Hybridized Agile Software Development of Flight Control Team Tools for International Space Station's Payload Operations Integration Center

Cerese M. Albers, Ph.D., PMI-ACP[1]
COLSA Corporation, Huntsville, AL, 35806, USA

Ground systems operations at the National Aeronautics and Space Administration's (NASA) Payload Operations and Integration Center (POIC) at Marshall Space Flight Center (MSFC) recently increased via a High Operations Tempo (HOT) initiative, in order to support more science activities with a fourth crew member on the International Space Station (ISS). The Flight Control Team's (FCT) need to support this increasing pace of payload science operations was the impetus for creating a series of new tools. While their need was clear, the full scope and user experience for each tool was not as well-understood, thus establishing a fixed set of initial requirements was not feasible. A hybridized Agile Software Development (ASD) paradigm was created to take advantage of this uncertainty, plan for it, permit the exploration of novel concepts, and also facilitate a rapid and flexible response to inevitably changing requirements. The POIC's hybridized ASD approach places preeminent focus on providing customer value through the delivery of high quality, customer-focused solutions in short timeframes. This has been successfully achieved through creating unprecedented modes of cooperation and collaboration between operations and software development teams, frequent user evaluations of the software with well-defined feedback mechanisms, increased human factors involvement, and a dedication to successful outcomes by the whole of the POIC. Since space science operations and software development are not typically so closely linked, this paper discusses an approach that offers an optimal way to provide an increased return on investment and a faster time-to-completion than traditional software development paradigms, while aiming at delivering high quality products and customer-driven value.

I. Introduction

In mid-2014, ground systems support personnel at the National Aeronautics and Space Administration (NASA), Marshall Space Flight Center's (MSFC) Mission Operations Laboratory (MOL) identified the need to support an anticipated large increase in payload science experimentation time due to the upcoming ability to staff an additional crew member aboard the International Space Station (ISS). The "fourth crew" member provided the opportunity to help NASA have a higher return on investment with the ISS, by supporting many more micro-gravity payload science experiments in near-Earth orbit. Thus, MOL kicked-off a new Fourth Crew Tools effort, also known as the High Operations Tempo (HOT) initiative. Its purpose was to review and mature concepts for operations, Flight Control Team (FCT) structure and organization, processes and procedures to support ground control of commanding and telemetry, and to investigate additional tools and automation capabilities to support a faster cadence of operations. One catch was that the anticipated start of this tempo increase- to around 100 hours of crew-time payload science activities per week- was slated to begin in 2017. This did not leave much time to evaluate if additional positions were necessary and hire additional personnel, overhaul processes, and develop new tools for automation and simplification of information flow for the FCT. This meant that the HOT tools had to be developed quickly and provide a complete and useful solution the first time they were delivered, with little time anticipated to accommodate feedback during operational use before the tools would become highly necessary.

The unique challenge of delivering five of the HOT tools ready for usage by 2017 was ultimately met through the dedication and hard work of many operations and software development personnel willing to embrace a previously unexplored mode of collaboration; hybridized Agile Software Development (ASD). Its success has led to the origination of concepts for additional tools, the innovation of new methods of automation, the ability to support

[1] Software Development Build Coordinator and Agile Consultant, NASA MSFC HP27

an increased operations tempo which is now an ongoing manageable pace, as well as increased job satisfaction for both operators and developers on these projects, and a suite of tools that is rapidly gaining the attention of other NASA operations teams.

SpaceOps brings together experienced professionals in ground systems engineering and data management, and opportunities abound to exchange ideas and lessons learned for how to design, develop, and deploy tools to support science activities in space. This presents a one-of-a-kind venue to share success stories and glean insights from how other partners in the industry are developing tools to support an increasing pace of payload science operations. ASD is an optimal approach that offers higher return on investment and faster time-to-completion than traditional software development paradigms, while aiming at delivering high quality products and customer-driven value.

II. The Challenge

Innovating a new way of involving ground systems operational support personnel in the software development process to achieve greater outcomes and better tools is a formidable but worthy challenge. Doing so in a short amount of time with little room for failure is a challenge that, if successfully met, promises rich rewards. However failure to meet the task with enough support from both sides of a project- operations and software development- could result in a shortcoming in the ability to support a higher operating tempo, a loss of enthusiasm for a new collaborative paradigm, and a missed opportunity to create a new synergy and unity of purpose between the teams from which multiple beneficial initiatives could arise. With so much at stake, NASA MOL recommended that the Payload Operations and Integration Center (POIC) undertake an Agile Project Management (APM) approach to developing the HOT tools. It was hoped that a custom implementation of APM could overcome the obstacles to incorporating FCT personnel and their feedback into the POIC software development process for the first time.

A. Why Not Waterfall Software Development?

Comprehension of several factors is required to understand the necessity of hybridizing an APM approach for the HOT initiative, instead of simply accommodating the need in the existing POIC Waterfall Software Development (WSD) implementation. These were obstacles that presented an imposing challenge to successful collaboration. Some were more difficult to surmount than others. Figure 1 details the top ten temporal, cultural, and process complications that any chosen approach to HOT tool development had to overcome; many are common across space operations-development collaborations.

