Agile
Planning
and Analysis

SYNERGIZING TO
DELIVER VALUE

by Ellen Gottesdiener
and Mary Gorman

Agile is about the continuous incremental delivery of valuable, market-ready software. Your agile team iteratively explores and evaluates product needs—commonly referred to as requirements—by planning and analyzing what to build, defining acceptance criteria, and then building and testing product increments. A crucial aspect of your work is planning—and planning to plan—while integrating just-enough, just-in-time analysis.

Analysis and planning are synergistic. They are coordinated efforts, and one feeds the other. Analyzing requirements deepens your understanding of product needs so that you can identify and select the most valuable ones. Planning is the allocation of those product needs into delivery cycles, given your limited capacity (people, time, money, resources). Together, planning and analysis seek to maximize business value.

Whichever agile or lean framework, method, or technique you use to analyze the backlog, you may also employ *artifacts*, such as personas, a data model, a story map, or business rules. These artifacts, kept as lightweight as possible, can be very helpful as the team explores, designs, builds, and tests a slice of the product.

Power of Perspectives

Many technical and business people think of requirements as specifications that get defined and then "thrown over the wall" to the technical people. But on agile teams, this classic view is altered. Product needs are explored and evaluated through a partnership of technical and business people so that team members can collaboratively understand and deliver business value.

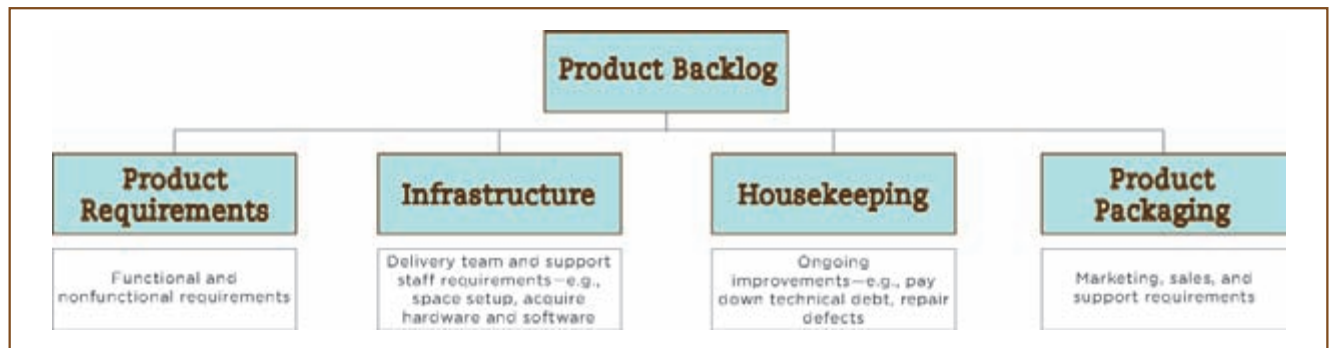It takes a shared understanding to plan and analyze

Figure 1: Product backlog categories

The Backlog: The Basis for Planning and Analysis

The *backlog* is a master catalog containing a prioritized list of unfulfilled product needs at varying levels of granularity. Figure 1 shows one way of categorizing backlog items.

Typically, most of your backlog items will be product requirements in various formats: user stories, one-line titles or story descriptions, drawings or sketches, and so on. Items in a healthy agile backlog are what Roman Pichler calls DEEP: detailed appropriately, estimated, emergent, and prioritized [1]. Note that "detailed appropriately" means that, at any given moment, some items will be highly detailed and others less so.

The backlog is dynamic. Items are added, removed, altered, reprioritized, deferred, decomposed, or prepared as needed. This ongoing planning and analysis of backlog items is known as *grooming*, *pruning*, or *refining* the backlog. As you do this work, you must be vigilant to ensure that the backlog items align with the product's vision and business goals, realizing that goals may change over time as the organization, market, and competitors evolve and you get feedback from users.

Grooming maintains a runway of product needs that are ready to pull into planning for the next and future delivery cycles. The trick is to balance current and future planning. We find that teams typically work two to four iterations ahead; the further ahead you're planning, the less detailed the requirements will be.

product needs, including perspectives from cross-functional disciplines. The team needs to incorporate the input of the internal and external stakeholders listed in figure 2.

