

Air Force Institute of Technology AFIT Scholar

Theses and Dissertations

Student Graduate Works

3-24-2016

Application of System Engineering Leading Indicators to Scrum Agile Projects

Eric M. Shirley

Follow this and additional works at: <https://scholar.afit.edu/etd>

 Part of the [Systems Engineering Commons](#)

Recommended Citation

Shirley, Eric M., "Application of System Engineering Leading Indicators to Scrum Agile Projects" (2016). *Theses and Dissertations*. 411. <https://scholar.afit.edu/etd/411>

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.



**APPLICATION OF SYSTEM ENGINEERING LEADING INDICATORS TO
SCRUM AGILE PROJECTS**

THESIS

Eric M. Shirley, PMP,CSM, Civilian

AFIT-ENV-MS-16-M-183

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

**DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED**

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-MS-16-M-183

**APPLICATION OF SYSTEM ENGINEERING LEADING INDICATORS TO
SCRUM AGILE PROJECTS**

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Systems Engineering

Eric M. Shirley, PMP, CSM, Civilian

March 2016

**DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED**

AFIT-ENV-MS-16-M-183

**APPLICATION OF SYSTEM ENGINEERING LEADING INDICATORS TO
SCRUM AGILE PROJECTS**

Eric M. Shirley, PMP, CSM, Civilian

Committee Membership:

John M. Colombi, Ph.D.
Chair

Kyle F. Oyama, Ph.D.
Member

Brent T. Langhals, Ph.D.
Member

Abstract

SCRUM Agile metrics and System Engineering Leading Indicators (SE LIs) are measures used to evaluate the effectiveness and performance of a project across its lifecycle. The application of metrics can provide information on both internal resources and external project stakeholder requirements and constraints, showing past changes, trends, and potential future impacts. Both SCRUM Agile and SE LIs are designed to give a systems engineer/project manager insight into development progress allowing for progress to be checked at regular intervals as well as early corrective actions to be taken if necessary. This research applied SE LIs to a SCRUM Agile project to demonstrate if it was possible to integrate and utilize the two different sets of performance metrics to enhance a SCRUM Agile project. This thesis identified 16 of the 18 SE LIs could be integrated and utilized in an SCRUM Agile project. Those 16 identified leading indicators were applied to a representative software development scenario in order to demonstrate how it is possible to integrate SE LIs into a SCRUM Agile project.

Table of Contents

	Page
Abstract.....	iii
Table of Contents	iv
List of Figures	vi
List of Tables	viii
I. Introduction	1
Background	1
Problem Statement	1
Research Objectives and Questions	3
Methodology	3
Assumptions/Limitations	4
Implications.....	4
Preview.....	4
II. Literature Review	5
Chapter Overview	5
Agile Method and SCRUM.....	5
Leading Indicators and Agile Metrics.....	6
LI Mapping to SCRUM Agile Metrics	11
LI Integration with a Modern SCRUM Agile Framework.....	12
Application to SCRUM Agile, Iterative Software Development within the DoD.....	13
Summary	15
III. Methodology	16
Chapter Overview	16
Applicability of Leading Indicators	16
<i>Pre-Sprint Phase</i>	18
<i>During the Sprint Phase</i>	21
<i>Post-Sprint Activities</i>	25
<i>Leading Indicators Not Included in Sprint Activities</i>	27
Applicability of Leading Indicators Summary.....	28
Qualitative Comparison Between SE Leading Indicators and SCRUM agile Metrics	28
<i>Product Backlog Development</i>	29
<i>Sprint Backlog Development</i>	29
<i>Team Velocity</i>	30
<i>Sprint Burn-Down</i>	31

<i>Release Burn-up</i>	32
<i>Delivered Business Value</i>	33
<i>Retrospective</i>	33
Qualitative Comparison Between SE LIs and SCRUM Agile Metrics Summary	34
Systems Engineering Leading Indicator Integration.....	34
<i>Selective Integration</i>	35
Representative Scenario	36
Summary	38
IV. Analysis and Results.....	41
Chapter Overview	41
Representative Results	41
Post Sprint-Team Retrospective.....	87
Investigative Questions Answered.....	87
Summary	89
V. Conclusions and Recommendations	90
Chapter Overview	90
Conclusions of Research	90
Significance of Research.....	91
Recommendations for Future Research	91
Summary	92
Appendix A.....	93
Appendix B	95
Description	95
Bibliography	96

List of Figures

	Page
Figure 1. V-Model of a Conventional, Large-System Dev Process (Turner 2007)	8
Figure 2. SCRUM Framework (SCRUM Alliance)	9
Figure 3. SCRUM Activities/Metrics Diagram	17
Figure 4. Projected Systems Engineering Staff and Skills	44
Figure 5. Planned Requirement Validation Trend	45
Figure 6. Planned Requirement Verification Trend.....	46
Figure 7. System Definition Change Backlog Trend Template.....	47
Figure 8. Risk Exposure Trends.....	51
Figure 9. Risk Handling Tracker Template	52
Figure 10. Sprint 1 Burn Down Chart.....	56
Figure 11. Sprint 1 Planned vs Actual Requirements Validated Trend.....	58
Figure 12. Sprint 1 Planned vs Actual Requirements Verified Trend	59
Figure 13. Sprint 1 Requirements Growth Trend	61
Figure 14. Sprint 2 Burn Down Chart.....	62
Figure 15. Sprint 2 Planned vs Actual Requirements Validated	63
Figure 16. Sprint 2 Planned vs Actual Requirements Verified.....	64
Figure 17. Sprint 2 Requirements Growth Trend	66
Figure 18. Sprint 3 Burn Down Chart.....	68
Figure 19. Sprint 3 Planned vs Actual Requirements Validated	68
Figure 20. Sprint 3 Planned vs Actual Requirements Verified.....	70

Figure 21. Sprint 3 Requirements Growth Trend	71
Figure 22. Sprint 4 Burn Down Chart.....	73
Figure 23. Sprint 4 Planned vs Actual Requirements Validated	74
Figure 24. Sprint 4 Planned vs Actual Requirements Verified.....	76
Figure 25. Sprint 4 Requirements Growth Trend	77
Figure 26. Technology Maturity Measurements.....	78
Figure 27. Overall Technical Measurement Progression.....	79
Figure 28. Project Compliance	80
Figure 29. Project Defects and Errors.....	80
Figure 30. Release Burn Up Chart.....	82
Figure 31. Cost Schedule Pressure Limits Diagram	83
Figure 32. Work Product Rejections.....	84
Figure 33. Review Action Closure.....	85

List of Tables

	Page
Table 1. Systems Engineering Leading Indicators/Key SCRUM Agile Metrics.....	7
Table 2. Product Backlog.....	42
Table 3. Sprint Backlog	42
Table 4. Architecture Base Measures	49
Table 5. Identified Risks.....	50

APPLICATION OF SYSTEM ENGINEERING LEADING INDICATORS TO SCRUM AGILE PROJECTS

I. Introduction

Background

Execution of multiple complex projects presents a variety of management and engineering challenges. These challenges are exacerbated by the fact that each individual project has a unique environment which is very distinct from other project environments that it might interact with over the course of its lifespan. Unfortunately, there is not one universal management and performance assessment methodology that can be used to cover all possible project environments. One performance assessment methodology that has been utilized for assessing the development progress of large, complex systems is the systems engineering (SE) methodology with leading indicators (LIs) metrics. SCRUM Agile is another methodology that is used primarily for assessing software development progress. This methodology utilizes a different set of metrics to ascertain the performance of a software development project. This thesis examines how these two individual methodologies, which were individually created for one specific environment each, can complement each other and be utilized together in a joint project environment.

Problem Statement

Throughout the Department of Defense (DoD) and commercial industry, different methodologies are used in order to predict and measure the performance of a project throughout its development lifecycle. Unfortunately, the vast majority of the available methodologies are targeted specifically for either physical systems or software systems

development. However, there may be times when one systems methodology or metric could be used to complement a software methodology or metric. This could be accomplished by studying various different metrics used by systems engineering and software developers and then determining if any of the metrics are complimentary to each other.

However, there is limited published research that illustrates the ability to take the performance metrics of the systems engineering methodology and utilize them directly in a modern software development methodology. Predictive estimates on performance of future systems within the DoD and commercial industry must be evaluated and analyzed to determine if the best practices of one methodology can be combined to refine and improve the performance of another methodology. Through this analysis, the reader will be able to determine whether or not SE methodology and LI metrics can be applied to the SCRUM Agile project methodology and as a result will be able to decide if the new combined methodology and performance metrics are better at predicting and measuring the performance of a software development project.

This research will take a practical approach and look at a realistic scenario in order to determine whether or not the SE LIs can be used in conjunction with the SCRUM Agile metrics. This study will look at two different sets of predictive and performance metrics that exist in the SE and SCRUM Agile environments. By comparing and contrasting these different metrics, this thesis will examine whether or not the SE LIs can complement and enhance the SCRUM Agile metrics and vice versus.

Research Objectives and Questions

The primary focus of this research is to identify which of the 18 SE LIs can be applied to software SCRUM Agile projects in a beneficial way. As such, the thesis objective is to identify and qualitatively assess a set of SE LIs applied during active simulated software project. Any findings on utility could have broad applicability to many efforts that are currently underway in both the DoD and the commercial communities. Some investigative questions related to the primary objective include the following:

1. Can all the recognized SE LIs be utilized in modern, SCRUM Agile, iterative software development or is there a limited set that can be used?
2. How well does the recognized SE LIs qualitatively compare to SCRUM Agile metrics?
3. How can the SE LIs be successfully integrated into the modern, SCRUM Agile, iterative software development process?

Methodology

This research will address the applicability of a possible relationship between the recognized SE LIs with SCRUM Agile metrics. This relationship will be studied through a general qualitative research methodology, but using a representative quantitative scenario. The methodology will be accomplished by first identifying the recognized SE LIs and SCRUM Agile metrics and discussing whether or not the recognized SE LIs can be utilized in a modern SCRUM Agile, iterative software development project. Next, the identified SE LIs will be analyzed and compared to the SCRUM metrics. Finally, a

discussion will proceed on which of the recognized 18 SE LIs can successfully be integrated into the modern, SCRUM Agile, iterative software development process.

Assumptions/Limitations

One assumption for this thesis, used in the simulated scenario, is that projects in the DoD environment and the commercial environment function in the same way. Other assumptions for this thesis include that a project's scope and timeline remain constant during the duration of the project and that a project's stakeholders remain constant and do not modify the scope of a project at any point during the project's lifecycle.

Implications

This thesis potentially has broad implications spanning both the DoD and commercial development communities. Most notably, this thesis may provide ways in which SCRUM Agile projects can be better managed with more concrete or more detailed performance metrics, such as the SE LIs.

Preview

Chapter II will present the literature review, which includes research on the 18 SE LIs as well as the SCRUM Agile metrics. Chapter III will cover the methodology used to determine the relationship between the recognized system engineering LIs with SCRUM Agile principles. Chapter IV will present a simulated scenario in which it will be demonstrated which of the 18 SE LIs could potentially be integrated and utilized in a hypothetical SCRUM Agile project. Chapter V concludes the thesis and will provide a discussion of the results of the simulated scenario presented in Chapter IV.

II. Literature Review

Chapter Overview

The purpose of this chapter is to address the resources used during the information gathering phase of this thesis. Chapter II will begin by first describing what is meant by Agile. Chapter II then will discuss what SCRUM is and how it relates to the idea of Agile and an agile project. Chapter II then will supply background information on SE LIs and SCRUM Agile metrics and identify ways in which they can map to each other in order to give insight into how SE LIs and SCRUM Agile metrics can successfully integrate with each other. Finally, a brief discussion on SE LIs application to SCRUM Agile, iterative software development within the DoD and commercial industry will be presented.

Agile Method and SCRUM

Agile is a specific type of project management that was originally developed specifically for use in software development. However, project managers have since found that applying Agile principles to non-software specific projects can be beneficial. Agile is characterized by its division of tasks into short phases of work, i.e. phases or sprints, which allow for flexibility and quick adaptations to ever changing project requirements.

SCRUM Agile is a type of Agile project management methodology specifically designed for software development. SCRUM Agile is defined as an iterative Agile framework characterized by two to four week sprints. SCRUM Agile is almost exclusively used for software development because it allows small project teams to

produce functioning software in a short amount of time. SCRUM Agile also gives a team the ability to be flexible in a rapidly changing software development process environment.

Leading Indicators and Agile Metrics

SE LIs and SCRUM Agile metrics are two different sets of criteria that both measure the effectiveness of a project. In Table 1, when listed side by side, both sets focus on separate measures, but both strive to identify problem areas within a project's lifecycle. "What is sometimes lost in Traditional World/Agile World discussions is the fact that both groups have the same goal—to deliver a quality product in a predictable, efficient and responsive manner." (Palmquist, Lapham, Miller, Chick, & Ozkaya, 2013) The main observation is both sets of metrics use very different approaches to achieve the same outcome - a successful project that meets a customer's expectations.

Table 1 also shows that the primary focus of SE LIs is on the activities associated with a large scale project and the outside factors that influence it. According to the Systems Engineering Leading Indicators Guide version 2, "SE LIs are a measure for evaluating the effectiveness of how a specific activity is applied on a project in a manner that provides information about impacts that are likely to affect the system performance objectives." (Jones, Rhodes, Roedler, Schimmoller 2010)

Table 1. Systems Engineering Leading Indicators/Key SCRUM Agile Metrics

Systems Engineering Leading Indicators	Key Scrum Agile Metrics
1. Requirements	1. Team Velocity
2. System Definition Change Backlog	2. Sprint Burn-Down
3. Interface	3. Release Burn-Up
4. Requirements Validation	4. Delivered Business Value
5. Requirements Verification	
6. Work Product	
7. Review Action Closure	
8. Risk Exposure	Other Sprint Activities
9. Risk Handling	1. Product Backlog
10. Technology Maturity	Development
11. Technical Measurement	2. Sprint Backlog
12. Systems Engineering Staffing & Skills	Development
13. Process Compliance	3. Retrospective
14. Facility and Equipment Availability	
15. Defect and Error	
16. System Affordability	
17. Architecture	
18. Schedule and Cost Pressure	

The SE LIs also use a root cause analysis approach to improve the success of a large scale project’s performance. According to Rhodes, Ross, and Valerdi, “Leading indicators use an approach that draws on trend information to allow for more predictive analysis.” (Rhodes, Ross, Valerdi, 2009) As such SE LIs are subsequently applied throughout the life cycle of a project. This can be shown by the SE “V” diagram outlined in Figure 1.

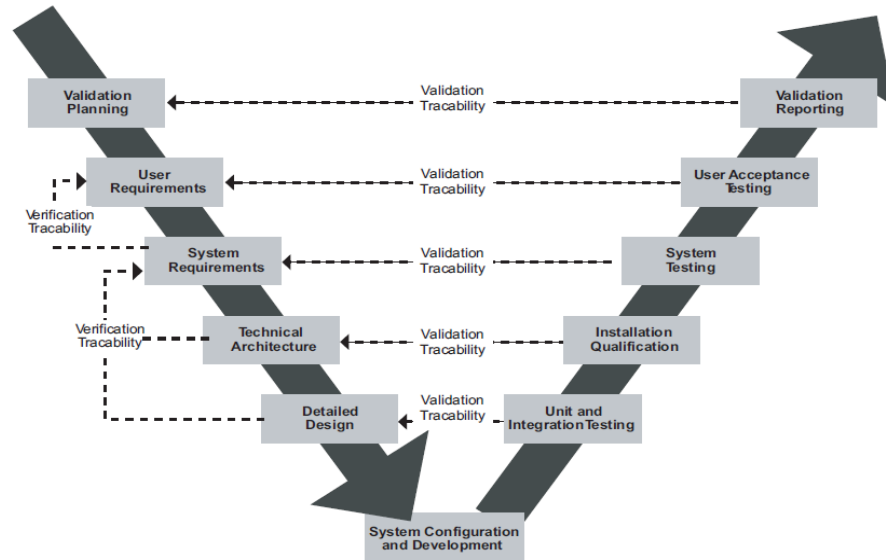


Figure 1. V-Model of a Conventional, Large-System Development Process (Turner 2007)

There are several points throughout a project’s lifecycle where the SE LIs can, and usually are, applied. Unfortunately, the application of root cause analysis at various points throughout a project’s lifecycle adds waste to the overall process and interjects points of leadership and team member frustration within both the DoD and commercial industry environments that can generate inefficiencies into a project that could produce delay, cost overruns, and scope creep. Dr. Turner states that, "as systems grow larger and more complex, new ways of dealing with abstraction, concurrency, and uncertainty need to be developed and Agile approaches do offer reasonable and elegant ways of evolving systems and software engineering toward handling these issues." (Turner, R. 2007)

SCRUM Agile metrics focus more on the team and the team’s performance and how the team’s performance impacts the overall success of a project. According to the Agile Manifesto, “individuals and interactions are more important than processes and tools.” (Agile Alliance) This declaration is best seen in Figure 2, which shows the

lifecycle flow of a SCRUM Agile project sprint and how the team fits into the project sprint.

Figure 2 also shows each of the project iterations is quick, two to four weeks, and is team focused and driven. SCRUM Agile metrics work within this short project sprint timeframe and allow the SCRUM master and the team as a whole to identify where inefficiencies and problems that are influencing a team's overall effectiveness exist. SCRUM Agile metrics also show a team whether or not everyone on that team is fully committed to the team's efforts and/or whether or not additional steps need to be taken to correct a team member's performance and thereby improving the efficiency of the next project sprint.



Figure 2. SCRUM Framework (SCRUM Alliance)

As previously stated, individually each set of metrics function to predict effectiveness of a specific activity in order to see both the positive and negative impacts on performance of a project. However, individually both SE LIs and SCRUM Agile metrics ability to successfully predict the performance of a project is completely

dependent on their individual environment. “Current systems engineering guides and standards provide a waterfall-like structure and key systems engineering characteristics that are imperative for successful system development.” (Kennedy, Umphress 2011) Also the SE LIs are geared towards a root cause analysis of a systems performance to determine the project’s performance success. However, in the SCRUM Agile environment the SCRUM Agile metrics are more geared towards an analysis of the team and thus the metrics are designed towards improving the team’s performance in order to improve project’s sprint success.

Also, they differ in the fact that the SE LIs use root cause trend analysis that focus on reactionary approaches. “They do not provide a framework for planning and managing projects that allow systems engineers to rapidly respond to the changes.” (Kennedy, Umphress 2011) On the other hand, the SCRUM Agile metrics look at the individual and analyze whether or not their future performance is detrimental to the team. This approach, unlike the SE LIs, allows for the team a framework to plan and manage their project(s) in a quick environment thereby allowing for rapid responses to change. This stark contrast in focus is what truly sets these two sets of metrics apart from each other.

Finally, SE LIs are used to be informative to leaders so they can make a decision for the team on the best course of action to take. “They are designed to assist program leadership in delivering value to stakeholders, assisting in interventions and corrective actions to avoid problems, rework and wasted effort.” (Rhodes, Ross, Valerdi, 2009) On the other hand, SCRUM Agile metrics are used to inform the team of their performance on the previous project and identify ways in which they can better themselves and

become a better, more efficient team for the next project. This contrast between a top down approach vs. a self-managing team approach to change is fundamental to each individual set of metrics. It defines how each metric is viewed by the project's stakeholders and drives the project team. This difference is also the main source of inefficiencies on a team if the approach is not handled properly but instead handled in such a manner as to cause detriment to the project team. However, according to Dr. Richard Turner, "while it may impact the management control residing with some of the stakeholders, providing the systems engineering team with the authority and flexibility of owning their own process could radically improve their effectiveness." (Turner 2007)

Both approaches can be successful if applied correctly to its respectful individual environment. However, the efficiencies gained by each in their individual environment vary and depending on how they are applied can lead to inefficiencies throughout the lifecycle of a project. As such a qualitative mapping between the two sets of metrics needs to be identified and understood if such inefficiencies are to be minimized/mitigated.

LI Mapping to SCRUM Agile Metrics

Despite these differences SE LIs share a commonality with the metrics of the SCRUM Agile framework. According to Misra and Omorodion, "The interrelationship between Agile and Traditional Metrics lie in the broad classification of metrics with differences in approaches noticed in the sub-classes based on the nature of the process model involved and the needs of the team." (Misra, Omorodion 2011) As such, the SE LIs and the SCRUM Agile metrics can be linked together. In fact, "The Agile World

uses the same building blocks—it just looks at these things differently than the Traditional World.” (Palmquist, Lapham, Miller, Chick, & Ozkaya, 2013) This is clear when one looks at their metrics side by side.

Their mutual relation also lies in the fact that the SE LIs are inherent action items that are almost automatically subconsciously performed within the context of a SCRUM Agile project. According to Bargh and Morsella, "There is substantial evidence that complex actions can transpire without conscious mediation." (Bargh & Morsella, 2011) In other words, the SCRUM Agile team recognizes that all of the SE LIs are important but in the spirit of team self-management chooses subconsciously to plan/solve the complex project issues associated with each SE LI, either individually or as a team. This subconscious behavior is done in such a manner as not to not affect the project sprint and appears to the team as natural.

LI Integration with a Modern SCRUM Agile Framework

Since there is a direct correlation between SE LIs and the SCRUM Agile metrics, integration of the two should be simplistic. However, SE LI integration to a modern SCRUM Agile framework is a fairly complicated process. Trying to merge the rigid SE framework into a flexible Agile, team centric framework is a challenging endeavor. In fact, if such an endeavor is to be undertaken one must understand the whole project environment. According to Rhodes, Ross, Valerdi, “A holistic evaluation of the environment within which a system operates, the environment's possible states of flux, and system impacts on that environment provide valuable insight that will influence the strategic design and operation of that system.” (Rhodes, Ross, Valerdi, 2009)

SE LIs require a project to have a distinct life cycle process that has a fundamentally set course of action. There are complicated metrics with interdependencies woven in which can, and often do, draw out the development phase of a project. In addition, integrating SE LIs into a modern SCRUM Agile framework requires a team to not only possess the knowledge of leading indicators, but possess the capability to integrate this understanding of the SE LIs into a SCRUM Agile environment without cause chaos which could lead to the failure of a sprint.

However, despite these challenges, it is possible to integrate the two frameworks utilizing the given metrics of both and as such integration should be pursued. In fact, according to S.M. Joseph-Malherbe, “adopting new methods and philosophies and integrating them into current SE practices can significantly improve the capability of engineering systems.” (Joseph-Malherbe 2011) The successful integration of the SE LIs with the SCRUM Agile metrics, however, lies with how well the team can adapt to a merger of the two and utilize both sets of metrics successfully during an Agile sprint. This understanding is crucial if the team is to maintain the SCRUM Agile philosophy and be successful.

