Getting Started on the Right Foot: Developing Requirements for Constellation's Next Generation Space Suit

Louis S. Wheatcraft
Compliance Automation, Inc.

Terry R. Hill
NASA – Johnson Space Center

Abstract. A major challenge identified by the General Accounting Office and NASA's Inspector General is a project's inability to fully define project requirements prior to entering into contractual arrangements, placing projects at risk of significant cost overruns, schedule delays, and performance shortfalls. In 2007, the Constellation Space Suit Element (CSSE) was tasked to produce their Level IV Element Requirements Document (ERD) and baseline it within a 3-month period. Aware of the consequences resulting from a poor set of requirements, NASA's Space Suit Project instituted a continuous requirement validation process that would allow the project team to develop a correct, consistent, and complete requirement set within tight schedule constraints. The result was an order of magnitude reduction of review comments against the ERD compared to the number of comments against the parent Level III System Requirements Document (SRD).

Overview

The emphasis of this paper is on the critical role requirements contribute to a project's success. To clearly illustrate this critical role, the authors present a case study and lessons learned associated with the development and management of requirements for NASA's next generation space suit being developed as part of the Constellation Program (CxP). In the first part of the paper, the case is made concerning the importance of requirements from a risk mitigation standpoint and the importance of having a continuous requirement validation process in place to mitigate these risks. Next, the actual case study is presented. The process used to develop and baseline the initial set of CSSE system requirements is discussed along with the positive results realized by following this process. This process included training the entire team how to develop defect-free requirements and a continuous requirement validation process was implemented that incrementally removed requirement defects before the requirements were included in the official Suit Element requirements document (i.e., the ERD). Both NASA management and potential bidders for the development of the suit expressed their praise for the Suit Element requirement development team regarding the resultant superior quality of the requirements in the ERD. Due to the success of this element-level process, the Space Suit Project integrated it into their continuous requirement validation process to ensure a

correct, consistent, complete requirement set as it matures. This paper concludes with a discussion of lessons learned from this effort.

Risk and Requirements

It is safe to assume that project teams begin their projects with the intent of being successful. To help achieve success, various organizations responsible for creating product development and system engineering processes, recognize the need to define a "good" set of requirements at the beginning of the project. These processes are documented in various standards including the international standard, ISO/IEC 15288, *Systems Engineering – System Life Cycle Processes;* the International Council of Systems Engineering (INCOSE) – INCOSE-TP-2003-002-03, INCOSE *Systems Engineering Handbook*; The National Aeronautics and Space Administration (NASA) NPR-7123.1A – NASA *Systems Engineering Processes and Requirements*; and Carnegie Mellon Software Engineering Institute (SEI), CMU/SEI-2006-TR-008 – *CMMI for Development – Improving processes for better products.* These standards stress upfront processes for defining product scope and requirements.

The result of following such product development and system engineering processes is a set of requirements that describe the characteristics, capabilities, functionality, performance, and quality of the system being developed needs to have to meet stakeholder expectations. These requirements provide the foundation upon which the product design is based. With this emphasis on requirement development and management, it is logical to assume that project managers would want to avoid the risks associated with having a poor set of requirements, and therefore, they would follow the above processes to ensure a set of well-written requirements is established before contracting the design and development of products. Unfortunately, however, this does not happen as often as it should.

As required by the Reports Consolidation Act of 2000, NASA's Office of Inspector General (OIG) reports on the assessment of the most serious management and performance challenges facing NASA as a means to draw attention to areas within the Agency's key programs and operations, which need to achieve greater economy, efficiency, effectiveness, and accountability. The November 2008 report [OIG 2008] focused on five major challenges; one of which concerns acquisition and contracting processes. The report states that one of NASA's major challenges is to:

> Ensure that adequate requirements and cost estimates are developed, program costs are adequately managed, and the most advantageous acquisition and procurement strategies and safeguards are in place to promote competition and ensure programs and projects are within schedule and performance parameters.

Furthermore, in this report, the OIG states: "Although NASA has made fundamental improvements to its acquisition approach, weaknesses in the execution of that approach continue to be reflected in the application and timing of project milestone events and NASA's inability to fully define project requirements prior to entering into contractual arrangements."

The United States General Accounting Office (GAO) [GAO 2009] has also reported on systemic issues involving NASA's acquisition process. Given that NASA spends approximately 85 percent [OIG 2008] of its budget on contracts, these systemic weaknesses pose significant challenges to NASA's ability to make informed investment decisions.

The GAO testified that “NASA’s acquisition strategy of awarding a long-term contract for the design, development, production, and sustainment of a major project before developing a sound business case placed the project at risk of significant cost overruns, schedule delays, and performance shortfalls.”

The GAO noted that gaps in one of NASA’s major projects “included inadequate knowledge of requirements, costs, schedule, technology, design, and production feasibility.” The OIG concluded that:

> NASA must be vigilant in its process of establishing and validating project requirements. Program risks increase when contractual obligations are established prior to the completion of research that would help define requirements. Effective risk management, safety, and mission assurance controls are key to supporting robust and reliable operations in the context of very challenging launch and mission schedules.

