Transcript

Prioritizing User Stories

Learning Objectives

After completing this topic, you should be able to

- *use the MoSCoW model to create a prioritized user story backlog*
- *recognize how various factors influence the prioritization of user stories*

1. Prioritizing user stories

Once an agile team has collaborated with the customer to ensure all project requirements have been identified and, if needed, expressed in the form of user stories, the customer – or product owner – finalizes the order of the stories based on their business priority. The team can then focus on developing stories with the highest priorities first.

Prioritizing user stories ensures that the most important product features are delivered to the customer early on, so that the customer can begin deriving value from them. So it's in keeping with the agile principle of maximizing the value of a project for the customer, from the outset.

But how do you determine the relative priority of each user story?

This isn't always a simple process. For example, a user story might describe a product feature that's of high value to the customer. But what if the cost of developing this story early on will mean there's little budget left for the rest of a project? Although prioritization is necessary, it can be challenging.

Factors a product owner may use to prioritize user stories include their value to the customer, cost to develop, and technological and market risk. They also include the extent of the learning that will occur during story development, any technical dependencies, and level of releasability.

Select each prioritization factor for more information about it.

Value

The value to the customer of the functionality described by a user story is always one of the key criteria for establishing its priority. In addition, you should ensure that the stories you prioritize have sufficient financial value for the organization to ensure you're able to finance their production.

The value of a particular deliverable can be assessed in different ways. For instance, the customer might estimate how many units of a released product it can expect to sell or the average price it can charge for each unit commercially.

Value-based prioritization in large projects may involve a detailed evaluation of the expected return on investment of the product, taking into consideration the net present value of investments, and internal rate of return over time.

Cost

The cost of developing a feature may influence its priority. Cost in this case includes the total cost of developing the relevant functionality or product component, as well as the cost of supporting it once it has been developed.

For example, introducing a new product feature in the customer's business environment may make it necessary to provide users with expensive training, or require a costly change in business processes. The associated costs reduce the value of the product feature and may prompt the customer to give the relevant user story a lower priority.

Risk

Risk is an important factor to consider when prioritizing user stories. Generally user stories associated with high risk, high value features should be given higher priorities so they can be addressed early on in a project.

Market risk refers to uncertainty about how a market will respond to a product or product feature. The earlier a team releases a product with high market risk, the sooner it can begin receiving feedback about the market's reaction. Plenty of time is then left to adjust the product or feature as necessary, to minimize risk.

The same is true for technological risk. If developing a user story is associated with a high level of technical difficulty and risk, addressing the story as early as possible gives the team the most time to iron out any problems. This also removes a large part of the project risk early on, which is desirable.

Learning

It's appropriate to consider what learning the team will acquire as a result of developing a user story when deciding on the story's priority.

For example, developing a particular user story might involve having to learn specific skills that will prove useful in subsequent iterations. The user story may be assigned a higher priority for this reason.

Dependencies

Certain user stories may have to be developed in a particular sequence because of their technical dependencies on one another. It's important to factor in these dependencies when prioritizing stories.

For instance, it may be possible to develop the functionality described by one user story only once another user story, which involves creating a particular interface, has been completed.

Releasability

Releasability refers to the extent to which a deliverable can be released – without requiring further development work or additional components – and so generate value. You should give higher priorities to user stories describing deliverables with high releasability. This is because the earlier a product is released, the sooner it can start earning a return on investment.

Early releases also give a development team time to obtain feedback from customers and users, and to adapt based on the feedback in later iterations.

Question

You're the product owner for a project that involves developing a new suite of database management software. Currently you're deciding on the priority of a particular user story. It's likely to be relatively expensive, although not prohibitively so, to develop the feature described by the user story. During development, the team will have to become familiar with a new technology – and this may be of benefit for developing other stories. The customer feels the feature will add to the usefulness of the software, but it's not imperative to include it early on.

What priority level should you give this user story?

