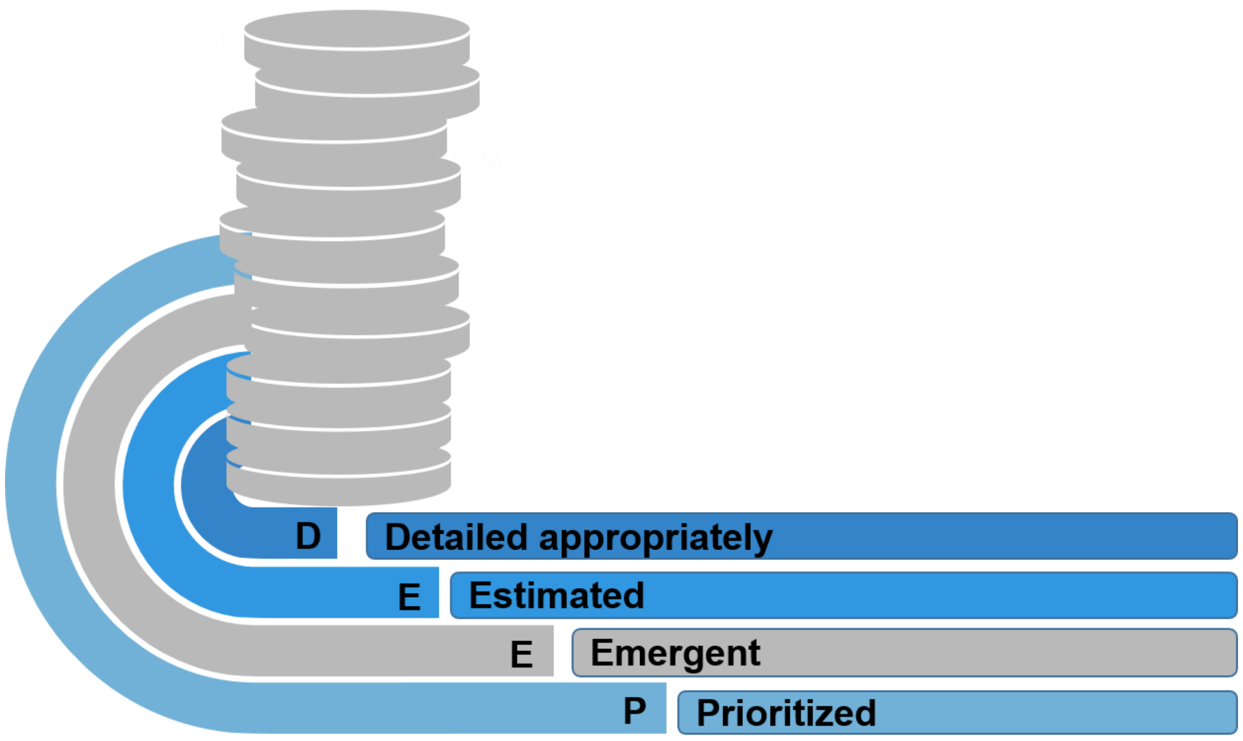
PRODUCT MANAGEMENT

PRODUCT BACKLOG MANAGEMENT

DEEP Qualities of the Product Backlog



- Product Backlog has four qualities.
- These four qualities are known as DEEP.

