AGILE IN ACTION:

Enhancing Speed from Ideation to Production

Bernie Pineau

Vice President of Application Services

RIVA

Introduction

Driven by the remarkable speed and efficiency enhancements displayed by Agile methods in the private sector, federal agencies have embraced Agile as a vital tool for modern software development, underscoring a universal emphasis on rapid delivery and value. While most agencies utilize Scrum, many teams grapple with efficiently delivering features to production. The challenge often lies in the varied implementation of Agile, as some teams don't fully value the preparation and planning of features and user stories, while others mistakenly believing that the incremental nature of Agile allows for ambiguity around requirements.

The goal of achieving a high-performing team that can collaborate, maintain accountability, and swiftly deliver ideas to production is often elusive. This white paper seeks to address this challenge by arming development and product teams with methods and techniques to maximize the output of application development teams. By understanding and implementing Agile principles effectively application development teams can successfully increase both their volume and quality of software development.

Table of Contents

The Challenges of Agile Delivery

Agile development presents a multifaceted landscape where conflicts in culture, planning, and continuous improvement intersect. From the collective responsibility of the team and the persistence of silos to the intricacies of planning and the pursuit of ongoing growth, these challenges reflect the complex nature of implementing Agile principles. By addressing the following three challenges, teams can better navigate the Agile development process, aligning with the methodology's core values and maximizing their potential for success.

Challenge 1: Fostering Team Culture

Because Agile teams often consist of siloed teams working on the same product, one team may not have transparency into another. This could increase complexity and dependencies if there is a lack of communication or a culture of blame. Successful Agile teams acknowledge a shared responsibility and embrace a growth mindset to deliver high-quality work products. This focus on collaboration will be especially important as we welcome the new teammate: AI. While the AI revolution will yield new tools and processes, Agile team culture will need to be fostered by good, old-fashioned humans.

Challenge 2: Busting the Myth that Writing Code is More Important Than Planning

The team breakdown above can also lead to a lack of shared awareness and understanding of the big picture. When teams are not aligned on the overall product goals and priorities, it weakens their ability to plan and maximize feature development. Poorly crafted user stories created solely by the product owner and not reviewed with the team until the planning session can lead to many questions and an unproductive planning session. Definition of Done (DoD) and Definition of Ready (DoR) must be paired and in balance. Focusing on a DoD before ensuring a clear DoR can result in commitments to work that the team doesn't fully understand. The assumption that "the team will figure it out as we go" may seem aligned with the flexible nature of Agile but, it can cause delays and misalignment.

Challenge 3: Metrics are Only Aligned to Output

The pursuit of continuous improvement is a core principle of Agile development, yet it can often be implemented superficially. One manifestation of this challenge is in the handling of metrics.

While metrics are often captured, they may not be consistently reviewed and used as a basis for ongoing improvement. This lack of regular assessment can lead to stagnation, where the same issues persist over time without being addressed. To be valuable, metrics must lead to actions, not just data points, that can drive positive change to enhance the development process.

Another aspect of this challenge relates to the narrow focus when measuring flow which refers to the smooth and efficient movement of work from conception to completion. If flow is measured solely by the time it takes to develop a user story, it overlooks the broader context of the development process. This narrow perspective can limit the understanding of potential bottlenecks, inefficiencies, or areas for improvement.

Accelerating Production: Strategies for Increasing Speed

This might sound counterintuitive, but if you want to go faster, spending more time planning will help mitigate the challenges listed above.

Focus on Optimal Outcomes and a Shift to a Growth Mindset

The Agile development process thrives on a culture that emphasizes growth, innovation, continuous improvement, and achieving optimal outcomes. To foster this culture, organizations must make deliberate efforts to empower and reward team members. Here's how this can be achieved:

- Empowerment to Say No: Team members will be empowered to respectfully push back when things are not ready if there is a culture of joint ownership. Encourage a culture that receives this feedback openly with a focus on the greater good.