Application to SCRUM Agile, Iterative Software Development within the DoD

With a successful integration into a SCRUM Agile framework SE LIs can be applied to iterative software within the DoD and commercial industry. By design the DoD and commercial industry share similar hierarchal leadership styles except for two big differences: the leaders of the DoD have legal authority by law to dictate what actions are to take place on a project and unlike the DoD, commercial industry is profit driven

and as such is dictated by revenue generation which has a direct correlation to customer satisfaction. As such, it is clear to see that in fact it is leadership/management that dictates whether or not a method is applied. “While DoD policy does not specifically prohibit Agile software and systems engineering methods, neither does it make Agile methods easy to implement.” (Wrubel, Miller, Lapham, Chick, Nidiffer, Walker 2014) This is different in commercial industry where leadership/management has moved towards implementing Agile practices and made it simplistic to apply such practices into day to day operations.

However, despite the leadership style of the DoD, SE LIs can be applied to iterative SCRUM Agile software developments in the DoD. However, there are major obstacles that must be overcome in order for successful application of Agile principles to occur. Moe, Dingsoyr, and Dyba found that, "A lack of system for team support, and reduced external autonomy to be important barriers for introducing self-organizing teams." (Moe, Dingsoyr, Dyba 2008) As such DoD leadership/management would need to fully embrace Agile principles and allow the project teams to self-manage. By allowing for Agile principles to be adopted and the Agile teams to self-manage and develop a subconscious utilization of the SE LIs, the major obstacles of lack of team support and reduced external autonomy, that occur naturally within the DoD, would be removed and as such the created Agile teams would be able to successfully apply both sets of metrics into an Agile sprint. This approach will meet all the requirements set forth by DoD and thus will allow for more efficient and better managed teams that produce high quality software.

Summary

This chapter reviewed the background on SE LIs and SCRUM Agile metrics, and the ways in which they are potentially related. Insight was provided into how SE LIs and SCRUM Agile metrics could be integrated. Finally, application of SE leading indicators to SCRUM Agile, iterative software development within the DoD and commercial industry was discussed. Chapter III will cover the methodology used to determine the applicability relationship between the recognized system engineering LIs with SCRUM Agile principles.

III. Methodology

Chapter Overview

The purpose of this chapter is to identify exactly how each of the SE LIs can be applied to a modern, SCRUM Agile, iterative software development process. Chapter III will begin by identifying which of the 18 SE LIs relate to the identified SCRUM Agile metrics, and are thus applicable to the overall modern software development process. By properly identifying each of the applicable 18 SE LIs, one can narrow down the list of 18 SE LIs to just the relevant ones. Once the relationship between the SE LIs and SCRUM Agile metrics is identified, this Chapter III then will discuss how to successfully integrate the identified SE LIs into the modern, SCRUM Agile, iterative software development process. Finally, Chapter III will conclude with a description of the representative scenario that will be used as the basis for Chapter IV. This representative scenario, based on the author's software engineering and management experiences, will show that the identified SE LIs and SCRUM Agile metrics can indeed be successfully integrated into a modern, software development process. This scenario drives the evaluation of the representative metrics. The results and findings of this scenario will be discussed in detail in Chapter IV.

Applicability of Leading Indicators

The SE methodology has a distinct set of metrics that help guide a project team during the SE developmental process. To understand how each SE LI guides a project team, each individual SE LI will be analyzed so that it can be better understood whether or not it can successfully be integrated into the SCRUM Agile methodology and

consequently applied to an iterative software development process. In order to show where each SE LI could possibly be integrated into the SCRUM Agile methodology, the pre-sprint, during the sprint, and post-sprint SCRUM Agile phases, as well as the key SCRUM Agile metrics and other sprint activities associated with each phase of a sprint first need to be identified.

Figure 3 identifies the key SCRUM Agile metrics as well as the other sprint activities associated with each phase and graphically illustrates the flow of the phases and activities during the modern SCRUM Agile, iterative software development process. The key metrics as well as the other sprint activities are listed below each phase.

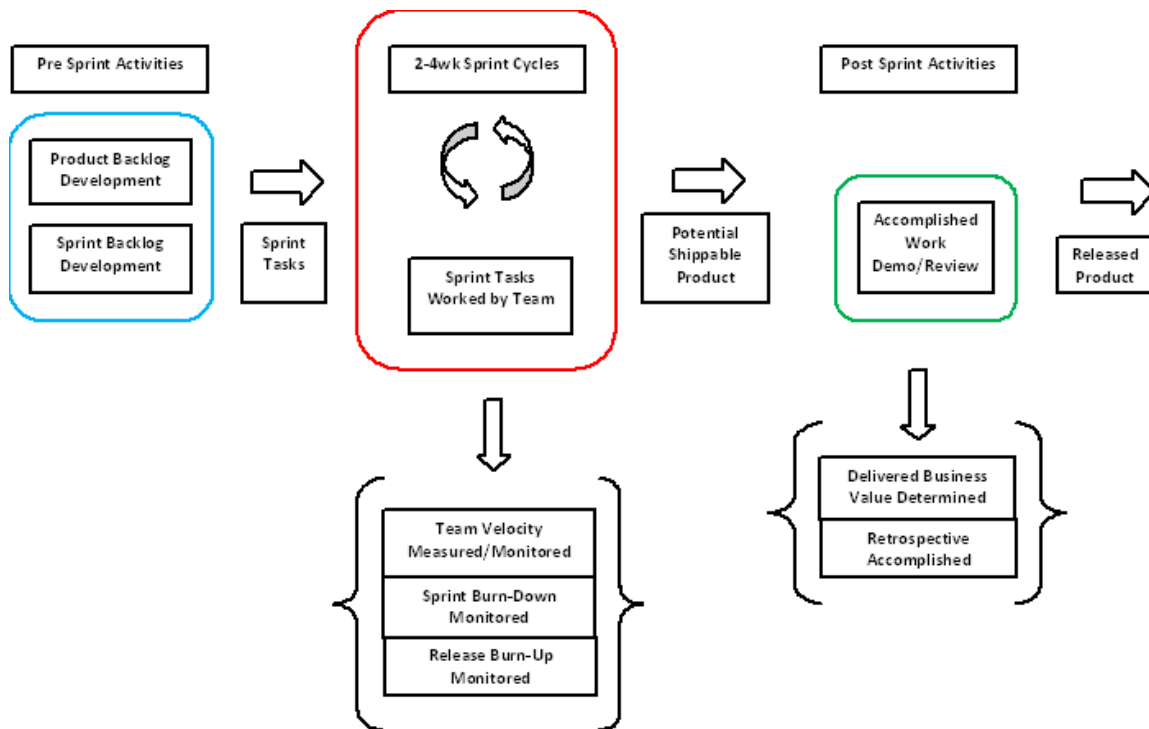


Figure 3. SCRUM Activities/Metrics Diagram

Utilizing the above activity diagram depicted in Figure 3, it will now be easy to identify which of the 18 SE LIs fit appropriately into the modern, SCRUM Agile, iterative software development process. Also, it will be easy to define where exactly

each of the 18 SE LIs fits in the modern, SCRUM Agile, iterative software development process. To do this each sprint phase and activity will be briefly discussed and each SE LI that could possibly fit into each phase and activity of a sprint will be identified.

If there are any SE LIs found that cannot appropriately be applied to the modern, SCRUM Agile, iterative software development process, they will be identified after each phase and activity is discussed and it will be discussed as to why those SE LIs could not be appropriately applied to the SCRUM Agile methodology.

Pre-Sprint Phase

Before a SCRUM Agile sprint begins, there are two pre-sprint activities that must first occur. These activities are the development of the product backlog and the development of the sprint backlog. There are three SE LIs that relate to the SCRUM Agile activities of product/sprint backlog development. Below the product backlog and the sprint backlog activities are briefly discussed. The SE LIs that relate with the product/sprint backlog development activities are also identified and briefly discussed.

Product Backlog Development

The first activity in the modern, SCRUM Agile, iterative software development process is the development of the product backlog. Product backlog development is the process by which the product owner/sponsor identifies the specific features that they want for a particular product in a prioritized format. This development is a high level business look at the project designed to develop broad business requirements the project team(s) are to work on during a given project.

The product backlog development relates to the requirements verification and requirements validation SE LIs in the systems engineering methodology. The

requirements verification and requirements validation SE LIs are used in the systems engineering methodology to identify whether or not the work that is to be done in a given project is relevant for that project. These SE LIs are usually periodically reviewed during the life of a project, much like the sprint product backlog, to ensure that the work at any given time during a project is still relevant or not for that project and whether or not based on their relevance should be retained, modified or discarded all together.

Sprint Backlog Development

The sprint backlog development is the next activity to take place in the modern, SCRUM Agile, iterative software development process. This activity usually takes place after the product backlog is developed. The sprint backlog development activity is the process by which the project team takes the prioritized product backlog that was previously developed and breaks down each of the broad business requirements into actionable workable tasks. The sprint backlog is periodically refined and updated as business requirements change on the product backlog.

The development of the sprint backlog is critical in the sense that it will lay out the required work that is to be done in upcoming sprints in a more defined detail format. Utilizing the sprint backlog the project team can gauge the level of effort needed for each work item and thus decide what work can be accomplished satisfactorily during an upcoming sprint. The project team owns the sprint backlog and as such each team member has input into its development and upkeep. The development of the sprint backlog is arguably the most important step in the entire SCRUM Agile methodology and therefore great care is taken during its development.

The sprint backlog development process relates closely to the architecture and system definition change backlog LIs in the systems engineering methodology. The architecture and system definition change backlog LIs are designed to ensure that the systems architecture as well as any of the desired changes to that system are accounted for and any negative trends resulting are quickly identified and corrected. The architecture and system definition change backlog LIs perform a similar function for the systems engineering methodology as the development of the sprint backlog does for the SCRUM Agile methodology.

Other Pre-Sprint Activities

In addition to the development of the product backlog and sprint backlog, the development of a staffing and skills plan as well as a project risk strategy are also done before a SCRUM Agile project begins development. The development of a staffing and skills plan is done to ensure that the project team is not lacking and necessary skills needed for the work that is scheduled to be accomplished during a sprint. If a skill shortfall is identified, then the SCRUM master will ensure that the proper human resource(s) are added to the project team before the sprint begins so that the skills needed for the project are fulfilled and the sprint is not negatively impacted. The project risk strategy is developed before the sprint begins in order to identify any all risks that could possibly impact the sprint negatively. It provides a framework for a SCRUM master to use quickly reference during a sprint as well as utilize to mitigate any identified risks should they occur during the sprint. It is important that staffing and skills plan and project risk strategy are done before a sprint begins so that valuable time is not wasted on non-sprint items during the sprint.

The staffing and skills plan and project risk strategy pre-sprint activities relate closely to the staffing skills, risk exposure, and risk handling SE LIs. These SE LIs are utilized to provide the basic documentation structure for a project using the systems engineering methodology. Since there is no formal process to complete these task in the modern, SCRUM Agile, iterative software development process, the staffing skills, risk exposure, and risk handling SE LIs can be combined with the staffing and skills plan and project risk strategy pre-sprint activities to form a more comprehensive staffing skill solution and risk strategy plan and thereby enhance their overall effectiveness.

During the Sprint Phase

Once the product and sprint back logs are developed the actual SCRUM Agile sprint methodology can proceed. During the sprint the three performance metrics of team velocity, sprint burn-down, and release burn-up, play a significant role in tracking the performance and overall success of the SCRUM Agile sprint. In this section each of the three metrics are discussed as well as each SE LI that relates to each of the three SCRUM Agile metric.

Team Velocity

The first SCRUM Agile metric to be discussed is “Team Velocity”. Team velocity is a measurement of the average amount of work a SCRUM Agile team can reasonably be expected to accomplish during a given sprint based on past work of a similar nature. It is ideal that a SCRUM Agile team’s velocity will increase linearly upward from sprint to sprint during the course of a project’s lifecycle. This linear increase is due to gained knowledge of the system environment by the project team as a whole as well as individual developer growth. The goal of the SCRUM master is to

develop the project team's knowledge and understanding of a project so that more work can be accomplished in subsequent sprints without changing the duration of the sprint cycle. This increase in work should be continued until a maximum velocity is achieved and the team is said to performing to its max ability and potential.

However, sprint teams can experience a negative downward linear trend in team velocity. This negative trend could potentially be because of a shift in team personal dynamics or a change in team composition (loss or addition of a new team member). As such, it is important for SCRUM masters to monitor a team's velocity in order to quickly identify any slippage in a sprint team's performance as well as the team's ability to maintain that upward linear trend towards maximizing their performance ability and excelling to their greatest potential.

A sprint team's velocity relates closely to the systems engineering staffing and skills, risk exposure, risk handling, and work product SE LIs. This is evident in the fact that a sprint team's velocity is tied closely to a team's staffing and skill as well as the risks the team is exposed to during the duration of a sprint and whether or not that team can handle the identified risks positively. Utilizing the staffing and skills, risk exposure, risk handling, and work product SE LIs during the sprint will give a SCRUM master greater insight into a sprint team's recorded velocity and thereby provide much needed data points as to where a SCRUM master could adjust the sprint team and enhance the their performance.

Sprint Burn-Down

The next SCRUM Agile metric to be discussed is the "Sprint Burn-Down". A sprint's burn-down metric measures the rate at which the work that the team committed

to during a given sprint, i.e. sprint backlog items measured in either story points or hours, is being accomplished. The ideal trend is a downward linear trend towards zero at which point no work is left at the end of the sprint. This ideal trend shows that the sprint team has completed all their committed work items within the given sprint time constraints at a constant steady downward rate of progression.

However, influences such as the team's skill level, team's work ability, internal team dynamics, and shifting project priorities of a product owner mid-sprint could cause a sprint's burn down rate to exhibit an upward linear trend during the sprint thereby showing that the sprint is experiencing a negative trend caused by additional work being added to the sprint. This negative trend must be corrected by the SCRUM Master or Product Owner in order to get the sprint back on track and on a positive downward linear trend in order for the sprint to finish with the ideal zero committed work left at the end of the sprint.

A sprint's burn-down metric closely relates to the requirements, requirements validation, requirements verification, interface, technology maturity, technical measurement, process compliance, and defect and error SE LIs. Each one of these SE LIs indicates, much like a sprint's burn-down rate, the rate at which the project as a whole is progressing forward. Each of the aforementioned SE LI focuses on specific areas of a project that could adversely affect a project if not monitored closely. By utilizing these SE LIs during a sprint, a SCRUM master could diagnose the issue that is causing a negative sprint burn-down trend to occur and thus be able to mitigate the issue causing the negative trend in an efficient manner.

Release Burn-Up

The third SCRUM Agile metric to be discussed is the “Release Burn-Up” metric. This metric measures the rate at which the product that is being worked on during sprints is close to being completed and ready for release. This rate is general a compilation of work completed during multiple sprints and shows the overall progress of a particular product. Combined with the sprint burn-down metric, the release burn-up metric will give the SCRUM master and consequently the product owner a holistic picture as to the overall progress of a project’s product towards completion

The release burn-up metric generally follows an upward linear trend over several sprints and depicts the ever increasing functionality of the product at a steady rate. A downward trend usually means that the work on a particular feature of a product is suffering a setback and as such the product’s functionality is not progressing as it should be. A downward trend could also indicate the decreasing value of a particular product by depicting the shifting in a Product Owner’s priority. In other words, a product owner could decide to discontinue work on a product feature because of shifting company priorities and thus work on the product is terminated and subsequently the release burn-up metric will show this drop in product work progression.

A sprint’s release burn-up metric closely relates to the schedule and cost pressure SE LI. Like the LIs which related to the sprint burn-down rate, the schedule and cost pressure SE LI describes the overall rate at which the project as a whole is progressing towards completion and whether or not the project as a whole is within time constraints and within budget. The schedule and cost pressure SE LI focuses on the outside influences on a project’s schedule and cost that could adversely affect a project’s

completion if not monitored closely by the project's manager. Used in conjunction with the release burn-down metric, a SCRUM master to determine whether or not a project's schedule and or cost is the reason for the trend (positive or negative) is important because if a SCRUM master can eliminate schedule and cost as a influencer, then the SCRUM master could focus their attention on the true influencer and not waste valuable time and effort.

Post-Sprint Activities

After the SCRUM Agile sprint there are two activities that take place. The first activity is an analysis on the SCRUM Agile sprint's return on investment or ROI. This analysis is realized through the perception the product owner views the delivered business value. The second activity that takes place after a SCRUM Agile sprint is the SCRUM team's sprint retrospective.

Below each of these two activities is discussed in detail. Also, the SE LI that relate to these two activities are identified as well as their relationship to the SCRUM Agile post-sprint activities are discussed.

Delivered Business Value

The "Delivered Business Value" metric is the last SCRUM Agile metric to be considered during a SCRUM Agile sprint cycle. Delivered business value is either the actual or perceived value gained from the completion of a SCRUM Agile sprint. The overall success of any sprint can best be represented by the product owner's perceived delivered business value of that sprint's deliverables and whether or not the deliverable meets the product owner's desired end goal. In the modern, SCRUM Agile, iterative software development process, this is by far the most critical metric to consider because

the product owner's satisfaction is the top priority of the sprint team. Therefore, the delivered business value metric is the chief metric used to vector a sprint team's next sprint direction.

The SE LIs that relate to the delivered business value metric are work product and review action closure. Individually work product and review action closure SE LIs gives insight on a different aspect of a project during a specific timeframe in that project's lifecycle. However, combined the work product and review action closure SE LIs show whether or not the project was a successful return on investment for the project sponsor and as such they are an invaluable gauge that can be used by a project manager to determine whether or not added delivered business value was achieved.

Combine with the delivered business value SCRUM metric, the work product and review action closure SE LIs will give the SCRUM additional insight into the individual sprint's return on investment. However, when used over the course of an entire project, the delivered business value SCRUM metric and the work product and review action closure SE LIs can give a SCRUM master as well as a product owner a holistic view at the actual value that was attained from the work done during a project.

Retrospective

The sprint retrospective is not a SCRUM Agile metric; however, it is worth noting because it provides invaluable insight to how the team as a whole felt the sprint went. The SCRUM team's sprint retrospective activity is the event where the SCRUM team, without the product owner being present, can sit down and discuss what worked and what did not work for the team during the course of the sprint. The sprint retrospective is the point where a team's lessons learned are captured and recorded and where the team can

grow together as a collective unit in the hope to continuously improve themselves as a team for future sprints. If done properly, the sprint retrospective can provide a SCRUM master as well as a product owner key knowledge that they can use to improve the team's overall performance during the next sprint by harnessing and leverage what the team feel like it did well/not so well as well what the team feels like they can improve on as a group.



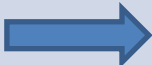




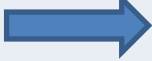
Leading Indicators Not Included in Sprint Activities

Finally, there are two SE LIs; facility and equipment availability, and system affordability that, in the context of this research, do not occur in a SCRUM Agile sprint but rather occur outside of the sprint and as such are not measured by any of the modern, SCRUM Agile, iterative software development process metrics. This is not unexpected simply because the systems engineering process methodology as a whole was developed separately and before the modern, SCRUM Agile, iterative software development process methodology. This, though, is not implying that the systems engineering methodology is flawed; just that there were separate considerations taken into account during the development and refinement of the SE process. However, even though these SE LIs do not relate to the modern, SCRUM Agile, iterative software development process metrics, they could potentially be useful at some point during the modern, SCRUM Agile, iterative software development process and therefore they are worth mentioning

Applicability of Leading Indicators Summary

As was detailed above, a significant number (16) of the SE LIs are applicable to the SCRUM Agile Metrics. Table 2 below shows which SE LI is applicable to which SCRUM Agile metric in tabular form.

Table 2. Systems Engineering LIs Mapped to the SCRUM Agile Metrics

Systems Engineering Leading Indicators (SE LI)		SCRUM agile metrics
1. Requirements 4. Requirements Verification 5. Requirements Validation		Product Backlog
17. Architecture 2. System definition change backlog		Sprint Backlog
12. Systems Engineering staffing skills, 8. Risk exposure, 9. Risk handling		1. Team Velocity
1. Requirements 4. Requirements Verification 5. Requirements Validation 3. Interface 10. Technology Maturity, 11. Technical measurement 13. Process compliance 15. Defect and error		2. Sprint Burn-Down
18. Schedule and cost pressure		3. Release Burn-up
6. Work product 7. Review action closure		4. Delivered Business Value
N/A		Retrospective
14. Facility and equipment availability, 16. System affordability		N/A

Qualitative Comparison Between SE Leading Indicators and SCRUM agile Metrics

Now that it has been determined which of the SE LIs can be applied to modern, SCRUM Agile, iterative software development process, its critical to explain how each one of the identified SE LIs has a positive, productive relationship with each previously

identified SRCUM Agile metric as well as the modern, SCRUM Agile, iterative software development process as a whole. In order to accomplish this, this section will take a look at the pre, during, and post sprint activities and describe how each of the identified related SE LIs relates to either the activity listed or the identified SCRUM Agile metric.

Product Backlog Development

Requirements, requirements validation, and requirements verification SE LIs relates to what takes place during the initial product backlog development and subsequent backlog refinement ceremonies in the modern, SCRUM Agile, iterative software development process. The main differences occur in that in the systems engineering process the requirements, requirements validation, and requirements verification LIs are reviewed in context of the entire project over the entire span of the project whereas in the modern, SCRUM Agile, iterative software development process the product backlog is developed and refined in a more fluid process before each SCRUM Agile sprint by which the requirements could change more rapidly depending on what the product owner views as a priority.

Sprint Backlog Development

Architecture and system change backlog are closely resembles the sprint backlog development because during the sprint backlog development system architecture and system changes are taken into consideration during the development and refinement of work items. Also, during the sprint backlog development, a change control board is sometime utilized to verify that the work items developed for the sprint backlog are relevant and to necessary detail in order to ensure success of the sprint. The biggest

difference between the architecture and system change backlog development SE LIs and the sprint backlog development process is that the architecture and system change backlog development SE LIs, a more firm process that is resistant to change as the project progresses exists whereas in the sprint backlog development process there is more flexibility and acceptance of changes built in so the team can adapt to the ever changing project environment thus allowing for a higher chance of project outcome satisfaction from the product owner.

Team Velocity

The team's sprint velocity is defined as the actual completed hours per sprint the team was able to complete in a given sprint timeframe. The average team velocity is a metric that will help a team gauge their ability as a group to get work done. This metric can be utilized by a team during the sprint planning event to help figure out how much work to commit to during a given effort. It can be used by a SCRUM Master when dealing with a Product Owner demonstrated by the following equation:

$$T_{\text{avg velocity}} = \frac{\sum \text{Sprint Completed Hours or Story Points}}{\text{Total Number of Sprints}}$$

The rate of completed hours (or story points) vs. committed hours (or story points) is demonstrated by the equation:

$$V_{\text{Rate}} = \frac{\text{Completed Sprint Hours or Story Points}}{\text{Total Committed Sprint Hours or Story Points}}$$

With a rate of one being all committed hours completed and zero being no committed hours completed. The goal of any team is attain a rate as close to one as possible.

