Proper Product Backlog Prioritization

To ensure Agile project success, Scrum teams need to understand key organizational imperatives and work effectively to maintain focus on priority items with the greatest business impact, even as circumstances change.

Executive Summary

The success of any Agile project depends on the ability to deliver the most valued items to the business at the earliest. Thus, it is imperative that the right requirement is chosen, one that will deliver value to the customer as early as possible. At the same time, a lower priority is to be assigned to more volatile requirements so that the rework by the development team is minimized. And there are many other factors which drive the product backlog prioritization in an Agile project.

This paper offers guidelines to Scrum product owners and Scrum development teams on prioritizing product backlog correctly and consistently.

Business Challenge

Product backlog prioritization is one of the key success factors for an Agile project; however, most product owners and project teams face challenges in coming up with well prioritized lists of backlog items. This often results in a change in the Sprint backlog amid each Sprint. The reason: A lack of understanding by customers, end users and the project team of the value of properly prioritizing requirements. As a result, customers and end users may look at the prioritization question with a mindset of “everything is a priority for me.”

If everything is an equal priority, then nothing has true priority. But if we deep dive into the requirement list, we may find items that are necessary to build the basic architecture of the product or items that are business-critical and thus should be kept high on the priority order. In the following sections we will examine why we should prioritize the product backlog and offer guidelines for proper prioritization.

Why Prioritize?

The speed of a customer receiving return on investment (ROI) is driven primarily by how well the items carrying the most business values are developed and released to the customer. So before going into the details of how we prioritize, let’s first understand the reasons for focusing on prioritizing the backlog items.

• **Improve customer satisfaction:** The first and foremost goal for any project is to delight the customer. The customer and the system's end users have certain goals in mind while deciding on making the investment on the project. The sooner the goals start to be realized, the more delightful the experience will be for the customer.

• **Reduce risk:** During prioritization, the risks and key performance parameters are analyzed
critically so problem areas surface early in the project. This provides more time for reducing the risk associated with hard technical problems and key performance parameters.

- **Keep the product backlog delivery focused:** During prioritization, each requirement is analyzed to see if it will add value to the business or is necessary for the technical architecture. If it does not fall into either of these categories, it is pushed to the lowest end of the backlog or may even be eliminated. This helps to ensure the backlog contains only mandatory candidates for implementation.

- **Manage your constraints:** It is very likely that some projects may have a constraint on available time, resources, funds, etc. By allocating resources and efforts towards the highest priority items, organizations make sure that even if all the items are not finished the most important ones are delivered.

- **Create system architecture:** While creating the prioritized requirement list, the technical importance of each requirement is also analyzed. That, in turn, helps in identifying those items critical for building sound system architecture. Note that the backlog prioritization should involve technical input along with business case input.

- **Keep on schedule:** A well-prioritized backlog helps in maintaining the schedule and keeping the Sprints meaningful, in case there is a chance of missing some deliverables. Think about a situation when the team is nearing the end of an iteration and finds that all features cannot be implemented in the specified time box. This can be handled by delaying some low-priority features instead of delaying the schedule or compromising on QA activities, which could endanger the delivery.

- **Nonfunctional and performance requirements:** During the prioritization, end users and systems developers must articulate their usability, security and performance expectations and rate the importance of each item. This can yield huge benefits as the team comes to grips with the nonfunctional requirements early in the project and thus builds the underlying framework accordingly from the get-go.

**Prioritization Considerations**

Achieving proper prioritization is not easy. It requires clear goal setting, focus on the business outcome and technology knowledge to arrive at a priority list that gets approved by the business and is feasible for implementation. Key considerations and guidelines for prioritizing the product backlog include the following:

- **Customer satisfaction:** Features that will help the customer realize its ROI and, in turn, increase customer satisfaction should get high priority. This requires a thorough understanding of customer preferences to arrive at a satisfaction figure for each feature.

  Professor Noriaki Kano developed the Kano model to help analyze this with the help of simple questions such as those presented in the box:

  - **What will be the customer’s reaction about this feature?**
    - **Favorable:** Does this feature increase the customer satisfaction?
    - **Critical:** If this feature is not delivered, will it cause customer dissatisfaction?
    - **Indifference:** Will the customer look at this as an expected basic feature?
    - **Questionable:** Is the customer doubtful about the usefulness of the feature?
    - **Unfavorable:** What if the customer dislikes the feature?
    - **Will this feature be the life-point for this product?** Can the product be released without this feature?

  By plotting the features in the Kano model, an organization can gain an idea of what features are critical to customer satisfaction and thus should be high priority (see Figure 1).

- **Measure of success:** Understand the critical success factors defined by the customer and thus satisfy the customer’s critical needs first.

- **System architecture:** The features with a large impact on system architecture should be addressed with highest priority. Any change in these items will cause changes in many other entities and thus lead to rework and system instability.

- **Business value:** Understand the business value of each of the requirements. Some may be important for the customer from the long-term business strategy point of view whereas others
Gauging Customer Satisfaction

![Kano model diagram](image)

**Note:** Prof. Noriaki Kano from Tokyo Rika University first published the concept of the Kano model in 1984 in the article “Attractive quality and must-be quality,” in the journal of the Japanese Society for Quality Control, April, pp. 39-48. He received an individual Demming Prize in 1997.

Figure 1

... may provide an immediate edge over the competition. Based on the key sponsors and drivers of the project, the precedence changes. For example, the CEO of the end-user organization may be more interested in the strategic points while the marketing team may be focused on the immediate value adds.

