January 2011

Redesigning SDLC for HTG

Khushbu V. Patel
Worcester Polytechnic Institute

Stephanie M. Manson
Worcester Polytechnic Institute

Trevor Royce Podber
Worcester Polytechnic Institute

Redesigning SDLC for HTG

MQP BXT - 1001

A Major Qualifying Project Report:

submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements

for the Degree of Bachelor of Science

by

Stephanie Manson
Khushbu Patel
Trevor Podber

Date: December 14th, 2010

Advisors:

Professor Bengisu Tulu

Professor Vance Wilson

Professor Amy Zeng

Table of Contents

List of Figures

List of Tables

Abstract

The Hanover Technology Group (HTG) develops software applications for Hanover Insurance internally. They currently use a waterfall methodology that is very document oriented and inhibits HTG's speed to market. This project explored process redesign opportunities to increase the speed of HTG's delivery process by increasing flexibility. We collected data through staff interviews, document analysis, and research on current SDLC methodologies. We developed redesign recommendations for Scrum and hybrid methodology with guidelines to measure success after implementation.

Acknowledgements

We would like to thank our sponsors Kathleen Mills, Karin Winsky, and Veronica Mulcahy from Hanover Insurance for their support and time. We would also like to thank Professors Bengisu Tulu and Vance Wilson for their support and direction while we developed and completed our project. In addition, we would like to thank Professor Amy Zeng and all HTG employees who took time out of their busy schedules to help us with our project.

Executive Summary

The following provides an executive summary of our work for and with the Hanover Technology Group (HTG) during the period of August 28th through December 15th. HTG operates as a business unit of Hanover Insurance, with the responsibility of creating and implementing all software for internal use. The problem faced by HTG was having a process for software development that was lengthy and difficult to modify. Our core project objective was to develop recommendations that would reduce the time it takes to develop software.

A Software Development Life Cycle (SDLC) is a set of procedures used to create software through various phases such as planning, analysis, design, testing, and implementation. There are multiple SDLC methodologies, each with its own unique elements and designed to meet different business requirements and needs. The current system at HTG utilizes a document-driven methodology which follows the guidelines of waterfall, a process in which all work must be completed in each phase before proceeding to the next. In recent years, agile methodologies have emerged as a new and faster approach to software development. HTG is interested in transitioning from waterfall to an agile methodology.

We utilized, DMAIC, a five phase Six Sigma methodology (Define, Measure, Analyze, Improve, and Control) for identifying and reducing defects in a system. Our work plan included, conducting research concerning different types of SDLCs, and examining case studies from other technology organizations that transitioned from waterfall to agile. We also evaluated the current process at HTG through interviews with HTG personnel, analyses of the documentation used in the process, and by examining the types of SDLC process change requests being received following project initiation.

We researched three types of SDLCs, structured design, rapid application development (RAD), and agile methodologies. Structured design methodologies involve large amounts of documentation, gate meetings, and sign-offs, and require each phase be completed prior to beginning the next. The current SDLC process at HTG follows a structured design methodology, more specifically waterfall. Another structured design methodology we researched is parallel development, in which the project is divided into separate components and is completed simultaneously, a process that HTG was utilizing on multiple projects through process change requests. RAD methodologies involve the creation of prototypes and constantly showing the business sponsor each prototype to ensure project progress is satisfactory. However, we did not find any projects utilizing these methodologies at HTG. Agile methodologies are based on simple iterative development, in which components of the software go through all phases after planning and then the process iterates for the next requirement. Since HTG was interested in agile methods due the flexibility built into the process and ability to deliver results in a shorter period, we looked at two of the most popular agile methodologies, Extreme Programming and Scrum, to better understand the process and how it can be incorporated into HTG's workflow. We also reviewed literature from publications about other companies which had transitioned from waterfall to agile methods, specifically Scrum, in order to develop an understanding of best practices and the major challenges we would likely face.

We interviewed 15 HTG personnel with different roles in the SDLC to help us better understand how the process works in practice, critical factors currently slowing down the SDLC, and how HTG personnel perceived agile methodologies. As a result of the interviews, we identified five causes which were slowing down the process: multi-allocated resources, Alternative Practice Requests (APRs), signs-offs and gates, change requests not related to the

process, and lack of business sponsor involvement. We created a fishbone diagram to illustrate current root causes and identify specific sub-issues for each category and utilized this information when developing our recommendations.

APRs are forms which must be completed if a Project Manager wishes to alter the process. In 2009-2010 there were 89 APRs submitted, almost one for every project. We examined these to identify where changes were being made to the process and found 22 relevant requests. We categorized all the APRs, but focused mainly on the 22 most relevant ones to derive our recommendations. These APRs fell into two categories, phase overlap and tracks. Phase Overlap category requests focused on a project team's need to work on the same documentation in multiple phases. Tracks category requests focused on dividing a project into sub-projects to be worked on in parallel. These changes represent characteristics of agile methodologies and parallel structured design respectively. We created a chart that compares all of the APRs by size of the project. In addition we created a concentration diagram of the 22 relevant APRs to illustrate the phase where most change requests occurred.

We formulated two recommendations using the above data: (1) a set of criteria for a pilot project utilizing Scrum and (2) a hybrid SDLC methodology. Scrum is one of the most popular and easy to use agile methodologies. Through our analysis of current resources at HTG and employees perceptions and knowledge, we concluded that Scrum can be successfully implemented, and its implementation would decrease the time required to develop software at HTG. The hybrid methodology we designed can address more specific concerns at HTG, and based on previous years APR analyses, should eliminate 25% of APRs. The hybrid methodology allows for both categories of change requests, phase overlap and tracks. In the hybrid methodology, we modified the current SDLC to include four evaluation points in lieu of the 2nd

and 3rd gates. At these points, the core team meets with the business sponsor to assess the progress of the project. If the team feels the project meets the criteria for changing the process, then decisions are made on how to move forward.

As Hanover strives to remain competitive in the insurance market, an information intensive industry, it is critical that they efficiently develop software applications that serve the needs of the business. The implementation of either or both of our recommendations will allow HTG to decrease the time required to develop software, and provide more value to the business partners.

Authorship

Abstract - All

Acknowledgements- Trevor

Executive Summary - Trevor

1.0 Introduction - Stephanie

2.0 Literature Review & Background – All

2.1 Insurance Industry and Hanover - Trevor

2.2 SDLC Practices - Khushbu

2.2.1 Structured Design – Khusbu

2.2.1.1 Structured-Waterfall System – Khusbu

2.2.1.2 Structured-Parallel Development – Khusbu

2.2.2 Rapid Application Development – Khusbu

2.2.2.1 RAD-Phased Development – Khusbu

2.2.2.2 RAD-Prototyping – Khusbu

2.2.3 Agile Methodologies– Khusbu

2.2.3.1 Agile-Extreme Programming– Khusbu

2.2.3.2 Agile-Scrum– Khusbu

2.2.3 Hybrid Methodologies– Khusbu

2.2.4 Selecting the appropriate Development Methodology– Khusbu

2.3 Measuring Success of an Agile Project - Trevor

2.3.1 Organizational Factors- Trevor

2.3.2 People Factors- Trevor

2.3.3 Process Factors- Trevor

2.4 Case Studies of Companies that Transitioned from Waterfall to Agile – Stephanie

2.4.1 Farm Credit Services - Stephanie

2.4.2 Capital One Auto Finance - Stephanie

2.4.3 Yahoo! – Stephanie

2.4.4 Microsoft IT – Stephanie

3.0 Methodology - Stephanie

3.1 Define – Stephanie

3.2 Measure - Stephanie

3.2.1 Interviews - Stephanie

3.2.2 Document Analysis - Stephanie

3.3 Analyze – Stephanie
3.4 Improve - Stephanie
3.5 Control - Stephanie
4.0 Results - All
4.1 Define - Trevor
4.2 Measure - Trevor
4.2.1 Hanover SDLC - Trevor
4.2.2 Definitions - Trevor
4.2.3 Structure - Trevor
4.2.4 Core Team Members - Trevor
4.2.5 Project Team Members - Trevor
4.2.6 Change Management Procedures - Trevor
4.3 Analyze – Stephanie & Khushbu
4.3.1 Interview Analysis – Stephanie
4.3.1.1 Resources - Stephanie
4.3.1.2 Procedures - Stephanie
4.3.1.3 Business Sponsor - Stephanie
4.3.1.4 Change Requests - Stephanie
4.3.1.5 APRs- Stephanie
4.3.2 APR Analysis - Khushbu
5.0 Reccomendations – All
5.1 Reccomendation One – Scrum - Trevor
5.1.1 Improve - Trevor
5.1.1.1 Barriers and Enablers to Scrum Adoption at HTG - Trevor
5.1.1.1.1 Location of Teams/Work Environment - Trevor
5.1.1.1.2 Allocation of Resources - Trevor
5.1.1.1.3 Training - Trevor
5.1.1.1.4 Business Sponsor Commitment- Trevor
5.1.1.1.5 Cultural Changes - Trevor
5.1.1.1.6 Enablers - Trevor
5.1.1.2 Pilot Project - Trevor
5.1.2 Control - Trevor
5.2 Reccommendation Two - Hybrid Methodology – Khusbu & Stephanie
5.2.1 Improve - Stephanie

