PRODUCT MANAGEMENT

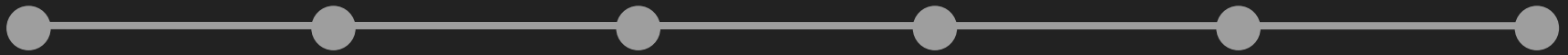
PDM 5.0

Documenting a Solution

GENERAL ASSEMBLY

Documenting a Solution

LESSON ROADMAP



Welcome + Warm-Up

Documentation

PRDs/Specs

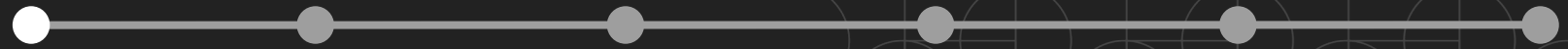
User Stories/Bugs

Acceptance Criteria

Bring It Home

WELCOME + WARM-UP

Documenting a Solution



Welcome + Warm-Up

Documentation

PRDs/Specs

User Stories/Bugs

Acceptance Criteria

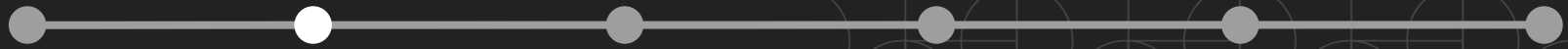
Bring It Home

LEARNING OBJECTIVES

1. Review and discuss Product Requirements Documents.
2. Document specifics of feature(s) for development and testing.
3. Differentiate different ways to document for development.

DOCUMENTATION

Documenting a Solution



Welcome + Warm-Up

Documentation

PRDs/Specs

User Stories/Bugs

Acceptance Criteria

Bring It Home

Documentation is a love letter that you write to your future self.

Damian Conway

WHAT IS PRODUCT DOCUMENTATION?

1. Documentation is a broad term that refers to the transcribing of information related to the description, functionality, development status and technical requirements of a product, feature or development team.

2. Documentation can help to structure information needed for prospective development or to memorialize information from a previous build.

3. Documentation is used among product teams, shared with stakeholders and, depending on the company or feature, customers/users.

TYPES OF DOCUMENTATION

Internal (shared within Product Team)

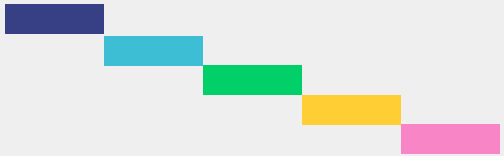
- ➔ Meeting Notes
- ➔ Wiki Pages
- ➔ Discovery Documentation
- ➔ Epics/User Stories

External (shared among stakeholders and colleagues)

- ➔ Product Requirements Documents (PRD)/Specs
- ➔ Epics/User Stories/Acceptance Criteria
- ➔ Bug Tickets
- ➔ Release Notes

DEVELOPMENT PROCESS vs DOCUMENTATION

PRDs are traditionally used in **waterfall development processes** where there isn't much iteration to the development work and the product's functionality is less likely to change.



- ➔ A finalized document is provided to development teams and they build what is documented.

In **Agile environment**, PRDs are shorter in length and are used more.

- ➔ Archiving the details of the product
- ➔ Providing a detailed overview that development teams can use to provide an estimate for development
- ➔ Knowledge sharing among other teams

Best Practices for Documentation: JUST DO IT.

DISCUSSION: Onboarding on the Fly

5 Minutes

Come off mute or type in the chat to share your thoughts and questions.

- Have you ever had to onboard for a new role or project without documentation?
- How did you get up to speed?
- What was your experience?
- What role do you think a documented overview of the role/project could have played?

BENEFITS OF DOCUMENTATION

The more notes you take, the better you become at taking notes and the less work you have to repeat. Effective documentation helps with:

- Knowledge sharing
- Preparation for the future meetings
- Reducing the need for additional meetings, offering teams an opportunity to communicate asynchronously

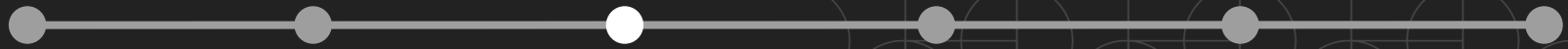
Documentation helps with onboarding and scale.

