



[Print](#)  
[Close](#)

## Product Qualities Approach, Agile Style

Ryan Shriver

February 1, 2010

In many agile organizations, the product owner is responsible for setting the team's priorities through the product backlog. Whether they want enhancements to in-house systems or shrink-wrapped products, product owners get input from customers and stakeholders to create product backlogs of prioritized features (or user stories). These backlogs contain functionality that can be estimated by developers and planned for releases.

While there's nothing wrong with this approach of functions-first planning, I have come to believe it's short-sighted in that it doesn't place product qualities on equal pairing with functions. Currently in the agile community, there's a tendency to focus too quickly on user-centric functionality instead of product qualities that can deliver real stakeholder value, often very quickly. Product owners who understand and leverage product qualities cannot only delight customers, but also help them achieve their organization's business objectives.

This article provides a how-to for progressive change agents interested in delivering products that generate measurable business value for their customers and stakeholders. You'll learn how product qualities differ from functions, how to identify the right ones, measure them and use improvements to drive business results. Along the way, I'll demonstrate how to integrate an agile development processes such as [Scrum](#).

### What are Product Qualities?

Whereas functions describe *what* a product does, product qualities describe *how well* the product performs. This can be along an array of technical and business dimensions.

- Technical dimensions refer to *how well* the system performs, often referred to as “non-functional requirements”. Common ones include availability, response time, throughput, storage capacity, security, maintainability and accuracy.
- Business dimensions refer to *how well* value is delivered to the stakeholders--the business results of the product. This includes market-facing product qualities important to paying customers as well as those related to operational objectives important to business sponsors.

How long does it take to record a business event? How much training is required for new hires? How much will our efficiency improve with your product? This article primarily explores the business dimensions, but the concepts are equally applicable to both, as you'll see.

### Why are they important?

In crowded marketplaces where competitors have almost identical functions, products that perform at higher quality levels differentiate themselves from competitors. They also tend to be sold based upon their value propositions rather than viewed as a commodity, thus resulting in higher profit margins for the seller. Performing at higher-quality levels also has the benefit of being recognized as a leader in your industry, something that can only help sales.

For software, desired product qualities are commonly financially driven: increase revenues and reduce costs. Yet some product qualities can also have non-financial objectives such as customer satisfaction, net promoter score and team morale. Often, these non-financial objectives are important leading indicators of future financial results and are thusly important to consider.

One organization that puts product qualities forefront in their product development approach is [Confirmit](#). They sell marketing research software and [report](#) their products-focused approach is one of their [keys to success](#). Another organization that markets using product qualities is [Unica](#), the marketing software vendor. While I've never used their products, I do think they market quite well using product qualities aligned with their customer's needs. These include generating higher sales, retaining more customers and reducing operating costs. Notice these product qualities aren't features or functions, they are the business objectives of their customer: marketing department executives.

Here's a graphic from their [website](#) on the value proposition of their products:



Unica must do well enough with their approach to marketing. Gartner has it at the [top of their magic quadrant](#) for multi-channel campaign management, ahead of SAS, Teradata, Oracle-Siebel and other industry leaders.

### Identifying the Right Product Qualities

So how do we identify the right product qualities? While I don't believe there's a single right way for everyone, I've found the following recipe works well for me. I encourage you to explore what works best for you:

**1. Identify the Product Stakeholders.** Pair up with a partner and identify all the possible product stakeholders you can think of. Cast a wide net and identify anyone who is impacted by the product. Often the objectives of your stakeholders make good product qualities! Key stakeholder types include:

- **Customer** – purchases your product,
- **Business Sponsor** – funds development of the product
- **User** – various roles, uses your product to accomplish a task
- **Operations** – provides infrastructure and servicing
- **Trainers** – trains new users on your product

**2. Identify the Product Qualities.** Organize meetings with the individual stakeholders to learn about their impact on the product.

Come prepared with a set of draft product qualities and center the discussions on the important aspects of the product to them. I've found good questions that help reveal product qualities include:

- *What are the reasons customers purchase your product?*
- *What are your customer's objectives and how does your product help them achieve these?*
- *What would it mean to you personally, the organization and your customers if <insert product quality> improved?*

**3. Build Consensus.** Organize a workshop and present the key stakeholders the information you've gathered. Encourage candid discussion, but work toward getting consensus on three things:

- Highest priority stakeholder to serve first
- Highest priority product qualities to improve first
- Available budget of time and money for next release

Be efficient with stakeholder's time but also flexible to explore certain areas for deeper discussions. The goal is to get consensus to the highest priority product qualities for improvement next--not forever. There will be time to re-prioritize later based upon feedback; all we're looking for is a starting place for improvement. Ideally we prioritize all the product qualities, but if that's not possible, identifying the most important one is better than none at all.