5.2.1.1 Evaluation Points - Stephanie

5.2.1.1.1 Evaluation Point 1- Stephanie

5.2.1.1.2 Evaluation Point 2 - Stephanie

5.2.1.1.3 Evaluation Point 3 - Stephanie

5.2.1.1.4 Evaluation Point 4 - Stephanie

5.2.1.1.5 Integration Point - Stephanie

5.2.1.2 Flow with Evaluations Points - Stephanie

5.2.1.3 Evolving Documentation - Stephanie

5.2.1.4 Implementation Plan - Stephanie

5.2.2 Control - Khushbu

6.0 Reflection –Khushbu & Stephanie

6.1 DMAIC - Khushbu

6.2 Constraints - Stephanie

6.3 Design Discussion - Khushbu

7.0 Conclusion- Khushbu

1.0 Introduction

The Hanover is an Insurance Company based out of Worcester Massachusetts that provides a variety of services to their customers. Hanover promises world class performance to its customers, using the trademarked slogan "Our policy is performance". They promise to achieve this by responding to individual needs, utilizing the latest technology, and by having great employee attitudes (Hanover, 2010). Their commitment to their customers is focused on doing what is promised, doing it well, and doing it with speed. They feel that they can accomplish this by having "quality underwriting, innovative products, powerful applied technology and responsive service" (Hanover, 2010).

In order to provide their customers and agents with the tools and software needed to provide world class performance, Hanover business units must enlist the help of the Hanover Technology Group (HTG). HTG is responsible for creating and implementing all software development projects. The HTG creates projects for the following business units:

- PL (Personal Lines) – Projects relating to insurance for individuals including home, automobile and property
- CL (Commercial Lines) – Projects relating to insurance for businesses ranging from small companies to large institutions
- EB (Emerging Business) – Projects for the business unit that focuses on new markets and technology
- Claims (Insurance Claims) – Projects for the business unit that handles insurance claims
- HTG (Hanover Technology Group) – Internal projects for the HTG
- Other – Any other projects Hanover Insurance requires (Donna Wallace, personal communication)

The Hanover is a company based on service. They are smaller relative to other Insurance companies. They compete on the basis of personal service and responding to individual needs.

In order to stay competitive in the insurance industry, HTG would like to increase their speed to market. This project investigated ways that the HTG's Software Development Life

Cycle (SDLC) can become more adaptable to changing business requirements to improve speed to market. For a technology organization, the SDLC is a set of core procedures that facilitate software development. By researching agile software development methodologies, we chose the features that are most appropriate for enhancing HTG's development time. The methodology for this project follows DMAIC, a Six Sigma tool for identifying and reducing defects in a system. We collected data through interviews and documentation analysis. Based on our analysis, we are recommending two different SDLC methodologies that contain agile features and we also developed guidelines for selecting the appropriate methodology for a given project.

The following chapters of this report contain a literature review and background section, a description of the methodology used, our results, recommendations and conclusions. The literature review and background section gives an overview of the insurance industry and SDLC methodologies. The methodology chapter explains how this project follows a DMAIC approach. The results chapter includes the results found through the DMAIC process. The recommendation chapter explains our two recommendations, and offers guidelines for implementation and metrics to measure the implementation. Our conclusion chapter summarizes the impact of our findings and our contributions.

2.0 Literature Review & Background

This section provides a background of the insurance industry, current SDLC methodologies, case studies of companies that have transitioned their SDLCs from waterfall to agile methodologies and the critical success factors for transitioning to agile methodologies.

2.1 Insurance Industry and Hanover

Insurance began when "Chinese merchants devised an ingenious way of protecting themselves against the chance of a financially ruinous upset in the treacherous river rapids along their trade routes" (Insurance Information Institute, 2010). Since then it has evolved into a way for people to protect against various events in the form of the following types of insurance:

- Property Insurance – provides coverage for damage to property that may come from accidents, theft or natural disasters
- Health Insurance – provides coverage for medical expenses and other health related issues
- Life Insurance – provides coverage for an individual that results in a payment if the individual dies
- Casualty Insurance – provides coverage for accidents that may include auto or boat insurance
- Travel Insurance – provides coverage for any problems that may happen while traveling (Insurance Information Institute, 2010)

For each of these policies, the risk for each individual or institution is assessed and the rates are based on this risk. Depending on the amount of coverage required, the rate will be correlated to the size of the policy. Payments are usually made monthly and are directly related to the risk factors that affect the kind of insurance each individual or institution wants. Risk factors differ for each kind of insurance but include: age, gender, geography, and history. For example, if two people of the same age wanted health insurance but one of them smoked cigarettes, the one who smoked would have a higher premium because the insurance company would be taking on a greater risk by giving the smoker an insurance policy.

The Hanover Insurance Group, Inc. was established in 1852 in New York City to protect businesses and homeowners against loss due to fire. By the early 20th century, Hanover expanded its business to include automobile and marine insurance. In 1969, Hanover became affiliated with State Mutual, which had just joined with Citizen Insurance of Michigan. This affiliation gave Hanover access to "new resources for product development, underwriting, data processing, investment, and claim, policyholder and Agent Services" ("Heritage", Hanover, 2010). Also around this time, Hanover moved to Worcester Massachusetts in order to lower its operating costs. By the end of the 1970s, Hanover became a recognized and trusted name in local markets. Today Hanover has a wide range of personal and commercial lines.

2.2 SDLC Practices

A Software Development Life Cycle (SDLC) is the process of understanding how an information system can support the business needs, designing the system, building it and delivering it to users. The SDLC has a set of four fundamental phases listed below (Dennis, Wixom, & Roth, 2006).

> **Planning** focuses on understanding why an information system should be built and determining how the project team will go about building it. Planning involves two steps, the project initiation and project plan.
> **Analysis** focuses on defining the user of system, what the system will do and where and when will it be used. An analysis strategy is developed to guide the team by which they can gather requirements. Finally the system proposal is produced at the end of this phase.
> **Design** focuses on deciding how the system will operate in terms of the hardware, software and software infrastructure. It includes screen layouts, business rules, diagrams, pseudo code and other documentation to determine exactly how the system will operate. This phase provides more information about the architecture design, database and file specifications for the program.
> **Implementation** focuses on building and testing the system to ensure it performs as designed. Installation is the next step followed, in which the new system replaces the old one. A support plan and training plan built by the analyst team is provided to the customer.

2.2.1 Structured Design

The formalized approach to implement an SDLC is known as a methodology. Structured methodologies adopt a formal step-by-step approach to SDLC that moves from one phase to the next. There are several different types structured design methodologies.

2.2.1.1 Structured-Waterfall System

The waterfall model is a sequential software development process. The name of the methodology was adopted because the pictorial representation, as seen in Figure 1, shows each phase flowing naturally into the next phase like a waterfall (Kasser, 2002). With this methodology, the analysts and users proceed sequentially from one phase to another. The key deliverables from each phase are presented to the sponsors for approval as they move to next phase. Once the work from one phase is approved, the phase is closed and the project moves on to the next phase. It is possible to make changes to the project after a phase is closed, but it is extremely difficult to go back. In waterfall, the processes are well defined and established resulting into a successful project when used sequentially (Dennis, Wixom, & Roth, 2006).

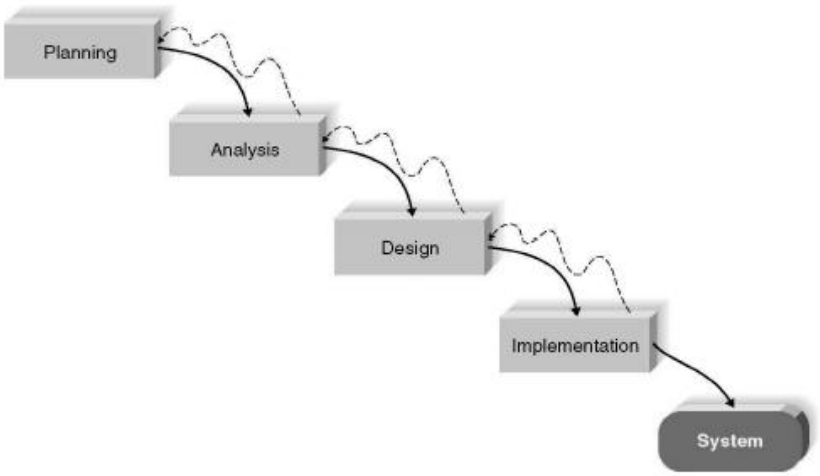

Figure 1: Waterfall Development-based Methodology (Dennis, Wixom, & Roth 2006 p.11)

Waterfall has well defined phases, it is easy to understand and use, and it provides structure to inexperienced staff. The requirements for the project are gathered at the front end so there is stability in the set of the requirements. This works well when quality is more important than the cost or schedule of the project.

