

# An Overview of Lean-Agile Methods

by Al Shalloway

AGILE agile  
LEAN scrum  
kanban

## **An Overview of Agile Lean-Methods**

### **A Net Objectives Essential White Paper**

Net Objectives Press, a division of Net Objectives

1037 NE 65th Street Suite #362

Seattle, WA 98115-6655

404-593-8375

Find us on the Web at: *www.netobjectives.com*

To report errors, please send a note to *info@netobjectives.com*

Copyright © 2011 Net Objectives, Inc. All Rights Reserved.

Cover design by Andrea Bain

Published by Net Objectives, Inc.

Net Objectives and the Net Objectives logo are registered trademark of Net Objectives, Inc.

### **Notice of Rights**

No part of this publication may be reproduced, or stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the written consent of Net Objectives, Inc.

### **Notice of Liabilities**

The information in this book is distributed on an “As Is” basis without warranty. While every precaution has been taken in the preparation of this book, neither the authors nor Net Objectives shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer or hardware products described in it.

10 9 8 7 6 5 4 3 2

Printed in the United States of America

This article is reprinted with permission from **The Agile Journal**, 07 Sep 2010. See [www.agile-journal.com/articles/columns/column-articles/3222-an-overview-of-lean-agile-methods](http://www.agile-journal.com/articles/columns/column-articles/3222-an-overview-of-lean-agile-methods).

## AN OVERVIEW OF AGILE METHODS

Life used to be simpler. In the 90s, if you wanted to go Agile, XP was the route of choice. And then Scrum became popular. And it was not too long before organizations began to hit the limits of these approaches due to their focus on teams. And then it became apparent that lean principles could be applied to software and Lean Software Development and later Kanban were added to the mix. Now, if you want to go Agile, you have a great many choices: Not just about which method to use, but where to start, whether to go top-down or bottom-up, and what should be the scope of your effort.

It can be daunting. Let's see if we can make it less so.

## WHAT IS AGILE?

Let's start by considering what is meant by "Agile." Many people would tell you that Agile means building software in small steps (iterations), measuring the progress and health of the project in terms of developed software, and working in close contact with the customer to ensure that what is built is truly useful. This is what I would call "team agility" because its focus is on the team and how the team works. It is the most common form of agility in use at present. While it is very effective for many, it should not really be the goal you strive for.

The best measure of success – the real goal of our work – must focus on how responsive the business unit that the development organization serves can be. I call this "business agility." It encompasses more than merely the team. Unfortunately, it is much harder to achieve. Business agility is the ability for a business (or business unit) to deliver increments of business value to their customers (for product companies) or the business side of the organization (for IT organizations). Business agility enables companies to respond quickly as needed when market conditions change, new technologies arise, or new ideas are developed.

Business agility is like having a car that is nimble, efficient, and goes where it needs to. Team agility is like having a finely tuned engine that enables the car to do whatever the driver wants. In other words, business agility is the real goal; team agility is a means to that end. Now it may be that in small enough organizations, focusing on the team alone may be adequate; however, in most organizations of size, trying to achieve business agility by focusing on the team is like trying to get a better performing car just by tuning and re-tuning the engine: it can help but you might see better results by improving the transmission or brakes.

## POPULAR AGILE METHODS: SCRUM AND XP

As I said, Scrum and now, to a lesser extent, eXtreme Programming, have been the most popular methods in the Agile movement. XP is rooted in development practices such as pair-programming, test-first, continuous integration, collective ownership and more. Scrum is primarily a process framework that exposes an organization's impediments. Its underlying assumption is that if you can see what you are doing wrong, you can fix the problem. Scrum's power is that you will discover problems quickly – in a matter of weeks. This is a vast improvement over waterfall methods. The