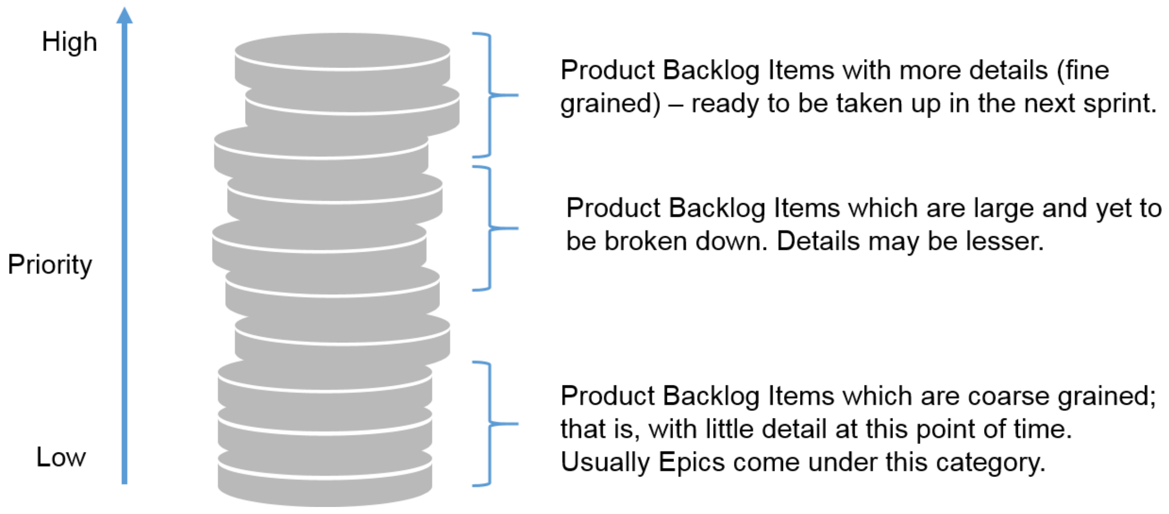
DETAILED APPROPRIATELY

- Product Backlog Items are detailed appropriately. The items with higher priority are provided with more details when compared to items with lower priority.
- This ensures that the items that might be taken as part of the next sprint are ready with more details than the ones which might be implemented in much later sprints.
- Throughout the project, requirements are discovered. So, the lower priority items will get enough details as we progress through the project.

Example	Let us observe two Product Backlog Items shown below as an example. The items are shown with an ID for illustration purposes. 1. **ID: 76435** As a supplier, I want to access the partner portal so that I can see which orders I must work on and make arrangements for requested material in required quantity. **Acceptance criteria**:

	1. Supplier must already be registered in the system 2. A supplier should be able to see only the material that is allotted to their company. No supplier should be able to see the material allotted to other suppliers. 3. Procurement Manager should be able to know which Supplier has looked at material requests. This Product Backlog Item is described with enough details for the development team to work on in next upcoming sprint. 2. **ID: 80923** As a Procurement Manager, I want to see the list of suppliers <and TBD> and the materials requested against them so that I can plan my activities <and TBD>. 3. **Acceptance Criteria:** 1. TBD (TBD indicates “To Be Decided”) This Product Backlog Item is of low priority. It need not have enough details at this point of time as it will be implemented in later sprints.

- This “D” is shown graphically below:


High
Priority
Low
Product Backlog Items with more details (fine grained) – ready to be taken up in the next sprint.
Product Backlog Items which are large and yet to be broken down. Details may be lesser.
Product Backlog Items which are coarse grained; that is, with little detail at this point of time. Usually Epics come under this category.

ESTIMATED

- Product Backlog Items are estimated.
- Estimates are expressed in Story Points or Ideal days.
- Task level details and effort estimation for these items will be done as part of Sprint Planning, and Sprint Backlog (which is a subset or child of Product Backlog) will contain more details and tasks.

