

Other teams have their way of measuring the quality of their work. Software engineers utilize peer review of code to bring consistency and best practices to everyone on the team. Testers, who usually have the role of quality assurance, can look at the number of bugs found during manual and automated testing to determine whether the code produces an expected result. Customer support teams can look at response or turnaround times to ensure we’re responding to issues and concerns from our users.

**Measuring Quality**

In the prior article I talked about why quality is essential in the Product Management team, and why Product Management should be the steward of quality for the organization. In this article, I’ll give you some specific ways to measure and improve the quality of the product.

**How Product Management Can Measure and Improve Product Quality**

**By Jonathan Hotz**

Subject7, Inc. Copyright © 2022

|  |
| --- |
|  |

|  |
| --- |
|  |



* Quality communication is probably the hardest to measure. Still, you can get a sense by the number of times you need to explain information, or how many corrections need to be made to other teams’ work.
* Business goals are typically related to revenue, and you can measure the annual recurring revenue (ARR) or the customer lifetime value (CLV). They may also relate to the number of active user accounts or sign-ins a quarter.

To measure quality, we need to have something against which we can measure. This could include historical information, industry standards, and competitive/peer measurements. We also need to consider whether we use **lagging measures** or **leading measures**. Most of what has been described above could be regarded as lagging measures – telling us if we achieved the goal. We also need to consider the kind of leading measures we can employ, to determine if we are likely to achieve the goal. And for product managers, there are two leading measures and one lagging measure that we employ.

The first leading measure involves writing user stories that include acceptance criteria. In Agile software development, user stories are the primary way that product managers can convey information to the other teams about the user’s needs and the value we want to deliver to the user. These stories do not need to be extensively long and verbose, but instead they have enough information for people to understand the idea and encourage additional discussion among the team to clarify the details. Acceptance criteria is also something that product managers can and should include in their stories, to help the team as they discuss and determine the definition of done for the story.

When I first started writing software requirements, they were very detailed and directed at the development team, which didn’t provide an opportunity for the developers to consider other solutions and rarely did they engage in a conversation about the requirements. In Agile development, though, we value those interactions, and the stories I wrote provided for that interaction. The acceptance criteria can be as simple as stating what the application will do when the user performs some action. By adding acceptance criteria to the user story, we expanded the target audience to include the testers, and increased the opportunities for interaction.



**Acceptance Criteria in User Stories**

But how does product management measure quality in the software? This links back to what I mentioned about the success of product management, so we measure quality by delivering value to our users, communicating clearly to other teams, and aligning the product with business goals. The challenge is that these are mostly subjective, and often hard to quantitatively measure.

* Value can be measured using metrics related to product usage, like bounce or retention rates, conversion rates, or time spent on the platform, and it can also be measured through customer satisfaction surveys

Subject7, Inc. Copyright © 2022

**DEVELOPERS AND LOW-CODE TOOLS**

It's easy enough to create a simple Create Read Update Delete (CRUD) application based on database tables. In Visual Basic you can get a database control and connect text fields; in Ruby on Rails, you give the table a name and have the application render itself. The low-code tools take things to a new level.

There is often no software to install; the low-code tool runs in

your web browser in the cloud. Upload a spreadsheet and the

software will create a database and website for you. Add your company CSS to automatically style it.

For simple, straight-forward workflow automation, low-code tools are gaining a place. You can even drag and drop complex flows. That might include selecting flight, seat assignment, adding credit card information, then clicking to change only the return flight information.

Subject7, Inc. Copyright © 2021



The second leading measure is for product managers to participate in the Agile rituals. Product management does not need to lead each of these rituals, however, over the years I found a good balance with the engineering manager on the team where I would lead the backlog grooming and retrospective, and the EM would lead the rest. Whether or not you’re leading the ritual, participating in the sprint planning, daily standup, backlog grooming, sprint review, and retrospective increases the chance that you’ll catch something before it goes astray and means you and the team do not have to be reactive to issues and situations.

Some teams include the story estimation process, sometimes called planning poker, in sprint planning, and while I’ve found it more effective as part of the backlog grooming, participating in the story estimation process is important. It ensures that the expected scope or complexity of the story matches up with the estimate – when the expectation does not match the estimate, this is an opportunity for discussion and for the team to learn from each other about what is needed. In some cases, I found the interface design made the development very complex and time consuming, and we could make small changes to greatly reduce the effort. In others, the complex development was not needed, and we found a simple way to implement the design.

Validating UI designs and prototypes is generally a manual process. Still, we have an opportunity with the completed code to look at better and more efficient ways to validate, and that is with an automated tool. For simple products or functionality, it may be tempting to validate it manually. But the product will become more complex, you will have a variety of personas and users, and the data used may present different experiences, and if you must test all these permutations manually it can take a very long time. Product managers can utilize an automation tool, like Subject7, to build test scenarios that can go through these permutations, and you can do this in one of two ways: work with the QA team to build the tests, or build the tests completely on your own.

The lagging measure used in product management is to validate the work products delivered from each team: design, prototypes, and completed code. There are specific milestones in the design and development process, and at each of these points you can validate whether the work products align to the user needs and user stories. The earlier in the process you can validate the work, the less wasted time and effort if something needs to be adjusted. Product management can review the UI designs to make sure it matches up with the user needs, can review the prototypes to ensure the interaction will deliver value to the user, and review the completed code to ensure it matches up with the designs, prototypes, and user stories, which again lead back to the user needs and delivering value to the users.

**Participate in All Agile Rituals**



**Incorporate Automated Testing into Product Management**

**Validate the Work Products Delivered**

Subject7, Inc. Copyright © 2022



While it’s certainly helpful to work with the QA team, the tests they create often target specific functionality, and may not look at the end-to-end experience to ensure it meets a user’s need. Think about the workflows a user will perform from start to finish, and what they are trying to accomplish, and then build an automation flow that matches. Think about how different users will vary in this flow, and then layer on the data (different users and input values) to cover the permutations. By approaching the test from the perspective of the user’s need, and to validate that the feature will deliver value, you can more accurately measure the level of quality in the product.

As a product manager, take some time to step back from your day-to-day, and ensure you have these in place to deliver a quality product:

1. What aspects of quality can you measure in your work?
2. Do you have baseline values, or values to measure against?
3. Have you put in place leading measures?
4. Do you have milestones for the lagging measures?

Each one of us wants to deliver quality in our work and wants our product and organization to succeed. As the steward of quality for the organization, product management uses these leading and lagging measures to objectively measure quality and greatly increase the success of your team, product, and organization.

Subject7, Inc. Copyright © 2022