BITS Pilani presentation

BITS Pilani
Pilani Campus

K.Anantharaman
kanantharaman@wilp.bits-pilani.ac.in

innovate achieve lead

BITS Pilani
Pilani Campus

SE CZ 544 , Agile Software Process
Module-5 – Prioritization Techniques and Planning with User Stories

innovate achieve lead

BITS Pilani
Pilani Campus

Why Do We need to prioritize Product Backlog?

Why Product Backlog Prioritization Matters?

innovate achieve lead

- Prioritization is part of product backlog grooming.
- It directs the team's work by focusing the team on the most important items – that are on the top of the list.
- As items are detailed according to their priority:
  - Flexibility is built into the process and allows delaying decisions about the lower priority items, buying you more time to evaluate options, gather feedback from customers, and acquire more knowledge.
  - This ultimately results in better decisions and a better product.

Source:: https://www.romanpichler.com/blog/prioritising-the-product-backlog/

What are these factors to be to considered when prioritizing?

innovate achieve lead

- **Business value:**
  - What is the actual vale ($$$) that I'm going to get out of releasing this into the market?
  - Most valuable items first. But what makes a product backlog item valuable?. An item is valuable if it is necessary for bringing the product to life
- **Return on investment (ROI):**
  - Cost of building a feature compared to other features of the same value
  - Cost delaying this feature
- **Feature grouping:**
  - Releasing a group of features together compared to realizing independent features
- **Risk & Dependencies**
  - Things that we don't know, If this is going to pan out or not
  - Two dependent items may have to be implemented in the same sprint, for example

Prioritizing Product Backlog/Work …

innovate achieve lead

- **Releasability**
  - If you are uncertain about if and how a feature should be implemented, then early releases can answer this question. A partial implementation is often sufficient for early releases.
- **Architectural runway:**
  - There might be code spikes, and things that we have to build so that we can actually build functionality on top of them
  - API Versions.
- **Regulations & Compliance:**
  - In regulated industry, and we have to be compliant with certain things. If we are not compliant by this date, we could have big repercussions on my company.
- **Organization Politics:**
  - Pet projects, Customer preferences
- **Knowledge** Creation, Need to consider Users, Customers, other Stakeholders

Defining Business Value

- **Increase Revenue**
  - Experimenting new features and learn from feedback.
- **Protect Revenue**
  - Competitors have these features. We actually may lose revenue by not having this new competitive advantage
- **Reduce Cost**
  - Doing something internally, Eg. Automate something to reduce cost.
- **Avoid Cost**
  - Adhering to regulator compliance. Otherwise fine us.

Start Prioritizing big items

- Individual product backlog items can be very small and therefore difficult to prioritize.
- It’s useful to prioritize themes and epics first and open it up to optimize release contents.

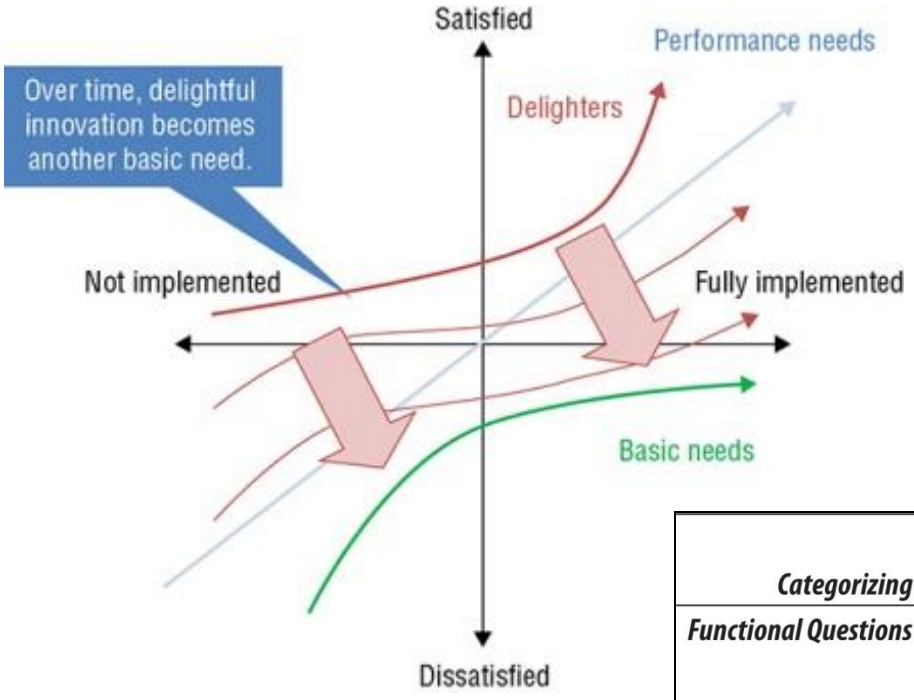
- We then prioritize the items within and, if necessary, across themes.

