

# Ten Pitfalls of Enterprise Agile Adoption

Jenny Stuart, Vice President of Consulting, Construx Software

Steve McConnell, CEO and Chief Software Engineer, Construx Software

Earl Beede, Senior Fellow, Construx Software

Version 1.5 3, July 2018

While many individual teams have transitioned successfully from traditional software development methodologies to Agile, larger organizations struggle with successful adoption across the enterprise. This white paper discusses ten pitfalls that organizations commonly face as they seek to adopt Agile throughout their enterprise.

# Copyright

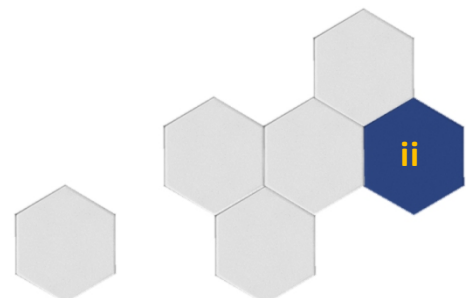
© 2012-2018, Construx Software Builders, Inc. All rights reserved.

Construx Software Builders, Inc.  
10900 NE 8th Street, Suite 1350  
Bellevue, WA 98004  
U.S.A.

This white paper may be reproduced and redistributed as long as it is reproduced and redistributed in its entirety, including this copyright notice.

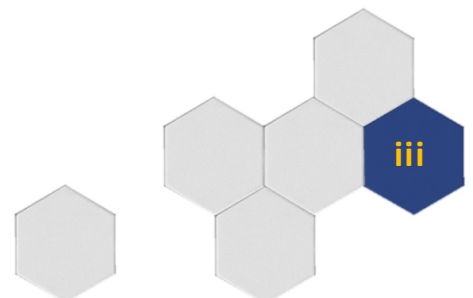
Construx and Construx Software, are trademarks of Construx Software Builders, Inc. in the United States, other countries, or both.

This paper is for informational purposes only. This document is provided “As Is” with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Construx Software disclaims all liability relating to use of information in this paper.



# Contents

- 1. Unclear Definition of Agile ..... 4
- 2. Unclear Vision for the Adoption..... 4
- 3. Accidentally Trading Flexibility for Predictability ..... 5
- 4. Failing to Account for Geographic Distribution..... 6
- 5. Inadequate Focus on Producing Value ..... 6
- 6. Assuming Traditional Requirements Problems will be Addressed..... 7
- 7. Failing to Implement Necessary Large Project Infrastructure..... 8
- 8. Failing to Align Agile with Organizational Governance..... 9
- 9. Insufficient Investment in Adoption Support ..... 10
- 10. Lack of Focus on Transformation..... 11
- Contributors ..... 12



# 1. Unclear Definition of Agile

Agile is an umbrella term that includes many techniques and approaches. It encompasses software development methodologies, including Feature Driven Development (FDD), Scrum, and Extreme Programming (XP). It includes specific practices such as daily scrums, continuous integration, pair programming, test driven development, time boxing, and so on.

One common failure of all Agile adoptions, which is exacerbated for enterprise Agile adoptions, is the lack of shared understanding about the specifics of which Agile approaches, frameworks, and practices are being adopted.

Without that shared understanding, teams throughout an organization start to use different practices and approaches. One team is doing daily stand-ups within the software development approach they have used for the last ten years. One team is using Scrum. One team is using XP. All of the teams call it “Agile.”

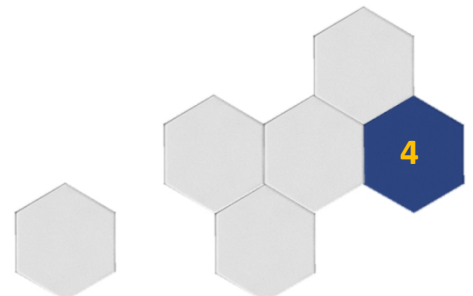
If Agile is adopted on a single team or on independent teams, it can work fine. When adopting Agile in the enterprise, however, the ambiguity of the word Agile causes confusion, even friction. The confusion can impede coordination and communication between teams.

Common agreement on, “What does Agile mean in our company?” is critical to successful adoption. Key elements such as how teams coordinate, how acceptance of completed incremental software is done, and how requirements work is performed need to be agreed upon.

Most organizations select one or two specific Agile methodologies or frameworks and a set of core practices that will be adopted throughout the entire company.

# 2. Unclear Vision for the Adoption

An additional complication of the ambiguity of the word Agile is that a wholistic vision of *why* the organization wants to be agile is frequently missing. In many organizations, management and technical staff have different, sometimes conflicting, visions of what it means to adopt Agile. This causes people to work towards different end states or believe that others are not supportive, when they actually are.