Options:

1. Very high priority
2. Fairly high priority
3. Very low priority

Answer

Option 1: *Incorrect. The user story shouldn't be assigned a very high priority level because the feature it describes isn't of high value to the customer.*

Option 2: *Correct. It's appropriate to assign the user story a fairly high priority level because it represents a learning opportunity for the development team – one that may have benefits during the development of other stories. It will be fairly expensive to develop but the cost isn't prohibitive, and it will provide a feature of some value to the customer.*

Option 3: *Incorrect. There's no reason to assign a very low priority to the user story in this example. It represents a learning opportunity for the team and, although it will be relatively expensive to develop, describes a feature that's of value to the customer.*

Correct answer(s):

2. Fairly high priority

2. The MoSCoW model

A simple but popular model for prioritizing user stories is known as the MoSCoW model. Using this model, a customer groups user stories into four categories – must have, should have, could have, and would have.

Select each of the four categories for more information about it.

Must have

User stories in the "must have" category should be assigned the highest priorities. They describe features that are considered critical.

Should have

User stories in the "should have" category describe features that are of high value to the customer, although they're not considered critical and short-term workarounds can be implemented for them if necessary. These stories should be given fairly high priorities.

Could have

User stories in the "could have" category describe features that are considered desirable but that will be included only if time and funds allow for this. These stories should be assigned lower priorities than those in the "must have" or "should have" categories.

Would have

User stories in the "would have" category should be assigned the lowest priorities. They describe features it would be nice to include but that aren't of high importance, and so are marked for probable inclusion in a later release or project.

An agile team may decide to use the MoSCoW method to create initial groupings of user stories, and then use another method to fine-tune the priorities of the stories within each category.

Question

Match letters in the MoSCoW acronym with what they stand for. Some descriptions won't be used.

Options:

A. M

B. S

C. C

D. W

Targets:

1. Must have
2. Should have
3. Could have
4. Would have
5. Might have
6. Critical

Answer

The M in MoSCoW stands for "must have." User stories in this category are considered crucial and have the highest priorities.

The S in MoSCoW stands for "should-have." User stories in this category are important but short-term workarounds can be implemented for them if necessary.

The C in MoSCoW stands for "could have." User stories in this category can be left out of a release if there isn't sufficient development time to include them.

The W in MoSCoW stands for "would have." User stories in this category are desirable but marked for inclusion in a later release or project. They have the lowest priorities.

The M in MoSCoW stands for "must have", rather than "might have." User stories in this category are assigned the highest priorities.

The C in MoSCoW stands for "could have." User stories in this category can be left out of a release if there isn't adequate time to include them.

Correct answer(s):

Target 1 = Option A

Target 2 = Option B

Target 3 = Option C

Target 4 = Option D

Target 5 =No Option Specified.

Target 6 =No Option Specified.

Consider how a team might use the MoSCoW model to prioritize stories in a project that involves creating a new application for tracking, buying, and selling stock online. First the team, together with the customer, picks out one theme – such as all user stories related to reporting stock price updates. Then the team spreads out the cards for the user stories in that category.

Next the team discusses the first user story in the group, identifying all the factors that should affect its relative priority level. For example, the feature described by the user story will save users time and so is of high value to the customer. It won't cost much to create a central reports module, which is common to many business applications.

Graphic

The relevant user story is "As a client, I want to receive constant stock price updates."

The team has experience in developing similar features, so both risk and the potential for learning are low. The feature isn't fundamental to the system architecture. Its main dependencies are that it should be created after the database and OLAP cubes are in place. It has low releasability because the reporting function isn't useful in isolation.

Once the team has considered all these factors, it decides to place the story in the "should have" category – meaning that it's desirable but not critical.

Question

One user story for a web site development project involves developing an interactive, introductory page for the web site. Although this feature isn't fundamental to the web site's architecture, it has high releasability. It won't be expensive to develop, but the customer feels it will add significantly to the effectiveness of the site. The web developers are familiar with building interactive web pages, so both risk and the potential for learning are low.

What MoSCoW rating should the team give this user story?