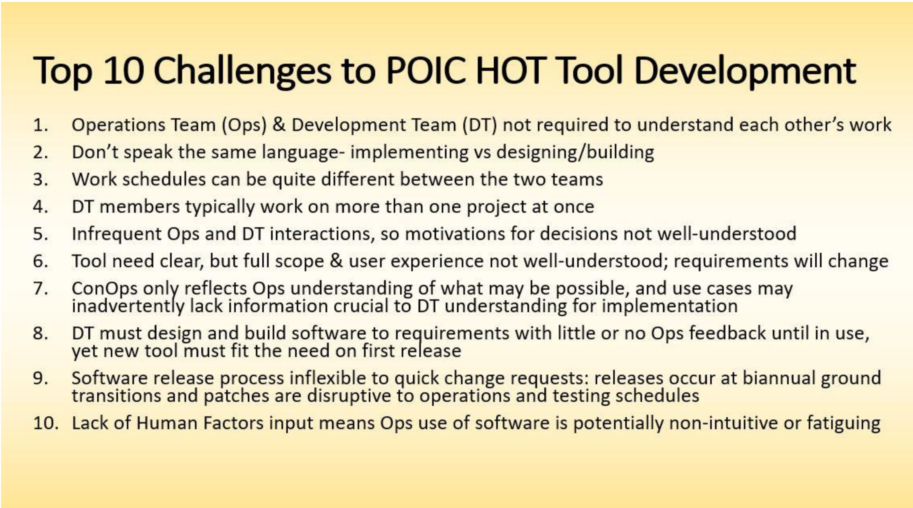

Fig. 1 Top 10 Challenges to POIC HOT Tool Development

At the inception of the HOT initiative, the existing mode of software delivery to operations was predominantly based on a WSD methodology. This was the kind of Plan-Develop-Test-Release process flow that had been in place for years, and which followed a rigid conformance to a plan, with requirements defined and fixed before software development started. While many prescriptive projects have clear, well-understood objectives, there are many others in which it is not possible to know the full scope, or to define the full user experience, at the outset of the project. Such efforts call for the ability to explore and adapt the product to iteratively-discovered refinements of the customer's understanding of requirements and user experience along the way. WSD avoids change mid-project, but there are other methodologies for developing and delivering software that can provide the flexibility needed to explore new solutions on complex, less well-understood projects. Agile Software Development (ASD) embraces change as inevitable and provides the necessary framework to accommodate user feedback, adapt, and grow as a team, collaborating more effectively, gaining a shared vision and implementing it, and providing capabilities for evaluation at several discrete points during product development. Customer-defined value is the most important goal of an Agile effort, so there is a natural connection with Human Factors Engineering (HFE, or HF for short) when creating tools for use in the operational arena. HF can provide the needed insight to conform to stringent operations standards, in order to avoid problematic usage of a tool, even if it meets requirements specifications. Examples include avoiding eye strain, fatigue, and minimizing the number of clicks to access needed information.

The tools to be developed to support a higher tempo of operations were just such complex concepts, without well-defined scope and in need of ongoing collaborative feedback with those who would be using the tool. They had to be developed quickly and incorporate changes to requirements when necessary, before first release. Figure 2 details the key differences between Waterfall and Agile approaches to product development when an operational user team and a software development team are isolated from one another in job function. Waterfall emphasizes end users providing an operations concept to a development team that then derives an implementation strategy and builds the releasable product based on requirements set forth at the outset. Oftentimes, if the concept is not well-understood, the design and functionality of the final product may reflect more of development's understanding than the customer's vision, but is considered successful as long as requirements are met. Thus, WSD works better for optimization projects with well-understood requirements and little need for exploration, or, when the users of the tool are unable to support regular evaluation of the product, but the delivery schedule is still fixed. In contrast, the Agile approach involves close collaboration between the two sides to achieve a shared vision, a set of better understood high-level capabilities, a sprint release schedule, a robust user evaluation strategy and schedule, and includes time and processes to adapt and respond to changes along the way. The result of the ASD method reduces complexity and also time-to-completion, and helps users get a valuable tool at first deployment by incorporating timely feedback.