- Encourage Calculated Risks: Allow the team to take calculated risks, especially when it can lead to innovations. This fosters a culture of creativity and exploration. For example, when a customer needed to upload multi-gigabyte CSV files but was limited to 50MB uploads, team RIVA developed an API using streaming technology. This not only solved the customer's problem but also added a new, innovative feature for machine-to-machine data uploads.

- Reward Growth Behaviors: Recognize and reward behaviors that contribute to growth and collaboration. For example, reward the team for acting on rather than ignoring problems, such as a broken build. Rewards do not have to be monetary; they could be something fun, additional time for exploring innovations or something as simple as public recognition.

- Set Achievable and Stretch Goals: For each sprint, set goals that are both achievable and challenging. Reward members who achieve stretch goals, reinforcing a culture of ambition and excellence.

- Promote Listening and Experimentation: Create an environment where team members listen to each other and are encouraged to experiment and learn. Promote flexible creativity time by creating space in the schedule for innovative thinking and problem-solving.

Teams that grasp their desired outcomes and swiftly embrace a growth mindset can accelerate their progress with enhanced openness, collaboration, focus, and adaptability. By amplifying these elements, they pave the way for high-performing teams that rapidly deliver exceptional value to their customers. This aligns with the agile principles and fosters a positive and productive culture where team members are motivated to excel, innovate, and contribute to the overall success of the development process.

Plan More, Code Less

In Agile software delivery, the mantra of "Plan More, Code Less" provides clarity in what needs to be built. It's essential for the development team to grasp both the immediate tasks and the broader picture. This means understanding the answers to pivotal questions: What is our end goal? How does this contribute to the business? Does it fit within our product and release framework? Is the work outcome-oriented? Having this clarity aids in the timely delivery of projects. Here are some ways this can achieved:

- Define Outcome-Based Metrics: Outcome-based metrics concentrate on the results, or the impact of the work product will have on the business. Avoid output based metrics which focus on the output such as story points, number of stories, etc. Agile teams focus on building value not requirements. Each progressive sprint should increase the value of the product and every sprint should define desired outcomes which align with overall product outcomes.

- Define "Ready": Definition of Ready (DoR) is a set of agreements, established by the team, that determine when a user story is ready for commitment. By enforcing a DoR you reduce the roundtrips that might arise from ambiguous user stories committed to a sprint.

- Define "Done": Definition of Done (DoD) is a set of agreements that define when a user story is completed. Like DoR, enforcing a DoD can help reduce ambiguity in user stories which also can lead to expensive roundtrips.

- Build Less with Minimum Viable Products: Rather than focusing on building more features, focus on building more value into fewer features. This promotes a streamlined approach to product development, ensuring that resources are invested wisely in creating a product that truly meets the needs of the user and the business and enables a faster time to market.

- Don't Skip Backlog Refinement: A well refined backlog drastically improves a team's ability to deliver features quickly. This critical process ensures that user stories are decomposed through regular refinement workshops led by the Product Owner. It includes defining clear narratives, acceptance criteria, outcome alignment, and operational attributes, ensuring that stories are "ready enough" for successful delivery.

- Use Visual Techniques and Tools: Visuals enable teams to quickly form shared understanding. A visual technic known as User Story Mapping can help accelerate the organization and prioritizing of complex products.

Defining "Ready" in Agile

In Agile development, it's crucial for teams to have a mutual understanding of two specific User Story states: the Definition of "Ready" and "Done." Each definition provides clarity on when a User Story is set to be included in a sprint and when it's completed. For a User Story to achieve the status of "Ready," consider adhering to the following best practices:

- The User Story aligns with sprint objectives and project outcomes
- The Acceptance Criteria encompasses functional, non-functional, technical, and User Experience aspects (as appropriate)
- The criteria are vetted by each team role (development, testing, user experience, solution architect)
- The User Story is articulated succinctly and lucidly
- All dependencies, risks, and blockers are noted within the User Story
- It's assessed by UX, Technical, Test, Solution Architect teams and supplemented with relevant implementation details such as UX wire frames or API specifications
- The Product Owner and Customer have given it their stamp of approval
- It's been sized appropriately
- The story's scope allows for its construction and testing within a single sprint
- It has the consensus of the key triad: Developer, Tester, and Product Owner
- An attached test plan has been reviewed and approved by the triad
- The User Story is cross-referenced with other stories to ensure architectural compliance

This checklist, akin to the definition of "Done," should be systematically applied to every story as decided upon by an Agile Team. Once all items are validated, the User Story can confidently be labeled as "Ready" for a sprint. Ideally, your backlog will visually differentiate User Stories by status, distinguishing between "Ideas," "In Refinement," and "Ready."