**Alan Shalloway** is the founder and CEO of Net Objectives. With over 40 years' experience, Mr. Shalloway is an industry thought leader in Lean, Kanban, Scrum, and design patterns. He helps companies transition to lean and Agile

methods enterprise-wide and teaches courses in these areas. Mr. Shalloway has developed training and coaching methods for Lean-Agile that have helped his clients achieve long-term, sustainable productivity gains. He is the primary author of *Design Patterns Explained: A New Perspective on Object-Oriented Design*, *Lean-Agile Pocket Guide for Scrum Teams*, and *Lean-Agile Software Development: Achieving Enterprise Agility* and is currently writing *Essential Skills for the Agile Developer*. Mr. Shalloway is a popular speaker at prestigious conferences worldwide and a cofounder and board member of the Lean Software and Systems Consortium. Mr. Shalloway has a master's degree in computer science from MIT and a master's degree in mathematics from Emory University. He can be reached at [alshall@netobjectives.com](mailto:alshall@netobjectives.com).

challenge is that sometimes the problems that are exposed are such that the solutions are not clear.

Both methods use iterations (building in 1-4 week intervals) and self-organizing, cross-functional teams.

Scrum has gotten to be more popular for a few reasons. First, it is easier to start because the team doesn't have to commit to the technical practices that are the core of XP. This level of change is too much for some teams to bear. Second, Scrum's certification program has spawned scores of "Certified Scrum Trainers" (CSTs) who both teach Scrum and promote it. As a result, many in the industry have come to think that "going Agile" means "taking a CSM (Certified ScrumMaster) class and adopting Scrum."

## CHALLENGES IN THE AGILE WORLD

Scrum has grown virally. Now, the challenges are showing up. In fact, one of the originators of Scrum, Ken Schwaber, said in an interview with Agile Collab that

*"75% of organizations using Scrum will not succeed in getting the benefits that they hope for from it." [1]*

There are many opinions about why this is the case. This article's author posited several reasons including,

- Many problems facing development teams actually originate outside the team; for example, overloading teams by giving them too many projects at one time.
- Many large organizations have poor deployment methods that make it difficult to get software out the door. While Scrum may expose these impediments, they are difficult to solve and Scrum, in itself, does not give any insights on how to solve them.
- Many organizations have difficulty creating the cross-functional teams that Scrum requires to work. People with specialized skills, knowledge of legacy code or domain expertise are often required to be shared across teams, breaking the power of the Scrum model.
- Many teams that are new to Scrum have never been taught the principles of product

development flow.

- While Scrum was originally designed for co-located teams, it is often necessary to use distributed teams.

Many Scrum aficionados claim that the real reason that organizations are not experiencing success with Scrum is that those organizations just don't have the discipline or motivation necessary to solve their problems.

Regardless of the cause, the problem remains. To meet this challenge, two other Agile-related methods have sprung up: Lean Software Development and "A Virtual Kanban System for Software Engineering" (or 'Kanban' for short). Both of these methods directly address the problems that organizations are having with Scrum. It is why I started doing Lean over five years ago and Kanban about 3 years ago.

## LEAN SOFTWARE DEVELOPMENT

Lean Software Development is founded on the understanding that the principles of Lean manufacturing and product development methods can apply to software development. Of course, the practices are different but the mindset, the way of thinking, is the same.

Its origins date back to the mid-90s with an article by Bob Charette. It became popularized with by Mary and Tom Poppendieck's books, the first one, Lean Software Development, An Agile Toolkit, being published in 2003 (Poppendieck & Poppendieck, 2003). Lean Software Development is based on the notions of eliminating waste (any effort that is not of value to the customer), optimizing the whole (going from "concept to cash"), delivering fast, building quality in, respecting people, improving relentlessly and deferring commitment until you know what you need to do.

A second camp, so to speak, has sprung up around the work of Don Reinertsen[2]. Reinertsen approaches Lean on the basis of appreciating the system that people are working in, fostering the skills of the people you have in that systems, and attending to time by seeing how work flows through the system. By "flow," he means to see where delays occur or where large queues form.

Regardless of which way you prefer to approach

Lean, it is useful to provide insights into how to solve problems both inside and outside of the development organization. Its model of flow (focusing on fewer, smaller things in your development pipeline to get value delivered quickly) suggests that one way to improve your software development value stream is to avoid overloading your development teams. When you consider how many development teams seem to thrash because they are doing too many things at once, you may intuitively get this concept.