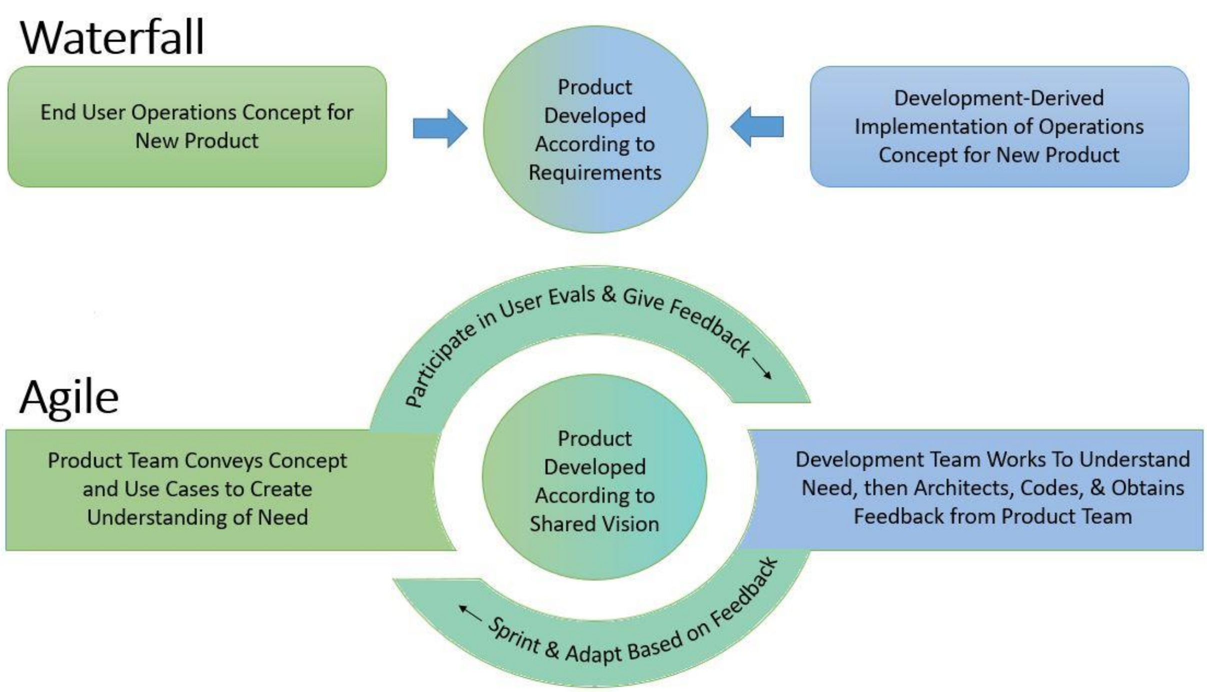

Fig. 2 Waterfall vs POIC ASD Bilateral Approaches to Product Development

In order to accommodate the spatial and temporal location and collaboration challenges, as well as the need for ongoing feedback on developing capabilities, the ASD implementation chosen would have to be able to plan scheduled interactions long in advance. Console operators would need managerial support to relieve their work and shift schedules around user evaluations of the software. Development team members would benefit from having these advance schedules as well, as workloads on developers and testers could be planned to accommodate supporting coding, delivering, testing, and adapting to changes around user evaluations of the software. Creating a shared understanding of each other's work, time constraints, and goals would not be as easy, and it would take proximity and time to build that understanding. Planning and implementing collaborative team-building exercises, and opportunities for as much reward and relaxed fun as possible together could provide openings to build trust. Getting to a shared vision was going to require more work up-front than either team was used to, so implementing new ways of meeting to understand and refine the Concept of Operations (ConOps) as much as possible was necessary, with increased elaboration of documented use cases by the users.

B. Creating a Useful Interface for Project Communication

The best way to ensure a flow of communication between the teams in an orderly fashion would be to designate a lead team representative on each side. In this case, a Product Owner would be the ultimate authority in decision-making and communication for the Product Team (PT) on the operations side, and the Change Package Engineer (CPE) (typically a Systems Engineer) for the project would serve as the coordination point of contact and high-level representative from the Development Team (DT). A set of responsibilities and procedures for each to follow would be needed as well. The Development Team would be composed of at least the following roles: CPE, developers, testers, trainers, network and security professionals, and at some point in the near-future, a Scrum Master. The Product Team would be composed of the following roles: Product Owner, Product Technical Lead (optional but recommended as a back-up to the Product Owner), users in the stakeholder community, Human Factors professionals, operational trainers, and other evaluators as necessary. Figure 3 is adapted from Jim Highsmith's 2010 text on APM [1] and details the ultimate POIC ASD interface between the PT and the DT. Arrows indicate the direction of project communication.

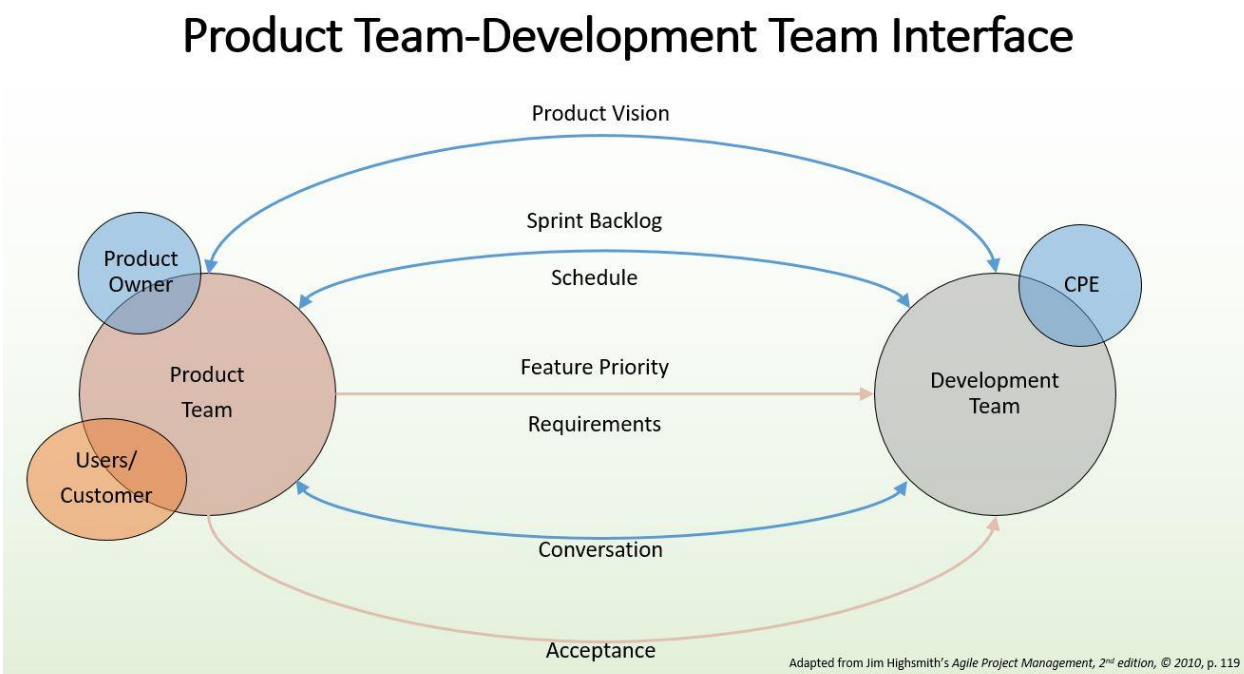

Fig. 3 The POIC Product Team- Development Team Interface and Communications Flow

In order for the tools to be the most useful in operations, Human Factors representation on the PT would be necessary. Also, the software release process was time-boxed, so whatever ASD implementation strategy was chosen would have to account for a fixed schedule and allow cost and scope to be more flexible. Management requirements for approval of the projects that were candidates for ASD would have to be relaxed compared to those for WSD projects in order to allow them to be flexible to exploration and changing customer requirements. The ASD projects would have to exist alongside of, but separately from WSD, so the ASD method would have to be hybridized to fit many of the complexities of a large organization supporting operations with a stringent release schedule. Also, new isolated environments would need to be created to load iteratively, and to let users test out the software without disrupting other development, testing, simulation, or operations work. However, the most important obstacle to success was the least tangible but could make the difference between success and failure- creating an Agile organizational mindset.

