

Agile Transformation Strategy

Experience has shown that there cannot be a prescriptive, one-size-fits-all solution for all organizations turning to Agile to improve product development. Just like Agile and Scrum itself, the process of transitioning organizations to Agile is best achieved using empirical techniques. We have defined a *Path to Agility* which represents the culmination of our experiences in the field working with customers. The *Path to Agility* is coupled with *Services Phases*. Each *Service Phase* is crafted to help the organization move down the *Path to Agility* as effectively as possible. This document describes the *Path to Agility* and the subsequent *Service Phases*, which together represent CollabNet's organizational Agile transition strategy.

Our approach to Agile transition is based on our experience with both successful and failed attempts at organizational Agile adoption.

As noted, one-size does not fit all when it comes to complex organizational change. Scrum says that difficult and complex projects require empirical process controls rather than simple linear steps. While our strategy is well-defined, the tactical day-to-day implementation is left to the knowledge, experience, and political savvy of the CollabNet consultant implementing the strategy.

PATH TO AN AGILE ENTERPRISE



Figure 1: The Path to an Agile Enterprise. Based on the original work of Kane Mar.

The CollabNet approach to Agile Transformation follows a common path we have identified in the course of our work with many organizations transitioning from non-Agile to fully Agile enterprises. The Path to Agility was formalized by Kane Mar, one of CollabNet's senior Agile consultants. Over the course of several years working in the field with a large cross-section of industries and organizations, Mar witnessed patterns in the way organizations adopted Agile and, more importantly, identified critical points most companies traverse in moving toward Agility.

We begin, at the start of a transformation project, with a non-Agile enterprise. At this stage, the majority of management, technical staff, and administrative employees are either unaware or unconvinced of Agile methodologies. We often find many misconceptions that cause skeptical employees to cling to the status quo, despite project failures and staff dissatisfaction.

There are several ways in which Agile methods can be initially introduced. Most often, developers seeking to improve their immediate situation read existing literature and implement some Agile practices in their small groups.

There are several ways in which Agile methods can be initially introduced. Most often, developers seeking to improve their immediate situation read existing literature and implement some Agile practices in their small groups. As these developers become more successful, they attract the interest of colleagues and management and the ideas gradually spread. Once sufficient momentum has been established, the organization may want to proceed with experimentation on one or several small projects. This phase of Agile adoption is usually limited in both scope and budget. By choosing smaller projects initially, the enterprise can experiment with this new way of doing things in a way that feels safe, minimizing risk.

Experimental projects help staff and management identify the advantages and disadvantages of Agile and, more importantly, teach everyone lessons on how Agile will be best implemented in the specific organization. We generally find that with demonstrated advantages and a refined and customized approach, most enterprises are ready to move forward with further transitions of more and larger teams and more complex and expensive projects. Evidence from the experimental projects creates buyoff from all levels of staff, building enthusiasm and willingness to participate in the next Pilot Phase. The Pilot Phase is more organized, but still limited in impact. Pilot projects are seeded throughout the group or department. These pilot projects are more significant than the previous experimental projects in terms of budget, team members, and visibility to both senior management and external clients.

The Pilot Phase amplifies the successes and lessons of the experimental phase and begins to bring organizational impediments to light. Organizational impediments can be defined as circumstances, structures, policies, and ways of doing things, sometimes outside of the scope of the project, that limit project success. Impediments rise to the surface over time. Initially, they are issues that directly affect the development team: poor tools, inefficient builds, and source code management issues. As project teams become more experienced, quality issues come into focus, including test-driven development, continuous integration, refactoring, etc. Even further into Agile adoption, organizational issues start to emerge, such as project staffing, individual compensation, and promotion (see below). Naturally, the emergence of these problems can create clashes between Agile team members (who are pushing for change) and those outside the team (who feel that their position is threatened). But, ultimately, resolving these issues results in a lasting change, rather than the “band-aids” that may have previously been applied, resulting in increased effectiveness and better products.