EMERGENT

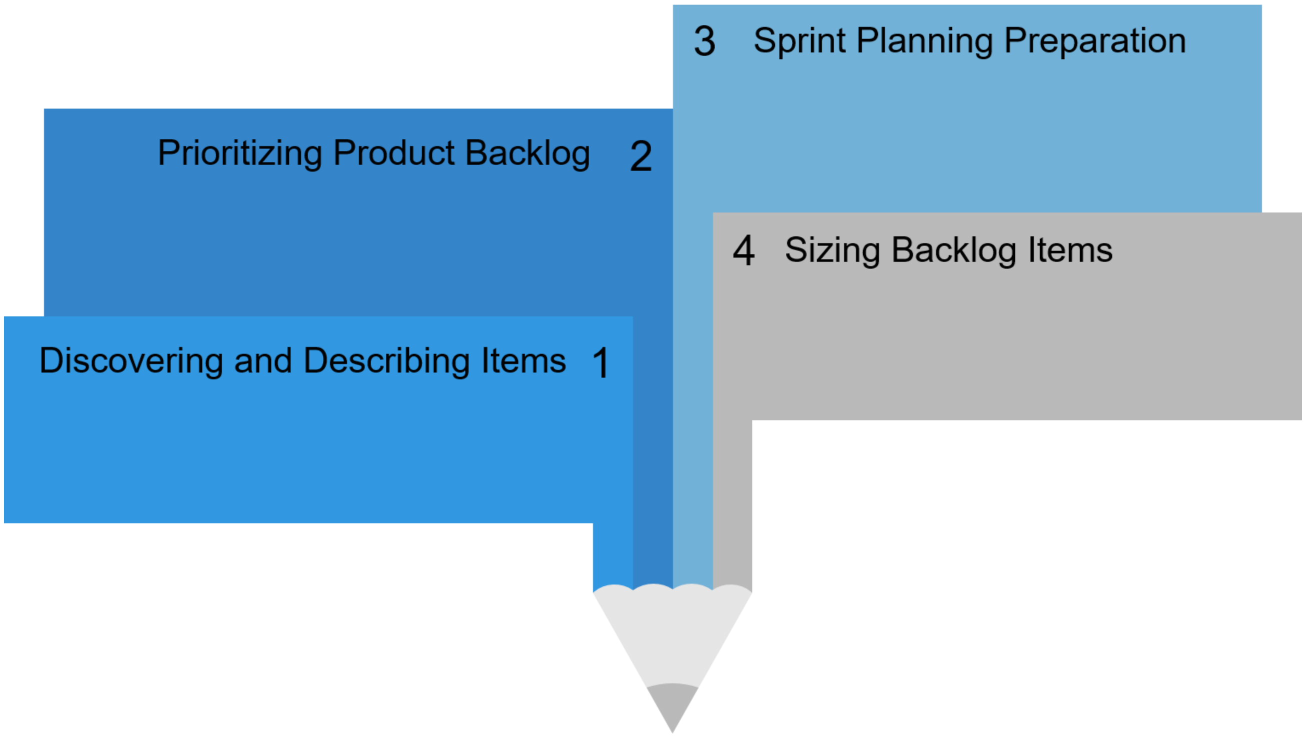
- Product Backlog is ***live*** throughout the project. It means that the Backlog items can be added, moved, modified, or removed as required by the customer.
- It changes continuously throughout the project.

PRIORITIZED

- All Product Backlog Items ***must*** be prioritized.
- High priority items will be on top and implemented first. Lower priority items will be at the bottom.
- Once a Product Backlog Item is completed, it will be removed from Product Backlog.

Backlog Grooming

- Agile software development is all about having the scope, that is, in the form of Product Backlog ready for changes as they emerge.
- Backlog Grooming is an ongoing exercise carried out throughout the project to:
    - Discover new items; that is, new requirements as they emerge
    - Change existing items—adding more details to a few items or moving a few items up or down the Backlog as part of reprioritization, and removing few items.
    - Prioritize items as per business requirements
- Product Owner, Scrum Master, and Development team members collaboratively work on Backlog grooming.
- The grooming of the Product Backlog depends on the project and team's decision. Few teams might groom the product backlog after their daily scrum. While a few teams may dedicate a day or two in a week for grooming during the sprint. Few other teams might do this toward end of the sprint.
- For Sprint Planning meeting to be successful, a well-groomed Product Backlog is mandatory.
- The following graphic shows the four-step Backlog Grooming process.