## KANBAN

Kanban is based both on Lean and the theory of constraints. It was created by David Anderson about five years ago and is achieving an upsurge in interest[3]. Kanban is based on a few basic premises. The first is that requiring an organization to undergo a large change at one time in order to start being Agile may be counter-productive. Telling people they have to adopt new roles and work in a different way often creates fear in those being forced to change. They feel devalued as they are told their former methods of working are to be abandoned.

The first few years of the Agile movement saw Scrum adopted where teams already existed and the developers/testers/analysts wanted to change. Many going the Agile route no longer fit into this class – perhaps another reason for the difficulties encountered.

Many Agilists will tell you that the way to overcome this high amount of change is to hire a trainer and coach. However, an alternative approach may be to undertake a smooth transition while starting down a road that gets you to your ultimate goal in a less disruptive manner.

The essence of Kanban is to start where you are and do three things:

- Make agreements with your stakeholders as to what your goals are and how many projects the developers will be working on at any one time.
- Create visibility into the development process by mapping its value stream. A “value stream” is the series of steps where an idea goes from concept to being defined to being built to being deployed to being used.

Then have explicitly defined policies of how work moves through the value stream. This is extremely important because it enables management to see how their actions affect the team and because the discussions around these explicit policies greatly accelerate learning.

- Manage the amount of work in progress (WIP) being done at each step. In particular, the team (and management) needs to look for places where the work gets backed up and then figure out how to unblock things. This clarity provides insights into solving the impediments the team hits.

Corey Ladas incorporated Kanban to the Scrum framework to create a hybrid called “Scrumban.” This has been a useful method for helping Scrum teams to adopt Kanban methods[4].

## ACHIEVING BUSINESS AGILITY: WHERE TO START

So, where do you start? It depends on where you are and where you want to go. My experience is that trying to start an Agile adoption through a team pilot project is not always best. Like that car’s engine, the team method may not be the problem. Even if it needs improving, you will always be better off by looking at the entire context within which software development is done and then choosing what to improve. Here are two recommendations:

- Begin by looking at the entire value stream. Where are the challenges, what needs to improve? This does not mean you are committing to changing more than the team, but doing so will provide insights that will ensure team progress translates to business progress, to business agility.
- Let the teams involved choose between Scrum and Kanban. They will intuitively know which is best for them. Part of respecting people means to allow them to choose how they do their work – within the context of the business, of course.

Considering the entire context is the difference between scaling agility and agility at scale. But that is a topic for another article.

If you are interested in learning more about

Kanban, Lean, Scrum, or XP, please visit the Net Objectives resources page at <http://www.netobjectives.com/resources>.

## WORKS CITED

Anderson, D. (2010). Kanban. Sequim, WA: Blue Hole Press .

Ladas, C. (2009). Scrumban: Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press .

Poppendieck, M., & Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit. Addison-Wesley Professional.

Reinertsen, D. (1997). Managing the design factory : a product developer's toolkit. New York: Free Press.

## FOOTNOTES

[1] [www.agilecollab.com/interview-with-kenschwaber](http://www.agilecollab.com/interview-with-kenschwaber)

[2] Don's book, Managing the Design Factory (Reinertsen, 1997) is still state of the art, a must read.

[3] David's recent book, Kanban (Anderson, 2010) Kanban book is another must read.

[4] If you are doing Scrum and facing challenges completing iterations effectively, I highly recommend Scrumban (Ladas, 2009).

## BUSINESS-DRIVEN SOFTWARE DEVELOPMENT

**Business-Driven Software Development** is Net Objectives' proprietary integration of Lean-Thinking with Agile methods across the business, management and development teams to maximize the value delivered from a software development organization. This approach has a consistent track record of delivering higher quality products faster and with lower cost than other methods.

Business-Driven Software Development goes beyond the first generation of Agile methods such as Scrum and XP by viewing the entire value stream of development. Lean-Thinking enables product portfolio management, release planning and critical metrics to create a top-down vision while still promoting a bottom-up implementation.