Options:

1. M
2. S
3. C
4. W

Answer

Option 1: *Incorrect. The letter M represents the "must have" category. Features in this category are functionally critical, which means that the software would fail if they were omitted. That isn't the case with this feature.*

Option 2: *Correct. The letter S represents the "should have" category. The feature described isn't a critical feature, but it should be developed because it would be inexpensive and low risk to develop, yet provides high value to the customer and is highly releasable.*

Option 3: *Incorrect. The letter C represents the "could have" category, which is for user stories that are desirable but have fairly low priorities. In this example, the user story should be given high priority.*

Option 4: *Incorrect. The letter W represents the "would have" category, for user stories with the lowest priorities. In this example, however, the user story describes a feature that should be given high priority.*

Correct answer(s):

2. S

The rating process continues until all user stories have been assigned a priority letter of either M, S, C, or W. The next step is to create a user story backlog. This can take the form of a whiteboard or wall to which user story cards can be pasted, or of a simple chart in a program like Excel.

If a team has the room for it and would like something more visual, it can create a large story map.

You use M-rated stories to create the backbone of a story map. You should then arrange the S-rated stories, and finally add C-rated and W-rated stories. Finally the project team should work through the map, identifying and discussing any issues.

Select each step to learn more about creating a story map.

1. Use M-rated stories to create the backbone

You start to build a story map by placing all the M-rated user stories in a line, sequenced according to their themes and logical order of development. For instance, structural themes may be on the left, progressing to user interface themes on the right.

This sequence of stories forms the backbone of the map. They describe fundamental features that will make the product work and meet the customer's most importance acceptance criteria.

2. Arrange the S-rated stories

As the second step, you place all the S-rated stories beneath the M stories – arranged per theme or according to their logical associations. For example, a story about a server-side application for stock price updates that the team has rated "S" would go below any M-rated stories related to the same stock price reporting feature.

3. Add C-rated and W-rated stories

As the third step, you add the C-rated user stories below the associated M- and S-rated stories. Finally, you add W-rated stories to the bottom of the hierarchy.

The map will then show all the user stories for a project in a hierarchy, and organized according to theme.

4. Work through the map

As the final step, team members should work through the completed map, progressing from left to right, and downward through the priority levels. They should discuss any issues that arise as they go, making changes where necessary – for example, if a technical dependency makes it necessary to change the order of stories.

The story map is then essentially the project plan. The team will create the highest priority items first, working their way down through the levels as a project progresses.

During later planning, it's best to allocate a mix of critical – or "must have" – user stories and stories with lower priorities for development during each iteration. The lower-priority stories provide a buffer if time runs out and something has to be dropped.

Within a particular iteration, it also becomes impossible to prioritize work if all user stories have the same priority rating.

Question

You're creating a user story map for a project that involves developing custom software for a client. To start, you've arranged and grouped all the M-rated user stories on a whiteboard.

What should you do next?

Options:

1. Group and arrange the C-rated user stories
2. Group and arrange the W-rated user stories
3. Group and arrange the S-rated stories
4. Work through the map, from left to right and top to bottom

Answer

Option 1: *Incorrect. You should add the C-rated stories only after adding S-rated user stories.*

Option 2: *Incorrect. You should add the W-rated stories, which have the lowest priorities, only after adding the S- and C-rated stories.*

Option 3: *Correct. Once you've laid out the M-rated user stories to form the map's backbone, you should add all the S-rated stories beneath the M stories. You should arrange the stories logically or according to theme.*

Option 4: *Incorrect. The team should work through the map, identifying and discussing any issues, only once all user stories have been added to it.*

Correct answer(s):

3. Group and arrange the S-rated stories

Summary

As part of an agile approach to project planning, the customer collaborates with the development team to prioritize user stories, ordering them based on the order in which they should be developed. Factors to consider when prioritizing user stories include their value to the customer, cost to develop, risk, releasability, the potential for learning they provide, and dependencies.

The MoSCoW model can help a team prioritize user stories by dividing them into four categories – must have, should have, could have, and would have.