Introduction

In recent years, Agile methodologies have steadily gained acceptance as a way to manage the risk of software development projects and to deliver greater value to customers. More teams and organizations are adopting Agile practices and transforming into Agile cultures.

While Agile offers great promise, it also presents challenges that can derail even the best laid plans.

The process of adopting Agile in any organization is challenging in many ways. It is especially challenging in larger organizations because of complex infrastructures, numerous legacy systems and mature organizational cultures. These larger organizations often underestimate the difficulty of getting Agile right.

This presentation will focus on the common challenges of Agile adoption. Tips are provided to help improve the chances of Agile adoption success.
Insufficient Agile experience

You need a team of people who can drive and enable an entirely new way of operating. If your existing staff lack Agile experience, it will be difficult if not impossible to successfully adopt Agile on your own.

Tip: Hire Agile as needed.

It sounds simple, but it means changing the way HR operates—the way they resource, interview, assess skills and more. For example, you’ll need to fill “scrum master”, “delivery manager” and other Agile-specific positions that will require new job descriptions. HR needs to understand what personality types fit readily into the Agile environment—that is, those who are comfortable with refactoring, evolving architecture, project rooms, writing and using stories, daily stand-up, etc.

Also, while coaching can help address insufficient Agile experience, to fully enable Agile, you also need to embed seasoned Agile practitioners within the development process to ensure that the Agile adoption sticks.
Lack of sponsorship at the executive level

Executive-level sponsorship is critical to driving the culture change that Agile requires. While a lot of the Agile adoption process is “bottom-up”, support from the “top-down” enables it. Without executive level support, an Agile adoption is at risk.

Tip: Find an Agile champion.

Ideally, you want one champion from the IT department and one from outside of it. If you can only get one, ensure that it is someone senior. It should be someone with the authority and commitment to break down the obstacles you will face during the Agile adoption process.
Attempting enterprise-wide Agile adoption without project-level Agile success

Agile requires “support from above” for a “bottom-up” approach to get things done. Trying to adopt Agile on all development teams at the same time is simply too much change at once.

**Tip:** Get executive support for an Agile development project.

Successful Agile adoptions start small. They budget for achieving a defined, initial functionality on one or two teams and just get started. Support comes from above to specific Agile teams (see diagram).
The traditional office environment

The traditional corporate environment of offices and cubicles makes Agile challenging if not impossible. Agile is based on individuals interacting with one another, ideally in face-to-face conversation. If a project team is split apart by the work environment, into separate offices and cubicles, communication will suffer, and it will be a struggle for Agile to be truly adopted.

Tip: Adapt the work environment

Altering the physical workspace is critical for an Agile adoption. The typical Agile approach puts everyone in the same project room or least contiguous space, including business, technical and operations people. You must facilitate open communication—with anyone at any time—and that can’t happen in a cubicle farm.
The tendency to micromanage

In non-Agile environments, there is a tendency to micromanage the development process. In Agile, development teams should be left to run the whole process using disciplined engineering practices and adaptive, collaborative control processes.

**Tip: Enable decision making.**

Let the team—managers, developers, quality assurance—work together and make decisions. In particular, business and domain experts assigned to a team must be enabled to make decisions on the project, such as prioritizing stories. They shouldn’t have to check with their boss every time a decision is needed. Keep in mind this principle behind the Agile manifesto: “Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”
Managers unwilling to change from commanders to team facilitators

With Agile, the project manager—whether a scrum master, delivery manager, or some other project leader—needs to be focused on removing impediments for the development team, not dictating what tasks the development team should carry out. We’ve seen organizations in multiple industries struggle with this transition, especially those with traditional project managers who are accustomed to a command-and-control system.

**Tip: Enable the team to make decisions.**

Much like our tip for combatting micromanagement, managers need to enable their teammates to make more decisions in Agile. It’s about what the manager can do for the team, not what the team can do for the manager. Coaching and embedding seasoned Agile practitioners can also help.
Unrealistic planning expectations

It is unrealistic to expect absolute certainty over project scope and budget before the project starts. Software development never actually worked that way; and now, with rapid innovations in technology and frequent changes in customer wants and needs, it’s impossible to envision the entirety of any substantial software project up front. As stated in the Agile manifesto, it’s about “responding to change over following a plan”.

Tip: Avoid comparing software development planning to other forms of planning.

The development of software is often compared to the construction of a building—a completely inappropriate analogy. A building is tangible. A significant change to the design of a building during the course of construction will be expensive and likely impossible. In contrast, software is intangible. Making changes in the course of development may or may not be difficult or expensive. For example, everyone has an intuition based on their experience of the physical world that it should be relatively cheap to change the colour of tile in a building foyer, but no one would ask to move an entire building “only” two feet to the left after the tenth storey has been added. But the same intuition does not apply to software: maybe “moving” the application from one server to another is cheap. Maybe it isn’t. No one would expect this to be impossible.
Underestimating the amount of planning in Agile

Plans for Agile projects, by their very nature, are flexible; stories can be pulled in or out of the plan as business requirements change. In Agile development, planning is an ongoing process that requires discipline to execute well.