Approaches to Prioritization

- Customer Valued Prioritization
  - Kano Analysis
  - MoSCoW Technique
- Relative Prioritization
  - Business Value Vs Risk Vs Effort
  - Relative Weighting Prioritization

Kano Analysis

Developed by Noriaki Kano (1984) , Prioritization features or requirements into four categories based on **customer preferences:**



Functional (Survey few users)

How would you feel if there was a free bottle of water in your hotel room?

[X] I would like

Dysfunctional

How would you feel if there WASN'T a free bottle of water in your hotel room?

[X] I expect to be always like this

Priority: M>I>E

Categorizing Responses		*Dysfunctional Question*				
		Like	*Expect*	*Neutral*	*Like with*	*Dislike*
Functional Questions	Like	Q	E	E	E	L
	Expect	R	I	I	I	M
	Neutral	R	I	I	I	M
	Like with	R	I	I	I	M
	Dislike	R	R	R	R	Q

M = Must have, L = Linear, E = Exciter, R = Reverse, Q = Questionable, I = Indifferent

Kano Analysis

- Mandatory quality (Basic expectations):
  - Expectations on this category are considered basic that should be present in the product. Their absence will frustrate the customer but their presence will not increase their satisfaction, since the customer expects that feature to be present. As an example we can mention the water heating system in hotels
- Desired quality (Satisfiers):
  - Satisfiers are characteristics that can increase or decrease customer satisfaction. They are usually linked to performance. A good indicator from this quality is "the more the better". For example the battery life of a cell phone
- Delightful quality (Delighters):
  - Delighters are characteristics that satisfy the customer when they are present in the product, but if they are not they will not cause dissatisfaction. Most of the time it does not take much effort to deliver it to the customer, but the degree of satisfaction grows exponentially. For example, Free water bottle in a Hotel room.

https://medium.com/@reinaldocamargo/product-backlog-prioritization-technique-kano-model-63bc5d28a1fe

Kano Analysis …

- How do I know if that characteristic is "Delightful", "Desired" or "Mandatory"?
- One way is to build a questioner for a group of people who are interested in the product to answer 2 kinds of question for each feature to be implemented:
  - A functional and dysfunctional one.

Functional
How would you feel if there was a free bottle of water in your hotel room?
[X] I would like
[ ] I expect to be always like this
[ ] Neutral
[ ] I can live with that
[ ] I dislike

Dysfunctional
How would you feel if there WASN'T a free bottle of water in your hotel room?
[ ] I would like
[X] I expect to be always like this
[ ] Neutral
[ ] I can live with that
[ ] I dislike

Categorize the answers

- Use the following table to categorize the answers (functional and dysfunctional):
- The analysis table tells you where a user would place a feature in the Kano Model based on how the functional and dysfunctional responses compare.

Functional Answer \ Dysfunctional Answer	Like	Expect	Neutral	Live With	Dislike
Like	Questionable	Delightful	Delightful	Delightful	Desired
Expect	Anti-feature	Indifferent	Indifferent	Indifferent	Required
Neutral	Anti-feature	Indifferent	Indifferent	Indifferent	Required
Live With	Anti-feature	Indifferent	Indifferent	Indifferent	Required
Dislike	Anti-feature	Anti-feature	Anti-feature	Anti-feature	Questionable

On the "free bottle of water" example, this feature would be "Delightful" category.