- Preventing key information from being siloed with an individual and/or lost
- Creating training references for new members of the technical team (product, design, development)

CHALLENGES		SOLUTIONS
People don't read it.	→	Do it anyway.
It isn't a one size fit all.	→	Communication has to be adjusted to your audience and the medium.
Delayed Gratification	→	You don't always see the immediate benefits - you and future members of your team will thank you in the future.

PRDs/SPECS

Documenting a Solution



Welcome + Warm-Up

Documentation

PRDs/Specs

User Stories/Bugs

Acceptance Criteria

Bring It Home

PRODUCT REQUIREMENTS DOCUMENTS (PRDs)

While roadmaps communicate a high level plan of the work that is planned for development - a more granular level of detail is required when working with design and development teams.

One of the methods for communicating with technical partners is to provide a PRD (Product Requirements Document).

A detailed document used to communicate the specifics, functionality and capabilities of a particular feature or product. These documents often include details on testing and launch plans.

PRDs are often referred to as specs and/or requirements.

ELEMENTS OF A PRD

Objective/Goal

Explain why are you building this and what do you hope to accomplish.

Release Plan

Phases/stages for the future launch

Features/User Stories

Include a description, goal and use case at a minimum. Additional details may be helpful or necessary depending on the complexity of the feature, such as out-of-scope items.

UX Flow & Design Notes

Mockups, wireframes; it can be used to describe the overall user workflow.

ELEMENTS OF A PRD

Analytics/Metrics

Success criteria; list of metrics that will be tracked for release; also may include any supporting analytical data

System & Environment Requirements

Which end-user environments will be supported (browsers, operating systems, memory, and processing power, etc.).

Assumptions, Constraints & Dependencies

List out what is expected of users, risks, limitations, dependencies and contingency plans

Future Work

An overview of next steps or plans for additional/related work

GROUP EXERCISE: PRD REVIEW

5 Minutes

Background:

AHA is a product development SAAS company - and they provide a PRD Template

As a group, each person take 3 minutes and write down your answers to the questions below.

Take 5-7 minutes to discuss/reflect with your group.

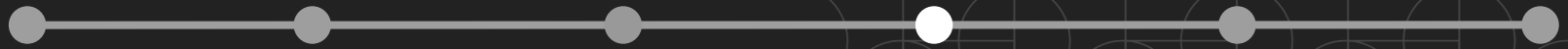
The remaining 5 minutes, we will discuss as a class and talk through any questions that didn't get answered or share overall thoughts:

- What questions do you have about the PRD Process?
- What thoughts do you have on the format/layout of the AHA Template?
- What sections do you think would be most helpful?
- Who do you think would most benefit from this document?

BREAK TIME

USER STORIES/BUGS

Documenting a Solution



Welcome + Warm-Up | Documentation | PRDs/Specs | **User Stories/Bugs** | Acceptance Criteria | Bring It Home

TWO PRIMARY TYPES OF TICKETS

Most organizations have a ticketing tool that tracks development work.

Jira

Asana

Trello

Why PMs create tickets?

To request new feature development (User Stories/Feature Requests)

To request a fix for current features (Bugs)

User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system.

Mountain Goat Software

USER STORIES

A type of feature request written with a very specific format

1. Broken down into pieces of business value that a team works to deliver during a specified development cycle, traditionally limited to 1 development cycle.

