W4156

Requirements Engineering I (via PRD)

Agenda

- ❑ Importance of Requirements
- ❑ Requirement Engineering
- ❑ Defining a Product Vision: 5 questions

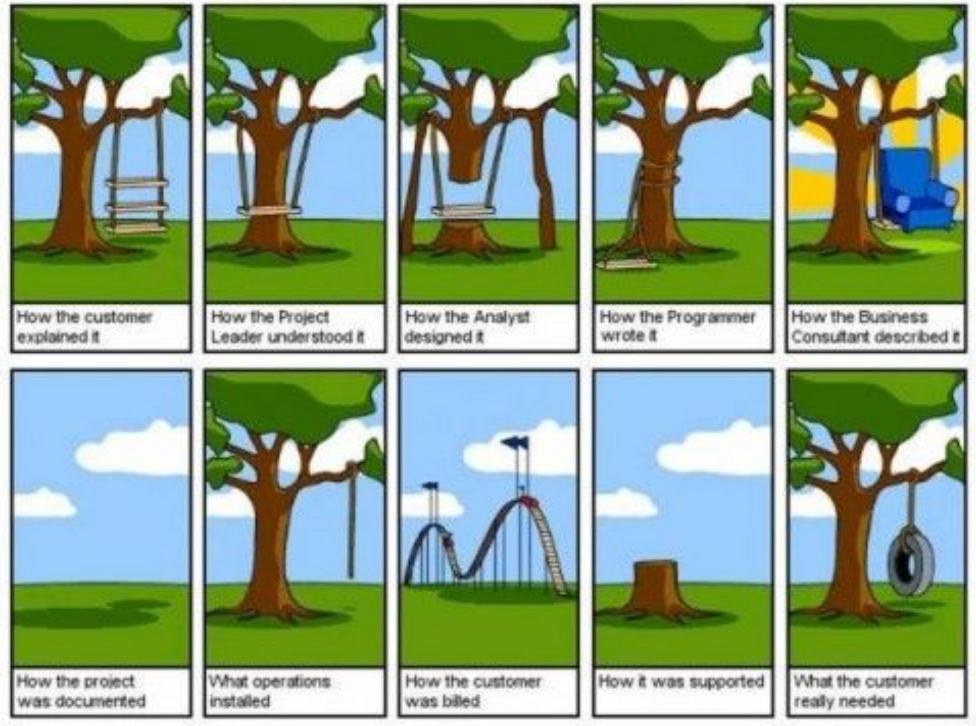
By the end of the next two lectures you will be able to write a very decent ‘Product Vision’ or ‘Product Requirements Document’

Requirement Engineering

“Are we building the right thing?”

versus

“Are we building it right?”



Importance of Product / Requirements

- **Poor requirements are the leading source of project failure**
- **The cost of fixing issues grows exponentially the later they are detected**
- **Most startups fail because they failed to fix a problem for someone**

“It's the same with technology. You hear all kinds of reasons why startups fail. But can you think of one that had a massively popular product and still failed?

In nearly every failed startup, the real problem was that customers didn't want the product. For most, the cause of death is listed as "ran out of funding," but that's only the immediate cause. Why couldn't they get more funding? Probably because the product was a dog, or never seemed likely to be done, or both.” [Paul Graham]

What are we building?

You need to understand requirements in *some form* for software

1. **For an ‘external’ customer**
2. **For an ‘internal’ customer**
3. **If ‘you’ are the customer**

Irrespective of whether you are

a. **A small company**
b. **A large enterprise**

The purpose is not to create the document.

