

# TechSur Solutions White Paper: “Aligning Agile Development Practices with Full Lifecycle Support Requirements”

In this white paper, we will:

- Discuss the complex portfolio of the USCG and the crucial need for consistent Agile application.
- Highlight the role of inter-program dependencies in fostering enterprise-wide collaboration.
- Delve into the significance of establishing enterprise-wide Agile standards.
- Make the case for prioritizing agile teams, tooling, and culture over rigid processes.

*The ultimate goal of aligning USCG Agile development practices with full lifecycle support requirements is to accelerate value delivery. Let’s get started.*

## Introduction

Navigating the vast seas of digital transformation, the [United States Coast Guard \(USCG\)](#) faces the immense challenge of managing its complex portfolio throughout their entire lifecycle. Agile development methodologies offer a compelling solution, enabling iterative, customer-centric, and collaborative practices. By integrating Agile principles into full lifecycle support, **the USCG can benefit from increased responsiveness to changing requirements, improved collaboration among stakeholders, and a faster delivery of high-quality software products.** This white paper charts the course for this Agile transformation, placing emphasis on building teams, tools, and a culture conducive to change.

## Considerations Unique to the Coast Guard

---

*The USCG's Complex Portfolio and the Imperative of Consistent Agile Application*

---

The varied mission objectives and legacy systems inherent in the USCG's technology portfolio demand a consistent and standardized approach to Agile. Consistency ensures predictability, aligning disparate projects and processes under a unified framework, facilitating smoother transitions and more efficient workflows.

---

*Inter-Program Dependencies and Enterprise Collaboration*

---

Given the interlinked nature of USCG programs, identifying and managing dependencies is paramount. A centralized Agile approach, complemented by effective tooling, provides a holistic view of interdependencies, allowing for real-time adjustments and more informed decision-making. This not only streamlines operations but also fosters a culture of enterprise-wide collaboration.

---

### *The Need for Enterprise-Wide Agile Standards*

---

While each mission or project might have its unique challenges, establishing enterprise-wide Agile standards ensures a baseline level of quality, performance, and delivery. It minimizes variability, aids in training and onboarding, and ensures that irrespective of the mission, the USCG can expect a standard of excellence.

---

### *Prioritizing Agile Teams, Tooling, and Culture over Processes*

---

While processes play a role in any organization's success, **the heart of Agile lies in its people, tools, and culture**. Building resilient Agile teams, investing in the right tooling infrastructure, and cultivating an adaptive culture are the cornerstones of true Agile transformation. When these elements align, processes naturally evolve to support the overarching Agile framework.

---

### *Ultimate Goal: Accelerated Value Delivery*

---

Incorporating Agile consistently across the USCG leads to one primary outcome: accelerated value delivery. By ensuring collaboration, recognizing interdependencies, and maintaining standards, the Coast Guard can swiftly pivot in response to changing requirements, resulting in faster, more efficient delivery of mission-critical objectives.

## Challenges in Aligning Agile Practices with Full Lifecycle Support

Implementing Agile methodologies within an agency's lifecycle support is not without its challenges. Although Agile was primarily developed for small teams and not necessarily tailored for maintenance or legacy systems, it can be effectively integrated with the right strategies. We explore potential issues and the corresponding strategies rooted in the key considerations shared earlier.

### 1. AGILE + LIFECYCLE SUPPORT CHALLENGE: Legacy Systems

Agile practices are most effective when implemented from the initiation of a project, as described in a [LinkedIn post](#) by a Software Quality Engineer at HP. However, [many agencies still maintain legacy systems](#) which are ingrained in the organization's IT infrastructure and processes, so introducing Agile practices to these systems can be challenging. These legacy systems, often non-documented,

complex, and brittle, pose difficulty due to their lack of flexibility. Adapting such systems to Agile practices requires efforts in rewriting or refactoring the system.

- Enterprise-Wide Collaboration: To address the integration of Agile with legacy systems, foster a culture of enterprise-wide collaboration. By pooling knowledge from different teams familiar with the legacy systems, the USCG can better navigate the introduction of Agile practices into these systems.
- Consistent Agile Application: Rather than a complete overhaul, a consistent and gradual application of Agile methodologies tailored for legacy system integration can ensure smoother transitions. This may involve identifying specific modules for refactoring rather than rewriting the entire system.