Our approach integrates business, management and teams. Popular Agile methods, such as Scrum, tend to isolate teams from the business side and seem to have forgotten management's role altogether. These are critical aspects of all successful organizations. Here are some key elements:

- Business provides the vision and direction; properly selecting, sizing and prioritizing those products and enhancements that will maximize your investment.
- Teams self-organize and do the work; consistently delivering value quickly while reducing the risk of developing what is not needed.
- Management bridges the two; providing the right environment for successful development by creating an organizational structure that removes impediments to the production of value. This increases productivity, lowers cost and improves quality.

## BECOME A LEAN-AGILE ENTERPRISE

**Involve all levels.** All levels of your organization will experience impacts and require change management. We help prepare executive, mid-management and the front-line with the competencies required to successfully change the culture to a Lean-Agile enterprise.

**Prioritization is only half the problem.** Learn how to both prioritize and size your initiatives to enable your teams to implement them quickly.

**Learn to come from business need not just system capability.** There is a disconnect between the business side and development side in many organizations. Learn how BDSO can bridge this gap by providing the practices for managing the flow of work.

## WHY NET OBJECTIVES

While many organizations are having success with Agile methods, many more are not. Much of this is due to organizations either starting in the wrong place, such as focusing on the team when that is not the main problem, or using the wrong method, such as using Scrum or kanban because they are popular.

Net Objectives is experienced in all of the Agile team methods (Scrum, XP, Kanban) and integrates business, management and teams. This lets us help you select the right method for you.

# LEARN TO DRIVE DEVELOPMENT FROM THE DELIVERY OF BUSINESS VALUE

What really matters to any organization? The delivery of value to customers. Most development organizations, both large and small, are not organized to optimize the delivery of value. By focusing the system within which your people are working and by aligning your people by giving them clear visibility into the value they are creating, any development organization can deliver far more value, lower friction, and do it with fewer acts of self-destructive heroism on the part of the teams.

## THE NET OBJECTIVES TRANSFORMATION MODEL

Our approach is to start where you are and then set out a roadmap to get you to where you want to be, with concrete actionable steps to make immediate progress at a rate your people and organization can absorb. We do this by guiding executive leadership, middle management, and the teams at the working surface. The coordination of all three is required to make change that will stick.

## OUR EXPERTS

Net Objectives' consultants are actually a team. Some are well known thought leaders. Most of them are authors. All of them are contributors to our approach.



Al Shalloway



Alan Chedalawada



Guy Beaver



Scott Bain



Max Guernsey



Luniel de Beer

## SELECTED COURSES

### Executive Leadership and Management

Lean-Agile Executive Briefing  
Preparing Leadership for a Lean-Agile/SAFe Transformation

### Product Manager & Product Owner

Lean-Agile Product Roadmaps  
PM/PO Essentials

### Lean-Agile at the Team

Acceptance Test-Driven Development  
Implementing Team Agility  
Team Agility Coaching Certification  
Lean-Agile Story Writing with Tests

### Technical Agility

Advanced Software Design  
Design Patterns Lab  
Effective Object-Oriented Analysis and Design  
Emergent Design  
Sustainable Test-Driven Development

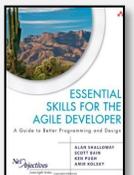
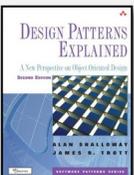
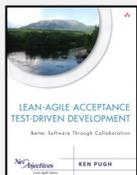
### DevOps

DevOps for Leaders and Managers  
DevOps Roadmap Overview

### SAFe®-Related

Implementing SAFe with SPC4 Certification  
Leading SAFe® 4.0  
Using ATDD/BDD in the Agile Release Train (workshop)  
Architecting in a SAFe Environment  
Implement the Built-in Quality of SAFe  
Taking Agile at Scale to the Next Level

## OUR BOOKS AND RESOURCES



## CONTACT US

info@netobjectives.com  
1.888.LEAN-244 (1.888.532.6244)

## LEARN MORE

www.NetObjectives.com  
portal.NetObjectives.com



Copyright © Net Objectives, Inc.