A sprint team's velocity closely relates to the systems engineering staffing and skills, risk exposure, risk handling, and work product LIs. A sprint team's velocity has

an interdependent relationship with the sprint team's staffing and skill. If the sprint team is not staffed with enough properly skilled people then the team's work capacity will be limited based on the size and skill level of the assigned team.

Also the risks the sprint team is exposed to and whether or not the sprint team can handle the identified risks affects a sprint team's velocity. Risk management and risk mitigation are crucial to the efficiency of a team's velocity. The better a team is at adapting to risk and managing identified risks the better the team will be able to overcome those risks and accomplish more work.

Finally, whether or not the work the sprint team produces during a given sprint is accepted in some manner by the product owner the team's velocity. If a product/project owner does not approve of the work product being presented to them during sprint reviews session, then there stands a strong possibility that the product/project owner will request a change or modification to what was not approved thus potentially affecting the team's ability to complete all committed tasks in a sprint thereby negatively impacting that team's velocity.

Sprint Burn-Down

A sprint's burn-down metric closely relates with several SE LIs. There are several interdependencies between a sprint burn-down's rate and requirements, requirements validation, requirements verification, interface, technology maturity, technical measurement, process compliance, and defect and error

The biggest interdependencies occur with the requirements, requirements validation, requirements verification, interface SE LIs. This because bit the sprint burn-

down rate and these LIs show the rate at which the sprint/project's requirements and project design are being completed and understood by the team.

Another big interdependency is with progression of technology. As the team develops code during the sprint they improve upon the overall system's capabilities. The technology maturity and technical measurement LIs can help gauge this progression during the course of each sprint. The technology maturity and technical measurement LIs are also interdependencies of a sprint team's burn-down rate. If a piece of technology underdevelopment is new and/or cannot be measured accurately with given equipment by the team, then the team's ability to successfully complete any assigned tasks will be degraded.

Finally, sprint burn-down has interdependencies with the process compliance and defect and error LIs. If the process is complicated and compliance is mandatory then the rate in which a team can complete a sprints committed work will be difficult. Also, if there are a lot of defects and errors (i.e. Bugs) then the resultant rework on the flawed items will severely hinder the ability of the sprint team to complete the sprint work items.

Release Burn-up

A sprint's release burn-up metric closely relates to the schedule and cost pressure, technology maturity, technical measurement, and defect and error LIs. Like the LIs which relate to the sprint burn-down rate, these LIs are share interdependencies with the release burn-up metric.

The schedule and cost pressure LI will affect the release burn-up in that if there is any change, good or bad, to the schedule and cost of a project than the rate in which a product is release will either increase or decrease respectively. Also, if the technology is

immature or the way to measure a piece of technology is not known or not up to specifications needed, than the release burn-up rate will also be affected.

Delivered Business Value

The work product and review action closure SE LIs relates directly to the SCRUM Agile delivered business value metric. Each one of the SE LIs share direct interdependencies with the delivered business value metric. Also, any one of the following LIs; Schedule and cost pressure, technology maturity, technical measurement, and defect and error, can significantly alter the way in which a product owner/project sponsor views the gained value of the effort and thus should be considered in the overall assessment of the delivered business value.


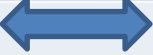

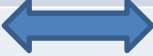

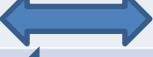

Retrospective

Despite the fact the sprint retrospective is not listed as a key SCRUM Agile metric it still must be discussed how it relates to any SE LIs simply based on the criticality of it. However, there is no SE LI that resembles a sprint's retrospective. However, each previously discussed SCRUM metric and each individual SE LI could potential be a point of discussion for the team during the retrospective ceremony. As such, because of this and the fact that the sprint retrospective is key to the success of future project in which the team is involved with, it is important, as well as no surprise, that all previously discussed SCRUM metric and SE LIs could potentially play a role in the team retrospective.

Qualitative Comparison Between SE Leading Indicators and SCRUM Agile Metrics Summary

As described in the preceding sections, there is a direct comparison between 16 of the SE LIs and the SCRUM Agile metrics. There was no SE LI that compared directly with the SCRUM retrospective activity. Table 3 outlines what was described in a tabular format.

Table 3. Qualitative Comparison Between SE LIs and SCRUM Agile Metrics

Systems Engineering Leading Indicators (SE LI)		SCRUM agile metrics
1. Requirements 4. Requirements Verification 5. Requirements Validation		Product Backlog
17. Architecture 2. System definition change backlog		Sprint Backlog
12. Systems Engineering staffing skills, 8. Risk exposure, 9. Risk handling		1. Team Velocity
1. Requirements 4. Requirements Verification 5. Requirements Validation 3. Interface 10. Technology Maturity, 11. Technical measurement 13. Process compliance 15. Defect and error		2. Sprint Burn-Down
18. Schedule and cost pressure		3. Release Burn-up
6. Work product 7. Review action closure		4. Delivered Business Value
N/A		Retrospective

Systems Engineering Leading Indicator Integration

Finally, now that it has been established that the identified SE LIs do indeed successfully relate to the SCRUM Agile metrics previously identified as well as the modern, SCRUM Agile, iterative software development process as a whole, it can now be determined how the SE LIs can be properly integrated into the modern, SCRUM Agile, iterative software development process. However, there is one optimal way in which the

identified SE LIs can be properly integrated into the modern, SCRUM Agile, iterative software development process. As such it is important to thoroughly discuss that integration method in order to explain how it would be the most appropriate and most successful method to utilize for any given integration situation between the identified SE LIs and SCRUM Agile metrics. Below the integration method is identified and thoroughly discussed how it could potentially be utilized in order to properly integrate the identified relatable SE LIs, into the modern, SCRUM Agile, iterative software development process.

Selective Integration

The ideal method to successfully integrate the SE LIs into the modern, SCRUM Agile iterative software development process is to only select those SE LIs most applicable to a specific SCRUM Agile sprint and disregard those SE LIs that would not be applicable and/or beneficial to the SCRUM Agile sprint. This method specifically calls for the SCRUM master/project manager to analyze the SE LIs that relate to the SCRUM Agile process, as depicted above, and make a judgmental decision as to which of the SE LIs relate to the current project and choose those SE LIs to complement the SCRUM Agile metrics. In other words, this method aims to just take those applicable SE LIs and use them to enhance each project SCRUM Agile sprint. This introduces some risks in the fact that by limiting the SE LIs utilized, a SCRUM master/project manager may limit the available information that could be beneficial when making an informed decision. However, despite this risk, this method of integration may prove to be the best choice given a specific project environment that may not allow for full integration of all

the SE LIs into a modern, SCRUM Agile iterative software development process due to outside managerial forces.

Representative Scenario

In order to show the above integration method is truly the ideal method to use to integrate the identified SE LIs and SCRUM Agile metrics a representative SCRUM Agile web development scenario needs to be developed and conducted. Below is the representative scenario that will be used for this study. It is modeled after a real life web development effort conducted in private industry.

Healthcare Web Development Claims Module Enhancement Scenario

Healthcare Web Development has been in business for approximately 10 years. Their chief product service is developing and maintaining a healthcare care coordination web application that tracks care coordination status/progress of care for patients as well as processes medical claims from start to finish.

The International Statistical Classification of Diseases and Related Health Problems (ICD) is a medical classification list developed by the World Health Organization (WHO). It contains codes for diseases, signs and symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or diseases. The latest ICD code list, ICD-10, was mandated by Congress to Go-Live on October 1st, approximately three months

In preparation of this Go-Live date, the Claims Department director has been tasked to ensure that the Healthcare Web Development claims module has been modified such that it meets compliance with the Congressional mandate; i.e. any claim filed with

the ICD9 format should be rejected after the Go-Live date. The intent is that upon receipt of any ICD9 claim with dates of service or discharge after September 30th, the claim should be rejected with a new error code for the institutional and professional claim.

The director has charged the SCRUM Master with this task. In addition to the task at hand, the director has side tasked you with finding out if there are any ways in which the 18 SE LIs could be utilized to improve the current developmental model that Healthcare Web Development currently uses. The director has made it clear that this is a critical piece of the overall effort and should be treated with the highest priority. The director expects any findings to be documented and justified in a clearly and concisely.

Team Composition:

1 Product Owner (Claims Department director)

1 SCRUM Master

1 Team Lead

1 Developer

Tasks:

1) Modify the existing Healthcare Web Development's Claims module such that it meets the intent of the modification, can successfully integrate with other necessary modifications, and is compliant with Congressional mandate. Subsequent tasks include developing a product backlog, sprint backlog, burn-down charts, release burn-up charts, and a lessons learned document outlining what could be used to improve the overall process. Average team velocity must also be tracked through the entire development cycle and reported at the end.

2) Research any ways in which SE LIs could be used improve the current developmental model that Healthcare Web Development currently uses. All findings are to be documented and presented to the Claims Department director.

Assumptions/Scenario Restrictions:

- Developmental team is set. No additional team members will be added nor any team members be removed during this scenario. In practice, however, developmental team resources are subject to the whims of managerial needs and maybe added or reduced according to priorities. This practice is forbidden in traditional SCRUM Agile development but does occur regularly in industry.
- The Product/Sprint Backlogs are not a true representation of what the actual work for a real life project of this caliber would be. It has been simplified in order to meet the intent of this exercise in a reasonable manner.
- The team will use Unit Test Cases in order to test their code in accordance with SCRUM Agile principles. In practice though, some companies use QA resources to do testing verification/validation. One must be aware that if QA resources are used that the QA process as a whole can lead to significant issues and delays in the developmental cycle.
- The testing process will run concurrent with development. In practice this may not always be the case. In practice testing may begin after all development has been done. Also, addressing any found bugs may wait to the end of all development. In this scenario, however, testing will be done concurrently with development and it will be assumed that no major defects or bugs were discovered. This limitation is in place in order to simplify the exercise and not introduce variables that could take this research effort out of scope.

Summary

This chapter began by identifying exactly which of the SE LIs can be applied to a modern, SCRUM Agile, iterative software development process. By properly identifying each of the applicable SE LIs we were able to narrow down the list of 18 SE LIs to just the relevant SE LIs that relate to the previously identified SCRUM Agile metrics and

focus solely on analyzing those LIs thoroughly to see how exactly each one can possibly be applied to the modern, SCRUM Agile, iterative software development process in a beneficial way.

After the relevant SE LIs were identified and thoroughly analyzed, this chapter then proceeded to identify and discuss how each individual SE LI qualitative compared/related to the identified SCRUM Agile metrics as well as a modern, SCRUM Agile, iterative software development process. This was accomplished by demonstrating how each one of the identified SE LIs has a positive, productive relationship with each previously identified SRCUM Agile metric as well as the modern, SCRUM Agile, iterative software development process as a whole. As shown, it is crucial to understand the interrelationships and interdependencies that could exist between SE LIs and SCRUM Agile metrics and as such its worth re-stating how extremely important that each SE LI related SCRUM Agile metric interrelationship and interdependency be completely understood in order to fully maximize the benefits that could be gained from the identified correlation of SE LIs and SCRUM Agile metrics.

Once the relationship between the SE LIs and SCRUM Agile metrics was identified this chapter then discussed how to successfully integrate the identified SE LIs into a modern, SCRUM Agile, iterative software development process. Successful integration of identified SE LIs is extremely important if a systems engineering project is to be successful and benefit from utilizing a modern, SCRUM Agile, iterative software development process. Therefore again it is vital that any potential approach or approaches that might be used in integrating SE LIs and SCRUM Agile metrics into a modern, SCRUM Agile, iterative software development process together in a systems engineering

type effort be thoroughly understood, analyzed, and discussed in order to ensure the maximum success rate possible of the effort be achieved.

Finally, this chapter concluded with a description of the representative scenario that will be used as the basis for Chapter IV. This scenario will show that SE LIs and SCRUM metrics indeed can be successfully integrated into a modern, SCRUM Agile, iterative software development process. This scenario will also be analyzed and the results and findings will be discussed in detail in Chapter IV.

IV. Analysis and Results

Chapter Overview

In this chapter we will take what was discussed in Chapter III and apply the applicable realistic scenario and develop representative results of the aforementioned real world situation. The overall intent of this section is to take the simulated scenario presented in Chapter III, with applicable constraints, and mirror as closely as possibly the real world. By doing this the investigative questions posed in Chapter I will be answered in sufficient manner. The scenario and the answers to the investigative questions will be discussed in the subsequent sections of this chapter.

Representative Results

After running the scenario through the hypothetical realistic timeline of the scenario, the below results were produced. This section will look at those results generated from a SCRUM Agile pre-sprint point-of-view, at weekly SCRUM intervals for the sprint, at a holistic sprint to sprint view point during the lifetime of the entire project, and at a post SCRUM Agile sprint view point. At each SCRUM Agile point-of-view the potential value that the previously identified SE LIs, discussed in Chapter III, could bring to the sprint will be discussed along with their utilization results.

Pre-Sprint Scenario Activities

At the beginning of the scenario the SCRUM Master met with the Claims Department director and developed Table 2, the Product Backlog table. In this scenario, there were only two Product Backlog Items (PBIs) that would need to be satisfied in order to accomplish the desired results.

Table 4. Product Backlog

1) Put in new EDI edits in the claims module program in order to reject institutional ICD9 claims
2) Put in new EDI edits in the claims module program in order to reject professional ICD9 claims

After the SCRUM Master and the Claims Department director developed the above Product Backlog table, the SCRUM Master took the developed PBIs and presented them to the entire SCRUM team for analysis and development of the Sprint Backlog table. In this scenario the team developed eight specific Sprint Backlog Items (SBIs) that would need to be accomplished in order to complete the two PBIs listed in Table 2. Table 3 shows the Sprint Backlog table with the eight agreed upon SBIs that were developed by the team and will be used as benchmarks to gauge overall progression of the project.

Table 5. Sprint Backlog

1) Make a new working storage variable for ICD-date to be checked in the program against the covers-from/covers thru date
2) New UB03855C error code to reject any ICD9 claims after 10/1 with error message as “Invalid ICD9 claims-cannot be filed”
3) Assign new error code UB03855C; create in test and request in production.
4) Check the covers-from-date and covers-thru-date against the ICD10-date to reject or accept the claim
5) Make a new working storage variable for ICD-date to be checked in the program against the earliest-from-date/latest-thru-date
6) New HB02222A error code to reject any ICD9 claims after 10/1 with error message as “Invalid ICD9 claims-cannot be filed”
7) Assign new error code HB02222A; create in test and request in production.
8) Check the earliest-from-date and latest-thru-date against the ICD10-date to reject or accept the claim

As stated before these two backlog developments happen in advance of the actual sprint beginning. This gives the Claims Department director, SCRUM Master, and Team

the necessary time to plan and develop the actual work items needed. It also allows for the proper preparation and pre-planning for resources, allowance for requirements and change modification mitigation step to be developed, as well as the pre-development of a list of potential risks and ways in which to mitigate them.

In this scenario the team found they could utilize SE LIs to properly prepare and pre-planning for resources, to develop steps for requirements and change modification mitigation, as well as to develop a list of potential risks and ways in which to mitigate them. The specific SE LIs utilized during the pre-sprint period are as follows: systems engineering staffing and skills, requirements validation, requirements verification, system definition change backlog, architecture, risk exposure and risk handling LIs. These SE LIs could also be utilized during the pre-sprint period to ensure that the developed PBIs and SBIs were appropriate and would successfully integrate into the overall effort. The below figures depict the team results of their utilization of these aforementioned SE LIs.

The first SE LI used was the systems engineering staffing and skills LI. This indicator showed the quantity of personnel assigned to the claims project as well as the skill and seniority mix and the time phasing of their application throughout the claims project's lifecycle. For this scenario there were three key personnel needed--a SCRUM master, Senior Developer, and a Developer. Figure 4, the systems engineering staffing and skills chart, showed the total time required for each of the four sprints broken down into individual components for each of the three key personnel that were used for the project.

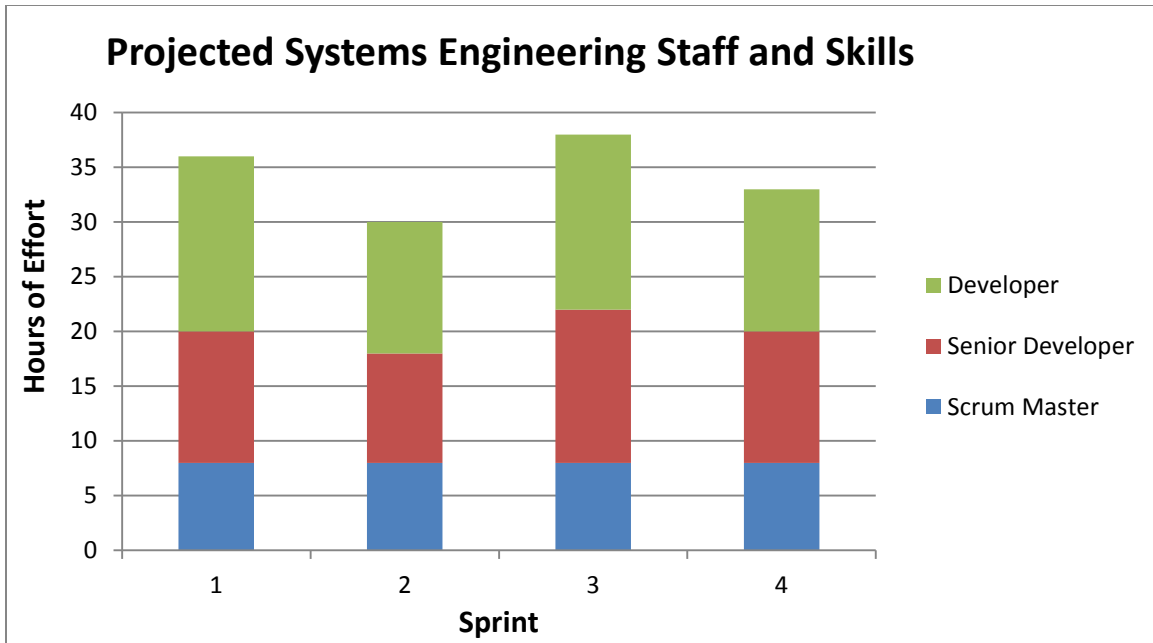


Figure 4. Projected Systems Engineering Staff and Skills

The value this gave the team was that it graphical represented each key contributor’s contribution to the overall project hourly budget. It also gave the SCRUM master a graphical representation of his allotted resource utilization and roughly which skill set is needed the most during the project. Finally, when this chart is combined with similar charts of other projects it will give the Claims director a better view of how his personnel are being used and whether or not a resource is being over or underutilized as well as whether or not there are any underlying skill deficiencies that are hidden within their department.

The next two SE LIs to be discussed are the requirements validation and requirements verification SE LIs. Figure 5 shows the team’s cumulative planned validations chart of their eight project requirements on a sprint by sprint basis. Figure 5 depicts the ideal trend line which shows the team has some understanding of the product

owner's requirement needs and that the overall planned project progress should go according to plan if the eight SBIs are able to be validated within the given schedule with little to no change. If the trend line were to dip or plateau, then it would show that there was a negative trend in regards to the understanding of the product owner's requirement needs and additional clarification is needed.

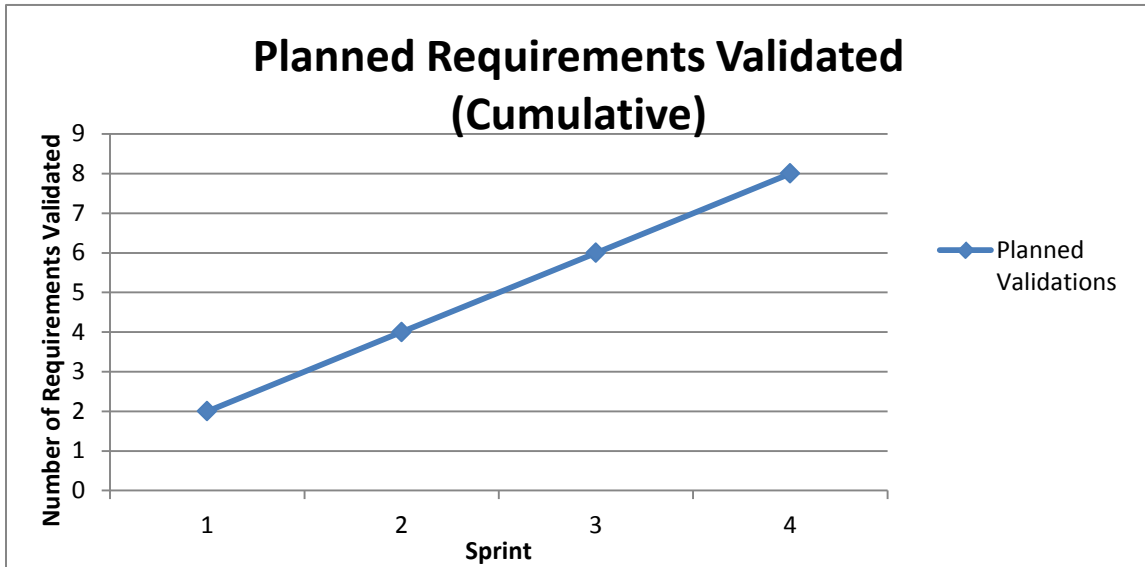


Figure 5. Planned Requirement Validation Trend

Figure 5 is important for the SCRUM master and Claims director because any deviations from this plan could impact the overall project design and planned project activities and thus adversely impact overall cost, schedule, and end result satisfaction.

However, whether or not this happens at this point in the project is yet to be seen and as such this SE LI will need to be re-visited each sprint to compare actual requirement validation with the planned requirement validations to ensure that the requirements are being validated as planned with no deviations.

Figure 6 shows the team's cumulative planned verification chart of their eight project requirements on a sprint by sprint basis. Figure 6 shows the ideal trend line and demonstrates that the team has some understanding of the project design needs and thus should be able to verify them within the given schedule with little to no risk. If the trend were to plateau or go downwards, then that would show that there was a problem understanding the design needs of the project and clarification of the needs of the project is needed.