To lead the exploration and analysis, many agile teams rely on a few people who have strong analysis and domain skills. In our experience, people with these skills include business analysts, product managers, testers, user experience experts, and the like.

The Three Views of Product Needs: Plan and Analyze the Backlog

As you pull and evaluate items from your backlog, the key concept is that the level of detail of any item will vary depending on the amount of lead time in your planning. The closer you are to building a product need, the more detailed it should be. You can't know all the details of all the backlog items up front, so you sketch out the long view of the product to establish a common focus and marshal organizational resources (people, money, space, governance). Then, you iteratively define what to build now and what to build next.

Product champions and development teams tend to think of and refer to the product backlog from three points of view based on where a given product need is within the timeline of the development cycle, as shown in figure 3. We call these three views the *big-view*, the *pre-view*, and the *now-view*.

Planning and analysis get increasingly fine-grained as you descend the view hierarchy. The big-view idea gets more re-

Internal Stakeholders

- Sponsor
- Product champion
- Business, project, and product managers
- End-users
- Indirect users—e.g., performance improvement staff, down/upstream interfacing systems staff and management
- Delivery team members—e.g., architects, data and database administrators, business analysts, developers, testers, user experience experts, etc.
- Support staff—e.g., help desk, installers, operational staff, maintenance staff, trainers, network operations, etc.
- Advisers—e.g., compliance, HR, legal, subject matter experts
- Sales and marketing staff

External Stakeholders

- Buyers (purchasers)
- Direct end-users
- Regulatory agencies
- Competitors
- Third-party suppliers and partners
- Shareholders and investors
- General public
- Media

Figure 2: Stakeholders who offer perspectives on the product backlog

View	Description
Big-view	• Provides a holistic understanding of product needs you may deliver over a long time horizon to satisfy the product vision and business goals
Pre-view	• Defines product needs for the near term
Now-view	• Defines immediate product needs in sufficient detail to estimate, build, and test what will be delivered next

Figure 3: The three views of product needs

fined at the pre-view level. Then, in the now-view, it is sufficiently detailed so that the delivery team can estimate the next delivery cycle, develop acceptance tests, and design the solution.

The big-view

A product need starts at the big-view as a general idea—for example, a feature that you think will fulfill some aspect of the product vision.

Note that the big-view reflects how the new product will fit with the other products in your organization's portfolio. Thus, the big-view should align with your organizational strategy and should be feasible.

The pre-view

The pre-view defines product needs in enough detail to support planning for the next release. The pre-view is informed by the big-view and the product roadmap (an artifact that describes your planning decisions). The pre-view for delivery teams with short release cycles can be as near term as a day or week. For other teams, it is one to several months. Each release contains chunks of consumable, marketable, valuable features.

The now-view

The now-view describes product needs in sufficient detail that your team can make reasonable estimates of the work

needed to deliver those requirements to a predefined set of acceptance criteria (conditions of satisfaction). The now-view is informed by your release plan and your most recent retrospective. Once you've planned your now-view and are in the midst of delivery, the now-view entails analyzing product needs under development sufficiently to design, test, and deliver the product.

Analyzing the Backlog Across Seven Product Dimensions

In partnership with the product champion and other stakeholders, the team works together to analyze the big-view and pre-view backlog items in just enough detail to estimate and plan. For now-view planning, the delivery team analyzes and estimates the activities needed to build, test, deliver, and validate the high-value product needs.

To assist in this analysis, we've developed the seven product dimensions shown in figure 4. They serve as a checklist that technical and business partners can use to make sure they're analyzing all the key aspects of a product backlog item. The seven product dimensions help you structure daily conversations about backlog items and give you a comprehensive understanding to guide planning. You can use this construct to analyze work in all three views (big-view, pre-view, and now-view).

The team looks at a backlog item, asking—in any order—Who is the user? What actions does the user perform with this item? What data goes in and out? What controls does this item enforce? What users, systems, and devices does it interface with? What quality attributes must be satisfied? What are the design and implementation constraints that apply?

There are a variety of ways to structure this conversation. We like to use an explore-and-evaluate technique in facilitated workshops; the product partners collaboratively identify timely product options (explore), assess the options and identify the highest-valued ones (evaluate), and slice each product need based on value.