Mandatory (Required)
Linear (Indifferent)
Exciter (Delightful)
Questionable (Q)
Reverse (Anti-Feature)
Indifferent (I)

https://medium.com/@reinaldocamargo/product-backlog-prioritization-technique-kano-model-63bc5d28a1fe

Group the answers

- Now you should do the same thing for all the answers and group them into a table like the following: Example of grouped answers and features to be implemented Example

		Categories					
		Delighful	Desired	Required	Indifferent	Anti-feature	Questionable
Theme	Feature A	3	11	41	1	3	2
	Feature B	4	21	20	6	1	0
	Feature C	22	9	14	5	1	3

Prioritization: would be A > B > C

https://medium.com/@reinaldocamargo/product-backlog-prioritization-technique-kano-model-63bc5d28a1fe

Analysis

- Based on that, we can see that "feature A" is a basic feature that customers hope to see.
  - But it is important to remember that, since this functionality has been developed, we should not add effort to this feature as this will not increase customer satisfaction. We should only maintain it. The "feature B" seems to be basic for some people and "the more the better" to another. The "feature C" was classified like a delightful feature. Therefore, the prioritization would be A > B > C.
- In order to keep the Product Backlog prioritized by this model:
  - Make the most of desired functionalities/characteristics
  - It is necessary that all mandatory features be in the RoadMap.
  - Try to leave a space for delightful features because they can increase the degree of customer satisfaction quickly in your favor.

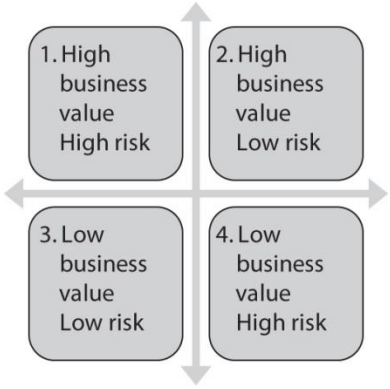
MoSCoW Prioritization scheme

- One of the ways that the most important features make it into the ultimate solution is to determine exactly what those features are using a **brainstorming approach** called MoSCoW.

- **M**UST have this requirement to meet the business needs
- **S**HOULD have this requirement if possible, but the project success does not rely on it.
- **C**OULD have this requirement if it does not affect the business needs of the project.
- **W**ON'T have this requirement, and the stakeholders have agreed that it will not be implemented in a release but may be considered for the future

Factoring Business Value and Risk

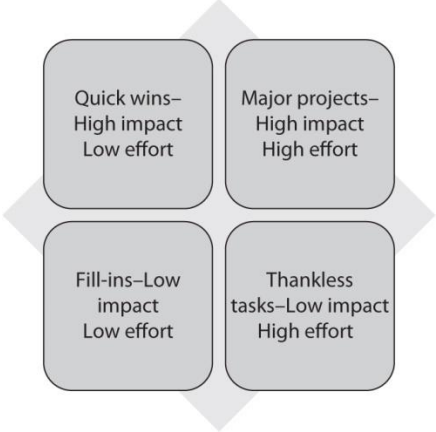
Business Value vs Risk



1.High business value and high risk*, if justified, being completed first-
2.High business value and low risk
3.Low business value, Low Risk
4. Low business values at high risk are done last and possibly avoided.

* The earlier this work is done, the sooner the team will move to mitigate the issues

Business Value vs Effort



- Other factors such as time to market may also be considered in a similar fashion
- "Buy a Feature" is useful for customer-valued prioritization.

Karl Wieger's Relative Weighting Prioritization Matrix

- Based on the expert judgment made by the product owner and supported by the agile team in ranking the score of features in the following way (a scoreboard from 1 to 9 is usually used)
  - Benefit from having the feature
  - Penalty for not having the feature
  - Cost of producing the feature

The priority is determined by dividing the value score as below:
(Value%) / (Cost%)

Feature	Relative Benefit	Relative Penalty	Total Value	Value (%)	Estimate	Cost (%)	Priority
Graph event times	8	6	14	42	32	53	0.79
Can upload photos	9	2	11	33	21	34	0.97
Post-autobiographical profile	3	5	8	25	8	13	1.92
Total	20	13	33	100	61	100	

100-Point Method

- The 100-Point Method was developed by Dean Leffingwell and Don Widrig (2003).
- It involves giving the customer 100 points they can use to vote for the User Stories that are most important.
- The objective is to give **more weight to the User Stories that are of higher priority** when compared to the other available User Stories.
- Each group member allocates points to the various User Stories, giving more points to those they feel are more important.
- On completion of the voting process, prioritization is determined by calculating the total points allocated to each User Story.