Backlog Refinement

A well refined backlog will greatly accelerate the planning and execution phases of the development process. The intent of backlog refinement is to decompose ideas first into Epics and then into user stories over time. Backlogs that meet the DoR can be planned in minutes, will require fewer roundtrips, and will have a greater chance of meeting the DoD during a sprint.

Figure 1 visualizes two work streams in the lifecycle of an application. The top shows a user story beginning as an objective and steadily maturing with stakeholder input to meet the DoR for the team. Enriched with implementation details, the bottom workstream shows the sprint planning and execution, leading a user story to the DoD. Just like a car would drive much faster on a road that's been pre-paved versus paving as you drive, the plan and execute workstream will be much more efficient if the user stories are "ready."

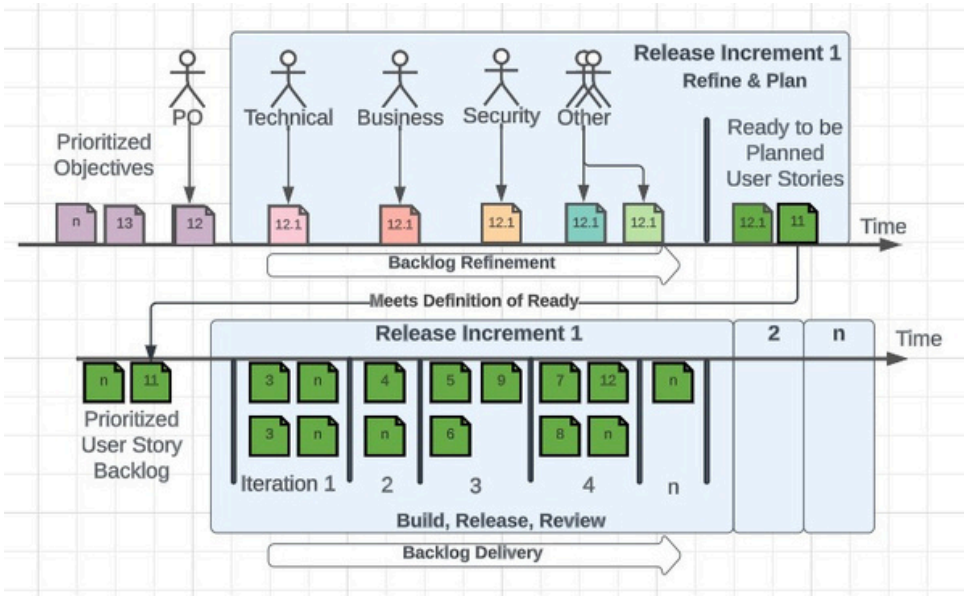

Figure 1

Many teams say they perform backlog refinement however if your teams experience some of the symptoms below, there is a good chance there are inefficiencies:

- Prolonged planning sessions
- Team confusion during active sprints
- Persistent rollover of User Stories across sprints
- Incomplete features over multiple sprints
- Reluctance to commit due to unease

One way to combat these inefficiencies is to form a forward-facing team that is two to three sprints ahead. This team would represent the top work stream from Figure 1, paving the road for the agile development team.

The second approach is to incorporate User Story refinement sessions at various stages of the Sprint. A good guideline for setting aside time for these sessions is 4 to 8 hours over the course of a two-week sprint. By dedicating adequate time to proper refinement, teams can complete sprint planning in less than 30 minutes. This efficiency allows the team to allocate more time to ongoing improvement efforts, such as Sprint Retrospectives.