## 2. AGILE + LIFECYCLE SUPPORT CHALLENGE: Technical Debt

Another challenge in aligning Agile practices with lifecycle support is managing technical debt, which refers to the implied cost of additional work caused by choosing an easy or quick solution over a better approach that would take longer. As pointed out in TechSur's article "[Charting a Course Towards Efficiency: Portfolio Rationalization and Reducing Technical Debt in the USCG](#)", technical debt often accumulates over time, especially when organizations prioritize speed over quality. While Agile methods aim for quick iterations and constant delivery, if not managed correctly, they can lead to an increase in technical debt.

- Accelerated Value Delivery: Addressing technical debt aligns with the goal of accelerated value delivery. By dedicating specific Agile sprints to tackle areas of high technical debt, USCG can ensure that value isn't compromised by suboptimal past decisions.
- Enterprise-Wide Standards: By establishing clear standards, the USCG can ensure quality is maintained, thereby reducing the creation of new technical debt.

## 3. AGILE + LIFECYCLE SUPPORT CHALLENGE: Frequent Changes

Agile methodologies thrive on change, with an emphasis on adaptability and continuous improvement. However, in a full lifecycle support context, frequent changes can be a double-edged sword, making the system more vulnerable to bugs and instabilities.

- Building Agile Teams and Culture: Equip teams with the tools and culture that prioritize thorough testing and quality checks, reducing bugs and instabilities despite frequent changes.
- Consistent Application: While Agile thrives on change, a consistent application ensures that it's not change for change's sake but for genuine improvement. The right balance can be struck with careful planning and retrospection.

## 4. AGILE + LIFECYCLE SUPPORT CHALLENGE: Large-Scale Systems

Large-scale systems present a unique challenge for Agile practices as highlighted in the literature review "[Challenges and Success Factors for Large-Scale Agile Transformations](#)." Agile methodologies were originally designed for small, co-located teams and projects. As organizations scale up and become more complex, with multiple teams often distributed across different locations, the application of Agile practices can become difficult.

- Inter-Program Dependencies and Collaboration: Large systems naturally have interdependencies. Recognizing and managing these can make the Agile implementation more streamlined.
- Tooling: Use of proper tooling can help in managing large-scale Agile transformations, enabling teams spread across different locations to remain in sync and work effectively.

Overall, while **Agile practices offer numerous benefits, including improved team collaboration, faster time-to-market, and better adaptability to changing business environments**, aligning these practices with full lifecycle support presents challenges. Careful consideration and proactive management of these potential obstacles are essential. TechSur Solutions helps agencies successfully adopt and benefit from Agile methodologies in the context of lifecycle support.

## Solutions to Potential Challenges

To effectively align Agile development with full lifecycle support, **we help agencies adopt several key strategies that promote collaboration, adaptability, and continual improvement**. These strategies facilitate seamless integration between Agile principles and the comprehensive lifecycle needs of modern and legacy systems:

- **DevSecOps Adoption:** use [TechSur's DevSecOps Platform](#) to bridge the gap between development and operations teams. DevSecOps emphasizes automation, continuous integration, and continuous delivery, enabling faster and more reliable software releases. By breaking down silos and fostering cross-functional collaboration, DevSecOps ensures that Agile development efforts smoothly transition into operational support and maintenance and keeps security at the forefront of every implementation.
  - *Commercial Example:*

With its Azure DevOps platform, Microsoft combines development, testing, and operations, promoting collaboration and automation across the entire software development lifecycle. By integrating security practices into each stage of development, [Microsoft ensures that security remains a top priority throughout the software delivery process](#), empowering teams to release software faster while maintaining the highest standards of security and reliability.
  - *TechSur Example:*

Our team supports DevSecOps pipelines using zero-trust and security-first philosophies. We incorporate multistage checks, including Twistlock and Fortify, to detect and prevent software vulnerabilities from being deployed to production. Our DevSecOps pipeline enforces security from the first code commit through each quality gate. Within their Integrated Development Environment (IDE), developers integrate tools, such as SonarLint, to receive the earliest feedback on code issues and address security defects.