You'll also want to do this same exercise with the technical leads for system qualities such as availability and security. The discussions should be around design ideas necessary to avoid the constraint levels--and ideally hit the target levels within the budgeted resources. If this cannot be done, the technical team's responsibility is to come back with alternative target or constraint levels that are achievable in budget, and use this for further stakeholder education and discussion. Depending on your resources, you can do this sequentially or in parallel to identifying product qualities.

These prioritized product qualities can go into your [results backlog](#) and serve as input into your agile release planning process. As to estimating how much these products will improve in the next release, we need to learn how to measure our product qualities.

### Measurable Product Qualities

While there's value in simply identifying and prioritizing the product qualities, the primary goal is to measure them. Why? [Tom Gilb](#) says it best:

*"The fact that we can set numeric objectives, and track them, is powerful; but in fact is not the main point. The main purpose of quantification is to force us to think deeply, and debate exactly, what we mean; so that others, later, cannot fail to understand us."*

Defining measurable levels of improvement to product qualities forces us to have open and honest discussion with stakeholders. This ensures expectations are aligned with how much better, faster, quicker or improved the new release will be while working within the resources--or else what additional resources are necessary to reach the desired levels.

I prefer to define product qualities with the following minimum attributes:

- **Name** – Brief unique identifier
- **Scale** – What's measured (units)
- **Meter** – How it's measured (method)
- **Targets** – Levels aiming to achieve
- **Constraints** – Levels trying to avoid
- **Benchmark** – Current or past performance levels

In order to fill in the details for the highest priority system qualities, I use a similar approach to before. Working with my partner, we create a draft set based on our working knowledge and then validate and fill in the gaps with stakeholders offline. We then gather stakeholders together again to present the information back and get consensus on each attribute. Depending on your project, you may find it more efficient to do a single workshop to accomplish all of this. The method used to gather the product qualities isn't as important as ensuring the information is complete and that consensus is reached amongst key stakeholders.

The following figures illustrate an example product company seeking to increase market share, monetary donations and volunteer

time donations. Below are the product qualities most important to the business sponsors.