If the enterprise is serious about Agile adoption and has gained support from senior management, some of the issues will be addressed. The first changes will be tactical; those changes that are seen to be immediately needed to support the Agile teams. This might include changes to use a simpler build process, the introduction of continuous integration or reducing the numbers of meetings that teams are expected to attend. These initial changes to the organization are typically limited in scale (confined to a particular group or department) and perceived risk.

With growing acceptance of Agile in the enterprise, more and more teams will be interested in adopting Agile on their projects, usually resulting in a sudden increase in the number of Agile projects. This sudden increase in Agile teams will result in different teams taking up different Agile practices, or adopting similar practices, but implementing them in different ways. For example, some teams will do two-week iterations, some will do three-week iterations, and some teams will do four-week iterations. Some teams will hold daily team meetings, while others will not. And different teams will estimate story points in different ways. The response to these variations will be greater formalization of the Agile methodology. This formalization will usually be undertaken by a project management office (PMO) who will want to understand what is meant by “Agile within the context of

With growing acceptance of Agile in the enterprise, more and more teams will be interested in adopting Agile on their projects, usually resulting in a sudden increase in the number of Agile projects.

In order to make a successful transition to an Agile enterprise, any risk (either real or perceived) associated with the introduction of Agile development needs to be resolved promptly by senior management.

the Enterprise.” The PMO will try to define Agile practices such as terminology, length of iterations, reporting, and metrics.

Following the formalization or standardization of the Agile process, the enterprise will be ready to attempt a large-scale rollout. This is a rollout of Agile methods across several departments or organizations. The types of projects attempted will be far more ambitious and may include Scrums-of-Scrums, significant visibility to senior management, and increased corporate risk (associated with the potential of large project failure).

These enterprise-wide projects will raise enterprise-wide issues (compensation, promotion, roles and responsibilities) in addition to the usual local problems (tools, build times, quality issues, etc.). These challenges will be the most difficult for the enterprise to address because they will directly challenge the culture of the organization and will require changes to employees' behavior and day-to-day practices. In addition, some individuals will perceive that their positions and authority within the organization are at risk. In order to make a successful transition to an Agile enterprise, any risk (either real or perceived) associated with the introduction of Agile development needs to be resolved promptly by senior management.

The enterprise that has successfully negotiated this path is an enterprise that is able to manage constant change within the organization; an enterprise that is closely in tune with its customers; an organization that can rapidly change according to changing business conditions at incremental cost (rather than exponential cost) and that has employees who are constantly learning and innovating.

COLLABNET AGILE TRANSFORMATION STRATEGY

As Agile methodologies gain wider acceptance, they will be taken up by larger numbers of organizations. The question of how to introduce an Agile methodology into an enterprise with the least amount of risk will become more and more common. The two approaches that are typically talked about are top-down, where senior management takes the initiative to introduce Agile, and bottom-up, where developers and testers take the initiative to introduce Agile.

Both of these approaches have flaws. The successful approaches we have witnessed use some combination of both the top-down and bottom-up introduction. Agile software development practices can force large changes in the corporate culture, and making the change to an Agile organization is only possible if there is support from all parties involved.

Ultimately, there needs to be some coordination of both these efforts. There needs to be some planned approach to deal with concerns raised by those who do the work, as well as those who lead the organization and make the strategy a reality.

Strategy Overview

This plan of action is broken down into three different phases — each phase in turn relies on a combination of both top-down and bottom-up introductions. The three phases are separated based upon their scope of influence as it ever increases outward, like ripples on a pond. In the Pilot Phase, only a limited number of individuals are directly affected and the projects are limited in scope and risk. In the Formalization Phase an entire department (many people and multiple projects) may be affected, but the rollout at this point is still confined within the department. The final phase is the Enterprise Rollout Phase where individuals across multiple departments are affected and projects involve significant budgets and risk to the organization.