TechSur advocates for the refactoring of applications from the legacy monolithic model to a cloud-native DevSecOps support model. We utilize cloud Infrastructure as a Service (IaaS) and continually infuse innovation into operations. This includes introducing serverless containerization platforms like EKS and piloting the GitOps methodology using ArgoCD, leading to a reduced change failure rate. Additionally, TechSur manages a shared platform infrastructure, including the CI/CD pipeline, which enables development teams to deliver functionality quickly and frequently. By applying state-of-the-art engineering practices, TechSur delivers new code and business functionality to production daily with zero downtime. We also offer the potential for supporting multiple deployments if required. This platform supports microservices web-based applications using a DevSecOps model and cloud IaaS,

including a help/service desk.

- Agile Architecture: we implement an Agile architecture approach that supports iterative development and frequent updates. Agile architecture focuses on creating a flexible and scalable design that accommodates evolving requirements. By continuously refining the architecture throughout the development process, teams can address technical debt, avoid rework, and maintain a system that aligns with the changing needs of stakeholders.
  - *Commercial Example*:  
Spotify, the digital music service provider, exemplifies Agile architecture in action. Spotify's "[Spotify Model](#)" emphasizes autonomous squads (small cross-functional teams) that can independently deliver parts of the product. This approach enables iterative development, frequent updates, and ongoing support for various components of their platform. By fostering a flexible and scalable architecture, Spotify successfully manages a vast user base while continuously improving and adapting their platform to meet changing user preferences.
  - *TechSur Example*:  
TechSur follows a sprint-based approach in project management, which is a time-boxed iteration typical of Agile methodologies, such as Scrum. A sprint is a short period during which a development team works on a set of prioritized user stories or tasks to produce a product increment. The Agile Scrum methodology enables a consistent flow of requirements to be planned and delivered in sprints, typically ranging from 2 to 4 weeks in duration. This Agile strategy facilitates the progressive rollout of systems and features to users. TechSur boasts a proven track record in successfully delivering projects with short durations. We prioritize collaboration, communication, and continuous improvement.
- Iterative Development: our engineers emphasize iterative development practices, where software is developed in small, incremental steps, thereby breaking down the Software Development Life Cycle (SDLC). Each iteration delivers a potentially shippable product increment. This approach enables early and frequent feedback from users, stakeholders, and end-users, allowing for rapid adjustments and improvements. Iterative development ensures that lifecycle support requirements are integrated into each stage of the development process.
  - *Commercial Example*:  
A prominent example of iterative development is Google's approach to developing and maintaining its popular products like Google Search and Google Maps. Google continuously releases small, incremental updates based on user feedback and data analysis. This iterative process allows them to [respond quickly to changing user needs, refine features, and address any issues that arise promptly](#). By embracing iterative development, Google ensures that their products consistently evolve to deliver the best user experience.
  - *TechSur Example*:  
Our Agile processes deliver high-quality, timely solutions that effectively meet the needs of our customers. During the Project Kick-Off and subsequent engagements, we validate the program's background and objectives, identify key stakeholders including Product Owners (POs) and the Integrated Project Team (IPT), and establish the program timeline. We also review the Agile operating model and tools used for delivery. Below are the steps we follow for a single sprint, noting that a project deliverable consists of multiple sprints:

- **Sprint Planning:** The Product Owner prioritizes the product backlog items (e.g., user stories, tasks, or features) based on business value. The development team then selects a subset of these backlog items to work on during the sprint, considering their capacity and capability. Together, they decompose the chosen items into actionable tasks. Our team then estimates the effort for each task (typically in story points), which assists in determining how many items can be finished within the sprint's duration.
  - **Sprint Goal:** The team sets a clear and concise sprint goal that encapsulates the objective or outcome they seek to achieve during the sprint. This goal serves as a focal point, guiding the team's decisions throughout the sprint.
  - **Sprint Execution:** During this phase, the team implements and delivers the agreed-upon functionality. They also hold daily Scrum meetings to update on progress, discuss issues, and synchronize their efforts.
  - **Collaboration and Adaptation:** The team collaborates closely, seeking clarifications from the Product Owner when necessary and adapting plans based on new insights or evolving requirements.
  - **Incremental Development:** The team emphasizes delivering a potentially shippable product increment, ensuring regular integration and testing.
  - **Sprint Review:** At the end of the sprint, a meeting is convened to showcase the completed work, solicit feedback, and engage stakeholders in suggesting modifications or voicing concerns.
  - **Sprint Retrospective:** The team takes a moment to reflect on their processes, teamwork, and outcomes. They identify potential improvements and strategize on enhancing future sprints and overall performance.
  - **Repeat and Iterate:** The cycle resumes with sprint planning, where backlog items are selected, and a new sprint goal is set. This iterative process ensures that value is delivered incrementally, and performance is continuously optimized.
  
