Agile Software Development

Lecturer: Raman Ramsin

Lecture 7

Scrum: Product Backlog
Product Backlog Items (PBIs)

- The product backlog is a prioritized list of desired product functionality.
- The product backlog is composed of backlog items referred to as PBIs, backlog items, or simply items.
- PBIs can be of various types:
  - Features, which are Items of functionality that will have tangible value to the user or customer;
  - Defects needing repair;
  - Technical work;
  - Knowledge-acquisition work;
  - Any other work the product owner deems valuable.
## Main PBI Types

<table>
<thead>
<tr>
<th>PBI Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>As a customer service representative I want to create a ticket for a customer support issue so that I can record and manage a customer’s request for support.</td>
</tr>
<tr>
<td>Change</td>
<td>As a customer service representative I want the default ordering of search results to be by last name instead of ticket number so that it’s easier to find a support ticket.</td>
</tr>
<tr>
<td>Defect</td>
<td>Fix defect #256 in the defect-tracking system so that special characters in search terms won’t make customer searches crash.</td>
</tr>
<tr>
<td>Technical improvement</td>
<td>Move to the latest version of the Oracle DBMS.</td>
</tr>
<tr>
<td>Knowledge acquisition</td>
<td>Create a prototype or proof of concept of two architectures and run three tests to determine which would be a better approach for our product.</td>
</tr>
</tbody>
</table>

[Rubin 2012]
Characteristics of Good Product Backlogs (1)

- Four characteristics have been pointed out for good product backlogs, summarized by the acronym DEEP.

- INVEST criteria are for judging the quality of user stories; DEEP criteria are for determining if a product backlog has been structured appropriately:
  
  1. **Detailed appropriately:**
     - PBIs that we plan to work on soon should be near the top of the backlog, small, and detailed so that they can be worked on in a near-term sprint.
     - PBIs that we will not work on for some time should be toward the bottom of the backlog, larger in size, and less detailed.
     - As we get closer to working on a larger PBI, such as an epic, we will break it down until we get a collection of smaller, sprint-ready stories.
       - Disaggregation should happen in a just-in-time fashion:
         - If we refine too early, we might spend a good deal of time figuring out the details, only to end up never implementing the story.
         - If we wait too long, we will impede the flow of PBIs into the sprint.
       - We need to find the proper balance of just enough and just in time.
Characteristics of Good Product Backlogs (2)

- **DEEP** criteria (Continued):

  1. **Emergent:** As long as there is a product being developed or maintained, the product backlog is never complete or frozen.
     - It is continuously updated based on a stream of information: Its structure is therefore constantly emerging over time.

  2. **Estimated:** Each product backlog item has a size estimate corresponding to the effort required to develop the item.
     - The product owner uses these estimates as one of several inputs to help determine a PBI’s priority (and therefore position) in the product backlog.
       - Also, a high-priority, large PBI (near the top of the backlog) signals to the product owner that additional refinement of that item is necessary.
     - Estimates must be reasonably accurate without being overly precise.
       - Items near the top of the backlog will have smaller, more accurate estimates.
       - As it may not be possible to provide accurate estimates for larger items, you may choose to use T-shirt-size estimates (L, XL, XXL, etc.).
Characteristics of Good Product Backlogs (3)

**DEEP** criteria (Continued):

4. **Prioritized:** Although the product backlog is a prioritized list of PBIs, it is unlikely that all of its items will be prioritized.
   - It is useful to prioritize the near-term items destined for the next few sprints.
   - As new items emerge during development, the product owner inserts them in the backlog in the correct order.

[Rubin 2012]
Grooming

- To get a DEEP product backlog, we must constantly manage, organize, and administer the product backlog through a process called *grooming*.

