



## Agile, Agile 2 and Agility, Part III: Why Agile Often Fails and What to do About It

### Description

In [the previous articles](#) in this series, we discussed the role that agile digital delivery capabilities plays in your company's competitiveness and why rapid delivery is so important. In this article, we will look at the many reasons that Agile adoptions frequently fail to deliver what companies expect and suggest some things that you should do to address them.

### Why Agile Fails

It will not require much effort for you to find innumerable sites that have published content on this subject. In spite of how obviously well-known and understood the issues are, they seem to persist. It's worth recognizing some of them here:

- Your organization has not adopted an agile management approach. You are still doing legacy funding and employing traditional project management techniques that preclude your realizing the benefits that agile delivery should provide.
- You are not **product** focused; you are **project** focused. Your oversight of the development process inherently focuses on **output** (how much work was done) rather than **outcomes** (how the product has evolved and how new features or functions were received.)
- You have not implemented appropriate tooling or repeatable agile and DevOps development processes, or you have not attained sufficient maturity with them.
- There are significant gaps in required knowledge and experience among the people who are, or should be, participating in delivering your products.
- Your middle management is pushing back, reverting to what it knows and is comfortable with, and is impeding the transition to Agile practices.
- Your product owners and product managers do not participate sufficiently with the development and delivery teams to provide continuous guidance. Instead, they only meet at intervals, often to plan and thereafter to review fixed scopes of work, such as that which was performed during sprints or program increments.
- You have not articulated how you will measure the performance of your agile organization or your agile digital delivery capabilities. You demonstrate a lack of empiricism and do not utilize the data

thrown off by your execution processes to identify and exploit opportunities or address problems.

- You do not have a commitment to continuous improvement; you do not routinely and systematically analyze performance metrics to identify where and how you can do better.
- You do not have sufficiently mature testing capabilities. Rapid agile and DevOps processes are gaited by the ability to build and deploy digital components with confidence that they will not break things. Immaturity here can impair the quality of your developed components, slow your delivery to accommodate rework and constrain your business agility.
- Your delivery capabilities are impaired by cultural issues. Successful agile delivery requires (a) consistent working behaviors, (b) knowledge and (c) constructive culture. Behaviors and Knowledge can be trained but culture must be developed and nurtured.
- You allow development that is inconsistent with or that over-complicates your architectural standards, including your Enterprise, Business, Technical, Infrastructure, Data, Knowledge and Process architectures.

## What You Need to Do

Addressing the issues that may impede your agile transformation or adoption should take place at two levels:

At the **Enterprise Level**, you need to transform into an agile company. You may remember from the previous article that an agile company treats the products and services it offers as a portfolio, which is diversified along a number of dimensions. It is explicit about its assumptions and the hypotheses it is working to prove or refute. It engages in constant experimentation to find ways to improve products or simply find ways of doing things better. It is structurally fluid, transforming itself to align to improvements in capabilities or become more responsive to the environment in which it operates. It is pragmatic, prioritizing activities so as to accelerate decision points in the lives of its products and is quick to terminate investments in products that demonstrate that they will not produce a return worth risking investment capital on. It does not blame people for their involvement in experiments that do not pan out; it just applies the knowledge gained from them to ongoing and future experiments.

At the **Operational Level**, an agile company adopts approaches that accelerate both ongoing operations and execution of developmental projects. A primary element of this is agile digital delivery, which these days is applicable to almost all products and services. The rate at which the company is able to iterate and evolve its products contributes directly to its overall agility and competitiveness.

### At the Enterprise Level

- You will need to articulate how you will measure progress and performance, implement systems and dashboards to produce the data necessary for you to do it and ensure that everyone is operating from the same source of truth.
- You need to establish a portfolio of developmental product initiatives, which may include the evolution of existing products as well as the creation of new ones. This portfolio or portfolios should be overseen and managed by the business units that will own the products that evolve from them.
- You should apply Eric Ries's™ *Lean Startup*[\[1\]](#) approach to developing the initiatives in the portfolio.

