BACKLOG REFINEMENT

SCRUM CHECKLIST

PREPARATION	SPRINT PLANNING (WHAT)	SPRINT PLANNING (HOW)	DAILY SCRUM (STANDUP)	**BACKLOG REFINEMENT**	SPRINT REVIEW	RETROSPECTIVE
1	2	3	4	**5**	6	7

Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Team collaborate on the details of the Product Backlog, refine and revise it. It ensures that the Team and Product Owner have sufficiently defined Product Backlog items so that Sprint Planning runs more smoothly and without surprises. A formal meeting or workshop activity mid-way through the Sprint ensures the Team and the Product Owner dedicate time to this Scrum event.

Time-box: 2 hours (total Backlog Refinement activities in the Sprint should not exceed 10% of the Sprint)

Agenda

- Scrum Master
  - Introduces the agenda.
  - Clarifies the rationale and the rules of the key meeting.
  - Indicates the time-box for the key meeting.
  - Indicates the time-box likely required for discussion of each Product Backlog item to get through sufficient Product Backlog items in the key meeting.
- Product Owner
  - Presents ideas and the likely vision for the *upcoming* Sprints.
  - Introduces the Product Backlog items and their Definition of Done.
- The Team
  - Discusses the Minimal Viable Product(s) (MVP) in the Product Backlog with the Product Owner.
  - Discusses each Product Backlog item that make up the MVP.
  - Estimates each of the Product Backlog items in turn using techniques like Planning Poker.
  - Discusses any points of divergence in relation to the estimate of effort for each Product Backlog items
  - Breaks-down the Product Backlog items into smaller pieces, through collaboration with the Product Owner. This ensures that each item be completed within a single Sprint. This also helps the team remove any unevenness that can lead to increases in wasted time and effort due to context switching.
  - Communicates to the Product Owner any consequence relating to *what* is being asked for and its rank-order for delivery, including any technical, design or business debt that may be incurred as a result.
- SMEs and other interested stakeholders
  - Observe.
  - May participate in estimation and discussion of the *what* through prior negotiation with the Scrum Master and the Product Owner to assist with increasing transparency of upcoming Products.
- Scrum Master

- Facilitates the meeting.
- Encourages participants to use established Patterns to break-down Product Backlog items.
- Encourages participants to create a level of granularity that ensures the top 20% of the Product Backlog contains fine grained requirements.

Key meeting participants

Role
Team
Scrum Master
Product Owner
Subject Matter Experts (by invitation)
Interested stakeholders (by invitation)

Key outcomes

- Higher ordered Product Backlog items are clearer and more detailed than lower ordered
- ones.
- Product Backlog is **DEEP** –
  - **“D”**etailed appropriately: To support conversation and commitment from the Team.
  - **“E”**mergent: Enabled to grow and be organised as new information arises.
  - **“E”**stimated: In terms of complexity.
  - **“P”**rioritised: Ranked, value prioritised.
- Stories should be:
  - **“I“**ndependent (of all others)
  - **“N“**egotiable (not a specific contract for features)
  - **“V“**aluable: Taking account of both organisational and end-user needs.
  - **“E“**stimable (to a good approximation)
  - **“S“**mall (so as to fit within an iteration)
  - **“T“**estable (in principle, even if there isn't a test for it yet)
- Granularity of the Product Backlog – Stories are of fine granularity (top 20%), Stories of medium-grained requirements (next 20%), and Epics (remaining 60%).
- Items are estimated and reviewed – effort and complexity by the Team, and value by the Product Owner. More precise estimates are made based on the greater clarity and increased detail; the lower the order, the less detail. Reviewing estimates is based on insight gained from delivering Stories in previous Sprints.
- Larger items toward the top of the Product Backlog are sliced or chunked into smaller pieces.
- ‘Importance’ and ‘priority’ are clarified in terms of value: Return on investment (ROI), business value, value to users, cost of delay, reduction in risk and other factors.
- Items in the next two Sprints worth of Product Backlog are (re)rank-ordered according to value.
- Known architecture, business rules and user-needs are articulated against the Product Backlog items.
- Known tasks are articulated against the Product Backlog items.
- Future capability needs are identified and knowledge transfer or training options are identified.
- Future capacity issues – leave, public holidays, part-time availability – are identified against future Sprints.
- Definition of Done for future Sprints is assessed, revised as necessary, and confirmed.

- An increase in knowledge is gained by the Team regarding what work is likely to come up at the next Sprint.
- Unknowns are explicitly identified and targeted for further investigation, analysis and definition.
- The Team agree that specific Stories are ready for selection for Sprint Planning.
- The Product Backlog is transparent to the Team and to stakeholders.

Key approaches and patterns

- **Top-down** – Start with the Project Vision and supporting documentation like the Business Case to populate the Product Backlog. Focusing on what will resolve current business issues in the next Sprint is the key. This approach creates what is analogous to the Product Breakdown structure from PRINCE2.
- **Story Mapping** – Examine the context and the steps required for a user to complete an activity. Break each activity into Stories.
- **Investigations & experiments** – After a formal, collaborative Backlog Refinement session, the Team members identify key areas in which to do some analysis, investigation and R&D. The goal of the activity should be clear and timeboxed with the outcome explicitly reported to the Team and recorded against the Product Backlog item. No actual work should be undertaken to progress a Story.
- **Design Spikes (options analysis)** – After a formal, collaborative Backlog Refinement session, the Team identifies which future Stories have significant unknowns and several potential ways of implementing them. These are targeted for a timeboxed options analysis with the results brought back and discussed with the Team at the next Backlog Refinement or future Sprint Planning session.
- **Cynefin** – Using the Cynefin sense framework by Dave Snowden, the Team assess whether Stories are ‘simple’, ‘complicated’ or ‘complex’. Patterns for Complicated Stories are identified. Design Spikes are identified for complex Stories. Best-practice is agreed by the Team for simple Stories.
- **To size or re**-size – If the Team has already sized a PBI, ask whether the Team has learned anything new by asking for a Roman Vote. If the Team has learned nothing new, and then move onto the next PBI.

Common obstacles to avoid

- People outside the Team consume the meeting with discussion about design or the solution.
- Too much time is spent determining *how* the Team will tackle the Product Backlog item over defining and clarifying *what* it is and the associated Definition of Done.
- General discussion of Stories rather than getting down to specifics.
- Discussion about Stories that are of low priority and unlikely to be included in the next one or two Sprints.
- In sufficient time is spent by the Team in preparing for the upcoming Sprint as a Backlog Refinement activity. As a result, too much time is taken up in Sprint Planning discussing scope boundaries of Stories.
- The Product Owner hasn’t prepared sufficient Product Backlog Items to discuss with the Team. The Product Owner should have at least two Sprints worth of items.

Using this Document

ZXM gives you permission to share and distribute this work under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

You are free to:

- Share – Copy and redistribute the material in any medium or format.
- Adapt – Remix, transform, and build upon the material.

Under the following terms:

- Attribution – You must give appropriate credit, provide a link to ZXM, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests that ZXM endorses you or your use.
- NonCommercial – You may not use the material for commercial purposes.
- ShareAlike – If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.