

# Session-1

Background, History & Overview:

Terms we are going to use

Roles:

Product Shift

Value per sprint - Maximizing the value delivery

Complexity - Understanding Complexity

Iterative Development - Iterative vs Incremental

Sprint Planning

Example: These are the steps GMo uses,

## Background, History & Overview:

1. Scrum started in Software, however now its everywhere (home, construction,...)
2. The examples will be relevant to Software Product development and projects.
3. To forget anything that we have learned about Scrum to attend the class.



There is no one perfect way of Scrum that works for everyone

Giora - Software Developer, Tester, and Project Manager. Used scrum to create a competitive advantage. Where the competition was giving them documentation, his organization was giving a working solution within the first few months of client interaction. Since 2005, Agile coach and trainer.

Worked with Nike, Blizzard, Sony,... Certified Scrum Trainer since 2006. March of 2020 started doing virtually. Invited for his class for free whenever it happens closely. Currently located 25

miles south of Boston, MA.



CSM Certification will not get you a job. Its a foot in the door for the interview to show/prove your knowledge

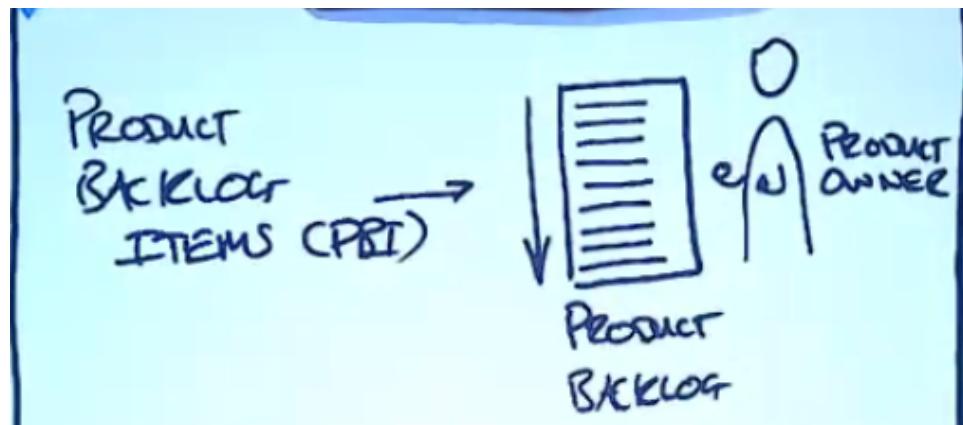
## Terms we are going to use

**Product backlog:** PO manages the log based on priority. It's a list of potential future investments. Just putting it there does not mean it will happen. It's kind of like a wish list. If any of the investments become important, they are updated accordingly.

Not everything in a product backlog is a user story. Not everything will provide a new capability for users.



The team should always have 1 and only 1 Product backlog



It's a living document. Something that is necessary is added and something that isn't, removed. It will help increase visibility and transparency.

*Question - can Product backlog grow so long that it's not achievable? how long can investments stay there without being worked on?* It's up to the PO. Different POs work differently. Some set a limit - say 150 ideas, if it's not in the top 150 ideas, it's not that important. This will also help the stakeholders to not present a laundry list of ideas but instead present just a few to compete with the 150 list ranking. The downside being, PO needs to communicate well with the client that it did not make the top 150. Some POs set an expiration date for each investment.

*Question - There are 280 in the backlog. Notetaking to CRs,... What should be the content of a backlog?*

Not everything invested will have customer value. Sometimes it might help the company, or help increase the team's efficiency or research and more. Having the list is easy to determine which investment and benefit. The backlog needs to be a list of nouns but not verbs.

**Product Backlog Item (PBI):** We are not worried about classifying the investments - CRs, Bugs, Tasks, Changes,... We only worry about the priority and the value they deliver.



Everything is a PBI, but not all PBIs are user stories. A user story is a type of PBI.

**User Story:** There will be no questions related to User Story. It's not part of Scrum. It's a tool to better collaborate and understand the needs of stakeholders and/or customers. A user story is something that provides value to users.



User story - It is context but not the Goal. It's defining the problem, not the solution.

**Sprint:** Every sprint starts with sprint planning. Sprints are typically less than a month. 2-3 weeks is a sweet spot. 4 weeks can be really long. 1 week sprints can be overly intensive. **It's a fixed time, variable scope (requirements) item.** Each sprint delivers an increment.