A good vision is a detailed articulation of the desired end state. For an enterprise Agile adoption, the vision needs to include:

- What do we mean by Agile?
- Why do we need to transform?
- How deep and wide is the change?
- What does the change mean to each individual?

Organizations need a clear, complete, and collaboratively-created vision of the end state to be successful.

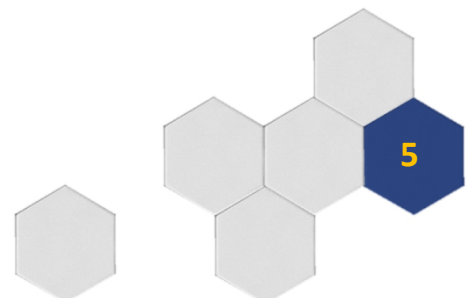
### 3. Accidentally Trading Flexibility for Predictability

A common misperception is that Agile means flexibility at all costs. In some cases, Agile teams refuse to provide cost, schedule, and/or feature commitments. This is difficult enough for a single team, but can be chaotic for businesses when projects scale to multi-teams, or when they seek to adopt Agile throughout the enterprise.

A common response to this behavior of refusing to make any commitments and calling it “Agile,” is executive resistance. This alone can kill an Agile adoption.

Most organizations need to make commitments to customers, stakeholders, shareholders, and/or business partners. When an organization moves to Agile, it needs to decide and communicate what it values most—predictability or flexibility. Is responding to change in the market place and rapidly changing the feature set most valuable? Or is the ability to commit to specific delivery dates with at least agreement on the marquee features most valuable?

The priorities of your business should drive the specific practices, techniques, frameworks, and methodologies you deploy. They drive how Agile you want to be. Organizations that value predictability will have a different approach than those that value flexibility.



## 4. Failing to Account for Geographic Distribution

Many Agile methodologies rely on the close communication and coordination of co-located teams. However, most of the companies that Construx works with today are geographically distributed. They have at least two sites, and many have six or more major locations. For most large organizations, having sites around the world is simply the way that business is done today.

Organizations and, more importantly, project teams that are geographically distributed need to account for this during an enterprise Agile adoption. The type and level of geographic distribution of your organization can impact how you deploy Agile. In some cases, organizations make changes to organizational structure or distribute work so that it requires less cross-site communication. In other cases, additional practices and infrastructure are implemented to support geographic distribution.

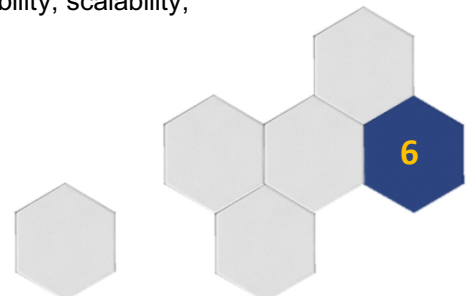
Issues with geographic distribution, along with common approaches that Construx sees organizations use to address these complexities, are discussed in our *Succeeding with Geographically Distributed Scrum* white paper.

## 5. Inadequate Focus on Producing Value

Many Agile methodologies are based on a single team focused on a specific business problem, with frequent feedback from a selected individual (Product Owner or Onsite Customer Representative) during the development process.

In a number of organizations, it's not that simple. There are often many stakeholders. It is not uncommon to have one or two dozen stakeholders, including end users, peer teams, component teams, management, quality, support, legal, and so on. These stakeholders typically have different, sometimes competing, priorities.

Beyond understanding and prioritizing the stakeholders, it is critical to know what they value. When asked, many stakeholders will quite naturally reply in terms of features, capabilities, or functions they would like to see. Few will reply with their underlying values. Stakeholders rarely understand the detailed capabilities of the tools the team is using to construct the solution. Consequently, their suggested feature, capability, or function is unlikely to be the best idea from many perspectives, such as cost effectiveness, maintainability, scalability, performance, or security.





Without a focus on understanding the stakeholders, understanding what they value, and clearly capturing and communicating that to teams, organizations can find they are back in a situation where a team or teams deliver to their customers only to find out they haven't built what those stakeholders actually need.

For more information, see Construx's *Value-Driven Delivery* white paper.

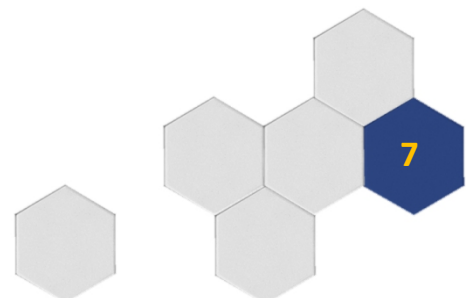
## 6. Assuming Traditional Requirements Problems will be Addressed

Having good requirements—complete, clear, unambiguous, and descriptive of what the customers really need from the system—has been a historically difficult challenge in the software industry. We all know the problems that come from the lack of good requirements—projects delivering functionality that is not what the customers really wanted.