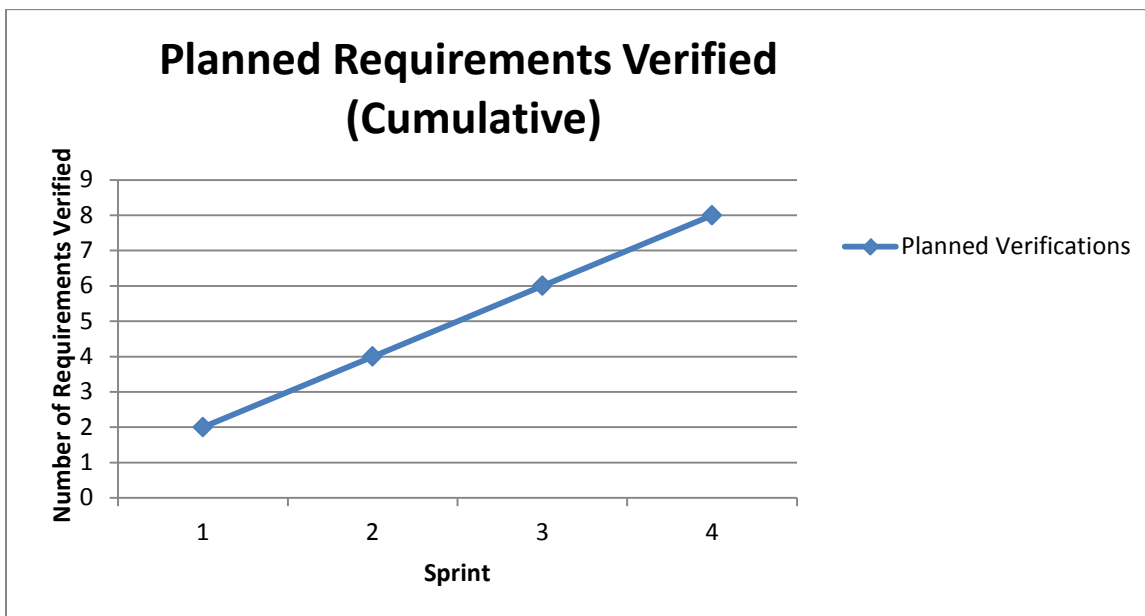


Figure 6. Planned Requirement Verification Trend

Just like with the requirement validation SE LI, Figure 6 is important for the SCRUM master and Claims director because any deviations found with the requirements validation could impact the overall project design and planned activities and thus adversely impact overall cost, schedule, and end result satisfaction.

However, as with requirements validation SE LI, whether or not this happens continuously throughout the project is yet to be seen. The requirements verification SE LI will need to be re-visited over the course of the project to compare actual requirement verifications with the planned requirement verifications, on a sprint by sprint basis, to ensure that the requirements are being verified on time as planned. Any variations could indicate changing customer needs and as such indicators of potential risk being injected into the project.

In order to ensure that change is properly accounted for so that no unidentified changes occur and adversely impact the project's requirement validation and verification steps, the team utilized the system definition change backlog SE LI and developed Figure 7 as a template to document the type of changes as they occur during the course of the project.

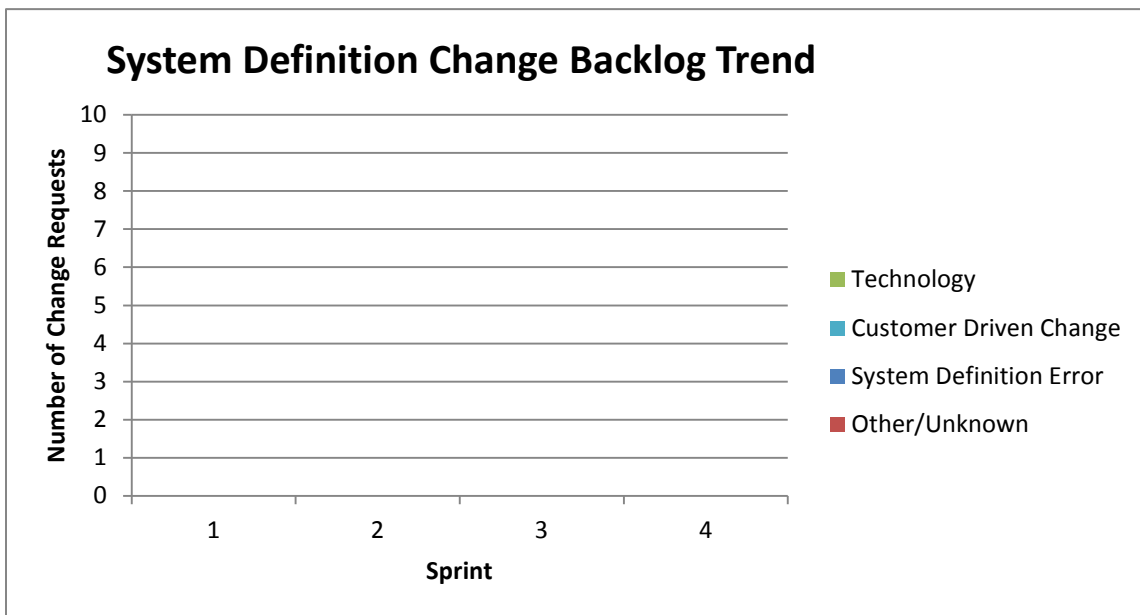


Figure 7. System Definition Change Backlog Trend Template

The template shown in Figure 7 is designed such that the team will be able to capture and document the total number of system changes that occur with the technology being developed, customer requirements, system definition or any other/unknown changes that could occur during the course of the project. If any changes were to occur during the project, the project team will be able to utilize this template to properly capture and document the number of changes and thereby ensure that they all are accounted for and not lost sprint to sprint.

The system definition change backlog is designed to be updated continuously over the course of the project and is in essence a “living backlog” that grows as changes occur if they occur. The ideal trend is a horizontal straight line trend as close to zero as possible depicting few changes. This backlog trend is important for the SCRUM master to monitor because when a project experiences too many changes, then there could be an adverse impact on the technical, cost and schedule baselines of the project as well as the overall scope and intent of the project. Naturally if this were to occur in this scenario then the Claims director must be notified immediately and the systems definition change process will have to be re-visited and modified in order for the project baselines to be corrected.

The next SE LI utilized by the team was the Architecture SE LI. This LI was used by the team during the pre-sprint phase to ensure that the team’s developmental tasks fit into the overall all-inclusive architecture design of the entire enterprise effort. Table 4 shows the enterprise effort that the team must keep in mind as they develop their individual portion. The basic idea behind the table is for a team to be able to properly gauge the level of maturity that the specific measures have on the enterprise effort where

versus where executive management wants the end state to be. The scores captured tell the team as well as executive management where their effort is. The higher the scores for each individual measure the better the architecture design of the enterprise effort.

Table 6. Architecture Base Measures
(Adapted from Systems Engineering Leading Indicators Guide, Version 2.0.)

Architecture Trend Base Measures					
Score	1	2	3	4	5
Base Measure 1: Commitment	• Enough resources exist	• Proper accountability for directing, overseeing, and approving the architecture design has been assigned	• Written and approved organization policy exists for architecture design development	• Written and approved organization policy exists for architecture design maintenance	• Written and approved organization policy exists for IT investment compliance with architecture design
Base Measure 2: Capability	• An office responsible for architecture design development and maintenance has been established	• An architecture design review board exists at the project level	• Architecture design processes undergo independent verification and validation	• An architecture design review board exists at the business unit	• Architecture design is critical component of enterprise business investment management process
Base Measure 3: Plans and Products	• Architecture design is being developed using an established framework and methodology	• Architecture design plans address the “as-is” and “to-be” architecture design	• Key stakeholder business drivers are properly documented	• Process for identifying, managing, and closing gaps between “as-is” and “to-be” is accounted for	• The relationships between the “as-is,” transition, and “to-be,” to investment planning are accounted for
Base Measure 4: Performance Metrics	• Basic Performance metrics exist	• Architecture design plans call for developing metrics for measuring progress	• Progress against architecture design plans is measured and reported	• Compliance with architecture design is measured and reported	• Performance measures are defined and linked to the service and technical portions of the architecture design
Base Measure 5: Strategic Direction Strategic Direction	• Architecture design demonstrates stakeholder buy-in is documented.	• Architecture design demonstrates management structure.	• Architecture design defines change and risk management strategy	• Architecture design defines a communications plan	• Architecture design demonstrates process for managing changes
Base Measure 6: Interoperability	• Interoperability standards are defined conceptually	• Interoperability standards are defined at the business function level	• Interoperability standards are related to business functions	• Business functions are aligned to components and services at the enterprise level	• Interoperability and sharing of information established as a backbone of the architecture design
Base Measure 7: Data	• Data architecture design is only broadly defined	• Data relationships and interdependencies defined at a conceptual level	• Data relationships and interdependencies defined at an operational level	• A well-defined data approach to business processes and mission established	• A well-defined approach to integrating data established
Base Measure 8: Security	• Security standards are conceptually defined	• Security standards align to a conceptual reference model	• Security standards align to a technical reference model	• Security standards are tightly defined and are presented as part of transition planning	• Security standards are tightly defined and are presented as part of investment planning

It is important to note, that whether or not each individual measure is included in the actual team portion of the enterprise effort development, each individual measure must be considered by the team if the team’s portion of the effort is to be successfully integrated into the overall enterprise effort. The SCRUM master must understand this and recognize the underlying interdependencies brought forward and captured so that the team can keep them in mind as they develop. These underlying interdependencies are

also important for the Claims director to know because they must know how their project(s) interact and affect the enterprise effort so that could communicate any issues effectively if they arise.

Finally, the team utilized the risk exposure SE LI to identify and approximate how much risk could be present for each sprint. To do this the team first developed a risk table that showed what the team perceived as the potential risks that could occur during the course of each sprint of the project as well as the probability and impact of each identified risk. The team then ranked each risk individually based on their potential and impact to the overall project. Finally, to complete the table the team listed the mitigation steps they would take if any one of the identified risks listed were to occur. Table 5 depicts the team’s results.

Table 7. Identified Risks

S/ No.	Risks	Probability (1=Low, 5=High)	Impact (1=Low, 5=High)	Risk Category	Ranking of Risks	Risk Reduction Measures
1	Management disregards project communications and meetings	2	4	Low/High	9	Keep Management informed of activities and progress on a regular basis
2	Uncontrolled changes and continuous growth of scope	3	5	Medium/High	3	Utilize a Change Control Board
3	Inaccurate estimates	3	4	Medium/Medium	8	Use past estimates as a baseline for
4	Missing activities from scope definition	2	5	Low/High	10	Work with Stakeholders to ensure scope is properly defined
5	A large number of change requests	3	5	Medium/High	4	Utilize a Change Control Board
6	Lack of critical tools	3	5	Medium/High	5	Procure any needed tools in a timely fashion once identified
7	Inaccurate change priorities	4	4	Medium/Medium	6	Utilize a Change Control Board
8	Stakeholders become disengaged	4	3	Medium/Medium	7	Keep Stakeholders informed of activities and progress on a regular basis
9	Resource performance issues	3	5	Medium/High	1	Require additional training
10	Technology components aren't interoperable	3	5	Medium/High	2	Ensure proper and appropriate testing is accomplished

After developing the risk table, the team then developed the risk exposure trend figure. In order to develop this figure (Figure 8), the team first divided took each risk identified and labeled them either a stakeholder risk or a team risk. Then the team separated each risk based on their risk probability/impact risk category (Low, Medium, and High). After they separated the risk by probability/impact risk category they assessed the likelihood of each risk occurring in each of the four sprints. The ideal risk exposure profiles would depict a trend line with little to no risks at any probability.

Once the probability/impact of the identified risks was determined by the team, the team was able to develop Figure 8. These profiles are important to all key stakeholders because they show the amount of risks exist and the probability/impact of each type of risk. By knowing this knowledge key stakeholders and decision makers will be able to focus their attention with the highest probability/impact risks and implement mitigation strategies that could save valuable team time and effort.

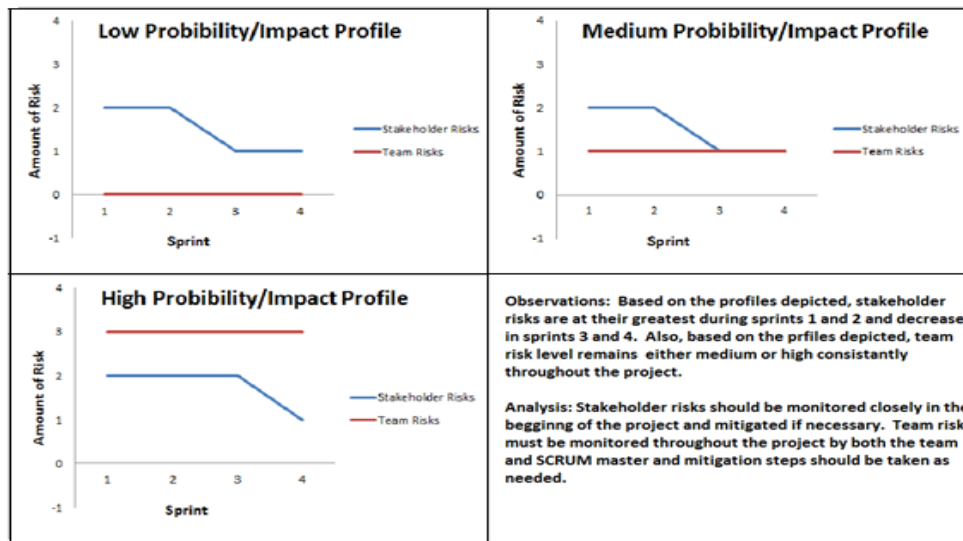


Figure 8. Risk Exposure Trends
(Adapted from Systems Engineering Leading Indicators Guide, Version 2.0.)

Based on the project risk exposure profiles above, the team will be able to utilize the risk handling SE LI to determine the best way to track any identified risk, from open to close, should they occur during the course of any of the project's sprints. The ideal trend for this SE LI is a downward linear line where the amount of risk actions open equals the amount of risk actions closed by the end of the project. Figure 9 depicts what the team generated prior to the project beginning. The team was able to use for the above risk tools effectively and were able to identify a minimal amount of risks that could potentially occur during each sprint and take appropriate actions to mitigate them fully without having any overdue issues.

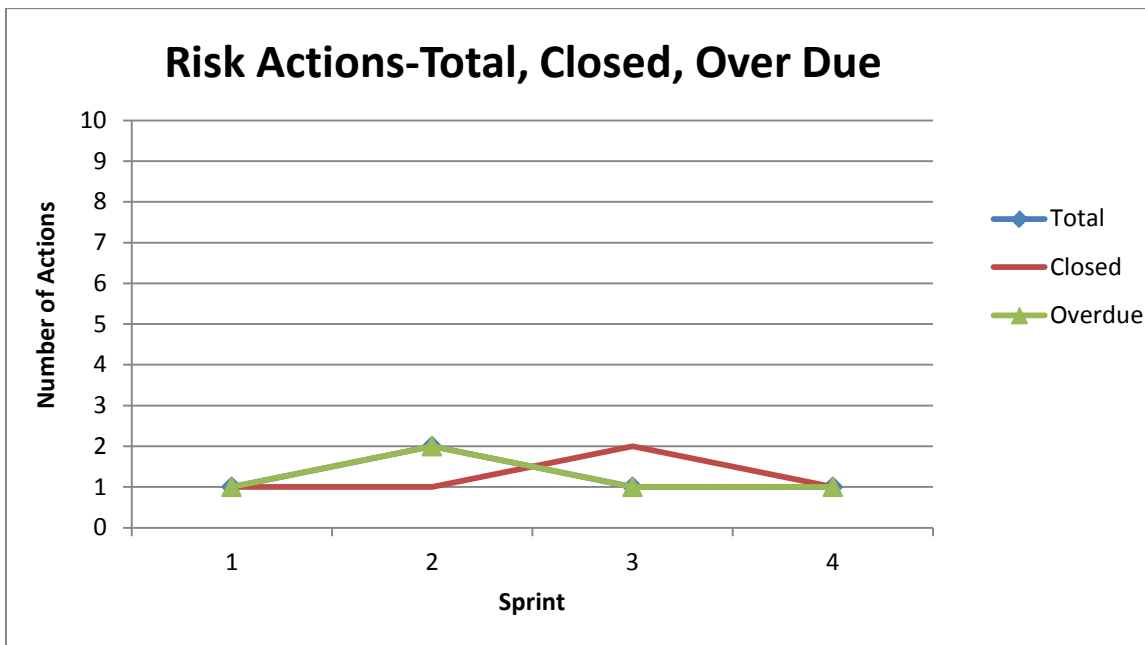


Figure 9. Risk Handling Tracker Template

As seen from the above risk artifacts developed by the team, the knowledge gained and tools developed by the by the team gives the team the insight and forward

thinking capability that they can use during the course of the project's individual sprints. It also gives the team the knowledge and tools they need to quickly resolve any identified risk(s) without having to take time for unnecessary team discussions about the method(s) of identifying and handling the identified risk(s). The SCRUM master must know how to properly utilize these artifacts and keep them up to date as new risks and mitigation steps are identified and employed by the team. The SCRUM master must also report any changes to these artifacts to the Claims director as soon as possible in order to keep them properly informed on the health status of the overall project.

As it can be seen, the utilization of the mentioned SE LIs enhanced the overall pre-sprint process taken by the team and provided the team with valuable knowledge and tools that can be used throughout the course of each individual project sprint in order to ensure that the overall project is progressing forward in a productive and efficient manner. However, it must be noted the utilization of these SE LIs is not included in the overall SCRUM process. As such, there could be substantial unnecessary risk to the project in using them, specifically the use of valuable limited time and resources necessary in order to effectively develop the tools and insights each of the mentioned SE LIs can provide. This is especially true if the team and/or SCRUM master are not accustomed to developing and using them.

However, in this case, the team and SCRUM master were familiar with SE LIs and therefore were able to save time by utilizing these SE LIs upfront in the beginning stages of the project. As such the team avoided any undue risk up front by ensuring that the work items developed were appropriate and able to be successfully integrated into the overall program, saving valuable time and effort. This utilization also ensured that each

sprint begin with as much insight planning as possible to ensure overall sprint success.

This revelation was recorded by the SCRUM Master and reported to the Claims director.

During Sprint-Sprint Burn Down

Once the PBIs and SBIs were developed and all the pre-sprint activities described above had been completed and artifacts developed, the team began development of the project. For the development effort, the team determined the best way to handle the work would be to break the SBI development up into four 2-week sprints of eight developmental days each (the first day was designated as a planning day with no development and the last day was designated the retrospective day also with no development as well). This format utilized by the team would allow for the maximum number of developmental days for each sprint while also allowing for non-intrusive sprint planning and closing ceremonies at the beginning and end of the sprint respectively. This is important because the planning and closing ceremonies for each sprint are crucial if the project's sprints are to begin and end appropriately.

The team also decided to focus on approximately two SBIs per each sprint. This would allow each developer to take on an appropriate level of work that meets the overall objectives of each sprint's as well as allows for continuous progress towards completion of the project while still retaining some level of capacity for other company projects if needed. This notion of "retaining capacity" is important because upper management may have a high priority project come down from executive leadership and the team must have available capacity to be able to flex and successfully fulfill that project's requirements in conjunction with the currently discussed project. This aspect is

important because rarely does a project team just focus their efforts on one project at a time.

The burn down progress for each sprint is a ratio of the team's committed sprint hours remaining versus the team's committed sprint hours completed. The figures specifically show the rate of completed work vs. remaining work for each developer at any given point during the sprint. The goal is to complete all work committed to before the official end of the sprint.

In addition to the sprint burn down figures, the team utilized the requirements validation, requirements verification, and interfaces SE LIs before the end of each sprint to ensure that the SBIs being developed, as well as the SBIs still pending development, were still relevant and still meet the Claims department director's overall intent and goals for the project. The team's findings on each of these SE LIs are also shown in separate figures below.

Sprint 1 Results:

Sprint 1 was the official start of the overall project effort. For this sprint the team decided to commit to approximately 28 hours of developmental work. Figure 10 splits these hours between the senior developer and developer so that each developer's burn-down rate can be seen in relation to the overall burn down of the sprint. The ideal trend is a downward linear trend where all hours committed to are reduced to zero by the end of the sprint.

As can be seen in Figure 10, the developmental team had a steady burn down rate throughout the course of the sprint with no additional work items having to be committed to and completed. The results of Sprint 1's burn down trend were noted by the

SCRUM master and presented to the Claims director.

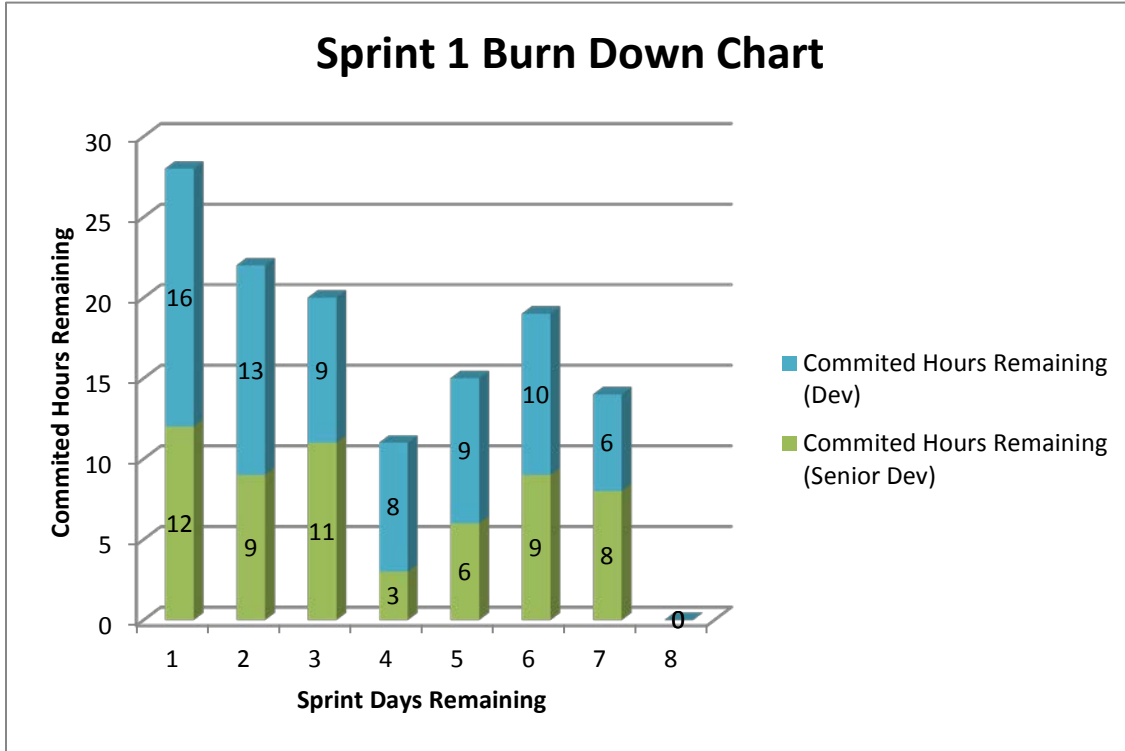


Figure 10. Sprint 1 Burn Down Chart

It is important to mention that the first and last days of the sprint were reserved by the team for the planning and closure ceremonies specifically so that there would be no interruptions during the course of development of the following sprints. The team found that this was the optimal format to utilize so that there could be seamless transitions in between each sprint.