2. A method for communicating what task a user is trying to accomplish and why the task is important.

3. A way for a PM to explain the “why” rather than the “how” to developers.

USER STORY FORMAT

1. As a **{type of user}**,
2. I want to **{goal}**
3. so that I can **{reason}**.

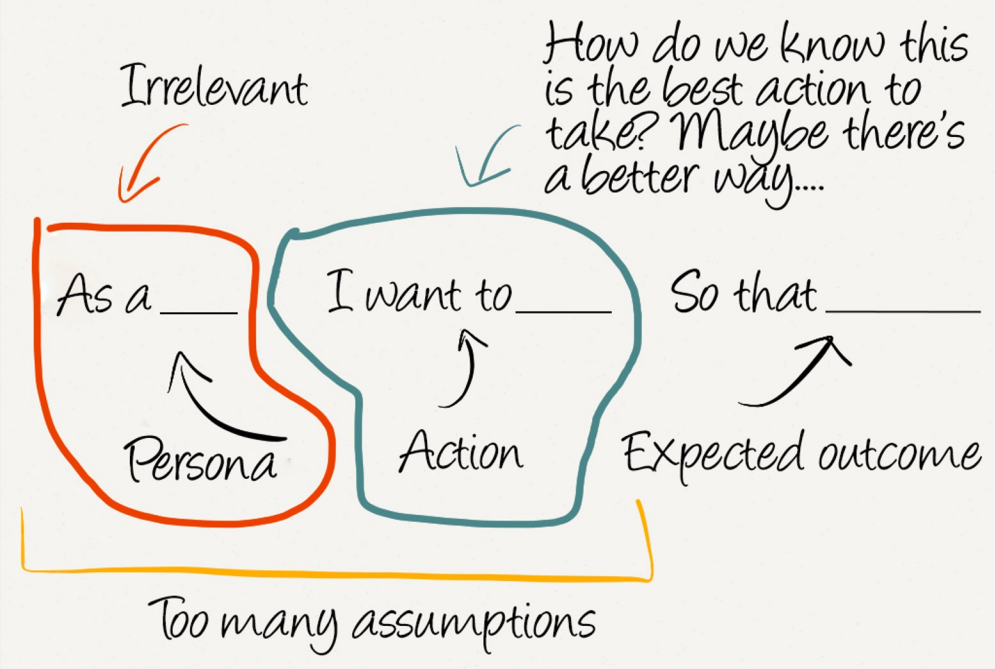
Who is this functionality for?

What should we create?

Why is it valuable to the user?

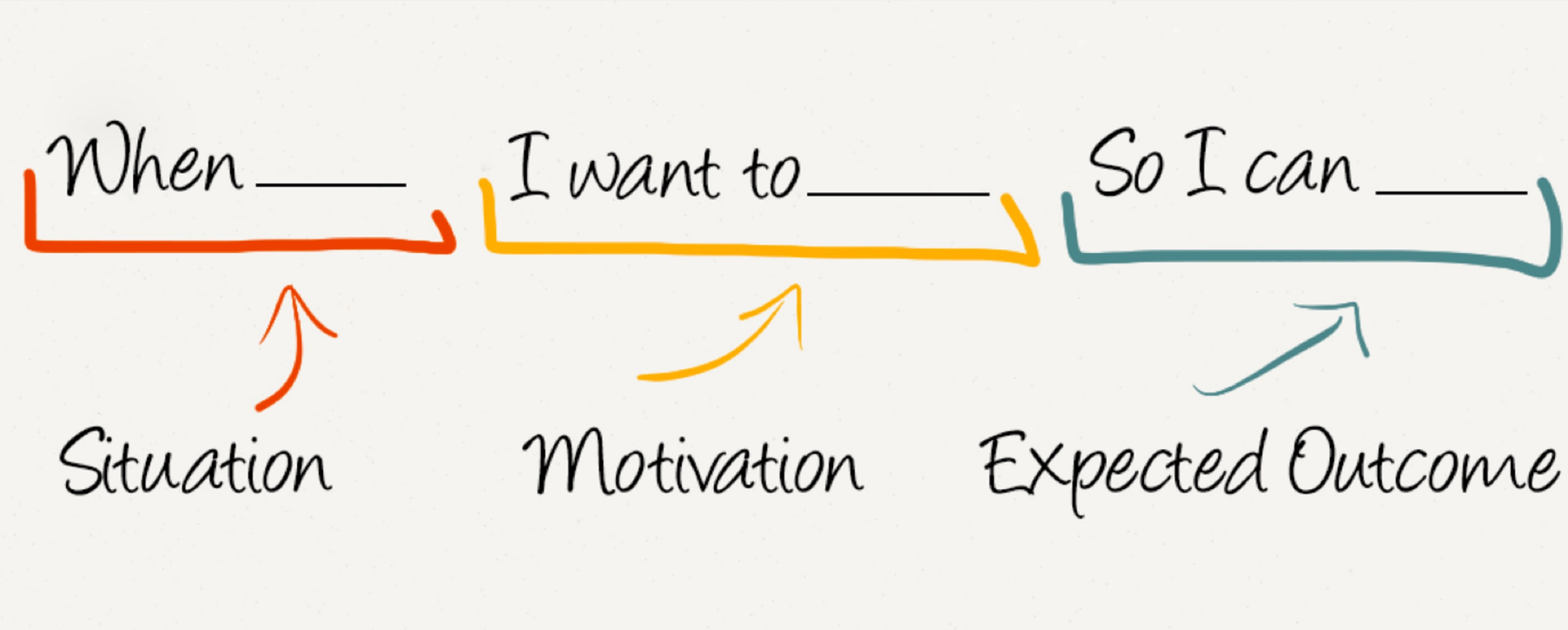
USER STORY FORMAT

USER STORY FORMAT - CRITICISMS



https://jtbd.info/replacing-the-user-story-with-the-job-story-af7cdee10c27

JOB STORY FORMAT



https://jtbd.info/replacing-the-user-story-with-the-job-story-af7cdee10c27

JOB STORY EXAMPLE

As a

(Training Supervisor in a large ski school)