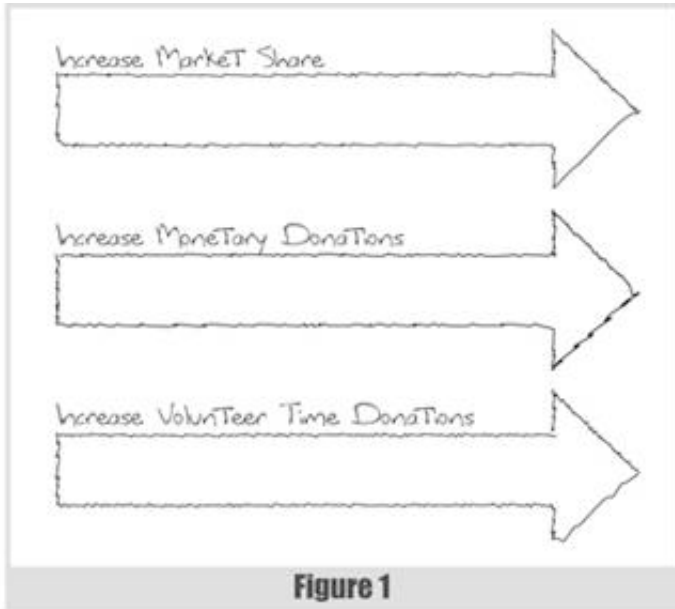


Figure 1 - Product Qualities

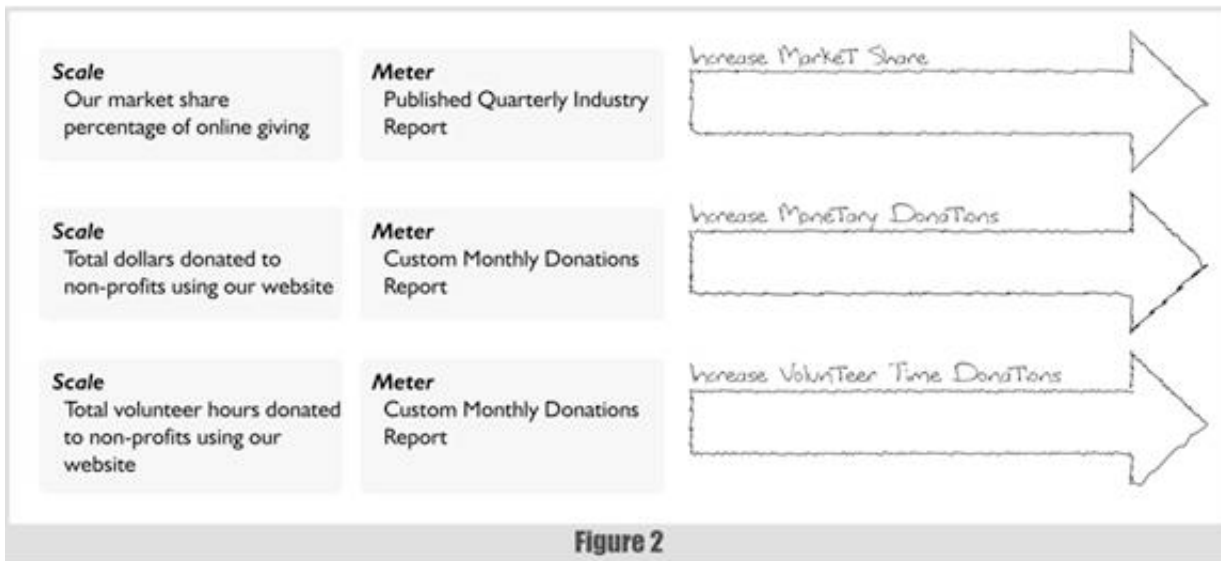


Figure 2 - With Scale and Meter added

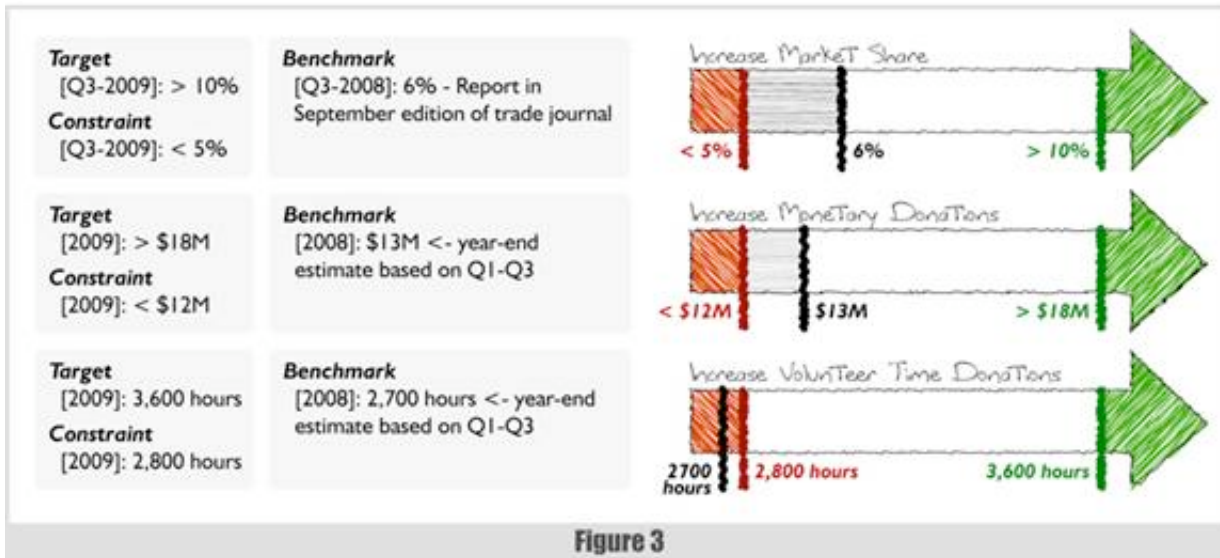


Figure 3 – Completed with Target, Constraint and Benchmark

Let's look at an example system quality using the same attributes:

**Name: Availability**

**Scale: Percentage of time system is available for accepting transactions excluding 1 hour weekly maintenance window**

**Meter: Weekly Monitoring Report from data center polling application every 3 minutes**

**Target: >= 99.9% (< 10 mins unplanned down)**

**Constraint: < 99.4% (> 60 mins unplanned down)**

**Benchmark [Date/Version]: \_\_\_% <- Benchmark Source**

Hopefully you can see from these examples that product qualities can be defined simply and succinctly. (In my experience, three product qualities can fit on one PowerPoint slide and up to 10 on a piece of paper!)