- Sprint Planning - the 1st step of the sprint is to see which of the backlog items will be worked on in the sprint. Developers need to understand and come up with an implementation plan, which is a sprint backlog.
- Sprint Backlog - Developers craft an implementation plan to deliver the backlog items and maintain the plan for the life of the Sprint. The items are chosen to accomplish the Sprint Goal.
  - Contains what to tackle
  - how to tackle
  - developers maintain and update the backlog
- Sprint Goal - What the team is planning on achieving at the end of the sprint. The entire team needs to agree. PO does not come the next day and change it. It stays consistent for the next 2 weeks. End of Sprint Planning, the sprint goal is decided by the entire team (PO, developers,...).
- Sprint Review -
- Sprint Retrospect -



POs own Product Backlog. Developers own the Sprint Backlog.



POs and Developers together work on and confirm the Sprint Backlog.



At the end of the sprint, an item will be added to the increment or back to the product backlog, there are no other states.

**Question - When to release?** If it's done before the sprint review, it can be released right away. It does not have to coincide with the start and end of sprints.

**Increment** - It's the latest version of everything we have. It's adding air to the balloon. It's everything we have completed in this sprint and all the sprints prior. Represents all the things that are complete. It needs to be **done and usable** to be part of the increment.

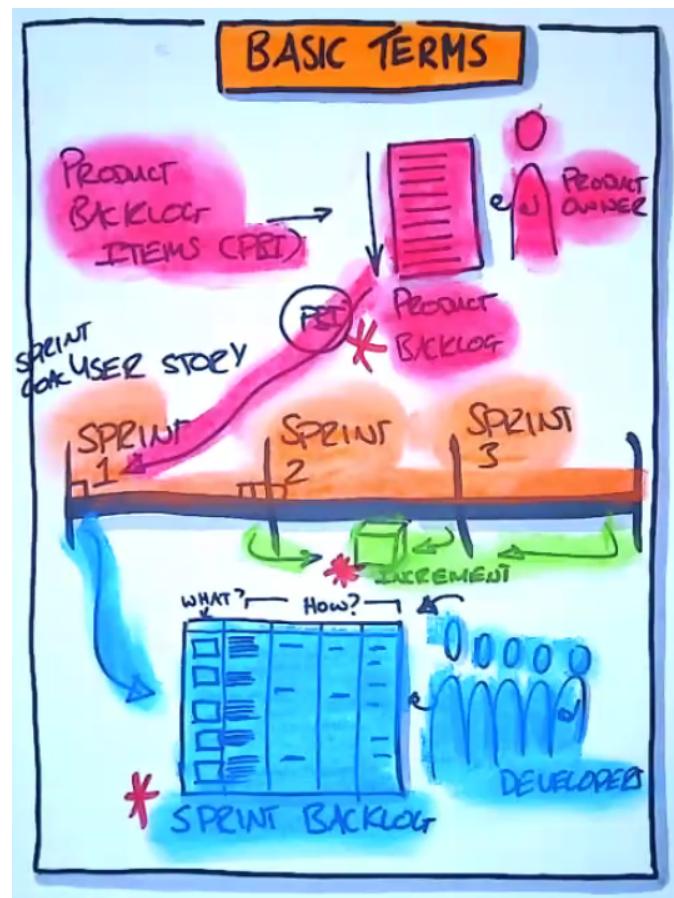
- Done - There is no standard definition of done. It's unique to different teams. It's extending the org standards. A starting point would be the org standard. Ex: It needs to be scanned to be called done, however, different teams might add more to it accordingly for that team to mark it as done.
- Usable - The user could use it.
- Useful - The user would use it.

**Question - How frequently can you change the sprint duration?** Having consistency will give cadence to the team.

**Question - What if the team is working on different projects?** There are no “projects” in Scrum. There are items in the Product backlog that can come from different projects which the PO needs to prioritize.



Product Backlog > Sprint Backlog > Increment



## Roles:

- **Product Owner** - Always suggested having a goal in mind. Held accountable for investment decisions. They define,
  - What are the acceptance criteria
  - What is important
  - Why are we addressing these needs over other
  - When are we going to work on other priorities
  - Anything related to - What, Why, and respective When
- **Developers** - Responsible for
  - Who is going to work on it?
  - How long would it take?
  - How much work can we take in this sprint?
  - When are we going to deliver?
  - Anything related to - Who, How, How much, and respective When
- **Scrum Master** - The role is to enable POs and Developers to work on their actions effectively. To facilitate what happens if it does not go as planned. Support and Coach POs and developers to be successful. CSM is a leadership, people-centric role. Pushing the team to make it better. Understand what the product is being used for, who the competitors are, which stakeholder priorities make sense,... CSMs work for their teams. Facilitating idea generation, and root cause analysis.