For a waterfall system, all the requirements have to be known up front. This can be difficult because many times clients are not sure about what they want. Sometimes there may be a misunderstanding in what a client wants, and because waterfall provides little opportunity for the client to preview, these misunderstandings may not be caught until late in the project.

Waterfall methodology is not flexible. Once the deliverables are created for each phase they are considered frozen. Many times this gives a false impression of progress. For instance, according to the deliverables, the team may be on the analysis phase, but in reality the design team may have already begun their work. The Waterfall system should be used when the requirements are well known, the product definition is stable, the technology is well understood, or it is new version of an existing product (Dennis, Wixom, & Roth, 2006).

2.2.1.2 Structured-Parallel Development

The parallel development methodology addresses the long time interval between the analysis phase and the delivery of the system. A general design of the whole system is performed, and then the project is divided into a series of distinct sub-projects that can be designed and implemented in parallel (Dennis, Wixom, & Roth 2006, p.11). After all the sub-projects are completed, the final integration of the project is done and the system is delivered (Refer to Figure 2).

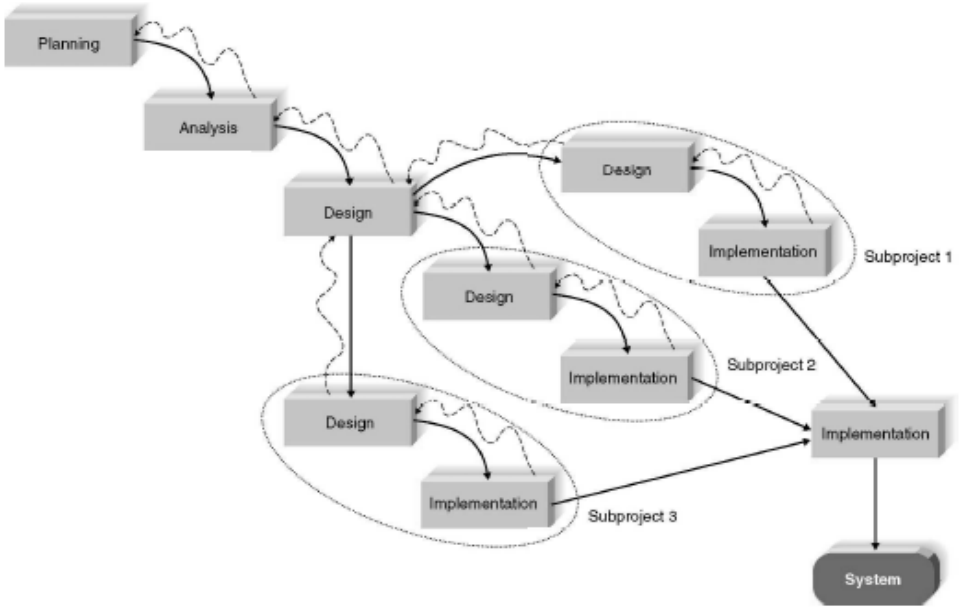

Figure 2: Parallel Development-based Methodology (Dennis, Wixom, & Roth 2006 p.12)

The major advantage of this methodology is that the delivery time of the system is shortened. The drawback of this methodology is that sometimes the sub-projects are not completely independent, and certain design decisions made in sub-projects may affect one another. Even though parallel development has a shorter delivery time than waterfall, there are still documented deliverables which add time to the project (Dennis, Wixom, & Roth, 2006).

2.2.2 Rapid Application Development

Rapid Application Development (RAD) adjusts the SDLC phases to get some parts of the system developed quickly and into the hands of the users. Using this methodology, the users can better understand the system and suggest revisions that bring the system closer to what is needed (Dennis, Wixom, & Roth, 2006). The RAD software model is a "high speed" adaptation of the linear sequential model in which rapid development is achieved by using a component-based construction approach. RAD compresses the waterfall development method into an iterative process. The RAD approach includes developing and refining the data model, process models, and prototyping in parallel using an iterative process (Refer to Figure 3).

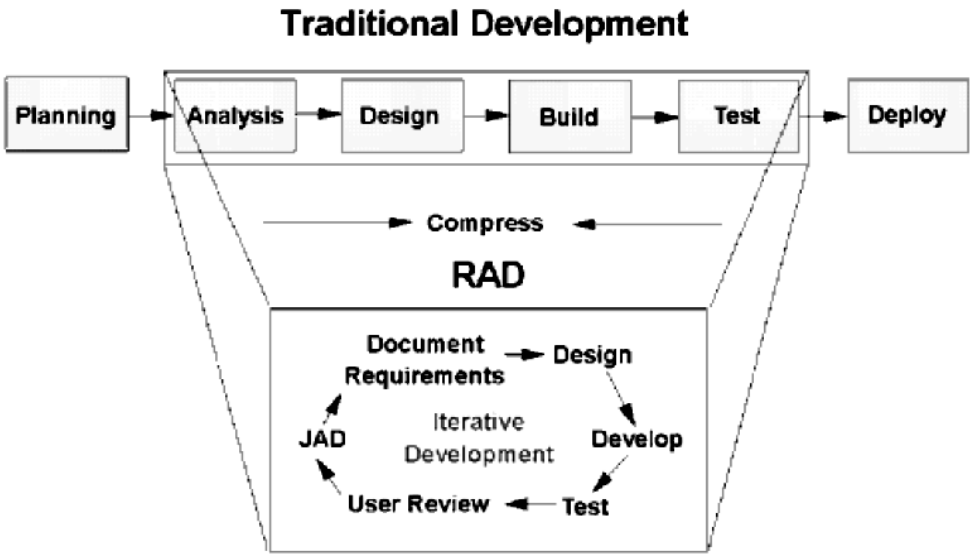

Figure 3: RAD Approach (CaseMaker, 2000)

There are two major types of rapid application development: phased development and prototyping.

2.2.2.1 RAD-Phased Development

The Phased development-based methodology breaks the overall system into a series of versions that are developed sequentially. The system requirements are divided into series of versions by the project team, sponsors, and users during the analysis phase. Important requirements of the system are listed under the first version of the system, then the project moves to the design phase and so on for requirements in the first version. Once the first version is completed, the analysis for version two is performed on the basis of previously formed ideas and additional new ideas and issues that arise from the users experience with version one. Each version goes through the development phase and is passed on to next version (Refer to Figure 4).

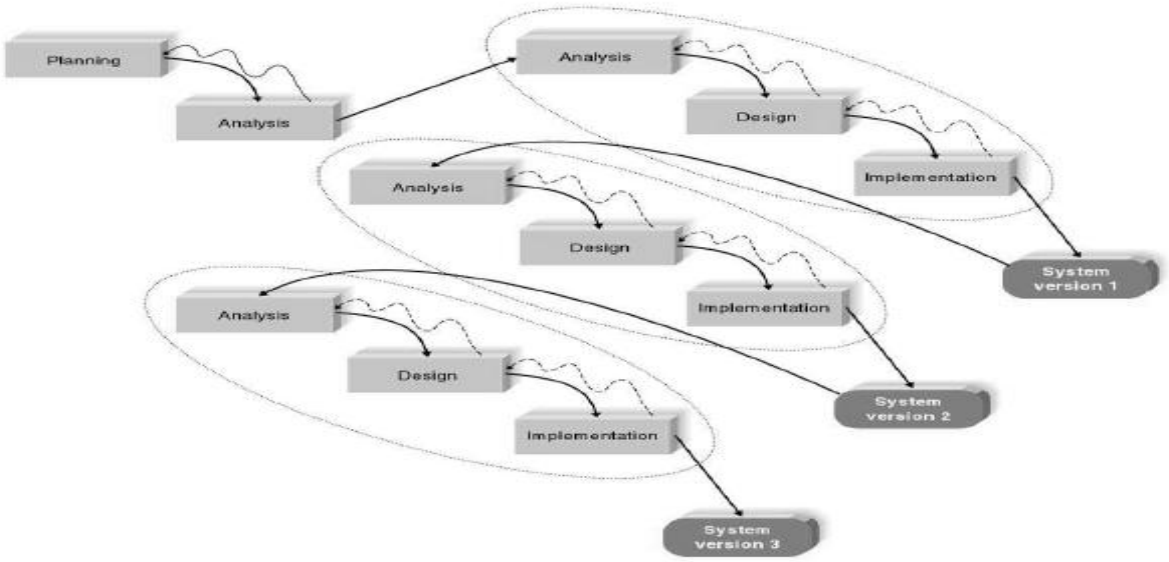

Figure 4: Phased Development-based Methodology (Dennis, Wixom, & Roth 2006 p.13)

The advantage of using a Phased development methodology is quickly getting a useful system into the hands of the client. This provides business value sooner than if the system was delivered only after all the requirements were completed. Even though the first version is a basic one and may not perform all the functions the client requires, it is critical to recognize the most

important and useful features and include them in the first version. This is due to the fact that the user begins to work with the system sooner and therefore can identify important additional requirements or can recognize flaws within the system earlier than if a structured design methodology was used (Dennis, Wixom, & Roth, 2006). The major drawback to creating an early version is that the sponsor may reject the system, which would be a waste of time and resources.