III. The Premise of Agile Software Development

The guiding "North Star" of ASD is that it is not as much a set of prescriptive processes as it is a shift in mindset; one that moves from planning *against* change, to planning *for* change, thereby iteratively growing software towards user-defined value. This is a process shift in the development, test, and release of software from one that is *prescriptive* to one that is *adaptive*, which is necessary for operations tools to have longevity. The application of ASD to the product development lifecycle at the POIC permitted the timely incorporation of customer feedback, and, allowed for continuous quality improvements. This resulted in a suite of tools that are efficient, user-friendly, and have enabled ISS ground systems personnel to support the increased pace of payload science operations.

A. Guiding Agile Principles

While this paper discusses the impetus and practical application of an ASD methodology, it is not a comprehensive review of Agile standards and practices. Although APM was brought about by improvements over manufacturing processes in the 1990s, the hallmarks of Agile approaches were first documented in the early and mid-2000s. These guiding principles were set forth in collaborative documents between the implementers and founding authors of the first Agile processes. There are two predominant, short, open-access resources which are helpful to understanding the Agile mindset. The first is the Agile Manifesto, which is a set of prioritized values that are centered on software development. Its focus is on teams of people, collaborations and interactions, releasable software, and responding to change. The second is a definitive guide for Agile practitioners known as the Declaration of *Inter*dependence, which is a set of principles focused on value, shared ownership, innovation, and people-focused self-empowerment of teams with shared accountability. These resources are provided in the Appendix for quick reference.

B. A Hybridized Agile Approach

The POIC created a custom process to facilitate work on the first five tools to support the fourth crew member operations; a hybridized a Scrum-derived, time-boxed ASD strategy. Developing iteratively in "sprints" via the Agile method of Scrum (a reference to Rugby) was selected, with user evaluations at the end of each sprint. However, a Scrum implementation for the POIC also needed to be flexible enough to provide the DT with a way to accommodate evolving requirements definition and new requests discovered during user evaluations. The POIC organizational culture at the time involved a mindset that user evaluations only took place just prior to a tool's release, usually only once, and typically after tool development had completed. The existing feedback mechanisms involved writing software problem reports or software change requests that could be assessed for inclusion in later releases, when development work on the tool could be accommodated again at some future time. Thus, in order for the new ASD method to be acceptable to the DT, it needed to provide them with the flexibility to devise the implementation details to accomplish high-level abstract requirements, and also permit development in iterative cycles with user evaluations occurring far earlier than ever before. Yet, it could not hinder the DT's ability to complete project objectives within an ISS release cycle. The main constraint for POIC software release has always been the rigid, biannual deployment of ISS software syncing with onboard crew transitions. Its strict adherence is owing to the major disruption a patch can have to operations (ops). This previously yielded a natural tendency toward WSD release constructs. ASD strategies in industry focus on rolling, iterative releases that provide incremental value to the customer by delivering complete chunks of software that work together with the prior release as they are created. Thus, providing value to the operations user community would have to be accomplished despite not being able to release new software more frequently than at ground transitions, which formed the largest basis of all for hybridizing a Scrum approach.

Figure 4 details the custom time-boxed ASD approach that was developed for use at the POIC. It showcases the processes alongside the Agile phases, and displays how iterative work occurred in sprint cycles after managerial approval, and what deliverables were outputs of each phase of the lifecycle. In the Envision phase, the ConOps was discussed and revised to provide use case elaboration and use Agile envisioning practices to develop a shared vision for the product. This resulted in a product backlog, the identification of project stakeholders, and development of user stories to accomplish the minimum success criteria emerging from the ConOps. The Speculate phase was characterized by resource estimation, schedule definition, and risk identification which yielded a communicable schedule for user evaluations and identified a delivery timeframe. Work was then broken into sprints to match the development-test software drop schedule in collaboration with the PT's evaluation time limitations. At this point, the CPE would create and gain DT approval of information on the project to take to NASA management for Authority To Proceed (ATP). If approved, the Explore phase of working in iterative sprints would begin as soon as the approved resources became available. Each sprint in the Explore phase was characterized by a cycle of design, develop, test, user evaluation, review and adaptation. In ASD methodologies, the Explore and Adapt phases go hand-in-hand for iterative development, and aren't truly two discrete phases, which is what facilitates the feedback process. Outcomes of these include adapting to identified changes, metrics generation, new risk identification and mitigation strategies, updating sprint and product backlogs, discussing any content trade-offs as needed, and documenting future ideas. Sprint Retrospectives can aid mining this information and also provide relaxed opportunities for the team to meet to discuss how to make the team and the work flow even better during the next sprint. During the Close phase, the final product is delivered out of development and test and into the hands of trainers and operations personnel to prepare for deployment. Lessons learned and metrics are gleaned, and shared. Requirements are frozen and any additional user requests and future ideas are documented to be revisited later. Once the tool is deployed to operations, the team keeps listening for feedback from other stakeholders as well. Operational use metrics, FCT feedback, new concepts arising from use, and needed process improvements are identified and documented. If available, the team can decide to raise any issues or leverage potential new synergies and opportunities to management when deemed appropriate.

