



Office of the Inspector General  
SOCIAL SECURITY ADMINISTRATION

*Audit Report*

Development of the Preliminary  
Claims System

*A-14-20-50912 August 2022*



# Office of the Inspector General

SOCIAL SECURITY ADMINISTRATION

## MEMORANDUM

**Date:** August 30, 2022

**Refer to:** A-14-20-50912

**To:** Kilolo Kijakazi  
Acting Commissioner

**From:** Gail S. Ennis, *Gail S. Ennis*  
Inspector General

**Subject:** Development of the Preliminary Claims System

The attached final report presents the results of the Office of Audit's review. The objective was to determine whether the Social Security Administration complied with Agile software development methods and industry best practices when it developed the Preliminary Claims System.

If you wish to discuss the final report, please contact Michelle L. Anderson, Assistant Inspector General for Audit.

Attachment

# Development of the Preliminary Claims System A-14-20-50912



August 2022

Office of Audit Report Summary

## Objective

To determine whether the Social Security Administration (SSA) complied with Agile software development methods and industry best practices when it developed the Preliminary Claims System (PCS).

## Background

In 2017, SSA began planning the development of the Consolidated Claims Experience product, which the Agency intends will become the single-entry point for employees to process all claims for the Agency.

In March 2018, SSA began its PCS project, the goal of which was to help employees uniformly evaluate the benefits an inquiring individual may be eligible to receive. PCS would share the information collected with the claims systems to eliminate duplicate keying and save time.

In July 2019, SSA delivered a Beta version of PCS to 8 field offices and, in November 2019, delivered it to an additional 40 field offices.

SSA developed PCS using Agile methods, which is an iterative, incremental approach to software development characterized by frequent releases developed in collaboration with customers.

## Results

SSA complied with most of the Agile software development methods and many industry best practices but could have made improvements while it was developing PCS. Team members were new to Agile development and were not familiar with the Agile tool used to manage the development of PCS. As a result, SSA was not able to accurately determine or predict the schedule or progress for PCS. The PCS Beta release was delayed, with reduced scope and increased cost.

In September 2020, SSA stopped developing PCS and began integrating its programming and functionality into a larger effort aimed at replacing all aspects of the claims process.

We have not included formal recommendations in this report; however, we have included suggestions to help SSA improve the management of future Agile projects, update its policies for Agile development, and ensure development teams follow the Agency's policies and guidance. We included recommendations, to be considered together with the suggestions from this report, for the Agency to improve and mature its use of the Agile methodology in another audit, *Agile Software Development at the Social Security Administration (A-14-20-50497)*, which we issued in August 2022.

SSA agreed to consider our suggestions as it continues implementing a scaled Agile framework based on industry best practices.

## TABLE OF CONTENTS

|   |     |
|---|-----|
| Objective.....                                      | 1   |
| Background.....                                     | 1   |
| Results of Review .....                             | 3   |
| Implementing Agile Frameworks and Methods .....     | 4   |
| Scaled Agile Frameworks .....                       | 4   |
| Consistent Sprint Length .....                      | 5   |
| Sustainable Development Pace.....                   | 6   |
| Retrospectives .....                                | 7   |
| Human Resources Management and Staffing .....       | 7   |
| Cross-functional Teams.....                         | 8   |
| Team Facilitators.....                              | 9   |
| Forecasting, Scheduling, and Planning .....         | 9   |
| Release Level Metrics and Tracking.....             | 10  |
| Risk Management .....                               | 11  |
| Quality Controls.....                               | 13  |
| Peer Review .....                                   | 13  |
| Automated and Integrated Testing.....               | 14  |
| Conclusion .....                                    | 14  |
| Agency Comments.....                                | 15  |
| Appendix A –Scope and Methodology .....             | A-1 |
| Appendix B –Commonly Used Agile Frameworks .....    | B-1 |
| Appendix C – Consolidated List of Suggestions ..... | C-4 |
| Appendix D – Agency Comments.....                   | D-6 |

## **ABBREVIATIONS**

|      |                                  |
|------|----------------------------------|
| GAO  | Government Accountability Office |
| OIG  | Office of the Inspector General  |
| PCS  | Preliminary Claims System        |
| SAFe | Scaled Agile Framework           |
| SSA  | Social Security Administration   |

## OBJECTIVE

Our objective was to determine whether the Social Security Administration (SSA) complied with its Agile software development methods and industry best practices when it developed the Preliminary Claims System (PCS).

## BACKGROUND

In 2017, SSA began developing the Consolidated Claims Experience product to become the gateway for employees to process all claims, including eligibility screening, initial claims intake and processing, and maintaining post-entitlement/post-eligibility activities. SSA plans for this modern consolidated user interface to decrease training time, streamline the process for the user, and ultimately enable the Agency to phase out several legacy systems.

In March 2018, SSA began its PCS project to automate the screening process for employees to evaluate the benefits an inquiring individual may be eligible to receive. Individuals' answers, combined with SSA data, would help employees investigate eligibility for Social Security Retirement and Disability benefits, Supplemental Security Income payments, and Medicare benefits. PCS would share the information collected with the claims systems to eliminate duplicate keying and save time.<sup>1</sup>

In July 2019, SSA delivered a Beta<sup>2</sup> version of PCS to 8 field offices and, in November 2019, delivered the Beta version to an additional 40 field offices.<sup>3</sup> With this version, employees were able to conduct pre-claim interviews with individuals and identify potential benefits they may be entitled to receive based on their own earnings history.<sup>4</sup> After feedback from Beta users, SSA discontinued PCS Beta testing in March 2020 and, in September 2020, began integrating PCS programming and functionality into the Consolidated Claims Experience product.

As of August 2022, SSA was still developing the Consolidated Claims Experience product. The Agency's development roadmap showed planned releases through the end of Fiscal Year 2027 but did not provide an estimated completion date for the project.

---

<sup>1</sup> This includes personal information, such as name, date of birth, and residence address; questions on disability and work activity; and earnings history.