There are no BAs, QAs,... It's just Product Owners, Developers, and Scrum Masters.



POs → Prioritize Customers. CSMs → Prioritize Teams.



Less conflict with DEVELOPER/CSM VS CSM/PO. If push comes to shove, developers can act as CSMs (dual role), but CMSs should not be both CSM/PO.

Product - 12-year-old kid's toy

User - Child (is it fun to play?)

Customer - Parent (is it safe + fun?)

## Product Shift

How do we rate projects? If it's delivered on time.

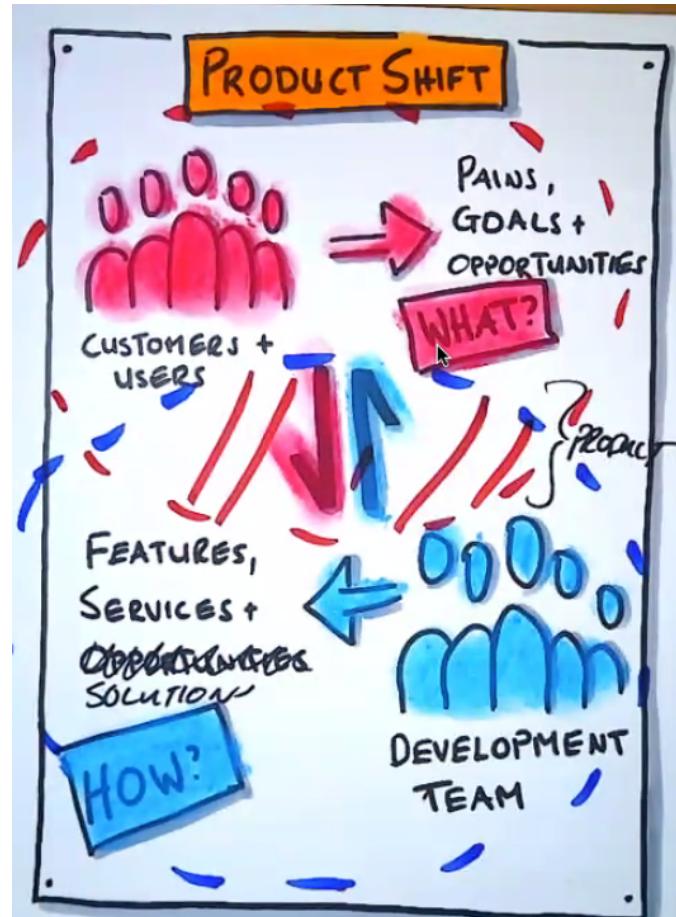
How do we rate products? If it provides value to the customers.

Whatever the team produces is the product.

Customers + Users' → Pains, Goals, Opportunities

Development team → Features, Services, Solutions

**Ex:** Kitchen remodeling. We invite the contractor and write the list of stuff needed in the kitchen - backsplash, refrigerator,.... the contractor will ask for the budget + timeline.



Product is what customers need + and what the development team can deliver.

POs need to establish the vision- what are we solving? what are we addressing? whose needs do we care about?

Portfolio Managers and higher-level people define the impact

- Ex: We want to allow mobile access for our customers.