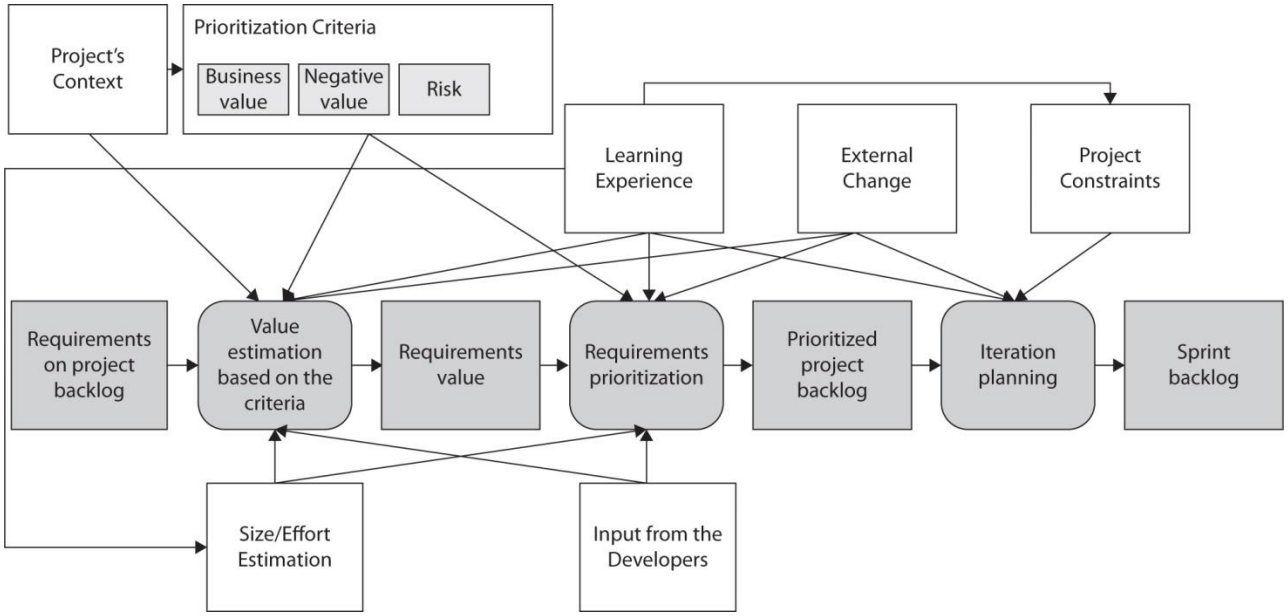
Structured Approach to Prioritizing Your User Stories

- While documenting the stories give them a priority based on the following three parameters.
- Customer Benefit:
  - Think of the story/feature from a customer's standpoint. And evaluate its importance on the below subsets
  - **Level of need:** From a customer's standpoint what do you think is the level of need of this story? Does it solve a major pain point? Or does it solve a moderate problem or only a minor annoyance?
  - **Frequency of need:** How frequently would this feature be used? Often or rarely?
- Opportunity Size:
  - For example you might think less than 10% of your customers might get impacted and none of them are major customers or you might feel more than 50% will get impacted and many among them are major customers.
- Competitive Positioning
  - Evaluate the stories based on Differentiation and customer response. Some stories give you clear differentiation from existing solutions in the market. But while they provide differentiation not many may be using it. So such stories will have a low score on competitive Positioning. While there could be stories that provide major differentiation or response by customers might also be high.

https://productcoalition.com/prioritizing-user-stories

A conceptual model by Racheva (2012) - Client-driven Agile requirement prioritization

- Requirements have been identified and the next step is to prioritize them in order to ensure the team first develops the features that bring the highest value to the business.



