VERSIONONE

ENTERPRISE AGILE MADE EASIER

Andy Powell and Lee Cunningham

INTRODUCTION

Delivering large, complex software systems is a major challenge for any organization, including those that have embraced agile software development. The popular agile methodologies, such as Scrum and Kanban, which work well in small team environments, don't readily address many of the issues that arise. For example, how do you respond when a critical initiative, which requires work in one team's area of expertise, exceeds the team's capacity? Or, how much detail is required before funding an initiative? With the amount of uncertainty in most software projects, questions such as these are continually asked. Unfortunately, different parts of an organization often use ineffective approaches to answer such questions and thereby limit their ability to add value to the business.

> Enterprise agile is the holistic application of agile principles in a large and complex software development organization that develops interrelated systems.

Enterprise agile is the holistic application of agile principles in a large and complex software development organization that develops interrelated systems , or as they are sometimes called, "systems of systems". At enterprise scale, strong leadership focused on an enterprise-optimized approach is a necessity to consistently deliver high-value, high-quality software within the given business constraints. Agile principles provide the solid foundation upon which software development leaders can implement such an approach. This document introduces key enterprise agile concepts that allow software organizations to reach their full potential.

CONTENTS

Introduction 1

Portfolio, Program and Project Management Overview 2

Agile Management Principles 3

- Value Focus 3
- High Performance Teams 4
- Tiered Workflow 4
- Continuous Improvement 6

Conclusion 6

Sources 7

PORTFOLIO, PROGRAM AND PROJECT MANAGEMENT OVERVIEW

As depicted in Figure 1, the traditional concepts of portfolio, program, and project management provide a useful framework for categorizing the types of management activity that occur in enterprise agile. The original goals for each of these management disciplines are still valid. The opportunity for improvement, though, lies in reconsidering the methods by which an enterprise attempts to accomplish those goals. For change agents, this is the common ground from which they can create a new vision for success.

Here is a brief summary of each management layer:

- **Portfolio Management** – assess, prioritize, and select business initiatives to pursue that will create the most value for the company. A key question for portfolio management is: Are we making investments that will allow us to best capitalize on our strategic objectives? Portfolio management authorizes the funding of business initiatives and reviews the progress of them in order to provide governance over spending. Enterprise agile requires portfolio management to select business initiatives with less information. However, it provides better insight for governance by demonstrating progress via working software.
- **Program Management** – coordinate resources in the most efficient manner to deliver on the selected initiatives. Program management is the glue that links business strategy to execution. Program management decomposes business initiatives in order to identify specific deliverables and the resources required to deliver them. In addition to driving the execution of the decisions made at the portfolio level, program management is also responsible for updating the portfolio level with information that might affect the value expected from a business initiative (e.g., the initiative will not be delivered on the expected date). With enterprise agile, program management is usually focused around a particular product or product line. A portfolio may contain one or more programs.

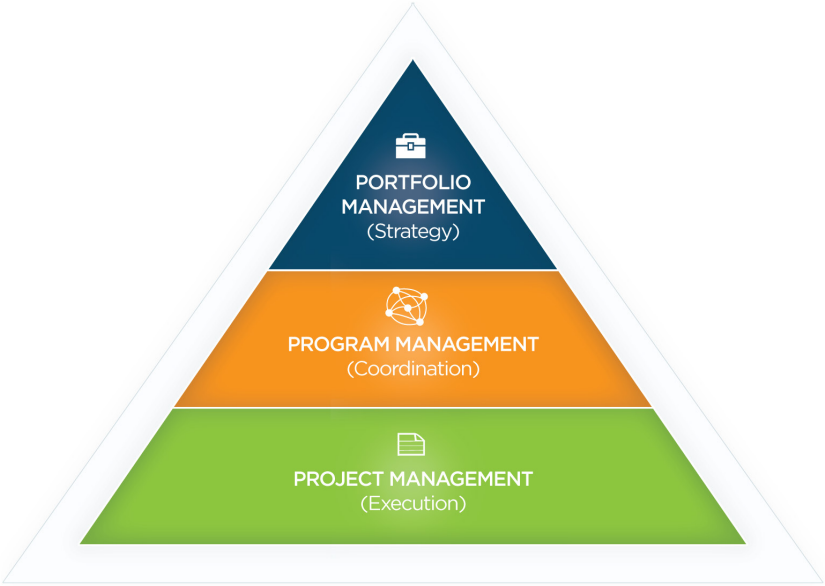

Figure 1: Three Level Management Framework

- **Project Management** – organize people and tools in the most effective manner to deliver high-quality, working software. Project management is the most tactical level of management and is responsible for managing creation of the actual product. Additionally, project management must inform program management about the status of the projects.