Figure 11 depicts Sprint 1's planned vs. actual requirements validated trend. This figure is a cumulative trend and will eventually show the progress of the requirements validation trend rate across all four sprints. It is designed to gauge whether or not the product owner's requirements are still valid and properly understood by a project team.

Any changes or deviations in requirements could negatively affect the team's overall understanding of the product owner's requirements and could adversely affect the development occurring during a sprint. The ideal trend is an upward linear trend where the planned for validations are completed each sprint. In this case, the team planned for two validations and was able to complete them both in the sprint.

As such, for Sprint 1, Figure 11 shows that the Claims director's requirements were found to be mostly still valid, only half of the requirements were completely understood by the team. This negative trend poses a potential adverse impact to the overall project design, however since it happened in the first sprint the team has time to recover from any impacts that occur. Also, based on Figure 11, the Sprint 1 activities that had corresponding impacts to overall project technical, cost and schedule baselines, as well as overall product owner satisfaction, were found to still within acceptable ranges, despite the negative trend that occurred. This is evident because there were no deviations from the planned requirements. In other words, since the requirements were consistent throughout Sprint 1, the baselines did not change. The results of Sprint 1's planned vs. actual requirements validated trend was also noted by the SCRUM master and presented to the Claims director.

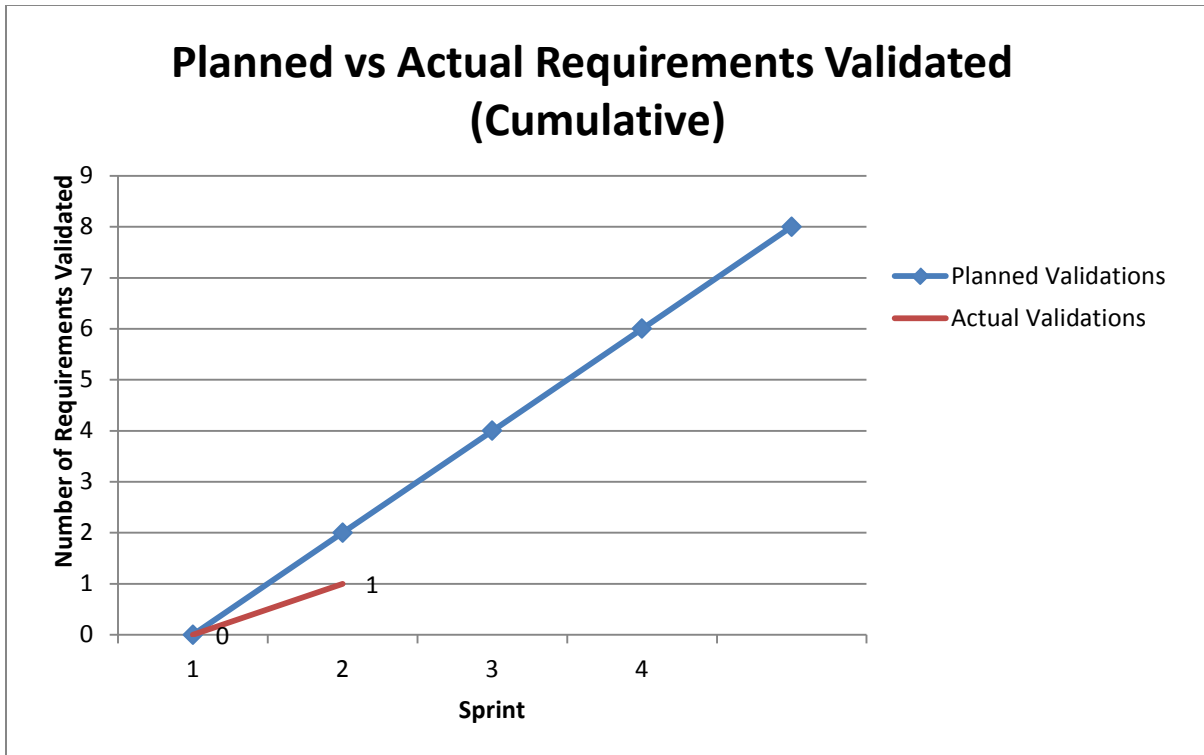


Figure 11. Sprint 1 Planned vs Actual Requirements Validated Trend

Figure 12 depicts Sprint 1's planned vs. actual requirements verified trend. Much like the requirements validation trend described above, the requirements verified trend is a cumulative trend and that will eventually show the progress of the requirements verification trend rate across all four sprints. It is designed to gauge whether or not the overall project design is still valid and still meets the product owner's requirements. Any changes or deviations in the project design could negatively affect the team's overall understanding of the product owner's requirements and could adversely affect the development occurring during a sprint. The ideal trend is an upward linear trend where the planned for validations are completed each sprint. In this case, the team planned for two validations and was able to complete them both in the sprint.

As can be seen in Figure 12, the overall design of the project changed slightly but still is moving in a direction that meets the specified requirements outlined at the beginning of the project by the Claims director. The trend present poses a slight negative impact to the overall project design, but since it happened early in the project there is an opportunity for the team to adjust and correct the design. However, the corresponding impacts to technical, cost & schedule baseline and overall customer satisfaction did not change and thus were still within acceptable ranges. The results of sprint 1's planned vs. actual requirements verified trend was also noted by the SCRUM master and presented to the Claims director.

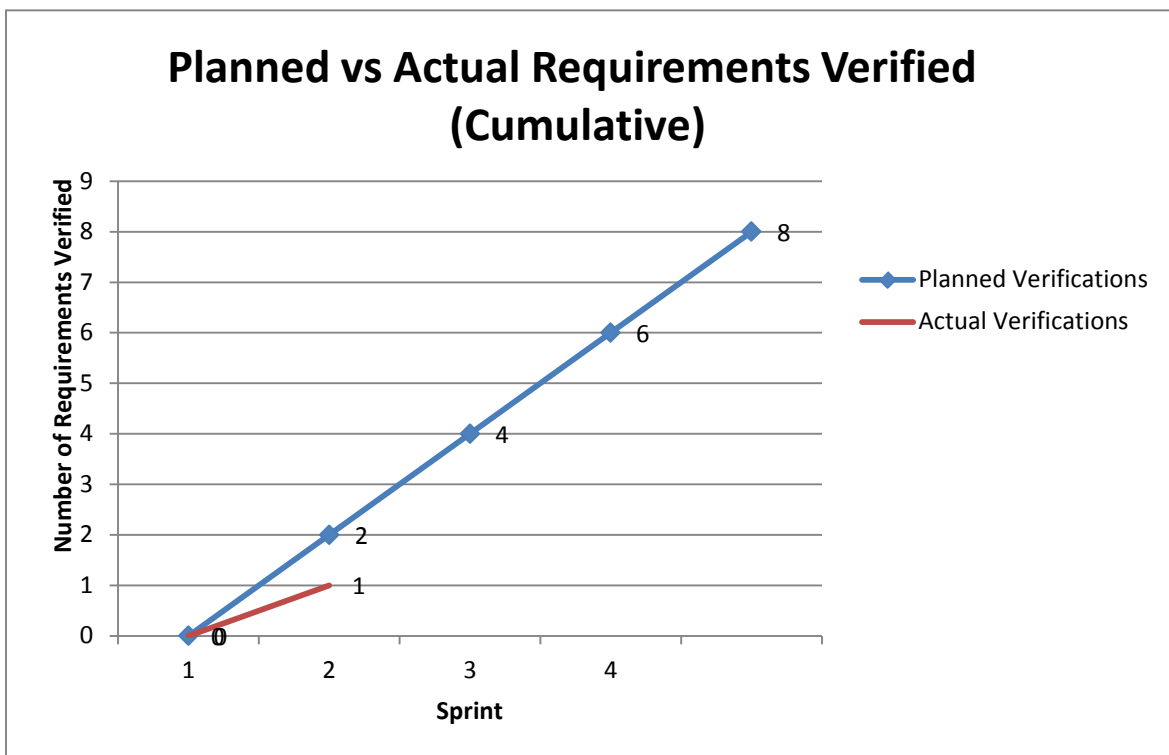


Figure 12. Sprint 1 Planned vs Actual Requirements Verified Trend

The final metric produced by the team for Sprint 1 was the interface specification trend or requirements growth trend. The purpose of this trend is to show the requirement closure rate of overall project requirements so the team could evaluate the trends in the growth, change, completeness and correctness of the definition of the project's requirements. The ideal trend line for this SE LI is again a linear upward trend where the amount of planned requirements equals the amount of actual requirements completed. Also, the projected requirements trend line should remain equal to the planned for requirements and not increase. See Figure 13.

Figure 13 shows the project's requirements were reduced during Sprint 1. This is happened due to the requirements not being completely understood by the team. This could pose an adverse impact to the overall system architecture and design which could pose negative technical, cost and schedule baseline impacts if not mitigated and should be of concern for the project team as well as the product owner.

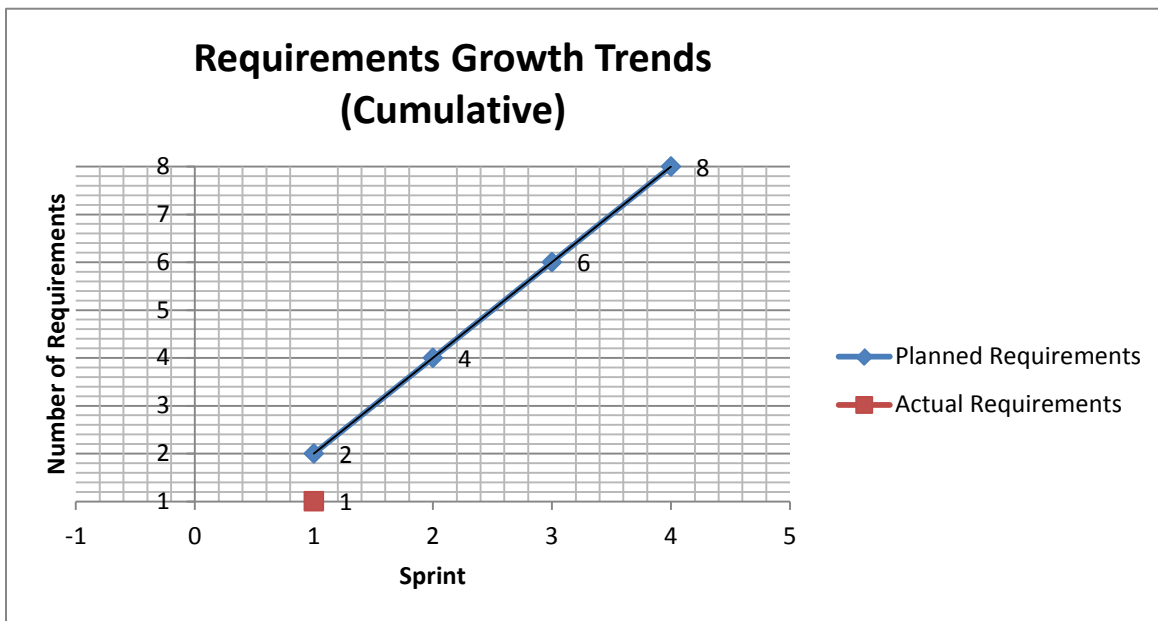


Figure 13. Sprint 1 Requirements Growth Trend

Again, much like the requirements validation and requirements verification trends described in Figures 11 and 12, the interface trend is a cumulative trend and will eventually show whether or not the project's requirements remained over the course of the project. The results of Sprint 1's interface specification trend was also noted by the SCRUM master and presented to the Claims director.

Sprint 2 Results:

The second sprint marked the halfway point for the overall project. For this sprint the team decided to commit to approximately 22 hours of developmental work. There is no particular reason as to why the team decided to commit to less work in this Sprint 2 than Sprint 1 other than the sprint backlog items committed to by the team developers required less work than sprint 1's committed backlog items.

This aspect is important to note because in SCRUM the team is allowed to decide how much work they wish to commit to in any given sprint. This aspect is difficult for some organizations to comprehend because it runs counterintuitive to most business structure models, especially in highly hierarchal authoritative entities, public or private, and can be a point of frustration for some in executive or upper managers, or individuals finding themselves in similar positions. This is because of the feeling of lack of control due to the lack of input they have on the work being committed to by the team.

Figure 14 depicts the burn down rate for the team during Sprint 2. Again, like Sprint 1, the figure splits the hours between the senior developer and developer so that

each individual's burn-down rate can be seen in relation to the overall burn down of the sprint.

As can be seen in Figure 14, the developmental team again had a steady burn down rate throughout the course of the sprint, in line with the expected trend, with no additional work items having to be committed to and completed for this effort. Again, the first and last days of Sprint 2 were reserved by the team for planning and closure activities so that there would be no interruptions to the schedules of the two remaining sprints. The results of Sprint 2's burn down trend were noted by the SCRUM master and presented to the Claims director.

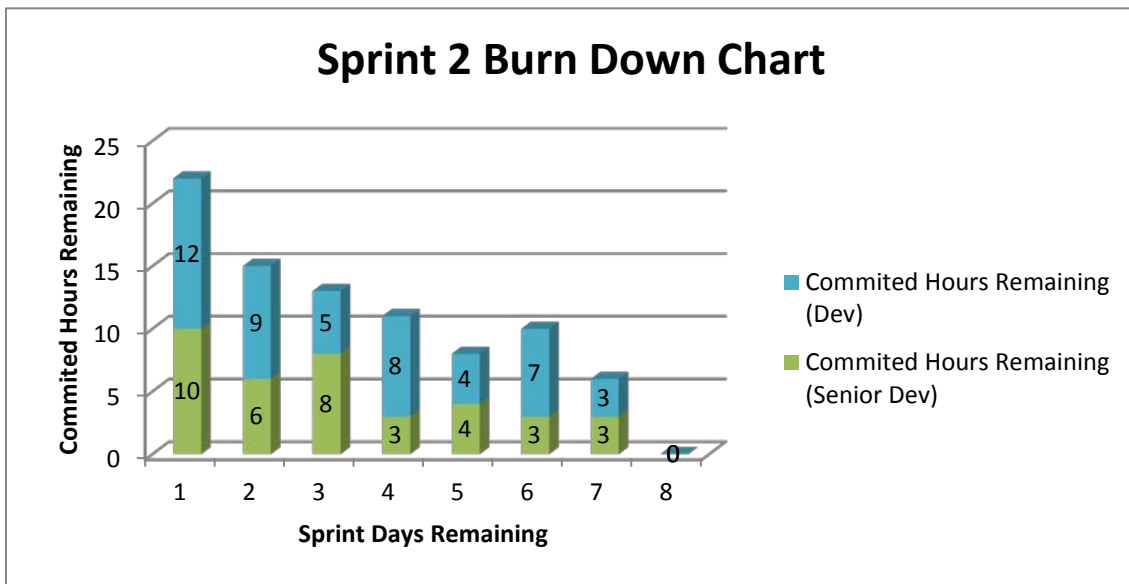


Figure 14. Sprint 2 Burn Down Chart

The next figure (Figure 15) graphs Sprint 2's planned vs. actual requirements validated trend. As can be seen, the team better understood the requirements and was able to adjust and get back in track in Sprint 2 through two consecutive sprints. No adverse trends that would pose a negative impact to the overall project were found during

this sprint. Also, the activity with corresponding impacts to technical, cost & schedule baseline and overall customer satisfaction were found to still be within acceptable ranges as well.

These results are important to note because it tells the team that after completing fifty percent of the overall project they are still on track and headed in the correct direction. It also tells the team that as of this point in the life cycle of the project no adverse effects have been experienced and thus no major deviations have occurred. The results of Sprint 2's planned vs. actual requirements validated trend was also noted by the SCRUM master and presented to the Claims director.

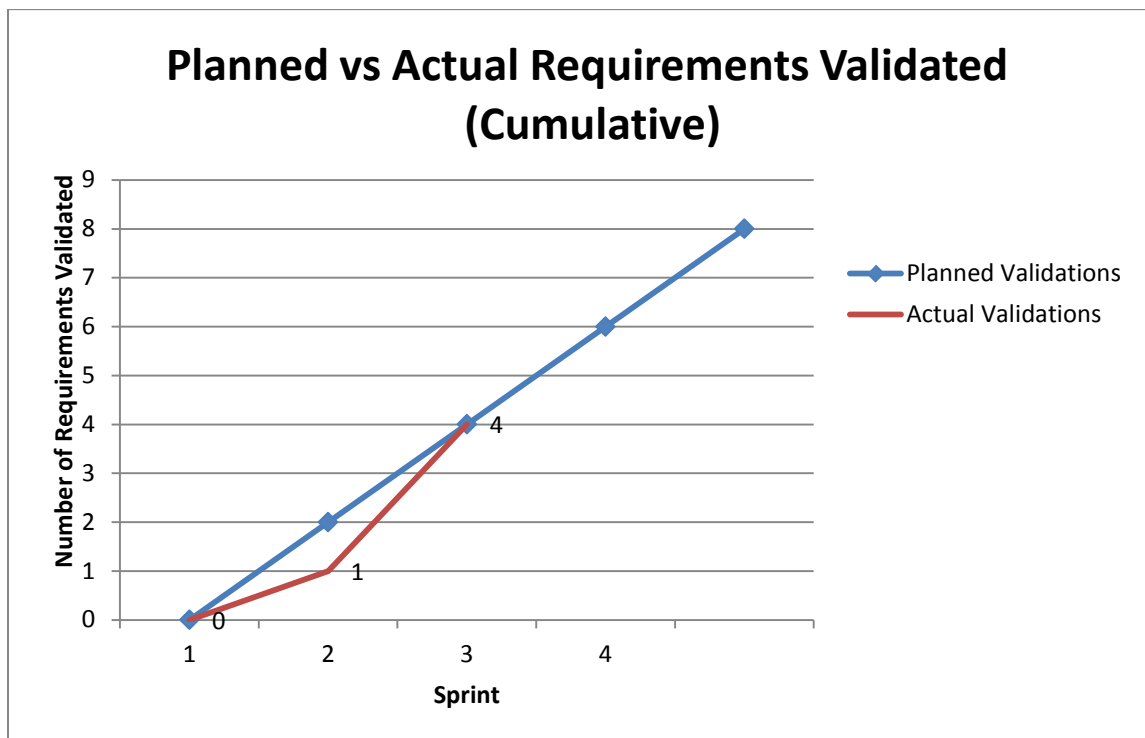


Figure 15. Sprint 2 Planned vs Actual Requirements Validated

Figure 16 depicts Sprint 2's planned vs. actual requirements verified trend. As shown, the team was able to adjust from Sprint 1 and get back on track. The overall design of the project did not change and thus still meets the specified requirements outlined at the beginning of the project through two consecutive sprints. Also, no adverse trends that would pose a negative impact to the overall project design were found during this sprint as well. As a result the activity with corresponding impacts to technical, cost & schedule baseline and overall customer satisfaction were found to still be within acceptable ranges.

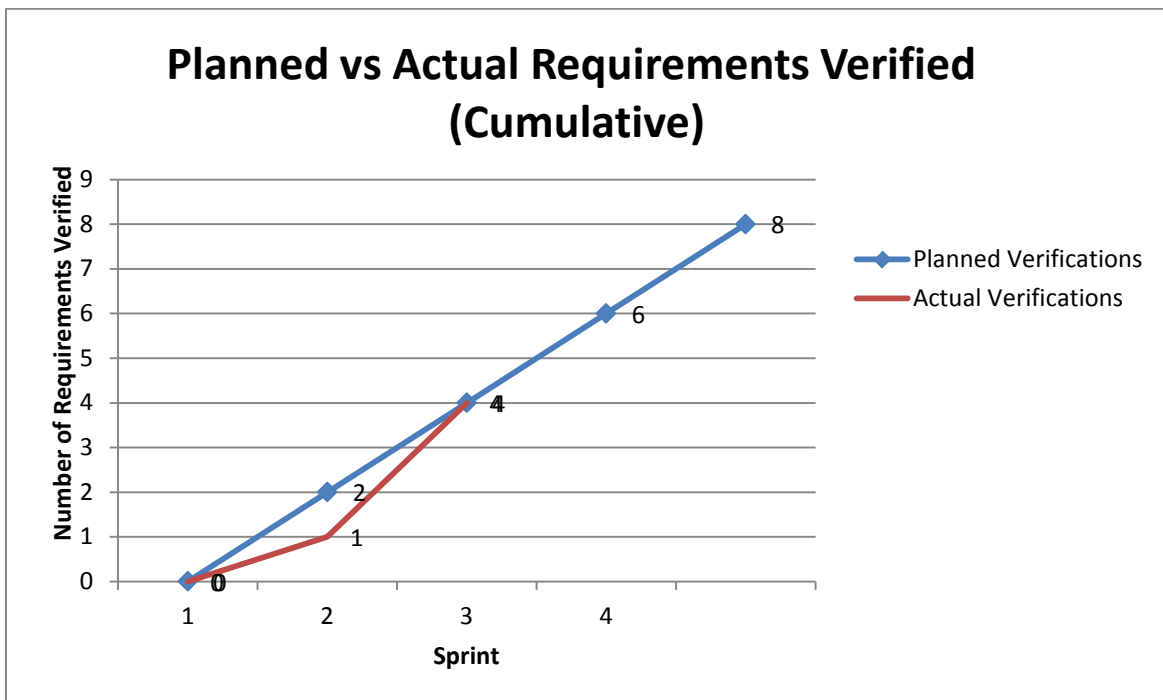


Figure 16. Sprint 2 Planned vs Actual Requirements Verified

Again, much like the planned vs. actual requirements validated trend, the results of the planned vs. actual requirements verified trend are important to note because it tells the team that after completing fifty percent of the overall project they were still on track

and headed in the correct direction with no adverse effects having been experienced. As a result there were no major deviations with the overall project design and the exhibited trend line was the expected trend line. The results of Sprint 2's planned vs. actual requirements verified trend was noted by the SCRUM master and presented to the Claims director as well.

The final figure produced by the team for Sprint 2 was the interface specification trend (Figure 17). Again, the team was able to adjust from Sprint 1 and was able to get back on track with completing the necessary requirements. This is good because any additional negative change in growth, completeness, or correctness of the project's requirements would pose adverse impact to system architecture and design, any of which could pose negative technical, cost and schedule impacts which, at this point in the lifecycle of the project, could be detrimental to the success of the overall project effort.