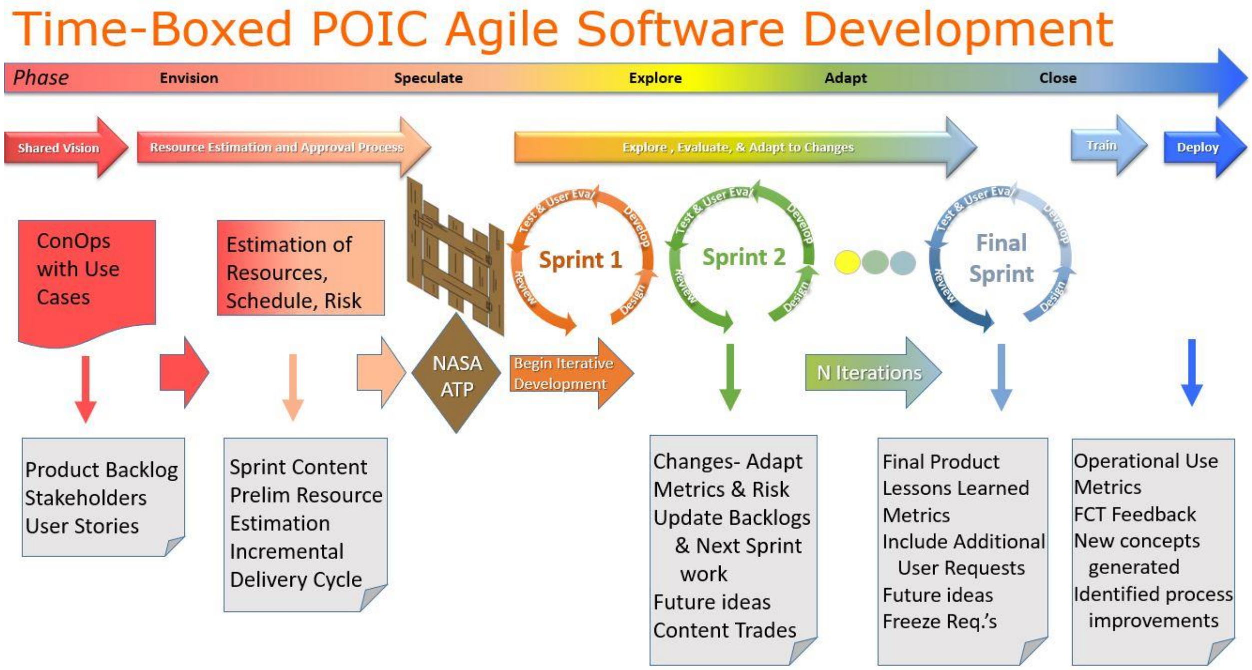

Fig. 4 The POIC Hybridized ASD Lifecycle

Under normal circumstances, implementing highly desired changes to ISS payload science ground-support software already in ops can take at least six months, but often over a year. Software defects occurring in ops are highly prioritized, however user requested changes are not typically as critical. This necessitated a different feedback process for the new HOT tools because of the ASD emphasis on providing value as perceived by the customer. However, from the outset there were unique considerations to establishing a mid-development organized user feedback process. For one, the user community was predominantly composed of professionals on rotating shiftwork schedules, so coordinating user evaluations during normal business hours proved challenging. This could only be overcome by strategically planning sprint lengths and providing event notifications, long in advance, to ensure adequate participation. This amount of strict scheduling is not typical of an ASD process. Also, there was no training in place to bring project participants and managers up-to-speed on the ASD process. This was created and delivered just-in-time before user evaluations to support understanding expectations for interactions across the various stakeholder communities.

C. User Evaluations

The process of obtaining feedback on the tools differs from other user evaluation approaches used by non-Agile projects at the POIC in that it is structured to minimize paperwork and turn-around time. Inputs from the PT are sought early and often, with the goal of creating reliable software that has both intrinsic (functional) and extrinsic (look, feel, experiential) quality. This process has been facilitated with the creation of a dedicated Environment for Value

Assessment and Learning (EVAL) for hosting the periodic release of sprint-developed capabilities for assessment by the PT. During a user evaluation, the Product Team gets hands-on experience with the software hosted by EVAL. A set of minimum success criteria drives high-level requirements for capabilities that are delivered each sprint, but leaves the feature implementation details to the talent and technical expertise of the DT. These features are then evaluated at strategic intervals in the EVAL environment. Users ensure that the features implement the required capabilities in useful ways that facilitate operations outlined by the tool's identified use cases in the ConOps, thereby achieving the MSC.

Organizational culture is very important to the success or failure of new ideas, implementing changes, and process improvement. The HOT tool projects were the first time such close collaboration between operations and software development was attempted, so there needed to be a consistent way to increase face-to-face meetings throughout the software development lifecycle in order to obtain reliable customer feedback early and often. This was achieved through more frequent and much earlier user evaluations in the POIC development cycle than previously attempted. For the HOT initiative, HF considerations for Graphical User Interfaces (GUI) and navigation were also important, and ASD offered a chance to incorporate those throughout the project. Despite software development's overall lack of experience working to achieve HF priorities, there was a willingness to break new ground and deliver tools that the users were excited to use, so these change requests were prioritized highly during the feedback and adaptation.

The POIC Agile User Evaluation Process was designed with several important goals in mind. First it had to be efficient, second it had to be only thorough as was necessary, third it had to minimize documentation, fourth it had to provide both teams with the information needed to make decisions to support the most successful outcomes possible for the final product, and finally it had to work with the diverse schedules of the evaluators and resources of the POIC.

