

Chapter 10. Becoming an Agile Enterprise

Continuous improvement is not about the things you do well - that's work. Continuous improvement is about removing the things that get in the way of your work. The headaches, the things that slow you down, that's what continuous improvement is all about. —Author Unknown

In This Chapter

This chapter discusses the challenge of becoming an Agile enterprise. This involves thinking clearly about where you are, where you want to go, and what is in the way. We consider three different organization types, based on what we have seen in helping companies transition to Agile. Fundamentally, the transition involves continual process improvement.

Takeaways

Key insights to take away from this chapter include

- There are many ways to achieve Agility throughout the enterprise.
- If you have well-established development teams, start by making them Agile; then improve the portfolio-management team, all the while improving both how your teams work together and their technical skills.
- If you don't have well-established teams, reduce the number of projects underway so you can create teams focused on completing prioritized requests that can be validated by the Product Champion.

Where Do You Want to Go?

Before you can decide on your path, you must know where you are and where you want to go. Where you want to go is probably clear: It is where most other companies also want to go. These include being able to do the following.

- Identify the needs of the market.
- Respond quickly to market changes.
- Create software that is tuned to the market: high quality and focused on providing the most valuable features.
- Create products (internal and external) that are long-lived, easily extended, and easily maintained.

To achieve this, Lean-Agile suggests enterprises do the following.

- Identify and define Minimum Marketable Features to build.
- Manage their development organization so that it maximizes throughput, minimizes cycle time, and produces high-quality software in the process.
- Ensure their teams follow Lean principles to the best of their capabilities within their constraints.
- Employ teams that are highly skilled in the engineering practices needed to sustainably build high-quality software, including test-driven development and design patterns.
- Adopt continuous process-improvement practices and become a learning organization.

All of this is based on the need for the organization to be business-driven. It is not enough for the teams to become Agile; the business must structure and lead teams to where they can add the most value for their customers. We call this “Business-Driven Software Development.”

What’s Gets in the Way?

Working with dozens of companies that have adopted Lean-Agile, we have seen a number of common impediments to making the transition. These include

- **Teams are not well formed.** A project is done by a collection of people assigned to work on it. Problems arise when these people are assigned to work on many other things as well.
- **Teams are not co-located.** Studies have shown that situating people more than 30 feet away from each other greatly reduces their ability to communicate and work together. As this is a reality today, we have to learn to deal with it effectively.
- **Annual planning cycles result in longer-than-necessary projects.** In addition, failing to focus on Minimum Marketable Features results in larger-than-necessary projects, which take longer to complete.
- **Large batches of unprioritized requirements are pushed through the organization.** There is no mechanism to limit work to resource capacity.
- **Program managers and business sponsors compete for resources rather than working together to maximize the return on them.** The preponderance of large projects means that different product lines compete with each other for budgets. This competition does not promote delivering the best value enterprise-wide.
- **Automated acceptance testing is not being done.** Test-Driven Development also is not being done. Testing is initiated too late in the development cycle.
- **Integration is done at project’s end.** Integration costs are high because teams work independently of each other for too long, resulting in waste.
- **Code quality is left up to programmers’ personal beliefs.** There is much known about what constitutes quality code. Unfortunately, most companies still allow developers to code based on their own preferences and have not set standards for what constitutes quality code.

- **Finding and removing the root causes of problems is not pursued aggressively.** Bugs are tolerated as a way of life in the software world. In fact, many organizations utilize bug tracking as status for release readiness.
- **Continuous process improvement is not practiced or valued.** Most companies are so busy trying to fix the latest crisis that there is no time to focus on process improvement to avoid causing the next one.

To become Agile, an organization must overcome these challenges.

Guidelines for the Transition

When companies transition to Agile, they must always consider the following three questions:

1. What are the pain points that can be most easily eliminated?
2. What cultural attitudes are in the way of the transition?
3. What metrics are in the way of the transition?

In addition, it also helps to remember these points:

- Trying to do too many things at the start can be counterproductive, even if all of them are useful.
- People undergoing transitions have a certain degree of fear.
- People must always understand what is in it for them.
- Adding significant work to the team about to make a transition may cause people to abandon the transition.
- We should look for the “trim tabs” that will help smooth the

Trim Tabs

We have always admired and been inspired by R. Buckminster Fuller, author of the groundbreaking book, Critical Path. Among other things, he coined the term “Spaceship Earth” and invented the geodesic dome.