Each phase is described below with an outlined list of activities that are commonly performed. For each of the activities, there are a number of associated questions that need to be addressed. The reason that the questions appear without the answers are listed is simply because every organization is different, with different problems, personalities and requirements. Some organizations will have to answer many of these questions in order to be successful, while others will only need a few.

Phase 1. The Pilot

The Pilot Phase is mostly concerned with the immediate rollout of Agile practices to a known team. Concern is often focused on how best to adopt Agile practices and answering practical Agile questions such as how granular should product backlog items be, how do we write stories, how do

we integrate QA, etc.

The Pilot Phase typically lasts between six months and a year and will involve only a small number of experimental projects (between five and 10). These projects will have similar profiles. They will be limited in budget (which implies limits in duration and staffing numbers), scope, and risk. These projects will typically be focused on delivering functionality to an internal client [limiting external exposure], and will have few (if any) dependencies.

Activities undertaken (and questions that need to be addressed) during this phase should include:

- *Introduce Agile software methodologies to several small teams using coaches.* Should the coach be CSM certified? Is it possible to certify an existing project manager and then have him/her act as the ScrumMaster? How many teams can a ScrumMaster manage at a time? How large should those teams be?
- *Introduce Agile practices and terminology to the teams.* What terminology should the team use — Scrum, XP or some combination of both?
- *Identify likely cultural issues and organizational impediments.* Is there some group or individual who feels most threatened by the introduction of Agile methods? Is there a Methodology or Software Development Process group that needs education?



Figure 2: Agile Transformation Overview

- *Identify tool issues.* Are the tools quick, efficient, and reliable? Do they leave the code base in a known state? Or are the tools cumbersome and require extensive baby-sitting? Do the tools meet the needs of the team or is there an alternative solution which better meets their needs? Is the choice of tools made by the developers, or by some third party that isn't responsible for delivering code?
- *Identify likely IP issues (open source tools, code).* Does the organization have a fear of GPLcode? Is there a tendency for the organization to re-develop tools that already exist in the marketplace?
- *Identify management issues.* How do functional managers (i.e., QA Managers, Software Analysis Managers, etc) fit into an Agile model? Who should be the Product Owner and what should the team do if the Product Owner doesn't want to engage with the team?
- *Identify and resolve reasons for initial failures.* What made some of the initial Agile projects successful? And how should the failed projects be addressed?
- *Physical location and layout.* How important is collocation to the success of an organization's Agile projects? Can teams retain their offices and communicate via IM, email, video conferencing, or other tools? What about teams in different locations or time zones?
- *Present Agile to interest parties and Senior Management.* Who should know about the benefits that Agile software development can bring? How should they be educated — in a series of lectures or by presentations from the team?
- *Have senior management promote Agile internally.*

In nearly all cases, Agile represents deep changes at nearly every level of the organization, including sensitive areas such as roles and career paths, company culture, and team dynamics.

The Formalization Phase is focused on how best to rollout Agile methodologies to a much wider audience in a consistent manner. The natural consequence of this is that the Formalization Phase is characterized by substantial codification of the organizational understanding of Agile.

Why We Avoid Traditional Assessments

In nearly all cases, Agile represents deep changes at nearly every level of the organization, including sensitive areas such as roles and career paths, company culture, and team dynamics. The prospect of traditional “consultants” coming in and “assessing” an organization in preparation for the next big initiative is enough to make most employees disengage and withdraw from the process. Traditional assessment approaches like interviewing leave a lot to be desired when trying to unearth the organization's real impediments to change. Interviewees are rarely candid and most often act in their own self-interest or to protect valued initiatives.

Our approach to assessment leverages Scrum's built-in capacity to unearth root problems quickly and accurately. Our approach is simply to start doing Scrum with a pilot project. While pilot project selection is important, the exercise of asking a team to deliver working software in 30 days or less nearly always reveals many of the key team-level and organizational issues likely to block healthy, organization-wide adoption of Agile.