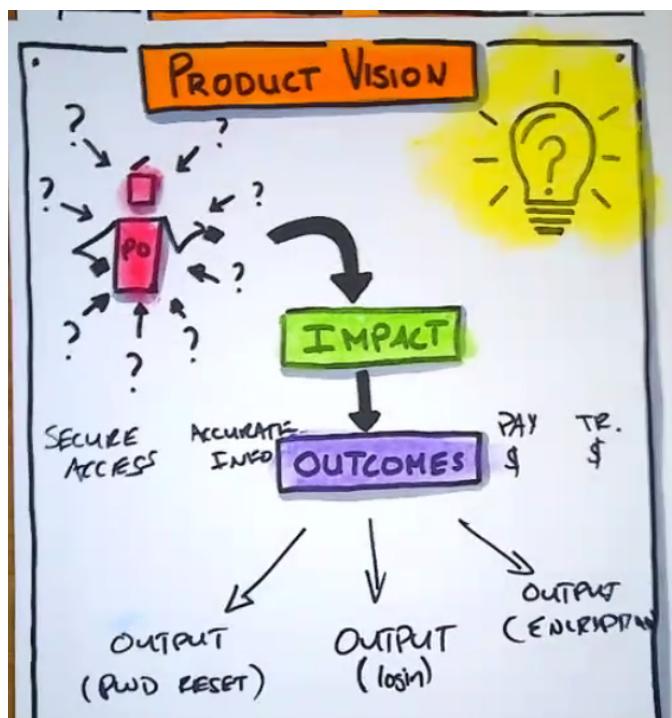
POs need to define the outcomes, ex

- Secure Access

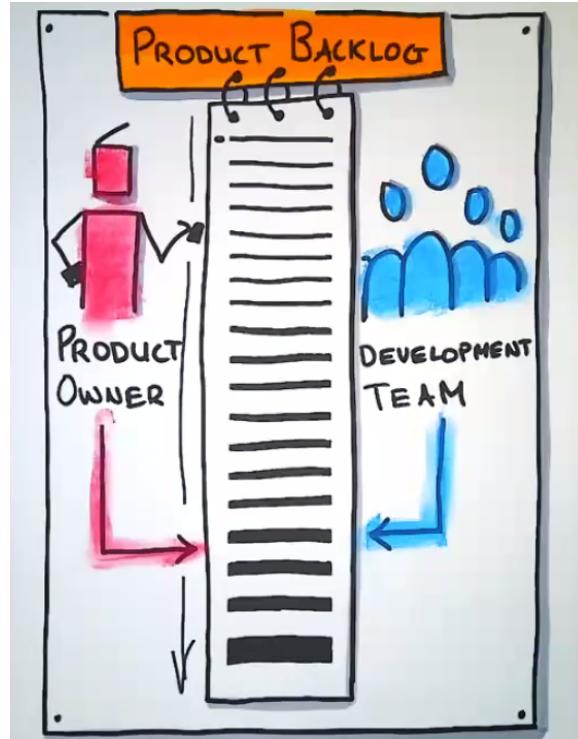
- Accurate Information
- Pay someone
- Make transactions

Developers produce outputs for the outcomes, ex

- Password reset
- Login
- Encryption



The product backlog is a prioritized list of items/outputs. The reason the items are in the Product backlog is that they make an impact. The items may not make a direct impact but sometimes enable us to make the impact. For example: In the above example, if the item is Encryption, we may research different encryption models that best suit the product. We invest in something today to do something tomorrow or later.



## Value per sprint - Maximizing the value delivery

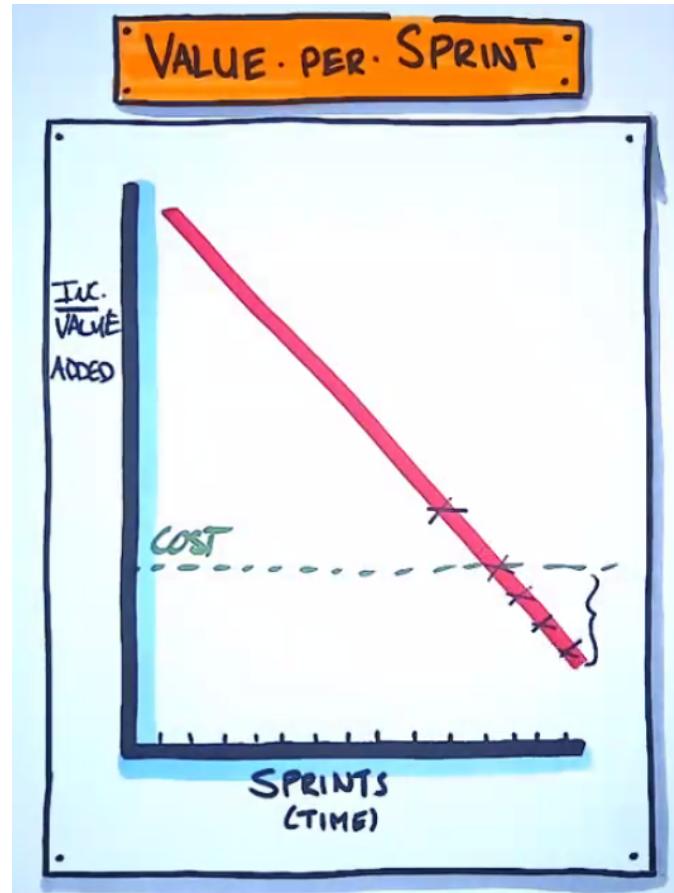
Imagine a project with a fixed time and fixed scope. How would the graph look like if you are looking at Time vs Value?