He reflected on trim tabs, which are used in aviation and shipping. They are attached to large control surfaces (e.g., flaps and rudders) that would otherwise be difficult to move.

Bucky once wrote about standing on a dock reflecting on how difficult it was to change society when he saw a large ship go by. He considered the effort it would take to change its direction. Pushing on the bow (front) of the ship was one way but it was not very efficient. Using the rudder is not as easy as it might appear. When the large surface area of the rudder is pushed against the water, the effort required and the stress on the rudder itself is very great. Just as Bucky was thinking, “It’s very difficult indeed to change the course of a great ship,” the stern (tail) of the ship went by and Bucky noticed what’s called a trim tab on the rudder.

A trim tab is like a miniature rudder. When you move it, it creates a low-pressure area that allows the rudder to turn much more easily. Bucky realized that with the trim tab, it took relatively little effort to move a great, big ship. If one is to make a difference in the world, Bucky realized, one had to look for the trim tabs in life. That is, those things that require little effort to produce a large effect.

transition.

You don't have to find everything that will help you, but you do need to find the few, most critical things that will make the transition work in your organization. You must also focus on the proper things, on the things you can change. Some people call this "picking the low hanging fruit." We prefer "finding the trim tabs" because it implies more than little effort, it also implies high return for the little effort.

Where Do You Start?

So where do we start? That depends on where you are.

To help you think about it, we will consider three examples that are based on our experience helping companies become Agile. Most of the companies we have worked with fall into one of three general categories:

- Product companies making software that is sold to those outside of their company
- IT organizations that have reasonably well-defined teams creating software that is used directly by their own staff, their agents, or their customers in support of their company's services
- IT-product companies (a hybrid of the first two) making software that is used in an administrative role of other companies; for example, a company that makes hospital-administrative software.

Do not be concerned if your company does not exactly fit one of the following descriptions. Instead, look at the issues and structures present and how we recommend the transition take place. See which issues are most important for your company. In all of the examples that follow, the organizations are comprised of 300 to 4,000 people and each product/service line has 50 to 500 people dedicated to it.

The Product Company

Product companies build software that individuals or other companies use. These products might be in the gaming industry (e.g., a PC game), in the business arena (e.g., a word processor), in the defense arena (e.g., missile-guidance software) or really almost anything. The biggest differentiator of product companies is usually whether or not they are writing embedded software (i.e., software running on dedicated hardware such as a car's controller).

Product companies typically have well-defined teams but often have people on more than one team. Very often people are on one team, building a new product or enhancement, while also on another team that supports legacy products. The company's biggest challenges are often improving product quality and increasing team efficiency. Although they often don't have the yearly planning cycles that plague IT organizations, they often have long-range plans for their products that result in building excess features.

The most common place to start in product companies is with improving their teams' efficiencies, as shown in Figure 10.1. This is often achieved by implementing Scrum or Kanban (whichever is more appropriate for the type of work they are doing). By learning how to build software in stages, it puts pressure on the program managers to select better product-enhancement definitions. When implementing Scrum, it is advisable to move quality assurance up to the front of the process. This improves team organization and communication.

