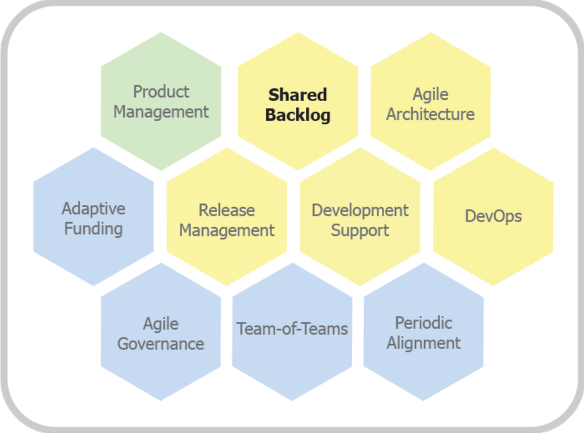
Shared Backlog Essentials

Part of the IJI Agile at Scale Practice Pack



Agile at Scale practices provide a starter kit that describes key common aspects of scaled agile development. Each practice contains cards that provide succinct advice on how to adopt and apply the practice

Practice Overview

Prioritize and marshal the delivery of product value requiring the collaboration of many teams.

Activities – the things we do

- **Refine Shared Backlog:** Get and keep the Shared Backlog visible, up-to-date and in good working order, with high priority items agreed and well understood.
- **Prepare a Shared Backlog Item:** Get a Shared Backlog Item ready for development by defining its constituent backlog items and how they will be integrated and tested.
- **Get a Shared Backlog Item Done:** Constituent backlog items are built, tested, integrated and it is verified that the Shared Backlog Item is correctly implemented.

Alphas – the essential elements that we progress

- **Shared Backlog Item:** An independently-releasable and valuable outcome that requires development contributions from many teams.

Work Products – the things that we work with

- **Shared Backlog:** Ordered list of outcomes that requires coordinated effort of many development teams.
- **Test Case:** Defines test inputs and expected results to evaluate whether a Shared Backlog Item is fully and correctly implemented.

Patterns - supporting practice guidance

- **Shared Definition of Done:** Have a clear definition of what it means for any Shared Backlog Item to be fully completed and ready for release.
- **Test-First Integration:** The Product Backlog Items that need to be developed in order to implement a Shared Backlog Item are specified by defining the tests that must pass for each to be considered "done".
- **Continuous Cross-Product Integration:** Code from different component products is continuously integrated so that there is always a tested, working integrated build of the composite releasable product.
- **Calibrated Estimating:** Where many teams work to deliver a Shared Backlog Item, we need to be able to compare like-with-like across team estimates, so we need a shared, calibrated measurement unit.
- **Dependency Management:** When many teams are working to evolve a complex product, dependencies between teams need to be managed.
- **Priority Alignment:** The priority of completing a Shared Backlog Item may need to be balanced with other items on the Product Backlogs of the different teams.

Creating winning teams.

Resources - referenced external sources of information and content

- This practice description uses the OMG Essence standard, with key concepts like Activities, Work Products, Alphas and Patterns being defined by this standard (http://www.omg.org/spec/Essence/).
- ***Shared Backlog***: the concept of a ***backlog*** as an ordered to-do list is common across many agile approaches (see for example http://guide.agilealliance.org/guide/backlog.html). A ***Product Backlog*** is part of Scrum (see for example http://www.scrumguides.org/scrum-guide.html#artifacts-productbacklog). The concept of a Product Backlog that is shared across multiple teams is a standard part of many scaled agile models and frameworks. Dean Leffingwell calls this a "Program Backlog" in his book *Agile Software Requirements* [Pearson Education Inc. 2011]. Large-Scale Scrum simply calls it "the Product Backlog", as its role remains unchanged irrespective of how many teams are collaborating to deliver the product. (See for example *Practices for Scaling Lead and Agile Development* by Craig Larman and Bas Vodde [Addison-Wesley 2010]).
- ***Calibrated Estimating***: Dean Leffingwell talks about the challenge of normalizing velocity in Chapter 8 of his book *Agile Software Requirements* [Pearson Education Inc. 2011]. Leffingwell suggests a "Hybrid Model" estimating approach for managing this challenge, as opposed to the Gold Standard model suggseted in this practice.