3 Sprint Planning Preparation
Prioritizing Product Backlog 2
4 Sizing Backlog Items
Discovering and Describing Items 1

DISCOVERING AND DESCRIBING ITEMS

- This is an ongoing exercise.
- Typically, in any project, at the very beginning, high-level scope in the form of Backlog Items will emerge. This could be the initial Product Backlog.
- As outlined in Minimum Marketable Product earlier, concentrate on bare minimum features that are mandatory for the Product to achieve its intended goals. The Product Owner, Scrum Master and Development Team collaboratively discover the Backlog items and add them to Product Backlog.
- At the beginning of the project, it is quite normal ***not*** to have many details about as many stories as possible. Remember that Product Backlog grows and more details emerge as the project progresses.
- User Stories are used to describe the Backlog Items.
- A User Story contains:
    - A name or an ID (for unique identification)
    - Brief narrative in “as a <role> I want <requirement> so that <benefit>” format
    - Acceptance criteria that must be satisfied for the Story to be deemed complete
- Backlog Items can be grouped together based on a few criteria. **Themes** can be used to relatively group User Stories together, and they act as placeholders for product functionality; they structure the backlog, aid prioritization, and make it easier to access information.

Example	Let us see a couple of examples for these Themes:

1. Typical IT Systems will have these four groups of requirements. You can use these four themes to group the related User Stories:
   - Masters (contain Master Data)
   - Transactions (contain actions carried out on Master Data)
   - Interfaces (contain information coming in and flowing out)
   - Reports (contain various reports generated for decision making)
2. In a Banking system, a few themes as given below can be arrived based on functionality:
   - Salary and Savings Accounts
   - Current Accounts
   - Cards
   - Loans

- Themes can be used for creating hierarchies of Backlog Items.

PRIORITIZING PRODUCT BACKLOG ITEMS

- Prioritizing Backlog items is the responsibility of the Product Owner.
- Backlog Items are detailed according to their priority; that is, higher priority items are described in more details than lower priority items.
- The following factors can be used to arrive at prioritization of Backlog Items:
  - Value
  - Knowledge, Uncertainty, and Risk
  - Releasability
  - Dependencies
- Let us explore them in detail:

Value-based prioritization

- Value is a most common and widely used prioritization factor.
- Please remember that this value is from the perception of the Customer and not the Development team.
- We aim to deliver higher valued items quickly and early.
- Each Product Backlog Item is given its value – the value it will add toward the final deliverable.

Knowledge, Uncertainty, and Risk-based prioritization

- Knowledge, Uncertainty, and Risk are interlinked.
- Lack of knowledge about the Product and its features lead to many uncertainties and risk.

Example	When a customer calls for Proposals (Request for Proposals, as it's popularly known), there would be certain criteria to evaluate the suppliers. Some criteria used by customers include: • Prior project experience in similar technology or domain • Asking for past project customer references and case studies These criteria help the customer to know how knowledgeable the suppliers are in similar projects and technologies. If suppliers provide credible references, it will give assurance to the customer that the project has less risk.

- Product Backlog items can be prioritized based on risk. The items with high risk are placed on top of the Backlog – to be addressed as soon as possible in next Sprint itself.

Release-based prioritization

- Release is deployment of certain features of a software product.
- Release-based prioritization delivers the functionality that is ready for release.
- Since Product increments are released frequently and early, customer feedback is made available to the team for Inspect and Adapt.

Example	For the roll out of a Global Financial Reporting system, the priorities were already set. The project is about six months old, with two releases completed. During the third release, the local government came up with a regulation (for example, introducing a new tax). The deadline to meet this new regulation can't be deferred beyond the required implementation date by the local government. In this case, the Product Backlog is adjusted to include Items that can deliver the required functionality to meet the new regulation as part of the next Release.

Dependencies-based prioritization

- Dependencies exist between User Stories.
- Functional Requirements often have dependencies with other functional and non-functional requirements.
- If a user story is dependent on another story that is not part of this Sprint, it is better to club these two dependent user stories to be part of the next Sprint.