This approach is still valid if your organization has already started using Scrum or Agile. It is our experience that organizations moving to Agile without any professional coaching sometimes relax the Agile rules that cause the most pain to the team or organization. By doing this, the impediments and problems holding it back from realizing its full potential with Agile are masked (not made visible). When CollabNet identifies a pilot project in an organization and enforces all rules strictly, the effect is largely the same: the issues holding the organization back bubble to the surface.

Phase 2. Formalization of Agile Practices

The Formalization Phase is focused on how best to rollout Agile methodologies to a much wider audience in a consistent manner. The natural consequence of this is that the Formalization Phase is characterized by substantial codification of the organizational understanding of Agile. This codification will try to address issues such as how long an iteration should be, what tools Agile teams should use, and what formats teams should use for reporting progress. Naturally, these decisions will be influenced by the successes and failures of the Pilot Phase.

Projects in this phase are usually much larger (than those in the Pilot Phase) in terms of budget, scope, and risk. A larger budget means that these projects will also have a larger number of staff and longer durations. The types of projects addressed in the Formalization Phase will likely represent a cross-section from within a single department. They will likely address many different aspects of the organization's business such as projects to address new functionality, software maintenance, database maintenance, and reporting.

The Enterprise Rollout Phase is characterized by focus on communication between projects in different departments; the management, organization and running of very large projects that comprises of multiple scrum teams; and issues related to compensation, responsibilities, and promotion.

Activities undertaken (and questions that need to be addressed) during this phase should include:

- *Codify the organization's understanding of Agile.* This includes establishing:
 - Usage of common terminology. What does a Sprint mean? Or an Iteration? What is the Scrum equivalent of Iteration 0?
 - Usage of common metrics. What scale should the teams use for estimating story points, a scale from one to 10, a scale from one to five, or something else? What is the meaning of "velocity" and should the organization compare velocities between two different teams?
 - Usage of common tools. What source control tool should the teams use? Should all the teams be using some form of Continuous Integration, and if so, which tool? What about IDEs and code coverage tools?
 - Usage of common reporting formats. Should the teams present their results as Burn Down charts? Is there any value to Gantt charts, or other traditional project management tools?
- *Formally establish a coaching model.* What is the organizational coaching model? How should coaches be brought on board and what career path should they follow?
- *Establish an office layout/collocation policy.* Are teams co-located? Is there sufficient space to create team rooms? Is there an expense involved with collocating teams?
- *Establish Agile forums within the organization.* What are the best ways to ensure different teams are communicating their experiences? Should this be an informal event or should there be some ceremonial process?
- *Establish Agile project selection and project scoping (size and cost).* What projects are suitable for Agile software development? What criteria should be applied to the project selection process? How should you use Agile methods to estimate the size and cost of these projects?
- Present Agile methodologies to interest parties and senior management. Who within the organization would help promote Agile methods? Who should be educated about the benefits that Agile methods can bring to the enterprise?

Phase 3. The Enterprise Rollout

The Enterprise Rollout Phase is characterized by focus on communication between projects in different departments; the management, organization and running of very large projects that

comprises of multiple Scrum teams; and issues related to compensation, responsibilities, and promotion. These challenges will be the most difficult to resolve and will require considerable persuasive and political skills.

The projects that are typical in this phase of Agile adoption are large, expensive, and potentially risky.

They may be projects that have multiple Scrum projects and managed with a Scrum-of-Scrum, or may involve some element of distributed software development. The Enterprise Rollout Phase is usually much longer than either of the previous two phases. It is possible for projects in each of the previous phases to be initiated and completed within a six- to nine-month period (depending upon the number of people impacted, the types of issues raised, etc.). Projects in the Enterprise Rollout Phase, however, may last anywhere from one to two years.