<sup>2</sup> Beta refers to software that is undergoing testing and has not yet been officially released. It is shared with users to try under real conditions to obtain feedback on the software's quality and usability and to identify issues.

<sup>3</sup> End-users tested the Beta configuration running in parallel with the existing system.

<sup>4</sup> Future versions of PCS were under development to identify additional potential benefit entitlements based on the individual's relationships (for example, marriage, child, and parent) for auxiliary or survivor's benefits. SSA, *Emergency Message 19020, Preliminary Claims System Beta Release* (July 26, 2019).

Under the traditional approach to software development, all requirements are established at the start of the program and their values are relatively fixed. The process starts with the development of a plan for all requirements and ends when those requirements have been completed.<sup>5</sup> Conversely, SSA developed PCS using the Agile development method, which is an iterative, incremental approach to software development, characterized by frequent releases developed in collaboration with customers.<sup>6</sup> Additionally, the development team continuously reviews and evaluates the software on its functionality, quality, and customer satisfaction, while working on it in iterations or sprints.<sup>7</sup>

Agile teams create user stories,<sup>8</sup> which are expectations that should satisfy the customers' needs for the system being developed.<sup>9</sup> Teams:

- show user stories as a road map and break them down into story points, which indicate the overall size and complexity of a user story;
- estimate the size of each story point, based on the effort involved in developing the feature, the complexity of developing it, and any risk inherent in it;
- prioritize story points in the backlog, based on the benefit to customers;
- focus on the highest priorities during each sprint; and
- ensure the user stories address customers' business needs and develops criteria to determine whether the requirements are met at the end of the sprint.

Agile methods require a strong commitment to ongoing feedback and refinement of requirements to meet the customers' changing needs, while continually managing the previously defined requirements and user stories. If there is not a strong commitment to ongoing feedback and refinement of requirements and user stories, the delivered software may not meet the customers' needs or address the evolving technical landscape.<sup>10</sup> Additionally, there are repeatable business practices that increase the likelihood a team will succeed when using Agile methods for its software development, including a retrospective meeting. A retrospective is a team meeting that typically occurs at the end of every sprint to review lessons learned and discuss how the team can improve the process and team dynamics.

---

<sup>5</sup> Government Accountability Office (GAO), *Agile Assessment Guide*, GAO-20-590G, pp. 69 and 86 (September 2020).

<sup>6</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 7 (September 2020).

<sup>7</sup> A sprint is a predefined, time-boxed, and recurring period in which working software is created. GAO, *Agile Assessment Guide*, GAO-20-590G, pp. 173 and 177 (September 2020).

<sup>8</sup> A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to customers. Rehkopf, Max, *User Stories with Examples and a Template*, Atlassian Agile Coach, atlassian.com (February 3, 2022).

<sup>9</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, pp. 178 and 179 (September 2020).

<sup>10</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 81 (September 2020).

To assist us in auditing SSA's Agile development of PCS, we contracted with a firm with expertise in Agile. To accomplish our objective, with the assistance of our contractor, we interviewed SSA staff; reviewed project-specific and centralized internal knowledge-sharing pages; examined project management data in the application life-cycle management tool; and reviewed the project source code and mechanisms in place to ensure the quality of code being developed. See Appendix A for additional information about our scope and methodology.

## **RESULTS OF REVIEW**

SSA complied with most of its Agile software development methods and many industry best practices but could have made improvements while developing PCS. The Agile process and technical approach the PCS team used were well-considered and sensible. The Agency developed and managed requirements for PCS and focused on the users' needs to ensure the system was usable and useful. In developing the Agile software, the PCS team followed best practices and sufficiently:

- documented the project's overall objectives and goals including high-level functions;
- captured the overall project scope and baselined and maintained the scope;
- structured the product roadmap appropriately and detailed business and technical items;
- prioritized user understanding, including making efforts to understand the PCS users' needs;
- documented user results and feedback to identify problems, opportunities, and needs;
- produced high-level frameworks and created a single-page goals document; and
- communicated regularly with the beta users.

Team members were new to Agile development and were not familiar with the Agile tool used to manage the development of PCS. As a result, SSA may not have been able to accurately determine or predict the schedule or progress for PCS. The PCS Beta release was delayed, with reduced scope and increased cost. In September 2020, SSA stopped developing PCS and began integrating PCS programming and functionality into a larger effort—the Consolidated Claims Experience—aimed at replacing all aspects of the claims process. We noted opportunities for improvement in the following areas:

- Implementing Agile Frameworks and Methods;
- Retrospectives (or lessons learned) Meetings;
- Human Resources Management and Staffing;
- Forecasting, Scheduling, and Planning;
- Release Level Metrics and Tracking;
- Risk Management; and
- Quality Controls.

# Implementing Agile Frameworks and Methods

When implementing Agile in a Federal environment, all Federal and contractor staff must work together to define the Agile terms and processes that will be used for their program.<sup>11</sup> When selecting a framework, organizations should adopt a deliberative process based on the given program's needs as well as the organization's culture and structure.<sup>12</sup> For PCS, SSA used two frameworks. See Appendix B for a description and structure of these frameworks.

## *Scaled Agile Frameworks*

Scaled Agile frameworks help larger programs with multiple teams deliver in a predictable way and with high quality.<sup>13</sup> According to our contractor, most Agile practitioners agree that effective coordination of work between multiple teams requires a scaling framework or collection of scaling patterns.<sup>14</sup> SSA began the PCS project with few staff but gradually increased the number of full- and part-time team members to 80 because of the complexity of the technical environment.<sup>15</sup> However, SSA did not alter its Agile process practices as the team size grew to include four Agile teams. Although SSA engaged in large Agile projects with multiple teams, the Agency's guidance focuses on single team-level Agile frameworks and does not advise how to scale beyond this level. The Agency should have used a formal scaled framework for PCS because of its size.