- Assuming the product backlog is prioritized based on the value
- Expert PO who prioritized the backlog well
- Expert developers who finish the sprints on time
- We are not talking about the incremental value, just the value of each sprint

So by definition, the initial sprint will deliver the highest value, and the last sprint will deliver the least value. However as the team is consistent, the cost will be consistent. So over time, in the end, there might be items that might not be worth the investment. So the final x's below the cost line, might not be worth investing.



Lesson 1 - not every idea is a high-value idea. Not every idea is worth implementing. Not every feedback, ticket, commitment, and yelling is worth implementing. Some are and some aren't.

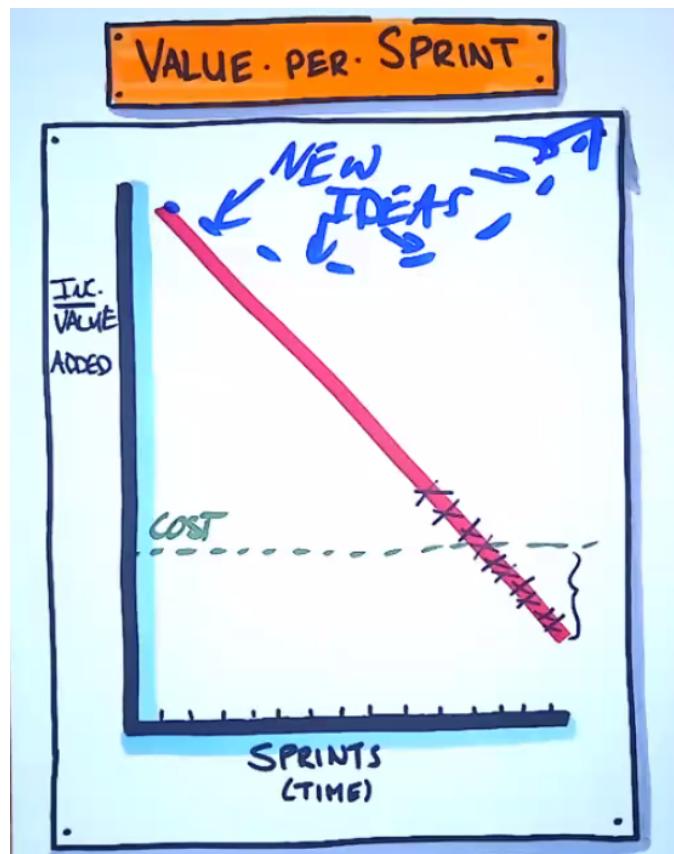


But because of this, the team is destined to deliver lesser value for each sprint as the value is going down. This might be depressing for the team. In order to keep the motivation up and to have the delivering better value, have to come up with new ideas.



Lesson 2 - Without being open to new ideas, you cannot discover new high-value ideas.

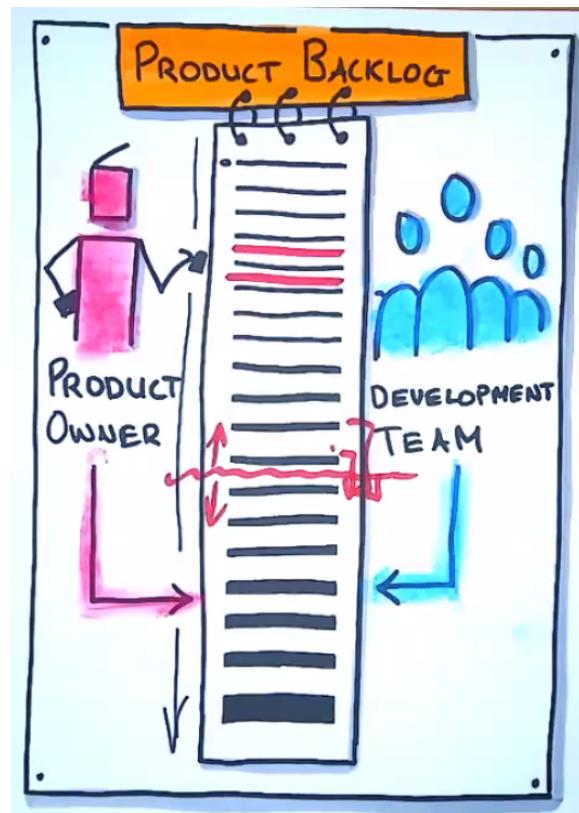
New ideas are good and the sources are vast - colleagues, government, customers, partners, vendors,...



What if we are going with the traditional waterfall model, and during the project, there are some new ideas, can we not swap the new high-value ideas for the old low-value ideas to still deliver within the same time and budget?