All projects need good requirements, and Agile projects are no different. Agile will introduce new requirements techniques into the mix and has a strong focus for a business representative to work with the team. However, most organizations find that the requirements challenges they had for large projects and throughout the enterprise are not addressed simply by a transition to Agile. Existing problems with getting good requirements remain. The main difference is that with Agile, organizations find they face their requirements problems sooner, since it is hard to avoid the gaps and issues with short iterations.

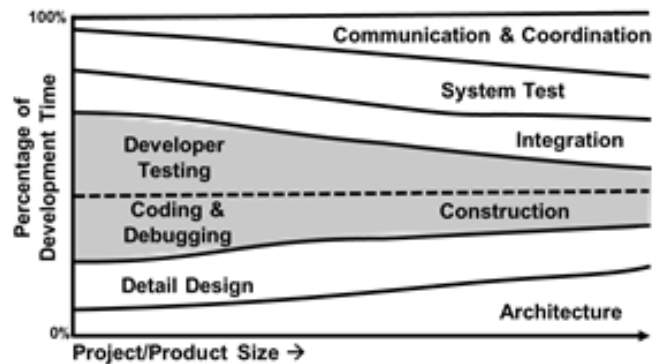
Switching to user stories from traditional requirements techniques does not magically result in good requirements. The staff still need to be proficient in the techniques. If there are skill gaps, these need to be filled.

During a transition to Agile, it is important that teams identify the current requirements problems they encounter; that they select requirements practices to help address them; that any larger organizational-level issues, such as prioritization across business units, are addressed; and that explicit decisions are made about how the business will participate and provide feedback to the teams.



## 7. Failing to Implement Necessary Large Project Infrastructure

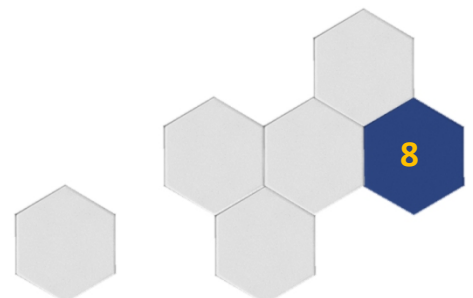
When organizations adopt Agile throughout the enterprise, they typically apply it to both large and small projects. As shown in Figure 1, the percentage of development time spent on different activities changes significantly as the size of the system increases. It is important to understand this dynamic and put in place the necessary infrastructure and practices to support large projects.



**Figure 1** Interrelationship between project size and development activities

The gap is that most agile methodologies—Scrum, XP, Feature Driven Development (FDD), and so on—are team level workflow approaches. These approaches can be highly effective at the team level, but they do not address large project architecture, project management, requirements, and project planning needs.

Enterprise Agile adoptions need to ensure there are appropriate practices in place to support small, medium, and large projects. There needs to be appropriate portfolio management and project planning to decompose the work, prioritize and assign it across teams, and manage progress. The approach structure and roles, such as Chief Product Owners and Scrum of Scrums, need to be put in place for large products. For more details, see Construx's *Succeeding with Geographically Distributed Scrum* white paper.





## 8. Failing to Align Agile with Organizational Governance

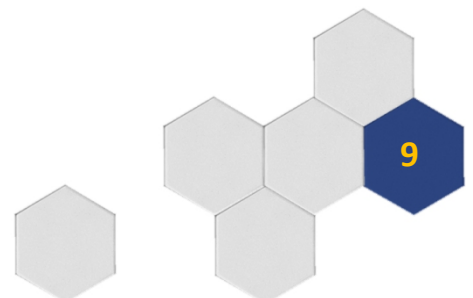
Most large organizations moving to Agile have governance considerations of some kind. Whether it is SOX compliance, FDA regulations, or a product governance process, there are typically some external constraints that must be accounted for.

During an enterprise Agile adoption, organizations need to evaluate their governance processes and Agile practices to find the appropriate balance. Construx finds the following types of problems as organizations seek to find this balance:

- Rigid waterfall-based governance models are not refined to support more Agile development practices. As a result, the organization is not able to obtain the benefits it wants from the change.
- Governance processes are thrown out because they are “not Agile,” which reduces the organization’s ability to make good business decisions on which projects to continue and which to close out.
- Governance devolves from watching over the investments of the company’s time and resources into process conformance. Both can be useful, but it’s important to separate them.

The key to success in this area is remembering what governance is for and working with the right set of people to do any needed modifications. Governance continues to be relevant in large organizations. Projects are an investment that should drive some return for the business. Governance is the tool by which the business monitors its investment. There may be modifications to the specific gates, deliverables, and timing of the activities, but governance continues to be important.

In fact, effective Agile adoptions can improve governance. The use of short iterations that deliver working functionality means that governance has “ground truth” in status. This increases the organization’s ability to determine the likely investment and whether that investment is justified by the likely return.