SSA's Agile Resource Center emphasized small, team-level Agile methods but had not endorsed a scaled framework for larger teams. It also did not provide related support or training for handling larger-scale efforts. Not adopting a formal scaled framework significantly hampered the PCS team's ability to coordinate dependencies and integrate the software among the teams. Scaling frameworks would have helped the team's structure themselves more appropriately into four or more smaller, cross-functional teams<sup>16</sup> and better coordinate their dependencies.

---

<sup>11</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 10 (September 2020).

<sup>12</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 12 (September 2020).

<sup>13</sup> Scaling is the ability of a system, network, or process to absorb a growing amount of work or its potential to be enlarged to accommodate that growth. If the design or system fails when the amount is increased, it does not scale. GAO, *Agile Assessment Guide*, GAO-20-590G, p. 51 (September 2020).

<sup>14</sup> A pattern is "A repeatably applicable solution to a problem that arises in a specific context." Mohammed Rowther, *Scaling Patterns*, Efficient Agile, efficientagile.com (February 3, 2022).

<sup>15</sup> The PCS team consisted of 61 staff and 17 subject matter experts, across four Agile teams.

<sup>16</sup> A cross-functional team includes individuals who have a mix of skills and ability to define, build, and test ideas into a working product. GAO, *Agile Assessment Guide*, GAO-20-590G, p. 172 (September 2020).

We suggest SSA:

1. Update the Agile Resource Center to include guidance on scaled Agile methods. The guidance should recommend that SSA implement a scaled Agile framework when projects warrant (for example, if there are more than 4 teams or 40 staff members).<sup>17</sup>
2. Train all team members in a scaled Agile method.

## ***Consistent Sprint Length***

The Agile Manifesto includes frequent delivery of working software as one of its principles.<sup>18</sup> According to GAO's *Agile Assessment Guide*, "Information obtained during these frequent [sprints] can effectively assist in measuring progress and allowing developers to respond quickly to feedback from customers, thus reducing technical and programmatic risk."<sup>19</sup>

SSA's Agile project schedule recommends sprints be a fixed length of 1 month or less. Although most of the PCS sprints were 2 weeks, the PCS project included two, 7-week sprints. One clean-up sprint began on June 5, 2019 and ended on July 26, 2019, and the maintenance sprint began during the clean-up sprint on June 26, 2019 and ended August 16, 2019. The Agency then resumed 2-week sprints until the end of the project. Generally, sprints for PCS did not overlap. This was the exception.

Each Agile team should have someone who facilitates sprint planning, ensures Agile practices are followed (such as maintaining consistent sprint length), and works with the Project Manager to ensure sprint planning and execution aligns to committed schedule and deliverables. A contributing factor to the inconsistent sprint length could have been the growing list of necessary work that was delayed to meet a deliverable or deadline.<sup>20</sup> The Agency accumulated this backlog because it developed PCS at an unsustainable pace, requiring the team to stop and stabilize for the two, 7-week periods, which overlapped and resulted in a total of 11 weeks.

Inconsistent sprint lengths make forecasting project completion and tracking progress difficult and reduce the effectiveness of Agile metrics, including those that measure the amount of work a team can deliver each sprint and progress in completing the tasks. This makes the project more difficult to plan and manage, as demonstrated when the PCS Beta release was delayed and later released with reduced scope and increased cost.

We suggest SSA sprints lengths be consistent with Agency policy.

---

<sup>17</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, pp. 11 and 12 (September 2020).

<sup>18</sup> Agile Manifesto, *Principles Behind the Agile Manifesto*, agilemanifesto.org (February 3, 2022).

<sup>19</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 7 (September 2020).

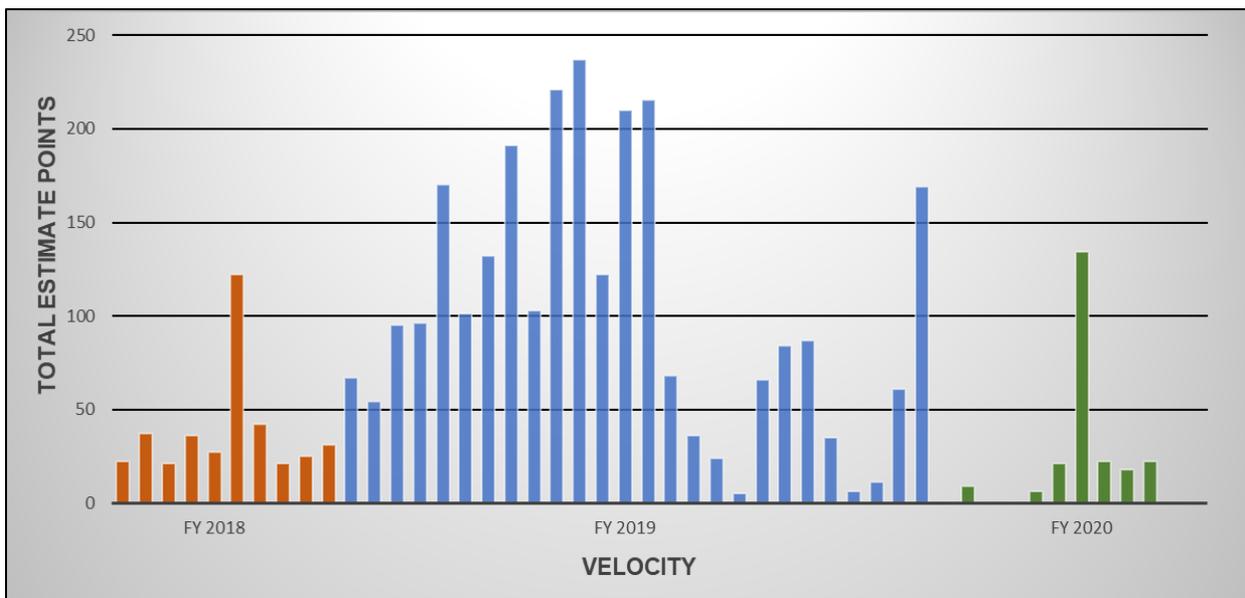
<sup>20</sup> The accumulation of delayed work can present obstacles to an Agile program if not properly managed. For example, as code base grows, additional functions will rely on the deficient code, reducing overall system performance. As the backlog of delayed work continues to rise, teams will devote more time to cleaning up errors instead of producing new features. GAO, *Agile Assessment Guide*, GAO-20-590G, p. 187 (September 2020).