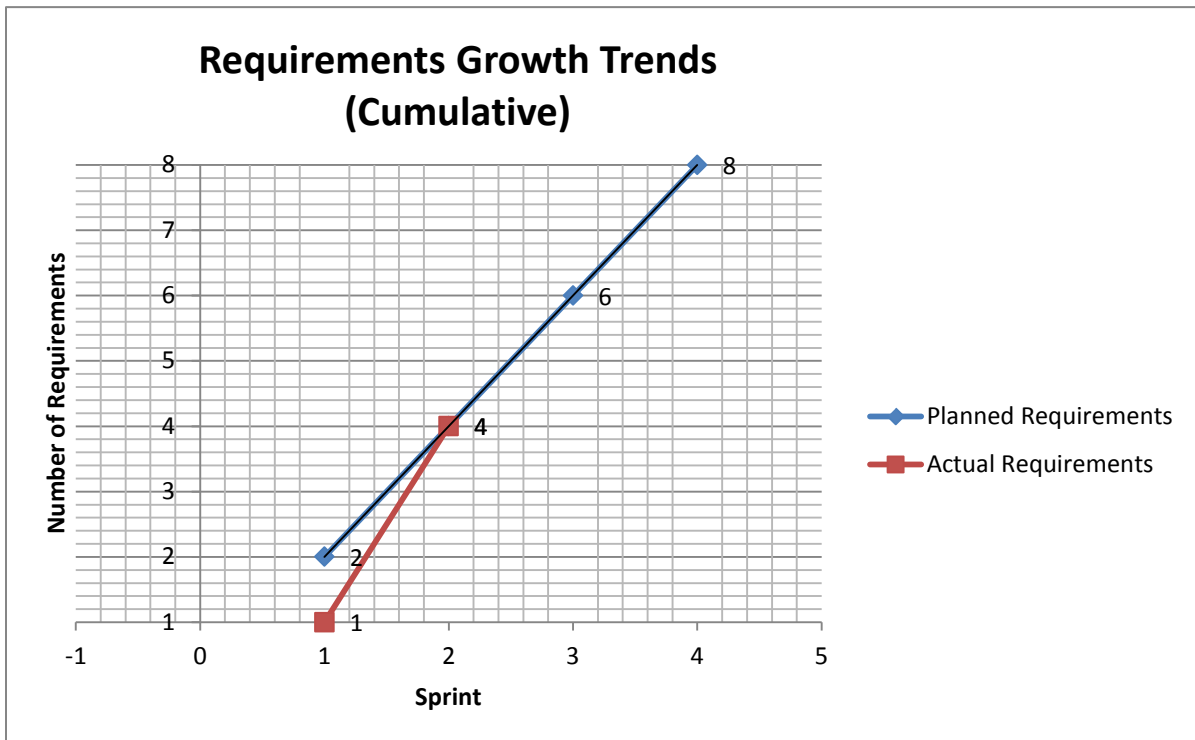


Figure 17. Sprint 2 Requirements Growth Trend

Again, the results of the interface trend are important to note because they tell the team that after completing fifty percent of the overall project effort they are still on track and headed in the correct direction. It also tells the team that no adverse effects have been experienced, in regards to requirement completion, and thus no major deviations have occurred with the project project's requirements or their expected development and completion schedule. This is shown again by the fact that the exhibited trend line was the one that was expected.

This aspect is important because it gives the product owner reasonable expectations as to when to expect all the requirements to be developed and completed thus allowing them to plan for future efforts more effectively. The results of Sprint 2's interface specification trend too was noted by the SCRUM master and presented to the Claims director.

Sprint 3 Results:

For this sprint the team decided to commit to approximately 30 hours of total developmental work. This is more work than either of their previous two sprints which indicates the team felt that more effort was needed to complete their assigned tasks than in the previous two sprints. Again, there is no particular reason as to why the team decided to commit to more work in this sprint other than the sprint backlog items committed to by the team was perceived by the developers to require more work than Sprint 1's and Sprint 2's committed backlog items.

Figure 18 depicts the burn down rate for the team for Sprint 3. Again, like Sprint 1 and Sprint 2, the figure splits the hours between the senior developer and developer so that each individual's burn-down rate can be seen in relation to the overall burn down of the sprint.

Also, just like the team experienced in Sprint 1 and Sprint 2, it can be seen in the figure that the developmental team had a steady burn down rate, in line with the expected trend line, throughout the course of this sprint with no additional work items having to be completed for this effort. This is good because it shows that the developmental team has matured to a performing level and thus has a firm grasp on what exactly the effort of work that needs to be completed.

Finally, the first and last days of Sprint 3 were again reserved by the team for planning and closure activities so that there would be no interruptions of development schedule of the final sprint. The results of Sprint 3's burn down trend was noted by the SCRUM master and presented to the Claims director.

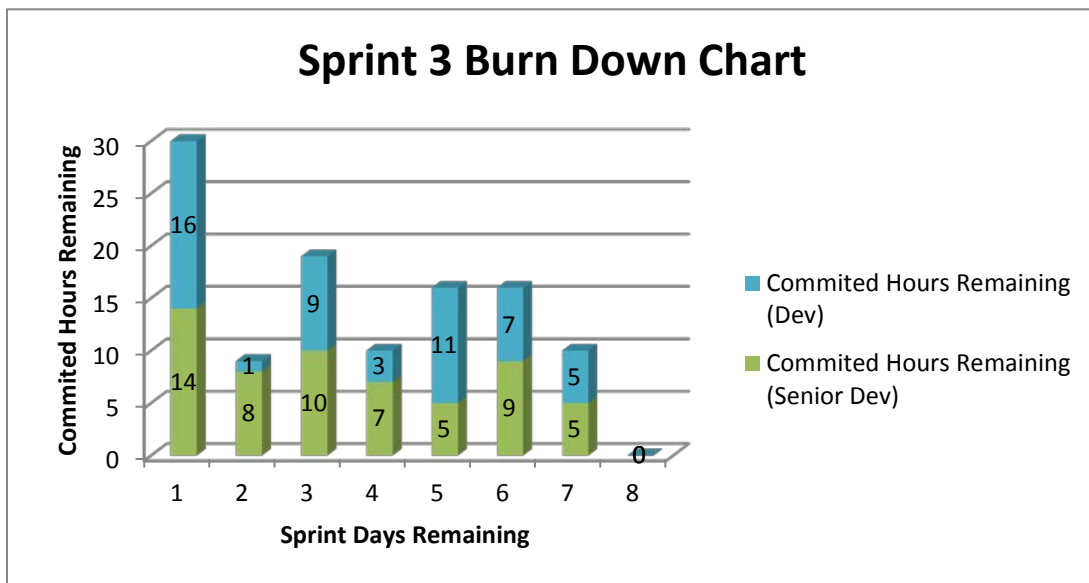


Figure 18. Sprint 3 Burn Down Chart

Sprint 3’s planned vs. actual requirements validated trend is provided in Figure 19. Again, the Claims director’s requirements were found to be still valid and properly understood by the team through the third sprint. The team was able to validate an additional requirement and thus a positive trend line was generated. No negative impact to the overall project requirements were found during this sprint and the trend line exhibited was better than expected. Finally, the activity with corresponding impacts to technical, cost and schedule baselines and overall customer satisfaction were found to be still within acceptable ranges in this sprint as well.

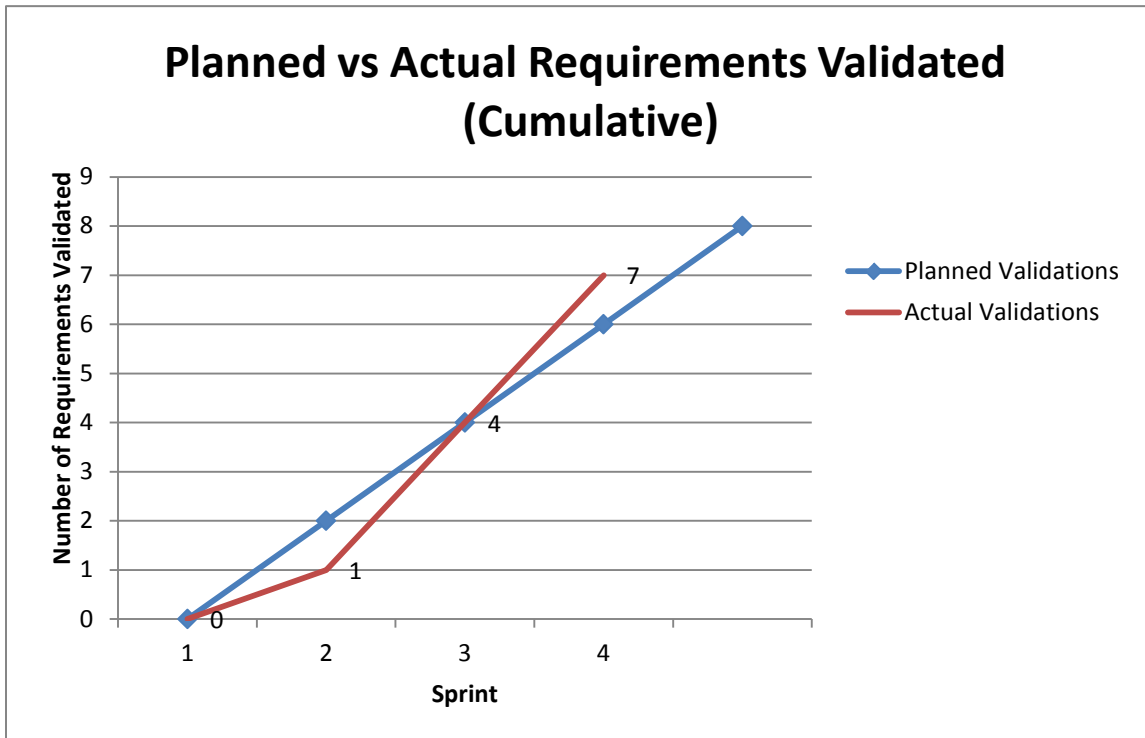


Figure 19. Sprint 3 Planned vs Actual Requirements Validated

These results are again important to note because it tells the team that after completing three sprints of the project and with only one sprint to go they are still on track and headed in the correct direction towards completing the project successfully. It also tells the team that as of this point in the life cycle of the project with no major adverse effects being experienced and thus no major deviations having occurred. The results of Sprint 3's planned vs. actual requirements validated trend were also noted by the SCRUM master and presented to the Claims director.

Next, Figure 20 depicts Sprint 3's planned vs. actual requirements verified trend. The overall design of the project still meets the specified requirements outlined at the beginning of the project through three consecutive sprints. The team was able to verify an additional requirement and thus a positive trend line was generated. No negative impact to the overall project requirements were found during this sprint and the trend line exhibited was better than expected. Finally, the activity with corresponding impacts to technical, cost and schedule baselines and overall customer satisfaction were found to still be within acceptable ranges.

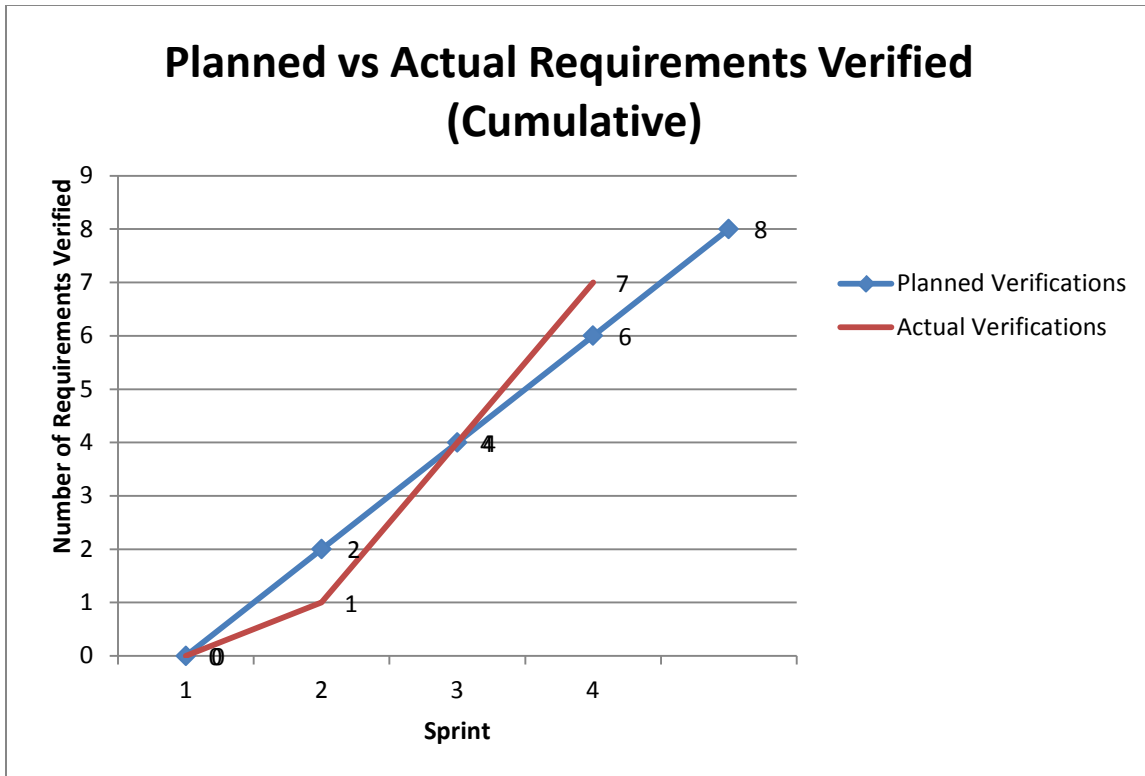


Figure 20. Sprint 3 Planned vs Actual Requirements Verified

Again, much like the planned vs. actual requirements validated trend, the results of the planned vs. actual requirements verified trend are important to note because it tells the team that after completing three of the four sprints of the project they are still on track and headed in the correct direction, as far as the design goes, with no major adverse effects having been experienced and the trend line occurring as expected. As a result, no major deviations have occurred. The results of Sprint 3’s planned vs. actual requirements verified trend was noted by the SCRUM master and presented to the Claims director as well.

The interface specification trend was the final metric measures by the team for Sprint 3. See Figure 21. The project’s requirements continued to remain constant and

did not change over the course of three consecutive sprints. However, the trend line exhibited was different than expected. There was a slight change in growth due to an additional requirement being added. This was determined to be acceptable because it made up for the earlier dropped requirement. The completeness, or correctness of the project's requirements did not exhibit an adverse impact to system architecture and design, thus did not pose any technical, cost and schedule impacts which at this point could be detrimental to the success of the overall project.

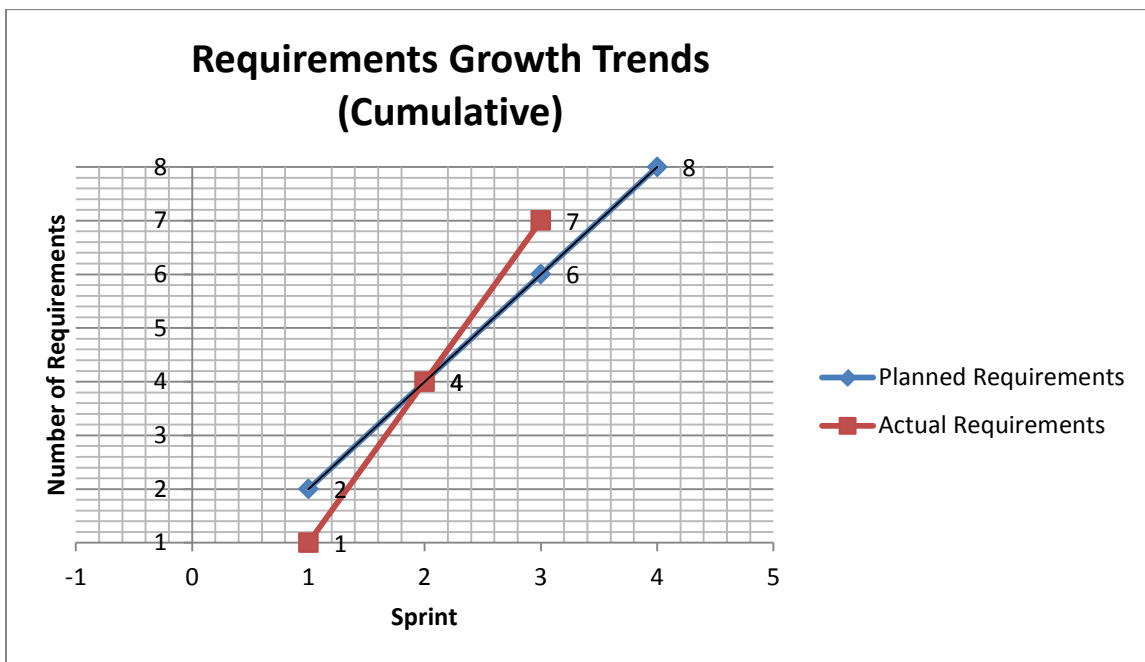


Figure 21. Sprint 3 Requirements Growth Trend

Again, the results of the interface trend are important to note because it tells the team that after completing three of the four project sprints they are still on track and headed in the correct direction with no major adverse effects have been experienced and thus no major deviations have occurred with the project's requirements. The results of

sprint 3's interface specification trend too was noted by the SCRUM master and presented to the Claims director.

Sprint 4 Results:

The fourth sprint marked the end of the project. For this sprint the team decided to commit to approximately 25 hours of developmental work. Again, there is no particular reason as to why the team decided to commit to less work in this sprint other than the sprint backlog items committed to by the team required less work than Sprint 3's committed backlog items.

The burn down rate for the team for Sprint 4 is shown in Figure 22. Again, like the previous three sprints, the figure splits the hours between the senior developer and developer. This is so that each individual's burn-down rate can be seen in relation to the overall burn down of the sprint can be seen visually by the SCRUM master.

Also, the developmental team again had a steady burn down rate throughout the course of this sprint, with the expected trend line being observed, with no additional work items having to be completed for this effort. This has been the case for the previous three sprints and thus shows the SCRUM master that this pproject team's has attained a high level of maturity and is well defined as a team. This is good to note because this information is invaluable to the SCRUM master and upper management when they are planning resources for future projects.

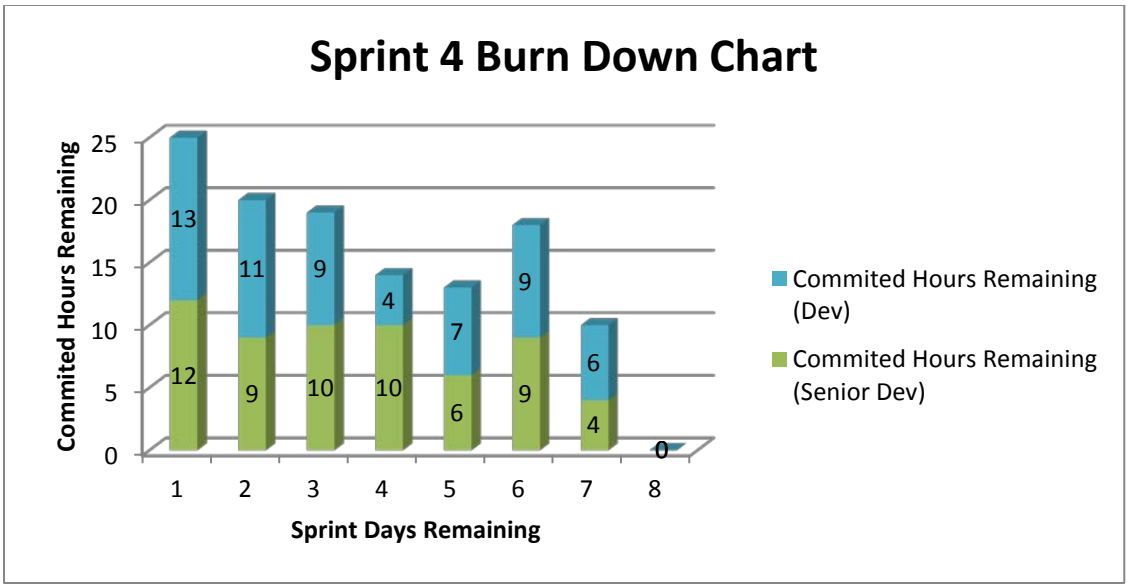


Figure 22. Sprint 4 Burn Down Chart

Similar to the previous three sprints, the first and last days of Sprint 4 were reserved by the team for planning and closure activities so that there would be no interruptions during the course of development as well as no interruptions during the closing ceremonies of the overall project by the team and the Claim director. The results of Sprint 4’s burn down trend were presented to the Claims director.

Figure 23 depicts Sprint 4’s planned vs. actual requirements validated trend. As can be seen in the figure, the Claims director’s requirements were still found to be still valid and properly understood by the team throughout the final sprint. This is good because it shows that the requirements were clearly defined and presented to the team in a fashion that was easily comprehended and well understood.

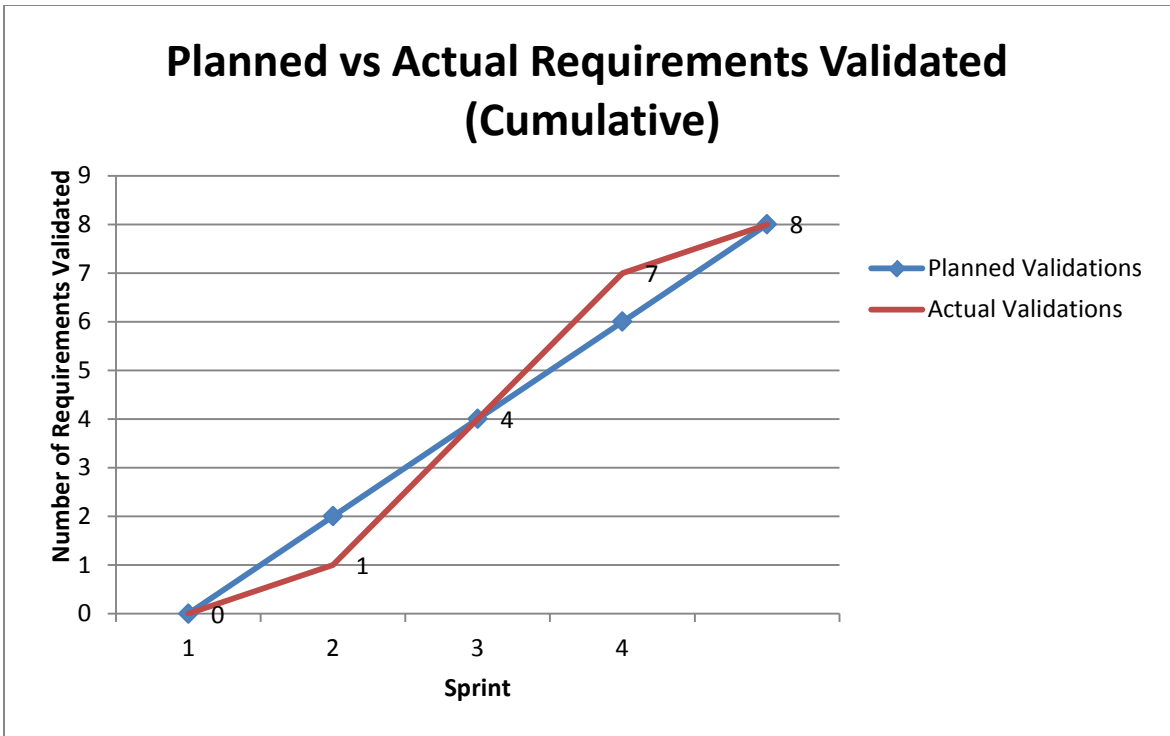


Figure 23. Sprint 4 Planned vs Actual Requirements Validated

No adverse trends that would pose a negative impact to the overall project requirements were found during this sprint and the trend line observed was expected. Finally, the activities corresponding to technical, cost & schedule baselines were within acceptable ranges and thru overall customer satisfaction was high. This relative consistency indicates that every aspect of the project remained within the planned baselines and as a result customer satisfaction remained constant throughout the entire project.