In this way, the team and product champion explore and evaluate options for each product need at the level of detail appropriate to the view. This approach engages you in creatively using right-brain thinking—visualizing options, relationships, dependencies, and flow. You might start by working on the wall, creating lightweight, organic analysis models such as a data model, state diagrams, a context diagram, a dependency map, decision tables, prototypes, and so on. Documentation, too, can be lightweight—posters, photos, wikis, and the like—to serve collective memory. These artifacts may be temporary or, if valued by the partners, may become the basis for building and packaging the product.

Throughout the conversations, the entire team gains crucial knowledge. As the product champion assigns a value to each product need, he asks the team to gauge the effort it will take and the risks of implementing it, improving his understanding of technical concerns and the development process. As team members discuss the product needs, they learn more about the business context for the work. Team members are responsible for questioning, challenging, and clarifying the product champion's filtering criteria, deepening their understanding of the business domain and the options that will provide value.

Together, team members balance their understanding of options for each product dimension of each backlog item with a hard-nosed assessment of which options are most valuable for the next delivery cycle. Ongoing analysis grooms the backlog and shapes the plan for each view.

Successful Synergies

Agile planning and analysis are interdependent and synergistic. Working hand in glove, they provide stakeholders with a flexible structure for continuous delivery of value. {end}

ellen@ebgconsulting.com
mary@ebgconsulting.com

Sticky Notes

For more on the following topic, go to www.StickyMinds.com/bettersoftware.
- References

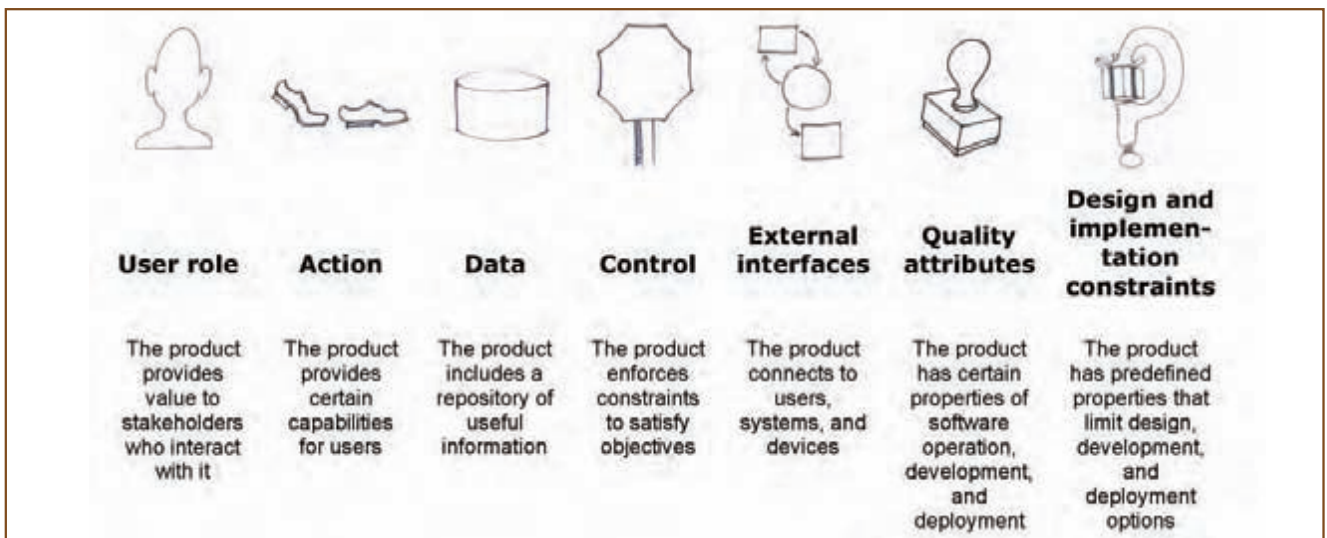| User role	Action	Data	Control	External interfaces	Quality attributes	Design and implementation constraints
The product provides value to stakeholders who interact with it	The product provides certain capabilities for users	The product includes a repository of useful information	The product enforces constraints to satisfy objectives	The product connects to users, systems, and devices	The product has certain properties of software operation, development, and deployment	The product has predefined properties that limit design, development, and deployment options

Figure 4: The seven product dimensions

Ranorex 3.0

GRAZ, AUSTRIA—Ranorex 3.0 has been released and is now available for download. This new version features a test suite view that allows easy test case management and simplifies the combination of recordings and code modules within tests. Ranorex Studio now supports data-driven testing via connectors to SQL, CSV, Excel, and simple internal data tables.