## Sustainable Development Pace

According to GAO, management should strive to ensure teams can maintain a sustainable development pace by prioritizing user stories, establishing an agreed-upon definition of “done” for those user stories, and reaching a mutual commitment on the work to be accomplished for each sprint. Management should emphasize, and encourage teams to, maintain a consistent development pace that can be sustained indefinitely.<sup>21</sup>

We noted inconsistencies in the pace of development for PCS, with some sprints delivering little value (for example, having a small amount of story points completed) and others delivering more. As shown in Figure 1, team output did not demonstrate a sustainable delivery pace.

**Figure 1: Story Points Completed per Sprint**



Source: SSA VersionOne

The PCS teams may have had difficulty maintaining a sustainable development pace because tasks that involve unfamiliar technologies are difficult to estimate, and some tasks may have been more complex than anticipated. If teams are not working at a sustainable pace, there is a risk of burnout, which can cause delays in the program. In addition, by not working at a sustainable pace, management does not have reliable historical data to estimate future efforts required in product development, as evidenced by the PCS Beta release delay.

We suggest SSA ensure teams balance the size of the user stories relative to one another, assess the complexity of work based on input from the product owner, refine user stories and estimates over time, and use prior estimates to inform future estimates.

<sup>21</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, pp. 50 and 51 (September 2020).

## Retrospectives

Retrospective (or lessons learned) meetings are an integral part of Agile planning, process, and product improvement. A principle of Agile is that “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”<sup>22</sup> Also, according to GAO, teams should hold a retrospective meeting at the end of each sprint to reflect on what went well and what could be improved for the next sprint.<sup>23</sup>

Although improvements may be implemented at any time, the retrospective provides a formal opportunity to focus on inspection and adaptation. Per SSA’s *Agile Team Roles and Responsibilities*, part of the Product Owner’s responsibilities is to work with the team facilitator to resolve issues discussed during the retrospective. By the end of the retrospective, the PCS Team should have identified improvements the team would implement in the next sprint.

In 18 of the 32 sprints, the PCS team did not document retrospective meeting summaries in SSA’s repository. In the remaining 14 sprints, the team properly documented 10 retrospective meetings but included 4 blank meeting summaries. According to SSA, some retrospectives were documented on notepads in the team’s physical workspace, and teams began documenting them in SSA’s repository as the teams became more familiar with its features and capabilities. Without documented retrospectives, future programs cannot benefit from the lessons learned.

We suggest SSA:

1. Remind Agile team members of the importance of properly documenting retrospectives.
2. Ensure the team facilitator and Product Owner are discussing results from retrospectives and implementing lessons learned after each sprint.

## Human Resources Management and Staffing

According to GAO, all team members and staff who will be developing software, supporting development activities, or involved in the acquisition process using Agile should possess the competencies, skills, knowledge, and process abilities needed to perform their roles. In addition, team members who use Agile methods should be trained in the specific Agile method they will be using. This training should include the Agile policy and procedures documented by the organization. Without training, there may be a lack of common understanding about the Agile methods to be used.

---

<sup>22</sup> Agile Manifesto, *Manifesto for Agile Software Development*, agilemanifesto.org (February 3, 2022).

<sup>23</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 44 (September 2020).

If management pays proper attention to staffing and human resources management, projects can be successful even if they are lacking in other areas. However, a well-planned project can be derailed if it does not have competent staff or has excessive turnover. The critical elements for staffing and human resources management controls are: (1) ensure project teams possess relevant skills, (2) implement effective training controls, and (3) implement effective retention controls.

## ***Cross-functional Teams***

When teams are structured in architectural silos (such as data access or computations) rather than feature teams,<sup>24</sup> batch hand-off to downstream teams may cause delays because those teams must catch up. This may have forced some PCS teams to stop new work until the backlog of in-progress work was cleared. The team may have also found it difficult to maintain a sustainable development pace because of its structure.

An Agile team should be structured to allow for its own autonomy so it need not rely on outside teams to complete its work. Collectively, the team should have the skills necessary to perform the work and represent the various sections of the organization that touch on software development, such as business subject matter expertise, quality assurance, and cyber-security.<sup>25</sup> If a team does not have all the requisite skill sets, it must rely on other teams that may have other responsibilities, which could delay the project.

According to SSA, there were four PCS component teams: two user interface teams, one data modeling team, and one computations team. These teams were composed of analysts, testers, developers, a product owner, and a facilitator. The PCS teams were not cross-functionally structured and organized for architectural silos, each of which was needed to deliver working software to an end-user. No team could independently deliver a user-friendly, fully functional product without reliance on other teams. As a result, SSA was unable to scale its efforts. Because of scheduling dependencies, the Agency was unable to forecast and plan because of the unknowable delays in coordinating work between teams. Dependencies between teams may have forced some PCS teams to stop new work until the backlog of in-progress work was cleared and contributed to the PCS Beta delay and later release with reduced scope and increased cost.

Cross-functional teams would have given SSA all the skills needed to deliver a piece of working software, and the teams could have seen the size of the entire project and developed more reliable estimates.

We suggest SSA structure teams to have cross-functional skills that allow for autonomy, so each team does not need to rely on outside teams to complete its work.

---

<sup>24</sup> Team members should have cross-functional skills that allow them to perform all the work rather than a single specialty.

<sup>25</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 30 (September 2020).