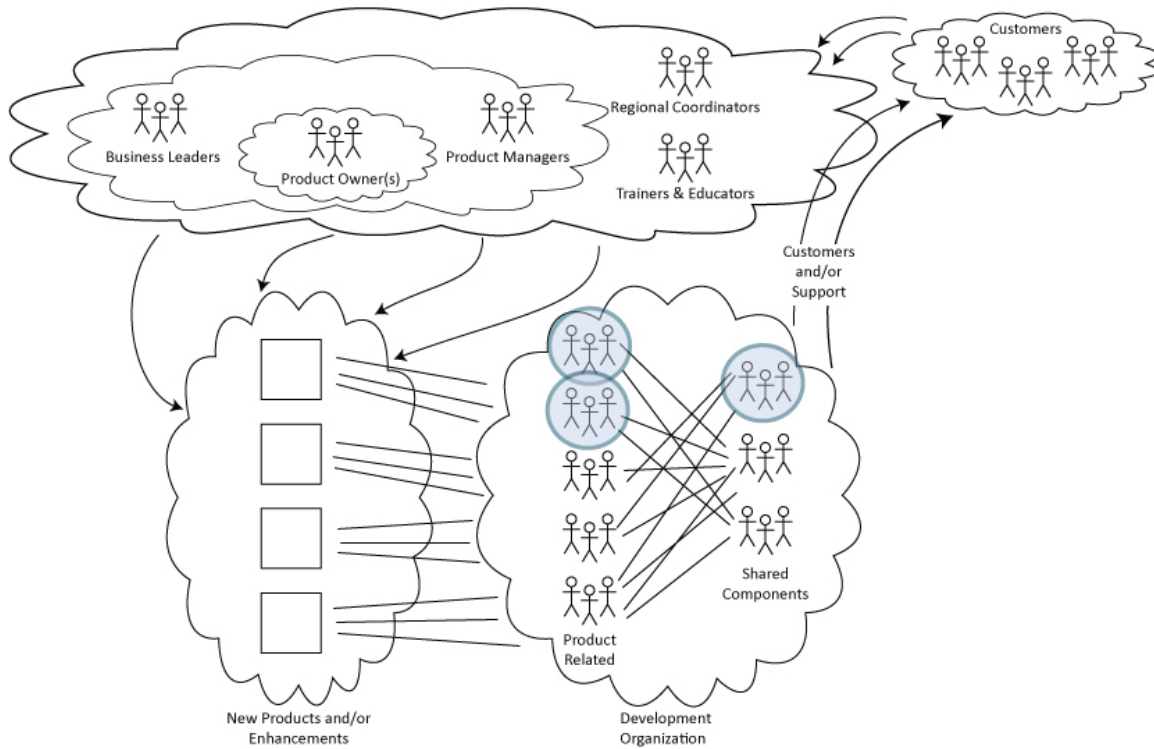


Figure 10.1. Create Agile teams

In a team's adoption of Scrum or Kanban, they will encounter many challenges that they can fix or improve directly. We have found frequently that a useful change is moving quality assurance up to the front of the development process (see chapter 9, *The Role of Quality Assurance in Lean-Agile Software Development*). After doing so, teams find that they can make significant improvements to their process and the quality of code they develop. However, after some period, they find that they are impeded by things outside their control. For example, the requested product enhancements may be over-defined and/or contain too many features (i.e., be bigger than necessary). This may require them to work on several product teams, causing thrashing.

While team training can be limited to the Scrum and Kanban methods required, once it becomes clear that the issues affecting the team are beyond the team's control, it is time for the next step. This is also a good time to train the teams in the wider principles of Lean—because at this point, one must look beyond the team.

The next step is coaching program managers on how to select smaller enhancements to build. That is, it's time to start building Minimum Marketable Features (MMF). This is shown in Figure 10.2.