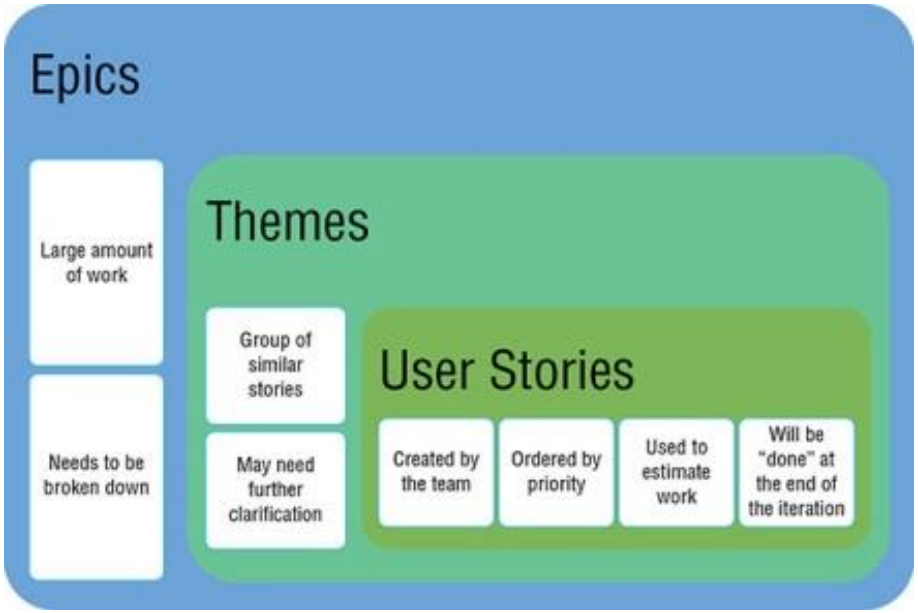
innovate achieve lead

BITS Pilani
Pilani Campus

Planning with User Stories

Epics, Themes, User Stories

- Most user stories do not start as user stories; they start as “epics.” An **epic** is simply a user story that is **too big to be designed, coded, and tested within a single sprint.**
- A **theme** in Agile is basically just a **collection of user stories or requirements with shared attributes** that will allow for grouping of like with like but can’t be produced in one single iteration .



Example Themes, Epics

innovate achieve lead

Theme:

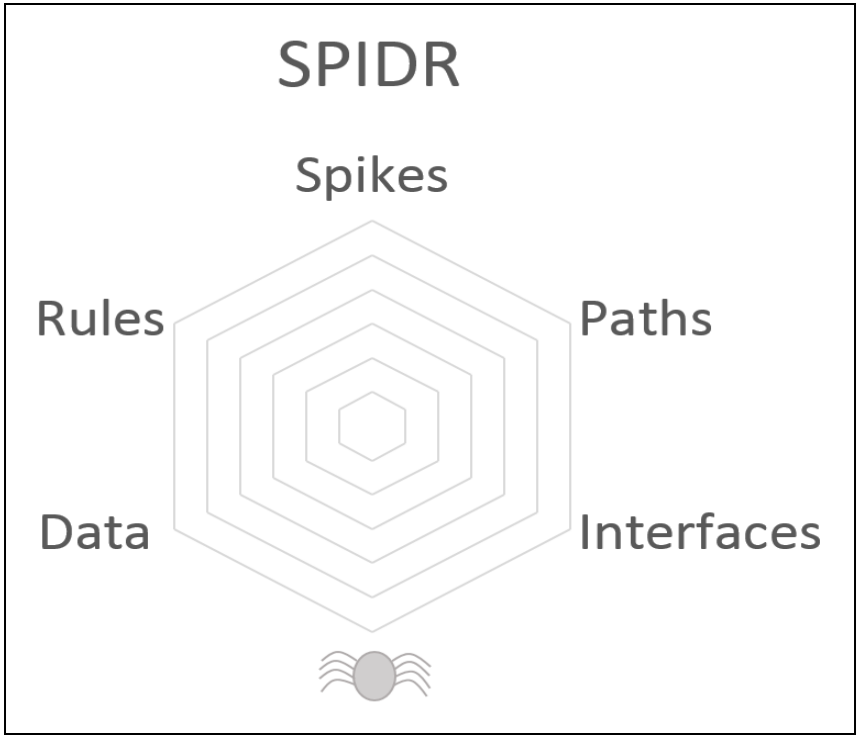
- If the customer says that they want to develop a website that would enable the user to **upload their music by category**, that **upload becomes the theme**..
  - Another aspect of themes is that you don't necessarily need to work on each one in order

Epic:

- As a lover of music, I want a website that allows me to access all of my music, upload multiple formats, and open it up to the public as a paid service so that they can upload and access their purchased music on the cloud wherever they are in the world.
  - Epics need to be broken down into several stories before being worked on.