## Forecasting, Scheduling, and Planning

GAO's Guide describes the 10 best scheduling practices for developing and maintaining a reliable, high-quality schedule.<sup>26</sup> The Guide also presents principles for auditors to evaluate certain aspects of Federal programs. An Agile program with a fully developed and executed schedule guides the team to focus on deadlines for specific goals and activities to ensure all required actions are planned to be completed.

According to GAO, while the Agile software development philosophy is different from that of traditional development methods, all Federal programs need a high-quality program schedule to be accountable for delivering a value-based outcome. Agile methods provide useful progress indicators to inform management about the status of high-priority features. Many measures used to manage Agile development programs can demonstrate the program is meeting GAO's scheduling best practices and can aid in assessing the program offices' planning for developing their schedules.

While Agile emphasizes that only near-term work is planned in detail (such as the next sprint), programs need to define their overall goal in a vision and plan the future releases needed to satisfy the vision. The detailed plan is subject to change, but the vision provides a high-level view and direction for the work to be accomplished for the entire program. Additionally, while the team organizes its own work, it must remain aware of dependencies with other teams, related Agile and non-Agile programs, and equipment.

GAO describes program monitoring and control, which is the ability to generate reliable estimates, as a critical program management function.<sup>27</sup> Typical estimates include: cost and schedule estimates that are updated throughout the program's life cycle; forecasts of costs at completion for work in progress; and plans to establish an Agile work breakdown structure to identify discrete features that can be monitored. Management and Agile teams can use a work breakdown structure or similar document to provide a clear picture of the total scope of work necessary to meet a program's vision and requirements.

SSA did not create a work breakdown structure or similar document for PCS early enough to establish a baseline cost and schedule estimate to deliver a minimum viable product.<sup>28</sup> Although a minimum viable product release and goal was written, the team did not translate this to a well-defined, high-level backlog that made forecasting possible. Without any way to forecast how much longer the project would take, leaders would not be able to make an informed decision whether to proceed (for example, not proceeding when 50 percent of the originally budgeted time or money was spent, with no end in sight). Because the Agency did not have a well-defined work breakdown structure, it:

- had to delay the Beta release, reduce the scope of the minimum viable product, and increase the cost as it had to add two development teams to meet the release timeframe;

---

<sup>26</sup> GAO, *Schedule Assessment Guide*, GAO-16-89G (December 2015).

<sup>27</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 66 (September 2020).

<sup>28</sup> A minimum viable product should have enough value that it is still usable, demonstrates future benefit early on to retain customer buy in, and provides a feedback loop to help guide future development. It is the simplest version of a product that can be released. GAO, *Agile Assessment Guide*, GAO-20-590G, p. 174 (September 2020).

- was unable to forecast milestones; and
- used a significant proportion of the time and money allocated because of these delays in discovering implications of the project's true size and scope.

We suggest SSA:

1. Train Agile teams on estimating size and team capacity requirements to complete a project.
2. Ensure teams track progress against forecasts at a high level.
3. Leverage tools and practices for generating forecasts that already exist in SSA.  
For example, the Agile Resource Center Agile Project Schedule Guidance provides an Agile Schedule Template in Microsoft Project that leverages that tool's ability to calculate and display overall percent complete and compare that to schedule completion percentage.
4. Follow the recommendations for forecasting and tracking progress in GAO's *Agile Assessment Guide*.

## Release Level Metrics and Tracking

GAO emphasizes the work Agile teams accomplish should be tracked and monitored to help inform and measure organization operations and results.<sup>29</sup> Performance information can be measured at various stages of software development and at different levels of an organization. Such information can be used to identify problems and take corrective actions, develop strategies, allocate resources, and identify and share effective approaches. Accordingly, regardless of their preferred Agile development framework, organizations and programs should establish an appropriate set of metrics and associated processes to measure their performance goals early in the development cycle.<sup>30</sup>

The developers should use relative estimation, which compares the current work with work of similar size and complexity, which enables teams to maintain a sustainable software development pace and predict work commitments. Management needs to be able to generate reliable cost and schedule estimates and update them throughout the project, forecast costs at completion for work in progress, and use an Agile work breakdown structure to identify discrete features that can be monitored. Agencies can use charts to track progress and measure productivity, costs, schedule, and performance.

The PCS project repository included about 1,000 user stories, but more than 300 did not include any story-point estimates, and more than 100 of these were part of a sprint. In addition, of the 1,700 tasks defined in the PCS project, nearly 700 did not have a detailed estimate in hours.

---

<sup>29</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, pp. 38 and 150 (September 2020).

<sup>30</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 150 (September 2020).

We identified many items that were carried into the next sprint, but we could not determine whether items were unfinished or finished but not updated in the Agency's project repository. The teams carried over split user stories, which skewed the data on the teams' performance. There is no best practice for splitting unfinished user stories, but the team needs to split the stories so the team can appropriately refine its planning. Additionally, the teams added new items after sprint planning to nearly every sprint so no sprint achieved its goal. The number of points the team committed to completing, and the number of points actually achieved, fluctuated for PCS.

According to SSA, the team continually evaluated backlog items for the highest priority stories to be completed, and the team facilitator emphasized the Agile best practice of prioritizing and ordering the backlog, recognizing that not all the user stories were expected to be completed because of the schedule constraints. However, there was no documentation to indicate the PCS team regularly reviewed important metrics, such as team progress charts.

Based on the data available, it is possible the PCS team did not add all items to the sprint backlog before it started the sprint, and it may not have closed out all items by the end of the sprint. Without complete story point and task hour estimates, the team had no way to measure its progress. A lack of traceability can lead to overlooking stories that are critical to the program.

We suggest SSA:

1. Review team progress charts at least once a day and immediately investigate any anomalies.
2. Analyze release-level metrics, in addition to the sprint-based metrics, and compare the metrics regularly to the high-level roadmaps to enable proper overall governance and oversight.