Increase Collaboration with Story Mapping

Story Mapping aims to gain consensus and provide clarity for a product's functionality and developmental trajectory, aiding in the prioritization and sequencing of feature development. Product teams often can lead the Story Mapping workshops with input from the agile development teams and other product stakeholders.

This technique, intentionally collaborative, is suitable for both virtual and in-person workshops, and is adept at highlighting gaps, risks, and dependencies throughout a product. Additionally, it ensures the alignment of releases with intended outcomes.

Story Mapping can be integrated at any phase of a product's lifecycle. Figure 2 shows an example Story Map created by our team for a fictional product named Alpaca Tours. The top of the map identifies the personas, followed by specific activities the user does. Each activity may have one or more epics which will in turn have one or more stories.

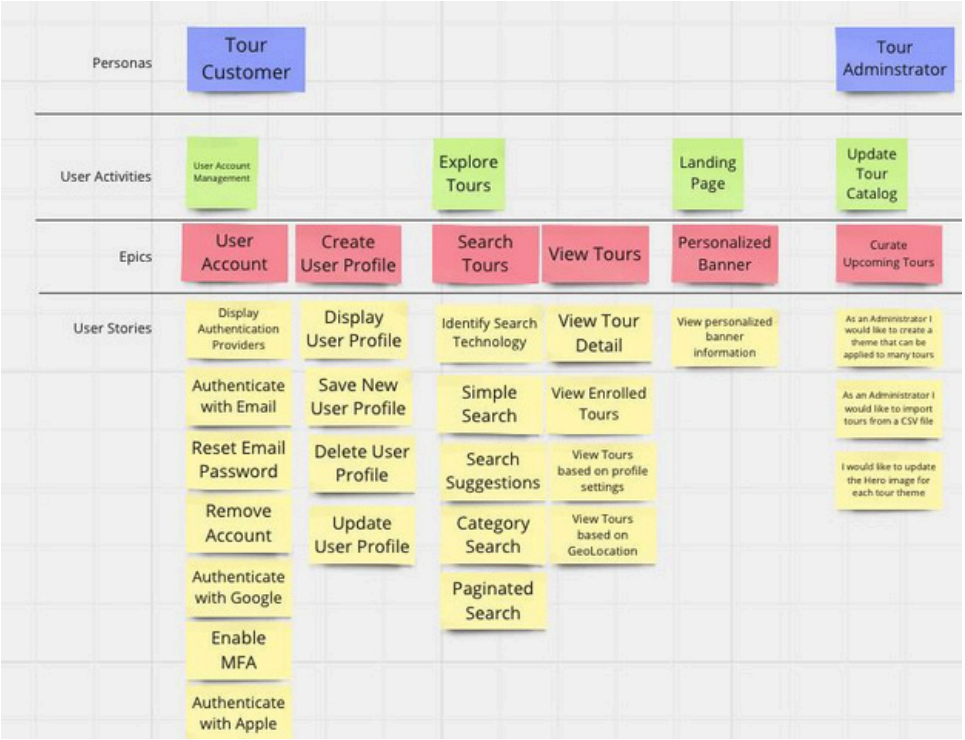

Figure 2

This visualization of the product narrative through the actions of its Users and can be a great way to expand on traditional planning tools like Jira Software. Integrating Jira Software with tools such as Miro, enables Miro notes to be turned into Jira issues with the click of a button.

Culture of Measurement and Reflection

Creating a culture rooted in measurement and reflection is paramount, not merely for tracking progress but for ensuring that every action is both purposeful and productive. Metrics not only provide the lenses for understanding the current state but also the foresight to anticipate future trajectories. When we dive deep into metrics, two measurement domains emerge: Outcomes, and Flow.

Measure Outcomes and Actionable Outputs

Metrics play a crucial role in providing insights into a development team's performance and the overall value the end user receives from using the product. While output metrics provide valuable information about the quantity and quality of work produced, they can focus narrowly on the team's immediate production and may not fully capture the impact of the code on the overall success of the sprint.