Example	In a typical IT Software Development Project, reports have dependencies on master and transactions data. So, any report development User Stories are taken for Sprints only after the masters and transactions are set up. In a similar way, transactions might require interfaces that bring and send data to upstream and downstream systems. So, before working on Transaction User Stories, Development teams will work on Interface User Stories to resolve the dependencies.

SPRINT PLANNING PREPARATION

- Product Backlog grooming is a mandatory prerequisite for Sprint Planning.

Sprint goal-based prioritization

- Each Sprint must have a Sprint goal.
- This Sprint goal is created by the team (Product Owner, Scrum Master, and Development Team together) and it offers the following benefits:
    - Creates necessary alignment between team members
    - Ensures the focus is on required functionality agreed for the sprint and minimizes variations from this baseline
    - Stakeholders are informed about what the team is working on
- Backlog Items might be added, moved, or removed per the Sprint goal.

Just enough items, just-in-time based prioritization

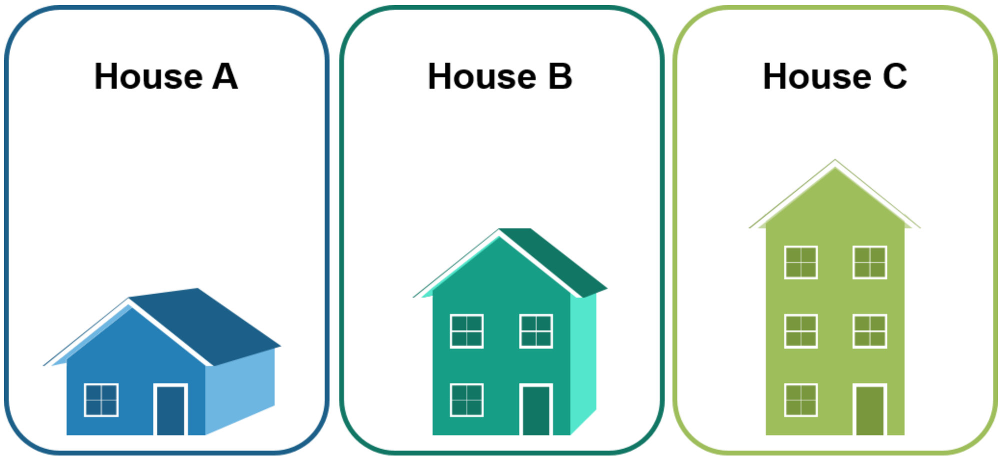
- Grooming is a cascading activity; grooming during one sprint facilities the upcoming sprint.
- By deliberately focusing on only a few items required for the upcoming sprint, the team can have just enough items in Backlog to support Just-in-time decisions and priorities.
- Larger items are decomposed into smaller ones as part of grooming.
- Small just-in-time items help the team to set realistic commitments to the Product Owner. This also helps the team to gain confidence, once the smaller items are delivered and accepted by the Product Owner – that they can meet the Product Owner's requirements.

SIZING BACKLOG ITEMS

- Size and effort are two different attributes of each Backlog Item.
- They help us to prioritize, track, and forecast project progress.

Story Points

- Story Points are relative measures of raw effort and size.
- Human beings are very good at relative estimation—using effort and size of one item as a reference to compare other items.
- To understand the relative sizing, consider this example:



- Let us assume ***it would take 10 hours*** to build House A.
- House B is almost double to the size of House A.
- ***How much time, most likely, will it take to build House B?***
- Because House B is almost ***double in size with relation*** to House A, most of us can answer that ***it would take about 20 hours to build House B.***
- House C is almost three times the size of House A.
- As you might have guessed by now, it would roughly take about 30 hours to build House C in ***relation with size*** of House A.