## Risk Management

Agile best practices dictate that teams should ensure project risks are systematically reduced by addressing the riskiest elements first to ensure risks do not recur and reduce the possibility of latent items surfacing as project risks. Typical project risks include not having appropriately skilled and knowledgeable personnel, not understanding the scope and requirements, or not being able to achieve scalability and usability goals.

According to SSA's Project Resource Guide and Risk Management Plan, the Project Manager will use the risk register<sup>31</sup> developed to identify, track, assess, and monitor risks associated with the project. The Project Manager periodically reviews the risk register, and updates it as appropriate to: identify any new risks; monitor the status of the predicted risks; and add or modify risk mitigation tasks. Significant or higher risks should be reviewed at least monthly, and the entire risk registry should be reviewed at least quarterly.

---

<sup>31</sup> A central record of current risks, and related information, for a given scope or organization. Current risks comprise both accepted risks and risks that have a planned mitigation path. National Institute of Standards and Technology, *Computer Security Resource Center Glossary*, nist.gov, (May 20, 2022).

The Agency documented and managed a risk register for PCS; however, risks captured for the project were outwardly focused and did not include risks related to Agile processes, team-level concerns, or inter-team coordination. Specifically, the PCS risk register focused on potential effects to the larger organization (such as a Government shutdown, unfamiliar technologies, or the feasibility of the project scope exceeding team capacity), but there was no documentation of team participation in risk-related discussions. Because the risk-management process did not delve into Agile process and team-level concerns, the risk register did not capture all issues, such as coordination between teams, breaking up larger releases into smaller pieces, or automating testing.

According to SSA, for PCS, the dependencies and concerns related specifically to Agile were discussed and addressed during daily stand-up and other meetings with guidance from the team facilitator and Agile coach. SSA had a risk management workgroup that was exploring a standard process for how risks are addressed. Identifying Agile team risks and recording them on the project's risk register provided transparency so risks that affect the project's scope, schedule, and/or cost can be addressed and allows for collaborative team planning.

We suggest SSA update the Risk Management Plan to:

1. emphasize the importance of active risk management and soliciting risks from team members who are closest to the work;
2. include scheduling a risk brainstorming meeting with all project team(s) at the beginning of the project;
3. assign a compliance group, such as Agile Resource Center or Project Governance and Quality Assurance, to review the risk lists to ensure they are being managed and updated; and
4. require that all project team(s) review current risks at the weekly project meetings and document notes in the appropriate tool, such as a risk register, that updates the overall project risk list.

## Quality Controls

Quality control processes help reduce the number and severity of latent errors in systems. According to GAO, adherence to coding standards and using automated and manual testing are necessary to improving the quality of code that is ultimately inserted into the continuous integration build process.<sup>32</sup> Software with a large number of defects or an inefficient structure affects system performance and requires that developers spend critical time and effort repairing defects.<sup>33</sup> Additionally, quality should be tracked and monitored based on established expectations. Finally, although development focuses on functional needs, the project strategy must also include nonfunctional requirements,<sup>34</sup> such as security and privacy.

### *Peer Review*

One quality assurance method is having a peer review, where a team member who did not develop the code reviews portions of the code base to assess its quality and adherence to defined coding standards.<sup>35</sup> Per SSA's procedures, a peer review is not complete until a peer review worksheet or an electronic equivalent is completed and distributed.<sup>36</sup>

SSA used an indirect peer review process for PCS. There were no formal meetings set up for a code review used during the indirect peer review, but there were ad hoc discussions based on comments from the reviewers. The Agency stated the process provided an indirect peer review and a resolution to any outstanding issues related to application development. However, there was no documentation to indicate this process was conducted.

Without documenting the peer review process for PCS, the Agency missed important opportunities to improve skills and shorten learning curves. A robust peer review process helps spread the knowledge from the teams structured in architectural silos to the other team members.

We suggest SSA:

1. Emphasize the importance of quality and reinforce that peer reviews are necessary and should be documented.
2. Spot check peer reviews during retrospectives and ensure they are properly approved.
3. Show teams examples of coding changes analyzed throughout the peer review process that were improved through multiple rounds of review.

---

<sup>32</sup> Continuous integration is when developers regularly merge their code changes into a central repository, after which automated builds and tests are run.

<sup>33</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 41 (September 2020).

<sup>34</sup> Non-functional requirements generally specify criteria that can be used to judge the operation of a system rather than specific behaviors. Typical nonfunctional requirements are reliability, scalability, maintainability, availability, quality, privacy, and security. GAO, *Agile Assessment Guide*, GAO-20-590G, p. 50 (September 2020).

<sup>35</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, p. 42 (September 2020).

<sup>36</sup> SSA, Office of Systems Procedure for Conducting Peer Reviews, Version: 11.9, p. 4 (October 2020).

## ***Automated and Integrated Testing***

Each stage of the continuous integration process should include automated tests of functional and non-functional requirements. Automation of repeatable processes allows software components that are added or modified to be continuously integrated into the system.

SSA's development environment included mechanisms for testing but did not offer a set of automated testing frameworks. PCS did not have automated unit and integrated testing; however, it is possible the team was using an older, manual approach to deployment that was not conducive to these types of test automation. Lack of automated stress, performance, and other testing increases the burden on manual testers and leads to larger teams and slower cycle times. Additionally, teams that overlook non-functional requirements may develop a system that does not comply with current Federal standards (such as for cyber-security). This could create unnecessary risks to business operations and delay implementing the software until these components have been addressed.

We suggest SSA:

1. Determine an automated scalability and performance testing strategy early in the project.
2. Include automated unit and integrated testing in the continuous integration and continuous development pipeline.
3. Identify ways to create separate parallel pipeline steps for testing so the team can get the benefits without having lengthy builds (for example, by running full performance tests daily).