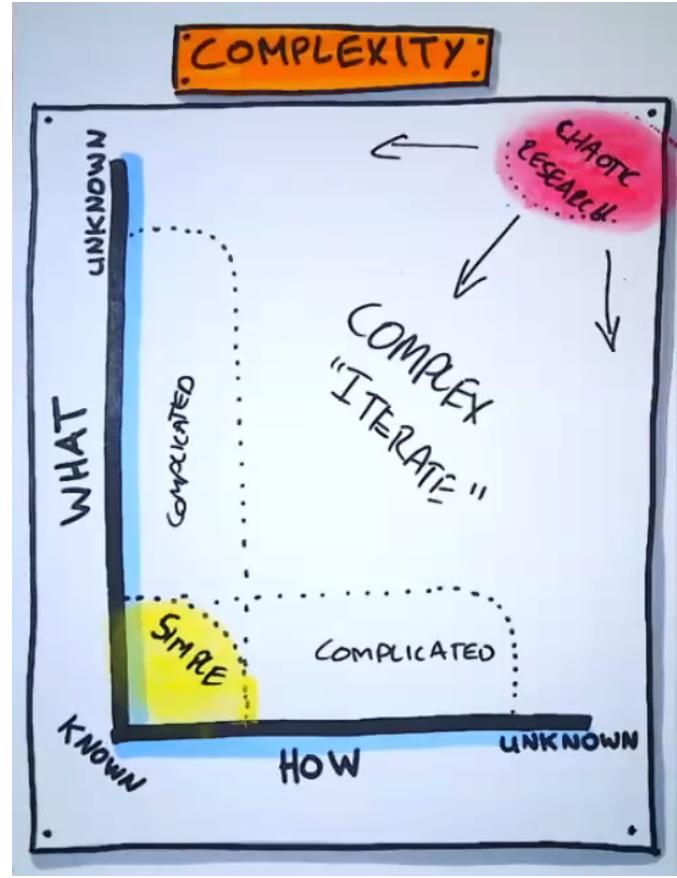
It was not possible in the traditional waterfall model, it's a big batch process where we start investing in the entire project and start working as a whole. Scrum, is a small batch model so

while the team is working on high-value items if someone comes up with new high-value items, it's easy to switch the old low-value items with the new high-value items.



*Question - Where does the scope creep happen in Scrum?* that is when the new scope is added after sprint planning is done, however, it needs to happen with the confirmation of the entire team.

## Complexity - Understanding Complexity



From a process standpoint,

- Simple - We know what to do and how (ex: data migration - it's not easy or cheap, however, we know that the data needs to be migrated from A to B). The best way to tackle them is to categorize them.
- Complicated - You want to build a house, however, you don't know how it looks like or if you know the solution, however, not sure how to apply the solution.
- Chaotic - No idea what to solve nor how to. Research, experimentation,... required to figure out and implement.
- Complex - There might be problems that you can analyze indefinitely or where from your analysis to design and implementation, the requirements changed (analysis paralysis). They are difficult to predict because of a high degree of unknown.

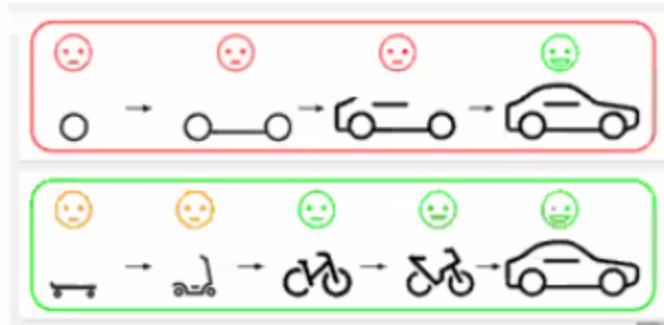


Solution → Iteration. Building something, getting feedback, making changes.

**Example:** Amazon gets a % from your shopping cart items purchased. That is what at the checkout, in the end, they will show you 3 items that you are most probably will add-on to your cart. Maybe if someone is buying an iPhone, recommending a case or checking their previous history and recommending new items or refills (toilet papers,...) or recommending based on seasonality or recommending based on age. Also, have to value the add-on, if purchasing 40\$, making an 80\$ add-on would not make sense. All these factors come into play. Even then it's not 100% accurate.

- Recommending case as soon as they add a phone - analyze the data
- Recommend a case after a month from purchase - analyze the data

## Iterative Development - Iterative vs Incremental



Incremental - Building something in pieces. Need to be really sure what it's going to look like. Need to make sure you have the time and funding to reach the goal because if the funding ends, you have nothing of value.