Over the years, the GAO has published numerous reports on government projects emphasizing the important role requirement development and management has in the success of a project. In a June 2003 report [GAO 2003], the GAO stated that a key to a successful project lies in the “ability to match users’ needs, or requirements, with the developer’s resources (technology and design knowledge, money, and time) when product development begins.” The GAO asserts their studies show that “doing so can prevent rework and save both time and money.” In conclusion, the GAO states:

The start of product development represents the point at which program managers make a commitment to provide a product that will perform as required and be delivered on time and within estimated costs. Our work has shown that programs are more likely to succeed if program managers are able to achieve a match between user needs, which eventually become requirements, and resources (technology, design and production knowledge, money, and time) at the start of product development. Conversely, if they do not match requirements with resources, cost overruns and schedule delays are likely to occur, reducing an organization’s buying power in other areas.

Effect of Requirements Definition Investment on Program Costs

As reported by Ivy Hooks [Hooks 2001], studies conducted by NASA revealed average cost and schedule overruns of approximately 65 percent on 29 programs. The graphic shown in Figure 1 was produced by NASA’s Comptroller's Office in the early 1990s depicting NASA programs from the 1970s and 1980s. The x-axis denotes the percentage of the program cost expended "up-front" in the requirements definition and design stage. The y-axis denotes the percentage overrun of the program costs. All programs identified on this graph overran its budget. The Orbital Maneuvering Vehicle (OMV), a space tug, was cancelled before it was finished, but at the time of cancellation, it was 200 percent overrun or three times target cost. The OMV suffered from requirements creep: Requirements were added until the last straw broke the proverbial camel's back.

In the commercial world, Boeing Aircraft historically spent approximately 15 percent on most new planes in the commercial sector for many years. For their 777 aircraft, the company spent 30 percent on

the engineering phase. While this chart only shows development costs, Boeing's expenditures early in the program were not only to reduce development costs, but also to reduce the operations costs associated with the continued production and the maintenance of planes.

If a project manager reviews his or her own projections, they may find that only about 5 percent was allotted for the up-front work. This almost guarantees a large overrun in cost. Much

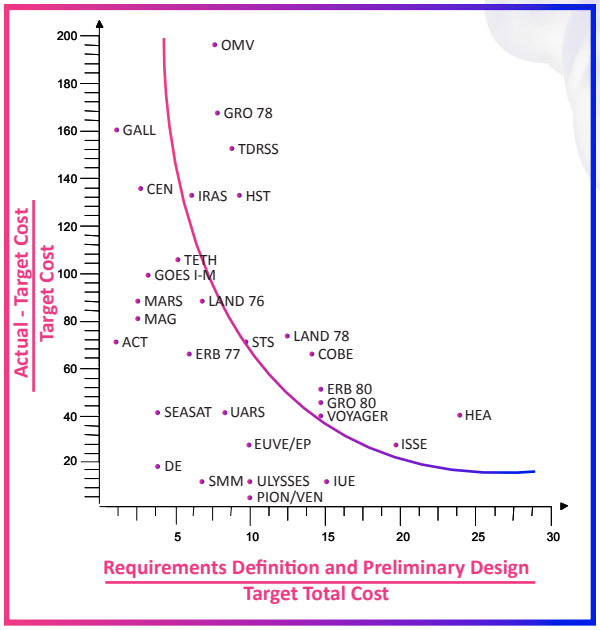

Figure 1: NASA Comptroller Cost Growth Chart

of the cost overruns shown in Figure 1 can be attributed to requirements changes. Much of that change was self-inflicted because of poor up-front work. This is referred to as the "you can pay me now or you can pay me later" chart. It does not take great mathematics to see that an up-front investment can pay off substantially in the development of a system.

INCOSE's System Engineering Handbook [INCOSE 2007] features similar charts that tell the same story concerning both cost and schedule overruns due to not adequately investing in the needed amount of system engineering effort at the beginning of a project. The SE Handbook authors conclude: "Systems engineering effort can be a positive factor in controlling cost overruns and reducing the uncertainty of project execution."

Similar impacts have been reported for software-intensive projects in the industry as well. The importance of using best requirement practices on project success was recently documented in a report by Keith Ellis [Ellis 2008]. In the report, Ellis studied over 100 companies with development projects in excess of $250,000. Using an "average" project as an example, he found:

> The companies using best requirements practices will estimate a project at $3 million and better than half the time will spend $3 million on that project. Including all failures, scope creep, and mistakes across the entire portfolio of projects, this group will spend, on average, $3.63 million per project.

The companies using poor requirements practices will estimate a project at $3 million and will be on budget less than 20% of the time. 50% of the time, the overrun on the project both in time and budget will be massive. Across the entire portfolio of successes and failures, this company with poor requirements practices will (on average) pay $5.87 million per project.

