MVP

May 29, 2020

1 Project Minimum Viable Product (MVP)

1.1 Introduction

As you learned throughout this course, the Product Owner is largely responsible for determining 'what' the development team works on and the priority of that work while the development team determines 'how' the work should be completed. It is very important that all members of an agile team understand and are able to create a good product vision, product roadmap, and user stories. In this project you are going to practice creating a product vision, product roadmap, release plan, user stories with acceptance criteria, and ultimately a MVP plan for a software product.

1.2 Scenario

You work at the Centers for Disease Control (CDC) on the infectious disease tracking & prevention team and there is currently a world wide virus outbreak that is at pandemic levels. The top public health official at the CDC has asked you to work with the software development team to develop a product that tracks the spread of the virus in an effort to help stop and prevent the virus from spreading further. There is currently no vaccine or cure for the virus and the CDC wants to know where to apply funding, where to send test kits, and eventually be able to track the status of vaccine and cure creation.

Here are a few clear and critical outcomes for the first release of the product to be successful: - It is critical that anyone in the U.S. is able to report that they have contracted the virus - It is critical to show heatmaps of the outbreak - It is critical to be able to track and compare the - Number of virus test kits that health care companies have created. - Number of virus test kits administered by the test centers - It is critical that medical examiners are able to report the number of deaths caused by the virus

Some less important and unclear outcomes for the product are: - Pharmaceutical companies are working on vaccines and cures and need to be able to report on their progress - State and Local Government Officials want to push out updates to the public through the product - Local Doctors need to be able to report successful remedies and cures for patients that have contracted the virus. - Having the ability to track the # of virus test kits that have been delivered to test centers

The Agile development team you are a member of is using the scrum framework. The team is working in 2 week sprints and has an established velocity of 8 story points/sprint. Assume the following for your team in this scenario:

More Simple To Build	vs	More Complex to Build
Manual Processes	vs	Automated Processes

More Simple To Build	vs	More Complex to Build
Web Portal	vs	Mobile App
View in Web/App	vs	Export from System to View or Send
Non-customizable & Non-interactive	vs	Customizable & Interactive

You need to plan a MVP that needs to be delivered in 3 sprints.

1.3 Step 1: Product Vision

Create a Vision Statement for the Product being delivered based on the scenario described in the **Product Vision** section in the **Minimum Viable Product Plan Vision & Justifications** Document template.

1.4 Step 2: Product Roadmap

Create a High-Level Product Roadmap for the Product on the **Product Roadmap** tab in the **Minimum Viable Product Plan** Spreadsheet template.

1. High-level Product Features are listed in **Column A**.
2. Based on the scenario, place the Product Features into the phases on the product roadmap. The product features should be ordered so that the highest value features are delivered to users first and the simple to implement features are delivered before the more complex features. Each cell in the four phases of the product roadmap has a dropdown list that allows you to select the product features that you want to appear in that phase of the roadmap. Every feature must be assigned to a phase of the roadmap.

Justify your placement of each High-level Product Featuures in the **Product Roadmap Justifications** section of **Minimum Viable Product Plan Vision & Justifications** Document template.

3. After assigning the features to the phases of the roadmap choose 1 feature to provide a simple justification for placing that feature into the phase that you put it in. (4 Justification statements total)

The focal point is to deliver the highest value to the customer at all times. Question what provides value: an update or a new feature.

1.5 Step 3: User Stories & Acceptance Criteria

Create User Stories and Acceptance Criteria on the **User Story** tab in the **Minimum Viable Product Plan** Spreadsheet template.

1. There are 7 user roles that have been identified in the **User Role** tab in the spreadsheet. There is a brief description for how each user will interact with the product.
2. Two user stories have been provided for you for each role in the **User Story** tab. Create at least 2 more user stories for each role based on the needs of each user in the same tab. You will have at least 28 total user stories when finished.
3. List at least 2 acceptance criteria for each User Story.

1.6 Step 4: Prioritize User Stories

All of the user stories that you created will appear in the **Priority** tab in the **Minimum Viable Product Plan** spreadsheet template.

1. Prioritize the user stories so that the highest value features are delivered to users first using the following parameters:

- There are 3 Releases: Release 1 (MVP), Release 2, and Release 3 (All contained in the drop-down menus in **Column D**)
- **Release 1 (MVP)** can only have 10 user stories within the 3 sprints:
  - Sprint 1
    will have 3 user stories
  - Sprint 2
    will have 4 user stories
  - Sprint 3
    will have 3 user stories
- **Release 2** contains 2 sprints. The release can only have 10 user stories within the 2 sprints:
  - Sprint 4
    and Sprint 5 will each have 5 user stories
- **Release 3** contains 2 sprints. The release can only have 5 user stories within the 2 sprints:
  - Sprint 6
    will have 2 user stories
  - Sprint 7
    will have 3 user stories)
- All remaining user stories are assigned to "Backlog for Future Release"
- Assign the sprint that the user stories will be in within that particular release in **Column E**. If it is is "Backlog for Future Release" the user story doesn't need to be assigned a sprint.

Please note that because we are doing this project virtually certain rules and parameters are being given to make the project a beneficial learning exercise. When you are working with your real development team the number of user stories included in a release and or sprint will be based on the team's capacity and the high priority items that need to be completed.

1. Question if a user role has too much functionality to be in the MVP release.
2. Question what is a nice-to-have vs high-value based on the scenario and vision.
3. Question if you should trade off- what is the worst thing that could happen if the user had to wait for that functionality to come later.
4. Refine, rewrite, or delete a user story in the blue "User Story" tab if it doesn't fit into your Release or MVP plan

1.7 Step 5: Release Plan and MVP

Create a Release & MVP Plan in the **Release Plan & MVP Plan** tab in the **Minimum Viable Product Plan** spreadsheet template. 1. Copy your 10 **Release 1 (MVP)** user stories into column C, Rows 3-12 in the appropriate sprint. 2. Copy your 10 **Release 2** user stories into column C, Rows 13-22 in the appropriate sprint. 3. Copy your 5 **Release 3** user stories into column C, Rows 23-27 in the appropriate sprint.

1.8 Step 6: MVP Explanations and Justifications

Complete the following justifications in the **MVP User Story Justifications** section in the **Minimum Viable Product Plan Vision & Justifications** Document template.

1. Pick one user story in the MVP and justify why that user story belongs in the MVP
2. Pick one user story that is not in the MVP and justify why that user story doesn't belong in the MVP

Provide the following explanations in the **Sprint Scenario Responses** section in the **Minimum Viable Product Plan Vision & Justifications** Document template. 3. Provide a simple explanation for adjustments that the team would likely take given the following unexpected sprint interruptions. Click the following buttons to see those interruptions.

In **Sprint 1**, one of the user stories has a dependency on another system that won't be available until the next sprint. What should the team do about this user story?

In **Sprint 3**, A software engineer is sick and will miss most of the sprint. What should you and the team do to account for their absence.

Please note that story points here are being randomly assigned because of the virtual learning environment. As a member of a real-world scrum team the story point estimates would be decided by the team based on the level of effort, scope, complexity, and knowledge of the team. Estimation/story pointing is always done by the team in the real-world and not randomly nor by an individual team member.