2.2.2.2 RAD-Prototyping

Prototyping methodologies overlap the analysis, design, and implementation phases all three phases are performed rapidly in a cycle until the system is completed. The basic analysis and design are performed, and then the work begins on the system prototype. Prototypes are constructed initially, as subsequent prototypes are developed, features of each prototype may be discarded while others are incorporated into the final product (Alexander & Davis, 2002). The first prototype is used by the client, who provides reactions and recommendations, which are used to create the next prototype. This process is repeated iteratively until the final system is delivered (Refer to Figure 5).

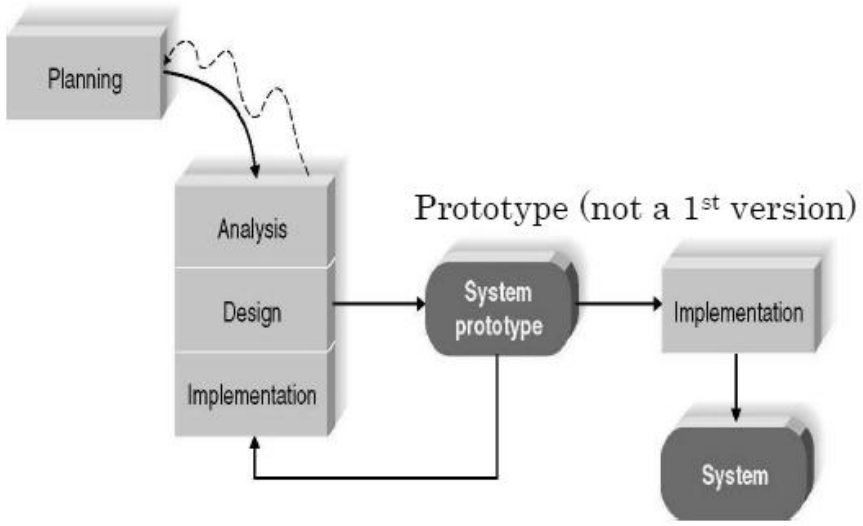

Figure 5: Prototyping-based Methodology (Dennis, Wixom, & Roth, 2006, p.14)

Prototyping methodologies quickly provide a system for the users to interact with. This approach helps to expedite the refinement of the real requirements. The users can interact with the prototype to understand what the system is capable of doing and what additional functionality it needs. An approved prototype is the equivalent of waterfall documentation on the basis that it provides a measure of progress. The advantage to prototyping is that misinterpretation of requirements can be identified earlier (Tripp & Bichelmeyer, 1990). The three major disadvantages are the time required for user participation, false user expectations, and increased development costs (Kinmond, 2002). Using a prototyping methodology can cause problems in the development of complex systems because the fundamental issues and problems may not be recognized until well into the development process.

RAD reduces cycle time and improves productivity. Clients are involved throughout the complete cycle minimizing the risk of not achieving the client satisfaction and business needs. RAD emphasizes reusing existing program components and creating reusable components. This

process emphasizes the program components that have already been tested, which minimizes testing and development time (Dennis, Wixom, & Roth, 2006).

2.2.3 Agile Methodologies

In the last few years agile methodologies have emerged as an alternative to traditional software development practices that are documentation driven (Ilieva, Ivanov, & Stefanova, 2004). Agile projects emphasize simple, iterative application development (Dennis, Wixom, & Roth, 2006). The agile methodologies discussed are: Extreme Programming, Scrum, and hybrid systems that employ features from multiple methodologies.

2.2.3.1 Agile-Extreme Programming

Extreme Programming (XP) is an agile methodology that involves close client collaboration, incremental software delivery and team based development done in pairs (Andersson, 2006). XP is founded on four core values: communication, simplicity, feedback and courage. XP employs practices that require communications such as: unit-testing, pair programming and task estimation. Simplicity refers to prioritizing work by the simplicity of the task. XP requires feedback on different time scales. Programmers give minute-by-minute feedback on the state of the system. Clients are given immediate feedback on the quality of their user stories. XP encourages taking drastic and unanticipated measures/actions such as throwing code away or breaking running tests in order to fix a flaw (Juric, 2002).

The key principles of creating a successful system using XP are; continuous testing, simple coding performed by pairs of developers and close interaction with end users to build the system. After a superficial planning process, analysis, design, and implementation phases are performed iteratively (Refer to Figure 6).

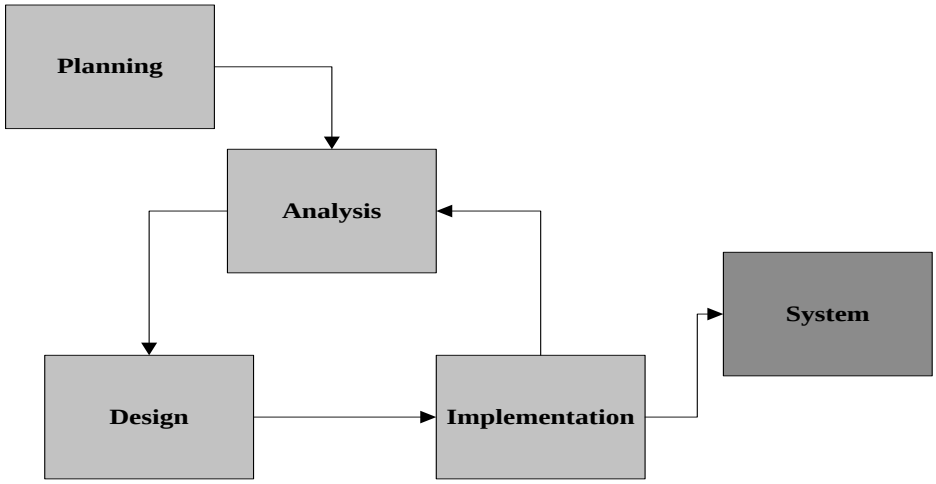

Figure 6: Extreme Programming

XP begins with user stories, which are small descriptions of what a system needs to do. Then the programmers code and test small, simple modules to ensure they meet requirements set in the user stories. Users are required to be on site to address any issues or questions that may arise. It is important to have standards so that an XP team can use a common set of names, descriptions, and coding practices in order to minimize the confusion (Dennis, Wixom, & Roth, 2006).

XP delivers results sooner than RAD approaches; the development team is rarely stuck in gathering requirements for the system. XP works well for small projects with highly motivated, cohesive, stable, and experienced teams, otherwise the likelihood of a successful project is reduced (Munassar & Govardhan, 2010).

2.2.3.2 Agile-Scrum

Unlike the traditional waterfall method, Scrum is an agile methodology designed to be flexible and adaptable to changing requirements. It provides control mechanisms for planning a product release and then managing variables as the project progresses. This enables

organizations to change the project and deliverables at any point in time delivering the most appropriate release (Schwaber, 1995).

Scrum is comprised of three roles, three documents, and three meetings (Stober & Hansmann, 2009). The three roles that are represented in Scrum are the Scrum Master, the product owner and the team. The Scrum Master maintains the processes (typically in lieu of a project manager). The Scrum Master has the responsibilities of checking in with the team and verifying that the test cases and code reviews are done. The Product Owner represents the stakeholders such as clients and business sponsors. Responsibilities include providing requirements, funding the project and signing off on the deliverables. The Team is a cross-functional group of about seven people who do the actual analysis, design, implementation, testing, etc (Stober & Hansmann, 2009).

The three important documents in Scrum are the Product Backlog, the Sprint Backlog and the Sprint Results. The Product Backlog is a list of all the requirements gathered. The Sprint Backlog is a list of work a team must accomplish during a sprint. This cannot change during the sprint. The Sprint Results are the use cases that are completed during a sprint (Stober & Hansmann, 2009).

The three meetings are the Sprint Planning Meeting, the Daily Scrums, and the Sprint Review. The Sprint Planning Meeting is a meeting at the start of the sprint which is divided into two parts. In the first part, the product owner presents the important requirements from the product backlog, in the second half; the team plans the next sprint in detail. The Daily Scrums occur each day during the sprint, and last about fifteen minutes. During this meeting each team member provides updates of their accomplishments since the previous meeting, their plan for the current day, and alerts the Scrum Master if they have any problems preventing them from

accomplishing their goals. The Sprint Review occurs at the end of each sprint in which the Scrum Master, Team, and Product Owner (along with the other stakeholders) meet again to review the results achieved during the iteration (Stober & Hansmann, 2009). A pictorial representation of Scrum is shown below (Refer to Figure 7).