Splitting User stories

- **We split user stories**:
  - When the user story is too large to fit within one sprint/iteration or in order to make estimates more accurate , If the user story is an epic story.
- **Guidelines** (The guidelines of Cohn M. (2004), can be adapted as follows):



https://blogs.itemis.com/en/spidr-five-simple-techniques-for-a-perfectly-split-user-story

Splitting User stories

innovate achieve lead

Spikes: Are small, prototypical implementations of a functionality that is typically used for the evaluation and feasibility of new technologies

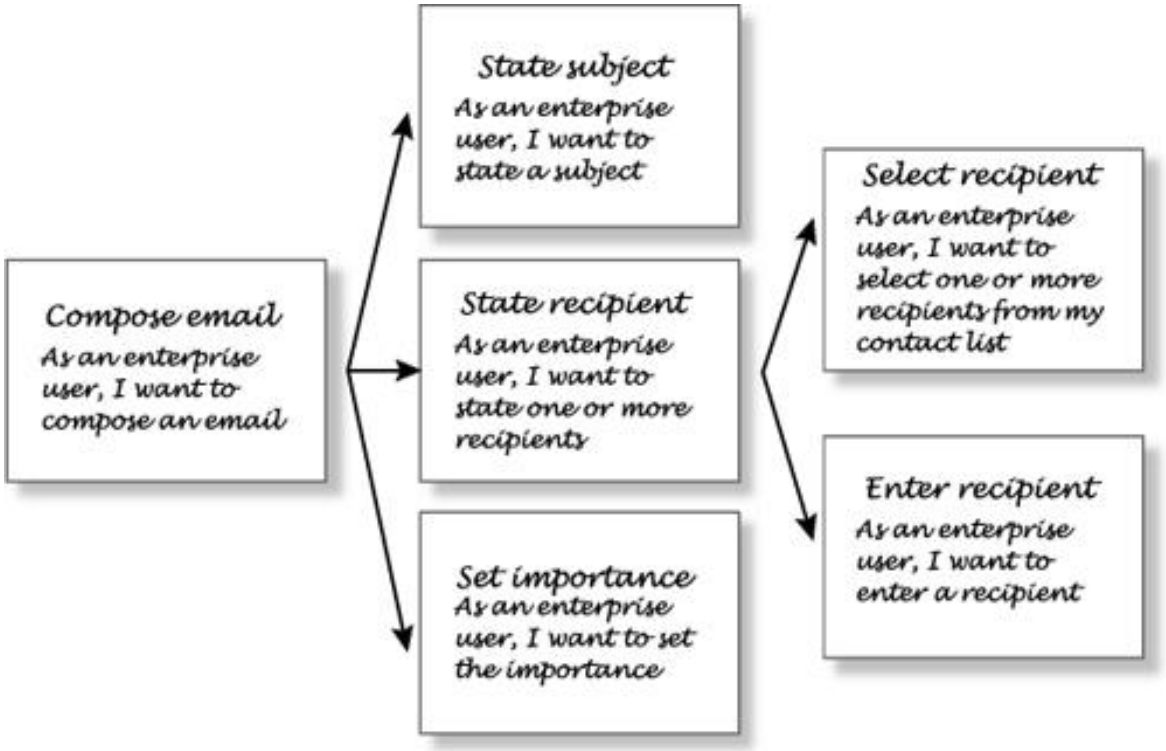
Paths: If there are several possible alternative paths in a user story, one option is to create separate user stories from some of these paths. Ex: Credit card , Pay pal Payment methods, Methods.(CRUD).

Interfaces: Interfaces in this context can for example be different devices or device types, such as smartphones powered by iOS or Android.

Data: when the initial stories refer only to a sub-range of the relevant data.
Ex: Tourist Website: Stories can split along important tourist attractions and tours.