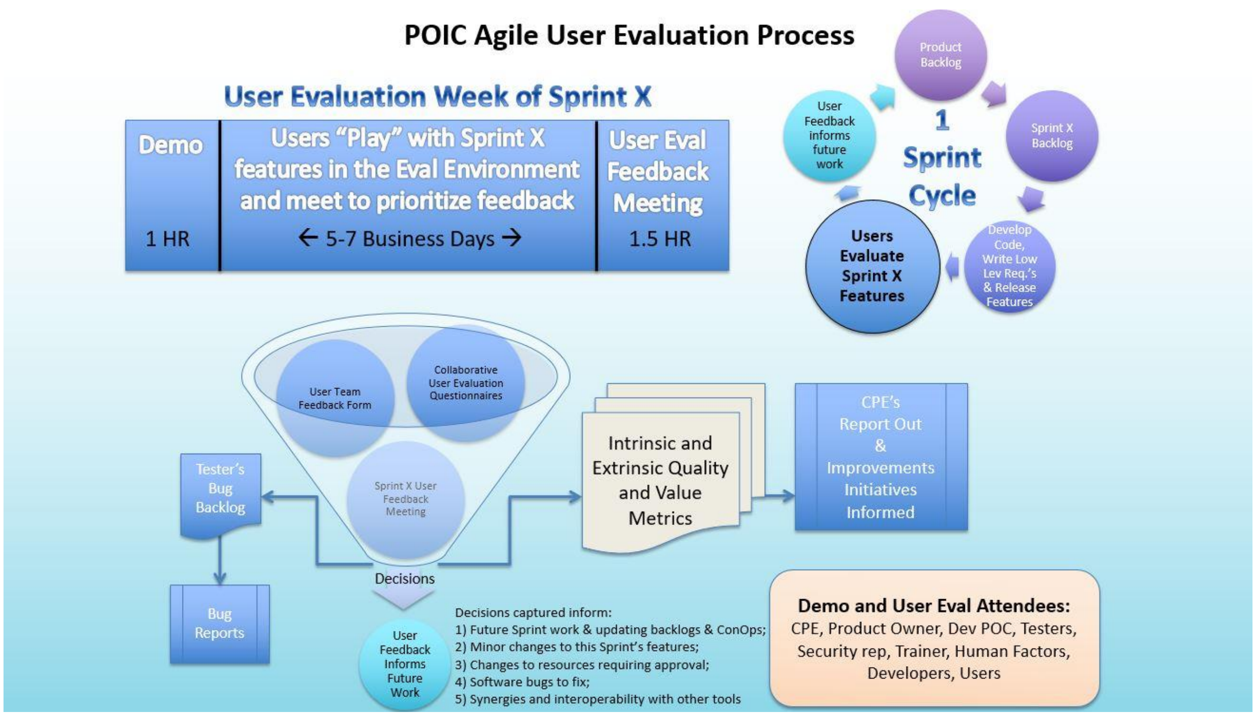

Fig. 5 The POIC Agile User Evaluation Process, Participants, and Outputs

Figure 5 depicts how the process addressed these goals. First, each user evaluation was efficient in that it was designed to accomplish gaining all of the needed sprint feedback in one week. User evaluations began with a demonstration (Demo) of the sprint's developed capabilities by the developers on the project. The first sprint of a project typically only involved Demo and Feedback meetings, addressing the second goal. For the second sprint and forward, the users were pointed to the EVAL environment and provided a work week to assess the software on their own time, satisfying the fifth goal of working around the diverse schedules of the evaluators. Feedback was collected during that time on an individual basis using a collaborative brainstorming and organizing tool from which reports and metrics could be generated. These reports were used by the Product Owner as a central point of communication to hold an internal meeting where users met to discuss their highest priority items. The Product Owner would then document feedback on the sprint's User Team Feedback Form for discussion with the DT at the sprint's User

Evaluation Feedback Meeting at the conclusion of the week-long timeframe. This feedback form was the central discussion piece for the meeting and also served to document on-the-fly agreements, discussions, action items, and Product Team acceptance of the sprint deliverables, accomplishing the third goal of minimizing documentation. Often, the DT would bring up the software in the EVAL environment to facilitate answering questions and gaining common understanding of user requests. Several outcomes from this one meeting accomplished the fourth goal of providing both teams with the needed information for decision-making about future work. Decisions captured identified: necessary ConOps and product backlog updates, minor changes to the presented features, software problems to document, if there were any major changes requiring additional resources that needed managerial approval, and new potential synergies and interoperability with other tools. There were also automatic ways of generating metrics for management informing improvements initiatives. Once the user evaluation week ended and go-forward actions were identified, decisions were made about the next sprint's work and the next iterative phase of development began on the next sprint.

IV. Practical Application of POIC ASD, Benefits, Overcoming Issues, and the Future

A. Example of the TIPS Tool Developed Using ASD Process

The Timeline Integration Product Summary (TIPS) tool [2] was actually not the highest priority cadre tool to be built for the HOT initiative, but it was the first project that had the personnel immediately available to support it from both the Software Development Team (DT) and the Product Team (PT) sides. TIPS had a relatively small collaborative project team (DT and PT) compared to other HOT tool concepts that were "waiting in the wings." It was hoped that ASD could provide the flexibility to respond to the need for a quick release of one fully-functional, valuable timeline planning tool in a single ISS-increment development cycle, while minimizing the risk of rework. To those helping craft the new POIC hybridized ASD paradigm, TIPS presented a small-scale opportunity to try out the new approach, and if it worked with the organizational culture, to then assess scaling it to larger projects. A full discussion of the TIPS tool, its unique architecture, and its amazing journey as the pioneer of the HOT tools and first POIC ASD project are detailed in the SpaceOps paper entitled, "Innovative Development of a Cross-Center Timeline Planning Tool," by Ramon W. Pedoto, this author, David N. Benjamin, and James F. Reynolds.

The vision for TIPS originated on the MSFC POI Planning Team, but after discussions with other FCT teams and the DT, the Concept of Operations (ConOps) was revised to reflect an even more robust tool that could display information of value to many teams outside of the payload activity planning function. All along, TIPS also benefited from the synergy created amongst the team by having regular mid-sprint tag ups to get early-look demonstrations of features in development. The TIPS team often collaborated so well that new ideas sprang up and were documented for later discussion. This was a big impetus for a second phase of the TIPS project. Also, regular testing earlier than during normal waterfall delivery cycles permitted solving software issues early. This left more time to adapt planned work to implement user requested changes. Due to such healthy collaboration and feedback, the TIPS team was able to deliver a very useful tool to flight operations that has been positively received by the whole FCT. In fact, it has been so successful that it has been requested for implementation at NASA Johnson Space Center (JSC) in the fall of 2018 to support their specific activities. Additionally, other FCT tool operations concepts have been drafted that will leverage TIPS to enable more cross-disciplinary functionality. The success of TIPS is due to the hard work of the team members who fully engaged in the new POIC ASD mindset and processes, and worked diligently to provide timely feedback at planned intervals.