Activities undertaken (and questions to be answered) during this phase should include:

- *Encourage internal communication regarding Agile.* What is the best approach for encouraging internal communication between different Agile teams? An Agile Forum? Or perhaps an email list is sufficient.
- *Anticipate change and have a plan to evaluate changing circumstances.* A new application is getting more traffic than anticipated. How can you exploit that to your best advantage?
- *Review and align compensation model with Agile teams.* Is everyone on the team adding value? How should project managers who don't facilitate a team (Scrum or Scrums-of-Scrums) be compensated? Is an architect who mentors a team more valuable than one who does not?
- *Review HR and hiring policies.* Are your existing hiring practices sufficient to find skilled staff that works well in an Agile environment? Does the team have any say in who joins the team? Or who should leave a team?
- *Establish parameters around very large projects with Agile.* What qualifies as a large project and at what point should a project be broken down into two or more sub-projects? Are there additional (financial) constraints that larger projects must meet? In a large project, who represents the Product Owner? Should there be a single Product Owner or is it okay to have multiple Product Owners?
- *Establish parameters around distributed projects with Agile.* How experienced should the team be? How is communication between teams handled? Should the Product Owner be located within the business or with the development team? Should the entire business be relocated to somewhere more cost-effective?
- *Establish promotion policies.* How should successful individuals be promoted? Should the promotion model be based on merit, influence, or some combination of both?
- *Establishing training model for coaches/Agile teams.* What are the training requirements of Agile teams? After doing some initial training, what else should Agile teams know?
- *Align funding with lines of business.* Funding for Agile teams is usually secured by the Product Owner. What does this mean for software development groups that have previously had their own source of funds? How will the management structure react to changes in the funding model for these departments?

COLLABNET SCRUM SERVICES

Our services are designed to move our customers down the Path to Agility by coinciding with the three phases of our Agile transition strategy. In each phase, we maximize the effectiveness of our involvement by combining elements of both top-down and bottom-up adoption through formal training, management mentoring, team coaching, and assistance with long-term strategy development and implementation. For a more thorough description of our services, please visit our Web site: <http://www.collab.net>.

Phase 1 Services:

- Certified ScrumMaster Training
- Product Owner Training
- Coaching through Initial Meetings
- Transition Strategy Coaching

A formal course introducing and solidifying Scrum's key principles is an excellent way to bring the majority of your staff up to the same level of understanding. Coupled with mentoring and coaching, this approach provides top-down support for the Agile initiative during this early phase of implementation. In general, pilot projects tend to cause bottom-up adoption of Agile based on the empirical results of the pilot teams. This success breeds interest from the rest of the organization and brings credibility to the methodology.

Phase 2 Services:

- Scrum and Agile Team Training
- Product Owner and Scaling Strategy Coaching
- Team Coaching to Resolve Organizational Impediments
- Agile Engineering Coaching

Formal courses on topics like Agile engineering practices and Agile product management demonstrate management's continued support for the advancement of the Agile initiative. As coaches and mentors transition to longer term fixtures within teams, their presence spurs bottom support for the initiative because they improve team dynamics and build a sense of collaboration between teams. In our experience, these courses are fun and instill excitement in the teams.

Phase 3 Services:

- Scrum and Agile Check-In Coaching
- Agile Engineering Coaching
- Long-Term Strategy and Scaling Coaching
- Custom Classes

ABOUT THE AUTHORS



This paper was collaboratively written by Victor Szalvay, Michael James, and Kane Mar. Szalvay serves as the Chief Technology Officer of CollabNet's Scrum business unit. Mar is an Agile coach and Certified Scrum Trainer, who has more than 15 years of experience as a software developer and project manager. James is a software process mentor and

Certified Scrum Trainer, who has worked in the software industry more than 20 years as a developer.

ABOUT COLLABNET

CollabNet leads the industry in Agile application lifecycle management (Agile ALM) in the Cloud. The CollabNet TeamForge ALM platform, CollabNet Subversion [software configuration management \(SCM\)](#) solution, and ScrumWorks project management software enable teams using any environment, methodology, and technology to improve productivity up to 50% and to reduce the cost of software development by up to 80%. Millions of users at more than 2,500 organizations, including Applied Biosystems, Capgemini, Deutsche Bank, Oracle, Reuters, and the U.S. Department of Defense, have transformed the way they develop software with CollabNet. For more information, visit www.collab.net.