- As shown in the example above, an item with one story point is half the size of an item worth two points. The relation goes like that.
- The following table shows commonly used range of Story Points

Story Point Value	Abbreviation (based on T-Shirt sizes)	Description
0	Freebie, effort required is negligible, not significant	
1	XS	Extra Small
2	S	Small
3	M	Medium
5	L	Large
8	XL	Extra Large
13	XXL	Double Extra Large
20	XXXL	Huge
40	--	Very Huge
100	--	Very Very Huge
?	Effort can't be identified	More details are required

- The range (as shown above) is selected by the team. If the value range and size are in relation, they can use that range of values.
- ***Please note: since these story points are relative and arbitrary, the Story Point range of one team can't be used with other teams.***

Planning poker

- Estimation is done by ***all team members*** in Agile.
- Planning Poker is a technique used by Agile Teams in estimating User Stories.
- In the following section, let us see how to play Planning Poker.
- **Required Items:** Each team member must be given a set of Planning poker cards
- **Facilitator:** Scrum Master facilitates this event
- **Participants:** Mandatory: Scrum Master, Development Team Members (all), Product Owner Others may include Subject Matter Experts, Functional managers etc.
- **Prerequisite:** Prioritized Product Backlog
- **Timing:** As part of each Sprint Planning
- **Steps:**
    - Product Owner takes the highest prioritized Product Backlog Item and reads it aloud for team members. If team members have any questions, the Product Owner should provide the required clarifications.
    - Each team member will pick a Planning Poker card (each card will have a value like the ones shown in the table above). ***Once a team member picks his/her card, it will not be shown, but it will be kept face down.***
    - Scrum Master checks that all members have chosen their cards (remember these cards contain the value that indicate the Story Point). After confirmation, Scrum Master will ask everyone to show their cards.
    - ***The team members who picked the smallest and highest Story Point values are asked to explain the rationale behind their values.***
    - All team members can ask questions and seek clarifications. Once it is done, a common value is agreed upon and that value is assigned to the selected User Story.
    - Product Owner picks the next high priority User Story and the process repeats. This is done until team capacity is filled in.

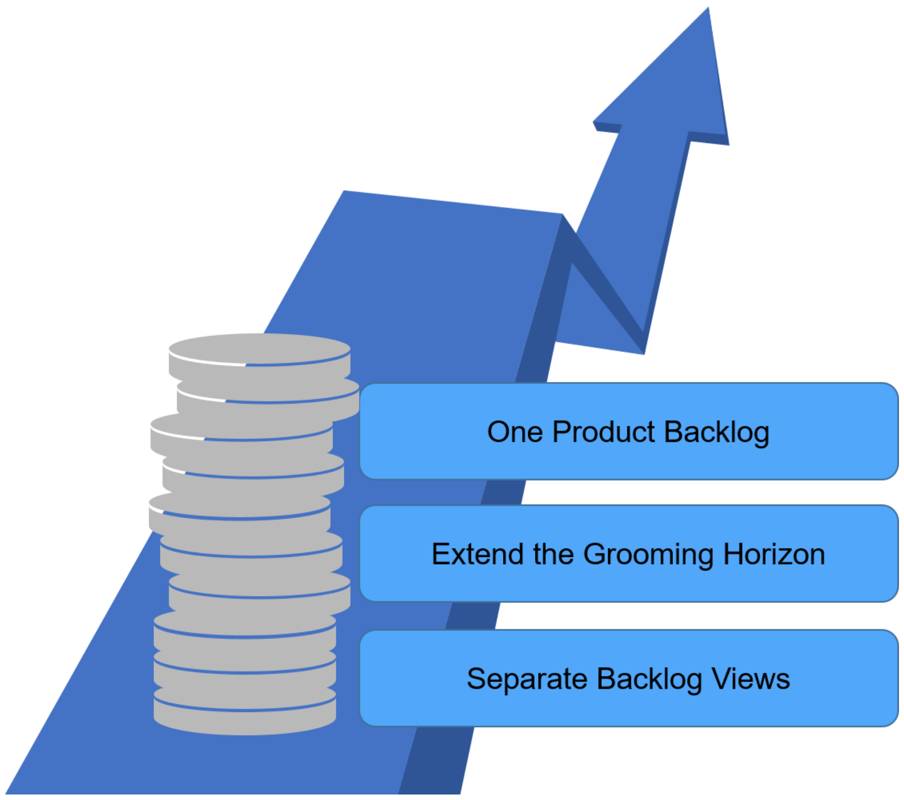
Dealing with Non-functional requirements

