Rule = A definition of what a Product Backlog is.

Good Practice = A practice which is commonly done and is good to do.

Avoid = A practice which, in most cases, is recommended to be avoided. But, for almost all of them, it's possible to imagine a scenario where that practice would make sense.

Rule

The Product Backlog is expected to reflect the PO's best knowledge at any time, but items committed to Sprint Backlog are locked to give the team some stability to focus on getting things done.

Rule

The Product Backlog can contain anything related to the product, including items of low value that are unlikely to be ever implemented, or items otherwise outside currently envisioned scope.

Rule

Multiple teams can pull items from the same backlog, but then they also have to refine, plan, review and retrospect together. They are separate teams only during Sprint execution.

Rule

While owning the priority, wise Product Owners listen to their teams and stakeholders to support their prioritization.

Rule

It is unfair to expect teams to prioritize their work between Product Backlogs. Prioritization is the work of the Product Owner and cannot be off-loaded onto teams in any circumstance.

Rule

Without clarity of relative priority, things get out of hand quickly. Everything becomes “important”, but nothing really is. Even when everyone agrees that there is no difference in the priority of two items, one must be placed more important.

Good Practice

Scrum is silent on who can contribute backlog items. The PO always holds the right to prioritize them as they see appropriate. Asking stakeholders and Team to contribute is a great way to engage them in the product.

Good Practice

Without understanding of e.g. product goal (vision) or desired targets (release plan or roadmap), it's too easy to lose sight of doing the right thing (priority and value) or being on track.

Good Practice

Ideally, the Team should never take just one item into a Sprint. If something goes wrong with it, there is little space to adjust and still deliver something valuable from the Sprint.

Good Practice

Scrum is silent how to write the backlog items. User stories is a great way to describe user requirements / interaction. But other types of backlog items are also useful, e.g. bugs, improvement tasks or various events.

Good Practice

The Product Backlog Items do not have to be estimated, but they often are, and the Team is the only party who can provide the estimates. The PO and stakeholders may certainly ask why certain item is of certain size, and the Team must be able to explain.

Good Practice

This activity is called Backlog Refinement and is a regular part of every Sprint. It helps the team to tackle uncertainties before the Sprint and typically increases velocity significantly.

Good Practice

Normally, there should be enough refined items for the team to include in the next 1-2 Sprints. Items later than that get progressively larger and unclearer, since we do not want to unnecessarily invest effort into them too early.

Avoid

This should be very strongly avoided since it implies that the Team must prioritize their work between the PO's. Instead, have one of the PO's (or someone else) be the one PO and other PO's become stakeholders.

Avoid

The Team certainly discusses technical issues even before an item is taken into a Sprint, but going too deep and committing too early to certain solution should be frowned upon.

Avoid

Backlog items should not be engineering tasks or architectural items, since they are very hard to deliver in short Sprints. However, removing technical debt is a sensible backlog item, if it requires effort beyond the normal refactoring that happens during development.

Avoid

A very large number of stories or old stories suggest that there is little value in most of them. Thus it is often better to just remove them (or at least hide well) and reintroduce as new items (with updated definition) if the requests reappear.

Avoid

Scrum is very much about cross-functional teamwork, and separate estimates for different disciplines strongly take away from that. Thus the team should always estimate the full effort of the whole team in each item.

Avoid

In most projects, the users and other stakeholders keep recognizing new needs, and it makes often sense to save some of them. Those items will remain in the Product Backlog after the last Sprint, stored for possible continuation.

Avoid

The PO should actively encourage stakeholders and the Team to contribute their suggestions as new backlog items, and them prioritize them appropriately. Adding an item in the Product Backlog does not mean it will ever get done. Actually, most ideas won’t.

Avoid

The Team should provide feedback on the value and effort of such technical backlog items, but the Product Owner is always responsible for prioritizing them relative to all other items in the Product Backlog.

Avoid

While it is possible that in some project the Product Backlog is quite well formed in the beginning of the first Sprint, in most cases much less is sufficient to get started. Actively avoid the urge to define too much up front.

Avoid

While sometimes it makes sense to collect bug information into a separate tool, never assume that they are not part of the Product Backlog and must be prioritized relative to other items in it.

Avoid

While sometimes the Product Owner specifically wishes to include such items in the Product Backlog, normally such items would confuse viewers and are better handled with some other mechanism.

Avoid

It is never sufficient to group rank, but such categorizations or tags can be very useful ways in understanding and maintaining the Product Backlog.

Avoid

While it may often make sense for the Team to list their technical desires (e.g. removing technical debt), such items are always part of the Product Backlog and must be prioritized by the PO in relation to other items in it.

Please send feedback and improvement ideas to petri.heiramo@gmail.com.