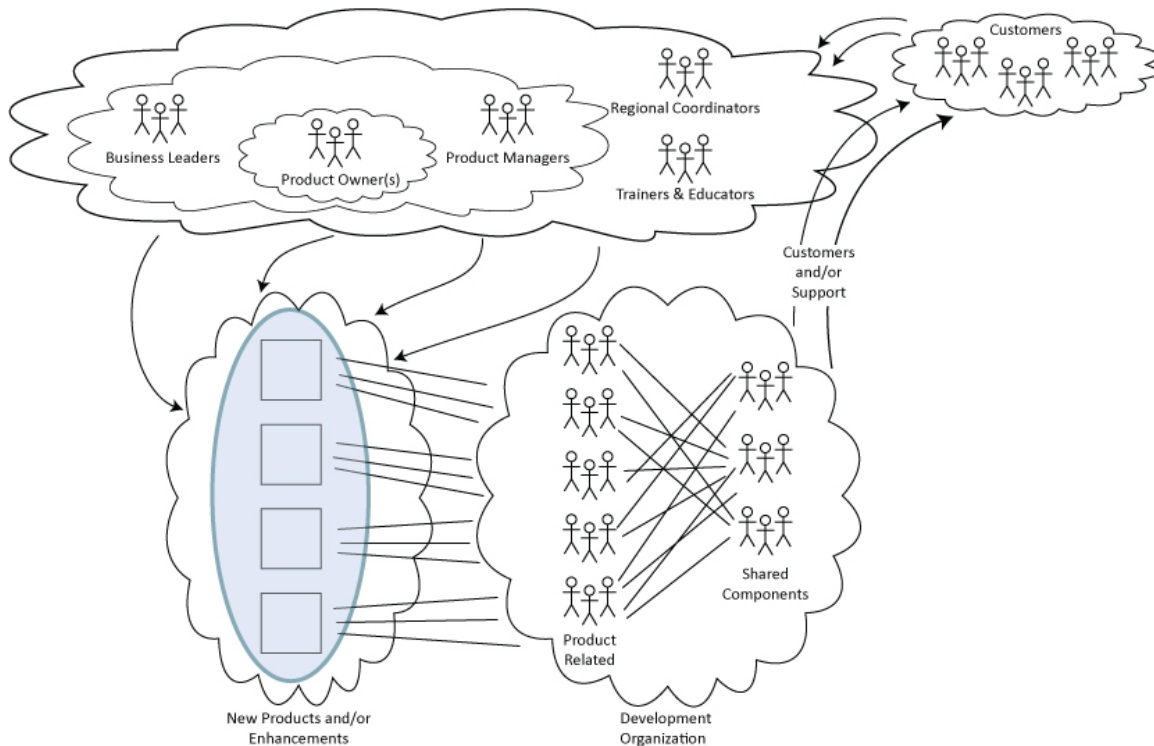


Figure 10.2. Improve the way product enhancements are selected

These two steps can actually be done at the same time but it is often difficult to get program managers to see the viability of it until after teams are able to shorten their development cycles. Once that is done, teams can make a case for more time with product managers or their proxies because they can demonstrate that doing without this contact impedes the team's ability to deliver high-quality code quickly.

Once an organization is building MMFs quickly, it should become possible for the product portfolio-management team to select MMFs based on maximizing ROI for the entire organization, not merely each product line. In other words, once program managers see smaller product enhancements implemented by the development teams, they will realize they are competing with each other for resources in a very direct way. As program managers clearly see the value of having fewer product enhancements in the delivery pipeline at any one time (to ensure maximizing development teams' efficiency), they understand the need to select product enhancements that align with the needs of the entire organization. This collaboration—rather than competition—among program managers is illustrated in Figure 10.3.

The following is an excerpt from *Lean-Agile Software Development: Achieving Enterprise Agility* by Shalloway, Beaver, and Trott. No portions may be reproduced without the express permission of Net Objectives, Inc.