- Operational requirements, qualities of the system, and constraints are known as non-functional requirements.

Example	In an IT System, the following can be examples of non-functional requirements: • User should be able to login within 3 seconds. • 99% of transactions must be completed within 2 seconds. • The system should be able to provide an audit trail of user actions carried out such as changes to data, deletion of records, etc.

- For each Non-functional requirement, User Stories can be created and added, prioritized in Product Backlog just like for any other item.

Product Backlog Scaling



- Large projects across an enterprise have one common challenge – how to scale the Product Backlog?
- The following three techniques can be used to scale the Product Backlog across an enterprise in a large project:

ONE PRODUCT BACKLOG

- For a large project, use one backlog.
- Avoid the mistake of using multiple or team specific or component specific backlogs.
- Feed all teams involved in the same project with one backlog.

EXTEND THE PLANNING HORIZON

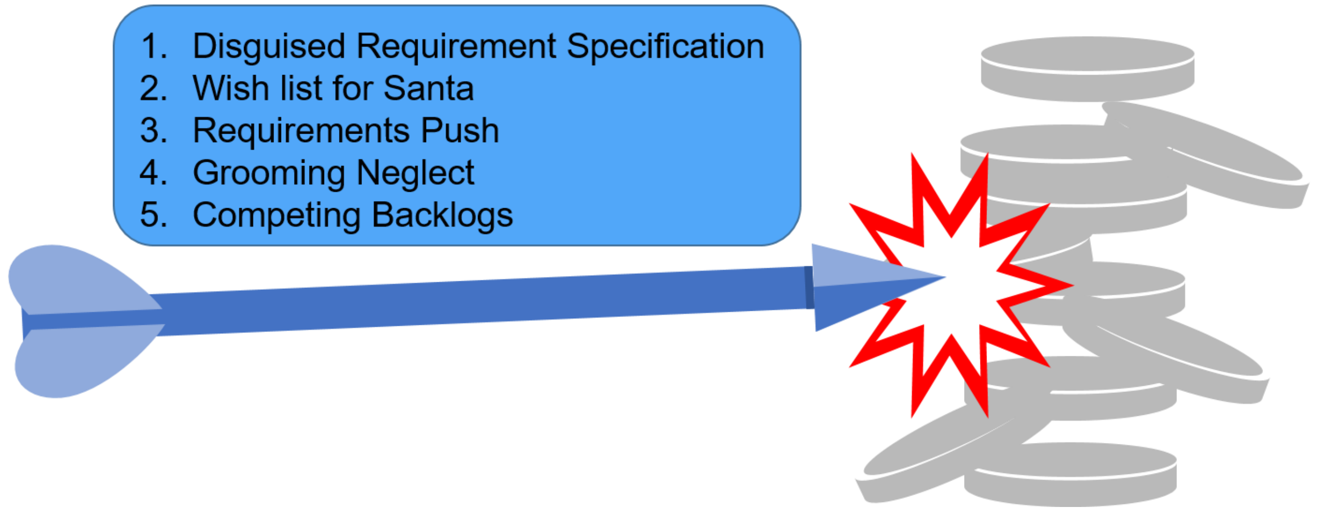
- Planning Horizon is the timeframe window within which you do planning the release.
- For smaller projects, this Planning Horizon can be for next sprint.
- For larger projects, smaller planning timeframe window would not serve the purpose well. Therefore, for large projects, extend the Planning Horizon to cover next two to three sprints or the next quarter.