When

(I am reviewing the performance of my training programs)

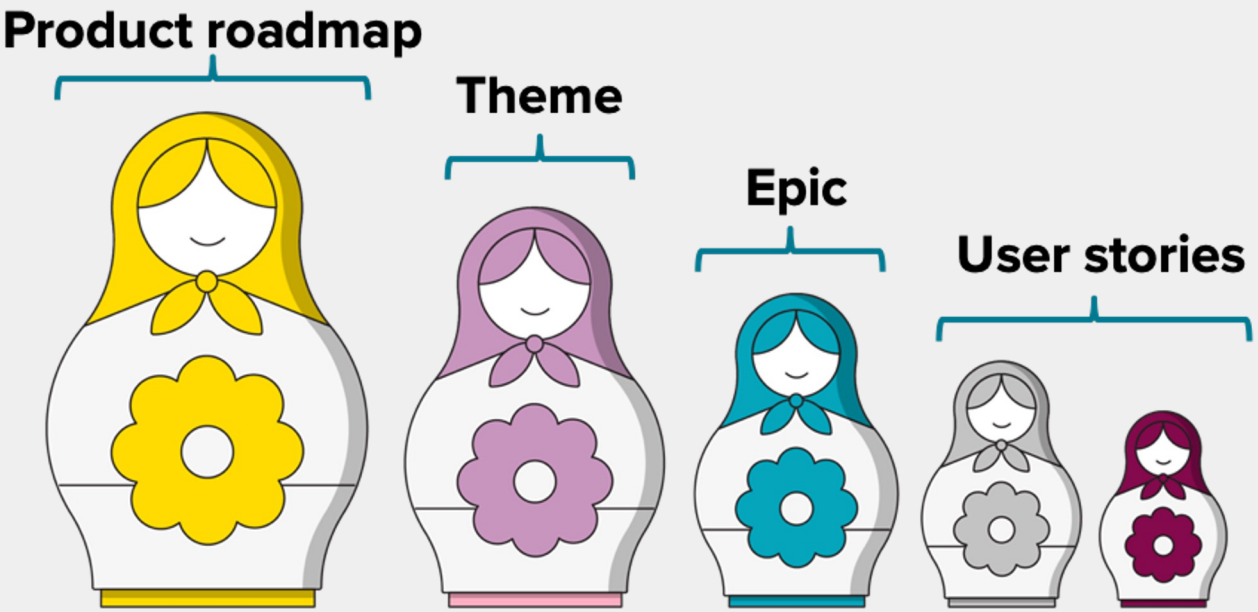
I want to

(Have up to date, detailed attendance and performance metrics for all of my programs)

so that I can

(Make sure I am maximise the use of my Trainers for the programs, topics, clinics and timeframes that are most valuable.)

