A Roadmap to Agile Development:  
A Strategy to Increase Adoption Success

**Executive Summary**

Agile development practices are becoming increasingly popular as organizations try to keep up with market demands. However, adoption is not quick. According to a Forrester survey in December 2008, of the organizations who have adopted Agile, only 35% reported mature implementation; 50% have just started or are midway through implementation.\(^1\)

While the philosophy behind Agile is appealing, it is neither a quick nor a straightforward change. Organizations that try to adopt Agile too quickly are often discouraged with less than stellar results, and they are left wondering why Agile was not successful.

As Ken Schwaber, the co-developer of the Scrum process noted, “Many CIO’s still think of Agile as more, faster. However, as organizations and projects flee the existing controls and safeguards of waterfall and predictive processes, they need to recognize the even higher degree of control, risk management, and transparency required to use Scrum successfully. I estimate that 75% of those organizations using Scrum will not succeed in getting the benefits that they hope for from it.”\(^2\)

So what is going wrong? According to the 2008 “State of Agile” survey by Version One, the leading causes of failure for Agile projects were a company philosophy or culture at odds with core Agile values (23%) and lack of experience with Agile methods (21%). To address these challenges, it is a good idea to gradually introduce Agile into your organization, starting with a few internal projects and then expanding to more departments and more risky projects (i.e. customer-facing projects).\(^3\)

It’s helpful to have a roadmap that guides you along the path to Agile adoption. While no two organizations follow exactly the same path or have the same timing, there are common trends you can extract from organizations that have successfully made the transition. This paper provides such a strategy and answers such questions as:

- What phases do organizations go through?
- What types of projects are best suited for each phase?
- How long will the stage last?
- What types of questions should you ask to make the transition as seamless as possible?

Overview: Path to an Agile Enterprise

While all organizations are unique, they follow similar stages as they progress from a non-Agile environment to an Agile one (Figure 1).

Figure 1: Roadmap of phases of Agile development

There are five phases (including Non-Agile enterprise), but this paper focuses on the latter three as that is where the vast majority of the issues occur: The Pilot Phase, The Launch Phase and the Enterprise-wide Rollout. The phases differ based upon their scope of influence: the more entrenched Agile development becomes in an organization, the more people, time, budget and risk are involved. (Table 1).

<table>
<thead>
<tr>
<th>Low Risk/ Low Budget</th>
<th>High Risk/Substantial Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot Phase</td>
<td>Launch Phase</td>
</tr>
<tr>
<td>Reach</td>
<td>Cross-section of projects within a single department</td>
</tr>
<tr>
<td>Duration</td>
<td>6 months to a year</td>
</tr>
<tr>
<td>Project Profile</td>
<td>- Cross-section of projects in a single department</td>
</tr>
<tr>
<td></td>
<td>- Addresses new functionality, software maintenance, database maintenance and reporting</td>
</tr>
</tbody>
</table>

Table 1: Overview of three latter stages of Agile development


Non-Agile Enterprise

The Non-Agile enterprise is an organization to which Agile practices have not been introduced, and the prevailing attitude is that Agile methods do not have any benefits over existing methodologies and practices. This attitude is often arrived at with little or no understanding of what Agile methods are or what benefits they can provide. Many organizations will never feel the need to move away from this state.

In order for an organization to consider change, there needs to be some event. This event can occur in a number of ways, such as a high-profile project failure. Other events include increased competitive pressure from rival companies, or loss of market share to competitors. The most successful transitions to Agile software development tend to be done by companies that have a significant incentive to change how they operate, or else face permanent loss of income and market share.

Regardless of the nature of the event, organizations typically start considering Agile approaches when someone within the organization (i.e. the Agile champion) asks “How can we better deliver business value?”

Experimentation

Once an organization understands the value of Agile development, they slowly begin to introduce Agile methods. Typically, developers read existing literature and implement some of the practices into their own work. This creates interest from other developers and management, and the ideas gradually spread. At this point, the organization typically experiments with one of several small projects, but they are usually limited in both scope and budget.

In the experimentation stage, organizations start by experimenting with one of several small projects, but they are usually limited in both scope and budget.
Pilot Phase

Once an organization completes a few experimental projects and sees positive outcomes, they accept that there are advantages to Agile methodologies and begin a more systematic exploration and rollout.

The pilot phase is primarily concerned with the immediate rollout of Agile practices to a known team. Teams focus on how best to adopt Agile practices and answer practical Agile questions such as the following:

- How granular should backlog items be?
- How do we write stories?
- How do we integrate QA?
- How do we plan Agile projects?
- When will we finish?

This pilot phase is more organized than the experimental phase, but it has limited impact. These initial changes are typically confined to a particular group or department and are limited in budget (which implies limits in duration and staffing numbers), scope and risk.

This phase typically lasts between a few months and a year and involves only a small number (between 5 and 10) of experimental projects, which have similar profiles. These projects are typically focused on delivering functionality to an internal client, which limits external exposure, and they have few (if any) dependencies.

The first changes that take place during this phase are tactical and are needed to immediately support the Agile teams. For instance, this might include using a simpler build process, introducing Continuous Integration or reducing the numbers of meetings that teams are expected to attend. All of the activities are listed on the chart on the next page.

This stage is also characterized by a lot of lessons about estimating and planning. Management is highly concerned about estimating final dates, and there may be intense pressure to report progress to senior management on a daily or weekly basis.
Activities during the Pilot Phase

**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 a 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?

**Identify tool issues.**
- Are the tools quick, efficient, reliable, and 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 that better meets their needs?
- Is the choice of tools made by the developers or by some third party who isn’t responsible for delivering code?