- **Cost-benefit analysis:** It is sometimes misconstrued that high-cost items should be placed lower in the product backlog. But the basic measurement factor should not be the cost alone; the focus should be on comparing cost with the benefit the product reaps. The importance of realizing ROI has created a misguided view that investment should be kept low to achieve a high ROI. Thinking rationally, a higher return also gives a higher ROI. So, consider the cost-benefit ratio for a better understanding of the value of each requirement. Cost should include money as well as other resources like time, labor, finances, overhead, infrastructure, shipping, etc. The high-cost, low-value items should receive lower priority.

- **Complexity:** Address the complex requirements first. Applying higher priority to complex items will ensure that the best resources are properly allocated.

- **Difficult-to-implement features:** Put high priority on items that are difficult to implement. This will ensure that enough time is given to implement and optimize each of these items.

- **Effort for implementation:** The items that require more time or effort to implement should be implemented first. The reasoning is similar to the point above and it avoids the situation that important features are missed due to time constraints toward the end of the project.

- **High-risk features:** Giving a high-risk feature higher priority ensures that they will be scheduled early on, thus reducing the overall risk for implementation.

- **Safety:** Critical safety features need to be prioritized first. Often this is led by regulations.

- **Frequency of use:** The features that are likely to be used more frequently should have high priority. These features most often qualify for optimization and thus scheduling them early will provide scope and time for optimizing them.

- **Reusable components:** Identify the reusable components and assign them higher priority.
These items will reduce future implementation efforts and thus need to be implemented early.

**Stability:** How a feature affects the stability of other features plays a key role in deciding its priority. Stable features should be implemented first. If the feature is dependent on external systems, and is likely to change in the near future, then the implementation should be assigned a lower priority. But in case this item is going to change other existing items in the system, then it will be necessary to give it a high priority so that the changes can be forced early.

**Ways of Prioritization**

Arriving at the correct set of prioritizations is not easy. It takes lot of negotiation with different stakeholders and users of the systems to arrive at the final priority list.

Speaking with the end users is the first and foremost requisite to understand what is important, and doing so is the primary driver in achieving project success. Then the other parameters highlighted in the section above need to be discussed with multiple teams.

Below we place some quantification around the priority analysis model so that the final priority list has a scientific basis instead of being qualitatively oriented. This will also ensure that at any later point of time the priority of a particular feature can be supported with enough justification and explained easily to stakeholders.

**Priority Quantification Model**

In the “Prioritization Considerations” section above, we identified the items that are crucial for backlog prioritization. We combine this with the concept of relative sizing and arrive at a model to quantify the priorities and to take informed implementation decisions. The step-by-step approach for this is explained below:

- Assign a number within the relative weight scale of 1 to 10 to each of the prioritization criteria based on their importance. For example, if customer satisfaction is the most important criterion, then assign it a weightage of 10; if the next important criterion is the business value, then assign a weight of 8 or 9. There may be a situation where the system architecture is the most important part of the system to be developed, which then means assigning it top status of 10. But for the time being, assume that we are able to decide on the relative scoring of few items like this:
  - Customer satisfaction = 10.
  - Complexity = 5 (meaning the importance of implementing complex features first is half that of customer satisfaction).
  - Business Value = 8, etc., as shown below, where the weightages are shown within braces.

<table>
<thead>
<tr>
<th>Customer Satisfaction (10)</th>
<th>Complexity (5)</th>
<th>Business Value (8)</th>
<th>Total Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog item 1</td>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Backlog item 2</td>
<td>8</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Backlog item 3</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Backlog item 4</td>
<td>4</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Backlog item 5</td>
<td>7</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

• Now rate each of the product backlog items against each of these criteria with a relative rating in the scale of 1 to 10. Note: Here we are using the relative sizing technique, which means comparing the backlog items against each other while deciding on the rating for these criteria. This rating is based on the relative contribution of that particular backlog item towards the prioritization criteria as perceived by the client or the business user. Similar to the relative sizing technique, it is important to decide on the baseline feature before starting the relative sizing as you will be rating each of the features or user stories against the baseline feature.

  - At the end of this exercise we will arrive at a matrix like this:
Multiply the rating against each cell of the backlog item with the weightage of the criteria and sum these across a row to arrive at the total score.

So, our backlog item 2 creates the highest value followed by items 5 and 4.

Some enhancements can be performed by introducing a negative sign before the weightages of the criteria which impact the priority in negative ways. For example, a requirement that has high dependency on some third-party interface should bring down the priority score for that item.

You may also bring the cost component into the picture and then compare the combined effect of the cost of implementation and the value delivered to take the final decision about the prioritization.

Looking Ahead

We have discussed the importance of having a properly prioritized backlog while working on an Agile project and discussed the points that should be considered to arrive at the priority list. We also quantified the priority model by combining the concepts of relative sizing and pointers in prioritization criteria. It is very important to understand that once you arrive at the prioritized product backlog you are not freezing the list - as priorities will change due to many reasons, such as:

- You gain a better understanding of the customer's needs as the project progresses.
- There is a change in stakeholders and/or the client's organizational dynamics.
- Various features are implemented, necessitating a reshuffling in the other features' priorities.
- The initial uncertainty in the system is resolved and the team is focused on more stable features.

So, treat the prioritization task as an ongoing activity and keep a constant eye on the drivers that may cause the priority change.

About the Author

Saikat Dutt is an Associate Director in Cognizant's Customer Solutions Practice. He has 14 years of experience in managing large-scale multilocation and mission-critical projects. He is the author of the book PMI Agile Certified Practitioner – Excel with Ease, which is published by Pearson. He is a Project Management Professional (PMP), certified by the Project Management Institute (PMI). He is also a Certified Scrum Master and is an active speaker on Agile best practices and project management methodologies in several forums. Saikat can be reached at Saikat.Dutt@cognizant.com.
About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world’s leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 50 delivery centers worldwide and approximately 150,400 employees as of September 30, 2012, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.