## **CONCLUSION**

SSA delivered and tested a Beta version of PCS while it maintained the existing production system. However, at the time of this audit, the Agency had stopped developing PCS and moved the project into a larger effort—the Consolidated Claims Experience—aimed at replacing all aspects of the claims process. We have not included formal recommendations in this report; however, we have included suggestions to help SSA improve the management of future Agile projects, update its policies for Agile development, and ensure development teams follow the Agency's policies and guidance. Refer to Appendix C for a consolidated list of suggestions. We plan to include recommendations for the Agency, to be considered together with the suggestions from this report, to improve and mature its use of the Agile methodology in another audit, *Agile Software Development at the Social Security Administration (A-14-20-50947)*.

## **AGENCY COMMENTS**

SSA agreed to consider our suggestions as it continues implementing a scaled Agile framework based on industry best practices (see Appendix D).

A handwritten signature in black ink that reads "Michelle L. Anderson". The signature is written in a cursive, slightly slanted style.

Michelle L. Anderson  
Assistant Inspector General for Audit

# ***APPENDICES***

## Appendix A –SCOPE AND METHODOLOGY

---

To accomplish our objective, we:

- Reviewed applicable Federal laws, regulations, and guidance related to use of Agile software development, including the following:
  - FY 2021 IT Budget – Capital Planning Guidance;
  - Government Accountability Office (GAO) Agile Assessment Guide, GAO-20-590G, *Best Practices for Agile Adoption and Implementation*, September 2020; and
  - Office of Management and Budget Circular No. A-130, *Managing Information as a Strategic Source*, July 2016.
- Reviewed the Social Security Administration’s (SSA) policies and procedures pertaining to Agile Software development, including the Office of Systems’ *Project Management Guidebook*, August 2019.
- Interviewed or contacted SSA staff from components related to the use of Agile software development, including the Office of Systems’ Office of Associate Commissioner.
- Examined project entry in the centralized investment management tool, project specific and centralized SSA SharePoint documents, project-specific and centralized internal wiki pages, and information from project tools. Reviewed the project source code, build scripts and continuous integration and continuous delivery automation.

We conducted our audit between September 2020 and December 2021. The principal entity reviewed was the Office of the Deputy Commissioner for Systems.

We assessed the significance of internal controls necessary to satisfy the audit objective. This included an assessment of the five internal control components, including control environment, risk assessment, control activities, information and communication, and monitoring. In addition, we reviewed the principles of internal controls associated with the audit objective. We identified the following four components and five principles as significant to the audit objective.

- Component 2: Risk Assessments
  - Principle 6: Define objectives and risk tolerances
- Component 3: Control Activities
  - Principle 12: Implement control activities
- Component 4: Information and Communication
  - Principle 14: Communicate internally
- Component 5: Monitoring
  - Principle 16: Perform monitoring activities
  - Principle 17: Remediate deficiencies

We conducted this performance audit in accordance with generally accepted government auditing standards. Those standards require that we plan and perform the audit to obtain sufficient, appropriate evidence to provide a reasonable basis for findings and conclusions based on our audit objective. We believe that the evidence obtained provides a reasonable basis for our findings and conclusions based on our audit objective.

## **Appendix B –COMMONLY USED AGILE FRAMEWORKS**

---

There are numerous frameworks<sup>1</sup> available for Agile programs. Commonly used Agile frameworks include Kanban, Scrum, Scaled Agile Framework, and Large-Scale Scrum. Kanban and Scrum are individual team frameworks, while Scaled Agile Framework and Large Scale Scrum are scaled frameworks intended to increase Agile processes so they can be applied to large, complex organization structures.<sup>2</sup> The Social Security Administration (SSA) used Kanban and Scrum to develop its Preliminary Claims System.

### ***Kanban***

Kanban seeks to alleviate the bottlenecks in Waterfall<sup>3</sup> development by limiting in-progress work to design and deliver products to customers efficiently and effectively and to prevent a team from committing to too much work. Since new work should not be started until the current work has been completed, bottlenecks blocking the completion of work should become more visible in the process. This framework focuses on the flow of work and was inspired by lean manufacturing.

There are no prescribed roles in Kanban, which allows for maximum team flexibility so members can work on each other's artifacts easily. Teams use a Kanban board to track their work, which can be either physical or virtual. A Kanban board maintains a clear, visual representation of the work through various stages of development.

### ***Scrum***

Scrum seeks to address complex problems while delivering high-value products frequently and effectively, in increments often referred to as sprints. Each sprint is a short, time-boxed iteration that is intended to provide distinct, consistent, and incremental progress of prioritized software features.

The Scrum framework is centered on Scrum teams where members fill specific roles and responsibilities. These members are responsible for various tasks, including developing Agile artifacts. Each team contains members that fit into one of three main roles: product owner, development team, or team facilitator.

With Scrum, teams are self-organizing and choose how best to accomplish their work, rather than being directed by management. Teams are also cross-functional, meaning they include members who have the capabilities to achieve the work without depending on someone outside the team.

---

<sup>1</sup> A framework is a basic structure that guides customers, rather than a prescriptive process. GAO, *Agile Assessment Guide*, GAO-20-590G, p. 10 (September 2020).

<sup>2</sup> GAO, *Agile Assessment Guide*, GAO-20-590G, pp. 11 and 12 (September 2020).

<sup>3</sup> Waterfall development sets an expectation that all requirements are established at the start of the program and their value is relatively fixed. It starts by developing a plan for all requirements and ends when those requirements have been completed. GAO, *Agile Assessment Guide*, GAO-20-590G, pp. 69 and 86 (September 2020).

During sprint planning meetings, the team determines the type of work to be done, prepares an ordered list of tasks to be accomplished during the sprint, and communicates expected responsibilities between team members. Teams meet daily during each sprint for a brief status update. Each sprint is intended to produce, among other things, completed increments of software features that are ultimately built into the final product solution.

A team progress chart is a public display of the remaining work in the sprint backlog. The team updates the chart daily to keep everyone informed of the status of tasks.