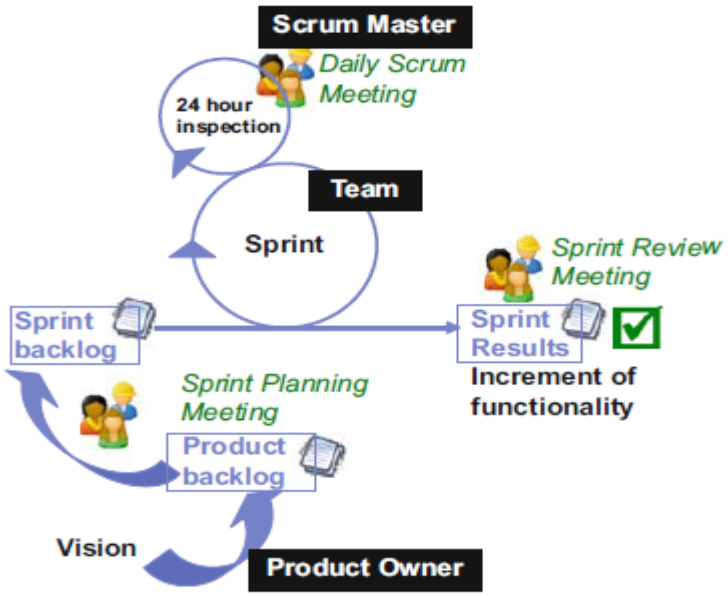

Figure 7: Scrum Methodology (Stober & Hansmann, 2009)

The advantages of Scrum are short iterations, no waste of time, and cross-functionality. By having short iterations and daily meetings, the team will always be able to measure their progress. By working on top priority features, Scrum makes sure that no time is wasted developing code that no one will use. In Scrum there are no defined roles and everyone is responsible for the whole product. By working together, the team uses their unique skills to create better software with higher quality (Bergstrom, 2008). Scrum is a widely used

methodology for software development, however transitioning to Scrum can be difficult. This is discussed in Section 2.3.

2.2.3 Hybrid Methodologies

For system development, no one methodology is perfect; every methodology has its own benefits and flaws. Depending on the requirements of the organization, different methodologies can be combined to achieve the most efficient SDLC (Alexandrou, 2010). Regardless of what type of methodology is chosen, the documentation is very crucial and is done in parallel to the development process. A basic guideline to build a hybrid system is given below.

In general, an SDLC methodology follows these steps (Alexandrou, 2010):

1. If there is an existing system, its deficiencies are identified. This is accomplished by interviewing users and consulting with support personnel.
2. The new system requirements are defined including addressing any deficiencies in the existing system with specific proposals for improvement.
3. The proposed system is designed. Plans are created detailing the hardware, operating systems, programming, and security issues.
4. The new system is developed. The new components and programs must be obtained and installed. Users of the system must be trained in its use, and all aspects of performance must be tested. If necessary, adjustments must be made at this stage.

2.2.4 Selecting the appropriate Development Methodology

It is difficult to choose a methodology because each has its own benefits and drawbacks. Some of the important criteria for selecting the appropriate methodology are listed in Table 1.

Table 1: Criteria for Selecting a Methodology (Dennis, Wixom, & Roth 2006 p.18)

Ability to develop Systems	Structured Methodologies		RAD Methodologies		Agile Methodologies	
	Waterfall	Parallel	Phased	Prototyping	Extreme Programming	Scrum
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
That are Complex	Good	Good	Good	Poor	Poor	Good
That are Reliable	Good	Good	Good	Poor	Good	Good
With a short Time Schedule	Poor	Good	Excellent	Excellent	Excellent	Excellent

Clarity of User Requirements It is difficult to understand and explain unclear user requirements in a written report when users are not sure about what they want a system to do. Users need to interact with the technology to really understand what the system can do and how it applies to their needs. RAD methodologies are appropriate when user requirements are unclear because they provide prototypes for users to interact with early in the SDLC. Extreme Programming may also be suitable if on-site user input is available. Scrum is appropriate too as the product owner is part of the agile team. Structured design methodologies are inept when the user requirements are not known up front or the requirements are prone to change throughout the development cycle. This is due to the fact that the requirements are gathered in the early phases and as they move forward, the requirements are frozen and it is difficult and expensive to go back and make any changes. Structured design methodologies work fine when the system requirements are well-known upfront (Dennis, Wixom, & Roth 2006).

System Complexity Complex systems require careful, detailed analysis and design. Agile methodologies are not appropriate for complex systems. Structured design-based methodologies

can handle a complex system due to the rigid documentation requirements. Given the depth of the requirements, the issue providing users with prototypes isn't as important (Dennis, Wixom, & Roth 2006).

System Reliability Structured methodologies and phased development are the most appropriate because they combine the detailed analysis and design phases. Prototyping-based methodologies lack the careful analysis and design phases that are essential for dependable systems (Dennis, Wixom, & Roth, 2006).

Short Time Schedules RAD methodologies are the best choice for projects on a short time schedule because they enable the project team to adjust the functionality in the system on the basis of a specific delivery date. Waterfall is the worst choice as it does not allow easy schedule changes (Dennis, Wixom, & Roth, 2006).

2.3 Measuring Success of an Agile Project

The success of an agile project can be measured in three ways using the quality of the product delivered, the time it took to create and how closely the process was followed. The quality of the product is determined by the customer, however, most agile methodologies involve the customer during the process so the result should be better. To measure the success of transitioning to an agile process we look at the three critical success factors: organizational factors, people factors and process factors.

2.3.1 Organizational Factors

Organizational success factors are all factors that are related to the organization that is implementing an agile SDLC. These include but are not limited to: strong executive support, committed sponsors, cooperative organizational culture instead of hierarchical, oral culture placing high value on face to face communication, collocation of the whole team, facility with

proper agile-style work environments, and a reward system appropriate for agile (Chow & Cao, 2007). In order for an agile project to succeed, the organization must provide a suitable environment for agile software development.

2.3.2 People Factors

The success of an agile project relies heavily on the personnel involved. Agile methodologies place a heavy emphasis on individuals and interactions, collaboration with customers and a quick response to change by customers (Subhas, Vinod, & Uma, 2009). The people success factors include: team members with high competence and expertise, team members with high motivation, managers knowledgeable in the agile process, managers with an adaptive management style, good customer relationships, and coherent individuals who are self-organizing (Chow & Cao, 2007). Agile methodologies are very interactive and, without people who are motivated and knowledgeable in the methodologies, success will be limited.

2.3.3 Process Factors

There are many types of agile software development practices, but defining the criteria and having a well-defined process are key to the success of an agile project. The success factors of the process are as follows: following an agile-oriented requirement management process, strong communication focus with daily face-to-face meetings, honoring regular working schedule (no overtime), strong customer commitment and presence, and giving the customer full authority (Chow & Cao, 2007).

2.4 Case Studies of Companies that Transitioned from Waterfall to Agile

Many companies are putting emphasis on the speed to market for their software development in an effort to stay competitive in the industry. In order to decrease the time to market many companies are redesigning their software development life cycles. This section

looks at the transition from waterfall methodologies to agile methodologies of Farm Credit Services, Capital One Auto Finance, Yahoo!, and Microsoft IT.

2.4.1 Farm Credit Services

The following section refers to an article from CIO Magazine, *From Here to Agility* (Weil, 2007), that discusses Farm Credit Services' transition to an agile SDLC. Farm Credit Services of America provides credit and other financial needs to farmers and ranchers in Iowa, Nebraska, South Dakota and Wyoming. They are part of the Farm Credit System, which is a nationwide agricultural network that provides credit and other financial services to farmers and ranchers across the United States. Farm Credit Services institutions differ from commercial banks because they do not take deposits. They raise their funds by selling system-wide bonds in capital markets. Farm Credit Services created software for both their customers and employees. Prior to 2005 Farm Credit Services followed a waterfall methodology for their software development life cycle.

The waterfall methodology was no longer meeting the needs of Farm Credit Services. Farm Credit Services CIO Dave Martin is quoted as saying "we got requirements and would build [the applications], and nobody was happy in the end" (Weil, 2007). One project that particularly failed with the waterfall methodology was a conversion from a mainframe-based customer application-processing system to a web-based version. This particular project was difficult because it had more than 200 pages of requirements and took nearly three years to complete. During the course of the project many of the requirements and business needs changed, and several members of the original business team left the company. The resulting system was full of defects and was discarded soon after release. Due to this failure and the

dissatisfaction of many other projects, Farm Credit Services decided to look into agile methodologies.

In 2005, CIO Dave Martin, with the help of Lou Thomas and Beth Schmidt, directors of applications development, decided to try a Scrum methodology. Their goal was to have a shippable product every two weeks so they had two-week sprints, daily meetings and regular iteration reviews and testing. They had six development teams that consisted of a business analyst, project leader, two or three developers, a database engineer, one to three business owner participants and a quality assurance engineer. Instead of pages of requirements, the teams wrote "user stories" throughout the project to convey the business needs. The introduction of agile methodologies reduced the number of defects per rollout from around one hundred to less than two. Martin explains that they "rolled out five key products with phenomenal results" and the business owners were ecstatic with the end results.