HOW IT ALL FITS TOGETHER



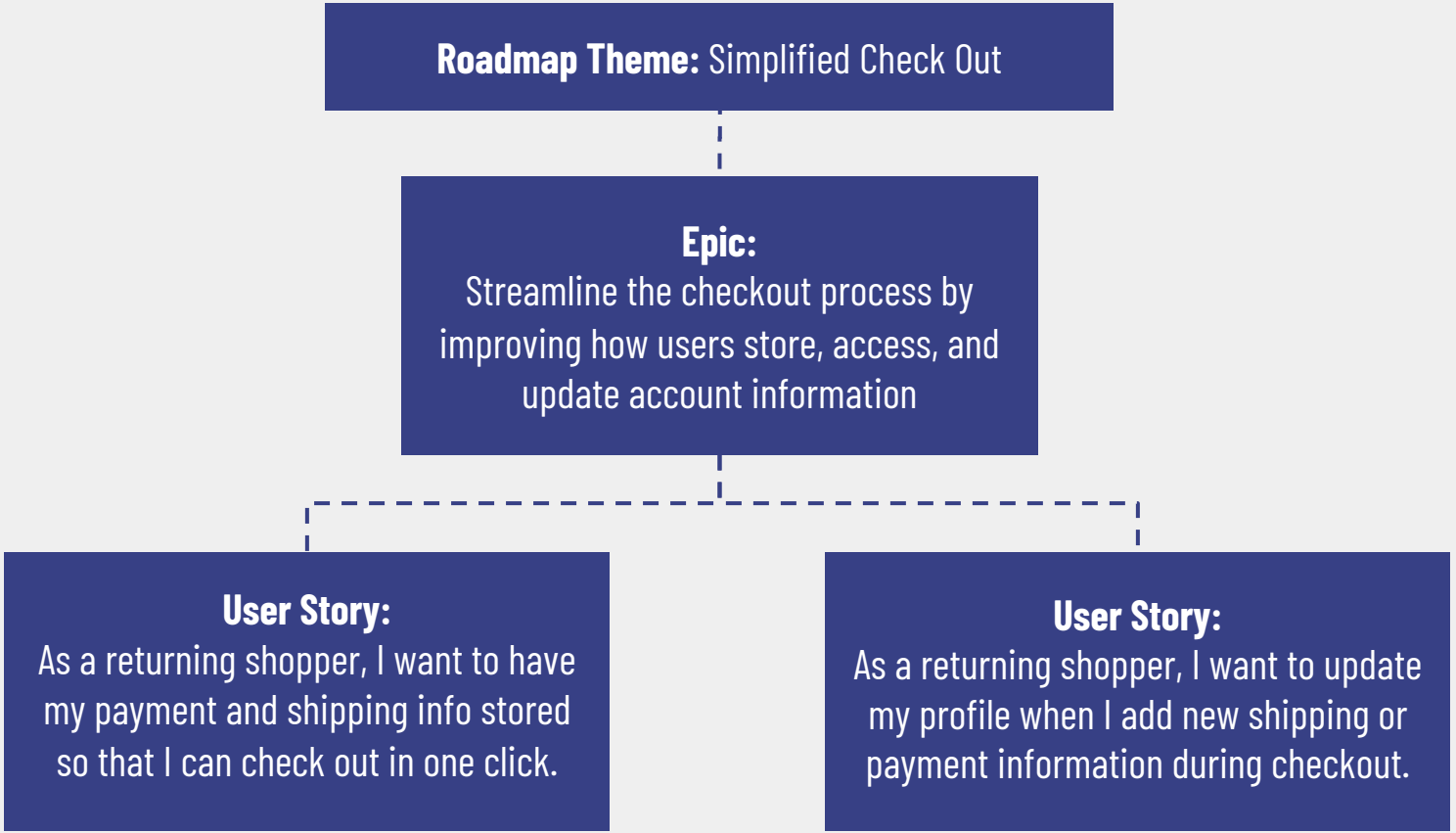
EPICS

A larger body of work that extends beyond multiple development cycles

Comprised of multiple user stories; the epic is completed when all user stories are released.

May represent one feature or several

Roll up into a Roadmap Theme.



10 Minutes

Referencing the following features, pick 5 features from the list below and write a user story (for each) that captures what the user needs from that feature and why it's important to them.

- Lyft Split Pay
- Auto Save Username/Password Functionality
- Biometric Phone Access
- Instagram Reels
- Siri
- Zoom Breakout Rooms
- Gmail Snooze Functionality
- Twitter Direct Messages
- Amazon "Buy Now"
- LinkedIn Emojis

User Stories are written to build; Bug Tickets are written to fix.

BUGS

Having varying levels of priority; ranging from non-critical to severe

➔ Severe usually indicates an issue where revenue/usage is impacted.

Tickets created to report inconsistencies or error that are appearing in the system.

High severity bugs take precedence over all feature development that's in progress.

Most non-critical bugs are triaged by PMs where they will work with developers to determine when to address.

Some bugs are never fixed.