B. Overcoming Issues, Processes, and Hurdles- CommDash as an Example

Every new process implemented undergoes some degree of acceptance and resistance to change in an operational environment, but with the right support and dedication from key stakeholders, it can overcome issues and obstacles as they are identified. In the case of the Communications Dashboard (CommDash) tool [3] this broke new ground as the ASD paradigm was scaled to a much larger FCT project than TIPS, and provided pivotal growth opportunities for process improvements.

CommDash was composed of a multi-ISS increment effort to build the applications that its dashboard provides one-stop streamlined access to. The first phase of the project developed the Payload Developer Status (PD Status) [4] application to display upcoming and current payload status information. The next phase developed several individual applications for use within the CommDash dashboard to provide To Do Lists, upcoming readiness, and chat features. The final phase developed the Flight Control Team Log in Console Log Tool (FCTL in CoLT) [5], which brought together existing and new capabilities to greatly facilitate console logging of operational activities, searching, and sharing those logs. Detailed descriptions of these tools, the CommDash challenges and successes, as well as operational benefits are discussed in-depth in another SpaceOps paper entitled "Marrying Social Media Approaches and Space Flight Control," by David W. Scott, this author, Hugh S. Cowart, Andrew J. Nichols, and Robert L. Roy.

CommDash presented its first challenge in terms of conveying an adequate description of its anticipated use that the DT could understand how to implement. Initially there just wasn't enough information on how CommDash would be used in simulations or in real-time operations scenarios. Through the process of clarifying what the DT needed to have in the ConOps provided by the PT, the need for a standardized ConOps template emerged. Also, the DT found it was most useful to have a set of Minimum Success Criteria (MSC), "Highly Desired but Not Required," and "Nice-to-Have" capabilities listed in a tabular format inside of the ConOps and cross-referenced to their high-level descriptions within the rest of the document.

Other hurdles arose from having inconsistent evaluators from one sprint to the next, across the many different console operating positions. This resulted in repetitive questions, longer meeting times, additional re-trainings, late feedback on previous sprints, and a lack of understanding of the kind of feedback that was valuable for evaluators to provide. To mitigate this, PT evaluators were required to identify their back-ups and communicate regularly to discuss sprint outcomes. Both evaluators for a position would be present for the same Agile Product Team training presentations at the beginning of the project's user evaluation process. Efforts were made to only change out evaluators for a project between phases, if at all, so that at least all sprints in a phase had consistent evaluators. Additionally, to avoid a lack of position-specific feedback until the tool was deployed to operations, Product Owners were asked to select evaluators carefully to ensure their participation was more beneficial than just being able to attend the meetings. They were asked to select evaluators who would be able to adequately represent the concerns and priorities of their FCT position. All of these measures served to reduce variances from non-evaluators, reduce redundancy, rework, technical debt, and also to generate more useful feedback at the times it was most beneficial.

During the evaluations of CommDash, it became apparent that there was a need for the evaluators to schedule a dedicated time for different FCT positions to simultaneously interact. Given the complexities of varying console operating schedules between the evaluators, a new concept of scaled-down cadre simulations was developed. These User evaluation-specific SIMulations (USIMs) provided advanced, coordinated team-level interactions with the CommDash tool in the EVAL environment where they could work side-by-side through a simulated scenario. A back-up operations room was utilized and the USIMs were held immediately following product demonstrations so that new knowledge of the tool's most recently developed capabilities was fresh in the minds of the evaluators. The DT developers, testers, and CPE were also required to participate in USIMs. They were available to answer questions, solve some issues, and document others as they arose. Sometimes unique software bugs were identified due to more load testing than the IV&V team could provide alone. New usage scenarios were identified as well, and kinks in the processes were identified long in advance of release to ops so that the PT could work together to figure out what training, process improvements, or standardizations might be needed to optimize the tool's usage once in ops. While it was still not the same as full operational use, and more feedback would eventually be obtained after release, the USIMs provided a valuable two-to-three hour window to glean feedback during the development phase that would otherwise be identified after deployment.

CommDash's development also provided tacit gains which were difficult to measure, but made the jobs of the PT and DT members more fulfilling. The close coordination and need to collaborate to solve problems together provided insight into each other's work constraints, time commitments, strengths, and also put faces to the people developing and using the software. This caused team members to invest more into their work on the project, to form new working relationships that bridged the gap between operations and development, and helped increase empathy toward each team's decision-making process on the tool. New modes of collaboration provided more productive meetings with time, increased the open sharing of questions and ideas, and formed new synergies that became the basis for innovative new tool concepts.

Several Standard Operating Procedures (SOP) have been developed as a result of the ASD implementation. Bridging such a large gap between the two teams for the first time gave rise to the need for more detailed procedures to follow for user evaluations and process expectations. Therefore, an SOP for the POIC ASD paradigm has been developed and is currently going through the approval process for inclusion in an operations handbook. The web portal upon which the HOT tools have been released has a privileging system which needs to be updated whenever tools are deployed with new position-specific permissions. An SOP has been developed which details this hand-off from the development to the operations teams to ensure that the privileges are provided in a timely manner to the correct team leads to manage. Additionally, documenting and standardizing the information to be detailed in new tool ConOps has been created to address the issues first discovered in the Envision and Speculate phases of CommDash. There are still other practices and team trainings that have been created out of necessity to communicate needed information to new team members so that they can understand and interact with the ASD process. As more tools are developed, the expectation is that there will be an increased need for more collaboration between these two diverse teams, but the benefits are well-worth investing time and energy into process improvements.