The test suite view simplifies the combination of simple modules and recordings within single test cases. In addition, one can directly connect test data—stored in tables or external data sources—with variables used as place holders within modules and Ranorex repositories.

One important goal of the new version is to reduce effort in maintaining test automation projects. The improved Ranorex repository browser now provides new functionality, allowing you to quickly search for and filter items and to clean up the repository in case of unused elements. In addition, the use of variables within RanoreXPath simplifies test automation of language dependent user interfaces.

ranorex.com/products/whats-new-in-ranorex-3.html

eggPlant Mobile Testing Solution

BARCELONA, SPAIN—TestPlant, in partnership with Real-VNC, has launched the eggPlant mobile testing solution, which extends eggPlant's reach into software running on smartphones and tablets across the major mobile operating systems.

eggPlant uses RealVNC's VNC Mobile Solution to connect directly to a device. A true user experience test is created, as eggPlant is capable of recognizing icons, colors, text, and fields on any display. A test script is automatically generated, as eggPlant is trained to move around an app using its image store to spot and compare display prompts. Multiple tests become effortless, and eggPlant reports test failures with helpful screen images of the exact point of failure. An automated test script can be created on one device (or emulator) to be repeated many times on other devices.

testplant.com

CollabNet TeamForge Project

BRISBANE, CA—CollabNet has announced the general availability of CollabNet TeamForge Project, an integrated suite of web-based development and collaboration tools for agile software development. The general availability release follows a four-month beta test by more than one-hundred organizations. TeamForge Project is a component of CollabNet's Codesion Professional Edition product, a secure, instant-on developer cloud that enables developers to instantly provision Subversion, Git, and other code repositories, as well as bug tracking, agile management, and full application lifecycle management capabilities.

open.collab.net/cloud

Sitefinity 4.1

WALTHAM, MA—Telerik has revealed details of its latest release for Sitefinity, the company's robust web content manage-ment system. The Sitefinity 4.1 release introduces a number of upgrades and improvements, including enhancements delivering performance increases of up to 35 percent. Also being debuted is support for Windows Azure, Microsoft's cloud services operating system, and native language packs for international Sitefinity users.

Enhancements incorporated into the Sitefinity 4.1 release include a new intelligent auto-caching algorithm enabling faster page loading and serving for websites; an upgraded virtual path provider, which streamlines and accelerates serving of dynamic content; and smarter auto-caching and an improved virtual path provider.

New language packs, providing authors, content managers, and content creators with a broad variety of native language support for backend and administrative tasks, are also being launched. Other features, upgrades, and enhancements in this release are:

- Comprehensive lists module
- Customizable user profiles for storing and displaying user-specific information
- Support for page redirects and multiple URLs
- Shared content blocks permitting greater content reusability
- Improved ORM mapping for greater extensibility

sitefinity.com

TestRail 2.3

Gurock Software has announced the release of TestRail 2.3. TestRail helps software development teams and QA departments manage, track, and organize their software testing efforts. TestRail 2.3 comes with major scalability and performance improvements for large teams and introduces new optimized views for test suites and test runs, and many other improvements.

With more than sixty changes, improvements, and bug fixes, TestRail 2.3 is the biggest update of the product since the release of TestRail 2.0. The new version comes with alternative views for the test suite and run pages that make it easier to work with larger numbers of test cases. An optimized user interface with a new section tree simplifies navigating and organizing test suites and rearranging test cases and sections via drag and drop.

The new version comes with improved support for third-party tools and Internet Explorer 9, and a new defect plugin for Axosoft OnTime to directly push and look up issues and bug reports. Other improvements include optimized Excel exports, easier attachment backups, and a new way to customize the subjects of email notifications.

gurock.com/testrail

GUI*dancer* 5.0

BREDEX GmbH has announced that version 5.0 of GUI*dancer* has been released and is available for immediate download.

GUI*dancer* 5.0 contains the new Test Style feature to offer