**Planning and Reporting with Product Qualities**

Now that we've quantified our product qualities, they can be integrated into planning and reporting activities such as:

**Balanced Scorecard** – For the executives, integrating product qualities reporting into the scorecards can clearly communicate progress toward targets on the highest priority product qualities (including what resources were used to achieve these results). Figure 3 above shows one visual representation of progress toward targets useful for reporting to executives (but often text is the simplest and easiest means to communicate results).

**Results Backlog** – In [my recent article](#), I discussed product owners using a companion to the Scrum product backlog called a results backlog. The idea was to create an artifact to manage and prioritize business objectives so the actual business results, in addition to the product features, could be managed. The results backlog concept applies to the product and system qualities as well because both measure ends, not means.

**Release Planning** – For the product owner, agile coach and the team, as you evaluate each new proposed feature during release planning, ask yourself:

- Which of our highest priority product qualities will this feature improve?
- Will this feature alone get us to our target performance level or do we need to consider additional or alternative features, too?
- What percentage of our resources (time and money) will it take to implement this feature (and what's remaining to improve

other quality levels)?

Although these questions can work at the user-story level, I find it's helpful to work at the feature level first--before breaking the feature down into user stories. If a feature has positive impacts on the product qualities, then the component user stories should as well.

**Value Decisions**– Sometimes referred to as an impact estimation table, [a value decision table](#) helps make informed decisions by assessing how means (such as technical design ideas, features or projects) impact ends (such as product qualities or business objectives) using some percentage of the budgeted resources (see Figure 4). The result is a benefit-to-cost ratio that indicates the “bang for the buck” delivered. In addition, by summing the impacts we can see the total impacts if all design ideas were implemented. For an example of how this can be integrated with Scrum, see my article [Measurable Value with Agile](#).

	Design Idea #1	Design Idea #2	Design Idea #3	Total Impacts
<b>Objectives</b>	Impact on Objective	Impact on Objective	Impact on Objective	Total Impact on Objective
<b>Resources</b>	Impact on Budget	Impact on Budget	Impact on Budget	Total Impact on Budget
Benefit to Cost Ratio	Ratio	Ratio	Ratio	

**Figure 4**

Figure 4 - Value Decision Table

### Deliver, Review, Adjust and Repeat

While much of this article has been about planning, there's still execution that must excel in order to achieve results. While I don't overlook this aspect, I have learned that well-coached agile teams can start delivering software on a frequent basis relatively quickly. While I know from first-hand experience that software delivery is challenging in its own right, I've learned that most agile organizations learn how to do the thing right early on but can struggle with knowing how to do the right thing for years!

It is important to review progress toward goals at key milestones such as each release or monthly or quarterly meetings. Pay special attention to the resources necessary to reach the performance levels. Does this reveal a new insight? Does this set (or reset) expectations on what's realistically achievable in the future? If the technical team was over confident the last time, now might be the time to lower target levels or increase budget in order to reach target levels. These are all topics ideal for discussion. Remember: The goal is to continually improve and adjust as you go, reviewing at the right times to make informed decisions. How this works in one particular organization will vary from another, but the important point is to do it repeatedly in order to gain the benefits of learning and continuous improvement.

### Summary

Today we've learned what product qualities are, why they are important and how to identify, prioritize and quantifying them succinctly. We've learned how to make better-informed decisions with numbers and communicate results to stakeholders.

Together, these techniques form the basis of a product qualities approach to development that can be integrated with agile

development teams. This approach can help your agile teams focus on what's most important to the stakeholders using clear terms and numbers everyone understands. The result is a win-win: Stakeholders get measurable results on their highest priorities and the team gets the satisfaction to knowing they are making a real difference.

[Ryan Shriver](#) is a Managing Consultant with [Dominion Digital](#), a Virginia-based process and technology-consulting firm. Based in Richmond, he leads the [IT Performance Improvement Solution](#) which includes [Agile Adoption](#), [Agile Engineering](#), [IT Process Improvement](#) and [IT Services Management](#). With a background in systems architecture and large-scale agile development, Ryan currently focuses on measurable business value and systems engineering. He writes and speaks on these topics in the US and Europe, posting his current thoughts at [theagileengineer.com](#). Ryan can be reached at [rshriver@dominiondigital.com](mailto:rshriver@dominiondigital.com).

Copyright © 2010 gantthead.com All rights reserved.

The URL for this article is:

<http://www.gantthead.com/article.cfm?ID=254127>