**Identify likely IP issues (open source tools, GPL code).**
- Does the organization have a fear of open source or GPL code?
- Is there a tendency for the organization to re-develop tools that already exist in the marketplace?

**Identify management issues.**
- How do functional managers (eg. QA managers, software analysis managers, etc) fit into an Agile model?
- Who should be the Product Owner be, and what should the team do if the Product Owner doesn’t want to engage with them?

**Identify and resolve reasons for initial failures.**
- What made some of the initial Agile projects successful?
- How should the failed projects be addressed?

**Physical location and layout.**
- How important is collocation to Agile projects?
- Can teams retain their offices and communicate via IM and/or email?
- What about teams in different locations or time zones?

**Present Agile to interested 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?
Launch Phase

The launch phase is focused on how best to consistently roll out Agile methodologies to a much wider audience. With growing acceptance of Agile, more people are interested in adopting this approach, usually resulting in a sudden increase in the number of Agile projects and teams. The challenge with this phase is that teams commonly adopt different Agile practices or implement them in different ways.

For instance, here are some examples of how teams may handle Agile practices differently:

- **Iterations**: Teams may decide to do 2-, 3- or 4-week iterations
- **Meetings**: Some teams will have daily team meetings while others will not
- **User Stories**: Teams will estimate story points in different ways

As a result, the launch phase is characterized by substantial codification or formalization of the Agile methodology. This formalization is usually undertaken by a project management office (PMO) who wants to understand what is meant by “Agile within the context of the Enterprise”. See the sidebar, “Activities during the Launch Phase,” which provides specific examples.

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 these projects have a larger number of staff and longer durations. The types of projects addressed in the launch phase are typically a cross-section of projects that are undertaken within a single department. They likely address many different aspects of the organization’s business such as projects to address new functionality, software maintenance, database maintenance and reporting.
Activities during the Launch Phase:

**Codify the organization’s understanding of Agile.**

- Usage of common Terminology:
  - What does a Sprint or Iteration mean?
  - 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 1 to 10, a scale from 1 to 5, 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 IDE's and code coverage tools?
- Usage of common reporting formats:
  - Should the teams present their results as burndown charts?
  - Is there any value to Gantt charts, or Microsoft Project?

**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, and 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 would be 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 to interested 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?
Enterprise-wide Rollout

Following the formalization or standardization of the Agile process, the enterprise is ready to attempt a larger-scale rollout, which is a rollout across several departments or the organization. The types of projects attempted will be far more ambitious. There may be multiple projects managed with a Scrum-of-Scrum, and/or they may include some element of distributed software development, which significantly increases visibility to senior management and corporate risk (associated with potential large project failure).

The enterprise-wide 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 6 to 9 month period (depending upon the number of people impacted, the types of issues raised, etc), but projects in this last stage may last from 1 to 2 years.

In addition to the usual local problems that were present in the launch phase (tools, build times, quality issues etc), the enterprise-wide rollout phase presents issues that will be the most difficult to resolve such as compensation, promotion, roles and responsibilities. These issues directly challenge the culture of the organization and will require changes to peoples’ behavior and day-to-day practices; resolution will require considerable persuasive and political skills. In addition, some individuals will perceive that their position 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.
Activities during Enterprise-wide Rollout:

Encourage internal communication regarding Agile.
- What is the best approach for encouraging internal communication between different Agile teams?
- Is an Agile Forum needed or is an email distribution list 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 the compensation model with Agile teams.
- Is everyone on the team adding value?
- How should project managers who don't facilitate a team (Scrum or Scrum 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 work well in an Agile environment?
- Does the team have any say in who joins or leaves the 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 with 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?

Establish a 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?
Conclusion

The path presented above is idealized, and successfully introducing Agile methodologies into an organization is a difficult task that requires substantial experience, strong persuasive skills and a great deal of “political collateral”. Whether an organization will be successful in the transition to an Agile enterprise depends upon a few things:

- It needs to be able to manage constant change within the organization.
- It needs to be closely in tune with its customers and able to rapidly change according to changing business conditions at incremental costs (rather than exponential cost).
- It has employees who are constantly learning and innovating.

Following the roadmap here is a start, but if are currently considering Agile development or struggling with challenges along the way, consider signing up for an Agile course: http://www.scrumology.com/courses-offered.

About Scrumology

Scrumology Pty Ltd is a private held boutique consultancy that specializes in Scrum Training, Coaching and consulting. In addition to teaching public courses worldwide, we have facilitated and coached at some of the world largest companies including: Microsoft, Oracle, CapitalOne Financial, Nationwide Insurance, Getty Images, Sony and Expedia. Started in February 2009, Scrumology is lead by Kane Mar.

Kane Mar has been developing, coaching and leading software projects for the last 20 years. He has been an active member of the Agile software development community since 2001, when he first had the opportunity of working with Ken Schwaber. Mar became one of the first 30 Certified Scrum Trainers in 2006, and one of the very first Certified Scrum Coaches in 2007. He has training and coached software teams throughout North America and Northern Europe. Prior to joining the Agile and Scrum communities Mar was a Rational Unified Process (RUP) and waterfall process Project Manager working extensively with PriceWaterhouse’s SMM methodology.

For more information about Scrumology, visit our website: http://www.scrumology.com/.