- **The purpose is to explore, create clarity, consensus and shared understanding**
- **The document is an unambiguous / crisp memorialization of the conversation**

Product Manager

We will discuss what methodologies and roles later. For now let's say there is often a person on the team called the product manager:

> *"The product manager has two key responsibilities:*
> 1. *Assessing product opportunities*
> 2. *Defining the product to be built"*
>
> [Marty Cagan - Inspired]

We are going to cover *requirements engineering* from the perspective of a product manager.
One way they often define the product to be built is via a 'product requirements document'
We will use the contents of this doc to explore various requirements engineering techniques

Product Vision in 5 simple questions

1. What is the ***value*** in building this / ***why***?
2. ***Who*** is it for and what are their ***goals***?
3. ***What*** must they able to ***do?***
4. How would it ***look/work***?
5. Which features are ***most important?***

Do we agree these are pretty sensible questions?

Anecdote/Variation: In the early days of Amazon they wrote the *press release* at *the start of a project*.
It hits many of the same goals/purpose. What is the product and value? What does it help people do and some product documentation.

Product Vision in 5 simple questions

Note:

A. Does ***not*** detail ***technologies*** or ***design***
B. Does ***not*** attempt to provide ***accurate dates*** or a plan
C. Dreaming partially allowed: needs to be implementable but speculate

- Technologists find A, B *hard* (they find C easy for some reason!)
- We spend N years *learning engineering* by solving *provided problems*
- Next M years in industry are spent *unlearning the reflex* to *build first*
- My boss' favorite question: "what are you trying to achieve?"

Q1: Value

Why spend time building this?

- Revenue Opportunity, Revenue Protection, Cost Save, Control, Regulatory

Simple description:

1. The world works this way and it is a pain (causes these inefficiencies)
2. The product we are building does X
3. It solves the pain and provides this value

Q1: Value

Bagels are a staple of the NY saturday morning. However, bagel shops experience peak usage within a compressed time period (10-11:30). Lines grow long, orders are often captured incorrectly and busy manhattenites have long waits in crowded shops. This situation limits Bagel shops revenue during peak hours and is a poor customer experience. Unfortunately, a delivery model does not work well for bagels as toasted bagels must be consumed within minutes.

BagelRun™ solves this problem by pairing consumers and their local bagel place. Using a smart phone app users enter an order in advance or when they are en route to the bagel shop. Using geo-location, estimated travel time and preparation tie the app instructs the bagel place to begin preparation at the correct moment to minimize wait time.

The advantage are lower wait times, less crowded shops and more accurate orders for the consumer. The advantage for the bagel shop is increased capacity and revenue at peak periods, establishing routines/loyalty with frequent customers, lower staffing to capture orders and higher order accuracy.

Q2: Who

User Persona: *“represent a class or type of user of a specific interactive product. A persona encapsulates a distinct set of behavior patterns regarding the use of a particular product” - Alan Cooper*

Personas

Persona Name	<real or synthesized>
Profile	<description of individual, background. Anything which is *relevant* to understanding them and how they may need/use the product. Technically sophisticated, Smart phone owner, Habits, Income, Frequency of Use, Location, etc, etc>
Goals	Many types (first is our focus) 1. End goals (what they want to do) 2. Experience goals (how they want to feel) 3. Life goals (user wants to be)

Persona Example

Persona Name	UWS Local: Troy McClure
Profile	30s, professional, married with young children. Smart phone owners and comfortable ordering / paying in-app and with geo-location. Some concerns over data privacy. Does a bagel run ~2 times a month but no set timing. Average spend 26$ on two Nova Scotia and two coffees. Lives 1km from bagel shop
Goals	End goals: - Accurate bagel order. Still hot and toasted Experience goals: - Bagel runs are an enjoyable experience. Scotty knows his local staff and does not want a ‘fast food’ or transactional experience.

Personas

Commentary:

- Does anything we learned in persona change/prioritize required features?
- Does Troy have different requirements than a NY tourist?
- Given it is an ‘experience’ purchase does it change our app feel?

As I started to write other personas:

- Are the bagel shop owner, cashier and line chef different personas?

Two ‘litmus tests’

1. Did the 10 minutes writing that persona generate any questions/insights?
2. Do we feel it was time well spent?

Personas: Lets Try

Break out for 4 minutes with people close beside you:

1. Who are the personas for AirBnB? (hint: suggest there are way more than two)

2. Does a product *need* to *solve* for all personas?

Q3: What can they *do*?