As a result of this study, Ellis found that for 68 percent of the companies evaluated, project success is "improbable." Ellis said, "projects might succeed – but not by design. Based on the competencies present, these companies are statistically unlikely to have a successful project." While these companies indicated they recognized that requirements are important to project success, they still failed to take effective actions to ensure a good set of requirements

and, by doing so, they tripled their chances of project failure. Ellis states:

> Organizations understand conceptually that requirements are important, but do not internalize this understanding and change their behavior as a result. The most successful of companies do not view requirements as a document which either existed or didn't at the beginning of a project, they view it as a process of requirements discovery. Only companies that focus on both the process and the deliverables are consistently successful at changing project success rates.

Based on the previous discussion, it should be clear that requirements are key to the success of a project and that when a good set of requirements is not developed, the project is doomed to failure from the beginning. The following two quotes clearly make these points.

Ivy Hooks, President of Compliance Automation, Inc. states: "People who write bad requirements should not be surprised when they get bad products. But they always are."

In July 2009, NASA Administrator Charles Bolden stated: "Putting forth the same effort, or using the same approach, then expecting different results is ... insanity."

A Winning Product

The goal of all projects should be to deliver a winning product. A winning product is defined herein as: "A product that delivers what is needed, within budget, within schedule, and with the desired quality." A simple and practical definition of risk is: "Anything that can prevent you from delivering a winning product!"

Given the importance of requirements to the success of a project, poor requirements represent a major project risk. One way to mitigate this risk is to have a clearly defined requirement development and management process. As part of this process, requirement validation must be addressed to ensure the best set of requirements possible.

Validating Requirements

Requirement Validation vs. Verification

It is important to understand the differences between requirement validation, verification, and system validation. "Requirement validation" is the process of confirming the completeness and correctness of the requirements. "Verification" is the process of confirming that the designed and built product meets the requirements. "System validation" confirms that the delivered and verified product meets stakeholder expectations.

The following is a summary of the subtle, yet important distinctions between validation and verification: "Requirement validation makes sure you are building the right thing; verification makes sure you built it right, system validation makes sure you built the right thing."

To prevent costly rework, a project manager must validate requirements early in a project, before the design phase. If this is not done, requirement validation and product verification must be exercised together. Because verification is guided by the requirements, the effort may not expose requirements that are, in fact, invalid. By the time components are built and tested, it can be expensive to discover that even though the product meets the stated requirements, the product fails to meet stakeholder needs and expectations.

Requirement Validation

Requirement validation helps ensure your requirements are needed, verifiable, achievable, clear, concise, correct, consistent, and complete. Requirement validation actually begins before requirements are developed; it starts as part of scope definition. Requirements communicate the stakeholder expectations, which are defined when a product scope, to the design team.

There are two types of requirement validation – continuous and discrete. Some requirement validation processes are applied continuously throughout the product life cycle, while other types of requirement validation processes, such as those supporting formal milestone reviews, occur at discrete points in the product life cycle. These milestone reviews include the Mission Concept Review (MCR), System Requirements Review (SRR), System Design Review (SDR), Preliminary Design Review (PDR), Critical Design Review (CDR), and Test Readiness Review (TRR).

During the requirements writing phase, both continuous and discrete requirement validation processes must be established to ensure that each individual requirement is validated as it is written and that the requirements are validated as a complete set. The focus of this paper is on continuous requirement validation.

Continuous Requirement Validation

Continuous requirement validation is performed across the entire product life cycle. The continuous requirement validation process applies to all written documentation of the product: general scope information, operational concepts, interface documentation, and all requirements from the system level down to the lowest level of the system architecture.

A project team needs to have a good understanding of the scope as well as templates, standards, and criteria that can be used to verify that the team is doing the right things. The process requires trained, experienced, and well-qualified writers and reviewers (gatekeepers). The writers have the responsibility to write the requirements along with the attributes needed to help understand and manage the requirements. These attributes include rationale, verification method, and trace to a parent requirement. Once a complete set of requirement attributes is developed, other attributes to help better manage requirements may be employed. These attributes include risk, priority, and the allocation of the requirements to the next level of the system architecture. Writers are responsible for ensuring that each requirement is submitted in the right format with all information required to validate the requirement. A checklist is beneficial tool to confirm that requirements are written correctly. An example of a good requirement checklist is included in the *NASA System Engineering Handbook*, NASA/SP-2007-6105, Rev. 1, Appendix C, "How To Write A Good Requirement."

Upon completion of individual or small "chunks" of requirements, writers must submit them to a "gatekeeper" function before the requirements are included into the formal set of system requirements. The gatekeeper can be an individual, but is often in the form of one or more teams. One team may focus on the editorial and "goodness" aspects of the requirements, while another team may be a more formal approval board that focuses on the technical validity of the requirements. Gatekeepers must have the responsibility and authority to ensure that proper requirements standards are being correctly applied. Likewise, if the requirements do not follow good writing standards, the gatekeepers can send them back to the writer(s) for correction.