Rules: Business rules or technological standards can be another splitting factor. Ex: Restrictions on buying tickets

Another Example

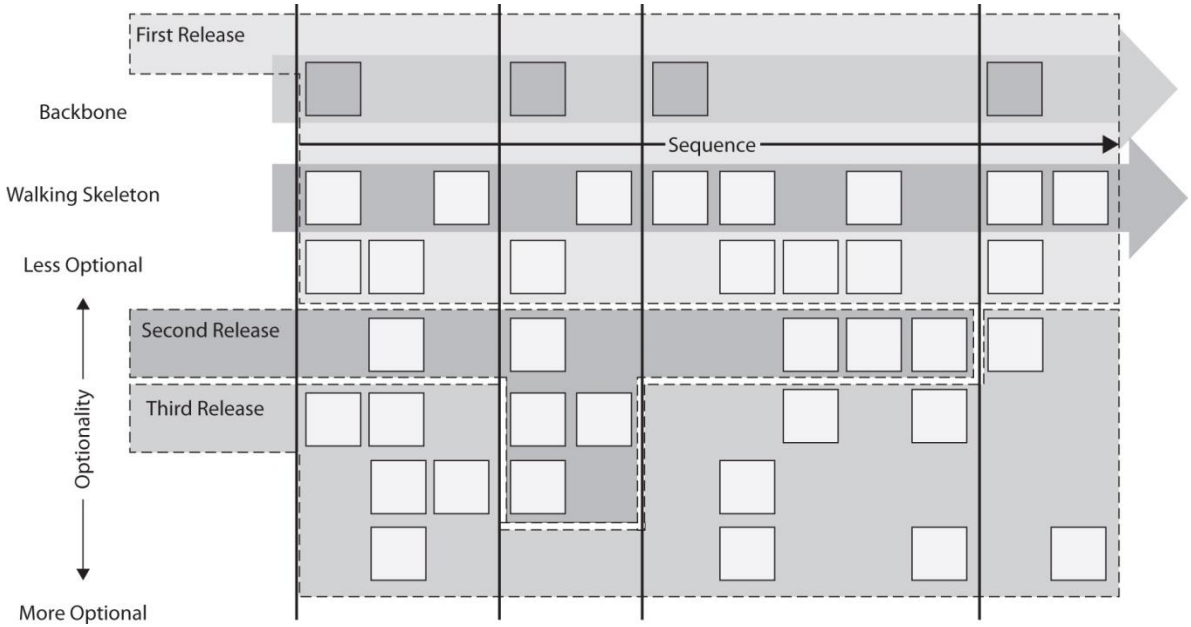


The epic is an example of a compound story, a user story that has more than one goal (Cohn 2004, 24–25). To decompose such a story, we introduce a separate story for each goal. “Compose email” is therefore broken into “State subject,” “State recipient,” and “Set importance.”

Source:: https://www.romanpichler.com/blog/prioritising-the-product-backlog/

Story maps help you prioritize your backlog

- Story maps or user story mapping is a technique developed to organize and prioritize user stories.



- **Backbone** The essential functionality of a product
- **Walking Skeleton** The smallest system that can work, i.e., minimum set of features that could create a working product
- **Additional Features** Any remaining features

An agile design prototype is worth a thousand user stories

innovate achieve lead

Type of Prototype	What	Best for
Paper	Sketch out the screens you want to test. Put them in front of a customer and when a customer taps or clicks on the screen, swap out the screen for a new one manually.	Very early rough concept validation.
Wireframe	Stitch together your wire frames with links that transition between each screen. Sketch will let you do this. Even keynote can be surprisingly effective.	Early validation of the information architecture, or common user journeys within a website or app.
Visual design	The same as previous but at a higher fidelity. If you are getting closer to shipping your designs, you need to start getting much closer to the actual designs customers will see. Tools like InVision are really easy to pickup and use, even for non designers.	Getting concrete feedback on proposed finished designs from customers.
Code	If you can code, or are comfortable with tools like framer or principal you can create the real thing.	The real deal. If you can get quickly made prototypes in code, these are as close to the real experience that you will ever get.