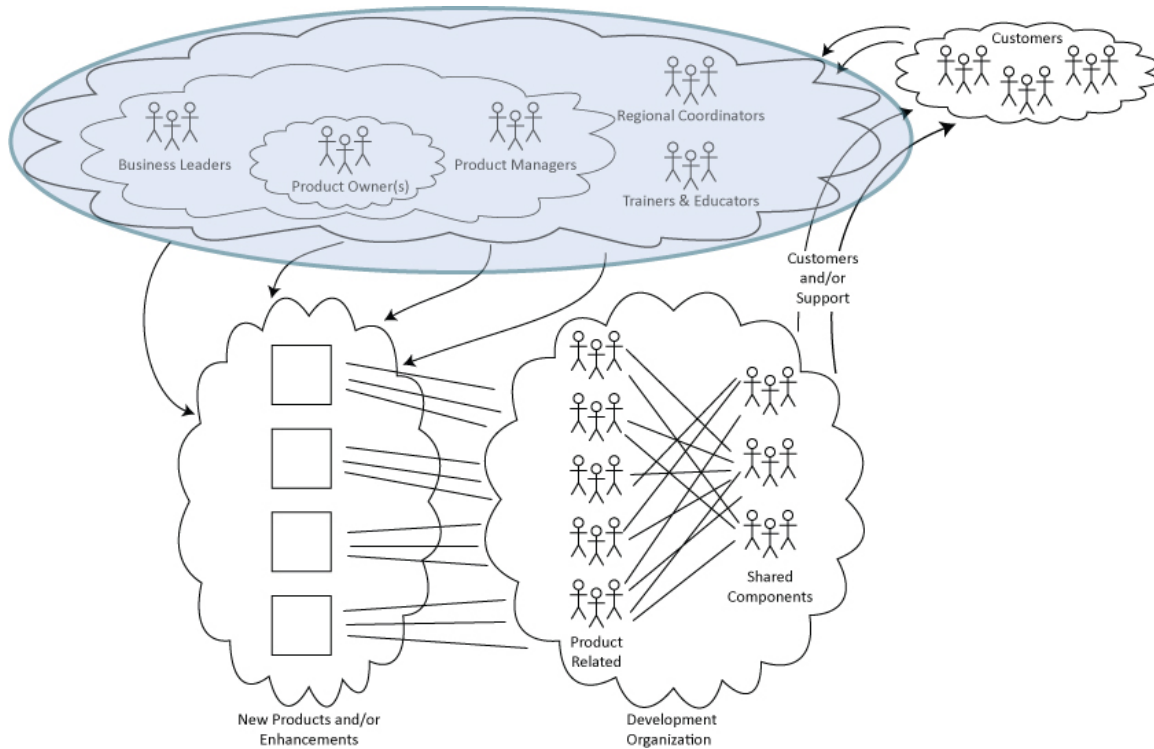


Figure 10.3. Program managers collaborating to pick MMFs that will “optimize the whole” of the development organization’s efforts

Throughout this process, the development side of the organization should be improving itself in two ways:

- Teams should be adopting engineering best practices, including test-driven development and design patterns tailored for their particular situation.
- Teams should be organized to work with other teams to lower integration costs and improve cycle-time.

This “last” step, to reorganize the way teams work with each other, is actually an ongoing step. It often starts after the others because it takes an understanding of flow to guide it. The process of improving how teams interact with each other is discussed further in chapter 12, The Product Coordination Team.”

When the Product Company Writes Embedded Software

When embedded software is being written, the transition becomes quite a bit more complex due to the added cost of integrating across both the software and hardware teams. Value-stream mapping early on can illuminate many issues. Lean’s focus on flow and minimizing cycle time provides insight on how to better integrate these teams. We discuss a method of coordinating different product teams in chapter 12, The Product Coordination Team.

The Product Company's Transition Path

Essentially, this is a path that often starts with the teams at the bottom and, guided by Lean, works its way to the top. The previously described path is not meant to be the specific way to do it as much as it is a guide on how one aspect of the transition leads to another.

The IT Company

IT organizations often have a greater degree of two significant challenges than what exists in product companies. First, teams are often organized in silos (by function); in addition, the IT organization itself—rather than the business side of the organization—decides which product enhancements to build. These are separate concerns. Typically, however, creating well-organized, cross-functional teams, which solves the first issue, often helps guide the second.

Ironically, starting Scrum on selected pilot projects may improve those projects while adversely affecting the rest of the organization. If the total number of projects does not change, and some teams are selected to work on only one at a time using Scrum, the result will be that the people not on these Scrum teams will end up with even more work than they had before. This causes even more project switching with its corresponding thrashing and lowers productivity even further. Looking at the results of the pilot projects alone often makes Agility look great, but the overall effect can be a decrease in productivity of the software development organization as a whole. If one mistakenly thinks that Scrum is helping (because of the success of the pilot projects), “scaling” Scrum may prove difficult and even counterproductive. In IT organizations, a combination of lowering the number of active projects and creating well-defined teams to work on them must be done simultaneously to avoid causing some teams to thrash (non-Scrum) while others succeed (Scrum).

IT organizations, however, tend to be more driven by the development (IT) side of the business than product organizations are. They also have a greater stake in building software that end users can actually consume. In product companies, releasing a product is often all that is needed to make it available to the business's customers. But in IT organizations, the software is tightly tied to the processes of the organization. Adding a capability that the service group is not prepared to implement adds no value to the company because the mere availability of the software is not sufficient for the service organization to use it.

The IT Product Company

The IT product company, although it looks like other IT companies, is actually a product company—it sells services to other companies. However, since it provides services that its customers' IT departments would otherwise provide, it is often organized like an IT organization. And as we have seen in many IT organizations, it may not have real teams at all. Instead, when a product needs to be enhanced, people who can take on the appropriate roles are selected to do it. The annual project budget lifecycle is very much alive in organizations of this type: Learning how to create smaller projects is key.

We have found that the best way for these organizations to transition to Agile is to educate the program managers right from the start. The first step is to identify the possible MMFs to build, then prioritize them according to ROI. Once that is done, teams must be created and assigned MMFs to build. This should be done for the entire product line—so that several teams will transition to Agile right from the start. Teams working on the same MMF can pull from a common product backlog.