- You must adopt *product* thinking and give up on *project* thinking. You must focus on and manage *outcomes* and not *output*.
- It is crucial that you decouple the evolution of your products from any fixed calendar. As Cliff Berg, Founding Partner of the Agile 2 Academy[2] has observed, development is an event-driven process. When an issue that impedes progress arises or a new idea that appears to offer a better result is discovered, it should be evaluated and addressed as soon as it is observed. Waiting until a sprint or PI completion date is antithetical to this and is, ultimately, anti-agile.
- In regard to the previous bullet, you should integrate Product Owners and Product Managers with the Development and Delivery teams. This will give them exposure to how the development is proceeding and make them available to reset product direction or resolve questions quickly.
- You will need to distribute investment authority, within limits, to the POs and PMs to eliminate time wasted in unnecessary bureaucratic permission-seeking and approvals.
- You need to establish a multi-disciplinary Architecture team to serve as a centralized service to all development teams. This integrated group should be accountable for providing solution patterns and minimizing creation of technical and other forms of debt.
- You need to achieve constructive culture and establish positive leadership norms within your organization.

## At the Operational Level

- Establish baseline agile development and DevOps capabilities. This will include installing tools, processes, knowledge and culture and leadership.
- Require that teams articulate their assumptions and the hypotheses that underlie their design work. Have them identify at which points in the development process they will revisit them to confirm or refute them and ensure that this happens.
- Establish robust and automated testing capabilities. As indicated previously, confidence that you can release software frequently without breaking anything is essential to your velocity.
- Actively manage cross-team collaboration. Many Agile techniques, such as Scrum[3], do not address how to manage multiple teams working simultaneously on large implementations. Scaling approaches, such as SAFe or LeSS[4], can induce bureaucracy and end up being anti-agile.
- Allow teams to develop and adopt their own ways of working. Every team is different and what works for one won't necessarily work for others. That said, teams require management and leadership and allowing teams to self-manage under an emergent leader will seldom produce the best results. Supportive management does not mean no management.
- Consider alternatives to standard sprint and PI intervals. Intermediate initiative targets should be tied to completion of value-driven goals, such as enabling product capabilities, and these may not align well with fixed intervals.
- Enforce design and specification standards that promote reuse and minimize creation of technical debt. There should be a defined role for the integrated Architecture team on all initiatives and they should be brought into the development process at appropriate points, especially during early solution design activities.
- Commit to retrospection and learning. Maturing your agile delivery capabilities is something that must be done a step at a time. Doing that requires that you evaluate how you are performing at reasonably frequent intervals and ensure that the information and knowledge that surface in retrospectives is saved and made available to everyone. Typically, this has taken place at sprint

or PI end dates, but retrospectives should be scheduled at appropriate points for initiatives that are not adhering to a fixed schedule.

## In Conclusion

Competing successfully in today's™ markets requires business agility more than it ever did. A variety of factors are conspiring to expose any part of your enterprise that cannot transform with adequate rapidity and technology and evolving business models are only going to accelerate change going forward. Keeping up will require that:

- You understand the Strategy-Execution Cycle
  - It is a cycle within a cycle—strategy enveloping and dependent on digital development,
  - The faster digital development goes, the faster strategy can be implemented and refined and the more agile, competitive and sustainable you will be,
- You establish how you will measure performance and ensure that the information necessary to support it is collected, analyzed and made available to everyone so that you can all work from a single source of truth.
- You adopt agile management practices:
  - Focus on products instead of projects; outcomes as opposed to output,
  - Establish a product portfolio composed of both developmental and evolutionary initiatives. Employ portfolio management techniques, such as multi-dimensional diversification.
  - Eliminate legacy funding practices and PM techniques where they are not appropriate.
  - Adopt an experimental mindset. Be explicit about assumptions and hypotheses and prioritize your development work to accelerate your ability to confirm or refute them and then act on what you learn.
  - Distribute investment authority to accelerate decision-making and relieve the management burden of unnecessary permission-seeking.
- You develop and evolve your agile development and DevOps capabilities:
  - Install agile processes and practices (lean and flow techniques, automated testing),
  - Adopt and become expert in relevant tooling,
  - Provide training and education in your selected practices and tools,
  - Establish performance metrics and commit to continuous improvement.
- You must support each team's™ way of working and establish constructive cultural and supportive leadership norms that are crucial to success.

---

[1] [https://en.wikipedia.org/wiki/Lean\\_startup](https://en.wikipedia.org/wiki/Lean_startup)

[2] <https://www.agile2academy.com/home>

[3] <https://www.scrum.org/>

[4] <https://www.digite.com/blog/scaled-agile-frameworks/>

### Date Created

2022/04/07

### Author

howardmwiener