On the other hand, outcome metrics, such as ' Improve customer satisfaction by 20% in the first quarter' or ' Achieve a 90% reduction in critical vulnerabilities detected in quarterly security assessments,' provide a more comprehensive view of the real-world effects of the team's efforts. This approach highlights more avenues to improve efficiency and identifies feedback cycles that might be hindering market responsiveness and competitiveness. Therefore, defining metrics for both outputs and outcomes is vital for helping the team grasp what success looks like throughout the entire process and how to measure progress effectively.

Measure Flow to Identify Bottlenecks

Flow focuses on the operational efficiency of processes. How seamlessly can we translate an idea into value for the customer? Metrics within this domain illuminate bottlenecks and operational hurdles. They allow teams to refine their workflows, ensuring that the transition from task inception to completion is smooth, efficient, and free from unnecessary impediments enabling teams to stay agile and adaptive. Flow metrics to measure and reflect upon are:

- Flow Distribution: This metric reveals the balance between technical debt, new features, and defect resolution, prompting teams to evaluate the reasons behind their work focus.

- Flow Velocity: Flow Velocity gauges work completed per sprint and aids in planning, allowing for performance tracking and future commitment forecasting.

- Flow Cycle Time: This measures the time taken from backlog to production, highlighting bottlenecks and offering chances for process optimization.

- Cumulative Flow: Looks at how long it takes for each work item to transition from idea to closed over time. Cumulative Flow is normally visualized through stacked line charts known as a Cumulative Flow diagram.

- Flow Efficiency: This ratio between active work time and total time points out bottlenecks and inefficiencies, guiding process refinement. High ratios signify efficient flow while low ratios indicate bottlenecks.

- Flow Predictability: This evaluates the team's forecasting accuracy for work completion, guiding estimations while acknowledging inherent uncertainties.

Benefits of Implementation

After reviewing the challenges, solutions, and implementation approaches, it's time to delve into the benefits. The objective was to accelerate the pace, and through collaborative efforts, meticulous planning, and ongoing refinement, teams can indeed achieve this speed in addition to the following value-adds:

- Enhanced Collaboration and Delivery Efficiency. Enhanced collaboration and efficiency are at the heart of Agile development. By fostering teamwork, shared responsibility, and clear communication, teams can work more cohesively and with greater alignment. Improved planning reduces delays and ensures goals and priorities are well-understood across the team. These elements contribute to faster development cycles, enabling faster agile delivery each sprint.

- Innovation and Quality Assurance. Encouraging creativity and risk-taking allows teams to explore new ideas and solutions, fostering a culture of continuous improvement. Quality assurance ensures products meet established standards and reduces defects, enhancing customer satisfaction. By enabling faster feedback cycles, teams can quickly test and validate their assumptions, leading to more valuable product delivery.

- Positive Organizational Culture and Team Satisfaction. Building a culture that emphasizes accountability, readiness, and pride in work enhances employee satisfaction and retention. Allowing teams to see the impact of their work leads to a sense of accomplishment and fulfillment. Happier teams are more productive and engaged, contributing to a more streamlined and successful process.

- Strategic Focus on Outcomes and Outputs. By aligning output to meaningful outcomes, teams can create products that resonate with the target audience. This strategic alignment ensures that every sprint delivers value, contributing to faster Agile delivery and building products that truly meet the needs of the customers.

Conclusion

Agile methodologies have revolutionized the landscape of software development, particularly within federal agencies over the past decade. However, the full potential of Agile is often not realized due to inconsistent implementation and misunderstandings surrounding agile ceremony intent. This white paper has aimed to equip development and product teams with actionable methods and techniques to optimize their Agile practices. By placing a strong emphasis on proper preparation, planning, and continuous improvement, teams can overcome common pitfalls and significantly enhance both the volume and quality of their software output.

The journey to becoming a high-performing Agile team is not a one-time effort but a continuous process of learning and adaptation. Organizations must commit to understanding the nuances of Agile, measuring their performance, and iterating their practices for sustained long-term success. By doing so, they not only maximize the output of their application development teams but also contribute to a culture ofexcellence, accountability, and collaboration.

Bernie Pineau
Vice President of Application Services

BPineau@rivasolutionsinc.com
www.rivasolutionsinc.com

RIVA