ELEMENTS OF A BUG REPORT

1. **Title**
   Overview of the issue

2. **Severity/Priority**
   Usually preset by the tech org; 0-5

3. **Description**
   Detailed description of what is occurring

   - ➔ Expected Result
   - ➔ Actual Result
   - ➔ Steps to reproduce

4. **Environment**
   Name of the system, browser, login name, etc

5. **Attachments**
   Screenshots, videos, text

EXAMPLE OF A BUG REPORT

Title:
Items are Being Erased from Cart

Severity/Priority:
0 (critical)

Description:

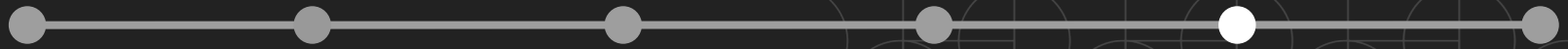
- Once a single item is added to the cart, the order insurance is automatically applied (via Route). When selecting additional items and attempting to check out, the cart begins to automatically refresh and removes all items from the cart except the insurance
  - If the insurance is removed before the second item is added, this refresh does not occur
- Expected Result: a customer should be able to complete their order with insurance
- Actual Result: when insurance is added, all items are being removed from the cart

Environment:
Site page; using chrome; cookies/cache cleared

Attachments:
Screenshots, videos, text

ACCEPTANCE CRITERIA

Documenting a Solution



Welcome + Warm-Up

Documentation

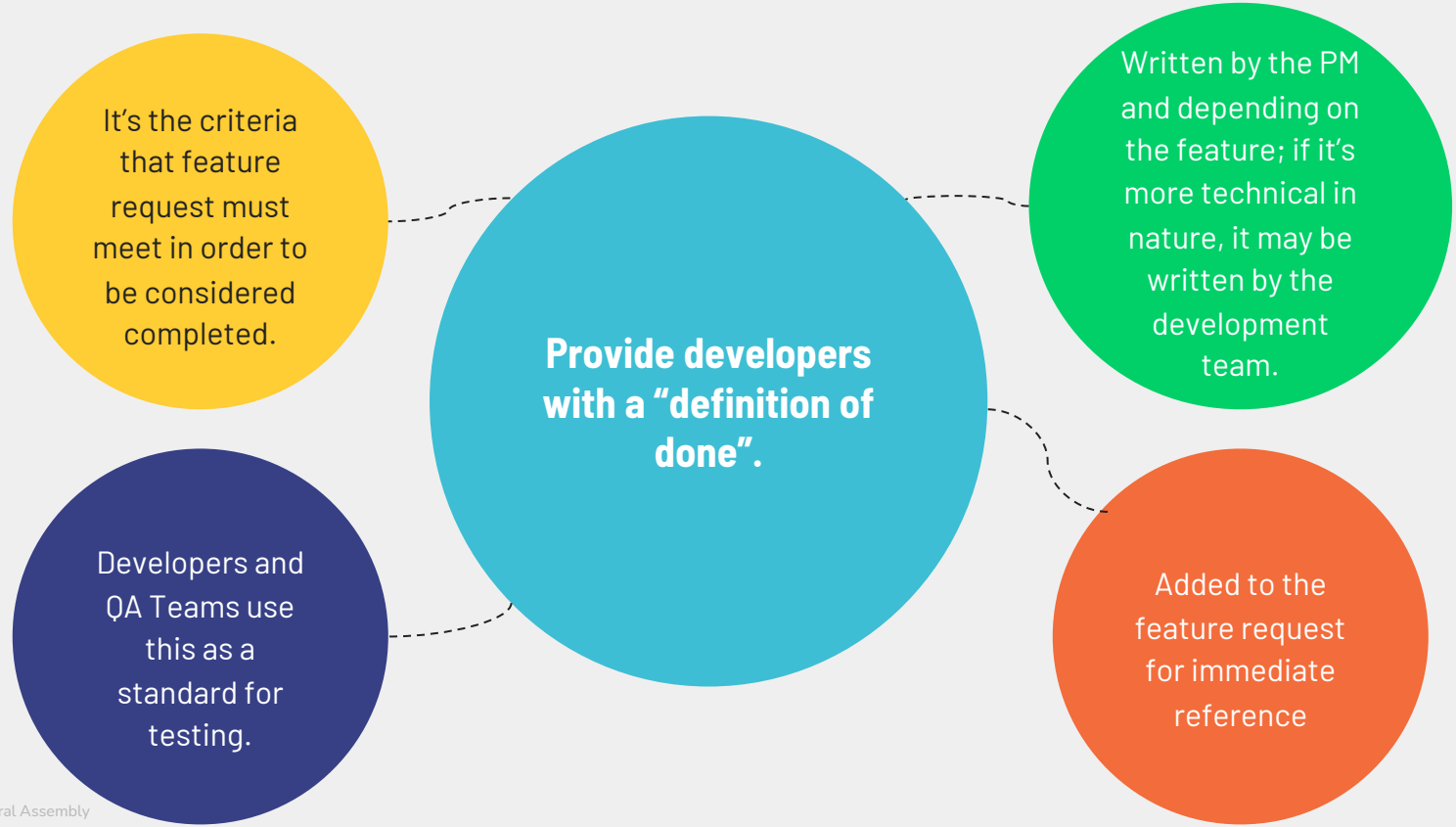
PRDs/Specs

User Stories/Bugs

Acceptance Criteria

Bring It Home

ACCEPTANCE CRITERIA



GOOD ACCEPTANCE CRITERIA ARE...