Regardless of an organization's size, these three levels of management activity are occurring, whether formally or informally. For a development organization with a single agile development team, lightweight portfolio and program management can be performed by a single person. In a large organization, each level will require many people, and some people may have responsibility in multiple layers. The natural tendency is for the decision makers to optimize at their respective levels, neglecting a more systemic end-to-end approach. This tendency is one reason why many large agile software development implementations fail to meet expectations. Implementing enterprise agile requires both a deep understanding of agile management principles and the determination to apply them at all levels so that the organization's deliverables are aligned with high-level objectives that meet its customers' needs.

AGILE MANAGEMENT PRINCIPLES

Although these management layers have different purposes and unique challenges, they also have many similarities. Capitalizing on those similarities will allow the different layers to better understand and communicate with each other, and can be a powerful way to reinforce agile principles throughout the enterprise. A simple, unifying management pattern for enterprise agile is:

> *Value-focused leaders provide vision so that teams can collaborate on work items that flow as process-focused leaders facilitate continuous improvement.*

Balanced, focused leadership is a critical success element for delivering large, complex systems. If an organization doesn't perform adequate analysis of a business initiative, it may rapidly deliver features that nobody wants to use. Likewise, if an organization accepts an inefficient process, it may be unable to execute on a solid strategy. As an organization improves its speed of delivery, any misalignment becomes more apparent and much more difficult to resolve. Leaders need to be aligned along a value-focus and process-focus as shown in Figure 2. The following sections outline each of the four principles that enable this management pattern: value focus, high-performance teams, tiered workflow, and continuous improvement.