With incremental, if funding stops, you do not have anything of value

**Iterative** - Iterative means versions.

- V1 = Skateboard - better than walking. However, accidents mess up the shoes and are not stable.
- V2 = Scooter - not fast, need to carry.
- V3 = Bicycle - has a handle, stable, however difficult uphill and issues during rain.
- V4 = Bike - issues during rain
- V5 = Car - all issues resolved.



With iterative, if funding stops, you have some product of value.



Stopping at a car makes sense as it resolves all the issues and not building a rocket as it's not necessary, and requires more money and time.



Sprint increment - each incremental or iterative development will end with a sprint increment.

**Exercise** - Current global pandemic - we are still not sure when it ends, real estate - what it will happen, supply chain - when it will settle down. Examples of companies that adapted to an iterative approach.

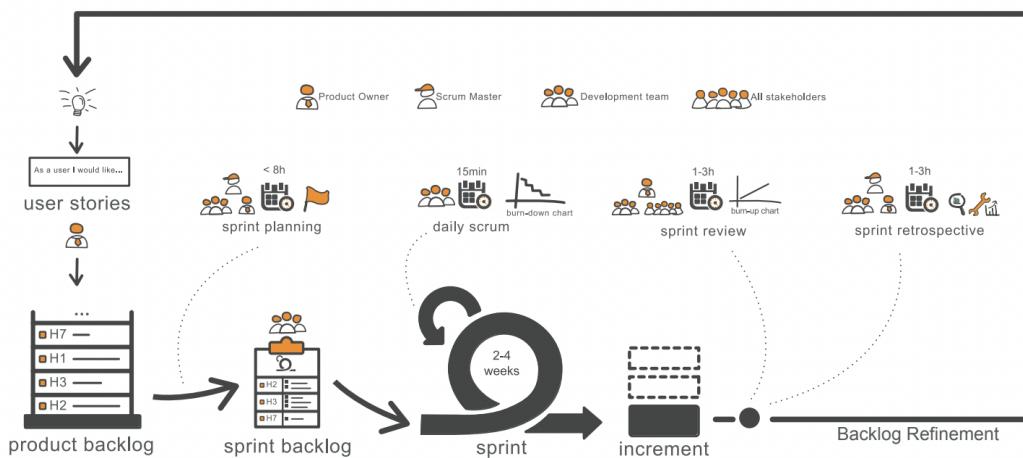
- Zoom - gave out the licenses to all the schools in the country. Imagine the confidence they have in their platform to give it out.
- Telemedicine
- Education
- Entertainment Movie streaming - introducing premium for an existing subscription

**Iterative process** - gives the tools to figure it out. One of the best examples is Tesla. You do not have a 2022 car or 2021 car, you just get Model S however based on the year you buy you get different versions of brakes or HVAC system or battery or more.