By selecting smaller product enhancements, it is more likely that each team will have all of the roles necessary to develop the features they are assigned. Co-locating teams will increase productivity even more. While this is a major transition for many companies, the creation of cross-functional teams will create immediate productivity gains which will more than offset any confusion the transition itself may cause.

The Importance of Continual Process Improvement

Lean is not about transitioning to a specific place; rather, it is more about learning that transitioning should be an ongoing process. However, transitioning for the sake of transitioning is not the goal; improving productivity and ROI is. Teams need to look at how to improve their own process and the dependencies they have between their teams. Management needs to continually seek ways to reduce cycle time and increase quality. This requires both coaching their teams to improve and discovering how the organization's structure impedes teams so they can fix it.

Summary

How a company transitions to Agile Software Development depends upon the where it is and what challenges it has to overcome. A combination of improving the teams, the structure within which they exist, and their ability to drive from business need will help it get there.

Try This

These exercises are best done as a conversation with someone in your organization. After each exercise, ask each other if there are any actions either of you can take to improve your situation.

- How many projects do the people in your company work on at any one time?
- What part of your organization is driving the business—the business/marketing side or the development side?
- How does this affect what is selected for development?

NET OBJECTIVES LEAN-AGILE APPROACH

INTEGRATED AND COHESIVE

All of our trainers, consultants, and coaches follow a consistent Lean-Agile approach to sustainable product development. By providing services at all of these levels, we provide you teams and management with a consistent message.

PROCESS EXECUTION

Net Objectives helps you initiate Agile adoption across teams and management with process training and follow-on coaching to accelerate and ease the transition to Lean-Agile practices.

SKILLS & COMPETENCIES

Both technical and process skills and competencies are essential for effective Agile software development. Net Objectives provides your teams with the knowledge and understanding required to build the right functionality in the right way to provide the greatest value and build a sustainable development environment.

ENTERPRISE STRATEGIES

Enterprise Agility requires a perspective of software development that embraces Lean principles as well as Agile methodologies. Our experienced consultants can help you develop a realistic strategy to leverage the benefits of Agile development within your organization.



Contact Us:
sales@netobjectives.com
1-888-LEAN-244
(1-888-532-6244)

*We deliver unique solutions
that lead to tangible improvements in software development
for your business, organization and teams.*

SERVICES OVERVIEW

TRAINING FOR AGILE DEVELOPERS AND MANAGERS

Net Objectives provides essential Lean-Agile technical and process training to organizations, teams and individuals through in-house course delivery worldwide and public course offerings across the US.

CURRICULA — CUSTOM COURSES AND PROGRAMS

Our Lean-Agile Core Curriculum provides the foundation for Agile Teams to succeed.

Lean Software Development

- Implementing Scrum for Your Team
- Agile Enterprise Release Planning
- Sustainable Test-Driven Development
- Agile Estimation with User Stories
- Design Patterns

In addition, we offer the most comprehensive technical and process training for Agile professionals in the industry as well as our own Certifications for Scrum Master and Product Champion.

PROCESS AND TECHNICAL TEAM COACHING

Our coaches facilitate your teams with their experience and wisdom by providing guidance, direction and motivation to quickly put their newly acquired competencies to work. Coaching ensures immediate knowledge transfer while working on your problem domain.

LEAN-AGILE ASSESSMENTS

Understand what Agility means to your organization and how best to implement your initiative by utilizing our Assessment services that include value mapping, strategic planning and execution. Our consultants will define an actionable plan that best fits your needs.

LEAN-AGILE CONSULTING

Seasoned Lean-Agile consultants provide you with an outside view to see what structural and cultural changes need to be made in order to create an organization that fosters effective Agile development that best serves your business and deliver value to your customers.

FREE INFORMATION

CONTACT US FOR A FREE CONSULTATION

Receive a free no-obligation consultation to discuss your needs, requirements and objectives. Learn about our courses, curricula, coaching and consulting services. We will arrange a free consultation with instructors or consultants most qualified to answer all your questions.

Call toll free at 1-888-LEAN-244 (1-888-532-6244) or email sales@netobjectives.com

REGISTER PROFESSIONAL LEAN-AGILE RESOURCES

Visit our website and register for access to professional Lean-Agile resources for management and developers. Enjoy access to webinars, podcasts, blogs, whitepapers, articles and more to help you become more Agile. Register at <http://www.netobjectives.com/user/register>.