- **Product Backlog Management:** we help you maintain a well-prioritized and continuously groomed product backlog. The product backlog serves as a repository of user stories, feature requests, and technical tasks. Regular backlog refinement sessions enable the team to assess priorities, adjust requirements, and align development efforts with evolving support needs. A dynamic product backlog facilitates seamless planning and execution of Agile development while accommodating lifecycle support requirements.
  - *Commercial Example:*

Atlassian's Jira software is a [leading example of effective product backlog management](#). Jira serves as a comprehensive tool for maintaining product backlogs, tracking issues, and managing development tasks. It facilitates backlog refinement sessions, enabling teams to assess priorities, adjust requirements, and align development efforts with support needs efficiently. Jira's flexibility and customizability make it an asset for Agile teams, streamlining the planning and execution of development while accommodating ongoing lifecycle support requirements.
  - *TechSur Example:*

TechSur recognizes the importance of maintaining detailed and comprehensive documentation for application development and O&M tasks. We consistently

update system roadmaps, product and release backlogs, and other pertinent documents in JIRA and SharePoint. This ensures both customers and Team TechSur have complete transparency and can actively collaborate.

Throughout the development process, we proactively engage with Product Owners (PO) to comprehend requirements. Together, we refine user stories and acceptance criteria to ensure they accurately represent desired functional and non-functional requirements. Our business analysts work closely with stakeholders, conducting the necessary business and technical analyses to maintain high-level system roadmaps. They also develop user stories which are incorporated into project epics, prioritized, and delivered incrementally based on an agreed-upon iterative sprint plan.

Our approach guarantees that our solutions cater to specific needs. Once a user story is finalized, it must meet the set acceptance criteria before being marked as completed. The PO, aided by subject matter experts and business analysts, determines the satisfaction of these criteria. This strategy ensures we consistently produce quality solutions aligning with the PO's expectations.

Our scrum team collaborates to refine requirements, business rules, and processes to develop the product backlog. Any new requirements supporting program goals are documented and maintained in JIRA's product backlog. We conduct backlog grooming sessions to validate understanding, acceptance criteria, and approval, organizing the backlog using epics and features. This organization aligns with the product roadmap, simplifying planning and tracing production releases.

Each two-week sprint commences with a sprint planning session. Here, Product Backlog Items (PBIs) are selected based on the PI planning session and the Product Roadmap. During sprint planning, our Scrum Master shares an agenda and a list of PBIs to discuss defects, requirements changes, technical needs, acceptance criteria, and item prioritization. This planning ensures no dependencies exist between user stories slated for the sprint. Our development team utilizes the Fibonacci method to assign points to stories, factoring in development time, testing phases, and documentation updates. Only approved stories with PO-sanctioned acceptance criteria are considered. Post-planning, key points are updated in the Product Backlog, and the Agile Project Sprint Plan is revised. Once a sprint is outlined, user story or scope changes within that sprint are not entertained. Any alterations are noted as new user stories for upcoming sprints.

To guarantee the reliability of our software, we advocate for a "test early, test often" methodology. Testing is integrated into each sprint, encompassing unit tests during individual user story development and integration testing for broader functional units. This ongoing testing identifies and resolves issues early, promoting stable releases and efficient resource utilization.

By adopting these strategies, agencies can bridge the gap between Agile development and full lifecycle support, fostering a culture of collaboration, adaptability, and continuous improvement. Agile practices will be integrated into the various stages of software development, [ensuring that modern and legacy C5I systems receive the necessary support throughout their entire lifecycle](#). The outcome is a more

responsive, efficient, and customer-centric approach that **aligns with the Coast Guard's IT objectives and empowers decision-makers to deliver value to stakeholders while effectively managing ongoing support and maintenance.**

### Mini Case Studies: Commercial Companies Provide Examples

Read on for concrete examples of how commercial organizations have successfully integrated Agile development with full lifecycle support. Each mini case study discusses the strategies used, the challenges faced, and the outcomes.