1 **Clear**
If your team can't understand it without asking you, they can't enforce it.

2 **Testable**
You need to be able to design a test for it. It can't be hypothetical.

3 **Pass / Fail**
The work must either pass or fail. Yes or no. No ambiguity or partial completion.

4 **Outcome, not Output**
Focus on the outcome needed, not the method for *how* something will be solved.

5 **Specific**
Be specific as possible.
There's a big difference between ("pages load fast") vs ("Pages load in less than 3 seconds").

GOOD ACCEPTANCE CRITERIA

User Story

As a returning shopper, I want to have my payment and shipping info stored so that I can check out in one click.

Acceptance Criteria

PayPal, Google Pay, Apple Pay, and all major credit cards can be used to complete the transaction.

Acceptance Criteria

User is prompted to log in if they aren't already.

Acceptance Criteria

If the user does not have default payment and shipping information, they are prompted to select a default from a list of past choices.

BAD ACCEPTANCE CRITERIA

User Story

As a returning shopper, I want to have my payment and shipping info stored so that I can check out in one click.

Acceptance Criteria

When a logged-in user adds an item to their cart, the “one-click checkout” button appears.

Acceptance Criteria

Payment information must be stored securely.

BAD ACCEPTANCE CRITERIA

User Story

As a returning shopper, I want to have my payment and shipping info stored so that I can check out in one click.

Acceptance Criteria

When a logged-in user adds an item to their cart, the "one-click checkout" button appears.

You are designing the solution and telling them HOW it should work.

Acceptance Criteria

Payment information must be stored securely.

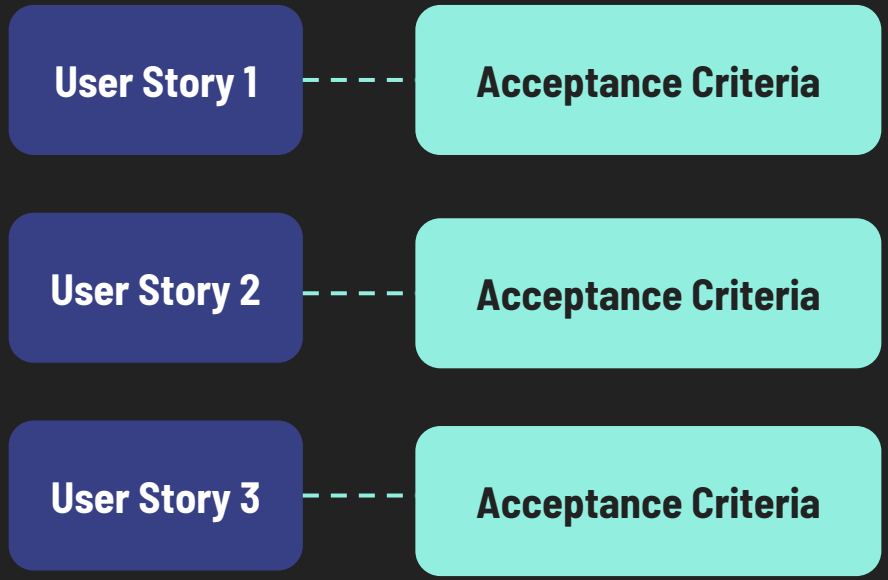
Define "Securely"!

"Abide by standard 10.3... uses only X storage practices...

