Agile Selection Framework

Which Playbook Is Needed In Your League?

Ben Reyes
Principal
(214) 557-4197
ben.reyes@parivedasolutions.com

Pariveda Solutions, Inc.
Dallas, TX
Table of Contents

- Background
  - Defining the Agile Selection Problem
- The Solution
- Next Steps
- References
Over the past five years, Agile development methods have gained increased levels of adoption to a point where IT organizations must utilize them to remain competitive

- Numerous in-depth statistics from a Forrester Research report of 1,300 IT professionals showed that Agile software development methodologies are reaching mainstream levels
  - 35% said that one form of Agile or another best represented their development process
  - 21% claimed the use of at least Iterative methods
  - 13% claimed the use of Waterfall methods

- Several publications, including *The Business Value of Agile Software Methods*, captured, analyzed, and synthesized hundreds of systemic research studies which show very compelling results and further support this increase of adoption
  - Agile methods are 100% to 700% less expensive than the largest traditional methods
  - Agile methods’ ROI is 2 to 17 times greater than that of large traditional methods
  - The NPV of agile methods is almost twice as large as that of traditional methods

- At the organizational level, a survey by the Economist Intelligence Unit showed compelling results
  - Nearly 90% of executives believe that agility is critical for business success
  - Half of CEO’s and CIO’s say that agility is not only important, but also a core differentiator

- From a consumer standpoint, customers have a much higher dependency and need for innovative products and services than in the past
  - Increased levels of business engagement and intimacy with consumers is now higher
  - As a result, IT organizations need to be more Agile by increasing their level of responsiveness to the business
The value and need for Agile development methods is evident, but one of the main struggles for IT organizations is selecting the most appropriate variation of Agile

Many variations of Agile exist, but the number of choices are overwhelming and confusing
- Each variation introduces new taxonomy, buzz words, and its own prescription
- Scrum, XP, Kanban, and Lean are the leading variations, but many others offer additional attractive benefits as well
- Numerous integrated development environments (Ex. Microsoft Visual Studio, Eclipse, Netbeans, etc) include support for their own variations of Agile methodology
- The time required to evaluate and narrow in on the right variation is time consuming

When anchoring the selection of an IT organization’s Agile solution to a set of commercial variations, we often see two common problems:
- The IT organization benefits, but the value does not get passed on to the rest of the organization
  - Most commercial methods are developer-centric and biased towards benefitting the IT organization
  - As a result, performance of the entire organization becomes suboptimal
- The Agile methods selected are often not optimally fit for the environment in which it’s implemented
  - The criteria for considering a specific methodology is often arbitrary and often leads to a gap in need vs. solution
  - The Agile methodology selected is too broad and often leads to implementing tactical processes that are unnecessary

To be most optimal in Agile delivery, two main areas need to be addressed by IT organizations:
- The Agile solution should be more focused on delivering value to the entire organization rather than looking at the IT organization in isolation
- A more meaningful process should be applied to select the right Agile methods to more optimally benefit the entire organization

... to create the right solution, Pariveda believes that IT organizations should shift their mindset to selecting and implementing Agile development processes more granularly
From our experience and deep understanding of the underlying principles behind Agile, we find that every Agile methodology is comprised of the same six **Agile Core Components**

- How are requirements maintained/managed?
- How is source code configured and deployed?
- How are tests and defects managed?
- How do team members collaborate?
- How formal are artifacts documented?
- How are iterations and releases managed?
- What are the processes that are automated?
- How aligned are IT Operations teams?
- How much work in progress and throughput exists?
- How are decisions made?
- How is cost, benefit, and waste evaluated?
- What is the product owner focused on?
- What is considered “done”?
- What is the intake cadence?
- How are work units estimated?
- How are work units evaluated and prioritized?
- Who participates in requirements sessions?
- How are work units elaborated or elicited?
- What is the value of what we learn?
- What are we trying to learn?
- What is the source of our learning?
- How are we learning?
- How are we applying what we learn?
- How are work streams organized?
- What are the communication channels?
- How is the customer engaged?
- What are the team skill sets?