User Story: As a <type of user> I want <feature> so that <goal>

Note: User stories are one of many requirement capture techniques.

User Story Example

User Story: As a <type of user> I want <feature> so that <goal>

As a BagelConsumer I want to browse the menu so I can decide what to order

As a BagelConsumer I want to be able to enter my order so I can ‘stage’ it in advance

As a BagelChef I want to receive tickets of Bagels to prepare so I can prepare them in right order/time

As a BagelShopOwner I want to have a report of sales with pivots of time, product, revenue so I can optimize my business

Note: We are probably papering over a lot of complexity for the moment - that is ok. We are still working on product / features

‘INVEST’ in Good User Stories

Characteristic	Meaning
Independent	Self contained / no dependency on another (we will prioritize independently)
Negotiable	They are not contracts. Leave space to negotiate the scope and interpretation
Valuable	Someone must say “I want this” (see MoSCoW)
Estimable	You must be able to estimate the effort to produce
Small	Appropriate amount of work. We want to be able to estimate and iterate
Testable	We must be able to test and verify it is done

Lets Try it Out

Another 4 minutes breakout. Can you give me a few user stories for AirBnB?

- …
- ...

Elicitation

- At some point we may have a customer to interview / get requirements
- Why is it called elicitation rather than ‘gathering’?
  - (Elicitation is a euphemism for forcible extraction)
- Elicitation is entire sub-discipline (interviews, workshops, etc)

Elicitation

- The simplest is to ask ***open*** questions and ***listen****
  - Tell me how the business currently works and pain
  - Walk me through the simplest scenario of how you want it to work
  - Note the use of open questions. We should be *listening and guiding*
  - <Then drill down>
  - User stories, BPMN, Prototypes (30 mins) are forms of communication

* When you do it well the ratio of talking/guiding to listening is 1:20

Q4: Show me

- Teams would work for a month (or more) to define ***textual*** requirements then go away and build
- Everyone *thinks* they are on the same page

Team proudly present to the user.

- User (being direct):
  - “This isn’t what I wanted”
- User (being nice / recognizing two different understandings):
  - “This isn’t quite what we need. I thought the workflow was going to work like X”
- User (recognizing this is what they asked for but now they see it not what they want):
  - “This is what we originally agreed but now I see it - this won’t work. It needs to work like Y”

Q4: What happened?

- You will shortly find ***building*** software is expensive/time consuming
  (you may also find you want to do something else with your life*)
- Textual descriptions are difficult for users/stakeholders to visualize / give feedback
- Some of the issues are *product issues* not engineering (what not how)
- Lower ***fidelity*** requirement formats enable different interpretations

* My lab partner in first year engineering quit after one semester to become a blacksmith making swords for LARPers. Production process included getting them enchanted by a wizard

Q4: Show me

The *partial*** solution is to:

- Produce a low cost visual *prototype* of the product
- Walking through and seeing a visual prototype is *vastly* more effective than textual descriptions
  - Flush out product issues (how it should work)
  - Get user, engineering and product on the same page
- Of course building a prototype takes time. However, we hope it is net positive

$$\text{Time}_{\text{Building Prototype}} + \text{Time}_{\text{Building Right Thing}} < \text{Time}_{\text{Building Wrong Thing}} + \text{Time}_{\text{Rework}}$$

** Partial solution in that it decreases but does not eliminate the issue

Prototyping

- There are a number of different prototyping methods & tools - wireframes to storyboards
- However, these days the prototype tools (that allow drag and drop creation of apps) are very quick
- I have arranged free licenses for proto.io for students for the duration of the course

Q4: Prototyping

Summary

- Visual prototyping tools help flush out product questions and establish common understanding
- You can have an ‘executable’ version of your app very quickly and agree with the user
- Visual prototypes do take *time* to produce. However, the time spent on the prototype eclipses building the wrong thing and rework.
- Visual prototyping also help the product person capture and express the requirements *without* them having the ability to code/heavy engineering.
    - My personal experience is this aids productivity
    - Product sometimes change their mind or revert to expressing requirements verbally
    - I say “Great - can you express that requirement back into the prototype”

Q5: Which features are most important?