**Tip:** Work the plan, not work to the plan.

In Agile development, planning occurs throughout the development cycle and involves the entire team. This avoids the situation where overly detailed plans made at the start of a project become out of sync with the technical and business needs as the project progresses. With Agile, you work the plan, not work to the plan. The result is a constant focus on business value, which demands ongoing planning.
Adopting Agile delivery process practices without adopting technical practices

This is challenge is all too familiar: adopting the non-technical elements of Agile while ignoring the technical elements. The benefits of Agile can not be fully realized without the adoption of the technical elements of Agile.

Tip: Adopt Agile technical practices

Agile means having the tools, skill sets, technology and engineering practices to support the overall concept. It goes beyond the ability to change what you’re building next week without jeopardizing what you’ve already accomplished. It means always having a production-ready product. Necessary capabilities include continuous integration practices and automated builds and tests. Adopt the practices of refactoring and evolving architecture. Agile technical practices are a critical yet often missed component of Agile adoptions.
Impatience

It took North American car manufacturers two decades to get a true handle on lean manufacturing; true enterprise-level Agile adoption will take many years. Large enterprises need to be patient and learn to expect that Agile is not a quick fix but instead a long-term investment.

Tip: Note where Agile is in the Hype Cycle

In Gartner’s latest Hype Cycle for Application Development (2014), Enterprise-Class Agile Development is at the peak of inflated expectations while Project-Oriented Agile Development Methodology is climbing the slope (of enlightenment). That means project-level Agile is nearly mainstream whereas enterprise-level Agile is still relatively nascent. Act accordingly.

To learn more about interpreting Hype Cycle’s, visit:
http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp
https://en.wikipedia.org/wiki/Hype_cycle

© 2015 Intelliware Development Inc.
Relying on an Agile tool to become Agile

“I’m using an Agile tool. Therefore, I’m Agile” seems to be the logic of some companies adopting Agile. While these Agile tools can certainly help, without an understanding of and support for the cultural change that needs to happen within the organization, relying on tools alone can be risky.

Tip: Get started without a tool.

There’s a lot more to Agile than tools. In fact, it’s stated right in the Agile manifesto: “Individuals and interactions over processes and tools”. Start with the basics of what it means to be Agile, and then use tools to support the initiative.
Celebrating a partial Agile adoption as though it were a full adoption

We’ve seen large companies view their Agile adoption as a success because they increased their speed of delivery, but they fail to realize that they’re still not reaping many of the benefits of a fuller Agile adoption. Speed is one part of agility; not the whole thing. Increased delivery of value and quality are also important.

Tip: Do retrospectives on your process and seek expert help.

If you know someone who has a credible perspective on Agile, get their take on your Agile adoption. There isn’t an instruction manual for Agile, so you can benefit greatly from the input of other Agile practitioners, inside or outside your organization.
Conclusion

Many companies are striving to be Agile. However, few will truly become Agile because of the challenges of adopting Agile. Yet these challenges can be overcome.

It’s critical to note that adopting Agile is not as simple as introducing and applying a methodology. Agile also requires a comprehensive culture change—a process that grows in difficulty with an organization’s size and scope.

For those who are ready for Agile, it’s important that they get this right. For those who aren’t ready, they ought to think long and hard about their organization’s responsiveness and ability to change. When your organization is ready to make the jump, make sure you understand the degree of culture change required to become Agile. When it all comes together, your ability to leverage Agile and adapt to external forces will accelerate, and the benefits of Agile — business, operational and financial—will be there for the taking.
More Information on Challenges to Agile Adoption


2. Rally’s 12 Failure Modes of an Agile Transformation: [https://www.rallydev.com/blog/agile/12-failure-modes-agile-transformation](https://www.rallydev.com/blog/agile/12-failure-modes-agile-transformation)

3. 12 Key Barriers to Agile Transformation according to Mike Cottmeyer: [http://www.leadingagile.com/2011/01/12-key-barriers-to-agile-transformation/](http://www.leadingagile.com/2011/01/12-key-barriers-to-agile-transformation/)
About Intelliware Development Inc.

Intelliware is a custom software, mobile solutions and product development company headquartered in Toronto, Canada. Intelliware is a leader in Agile software development practices which ensure the delivery of timely high quality solutions for clients. Intelliware is engaged as a technical partner by a wide range of organizations in sectors that span Financial Services, Healthcare, ICT, Retail, Manufacturing and Government.
Learn more about Agile in our *Agile Methodology Series*

- Agile Room (Team) Dynamics
- 7 Myths of Agile Development
- Agile Story Writing
- Release & Iteration Planning