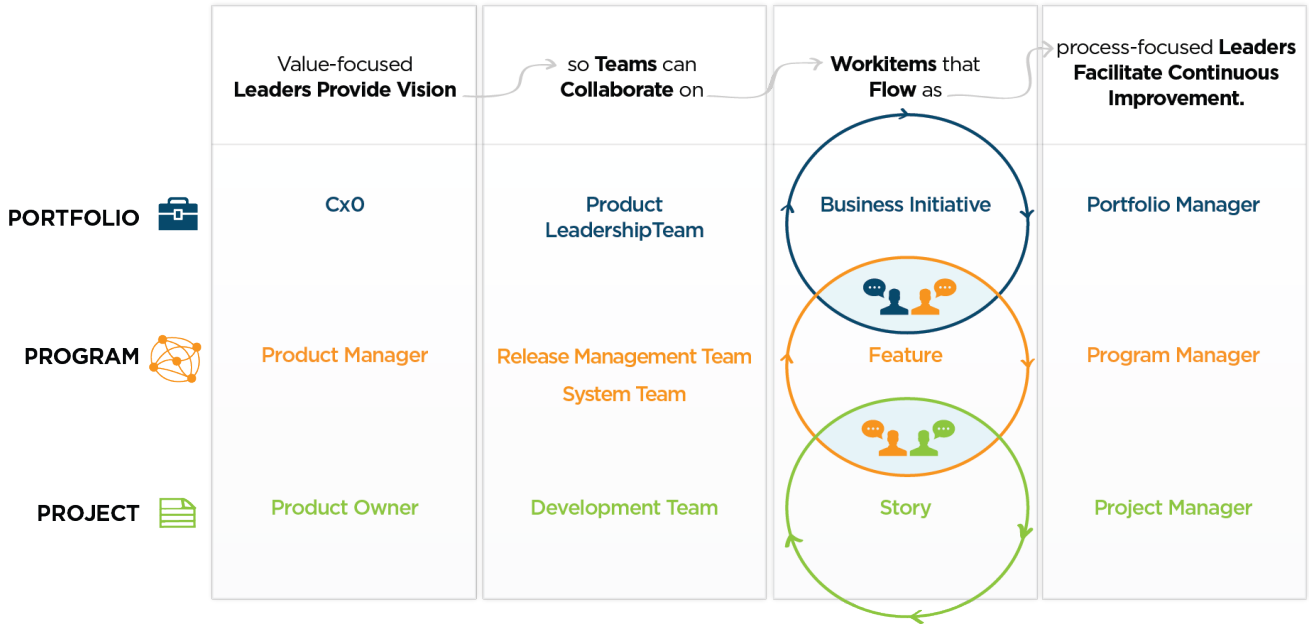

Figure 2: Management Pattern for Enterprise Agile

Value Focus

In an agile organization, teams don't just follow a plan and assume that it will provide value. Someone is focused on value, responsible for prioritizing the features according to market value, and responsible for ROI. In Scrum, that is the responsibility of the Product Owner, who also interfaces with a single development team as well as with customers. For large products or systems, a single person doesn't have the bandwidth to perform all of those functions. Therefore, the role often splits into two roles—a product owner (or business analyst) and a product manager. A product owner provides value-based direction to a single development team. A more customer-focused, strategic thinking product manager provides direction at the program level. Rather than having the responsibility for a single software development team, he or she is responsible for prioritizing a coarse-grained product backlog across multiple teams, each of which has a product owner. The product manager, then, provides rich customer insight and vision for multiple product owners.

When an organization has multiple products, there will be multiple product managers, and each product manager will naturally advocate for the customers in his or her market. Portfolio management is responsible for providing the proper balance of investment across products to capitalize on the market opportunities and maximize enterprise ROI. Additionally, cross-product initiatives that align to the company strategy may require investment in a product area that would not have been considered by a product manager otherwise, e.g., a change in underlying technology due to an acquisition. Figure 2 suggests the CxO as the visionary leader at the portfolio level because an executive is often asked to communicate broad vision across the organization.

Regardless of your specific organizational design, value-based decisions should be made at every level because agile software development is focused on continuous delivery of value to the customer. Moreover, since agile software development is inherently a process of continual discovery, the discoveries and decisions made at one level need to be communicated to the other levels.

High-Performance Teams

Because of this continual discovery, agile software development must be a highly collaborative effort. No one individual can know all of the details and communicate them to others continuously and effectively. Likewise, no one individual can understand all of the technology involved in order to deliver working software. Teams form to focus work effort and improve the communication process.

Organizations that practice enterprise agile recognize that high-performance teams require a clear purpose and empower teams to be self-directed in how they fulfill their purpose. Though team definitions are highly contextual, see Table 1 for some common team definitions:

Team Name	Members and Purpose	Management Level
Product Leadership Team	Business stakeholders, product managers, and development leaders that collaborate on portfolio direction	Portfolio
Enterprise Architecture Team	Senior technical leaders and architects that manage the technology strategy	Portfolio
Release Management Team	Product managers, business stakeholders, and program managers who determine the readiness of a product for release	Program
System Team	Testers and other people that build, integrate, and test a complete product or system (or subsystem)	Program
Development Team	Designers, coders, and testers that deliver working software	Project
Component Team	A type of development team that maintains a software component consumed by other development teams	Project
Feature Team	A type of development team that delivers product features	Project

Table 1: Common Team Definitions

> *A team should be as long-lived as possible, which requires giving them a purpose that is long-lived, as opposed to a specific deliverable.*

A team should be as long-lived as possible, which requires giving them a purpose that is long-lived, as opposed to a specific deliverable. For development teams, this approach has several benefits. First, it reduces the expense of forming new teams. New teams deliver slowly and the goal is high-performing teams. Second, it simplifies resource scheduling so program management can focus on teams instead of people. In other words, the human "resource" is a team, not a person. Finally, teams of stable composition are more predictable, allowing the organization to more credibly forecast duration and cost.

Tiered Workflow

Requirements decomposition is required for any software effort. What makes agile different is that it performs the decomposition with the goal of keeping the work items end-user focused as opposed to work-activity focused. This approach allows the value-focused leaders described earlier to provide clearer direction and make better tradeoff decisions.