- The excitement at this stage is generally palpable. People love envisioning the product.
- The conversation will turn to “when will it be ready?”
- *Resist the urge/pressure to throw out a date at this stage*
    - You will be held to it (and we are throwing out guessed plucked from thin air)
    - We only knew *what* needed to be built 5 minutes ago?!?! How could we have worked out *how* to build it and therefore *how many resources* and *how long?*
    - Product is generally so large that we don’t want to build it in ‘one go’ *anyway**

* for multiple reasons we will explore in later lectures including a) if we can deliver a subset of the functionality earlier does that provide value and b) as we start to deliver we will learn and change the product direction anyway

MVP

Minimum Viable Product (MVP): is a concept you may of heard of. At it's core, it says "lets not build everything but build a smaller version more quickly that provides value. We can use this to get feedback quickly and adjust (or stop!)"

Used loosely in industry

A) Warning: Customers use MVP to mean "everything I want" (defeats purpose)

Even within technology there are two definitions

B) "I have long defined minimum viable product as the smallest possible product that has three critical characteristics: people choose to use it or buy it; people can figure out how to use it; and we can deliver it when we need it with the resources available – also known as valuable, usable and feasible." [Marty Cagan]

C) "the minimum viable product is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort" [Eric Ries]

- 'C' is obviously weaker/less effort than B. 'B' is perhaps more common in industry
- My preference for startups/you is 'C' because 'C' is quicker feedback
- During the project you should control scope / aspiration to hit something 'viable'

Product Roadmaps

	Demo	MVP	MMP	Grand Vision
Goal				
Theme 1				
Theme 2				
Theme 3				

(Many roadmap templates and methodologies exist. Given limited time & context I have de-emphasized commercial aspects of roadmap planning)