## 9. Insufficient Investment in Adoption Support

While Agile processes such as Scrum appear simple, they are challenging to implement well. The staff needs new skills. Old roles need to be refined and revised. Existing processes also need to be refined.

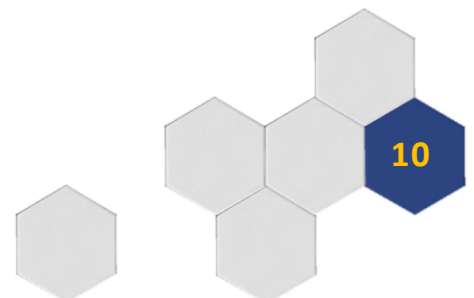
Many organizations find that their initial success with Agile in the small does not scale to larger projects or throughout the entire organization. This is often because the initial teams were made up of agile enthusiasts, whereas larger teams include people who are lukewarm about Agile or who are resisting Agile.

Enterprise Agile adoptions require investment in the staff to be successful. When Construx assesses failing or incomplete Agile adoptions, we typically find that the organization has not provided any training and coaching to its staff.

In many cases, the early adopters have trained themselves, which does not work across the entire organization. The ease of the early teams can lead the organization to believe that adoption across the entire organization will be easy. Instead, the training needs are higher for the later adopters. Later adopters are not interested in Agile, they do not train themselves, and they are often concerned that moving to Agile will change things that they have been comfortable with for years. Organizations also find they have a higher percentage of these later adopters as they scale up to larger and legacy teams. Beyond the training, teams — especially late adopters — need coaching to help them apply the techniques and work through the change period.

Training also helps the organization provide a shared vocabulary on the specific approach and techniques that have been selected for its Agile approach. A common set of terms helps to link Agile into any existing product lifecycle processes, governance processes, or common status reporting.

Organizational-level support, including agile training, team-level coaching and mentoring, executive training and coaching, is necessary for successful enterprise Agile adoptions.



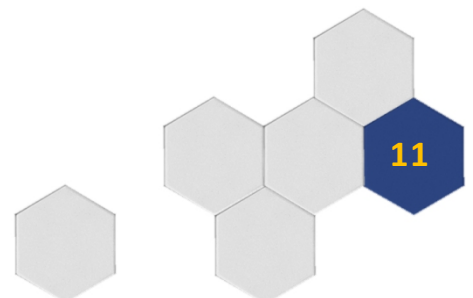
## 10. Lack of Focus on Transformation

There are two common interpretations of Agile transformation. One puts the focus on the word Agile. (Will we use Scrum? What unit test tool should we choose? When will we hold the daily stand-up?) The other puts the focus on the word transformation.

The Agile adoptions that put the focus on Agile tend to struggle and fail. The efforts that put the focus on transformation tend to be more successful.

Organizations that use Agile as a way to transform their organization outperform those that simply do Agile. The transformation to more Agile practices allows the organization to better understand and address any long-standing development issues.

For more information on how to succeed with a change, Agile or not, see Steve McConnell's *Agile Transformation Tips* YouTube videos and Construx's *Succeeding with Agile Transformation* white paper.



## Contributors



Steve McConnell, CEO and Chief Software Engineer

[steve.mcconnell@construx.com](mailto:steve.mcconnell@construx.com)

+1(425) 636-0107



Jenny Stuart, VP Consulting

[jenny.stuart@construx.com](mailto:jenny.stuart@construx.com)

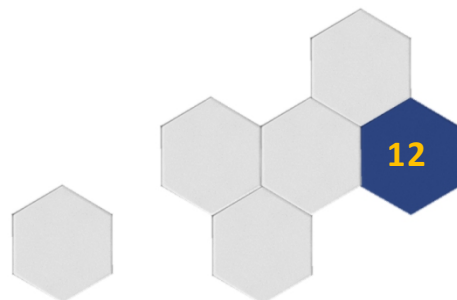
+1(425) 636-0108



Earl Beede, Senior Fellow

[earl.beede@construx.com](mailto:earl.beede@construx.com)

+1(425) 636-0114



## Construx

Construx Software is the market leader in software development best practices training and consulting. Construx was founded in 1996 by Steve McConnell, respected author and thought leader on software development best practices. Steve's books *Code Complete*, *Rapid Development*, and other titles are some of the most accessible books on software development with more than a million copies in print in 20 languages.

Steve's passion for advancing the art and science of software engineering is shared by Construx's team of seasoned consultants. Their depth of knowledge and expertise has helped hundreds of companies solve their software challenges by identifying and adopting practices that have been proven to produce high quality software—faster, and with greater predictability. For more information about Construx's support for software development best practices, contact us at [consulting@construx.com](mailto:consulting@construx.com), or call us at +1(866) 296-6300.