In a large organization, each management layer is usually responsible for defining and owning the workflow at their particular level within an overall work item breakdown process. Although the naming of the work item types isn't standardized, the purpose of the different work items usually is. For this document, let's use the following definitions:

- **Business Initiative**—a development investment that is undertaken to achieve a specific business objective. Business initiatives are managed at the portfolio level and may impact one or more programs. A business initiative may provide end user value or deliver a necessary change to the system architecture. A business initiative contains one or more features.
- **Feature**—a product enhancement that a user would find valuable. Features are managed at the program level and usually do not impact other programs. One or more development teams may be required to deliver a single feature. A feature is comprised of multiple stories.
- **Story**—a small, independent, testable product enhancement that a single development team delivers.

Each level has a unique workflow and that is dependent on the other levels. See Figure 3 for an example. As a business initiative moves into Approval, the product manager begins to define features. Once features are defined, the product manager moves the highest priority feature into Breakdown, so the product owners know to begin creating stories. This progressive elaboration is performed over time by multiple people.

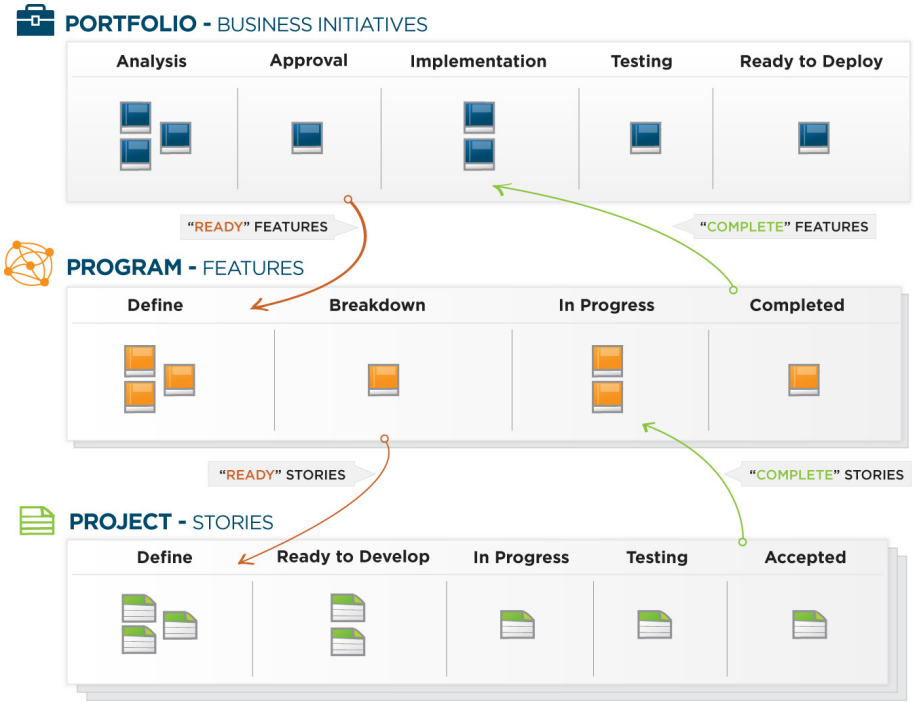

Figure 3: Three-Tier Workflow

Although work item elaboration provides knowledge that is essential for planning, it also pulls resources away from delivering software. Organizations that practice enterprise agile apply principles of product development flow[1] at each level in order to achieve optimal throughput. Those principles include:

- Making queues more visible, so they can be more effectively minimized.
- Limiting work in process, so that demand doesn't exceed capacity.
- Using short, regular planning and delivery cadences to limit the size of work items and adjust plans based on new knowledge.
- Synchronizing team cadences to facilitate communication and tradeoffs.

[1] "The Principles of Product Development Flow", Donald Reinertsen, Celeritas Publishing, 2009.

Kanban boards are a popular tool for visual management at all levels because they show progress, make queues visible, and reinforce work in process (WIP) limits.

Continuous Improvement

Process-focused leaders create a culture of continuous improvement. This requires leadership that is willing to both enforce the defined working agreements when necessary and quickly adjust those working agreements if they are reducing the organization's ability to deliver value. Many times a Program Management Office (PMO) is responsible for "owning" the process. Recognizing that self-organizing teams are more productive, an agile PMO gives development teams wide autonomy, but also realizes that local excellence has to serve the needs of the larger system.