Product Roadmaps

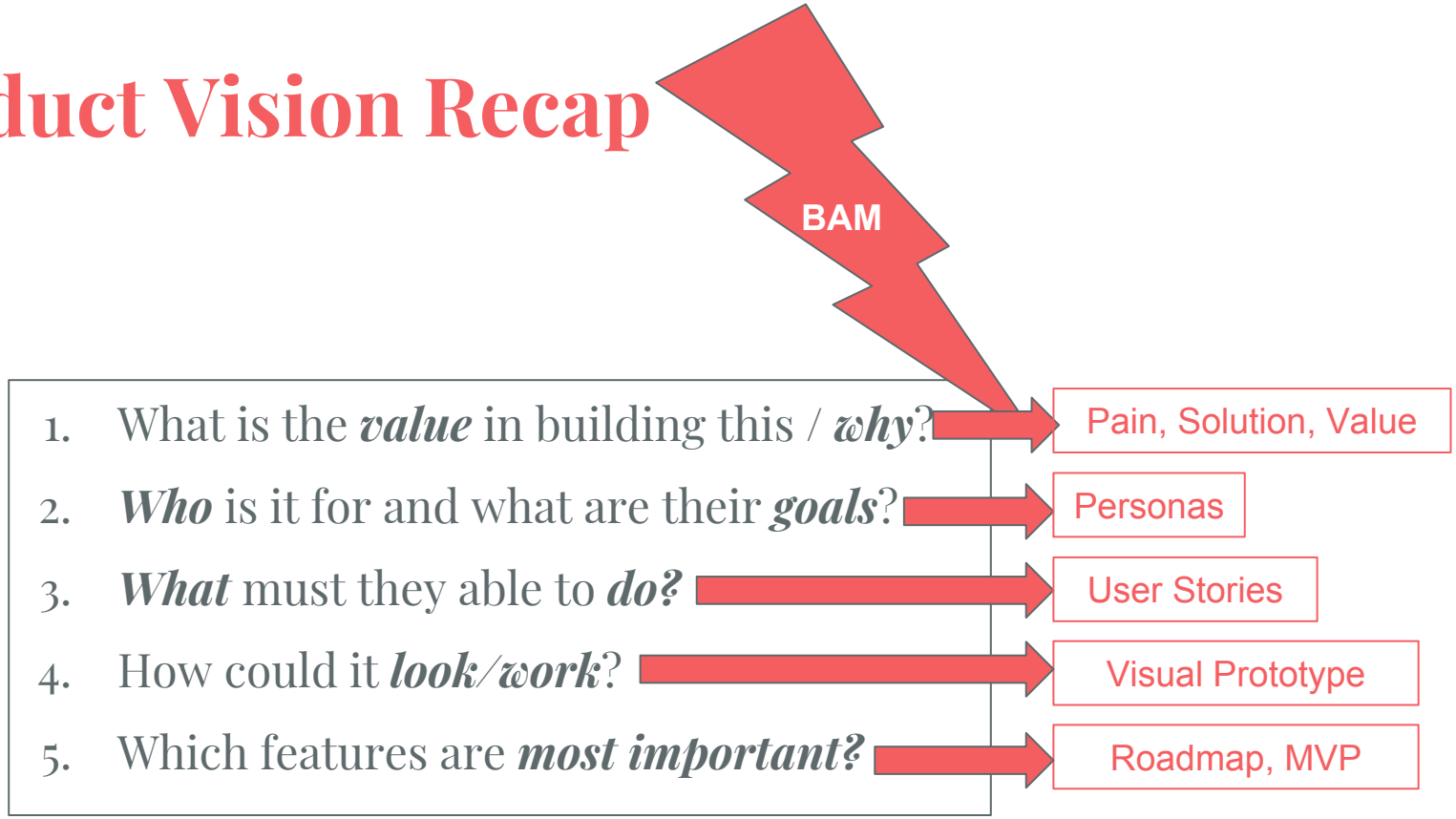
	MVP	MMP	Grand Vision
Goal	Basic end to end journey with basic time estimation	More sophisticated journey estimates and order management	...
BagelBuyer	- Email Login - Menu Browse - On demand orders	- OAuth - Scheduled Orders	...
BagelShop	- Prints order with estimated arrival immediately - Capture late/early metrics	- Releases order at correct time - In app payment	...
Geolocation	- Basic departure (assumes walking)	- Readjusted order release time based on travel - Mode of transport detection	...
Platforms	- iPhone	- Android	...

Product Vision Recap

We agreed the below 5 were reasonable questions to understand

1. What is the ***value*** in building this / ***why***?
2. ***Who*** is it for and what are their ***goals***?
3. ***What*** must they able to ***do?***
4. How could it ***look/work***?
5. Which features are ***most important?***