These results are again important to note because it tells the team that after completing all four sprints of the project they were able to remain relatively on track through the completion of the project. It also tells the team that there were no major adverse effects experienced and thus no major deviations occurred during the entire

project. The results of Sprint 4's planned vs. actual requirements validated trend were also noted by the SCRUM master and presented to the Claims director.

Sprint 4's planned vs. actual requirements verified trend is shown in Figure 24. The overall design of the project was met and the specified requirements outlined at the beginning of the project were accomplished throughout all four sprints. This was due in large by the well define project design that the team received. Also the team's maturity and understanding of the project design allowed for this trend. No adverse trends that would have posed a negative impact to the overall project design were found during this sprint and the observed trend line was expected. Finally, the activities with corresponding impacts to technical, cost and schedule baseline and overall customer satisfaction were found to still be within acceptable ranges.

Again, much like the planned vs. actual requirements validated trend, the results of the planned vs. actual requirements verified trend are important to note because it tells the team that after completing all of the project's sprints of the project they remained on track and in the correct direction during the last sprint, as far as the design goes, with no major adverse effects having been experienced and thus no major deviations having occurred.

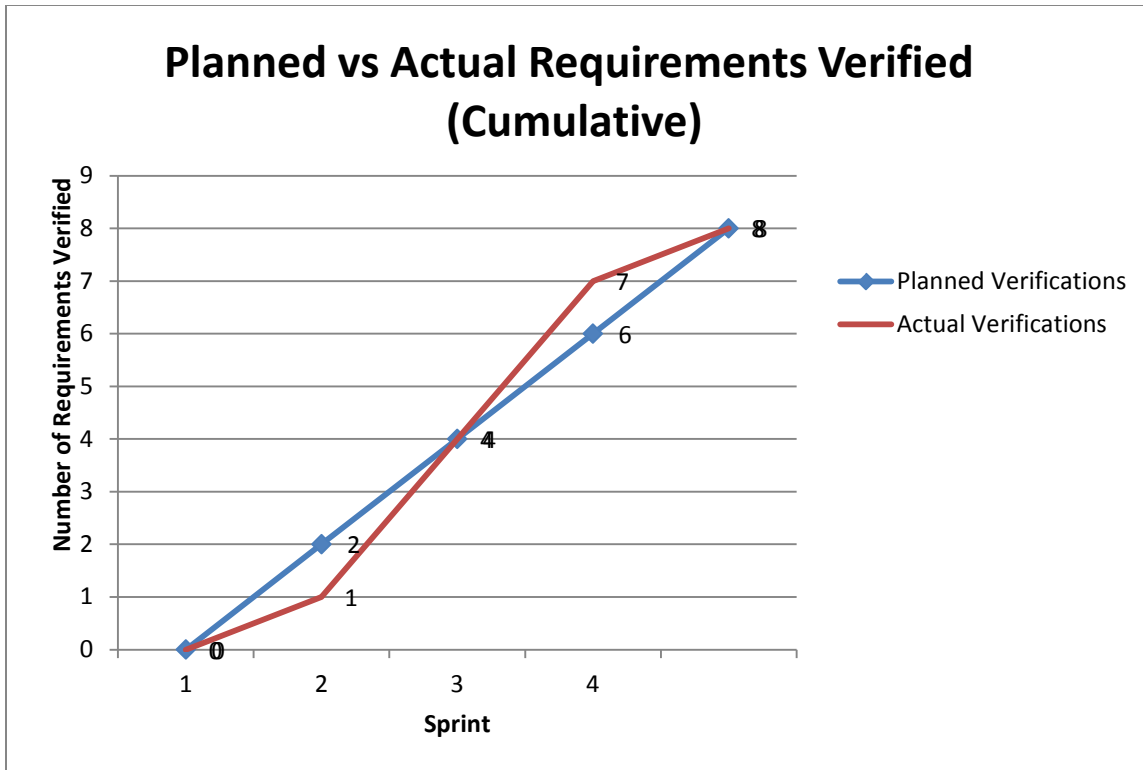


Figure 24. Sprint 4 Planned vs Actual Requirements Verified

The interface specification trend is final figure (Figure 25) produced by the team for Sprint 4. As can be seen, the project’s requirements continued to remain relatively constant throughout the lifecycle of the project and did not change. This is exhibited by the expected trend line.

This is good because any additional change in growth, completeness, or correctness of the project’s requirements could have had an adverse impact to system architecture and design which could have posed significant technical, cost and/or schedule impacts, which at this point could be detrimental to the success of the overall project and caused the developmental effort to be prolonged un-necessarily.

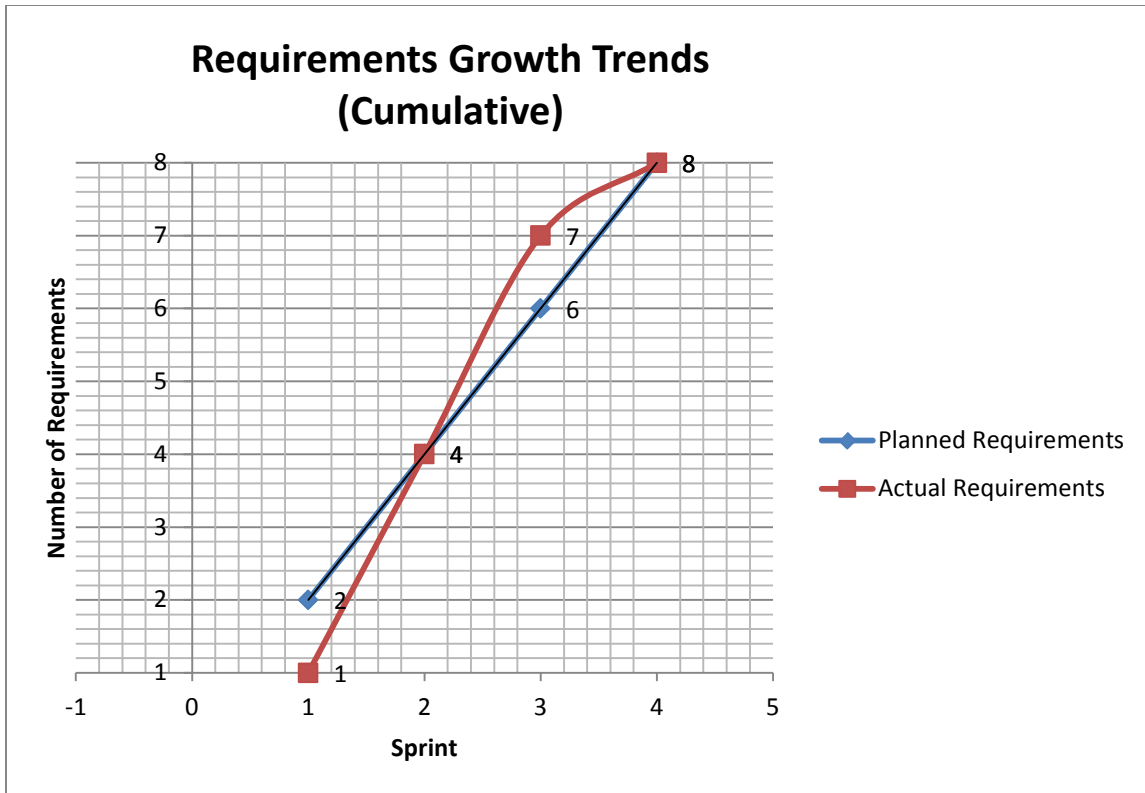


Figure 25. Sprint 4 Requirements Growth Trend

Again, the results of the interface trend are important to note because it tells the team that after completing all of the project’s sprints they remained on track and in the correct direction with no adverse effects having been experienced and thus no major deviations occurring with the project’s requirements. This is again important to note because if a team is to be truly successfully then the requirements must be clearly defined upfront and not changed during the course of a project.

Additional Results:

In addition to utilizing the above SCRUM Agile techniques and SE LIs to monitor the overall health and progress of the project, the team utilized the technology maturity

and technical measurement SE LIs to determine how much technical effort had progressed over the lifetime of the project. The ideal trend line for the technology maturity LI is an upward linear trend which shows an increase in the maturity of the technology. The ideal trend line for the technology measurement LI is a downward linear trend which depicts project progression moving towards “green” which stand for complete.

The team’s findings on the overall progression of technology maturity and measurement are presented in Figure 26 and 27.

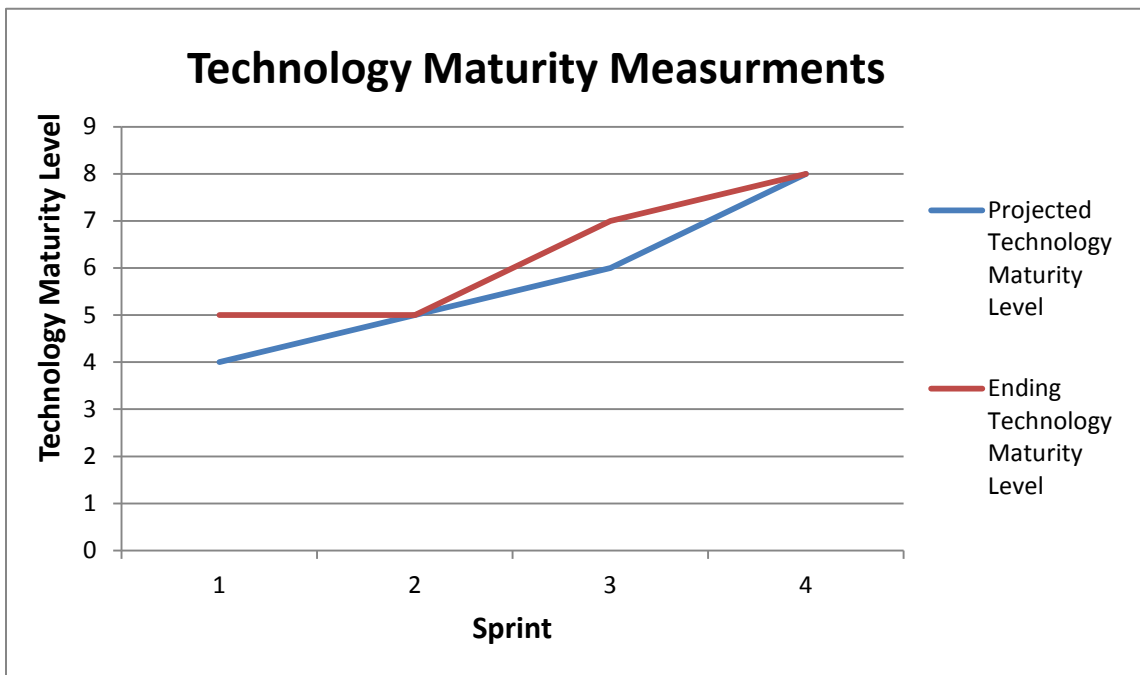


Figure 26. Technology Maturity Measurements
(Adapted from Systems Engineering Leading Indicators Guide, Version 2.0.)

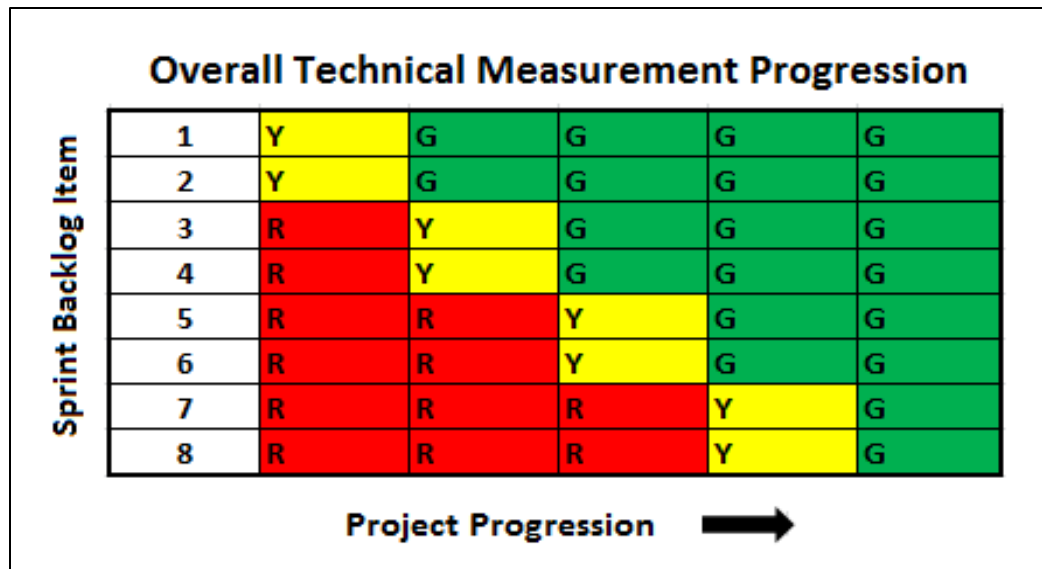


Figure 27. Overall Technical Measurement Progression
(Adapted from Systems Engineering Leading Indicators Guide, Version 2.0.)

The technical maturity and overall technical effort progressed at a steady rate throughout the course of the project. These figures also show that by the end of each sprint the team had added significant technical value to the overall enterprise system and did so systematically and methodically throughout the entire life of the project. These results were noted by the SCRUM master and presented to the Claims director as well.

Finally, the compliance SE LI was utilized to ensure that all work completed met all requirements set forth and the defect and error SE LI was utilized by the team to document and discovered code defects (i.e. bugs) that needed to be fixed in order to meet compliance. The ideal trend line for the compliance LI is a downward trend line which shows the compliance issues identified being completed. For the Defect and error LI the ideal trend line would be a steady linear trend line with little to no growth in defects and errors.

In order to track any compliance issues of the work completed as well as any defects and errors that might have occurred during the lifetime of the project, the team employed the following two graphs (Figures 28 and 29)

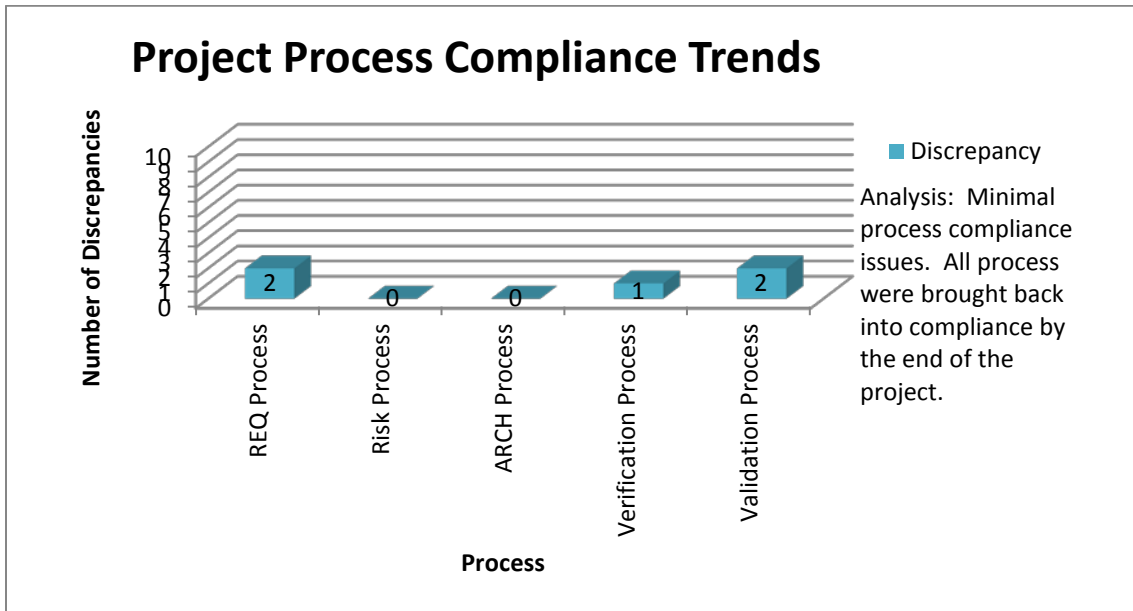


Figure 28. Project Compliance

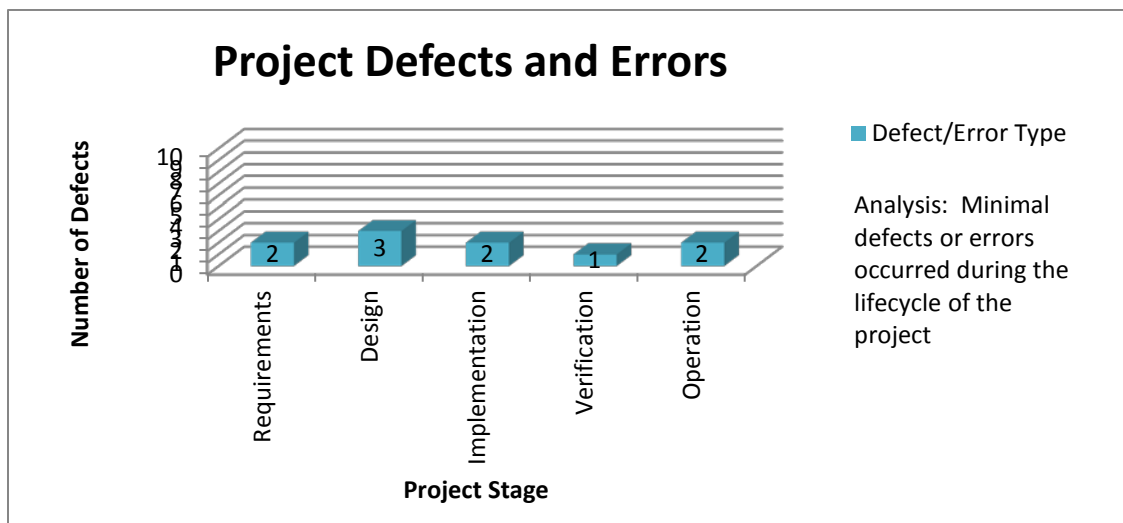


Figure 29. Project Defects and Errors

The graphs show that the team experienced five compliance issues with the work completed; demonstrating that the team stayed relatively within the regulatory requirements set forth by executive management, as well as a minimal number of defects or errors in their code that was developed. The trend lines observed for both were ideal. These observances were recorded by the SCRUM Master and reported to the Claims Department director.

During Sprint-Release Burn-Up

As the sprints finished, the SCRUM Master kept track of the overall effort's progression in the release burn up chart. The ideal trend for this SCRUM metric is an upward linear trend depicting the overall completeness of the SBL items.

Based on the chart, the SCRUM Master was able to determine that the effort progressed in a steady fashion and thereby remained on track to complete on time. Figure 30 is the team's aggregate release burn up chart. It covers the progress of the effort across all four sprints.

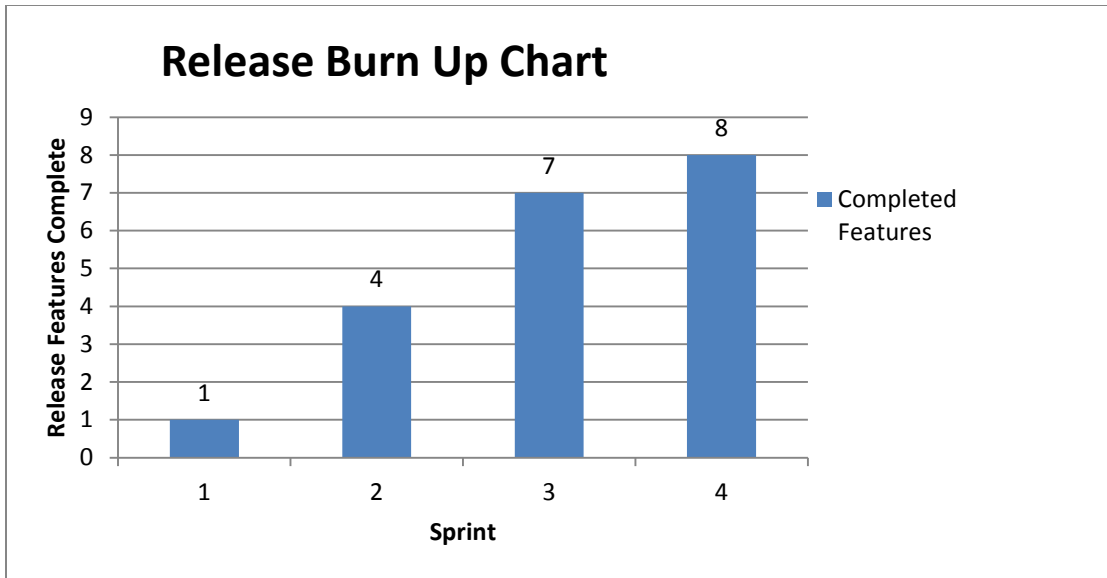


Figure 30. Release Burn Up Chart

Unfortunately the SCRUM Master discovered the release burn up chart only provided the total number of completed SBIs. It did not provide the holistic picture that the SCRUM Master needed to ensure that the effort was progressing as the Claims Department director required. As a result of this insight, the SCRUM Master combined the release burn up chart with the schedule and cost pressure SE LI to gauge the impact of the schedule and cost challenges to the execution of the project. Also, the SCRUM master utilized the schedule and cost pressure SE LI to develop the percentage differences between project estimates and any contracted values. To accomplish this, the SCRUM master developed Figure 31.