Process-focused leaders create a culture of continuous improvement.

The role of the project manager in enterprise agile as depicted in Figure 2 is not the same as the traditional project manager role to which many are accustomed. Regardless of the title worn by the development team's project manager (ScrumMaster, Agile Project Manager, etc.), the Scrum framework's ScrumMaster role provides a useful guideline for the entire PMO:

- Exercise servant-leadership.
- Facilitate. Don't drive or coerce.
- Trust in the team's desire to do good work.
- Focus on removing impediments and doing whatever it takes to help the team reach its potential.

Within a large organization, many impediments cannot be solved within the development team alone, and the team's project manager must raise them to the program level. This role is also responsible for holding the team accountable for the team's and organization's working agreements, which includes ensuring accurate visibility into the team's progress. Rather than dictating process improvements, an effective project manager facilitates team retrospectives, jointly working with the team to both identify issues and implement solutions.

The program manager and portfolio manager exhibit similar behaviors at their respective levels. Although providing status on work progress is often a top priority, the process-focused leaders' passion is enabling the organization to improve its method of delivering the work.

CONCLUSION

A combination of more productive tools and an agile approach makes it possible for development teams to deliver stories quicker than ever before. It's time for program and portfolio management to take full advantage of this new capability! Unfortunately, speeding up the process leads to plans that have more uncertainty and bottlenecks that arise faster. No one said enterprise agile was going to be easy. Focusing the organization on an enterprise agile management pattern can make it easier.

Value-focused leaders provide vision so that teams can collaborate on work items that flow as process-focused leaders facilitate continuous improvement.

SOURCES

"Kanban: Successful Evolutionary Change for Your Technology Business", David J. Anderson, Blue Hole Press, 2010.

"Organizational Patterns of Agile Software Development", James O. Coplien and Neil B Harrison, Prentice Hall, 2004.

"Agile Project Management" (Second Edition), Jim Highsmith, Addison Wesley Professional, 2010.

"Scaling Lean and Agile Development", Craig Larman and Bas Vodde, Addison Wesley Professional, 2009.

"Agile Software Requirements", Dean Leffingwell, Addison Wesley Professional, 2011.

"The Principles of Product Development Flow", Donald Reinertsen, Celeritas Publishing, 2009.

ABOUT THE AUTHORS

Andy Powell

As the Product Evangelist for VersionOne, Andy helps large software companies understand how VersionOne software best fits into their Application Lifecycle Management tools strategy. During his thirteen year career in the software development industry, Andy has assisted in numerous 500+ person agile tools rollouts with companies such as Siemens, Adobe, EMC and Sabre. Andy is both a certified ScrumMaster and certified Scaled Agile Framework Program Consultant. He received a Bachelor of Science in Mechanical Engineering from the University of Notre Dame.

Lee Cunningham

Lee Cunningham is an experienced Agile coach, trainer, and practitioner who is passionate about working with organizations to help them consistently deliver high-value, high-quality products and services. Lee is energized by working with people at all organizational levels, employing a common-sense approach to help make Agile practices relevant and effective. He has trained and consulted with hundreds of teams in organizations of all sizes in the US, Canada, and the UK, and typically works with several different organizations each month. In his role as an Enterprise Agile Coach with VersionOne, Lee's focus is on enterprise-scale agile transformation.

VERSIONONE SNAPSHOT

VersionOne is recognized by agile practitioners as the leader in agile project management tools. By simplifying the planning and tracking of agile projects, we help teams deliver better software faster.

COMPANY FAST FACTS

- Company founded in 2002
- Headquarters in Atlanta, GA
- 50,000 teams and 100,000 users in over 170 countries
- Customers include 35 of the Fortune 100
- 100% of employees are Certified ScrumMasters

MAJOR INDUSTRIES

- Software
- Technology
- Financial Services
- Government/Defense
- Healthcare
- Telecom
- Education/Non-Profit
- Manufacturing

MAJOR CUSTOMERS

- Boeing
- Cerner
- Fiserv
- GMAC
- Lilly
- Lockheed Martin
- Merck
- Motorola
- Novell
- Oracle
- Qualcomm
- SAP
- Siemens

See more VersionOne customers at:
www.VersionOne.com/Customers