Product Vision Recap



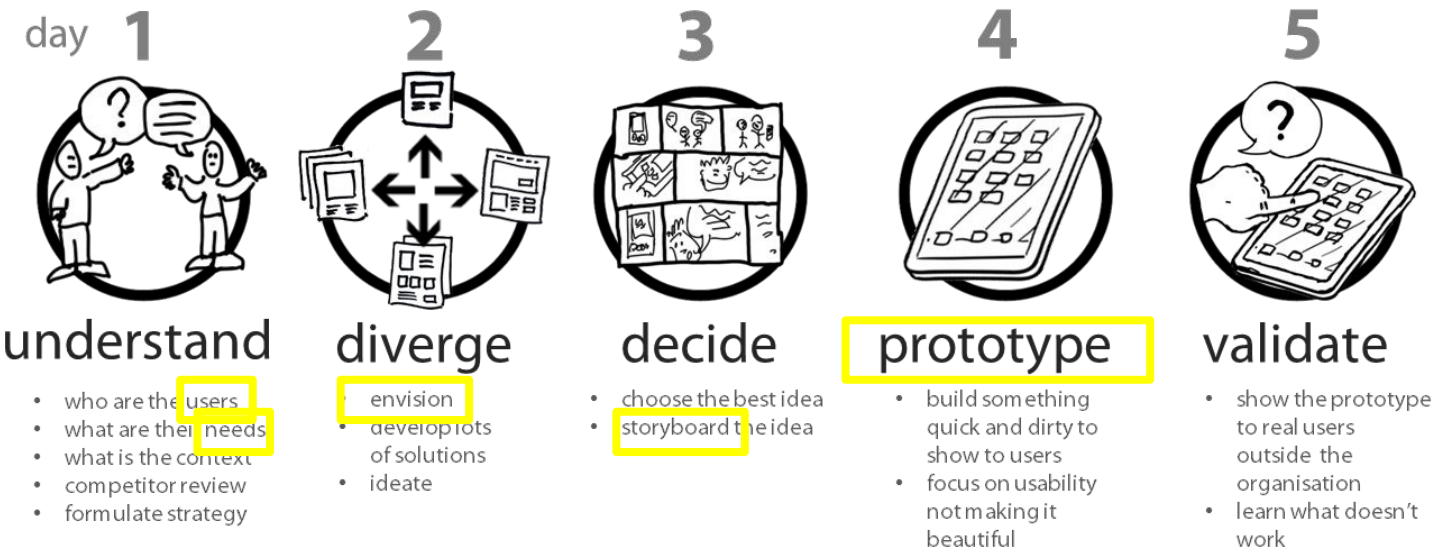
Process

- We understand the *contents* of a basic product vision document
- But how does a product vision document get created?
- Does it magically flow forth from the fingers of a product manager in a caffeine induced haze?

http://www.imdb.com/title/tt0315327/

Design Sprint

- Google *design sprint* one example of design process to solve problems/devise products
- We should be very comfortable with the techniques (yellow)
- We will link this process to theory later in the course (agile, envisioning, etc)



https://medium.com/i-want-to-be-a-product-manager-when-i-grow-up/the-design-sprint-92f61b18fb72

Recap / Breathe!

- We started by emphasizing the importance of ‘what’ vs ‘how’
- Product management is a role, activities, artifacts and culture part of many software teams
- Product vision is often the start of a new product process where we establish ‘why’ & ‘what’
- Basic product vision format (5 questions) and covered techniques to address each stage
- If you can gain a good grasp of these techniques and produce a product vision in this format it will be better start than ~60% of the requirements I have seen
- There are many requirements engineering techniques. However, if you grasp the core concepts you will recognize the overlap / essense

Pop Quiz

Question	Answer
5 Questions in PRD	**Supporting Techniques**
How do I extract information from users?	
Who is it for and their goals?	
What must they be able to do?	
How could it look/work?	
Which features are most important?	

Pop Quiz

5 Questions in PRD	Technique
How do I extract information from users?	Elicitation (spans all)
Who is it for and their goals?	Personas (inc goals)
What must they be able to do?	User Stories
How could it look/work?	Prototype / Wireframe
Which features are most important?	Feature Roadmap & MVP

Reading

Material	Optionality
Beginning S.E. Chapter 4 (across this and next lecture)	Required
PRD Guide	Required
MVP	Required
INVEST	Required
Templates	Recommended
Burton (first 5 mins prototyping)	Optional
Career advice - read this cover to cover	Optional

Homework

- I will not collect but practice the techniques
  - Pick an existing company
  - Sketch each of the 5 sections of a product vision document
  - You do not need to exhaust each section but write the value, 2-3 personas, 2-3 user stories each, sketch the most basic storyboard and identify the absolutely critical features

Project

- When you form teams the first thing you will do is write a Product Vision
- I recommend you block out a chunk of time and follow a compressed design sprint
- I have arranged for free licenses to proto.io which is an industry prototyping tool
  - You are free to use this or balsamiq / other