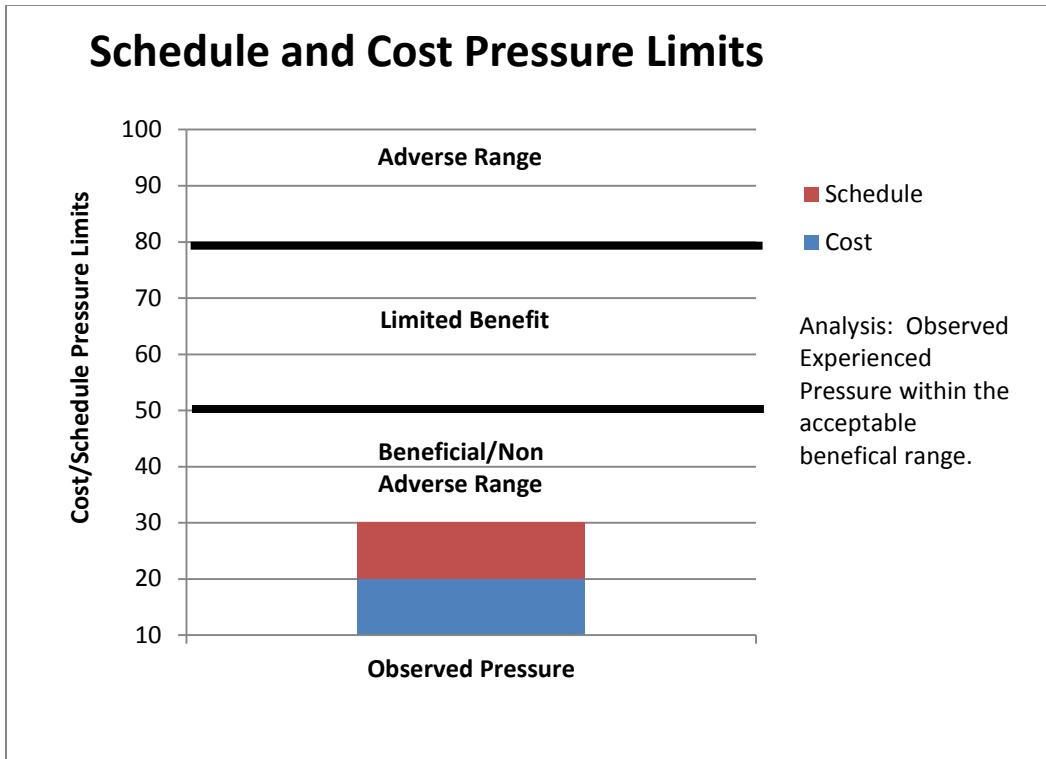


Figure 31. Cost Schedule Pressure Limits Diagram
(Adapted from Systems Engineering Leading Indicators Guide, Version 2.0.)

As it can be seen, the overall schedule cost pressure of the overall project was well within the beneficial range. In other words, the percentage differences between the project estimates of schedule and cost gathered before the beginning of the project and the actual values of schedule and cost were within acceptable limits.

In addition to this the SCRUM master took the findings of the release burn up chart and the schedule and cost pressure figure and combined them with the previously generated technology maturity, technical measurement, and defect and error data to get the holistic picture of the results of the effort. This complete package generated by the SCRUM master was reported to the Claims Department director and will be retained by the SCRUM master for future reference.

Post Sprint-Delivered Business Value Analysis

After the sprints concluded the SCRUM master and Claims Department director discussed the business value that was gained from this effort to ensure it was what was expected. The first item discussed was the work product rejections. Figure 32 was developed by the team utilizing the work product SE LI. The ideal trend for this LI is a steady low count trend reflecting the fact that none of the work completed was rejected by the product owner.

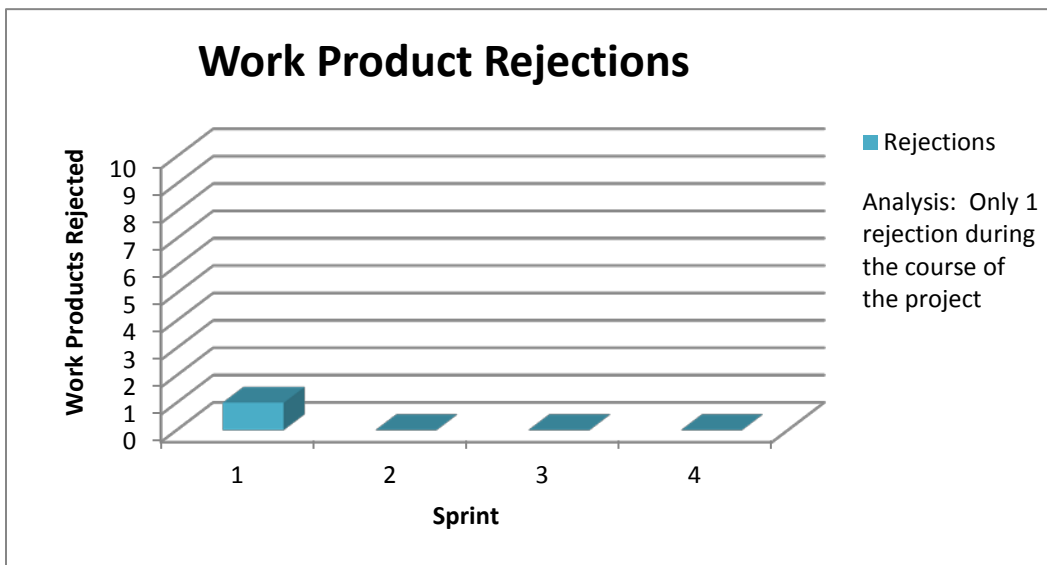


Figure 32. Work Product Rejections

This figure illustrates the total number of work product rejections on a sprint by sprint basis. As can be seen there was only one rejection of the work product developed during the entire course of the project by the product owner, which happened early during the project. This is not out of the ordinary for most teams because during this point in a project's lifecycle a team usually is still trying to understand what needs to be done during a project and thus may produce work not acceptable to the product owner.

However, in the case of this project, only one rejection indicates that overall high quality work or as well as excellent document review process each of which positively impacted cost, schedule and customer satisfaction.

Also, the SCRUM mast discussed the sprint backlog items that were closed and how many were closed per sprint. This was accomplished by utilizing the review action closure SE LI. The ideal trend line for this LI is an upward linear trend showing that all of the SBL items were closed during the course of the project. Figure 33 was developed by the team to present this information.

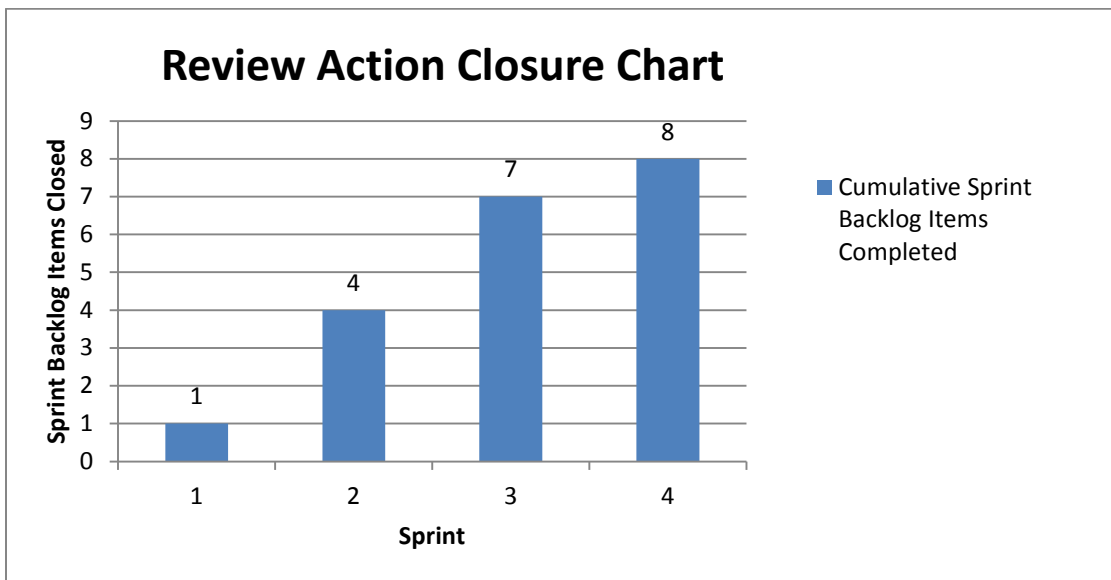


Figure 33. Review Action Closure

In addition to the discussions about the work product rejections and the review action closure, there were discussions the data previously provided by the aforementioned SE LIs of schedule and cost pressure, technology maturity, technical measurement, and defect and error were analyzed. Using these SE LIs, as well as the perceptions of the

Claims Department director, it was determined the gained business value outweighed the cost and impact of the effort.

It is important to note that the data provided by the SE LIs of schedule and cost pressure, technology maturity, technical measurement, and defect and error specifically, were critical in arriving at the resulting conclusion. This is because they provide actual data that the Claims Department director could use objectively vs. their subjective perceptions alone.

Post Sprint-Average Team Velocity Analysis

Team velocity for this effort was mentored and recorded at the end of each sprint. Using the equation

$$T_{Avg\ Velocity} = \frac{\sum Sprint\ Completed\ Hours}{Total\ Number\ of\ Sprints}$$

it was found that the Team's average velocity was 26.25 hours. Utilizing the equation

$$V_{Rate} = \frac{Completed\ Sprint\ Hours\ or\ Story\ Points}{Total\ Committed\ Sprint\ Hours\ or\ Story\ Points}$$

The SCRUM Master was able to determine that the team had a Velocity rate of one which indicated that the team was able to complete all committed hours within each sprint.

Knowing this the SCRUM Master, as well as the Claims Department director, can better plan for future projects. The SCRUM Master also realized during this scenario that when this information when combined with the SE LIs of systems engineering staffing and skills, risk exposure, risk handling, and work product that they would be able to estimate what resources will be needed for similar or larger project efforts with a degree of certainty using actual project historical data. They also realized that they would be

able to better gauge the potential risk exposure of taking on projects with larger hour/story point requirements as well as what measures should be taken to handle those risks due to having developed a listed of commonly occurring risks with IT type projects. These realizations were also documented and presented to the Claims Department director.

Post Sprint-Team Retrospective

Finally, at the end of the effort the team conducted a team a team retrospective. During this ceremony the team discussed what worked well as what could be improved upon for the next project. The Claims Department director was not present at this meeting (chiefly because Product Owners are generally not present in the SCRUM ceremony) therefore the SCRUM Master kept a log of all the observed positive and negative events as well as the remedies to the negative events.

During the team retrospective the team found that they could not only capture and document lessons learned but could also properly close the effort out cleanly and begin work on the next project seamlessly. The retrospective notes were documented and presented to the Claims Department director as well for their reference and assay.

Investigative Questions Answered

This chapter applied the various metrics to a SCRUM scenario. Recall the research questions were:

1. Can all the recognized SE LIs be utilized in modern, SCRUM Agile, iterative software development or is there a limited set that can be used?

It was found during the scenario that all but two of the recognized SE LIs could be utilized in the modern SCRUM Agile, iterative software development process. The two SE LIs that could not be used were: facility and equipment availability and system affordability. However, it should be noted that these two LIs should not be completely overlooked because they do play an important role outside the SCRUM Agile, iterative software development process.

Also, it was found that some of the SE LIs were able to be utilized more than others. However, it was observed that the impact of the less utilized SE LIs was not directly related to the frequency of use of those SE LIs.

2. How well does the recognized SE LIs qualitatively compare to SCRUM Agile metrics?

It was found that the recognized SE LIs do qualitatively compare to the SCRUM Agile metrics. The scenario used showed that there exists a definite mutual relationship or connection between the SCRUM Agile metrics and the recognized SE LIs.

3. How can the SE LIs be successfully integrated into the modern, SCRUM Agile, iterative software development process?

It was found during the scenario that indeed the SE LIs can successfully be integrated into the SCRUM Agile, iterative software development process with little sprint interruption. This revelation is important because smooth, seamless, non-intrusive integration is crucial if the recognized SE LIs are to be a positive contribution to the SCRUM Agile, iterative software development process.

Summary

This chapter took what was discussed in Chapter III and applied it to an applicable scenario and developed representative results of a real world situation. The overall intent of this section was to take a simulated scenario with applicable constraints and mirror as closely as possibly the real world. As such this chapter was able to answer the investigative questions posed in chapter 1 in a sufficient manner. Chapter V will further expand on the overall conclusions of this research effort as well as discuss the significance of this research and possible future research efforts that can further the topic discussed in this paper.

V. Conclusions and Recommendations

Chapter Overview

Chapter V will cover the results of determining which SE LIs can be applied to software SCRUM Agile projects in a useful way. Chapter V will discuss the conclusions found during the course of this research effort. Chapter V will also discuss the significance of this research effort to the greater systems engineering and SCRUM communities. Chapter V will dissect all the significant revelations and expound upon them and bring together the holistic significance and impact of this research effort. Finally, Chapter V will conclude with recommendations of potential future research efforts that can expand and further the insights into systems engineering and the approach to utilizing a modern, SCRUM Agile, iterative software development process during a systems engineering effort.

Conclusions of Research

The major conclusion of this research effort is that the majority of the SE LIs, 16 out of 18, relate to the SCRUM Agile metrics and can successfully be integrated and utilized within the modern, SCRUM Agile, iterative software development process. This research also shows that when a SCRUM Master or systems engineer utilizes these 16 SE LIs, they gain the ability to enhance their overall perceptions of how well a project is progressing and ultimately performing and thereby will be able to gain better insight into the health and wellbeing of a project and thus make better decisions on where to adjust if needed. Finally, by showing that 16 SE LIs can be successfully integrated into the modern, SCRUM Agile, iterative software development process, this research has shown

there is a possibility of hybridization and combination of the SCRUM Agile and systems engineering methodologies in order to improve the project management processes as a whole.

Significance of Research

The significance of this research is threefold.

- a) First it identified SE LIs can be applied to modern, SCRUM Agile, iterative software development. This was accomplished through first identifying the SE LIs individually and analyzing through scholarly interpretation which of the identified SE LIs could potentially be applied to, SCRUM Agile, iterative software development.
- b) It demonstrated the identified SE LIs qualitatively compared with SCRUM Agile metrics in such a way they can be successfully integrated together.
- c) It demonstrated SE LIs can be successfully utilized successfully in conjunction with a SCRUM Agile process.

Recommendations for Future Research

There exists several avenues of approach to continue this line of research and further developed the idea that SE LIs and SCRUM metrics can both be utilized together in a modern, SCRUM Agile, iterative software development process.

The most logical next step, however, is to take a simple systems engineering type project that requires some level of iterative software development and/or programming and actually run it through a modern, SCRUM Agile, iterative software development process.

Summary

In conclusion the goal of this research effort was to identify which SE LIs can be applied to software SCRUM Agile projects in a useful way. The thesis objectives sought to identify and qualitatively assess a set of SE LIs applied during a representative software project. It is hoped this research effort has demonstrated that not just one method can be utilized during a SCRUM Agile project but in fact multiple methods could be utilized in a productive manner that could make the overall effort more successful than if just one method was utilized.

Appendix A

SE Leading Indicator	Description
Requirements	Rate of maturity of the system definition against the plan. Also characterizes stability and completeness of system requirements which could potentially impact design and production.
System Definition Change Backlog	Change request backlog which, when excessive, could have an adverse impact on the technical, cost and schedule baseline.
Interface	Interface specification closure against plan. Lack of timely closure could pose adverse impact to system architecture, design, implementation and/or V&V any of which could pose technical, cost and schedule impact.
Requirements Validation	Progress against plan is assuring that the customer requirements are valid and properly understood. Adverse trends would pose impacts to system design and activity with corresponding impacts to technical, cost & schedule baseline and customer satisfaction.
Requirements Verification	Progress against plan is verifying that the design meets the specified requirements. Adverse trends would indicate inadequate design and rework that could impact technical, cost and schedule baselines. Also, potential adverse operational effectiveness of the system.
Work Product	Approval Adequacy of internal processes from the work being performed and also the adequacy of the document review process, both internal and external to the organization. High reject count would suggest poor quality work or a poor document review process each of which could have adverse cost, schedule and customer satisfaction impact.
Review Action Closure	Responsiveness of the organization in closing post-review actions. Adverse trends could forecast potential technical, costs and schedule baseline issues.
Risk Exposure	Effectiveness of risk management process in managing / mitigating technical, cost & schedule risks. An effective risk handling process will lower risk exposure trends.
Risk Handling	Effectiveness of SE organization in implementing risk mitigation activities. If the SE organization is not retiring risk in a timely manner, additional resources can be allocated before additional problems are created.
Technology Maturity	Risk associated with incorporation of new technology or failure to refresh dated technology. Adoption of immature technology could introduce significant risk during development while failure to refresh dates technology could have operational effectiveness / customer satisfaction impact.

Technical Measurement	Progress towards meeting the Measures of Effectiveness, Measures of Performance, Key Performance Parameters and Technical Performance Measures. Lack of timely closure is an indicator of performance deficiencies in the product design and / or project team's performance.
Systems Engineering Staffing & Skill	Ability of SE organizations to execute total SE programs Includes quantity of SE personnel assigned, the skill and seniority mix and the time phasing of their application throughout the program lifecycle
Process Compliance	Quality and consistency of the project defined SE process. Poor / inconsistent SE processes and / or failure to adhere to the SE process increase program risk.
Facility and Equipment	Availability of critical facilities and equipment needed. Composed of two metrics; one type that measures facility availability and the other that measures equipment availability.
Defect and Error	The amount of defects and errors over time for a project. A defect is a deviation of a product at any stage of its development, implementation, or operation from its requirements or applicable standards.
System Affordability	The estimate of the cost of the system at the end of the project with a given confidence and the customer's ability or willingness to pay that price for the project's deliverables.
Architecture	The progress that an engineering team is making towards developing comprehensive system architecture.
Schedule and Cost Pressure	The impact of schedule and cost challenges to the execution of the project. Also, the percentage differences between project estimates and contracted values.

* Reference: Rhodes, Valerdi, and Roedler. 2008. *Systems Engineering Leading Indicators for Assessing Program and Technical Effectiveness*.

*Reference: Roedler, Rhodes, Schimmoller, Jones. 2010. *Systems Engineering Leading Indicators Guide, Version 2.0*.

Appendix B

SCRUM Metric	Description
Team Velocity	Measurement of the average amount of work a SCRUM Agile team can reasonably be expected to accomplish during a given sprint based on past work of a similar nature.
Sprint Burn-Down	Measures the rate at which the work that the team committed to during a given sprint, i.e. sprint backlog items measured in either story points or hours, is being accomplished
Release Burn-Up	Measures the rate at which the product that is being worked on during sprints is close to being completed and ready for release.
Delivered Business Value	Either the actual or perceived value gained from investment.

* Reference: *SCRUM Alliance. (n.d.). Learn About SCRUM. Retrieved March 30, 2015, from SCRUM Alliance Web Site: <https://www.SCRUMalliance.org/why-SCRUM>*

Bibliography

- Agile Alliance. (n.d.). *Agile Alliance: The Agile Manifesto*. Retrieved March 30, 2015, from <http://www.agilealliance.org/the-alliance/the-agile-manifesto/>
- Jones, C., Rhodes, D. H., Roedler, G., & Schimmoller, H. (2010). *Systems Engineering Leading Indicators Guide, Version 2. LAI, INCOSe, PSM and SEARi*.
- Joseph-Malherbe, S. M. (2011). *Agile And Lean Principles And Systems Engineering – A Synergy?*, ISEM 2011 Proceedings (pp. 72-1-72-12). Stellenbosch: ISEM.
- Kennedy, M. R., & Umphress, D. A. (2011). *An Agile Systems Engineering Process: The Missing Link? Fort Belvoir: DTIC*.
- Misra, S., & Omorodion, M. (2011). *Survey on Agile metrics and their inter-relationship with other traditional development metrics*. ACM SIGSOFT Software Engineering Notes, 36(6), 1-3.
- Moe, N. B., Dingsoyr, T., & Dyba, T. (2008, March). *Understanding self-organizing teams in Agile software development*. IEEE 19th Australian Conference on Software Engineering 2008. (pp. 76-85)..
- Morsella, E., & Bargh, J. A. (2011). *Unconscious action tendencies: Sources of ‘un-integrated’ action*. *The handbook of social neuroscience*, 335-347.
- Palmquist, M. S., Lapham, M. A., Miller, S., Chick, T., & Ozkaya, I. (2013). *Parallel Worlds: Agile and Waterfall Differences and Similarities*. Fort Belvoir: DTIC .
- Rhodes, D. et. al. *Systems Engineering Leading Indicators for Assessing Program and Technical Effectiveness, Systems Engineering Volume 12: Wiley Periodicals, 2009*.
- Rhodes, D., Ross, A., Gerst, K., & Valerdi, R. (2009). *Extending Systems Engineering Leading Indicators for Human Systems Integration Effectiveness*. In 7th Annual Conference on Systems Engineering Research.
- Roedler, G. et. al. *Systems Engineering Leading Indicators Guide, Version 2.0. INCOSE-TP-2005-001-03. San Diego, CA: INCOSE, 29 January 2010*.
- SCRUM Alliance. (n.d.). *Learn About SCRUM*. Retrieved March 30, 2015, from SCRUM Alliance Web Site: <https://www.SCRUMalliance.org/why-SCRUM>
- Turner, R. (2007). *Toward Agile Systems Engineering Processes*. *Cross Talk*, 11-15.
- Wrubel, E., Miller, S., Lapham, M. A., Chick, T. A., Brey, D., Nidiffer, K., Walker, J. C. (2014). *Agile Software Teams: How They Engage with Systems Engineering on DoD Acquisition Programs*. Fort Belvoir: DTIC.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 24-03-2016		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From – To) January 2015-February 2016	
4. TITLE AND SUBTITLE Application of System Engineering Leading Indicators to Scrum Agile Projects				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Shirley, Eric M., PMP, CSM, Civilian				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENV-MS-16-M-183	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally Left Blank				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT SCRUM Agile metrics and System Engineering Leading Indicators (SE LIs) are measures used to evaluate the effectiveness and performance of a project across its lifecycle. The application of metrics can provide information on both internal resources and external project stakeholder requirements and constraints, showing past changes, trends, and potential future impacts. Both SCRUM Agile and SE LIs are designed to give a systems engineer/project manager insight into development progress allowing for progress to be checked at regular intervals as well as early corrective actions to be taken if necessary. This research applied SE LIs to a SCRUM Agile project to demonstrate if it was possible to integrate and utilize the two different sets of performance metrics to enhance a SCRUM Agile project. This thesis identified 16 of the 18 SE LIs could be integrated and utilized in an SCRUM Agile project. Those 16 identified leading indicators were applied to a representative software development scenario in order to demonstrate how it is possible to integrate SE LIs into a SCRUM Agile project.					
15. SUBJECT TERMS Leading Indicator; SCRUM Agile; Agile Metrics; Systems Engineering Handbook; Agile Manifesto					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Colombi, John M., AFIT/ENV
U	U	U	U	107	19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, x 3347 (john.colombi@afit.edu)