Although the transition to an agile methodology was successful for Farm Credit Services, they did have some trouble along the way. Many of the people in IT felt that agile was "the flavor of the month" (Weil, 2007). Some people at Farm Credit Services just said that they were not going to do it. Despite these obstacles, Martin feels his move towards agile was a great success and is quoted saying he "couldn't fathom going back to a waterfall methodology" (Weil 2007).

2.4.2 Capital One Auto Finance

This case was presented in a conference paper, (Noble & Tengshe, 2007), entitled *Establishing the Agile PMO: Managing variability Across Project and Portfolios*. Capital One Auto Finance is the third largest non-captive auto lender in the United States, and is the fastest growing division in the Capital One umbrella. Their IT Systems are "the key driver to bringing

new business initiatives quickly and reliably to market" (Noble & Tengshe, 2007, p.188). Before they started to move towards an agile methodology, Capital One Auto Finance followed a waterfall methodology. Following waterfall, projects took a long time to deliver the business's needs. The IT department decided it was time to try a more agile approach.

In 2005, CIO Dick Daniels gave the IT department permission to hire an outside agile coach and to start an agile pilot project. They began their move towards agile by using Scrum. The first pilot project began with a team of 5 members. The agile coach set the team up in an agile room so they were together, and their project sponsor was to be in the room with them for four hours every day. A Scrum Master candidate was identified and he and the team were trained on agile methods.

The pilot project quickly began to run into several obstacles. The first problem was that the team was not fully allocated to the project. This caused a major problem because when the project sponsor went to the agile room none of the project team was there. Another problem was that the team was still using all of the waterfall documentation in addition to the agile documentation. When the team was asked why they were still using the waterfall documentation, they explained that their managers had told them to. The agile coach now had to convince the managers that the waterfall documentation was non-value adding in an agile environment; this proved to be a difficult task. Due to the team not being fully allocated to the project and to use of the waterfall documentation, the first pilot project did not prove to be successful.

With the failure of the first pilot project, the agile coach learned several things: the waterfall documentation could not be part of the agile environment and the team members had to be fully allocated to the project and co-located for easy collaboration. In addition, they put a training program in place to help make the next pilot project more successful. At first, the

training classes did not have many participants. They encouraged employees to take these training classes by offering them credits towards a Project Management certification.

The lessons learned from the first pilot program and the training of employees and support from upper management helped Capital One Auto Finances successfully transition to using an agile methodology. By applying the agile methodologies, the IT department managed to meet their goals of reducing time to market by 50% and business sponsors reported that their customer satisfaction was 100%.

2.4.3 Yahoo!

This case was presented in a conference paper (Chung & Drummond, 2009) entitled *Agile @ Yahoo! From the Trenches*. Yahoo! offers its visitors many web-based options. These options "cover a very broad area of computer and Internet technology, such as social networking, content delivery, search, advertising, mobile, and cloud computing" (Chung & Drummond, 2009, p.113). The software development of these options have "common needs of flexibility, adaptability, fast time to market, and delivering applications" that satisfy the customers' needs (Chung & Drummond, 2009, p.113). Prior to Yahoo! adapting an agile methodology, no one methodology fulfilled the customers' needs.

Before Yahoo! moved to an agile methodology, they used a waterfall methodology with many gates and sign-offs. Project scope and release dates were assigned by upper management; the dates were often difficult for the project teams to meet with their current methodology, leaving the project team to work late hours to meet their release dates. As schedules tightened, many project teams began to ignore the strict waterfall structure.

In 2004, an engineering manager invited an agile expert to make a presentation at Yahoo! that discussed the advantages of adopting agile methodologies. There was a variety of reactions

to this presentation. Most of the developer community was against trying agile methodologies; however the senior director saw that agile methodologies may be able to solve some of the issues with their current methodology (Chung & Drummond, 2009, p.114). After learning more about agile, the senior director began to push agile methods as an alternative to their current waterfall methodology. To help him do so he hired an agile coach.

In 2005, the senior level managers were on board with developing an agile methodology. To start their agile pilot project, senior management decided to take volunteers who were willing to try Scrum and created four pilot project teams. Scrum training was then given to the pilot team as well as any employees that wished to attend. The pilot projects were a success and many teams within the company adopted Scrum as the new way of developing software (Chung & Drummond, 2009, p.116). Within two years of the first Scrum project, over 150 of Yahoo!'s development teams were using Scrum. Agile at Yahoo is still evolving and more development teams are moving towards agile when they realize the benefits it can offer.

2.4.4 Microsoft IT

Microsoft IT (MSIT) explains their move towards agile in the conference paper (Lewis & Neher, 2007, p.1) *Over the waterfall in a Barrel- MSIT Adventures in Scrum*. MS IT is not the people that develop Windows, Office or other Microsoft products. They are an organization "which develop, deliver, and maintain trustworthy technology solutions that protect corporate resources, increase employee productivity, and showcase the value of running Microsoft products" (Lewis & Neher, 2007, p.1). MSIT used a waterfall SDLC to develop, deliver and maintain their software. The waterfall methodology was "required by Microsoft corporate policy to be followed when developing applications that support Microsoft's business function" (Lewis & Neher, 2007, p.1).

In 2006, a development team in MSIT decided they wanted to try Scrum. In order to do this the team first had to get Scrum adopted alongside waterfall, which required the team to petition the SDLC governance body. The team first mapped waterfall terms with Scrum terms. From there they created an agile SDLC guide which described Scrum in terms common to the waterfall SDLC. The team was successful in petitioning the SDLC governance body and was gained permission to use Scrum. The project became MSIT's pilot project. The pilot project went through eleven sprints and was a success.

MSIT took several important steps when moving towards an agile methodology. They first changed their schedule management and phase reviews in their project management approach to create a more agile environment. Another important step is choosing an agile methodology that fit with the company, development teams and business needs. MSIT also found it very important to stay "focused and on track during the few sprints of the pilot project" because some members may want to change things (Lewis & Neher, 2007, p.2). Changing things would mean that the pilot project is no longer true Scrum and therefore cannot conclude that using Scrum was successful. One of the most important steps MSIT took was finding the right team members to participate in the pilot project. They needed a team that was willing to get rid of the waterfall culture and give Scrum a chance. It was also important for the pilot team to have the right coach. During the pilot program MSIT hired an outside consultant to act as the team agile coach.

After the successful pilot projects, MSIT created several agile teams. These teams are committed to agile and to spreading it throughout MSIT. Although many people would like to move completely towards agile, the waterfall SDLC is still an option for development teams and many still follow it.

3.0 Methodology

DMAIC is a methodology used in Six Sigma projects. DMAIC is an acronym for five phases as illustrated in Figure 8: Define, Measure, Analyze, Improve, and Control. DMAIC provides tools and techniques to help an organization improve its processes. It follows a lifecycle design to ensure that an organization first understands underlying problems, processes, root cause, and supporting data without jumping straight to the solution. We chose this methodology because it allowed us to add structure to our project and because this methodology focuses on understanding and achieving what the customer wants (Jacobs, Chase, & Aquilano, 2009).