- Grooming refers to a set of three principal activities:
  1. Creating and refining (adding details to) PBIs
  2. Estimating PBIs
  3. Prioritizing PBIs
Grooming: The What

[Rubin 2012]
Grooming: The Who

- Grooming is a collaborative effort led by the product owner and involving internal/external stakeholders, the Scrum Master, and development team.
  - Ultimately there is one grooming decision maker: The product owner.
  - Stakeholders should allocate a sufficient amount of time to grooming based on the nature of the organization and the type of project.
  - As a general rule, the development team should allocate up to 10% of its time each sprint to assisting the product owner with grooming.
- The team will help create/review emergent PBIs as well as progressively refine larger items into smaller items.
- The team will estimate the size of PBIs and help the product owner prioritize them based on technical dependencies and resource constraints.
Grooming: The When

- Grooming is an **ongoing** activity.
  - Initial grooming occurs as part of the release-planning activity.
  - During product development, the product owner meets with others at whatever frequency makes sense to perform ongoing grooming.
    - When working with the development team, the product owner might schedule either a weekly or a once-a-sprint grooming workshop during sprint execution.
    - Sometimes teams prefer to spread out the grooming across the sprint, rather than block out a predetermined period of time.
      - They take a bit of time after their daily scrums to do some incremental grooming; this grooming does not have to include all of the team members.
  - Teams naturally do some grooming as part of the sprint review.
Grooming: The When

[Diagram showing the process of grooming, including stages like Sprint planning, Sprint backlog, Sprint execution, Sprint retrospective, and daily scrum.]

Some up front then ongoing as necessary

Workshop during sprint

After daily scrum

Daily scrum

Sprint execution

Potentially shippable product increment

Sprint review

During sprint review

[Rubin 2012]

Department of Computer Engineering

Sharif University of Technology
Definition of Ready

- Grooming should ensure that items at the top of the backlog are ready to be moved into a sprint.

- Scrum teams can formalize this idea by establishing a **definition of ready**.

  - The definition of ready is a checklist of the work that must be completed before a PBI can be considered in a suitable state to be moved into the sprint.

  - A strong definition of ready will substantially improve the Scrum team’s chance of successfully meeting its sprint goal.
### Definition of Ready

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business value is clearly articulated.</td>
</tr>
<tr>
<td>Details are sufficiently understood by the development team so it can make</td>
</tr>
<tr>
<td>Dependencies are identified and no external dependencies would block the</td>
</tr>
<tr>
<td>PBI from being completed.</td>
</tr>
<tr>
<td>Team is staffed appropriately to complete the PBI.</td>
</tr>
<tr>
<td>The PBI is estimated and small enough to comfortably be completed in one</td>
</tr>
<tr>
<td>sprint.</td>
</tr>
<tr>
<td>Acceptance criteria are clear and testable.</td>
</tr>
<tr>
<td>Performance criteria, if any, are defined and testable.</td>
</tr>
<tr>
<td>Scrum team understands how to demonstrate the PBI at the sprint review.</td>
</tr>
</tbody>
</table>

[Rubin 2012]
Product Backlog: Release Flow Management

- The product backlog must be groomed in a way that supports ongoing release planning (the flow of features within a release).

- It is useful to actually partition the product backlog using two lines for each release, and thus partition the backlog into three areas:
  - **Must-have**: Items that we simply must have in the upcoming release or else we do not have a viable customer release.
  - **Nice-to-have**: Items that we are targeting for the next release and would like to include. If, however, we run short of time or other resources, we could drop nice-to-have features and still be able to ship a viable product.
  - **Won’t-have**: Items that will not be included in the current release.
Product Backlog: Release Flow Management

[Image of a diagram showing Must have, Nice to have, and Won’t have categories for items in a product backlog.]

[Rubin 2012]
Product Backlog: Sprint Flow Management (1)

- Proper product-backlog grooming is essential for effective sprint planning and the resulting flow of features into a sprint.

- When grooming for good sprint flow, it is helpful to view the product backlog as a pipeline of requirements that are flowing into sprints.
  - As items progress through the pipeline and move closer to when they will flow out to be worked on, they are refined through grooming.
  - If there is ever a mismatch or unevenness between the inflow and outflow of items, we have a problem.
    - If the flow of groomed items is too slow, eventually the pipeline will run dry and the team will not be able to plan and execute the next sprint.
    - Putting too many items into the pipeline creates a large inventory of detailed requirements that we may have to rework or throw away once we learn more.
  - Therefore, the ideal situation is to have just enough product backlog items in inventory to create an even flow but not so many as to create waste.
One approach that Scrum teams use to manage the flow is to have an appropriate inventory of groomed and ready items in the backlog.

- A heuristic that seems to work for many teams is to have about two to three sprints’ worth of stories ready to go.
- This extra inventory ensures that the pipeline will not run dry, and it also provides the team with flexibility if it needs to select PBIs out of order.
References