In some organizations, the gatekeeper function includes formal "inspections." The inspection process is done incrementally on parts of a work product to help identify defects early and has proven to be very effective in increasing quality. A discussion of formal inspections is beyond the scope of this paper. For more information on work product inspections, refer to the paper written by Larry Fellows [Fellows 2001], *Increase Product Quality, Decrease Development Cost.*

For continuous requirement validation, everyone associated with a project is responsible for what they do. However, management must ensure that processes are in place, and checklists are developed. Furthermore, management needs to see that team members receive proper training and have the correct experience and background required to carry out their tasks. Management must also assign responsibility and authority at the correct level. Perhaps most importantly, management must enforce discipline and hold everyone associated with the requirement development and management process accountable for good requirements.

Everyone is responsible for continuous requirement validation – not just the writer or gatekeeper. Anyone who sees a defect needs to communicate that problem as soon as it is discovered. This allows requirement defects to be spotted and corrected early, before effort and resources are expended based on a defective requirement. To make a big difference, a project manager must make sure everyone knows they have a responsibility to identify and communicate defects; no team member is allowed to ignore defective requirements.

It should be noted that mistakes will be made, as all project teams are composed of fallible human beings. Sometimes, team members are too close to the problem and need someone else to look at their requirements from a different perspective to uncover defects unwittingly built into requirements. The more people held accountable and the more a team looks at the requirements, the less likely it is that a requirement defect will go undiscovered. The intent is not to punish people for making mistakes; the intent is to end up with a good set of requirements. To make this happen, project teams need to adopt a culture that stresses the need for teamwork to produce defect-free requirements and, as a result, defect-free products.

Continuous validation ensures that the old saying, "Never time to do it right but always time to do it over," will never apply to a project. Including continuous requirement validation in a requirement development and management process can be the most effective way to realize process improvement. It reduces the time for the formal milestone reviews by stopping the creation of the "big bad" requirement document, providing a high-quality document instead. By removing defects from the requirement set early in the project, lost time and cost due to rework is minimized.

Project managers should not wait until the major milestone reviews, especially the SRR, to find out they have a bad set of requirements. There is always the danger that sub-par requirements will be baselined, especially if there is a multitude of problems with the requirements at the SRR and the schedule is tight. Baselining bad requirements always leads to wasted resources needed to correct the requirements [putting the project at risk of schedule and budget overruns, especially if a project uses outside contractors, as stated in the NASA OIG and GAO reports discussed previously.

CxP Suit Requirement Development Process

In the spring of 2007, the Constellation Space Suit Element (CSSE) was tasked to produce the Level IV Element Requirements Document (ERD) (CxP 72208) and baseline it within a 3-month period to ensure that the Suit Element-level SRR aligned with the subsequent CxP milestones and to be released in time for a prime contractor Request For Proposal (RFP) in mid-fall 2007. Since other project organizations were significantly further along than the EVA System Project at that time, the rather short and aggressive development schedule was necessary. Aware of the consequences of developing a poor set of requirements, as discussed in the first part of this paper, NASA's CSSE project leadership put in place a carefully designed continuous requirement validation process that would allow the CSSE team to develop a set of requirements within the tight schedule constraints, which reflected the stakeholders' expectations.

Suit Element management sought to kick-start this process and to reduce the risk of ineffective or poor requirement development by acquiring support and leadership from the same requirement training and consulting organization that is part of NASA's Academy of Program, Project, and Engineering Leadership (APPEL) program. This organization also supported the CxP in the development of Level I and II needs, goals, and objectives and the Level II program requirements. As part of this effort and process, the Suit Element team was relocated off-site for the majority of the summer of 2007 in order to free the team of the daily distractions, which would hamper the success-oriented schedule.

High-Level Suit Requirement Development Schedule

A summary of the team's schedule follows illustrating the tight timeframe the CSSE team was operating under and the necessity of doing the job right the first time. It should be noted that based on team requests, an additional week was added to the original schedule for the Suit Element SRR due to the pace of the summer's activities and the need for more time to ensure the quality of the requirements and coverage was sufficient for the project milestone. After the one-week delay, it was determined that all products met expectations and the CSSE Project was given the go-ahead to proceed to SRR.

To initiate this effort the entire requirement development team attended a CSSE team kick-off meeting on May 31, 2007. During this meeting, the team was trained in the proper method for developing requirements and given examples of good requirements. They were also introduced to the requirements development process and the schedule.

Following the Suit Element kick-off meeting, each of the subsystem teams held kick-off meetings. In one of these meetings, the Pressure Garment Subsystem (PGS) team lead clearly communicated her views on the importance and criticality of the task they had been assigned. The subsystem manager told the team the following:

- The requirements document is probably the single most influential piece of paper that we have control over in the entire Constellation Program [Suit Element].
- This is our chance to make sure that we are asking for what we really want. Let's get it right.
- This is a big, fat, hairy deal. If we don't get this right, folks, 20 years from now people will be shaking their heads and saying, "What were those yahoos thinking?"
- I'll be around and don't want to go to that meeting."

