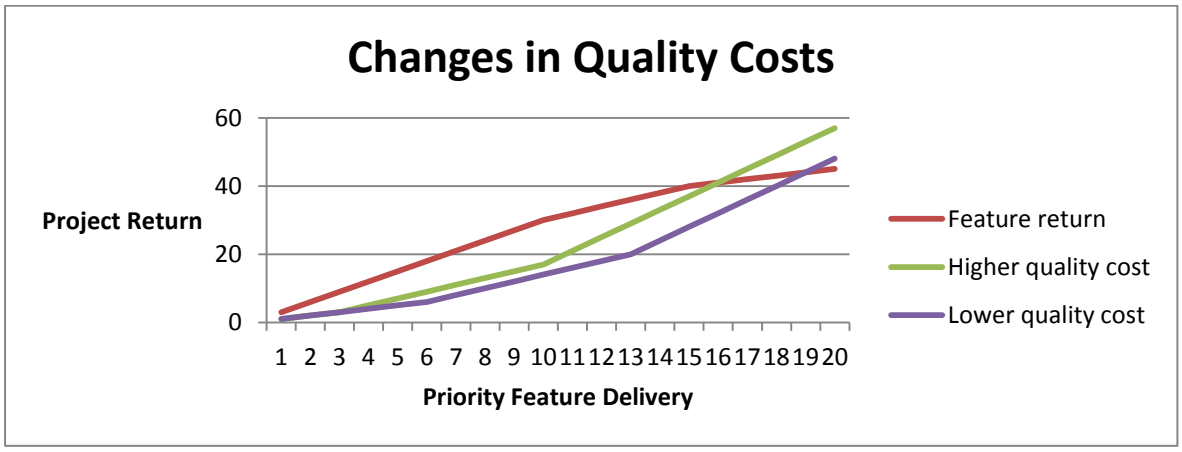
However, the profile for the cost of quality will change significantly depending on how the team is conducting its testing. In a typical environment however, you would start to see the cost of testing continually getting more expensive as more and more regression testing is created (while keeping the same level of quality). It should be noted that this effect can be slowed by the use of automated testing.

The cost of quality then needs to be added to the cost of developing the feature code of course. We will assume for this article that it will grow in line with the features developed.

The perfect quality situation is where you have reduced risk to the acceptable point, no more, no less. If you under invest, your risk profile will be too high. If you over invest, you are wasting resource. The graph below shows that as we go past "acceptable quality" all we are doing is damaging the overall project return.



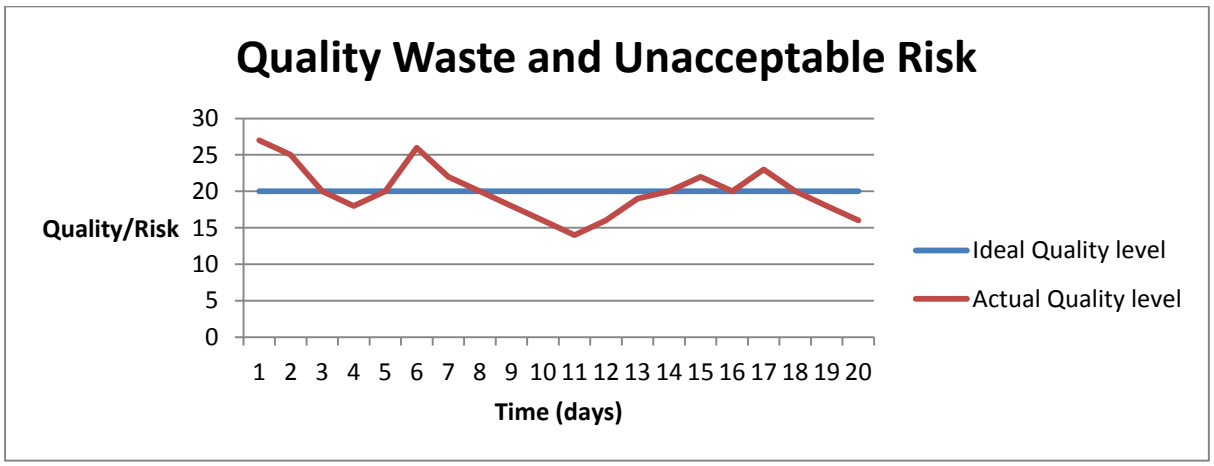
If we accept that our goal is to get the best return from a project, then we want to find the feature point with the optimal cost-benefit. The graph below shows what you may expect – if you lower the cost of quality, you can extract more project value. In the example, with "lower cost quality", you would say your optimum point to stop would be after feature 14 perhaps as there is a wide gap between the cost of quality and the overall return. However, with "higher cost quality" you would probably stop around feature 10.



Our aim is to reduce the cost of quality while staying exactly at the "acceptable" level.

How can we do this? The simplified knee-jerk answers of "cut the quality team" or "cut testing time" only answer the basic question of "how do I cut testing costs?" It must be noted that is not the same as "How do I cut project costs?"! The only time this is a good answer is if you are operating above "acceptable quality levels".

What we really need is the right information to flatten out a quality curve which may fluctuate between under and over investment in quality over time (too high a risk and waste respectively). This may look like the following:



To be able to flatten the quality curve, we need to look at the following:

1. What are the right areas to test to maximise most business value?
2. Is the risk profile of any areas too high at the moment?
3. Have we over-tested any areas and how are we going to prevent that in the future?
4. If current quality is acceptable for all features, don't just expand current testing, work on ensuring future quality in those features, for example by preparing test automation.

Conclusion

To maximise the business value of your quality processes by optimising your investment in quality, we need to be able to answer the following questions:

1. Are we as consistently at our "acceptable level of quality"?
2. How can I keep quality at the "acceptable level" for a lower cost?

In an area awash with data, the consequential data needed to answer these questions are:

- The business value of the features in your product.
- A quantifiable statement of your team's appetite for risk (leading to a calculated acceptable risk score for the project features).
- A calculated risk score for all of the features in your product.
- Cost data for the quality investment in each of those features.
- Efficiency ideas such as test automation.

Your goal should be to be able to extrapolate and analyse this data set.