## ***Scaled Agile Framework***

Scaled Agile Framework (SAFe) is a governance model used to align and collaborate product delivery for modest-to-large numbers of Agile software development teams. The Framework provides guidance for roles, inputs, and processes for teams, programs, large solutions, and portfolios. It is also intended to provide a scalable and flexible governance framework that defines roles, artifacts, and processes for Agile software development across all levels of an organization. There are 10 principles in SAFe:

1. take an economic view;
2. apply systems thinking;
3. assume variability;
4. build incrementally in cycles;
5. base milestones on evaluation of working systems;
6. visualize and limit work in progress;
7. apply cadence;
8. unlock motivation of workers;
9. decentralize decision making; and
10. organize around value.

There are four different configurations of SAFe: essential, large solution, portfolio, and full. These configurations allow for different scales of teams to adopt SAFe, depending on the size and complexity of the product being developed. These levels allow teams to perform iterative processes using Agile frameworks to develop features to be used by a larger program. SAFe uses many of the same tools as other Agile methods, such as backlogs, development teams, and time-constrained iterations.

Depending on the scale, the Framework is divided into different levels, each with its own responsibilities and processes that connect the different levels. Development teams in SAFe align with the selected Framework and are advised to embrace the traditional cross-functional team mentality. At the program level, these Agile teams come together to create a release train that reflects specific roles and responsibilities.

## ***Large Scale Scrum***

Large Scale Scrum is a scaled-up version of one-team Scrum but involved multiple teams. It maintains many of the practices and ideas of one-team Scrum, including:

- a single prioritized backlog;
- one definition of done for all teams;
- one product owner; and
- many complete, cross-functional teams with no single specialist teams.

In Large Scale Scrum, all teams are in a common iteration to deliver a common, shippable product.

## Appendix C – CONSOLIDATED LIST OF SUGGESTIONS

---

The Social Security Administration (SSA) delivered and tested a Beta version of the Preliminary Claims System (PCS) while it maintained the existing production system. However, at the time of this audit, the Agency had stopped developing PCS and moved the project into a larger effort—the Consolidated Claims Experience—aimed at replacing all aspects of the claims process. We are not including formal recommendations in this report; however, we have included suggestions to help SSA improve the management of future Agile projects, update its policies for Agile development, and ensure development teams follow the Agency’s policies and guidance. We suggest SSA:

- Update the Agile Resource Center to include guidance on scaled Agile methods. The guidance should recommend that SSA implement a scaled Agile framework when projects warrant (for example, if there are more than 4 teams or 40 staff).
- Train all team members in a scaled Agile method.
- Ensure sprints lengths are consistent with Agency policy.
- Ensure teams balance the size of the user stories relative to one another, assess the complexity of work based on input from the product owner, refine user stories and estimates over time, and use prior estimates to inform future estimates.
- Remind Agile team members of the importance of properly documenting retrospectives.
- Ensure the team facilitator and Product Owner are discussing results from retrospectives and implementing lessons learned after each sprint.
- Structure teams to have cross-functional skills that allow for autonomy, so each team does not need to rely on outside teams to complete its work.
- Train Agile teams on estimating size and team capacity requirements to complete a project.
- Ensure teams track progress against forecasts at a high level.
- Leverage tools and practices for generating forecasts that already exist in SSA. For example, the Agile Resource Center Agile Project Schedule Guidance provides an Agile Schedule Template in Microsoft Project that leverages that tool’s ability to calculate and display overall percent complete and compare that to schedule completion percentage.
- Follow the recommendations for forecasting and tracking progress in the Government Accountability Office’s Agile Assessment Guide.
- Review team progress charts at least once per day and immediately investigate any anomalies
- Analyze release-level metrics, in addition to the sprint-based metrics, and compare the metrics regularly to the high-level roadmaps to enable proper overall governance and oversight.

- Update the Risk Management Plan to:
  - emphasize the importance of active risk management and soliciting risks from team members who are closest to the work;
  - include scheduling a risk brainstorming meeting with all project team(s) at the beginning of the project;
  - assign a compliance group, such as Agile Resource Center or Project Governance and Quality Assurance, to review the risk lists to ensure they are being managed and updated; and
  - require that all project team(s) review current risks at the weekly project meetings and document notes in the appropriate tool, such as a risk register, that updates the overall project risk list.
- Emphasize the importance of quality and reinforce that peer reviews are necessary and should be documented.
- Spot check peer reviews during retrospectives and ensure they are properly approved.
- Show teams examples of coding changes analyzed throughout the peer review process that were improved through multiple rounds of review.
- Determine an automated scalability and performance testing strategy early in the project.
- Include automated unit and integrated testing in the continuous integration and continuous development pipeline.
- Identify ways to create separate parallel pipeline steps for testing so the team can get the benefits without having lengthy builds (for example, by running full performance tests daily).

## Appendix D – AGENCY COMMENTS

---



### SOCIAL SECURITY

#### MEMORANDUM

Date: August 11, 2022

Refer To: TQA-1

To: Gail S. Ennis  
Inspector General

A handwritten signature in blue ink, appearing to read "Scott Frey".

From: Scott Frey  
Chief of Staff

Subject: Office of the Inspector General Draft Report "Development of the Preliminary Claims System"  
(A-14-20-50912) — INFORMATION

Thank you for the opportunity to review the draft report. We will consider OIG's suggestions as we continue to implement a scaled Agile framework based on industry best practices.

Please let me know if I can be of further assistance. You may direct staff inquiries to Trae Sommer at (410) 965-9102.



**Mission:** The Social Security Office of the Inspector General (OIG) serves the public through independent oversight of SSA's programs and operations.

**Report:** Social Security-related scams and Social Security fraud, waste, abuse, and mismanagement, at [oig.ssa.gov/report](https://oig.ssa.gov/report).

**Connect:** [OIG.SSA.GOV](https://oig.ssa.gov)  
Visit our website to read about our audits, investigations, fraud alerts, news releases, whistleblower protection information, and more.

Follow us on social media via these external links:



Twitter: @TheSSAOIG



Facebook: OIGSSA



YouTube: TheSSAOIG



Subscribe to email updates on our website.