  - CxP EVA Suit Element PGS Team Requirement Kick-off Meeting, May 2007

To develop the CSSE requirements, the CSSE team engaged in the intensive development process outlined in Figure 2. Starting on June 1, the team began the requirement generation activities, which ended June 22. During the first week, the requirement consultant, who provided initial training during the kick-off meeting, attended each of the subsystem team meetings to offer real-time, hands-on training and mentorship. A continuous requirement validation process was in place, which required each team to submit their requirements weekly for a quality check. With this set of validated requirements, the development of the draft CSSE ERD took place the last week of June, and the ERD was updated for the SRR the first week of July.

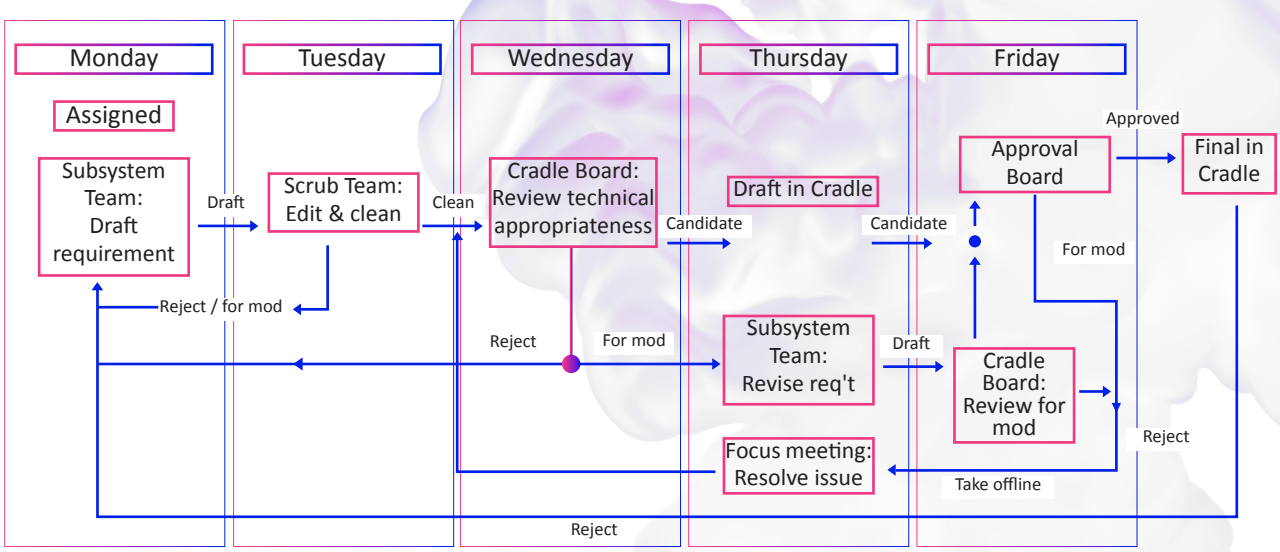

Figure 2: Requirement Development Process Used for CSSE ERD Development in 2007

The CSSE SRR kick-off occurred on July 10 and the SRR wrapped up with the closure of the SRR approval board on August 7. The SRR actions and the CSSE ERD update took place between August 9 and October 20; actions were closed out with the baselining of the ERD on October 23. Because the CSSE team was concerned with producing the needed product with the desired quality of requirements, they continued to work on requirements concurrently with the SRR activities and performed a significant update to the ERD by releasing revision A to the ERD on October 29, in time to support the release of the prime contractor RFP release.

Ground Rules

To ensure that the quality of the development process met expectations, a set of ground rules was established. These ground rules were based on the CxP's Requirement Engineering Management Plan (REMP). The rules included: requirements shall meet the criteria for good requirements in the REMP "Checklist for Good Requirements," each requirement shall have a rationale statement – no copying parent requirement with a noun change, and all requirements shall be verifiable, clear, concise – if it can be interpreted in more than one way, it is not ready for acceptance. The requirement management tool used by the CxP is CRADLE. This tool was used to document requirements prior to baseline; however, post-ERD baseline, the draft requirement revisions were managed outside of CRADLE until change approval by the Suit Control Board (as shown in Figure 3), at which time the revisions were entered into CRADLE.

Teams and Process

The four subsystem requirement teams [representing the three suit subsystems and the Systems Engineering and Integration (SE&I) team] were responsible for the draft and modification of new requirements, providing the core technical expertise, and providing the grass-roots requirement decomposition and traceability effort. As seen in Figure 2, the products of the requirement teams then flowed to other teams providing the gatekeeper function, which then evaluated and acted on the draft requirements. Figure 2 also illustrates the criticality of the different teams to develop the requirements in a very deliberate and timely manner in order to meet the tight schedule.

