Ready, Set, Go...and Ready Again: Planning to Groom the Backlog

In an agile project, the backlog--the prioritized set of requirements--is the main input to iteration planning. Many agile teams are as careful in specifying the "definition of 'ready'" as they are in specifying the "definition of 'done'." The product owner must ensure that priorities are thought through, stories are at the Goldilocks level of granularity ("not too big, not too small") and stakeholders are prepared to discuss details.

Getting the backlog ready and the related concept of "grooming the backlog" doesn't come for free. You need to plan and budget for this work. Here are five aspects to consider.

Keep Two Views of the Backlog

1. ***User Story Mapping*--The Business View:** This view lets the business stakeholders keep their eyes on the prize. In this view, they see user stories of all different sizes--some very large that are broken down into multiple levels of detail, some small and specific. Also, collections of stories are grouped into business scenarios. These dependencies among stories clarify how individual capabilities provide value to the users and stakeholders.
2. ***Prioritized List of Stories--The Classic View:*** The top items on the backlog get developed in the next sprint. The heart of iteration planning involves determining how far down the backlog each sprint will reach. To be ready for the upcoming iteration, the top items must be in "development-sized chunks". If an item is too large to be developed in a single, fixed-length iteration, it must be broken down into smaller partial stories.

These two views work together: While the prioritized list gives developers the roadmap for a particular sprint, the user-story mapping puts those development-sized chunks in context.

When is the work done?

Getting the backlog ready overlaps almost all the coding and testing. It begins shortly before coding starts to get "just enough" of the backlog ready. In many projects, those initial highest-priority items can be easy to find, even though prioritization down the line will be more contentious. In other projects, even those initial priorities require significant discussion; adjust the lead time before the first coding iteration accordingly.

Expect grooming the backlog to continue almost to the end of the release. Details about stories emerge over time. As stakeholders review already developed stories, priorities change and the backlog needs more grooming.

There are two milestones to aim for:

1. ***Minimally Marketable Features* Defined:** As you break down top-level features in the user story map, you eventually identify what portions of stories are needed for the release to be useful. These may still be too large to develop in a single iteration, so they may need to be broken down further. Other portions of the original features may also still be included, but these would have lower priority. You need to reach this milestone quickly, since it guides the prioritization of early iterations.
2. ***Release Backlog Finalized:*** Toward the end of the release, you've made the decisions of what stories will stay in this release and what can be deferred. The user story will remain stable after this point.

The Work Contour

There are four tasks to groom the backlog:

1. Adding and removing stories from the backlog
2. Refining the user story map: breaking down high-level features into more concrete stories and grouping them into business scenarios
3. Prioritizing “development-sized chunks” for each iteration
4. Specifying the details of the stories

These don’t all occur at a steady rate over the course of the project, and the people involved and their degree of involvement change. Early in the project, much of the work will be geared toward refining the user story map. This requires a high level of involvement from business stakeholders; getting consensus among the stakeholders is both difficult and critical. Their involvement will stay high at least until the minimally marketable features are defined. Stakeholders will still be needed to provide details, but once the main parts of the story map are stable this can be delegated to a working group of subject matter experts.

The development team will be involved at a fairly steady rate, primarily in discussions of story details. Overall, the work is front-loaded and intense until the minimally marketable features are defined, less intense as the User Story Map evolves, with only minor work remaining after the release backlog is finalized.

“Ready Again”: Plan for revisions to previous work

Stories developed in earlier iterations may need to be revised based on new requirements. For example, in an online commerce site, “checkout” may have been defined and implemented based on each registered user having a single address. Later in the project--when the story was added to allow a user to store multiple addresses--the checkout scenario needs to be revised. The business stakeholders will need to provide additional detail regarding business rules around billing addresses and shipping addresses.

Although many of us have been conditioned to avoid rework at all costs, Kent Beck years ago described some different economics in *eXtreme Programming eXplained*. By keeping things simple at first and adding complexity only later when you need it, you cover the cost of rework with two sources of savings:

1. You benefit from the simpler design making it faster to write code that will still work with the later, more complex design
2. You don’t spend the time designing for complexity you may never need (The “YAGNI” Principle: “You Ain’t Gonna Need It”)

Beck was primarily describing coding techniques such as refactoring and automated unit testing, but the same principles apply to getting the backlog ready for development: don’t spend time debating details abstractly until you know you need them. The time saved exploring in advance what *might* be needed makes up for the time spent later reworking only what *is* needed.

Planning and accounting for the work

When you are estimating the cost and resource requirements for a project or product release, include the work to groom the backlog:

1. Effort can be estimated as a proportion of development effort. Use history of past projects (either from your organization or industry-wide) as a guide. The proportion may be different for different types of

projects, so be sure to compare projects that are similar in nature, with similar development methods.
2. Schedule time with business stakeholders who will develop the user story map and participate in discussions of story detail. They will be needed throughout the project, but not at a uniform rate. More work will be needed upfront, with more detail work at the later stages--likely from people with more specialized knowledge. Plan for rework as stories are revisited throughout the project.
3. Account for work that coders will do getting the backlog items ready. Expect discussions between the business stakeholders (perhaps represented by the product owner) and the coders.

Planning for this in advance will keep your backlog groomed and your agile project humming along smoothly.

References:

1. Agile Alliance, Backlog, http://guide.agilealliance.org/guide/backlog.html
2. Agile Alliance, Definition of Ready, http://guide.agilealliance.org/guide/ready.html
3. Rubin, Ken, “The Importance of the Product Backlog on a Scrum Development Project,” http://www.informit.com/articles/article.aspx?p=1928232&seqNum=5
4. Patton, Jeff, “User Story Mapping," http://www.agileproductdesign.com/presentations/user_story_mapping/index.html
5. Shalloway, Alan, “Minimal Marketable Features: The Why of Enterprise Agility”, http://www.youtube.com/watch?v=P6fDFZHd5RE
6. Beck, Kent, “eXtreme Programming eXplained”, Addison-Wesley, 2000