## Sprint Planning

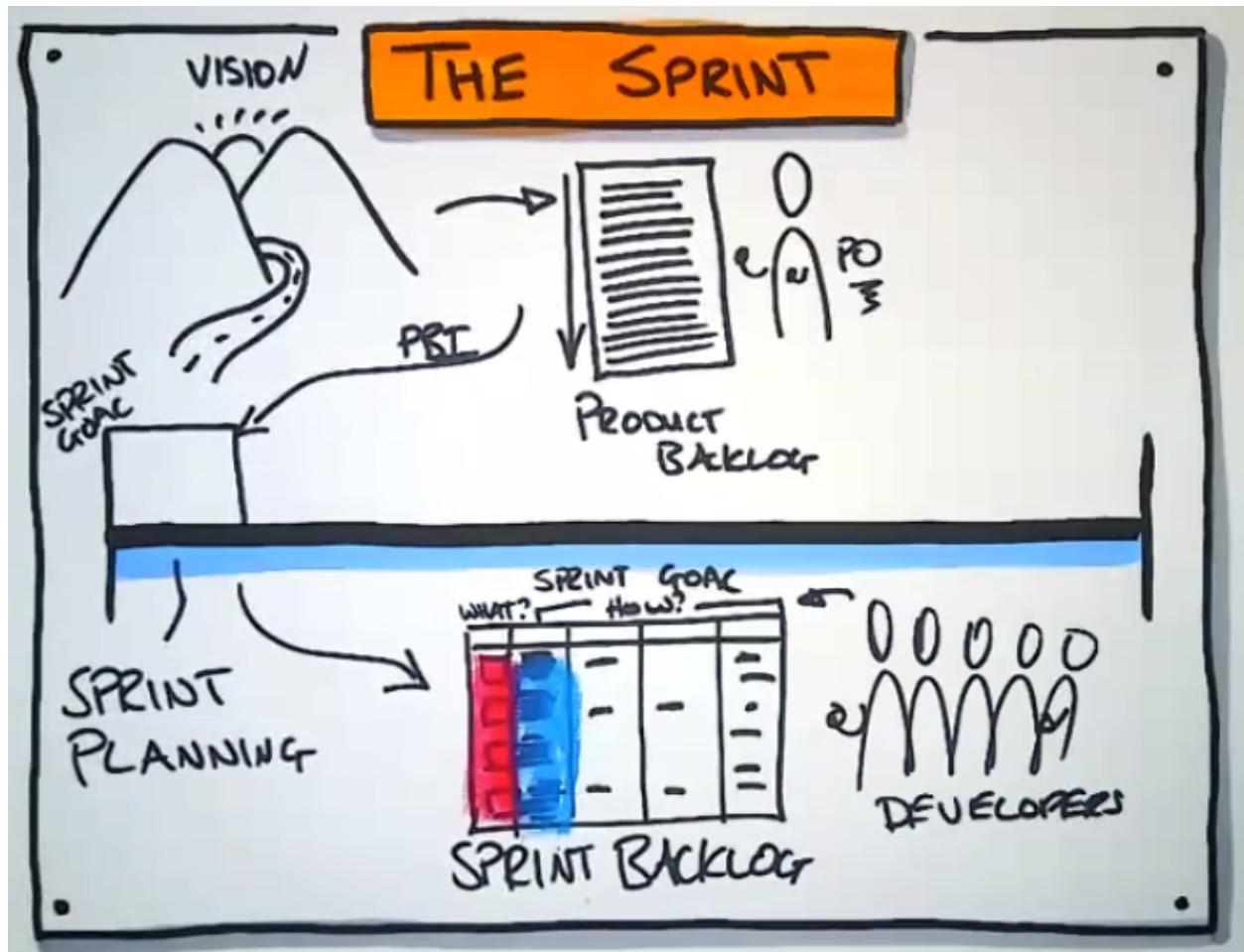
It's an opportunityThere are a lot of ways to draw the Sprint.

### THE SPRINT



- Vision - Why are we here? What are the problems we are addressing?
- Product Backlog - List of outputs from Vision that are prioritized. PO is responsible to define and manage the product backlog.
- Sprint planning - The team establishes the Sprint goal and lists out the Sprint backlog.
  - Sprint goal - what are we trying to achieve
  - Sprint backlog - Subset of product backlog
    - what are we going to work on
    - How are we going to work on
- Developers - Responsible for managing the Sprint backlog (maintaining and updating it). The product owner comes up with the problems that need to be addressed, while the

developers determine how to tackle the problem which begins in the sprint planning.



The items in the backlog are granular which can be done in a matter of hours.



Developers can't unilaterally decide that they don't want to work on it, POs can't unilaterally decide what more backlog items the developers can work on if they have some free time in the sprint - needs to be decided as a team. (except the how - which is defined by the teams internally).



POs can decide to cancel the sprint unilaterally. The reason to do that is if they cannot agree with the team.

JIRA, DevOps, ServiceNow - Not Facilitation Tools (they can house the plan but cannot facilitate it)

Whiteboard, PostIts, Markers, Miro - Facilitation Tools.

### **Example: These are the steps GMo uses,**

1. Draw a grid of who is available when
2. Requesting PO to Communicate or provide their idea of the sprint goal
3. Request PO to identify 2-3 items for starters and get into the details of each of the item
  - a. During the discussion, involve the developers detail down the How
4. Request developers to brainstorm all the tasks for all the backlog items
  - a. Once I have it, collect those and show them parallelly (horizontally)
  - b. Vertically, list out the tasks that are common across
5. Estimate - 3 options. Less debate.
  - a. Less than 8 hours
  - b. Less than 4 hours
  - c. Less than 2 hours

- d. If someone says its way more than 8, it's way too big
- 6. I will ask the team if they have a learning goal

### **Things to do to improve**

- 1. Introduce Parking lot - Anyone can add a topic that is not part of the agenda which can be brought up for discussion at the end. Needs to be accessible for everyone so they don't interrupt.
- 2. Emphasize the Sprint Goal and what are the things we need to do to pursue the goal
- 3. Present developers with problems instead of solutions so that the discussion is more interactive/engaged.
- 4. Getting food delivered or introducing food (surprise them)
- 5. The scrum team should be 10 people or less (according to the scrum guide).
- 6. Need to value collaboration