Scrub Team

The scrub team was the first line of "triage," as it was coined, due to the aggressive schedule. This team was responsible for the review of requirements from each of the four requirement teams and for editing and "cleaning" the draft requirements based on the ground rules and the REMP "Checklist for Good Requirements." The scrub team performed the gatekeeper function from a requirement editorial and goodness perspective. If they could "fix" a requirement and verify that the intent was understood, they would rewrite the requirement and pass it on to the CRADLE Board. If intent was not understood and the requirement needed more work, it was sent back to the appropriate requirement team for modification or rework. In an effort to implement continual process improvement, the scrub team kept a list of common defects and briefed the requirement teams daily at the morning team tag-up.

The scrub team membership consisted of a representative from the SE&I team, a CRADLE operator, a Compliance Automation requirement consultant, and a technical writer. Additionally, a subject matter expert (SME) from the applicable team was present to answer any questions regarding the intent of a given requirement. A primary requirement evaluation criterion was that if the intent of the requirement was not clear and concise or if it could be interpreted by the scrub team in more than one way, then it was sent back to the development team for rework or modification.

CRADLE Board

Once a set of requirements was approved by the scrub team, it was sent to the CRADLE Board. The CRADLE Board had two primary functions in the development process: (1) it was used to view the requirement traceability for the first time in the construct of the CRADLE environment and (2) it ensured all required data was present. Traceability was important to ensure the Level IV Suit Element requirements flowed down correctly from the level III EVA requirements and Level II programmatic requirements. Additionally, the CRADLE Board determined if requirements were technically appropriate, ensured that the correct mission phases were applied, and that any outstanding issues were resolved. The CRADLE Board was used as a dry-run for final approval. This board was composed of an SE&I team representative, Suit Element management, a CRADLE operator, subsystem leads, and additional SME support as required.

Approval Board

Once a set of requirements was approved by the CRADLE Board, it was sent to the Approval Board for final approval. The Approval Board provided consensus and final approval for a requirement to be included in the CRADLE database and ultimately, in the ERD. Approval Board membership included an SE&I team representative, Suit Element management, subsystem leads, a CRADLE operator, and specialized support as needed.

While it was evident that, as significant overlap

existed in team members supporting the different steps of the process and the various teams, and there were some perceptions of process overkill while working within a small team, the Approval Board provided a very effective way to process requirements and address issues given the different responsibilities of each team. Thus, it provided insight into how the meetings were run and managed. It enabled the teams to focus on the expectations of requirement development at each stage, helping them avoid being distracted by trying to resolve all issues with each and every requirement in each meeting. Additionally, with a limited attendance in each group, meetings limited the discussions to what was required while maintaining enough of a varied exposure to the different subsystems and groups that would be affected by a requirement.

Results

The result of this approach was an order of magnitude reduction of review comments/ Review Item Discrepancy (RIDs) against the Suit ERD compared to the number of RIDs against the parent level III System Requirements Document. For the Suit ERD SRR, 0.38 RIDs were received per Suit Element requirement. In comparison, the parent document had a 2.94 RID to requirement ratio at its SRR six months prior.

Both NASA management and potential bidders for the development of the suit publicly recognized the Suit Element requirement development team for their superior efforts by stating that the requirements were "... the most comprehensive and of the highest quality they ever remember seeing." and the JSC Engineering Directorate Crew and Thermal Systems Division (CTSD) Chief stated, "I can't say enough about how amazed I am by this set of requirement documents. As far as I know, no other Constellation Program has allocated and decomposed anywhere near to this level of depth. You are the first. I have also never seen anything like these from previous programs."

CxP Suit Continuing Requirement Management Process

Requirement Review Process Post-ERD Baseline

As a result of the extremely compressed requirement development schedule, there remained several areas that needed more work and issues that required resolution prior to preliminary design taking place. These open areas included addressing "To be Determined" (TBD) and "To be Resolved" (TBR) designations in the requirement set, finishing the verification requirements, defining the internal interfaces and maturing applicable interface requirements, and adding Ground Support Equipment (GSE) requirements to the ERD. Therefore, there was requirement development and maturation needed in order to continue this process. Given the success of the continuous requirement validation process used to develop the initial set of ERD requirements, it was decided to use a similar process to mature the ERD. While the basic functionality of the resulting process is the same, the process was adapted to function as more of a nominal project process and was adapted accordingly from what was used to develop the initial set of requirements. The most significant changes from the initial process to what is in use today, include the alignment of the process to more of an engineering development model where the scrub team and CRADLE Board functionalities have been replaced with the System Engineering and Integration Working Group (SEIWG) and processing any requirement changes takes place outside of the CRADLE environment. Once the changes are approved by the Suit Control Board (SCB) (formerly known as the Approval Board), they are then implemented within CRADLE (see Figure 3).