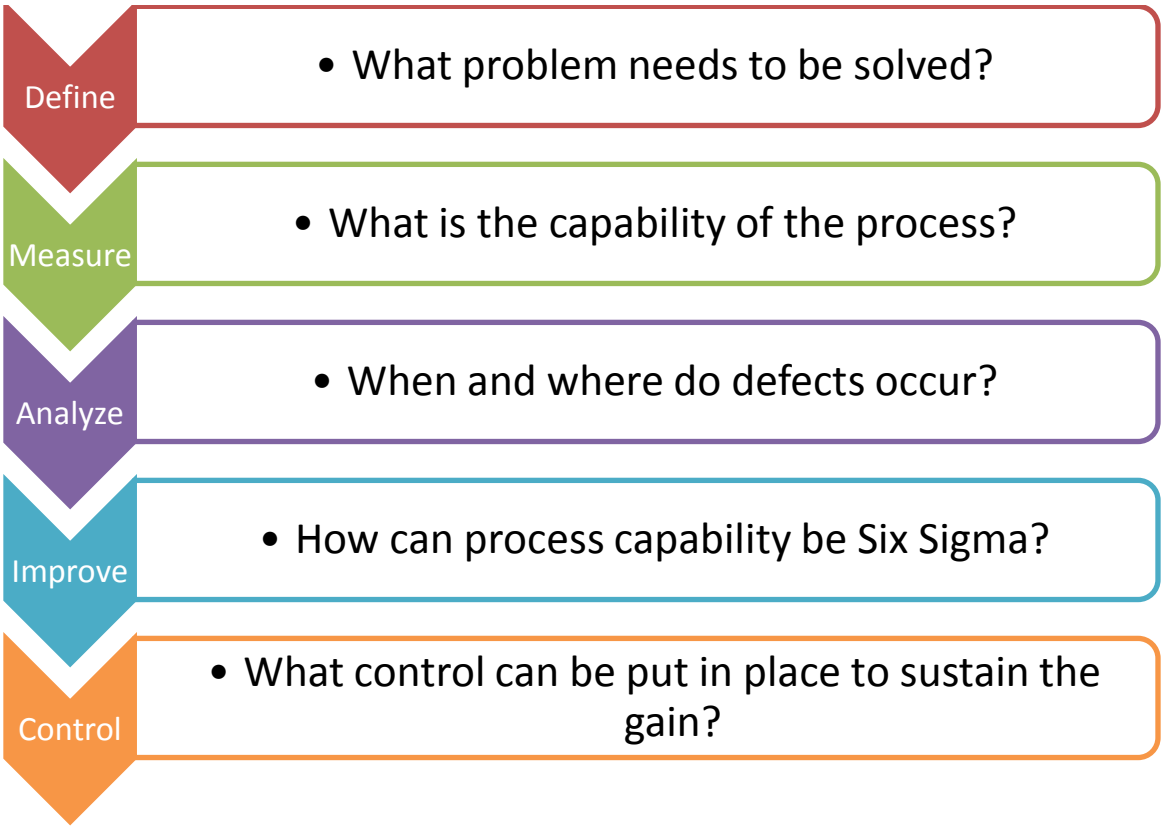

Figure 8: DMAIC Methodology

3.1 Define

The purpose of the define stage is to identify the customer and their priorities (Jacobs, Chase, & Aquilano, 2009). To identify the scope of HTG's problem, we had an initial meeting with our sponsors that involved an overview of the project and established initial delivery dates. During this meeting we were given the current Project Playbook and the Tailoring Matrix. Over the next few days, we analyzed the documents to gain a preliminary understanding of the current methodology. After careful review of these documents, we had unstructured interviews with one of our sponsors, a project manager and a lead architect to broaden our understanding of the process and compile a list of relevant issues in the current SDLC.

3.2 Measure

The measure phase of DMAIC is used to gather data on how the current process is performing and to find where the defects, which in this case are changes to the current system, occur in the current process (Jacobs, Chase, & Aquilano, 2009). There are two main sources we gathered information from, personal interviews with HTG personnel and documentation from within HTG.

3.2.1 Interviews

In an attempt to identify places where it could be improved, we conducted a series of interviews with various members of HTG. To begin our interview process, we formulated a list of roles within HTG that we felt would have important input for our project. This list was given to one of our project sponsors, who then contacted people of these roles for us to interview. We developed both general and specific questions, a list of these can be found in Appendix A. Before conducting our structured interviews with HTG personel, we chose a chair and a secretary. The chair was responsible for leading the meeting, and the secretary had the task of

writing the minutes during and after the interview. The purpose of these interviews was to further our understanding of the process and to gain the perspective of all those who would be affected by a change to the current SDLC. A list of the people we interviewed can be found below in Table 2.

Table 2: List of People Interviewed

Role	**Date**	**Chair**	**Secretary**
Project Manager	9/1/2010	All	All
Enterprise Architect	9/1/2010	All	All
Lead BA	9/8/2010	Stephanie	Khushbu
QA Project Lead	9/8/2010	Trevor	Stephanie
Delivery Manager	9/8/2010	Stephanie	Trevor
Implementation Manager	9/8/2010	Trevor	Stephanie
Project Manager	9/8/2010	Trevor	Khushbu
Solutions Architect	9/15/2010	Khushbu	Trevor
Program Director	9/15/2010	Stephanie	Khushbu
Senior Security Analyst	9/15/2010	Khushbu	Trevor
Project Manager	9/15/2010	Trevor	Stephanie
Program Director	9/22/2010	Stephanie	Trevor
Senior Business Analyst	9/22/2010	Khushbu	Stephanie
Technical Lead	11/3/2010	Trevor	Stephanie
Senior Business Analyst	11/17/2010	Khushbu	Trevor
Project Manager	11/17/2010	Stephanie	Khushbu

3.2.2 Document Analysis

The current SDLC is very document oriented so we examined various documents within HTG. To understand the specifics of each role within HTG, we looked at training manuals and descriptions of roles and responsibilities for team members. To further our comprehension of the process, we examined the current playbook and process flow diagrams. We looked at Alternative

Practice Requests (APRs), which are forms that must be filled out if a Project Manager (PM) wants to engage in an alternative method that is not described in the playbook, to see where PMs had been making changes to the current SDLC. For the APRs we found relevant to our project, we examined documents from their respective SharePoint sites to see how a change to the system affected the documentation.

3.3 Analyze

The purpose of this phase is to determine what causes the changes in the system and to identify why these changes occur and if there are any key variables in these changes (Jacobs, Chase, & Aquilano, 2009). We analyzed the information gathered to understand the defects and deficiencies with the current SDLC. The information gathered is in two forms, interviews and documents. The documentation analysis includes a review of two types of documents, the documentation included in the SDLC and APRs. For the process documentation, we evaluated both document templates and actual documents to understand the required information and decide what information is value adding.

To analyze the interview data collected during the measure phase we compiled the interview minutes into one document. After compiling the data we created a fishbone diagram to illustrate our findings. A fishbone diagram, also known as the cause and effect diagram or an Ishikawa diagram, is a Six Sigma tool used to identify the root causes of a problem or issue (Jacobs, Chase, & Aquilano, 2009).

To create the preliminary fishbone diagram, we first drew a horizontal arrow to represent the major issue of HTG's speed to market. Through analysis of the data, we identified five major causes that inhibited speed to market. The five major causes were resources, procedures, sponsors, change request, and APRs. These five causes were added to the fishbone diagram by

attaching arrows to the issue arrow using a fishbone structure. Next, we added reasons for the causes based on the information recalled from the interviews.

Once the preliminary fishbone diagram was created, quotes within the interview data document were highlighted in different colors based on the five identified causes. A table was created for each of the five causes; these tables can be seen in Appendix B. These tables include information such as the subcategory of the statement, the position of the employee that said it, and their statement on the topic. After the five tables were created the preliminary fishbone diagram was updated to make sure that it included all of the causes and the reasons for these causes that were presented in the interviews.

Once the five tables and the fishbone diagram were completed, it was evident that there was information from the interviews that was not used. Most of this information was not in the form of causes but was of interest because it offered suggestion on how the current SDLC can be improved. This data was then summarized into a suggestions table that was used to guide us in the Improve and Control sections of this report.

The APR documents show how Project Managers have been making changes, and in which phase the change happens. The review of the APR documents helped us understand what types of alternative methodologies HTG is currently using. We began by reading through all the APRs from 2009-2010 and summarizing them in an Excel sheet with the information about the cause for the request, the size of the project and in which phase the APR was submitted. The objective of doing this was to analyze the given information about the APRs and recommend a modified SDLC which incorporates these changes in the current waterfall SDLC. After summarizing the APRs we made a graph to help us understand how many APRs were filled out during that time and the number of APRs we found which pertained to our project. We selected

22 APRs, which were classified into two categories, phase overlap and tracks. Phase overlap describes projects that move onto new phases for parts of the project before the previous phase has been completed. Tracks describe projects that split the requirements so that the project can be done in sub-projects. Following this we created a concentration diagram, a visual tool that is used to analyze where changes occur in a given system, to have a graphical representation of what types of alternative practices are requested in which phase for the given size of the project.

3.4 Improve

The improve stage is used to identify a solution to the problem that the project aims to overcome (Jacobs, Chase, & Aquilano, 2009). For our improve stage we describe how we used the results of our analysis to make recommendations. The root causes of problems were reviewed and two solutions were created. The first recommendation proposed is for HTG to utilize Scrum as a methodology for projects that meet a certain criteria. For this solution we created a flow chart of the Scrum process. We discuss the major barriers to adopting Scrum at HTG and provide recommendations for ways that HTG can overcome these barriers, based on previous research. An implementation plan was created to assist HTG in making and adapting to the changes. The second recommendation is a modified version of the current playbook that incorporated the data from the APR and interview analyses. We created the new playbook by using a whiteboard to map out the current document flow and identifying the areas where either tracks or overlap could occur. We then transferred our design to a Visio flowchart. We suggested ways to audit progress and how to implement the new playbook.

3.5 Control

The control phase is usually used once the recommendations from the improve stage are implemented. This phase explains how to maintain the improvements, and to ensure that the

modified process meets the customer's needs. In most DMAIC processes there would be a continuous loop of improve and control until the process was within Six Sigma quality. However, because we were unable to implement our recommendations due to time constraints, we have provided a plan for implementation and ways to measure success in the control phase. In order to achieve this, metrics were developed to measure the effectiveness of the implementation. Some of the areas that are measured are time, monetary cost, communication, and proper documentation. At the end of this phase, all information and knowledge gained from the project is transferred to our sponsor at HTG.

4.0 Results

The following section describes the Define, Measure and Analysis phases of our project.

4.1 Define

Hanover strives to remain competitive in the insurance field. Their promise to their customers includes both new technology and speed of deployment. The problem that HTG faces is delivering software quickly. Some managers at HTG feel that performance is limited in this area due to their current SDLC methodology. Their current SDLC follows a waterfall methodology which makes it difficult to adapt to changing requirements. Finding a solution to this problem is important to HTG because they are constantly working on software development projects, most of which have a tight timeline. By redesigning the SDLC, they will be able to move from a strict waterfall methodology to a more flexible SDLC which will allow project teams, when appropriate, to utilize agile methodologies. A more flexible SDLC will improve their speed to market, which will give them a better competitive advantage. The objectives for this project are to reduce the time it takes for HTG to develop software and to increase flexibility within the SDLC by broadening the options for development.