SEPARATE BACKLOG VIEWS

- Large Agile Projects can leverage the advantage of separate backlog views for different teams.
- Please note: these are just views – not backlogs themselves.

- Today's electronic backlog management tools allow administrators to create multiple views easily by applying filters on data shown to a set or group of users.

Common Mistakes



1. Disguised Requirement Specification
2. Wish list for Santa
3. Requirements Push
4. Grooming Neglect
5. Competing Backlogs

DISGUISED REQUIREMENT SPECIFICATION

- In a few projects, the detailed requirements might be already documented in waterfall style documentation such as Software Requirements Specification and/or Business Requirements Document etc.
- Please note these exhaustive requirement specification documents are not a substitute for Product Backlog.
- Due to the detailed nature of these documents, they discourage the required face-to-face communication between Agile team members.
- Most likely, due to such detailed nature of requirements, they appear frozen or fixed—again losing out of Agile's capability of harnessing change.
- To combat this common mistake:
    - Check if there is Product Vision. If it exists, check the Product Backlog. If it is a disguised Requirement Specification Document or Use Cases (for example), work with Product Owner to create the necessary User Stories with required attributes.
    - If there is no Product Vision, start with a Product Vision first. Then execute the above step.

Example	In a customer supplier IT Application development engagement, the customer insisted on using Use Cases instead of Product Backlog because they were already created in detail by Customer Business Analysts. Instead of refusing outright the Use Cases, Supplier teams created User Stories from the Use Cases and populated them into an electronic backlog management tool. Cross references were provided for both Use Cases and User Stories. This helped the Customer to

	quickly realize the power of User Stories. More communications and discussions ensured that Customer could get Changes prioritized quickly and implemented easily. User Stories also helped the customer to forecast the Project progress much better than with Use Cases. Velocity and Release burn down charts helped the customer to assess project progress more accurately.

WISH LIST FOR SANTA

- Some Organizations create the Product Backlog with ***all*** the features of the future product. Sometimes, they also include outstanding work.
- Such a backlog is difficult to prioritize, and it limits the ability to adapt to customer feedback.
- To combat this common mistake:
    - Identify only the critical features as outlined in Minimum marketable product earlier
    - Discard any requirements that might appear as a wish list

REQUIREMENT PUSH

- In a typical Customer-Supplier engagement setting, there is a possibility that the customer's Product Owner might include Product Backlog items ***without*** the team's knowledge just at the onset of Sprint Planning meeting.
- This might come as a surprise to the team, thereby defeating the collaboration purpose of Agile with Product owner.
- To combat this common mistake:
    - Set up grooming sessions for each sprint with Product Owner
    - Allocate time for meetings between Product Owner and team (daily if required)

GROOMING NEGLECT

- Remember that ***a well-groomed Product Backlog is a mandatory prerequisite*** for Sprint Planning meeting.
- During Sprint Planning meeting, if the Product Backlog is not well groomed, the Sprint Planning meeting itself will not serve its purpose—shared goal might not be established, and the customer's high priority items might not be addressed correctly.
- To combat this common mistake:
    - Insist on backlog grooming prior to Sprint Planning meeting; do not start the next sprint unless backlog is groomed and ready.

COMPETING BACKLOGS

- There is a possibility that in a large project, there could be multiple products or services to be developed as part of the Project.

- This may result in multiple Product Backlogs.
- Project teams working on multiple Product Backlogs for the same project may not have cohesive sprint goals, may do a lot of task switching, and ultimately the customer may feel that no progress has been made as the team is working on multiple products and not focusing on delivering value to the customer.
- To combat this common mistake:
    - When there is a possibility of multiple products, ask the teams to work on one product per sprint, that is, the entire team needs to focus on one product per sprint.
    - Also, a series of sprints (say 3 to 4 continuously) can be used to deliver one product and then move on to the next product (if business priorities permit)