PARTNER EXERCISE: WRITE ACCEPTANCE CRITERIA

10 Minutes

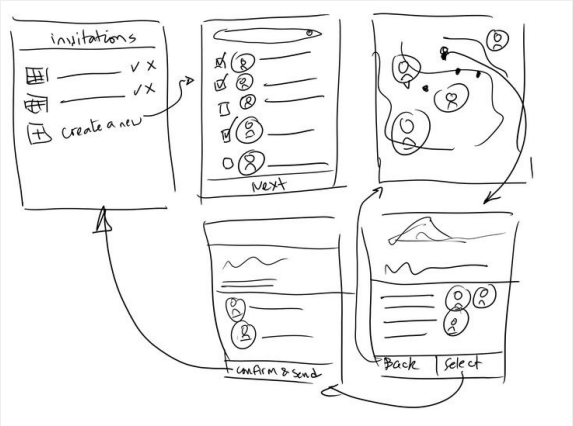
Given the user stories you previously developed, with your partner, write one acceptance criterion for each user story you have.



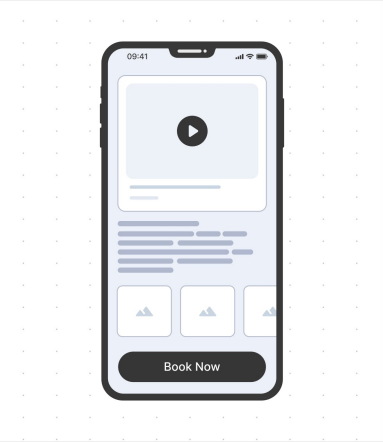
HIGH FIDELITY PROTOTYPES > ACCEPTANCE CRITERIA

What is better than a long written list of “You must do X in situation Y” acceptance criteria?

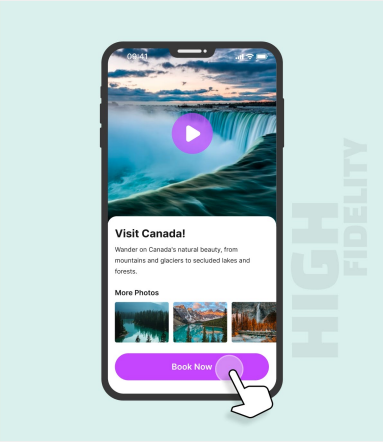
A HIGH FIDELITY, VISUAL PROTOTYPE THAT SHOWS YOU EXACTLY HOW IT SHOULD WORK!



SKETCH



LOW FIDELITY WIREFRAME



HIGH FIDELITY WIREFRAME

MARTY CAGAN SAYS SO...

“Here are what I consider the requirements for a good and useful product spec:

- *the spec must describe the full user experience – not just the product requirements but also the user interaction and visual design. By now hopefully everyone recognizes how closely intertwined the requirements are with the design.*
- *the spec must accurately represent the behavior of the software – and we need to acknowledge that words and pretty pictures are just too limited in their ability to describe this behavior.*
- *there are several critical consumers of the spec – engineering, QA, customer service, marketing, site operations, sales; as such, the spec needs to communicate the behavior of the product in a way that all of these groups get what they need.*
- *the spec will change – the rate of change should slow down dramatically once engineering gets started, but there will be decisions and issues that arise, and the spec should change to reflect the very latest decisions.*
- *there are a number of artifacts in the creation of a spec, such as lists of prioritized requirements, wireframes, and mock-ups, but there needs to be a single master representation of the spec, to minimize confusion, ambiguity and versionitis.*

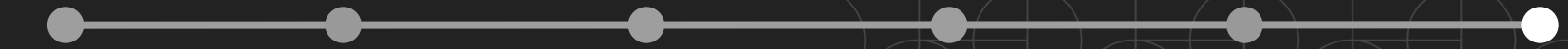
In my mind, there’s only one form of spec that can deliver on these requirements, and that is the high-fidelity prototype.”

- Marty Cagan

https://www.svpg.com/revisiting-the-product-spec/

BRING IT HOME

Documenting a Solution



Welcome + Warm-Up

Documentation

PRDs/Specs

User Stories/Bugs

Acceptance Criteria

Bring It Home

KEY TAKEAWAYS

Documentation is Key

The best thing is to just do it - your future self will thank you

Document for Your Respective Audience

Internal: within product org
External: stakeholders and colleagues

Documentation for Development

The following are referenced by engineers, designers, stakeholders for development:
PRDS, User Stories/Bug, Acceptance Criteria, Release Notes

Additional Resources

Practice Again	Digging Deeper
Writing Agile Stories • Agile Epic vs. User Story: What's the Difference? • How to Write Good User Stories in Agile Software Development	**Writing Good Stories** • 10 Tips for Writing Good User Stories • Useful Tips on How to Write Greatest Epics


GA
GENERAL ASSEMBLY