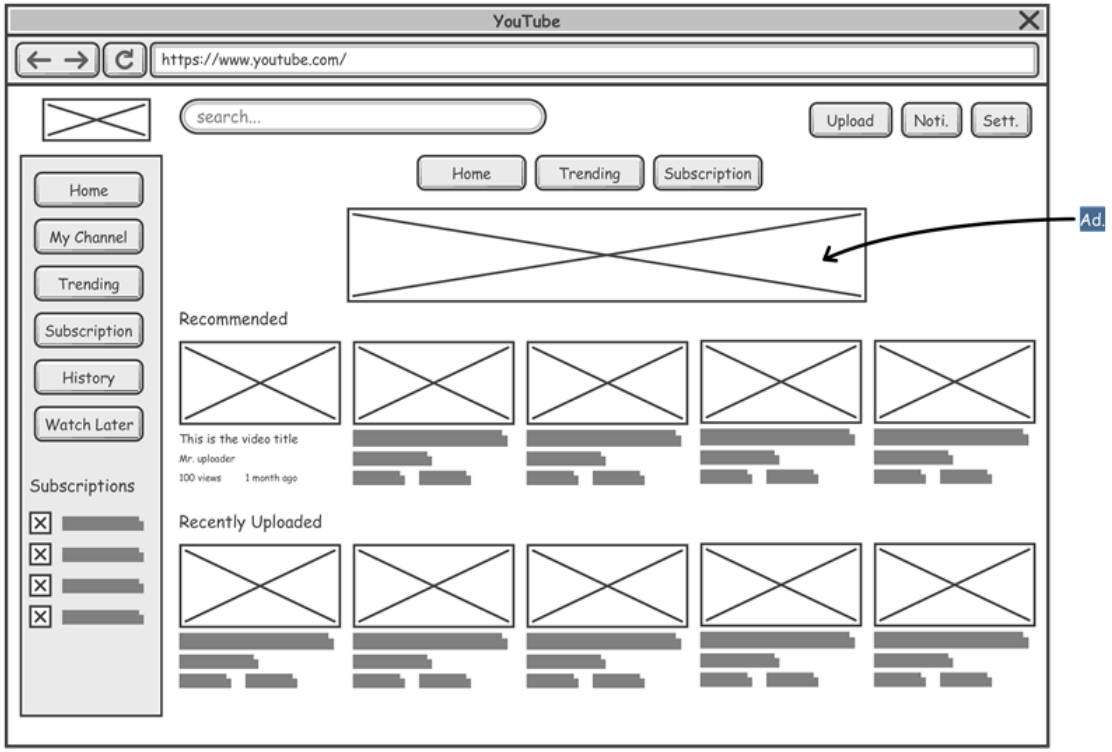
Source:https://www.atlassian.com/blog/agile/agile-design-prototype

WIREFRAMES

- Wireframes are **lightweight nonfunctional user interface designs**, created quickly, that show the major interface elements and how users interact with the system.

Reasons for using Wireframes for Agile projects :

- Information is displayed, Effective.
- Minimum requirements and functionality are illustrated.
- Prioritizations are visible.
- Effects of scenarios are easily displayed.



https://www.visual-paradigm.com/learning/handbooks/agile-handbook/wireframe.jsp

Personas help you get to know your users

- In Agile Software Development, teams tend to use personas and **extreme characters** for getting a better understanding of requirements.
- “Personas are simple descriptions or stereotypes for the roles different people will play when they use your software. They help bring some personality to the system. These are **real people, with real problems**, and understanding where they are coming from will help you meet their needs”
- “Personas are not replacements for requirements, but instead augment them”
- Example:
  - Experienced Gamer, Inexperience Gamer, Frequent Shopper, One-time shopper

Personas Creation

- Personas may include:
  - Name, Description, Picture of a fictional personal (user)—actors
- When designing personas:
  - Add details., Be grounded in reality.•, Generate focus., Be tangible and actionable.•, Be goal oriented, specific, and relevant.

Picture	• Works as product manager for a mid-sized company • Is 35 years old; holds a marketing degree • Has experience as a product owner on software product with Agile teams • Has had some Scrum training	• Has managed mature products successfully • Now faces challenge of creating a brand-new product • Wants to leverage his Agile knowledge but needs advice on creating innovative product using Agile techniques

Personas are great for: Getting to know the customer and adding details, Bringing personality and reality to the systems, Supporting user stories with alignment to the vision

Thank You