- In a collaborative project between Shell and IBM, the [Agile-based IBM Garage™ Methodology](#) was implemented to co-create, co-build, and co-operate the OREN project. Alongside Agile, the team integrated DevOps principles to maintain project focus and expedite delivery. This real-world application demonstrates how Agile practices can be seamlessly incorporated into full lifecycle support, with the project launched successfully within just eight months. The case underscores the growing relevance of Agile methodologies and open cultures in a rapidly evolving landscape.
- [Salesforce](#), a leader in customer relationship management (CRM) software, has integrated Agile principles into its engineering culture by offering a two-day Agile training course to all new engineers. The Salesforce Agile methodology, also known as the Adaptive Delivery Methodology (ADM), combines various Agile methodologies and frameworks like Scrum and Extreme Programming (XP) practices. Lean principles guide their overall product development, strategy, and decision-making, optimizing the value stream and encouraging self-sufficient, self-organizing teams. Additionally, Salesforce emphasizes respect and transparency, facilitating open communication through their chat product, Chatter, and encouraging employees to work on projects that interest them. They promote organic change and improvements by allowing employees to suggest ideas and gather organizational support, exemplifying a culture that thrives on change. This Agile approach at Salesforce keeps developers engaged and the process continuously evolving.

### Mini Case Studies: Federal Agencies Lead the Way

- Department of Veterans Affairs (VA) - Vets.gov: The VA embarked on an Agile journey to modernize their digital services for veterans. They implemented a user-centered approach, focusing on iterative development and continuous user feedback. Through Agile practices like Scrum and Kanban, they delivered incremental updates to their website, Vets.gov, ensuring that it met the evolving needs of veterans. The VA's Agile approach allowed them to [efficiently respond to user feedback, address issues promptly, and provide comprehensive support and maintenance to ensure a positive user experience.](#)
- General Services Administration (GSA) - 18F: 18F, a part of GSA, is a government digital services team that embraces Agile principles in their software development and support processes. They [prioritize user needs, employ iterative development, and leverage automated testing to ensure high-quality deliverables.](#) By integrating DevOps practices, 18F maintains a smooth transition from development to deployment and ongoing support. Their Agile approach has led to successful projects like the development of login.gov, a secure and user-friendly authentication platform used across various government services.



- Internal Revenue Service (IRS) - Online Services: The IRS embraced Agile development practices in creating and maintaining their online services, such as e-filing for tax returns and the "Where's My Refund?" application. Agile methodologies enabled the IRS to [deliver frequent updates, respond to changing tax laws, and ensure robust security measures](#). By maintaining a dynamic product backlog and fostering close collaboration with stakeholders, the IRS ensured that their online services remained accessible, reliable, and user-friendly.
- U.S. Digital Service (USDS) - Various Projects: The USDS, a team of digital experts from diverse backgrounds, has been instrumental in driving Agile transformation across multiple federal agencies. They have successfully integrated Agile development with full lifecycle support in various projects, such as the modernization of healthcare.gov and the improvement of immigration application processes. The USDS employs Agile practices like Scrum, user research, and usability testing to ensure that their [projects deliver value to citizens and government agencies alike, while maintaining ongoing support and maintenance](#).

## Future Trends

The future of Agile development and lifecycle support holds promising trends, including the integration of AI in Agile project management, which will leverage AI-powered tools for data analysis and decision-making. Additionally, advances in automation will streamline development cycles and support tasks, enhancing efficiency and reliability. This article highlights [how AI can help scale agile and scrum practices](#) to large and complex organizations by using frameworks and tools that coordinate and integrate the work of multiple teams such as the Nexus framework, LeanKit, and JIRA.

Additionally, the integration of security practices through DevSecOps will become more prevalent, and organizations will focus on holistic lifecycle management, considering the entire product lifecycle and its environmental impact. TechSur Solutions sat down with OrangeSlices.ai to outline 5 direct benefits of integrating this methodology in our article, ["Shifting DevSecOps from a Set of Capabilities to a Repeatable Platform: 5 Benefits to Federal Agencies."](#)

The transformation journey to align Agile practices with full lifecycle support, especially in a vast and complex organization like the USCG, demands more than a mere adoption of Agile processes. **It requires a shift in mindset, commitment to collaboration, and a focus on delivering genuine value.** The challenges, although significant, can be surmounted with the right strategy and approach.

TechSur Solutions, leveraging its Federal digital transformation expertise, offers a roadmap that doesn't just consider Agile in isolation but weaves it in the tapestry of the USCG's existing infrastructure and objectives.

Our aim?  
Seamless integration and optimized value delivery.