C. Benefits, Value to the Customer, and the Future of POIC ASD

The application of the ASD paradigm to the POIC development lifecycle has brought about unprecedented levels of coordination and cooperation, resulting in the delivery of highly valuable products that best meet the needs of the ground systems community for payload science activities. The team-level cooperation has been facilitated by introducing regular mid-sprint tag-ups, Agile Sprint Retrospectives, socials, and team-building activities that other projects do not experience. The metrics taken before and after the development of the tools have shown large increases in productivity, situational awareness, error reduction, time savings, and the ability to communicate more efficiently. Additionally, the user community has indicated that they have appreciated the opportunity to interact with the software very early and have their feedback incorporated so readily on the tools' usage, look and feel. Evidence of this positive feedback has been discussed in the other SpaceOps papers mentioned earlier for the TIPS and CommDash tools. Like the PT members, DT members report increased satisfaction in their work because it has been so positively received by the user community.

These results have provided sufficient evidence of success such that the customer has requested the application of ASD for additonal FCT tools going forward. In fact, ASD is the planned process for the upcoming effort to develop a second phase for another FCT HOT Tool. The tool is called SMARTSearch [6], and it will search data sources beyond the POIC to facilitate quickly finding information critical to planning and monitoring payload science activities. Two things have become clear above all else as the HOT initiative ends and that pace becomes the normal tempo of operations: the FCT and POIC DT enjoy collaborating through the ASD paradigm to successfully achieve valuable objectives together, and the projects undertaken have provided substaintial value to the FCT operations community.

V. Conclusion

The hybridized ASD approach to software development of the cadre tools for the High Operations Tempo initiative was undertaken as an optimal way to achieve unprecedented collaboration and cooperation. Its purpose was to accomplish the creation of several new tools in a short timespan that had to be right the first time they were delivered. The tools were needed to provide the ability to support additional activities in order to increase return on investment of onboard science payload operations. The projects utilized a hybridized ASD paradigm because they involved developing special tools to do specific jobs despite the customer not having the ability to know in advance exactly how they would specifically want to interact with those tools. ASD permitted the POIC to take advantage of that uncertainty, and plan for it, by providing a process that facilitated rapid and flexible response to changes in requirements and desired implementation details. It changed the mode of doing business at the POIC on these projects from a prescriptive development and release paradigm to one that is adaptive. The hybridized ASD implementation at the POIC also provided an avenue for strategic investigation and exploration of new technologies that solved customer concerns, and which was needed for these tools to have the longevity necessary to be useful for many years to come. Developing the HOT tools using this process permitted the incorporation of customer feedback throughout the product development lifecycle and allowed for continuous improvement of the quality of each tool so that the final products were released on time as useful, efficient, and user-friendly application of value to the customer.

SpaceOps brings together experienced professionals in ground systems engineering and data management, and opportunities abound to exchange ideas and lessons learned for how to design, develop, and deploy tools to support science activities in space. This presents a fortuitous occasion to share success stories across industry, government, and academia and glean insights from how other partners are developing tools to support an increasing pace of payload science operations. ASD is widely applied in industry because it promotes high return on investments by providing competitive market advantage with early and frequent release of valubale software. However, even a hybridized ASD strategy such as the POIC implementation demonstrates the potential to apply ASD methods beyond industry. It is an optimal approach in the unique sphere of space science payload operations because it offers faster time-to-completion and higher single-deployment customer satisfaction than traditional software development paradigms, while aiming at delivering valuable, high quality products. For this reason, it is gaining traction among the NASA user community. Given the typically unique constraints of government and academic environments, hybrized Agile approaches are often more appropriate than pure ASD, and can add a needed measure of flexibility to environments where very little may have existed before. Each software development project is unique, and each set of team members is unique, so teams will develop their own dynamic and will always have lessons to learn by the time a project completes. Nonetheless, ASD principles can break down barriers to collaboration, increase cooperation, facilitate understanding, and provide increases in work satisfaction for all team members. SpaceOps will provide the opportunity to facilitate a free-flowing exchange of ideas, lessons learned, and feedback that will help refine and increase the robustness of the hybridized ASD approach being employed by the NASA POIC. The author hopes that partners in the international community can benefit from the discussion of new ways of bringing space flight operations, payload science, and software development into closer and more effective collaboration than ever before, for the success of all.

Appendix

The Agile Manifesto

http://agilemanifesto.org/principles.html

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

The Declaration of Interdependence

http://pmdoi.org/

"We are a community of project leaders that are highly successful at delivering results. To achieve these results:

We increase return on investment by making continuous flow of value our focus.
We deliver reliable results by engaging customers in frequent interactions and shared ownership.
We expect uncertainty and manage for it through iterations, anticipation, and adaptation.
We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.
We boost performance through group accountability for results and shared responsibility for team effectiveness.
We improve effectiveness and reliability through situationally specific strategies, processes and practices."

Acknowledgments

Dr. Albers thanks NASA for sponsoring this paper and for the generous financial support of conference attendance to present this methodology and interact with the space operations community. In addition, the author thanks COLSA Corporation and QTEC Aerospace for their support of the author's development and implementation of the hybridized ASD methodology discussed herein. Finally, Dr. Albers wishes to thank the dedicated personnel at the NASA POIC for their hard work making this a successful process, and for their dedication to doing whatever it takes to provide high quality products and continuous operational support for payload science activities aboard the ISS and beyond.

References

Books

[1] Highsmith, J., *Agile Project Management: Creating Innovative Products*, 2nd ed., Addison-Wesley Professional, Boston, 2010, p.119.

Computer Software

[2] TIPS, Timeline Integration Product Summary, Software Package, Ver. 2.0, NASA MSFC POI, Huntsville, AL, 2017.
[3] COMMDASH, Communications Dashboard, Software Package, Ver. 1.0, NASA MSFC POI, Huntsville, AL, 2017.
[4] PD Status, Payload Developer Status application, Ver. 1.0, NASA MSFC POI, Huntsville, AL, 2016.
[5] FCTL in CoLT, Flight Control Team Log in Console Log Tool, Ver. 1.0, NASA MSFC POI, Huntsville, AL, 2017.
[6] SMARTSearch, Software Package, Ver. 1.0, NASA MSFC POI, Huntsville, AL, 2017.