... the key is to tailor each of these six components so that they best match the needs of the IT organization
We also find that most development organizations fit somewhere within a Development Organization Profile model, largely driven by the uncertainty and complexity in their domains.

- The model sets some initial guidelines for profiling an IT organization and helps us analyze the internal and external factors that could impact the Agile environment.

- In today’s highly innovative and competitive landscape, the needs of an IT organization’s customers are typically very UNCERTAIN:
  - Technological change is high
  - Multiple customers with multiple voices
  - Rapid shifts in market needs

- The variability and interactions of both internal and external stimuli causes IT project environments to be highly COMPLEX:
  - Involvement of numerous contributors and stakeholders
  - Dependency on third parties and external departments
  - Size and geographic distribution of teams
  - Compliance/Regulatory requirements
  - Pre-existing technical debt within system(s)

- The cross-section of these two constraints yields very distinct environments in which to operate:
  1. **Low Uncertainty / Low Complexity** environments consist of known and unmet work with mostly specific requirements and isolated teams
  2. **High Uncertainty / Low Complexity** environments consist of high external opportunity to improve performance and address any threats of competition or trouble
  3. **Low Uncertainty / High Complexity** environments consist of numerous internally known weaknesses due to the intricacy of people and systems involved
  4. **High Uncertainty / High Complexity** environments consist of internal constraints and external opportunities and threats
When the Agile Core model is adapted across the organizational profiles, we can define four different solutions that all apply the Agile components with varying degrees of focus.

**“Highly Dynamic” Solution:**
- High focus on customer responsiveness and feedback loops
- “Fail often, learn early” mentality
- Feature teams rather than component teams
- Values technical exploration and rapid prototyping to push innovation
- Assumption-based, gut-instinct decision making
- Little penalty for reacting rather than pondering

**“Balanced Adaptation” Solution:**
- SWOT-based planning and decision making
- Balanced velocity of consumer feedback and requirements intake of Agile teams
- High emphasis on adaptive technical architecture and reusability
- Technical Architecture leadership with strong business acumen
- Hybrid of component team and feature teams
- Investment in Market and Technology advisors to reduce risks

**“Efficiency Focused” Solution:**
- For high technical debt, iterations are focused on risk/cost reduction rather than net new value
- Ideal hours and days favored over story points
- Standard visibility mechanisms such as daily scrums and task boards
- Single resources wear multiple hats and are mostly generalists
- Product Owner fills a dual agile role requiring general understanding of the customer needs and technical details of the solution

**“Highly Predictive” Solution:**
- High focus on internal efficiency at the team level rather than keeping up with customer changes
- Ideal hours and days favored over story points
- Standard visibility mechanisms such as daily scrums and task boards
- Single resources wear multiple hats and are mostly generalists
- Product Owner fills a dual agile role requiring general understanding of the customer needs and technical details of the solution
The establishment of a sound framework and thought process for selecting the right Agile playbook is essential, but full adoption of Agile requires navigating the specifics of a game.

**Agile Adoption Approach**

**Play Formation**
- Selection Framework Development

**Game Navigation**
- Assessment
- Strategy Development
- Plan

**How do we think through the problem?**
- Establish a framework to guide Agile adoption and decision-making

**Where are we now?**
- Assess the strengths and weaknesses of the IT organization
- Assess the complexity across business and IT teams
- Assess the level of risk and uncertainty that could impact delivery
- Identify and evaluate the current Agile capabilities of the organization

**Where should we go?**
- Select an Agile playbook
- Define the skillsets and structure of the team
- Define the delivery tools that are needed to support the organization
- Outline the processes to engage with customers

**How do we get there?**
- Establish a roadmap for Agile adoption
- Initiate an Agile pilot group for initial rollout
- Test high-value tactical processes
  - Facilitate user story, estimation, and release planning workshops
  - Automate build, deploy, release processes and tools
  - Initiate pair programming techniques
  - Configure collaboration and Project Management tools