SE&I Working Group

The SEIWG currently meets once a week to discuss new and existing requirements and/or requirements issues such as TBDs/TBRs, action items, allocations, traceability, verification requirements, and technical correctness of all requirements and requirement changes. The SEIWG performs the scrub team function addressing editorial and requirement cleanliness to ensure the requirements are complete and in the correct format. If a requirement or requirement change is determined too messy or incomplete, it may be sent back to the applicable subsystem team to be rewritten. This venue is also where any requirement changes or added requirements can be suggested for review. The SEIWG provides the SE&I pre-coordination function for the SCB.

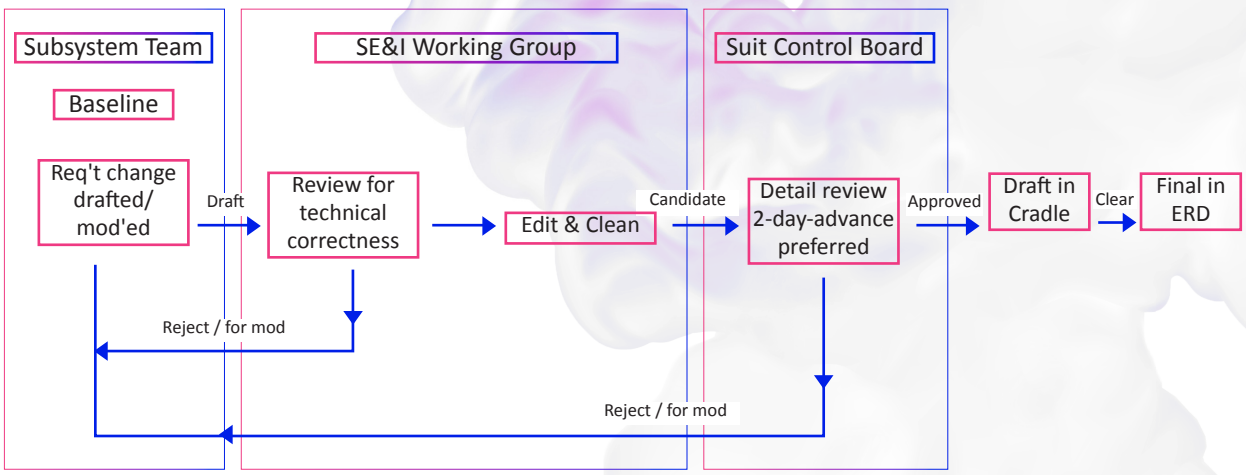

Figure 3: Post-ERD Baseline Requirement Maturation Process

Suit Control Board

The SCB meets for a detailed review of all requirements and requirement changes to determine if the requirement or change is acceptable. It provides consensus and final approval for requirements and changes to requirements prior to being added into CRADLE and the ERD.

What We Could Have Done Better

In hindsight, there are not many things the Suit Element Project would have done differently, given the overwhelming success and product generated as a result of the continuous requirement development process that was put in place. One of the major shortcomings of an extremely challenging schedule-driven process was the amount of open work.

At the time the CSSE ERD was baselined, a large number of open TBDs/TBRs existed; there was no time to perform the research, testing, or analysis required to fully define or validate requirements at the time of writing. However, this choice was evaluated by the team and deemed acceptable based on program schedule constraints.

Additionally, at the time requirements were initially written, the nature of the Suit Element modular system, as well as the inherent number and importance of interfaces, were not known. Furthermore, Suit Element interface identification and definition was incomplete. Full requirement traceability from parent and guiding documents was not as complete at the beginning of the requirement generation effort as it should have been. While there were requirements that had to be revisited to ensure proper parents or children, the percentage of such requirements was small. Moreover, given the rapid pace of requirement generation, it was difficult to keep requirements at the right level and some requirements may not have been needed. In the final analysis, it was decided that it was better to develop too many requirements than to not provide enough requirements, as requirements are cheaper to delete than to add to a contract. Another problem was the quality of the verification requirements. During the maturation process, missing verification requirements were added and existing verification requirements were rewritten.

Parting Thoughts and Lessons Learned

- In summary, this case study demonstrates the benefits of having a continuous requirement validation process in place and executed. This approach has proven to be successful for both large and small development projects. Below are some parting thoughts and lessons learned from a project management perspective.
- Address requirement risk at the beginning of a project. It is worthwhile to do the best job up-front to ensure a quality set of requirements. Changes to requirements result in design changes, which impact schedule and budget. Design changes will result in some level of recertification, which can impact cost and schedule. Frequently, these design changes are larger and more expensive than planned.
- Develop, implement, and enforce a formal requirement development process that includes continuous requirement validation. This is critical during the initial development push as well as during final requirement development, development of the corresponding verification requirements, and to sustain and maintain the requirements.
- The operational concepts should be thoroughly thought out at the beginning of a project to allow the writing of better and more comprehensive requirements. This will eliminate rework and multiple recertification cycles later in the product lifecycle, preventing cost overruns.
- Train a team and enforce the process through project leadership. Do not only send team

members to training, but have them use the language taught in training, set up processes to match what is presented during training, and invest in either continual "refresher training" or provide a mentor to ensure a learned behavior is followed by the team; the process must be learned and it does not come naturally to most. One of the metrics for evaluating the success of a team's training may be when a team member says, "What would [the requirement consultant] say about this requirement?" This would be followed by the team applying what was learned without the requirement consultant present.

- Allocate the time and resources needed to do the job right – the first time. Small investments early on will provide large dividends later in saved or avoided costs and schedule slips.

References

CAI Requirements Development and Management, Seminar Workbook, October 2009, Compliance Automation, Inc. 2009.

Ellis, Keith. Business Analysis Benchmark – The impact of Business Requirements on the success of technology projects, IAG Consulting, 2008.

Business Analysis Benchmark – The Path to Success, IAG Consulting. 2009.

Fellows, L. A. *Increase Product Quality, Decrease Development Cost*, Compliance Automation, Inc. 2001.

GAO-03-598, *Matching Resources with Requirements Is Key to the Unmanned Combat Air Vehicle Program's Success*, United States General Accounting Office, June, 2003. https://www.gao.gov/assets/gao-03-598.pdf

GAO-09-844, *Constellation Program Cost and Schedule Will Remain Uncertain Until a Sound Business Case Is Established*, United States General Accounting Office, August 2009. https://www.gao.gov/assets/gao-09-844.pdf

Hooks, I. F. and K. A. Farry. *Customer-Centered Products: creating successful products through smart requirements management*; AMACOM Books, NY, NY, 2001.

INCOSE. *Systems Engineering Handbook - a guide for system life cycle processes and activities*, Version 3.1, INCOSE-TP-2003-002-03.1, August 2007, ed, Cecilia Haskins.

ISO/IEC 15288, *System Engineering-System Life Cycle Processes*, October 2002.

NASA OIG, Inspector General, *NASA's Most Serious Management and Performance Challenges*, Report, November 2008. https://oig.nasa.govNASA2008 ManagementChallenges.pdf

NASA, Requirements Engineering Management Plan, CxP 70016, Rev. C1, May 2007.

NASA, *System Engineering Handbook,* SP-2007-6105, Rev. 1, December 2007. https://www.nasa.gov/sites/default/files/atoms/files/nasa_systems_engineering_handbook.pdf

NASA, *Systems Engineering Processes and Requirements*, NPR 7123.1A, March 2007. https://nodis3.gsfc.nasa.gov/displayAll.cfm?Internal_ID=N_PR_7123_001A_&page_name=all

NASA, *Space Flight Program and Project Management Requirements*, NPR 7120.5D, March 2007.

NASA, Space Suit Element Requirements Document, CxP 72208, Rev C, October 2009.

Software Engineering Institute, *CMMI for Development – Improving processes for better products*, Version 1.2, CMMI Product Team, Carnegie Mellon, August 2006.

Wheatcraft, L. S. and I. F. Hooks. *Scope Magic*, 2001.

Wheatcraft, L. S. *The Importance Of Scope Definition Prior to Developing Space System Requirements*. INCOSE INSIGHT, Vol. 4 Issue 4, January 2002.

Wheatcraft, L. S. *Developing Requirements for Technology-Driven Products*. Presented at INCOSE 2005, July 2005.

Wheatcraft, L. S. *Delivering Quality Products That Meet Customer Expectations.* Published in CrossTalk, The Journal of Defense Software Engineering, January 2003, Vol. 16 No. 1.

About the Authors

Lou Wheatcraft

Lou Wheatcraft has over 40 years experience in the aerospace industry, including 22 years in the United States Air Force. Over the last 5 years, Lou has worked for Compliance Automation, where he conducts seminars on requirements activities, writing good requirements, and managing requirements for NASA, Department of Defense (DoD), and industry. Lou has a Bachelor of Science (B.S.) degree in Electrical Engineering, a Master of Arts (M.A.) degree in Computer Information Systems, a Master of Science (M.S.) degree in Environmental Management, and has completed the course work for an M.S. degree in Studies of the Future. Lou is a member of INCOSE, co-chair of the INCOSE Requirements Working Group, a member of PMI, the Software Engineering Institute, the World Futures Society, and the National Honor Society of Pi Alpha Alpha.

Terry Hill

Terry Hill serves as the Engineering Project Manager and Constellation EVA System Suit Element Deputy Lead at NASA, Johnson Space Center, where he is responsible for the development of the functional, performance, and quality requirements, preliminary design, and final design oversight of NASA's next generation space suit system. Terry has a B.S. in Aerospace Engineering and an M.S. in Guidance, Navigation & Control Theory from the University of Texas at Austin. While at NASA, Terry has worked on projects and programs spanning verification of International Space Station (ISS) navigation software, Shuttle Design Test Objectives (DTO) and back room mission support, X-38 Crew Return Vehicle navigation algorithm development, Space Launch Initiative technology development, Orbital Space Plane project office ISS-Prime integration, STS-107 Return to Flight Tile Repair capability development, and Constellation Program Space Suit System leadership.