4.2 Measure

The following section describes the data we collected from HTG. This includes a description of the process and information about change requests.

4.2.1 Hanover SDLC

The HTG develops software for all of Hanover and their employees. As a need arises or an update is needed to current software, the HTG assigns a Core Team and an extended project team to work on each project. They develop the requirements based on the business needs and create software consistent with these requirements. Oversight of a project is conducted by the

Process Action Board (PAB). The PAB is responsible for conducting an audit of every project; this involves verifying that all required documentation has been completed.

4.2.2 Definitions

In terms of Software Development, Hanover creates all software required for their business internally. Projects are named based on the business unit they are being developed for, followed by a number; for example the first project for Personal Lines will be PPL01. Projects range in size and resources required. Based on this information they are categorized into four different groups. Table 3 shows the hours required, estimated cost, and amount of documentation relative to the complete list.

Table 3: Size of Projects (HTG, 2010)

Type	Small Baseline	Small Projects	Medium Projects	Large Projects
Hours Required	1-60 hours	61-2000 hours	2001-4000 hours	4001+ hours
Estimated Cost		< $150,000	$150,000-250,000	>$250,000
Documentation	Low	Medium	High	High

4.2.3 Structure

The existing methodology for Software Development follows a standard waterfall procedure. It is done in eight phases (pre-initiation, initiation, business systems analysis, test planning and preparation, technical design and construction, testing, implementation, and warranty and closure) and, based on the current playbook, all documentation and requirements must be finished before proceeding to the next phase.

There are three gates that are required for all medium and large projects (HTG, 2007). Within each of these gates the core team meets with the business sponsor to request approval to continue with the project. At each gate meeting the Project Manager is responsible for making

sure the key milestones and deliverables are met, the stakeholders in the project attend the meeting, and an updated estimate is provided along with a checklist of upcoming deliverables (HTG, 2007). Shown below in Figures 9 - 11 is a representation of the current SDLC at Hanover created in Microsoft Project. This depicts documents in blue, processes in yellow, consolidated documents in green (documents that are created from other documents), double consolidated documents in pink (documents that are created from consolidated documents), documents that are updated in purple, phases in black, gates in red and documents that require signoffs with slashed bars.

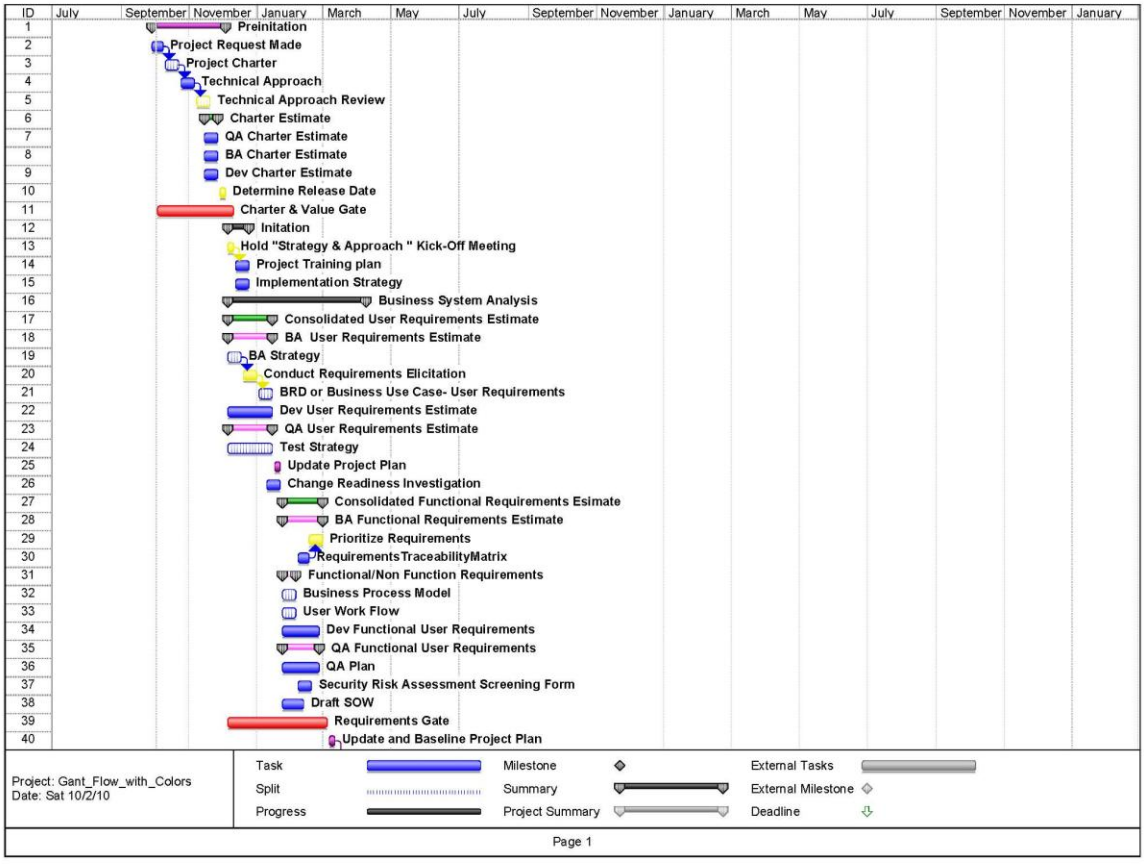

Figure 9: MS Project Representation of Current SDLC Page 1 (HTG, 2010)

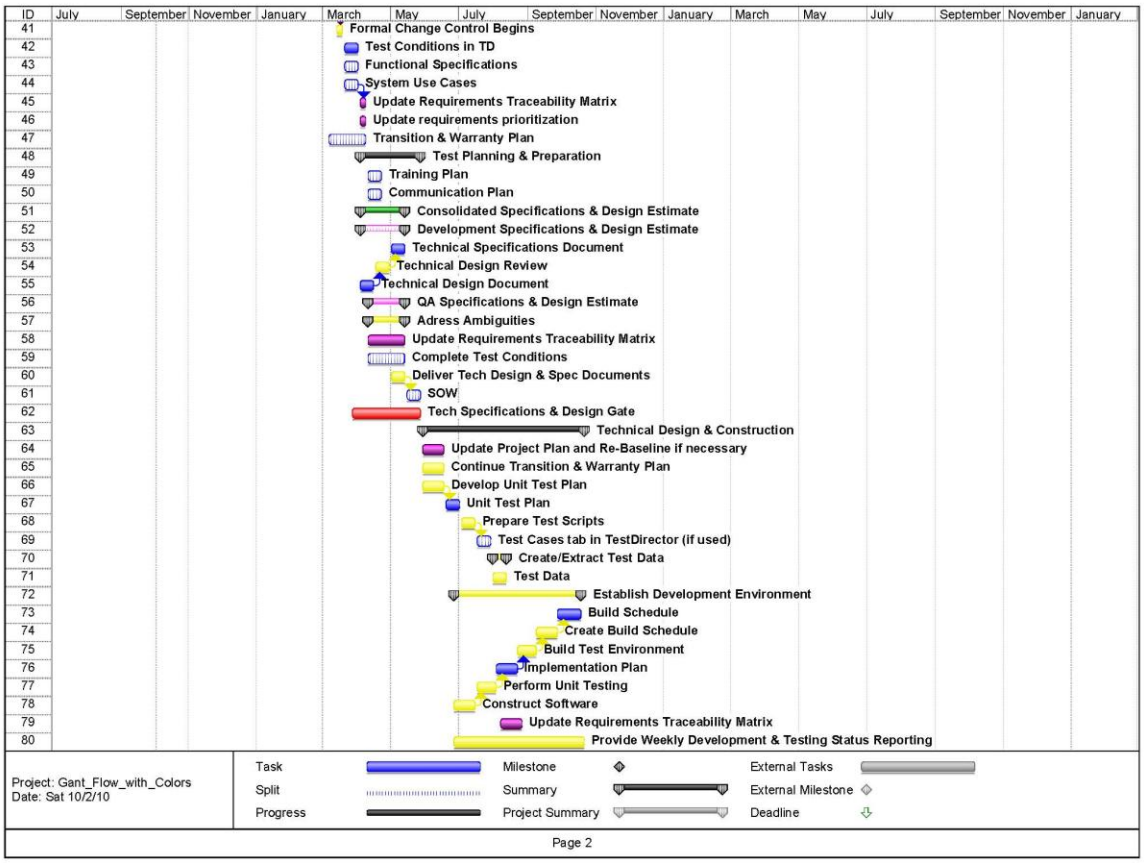

Figure 10: MS Project Representation of Current SDLC Page 2 (HTG, 2010)

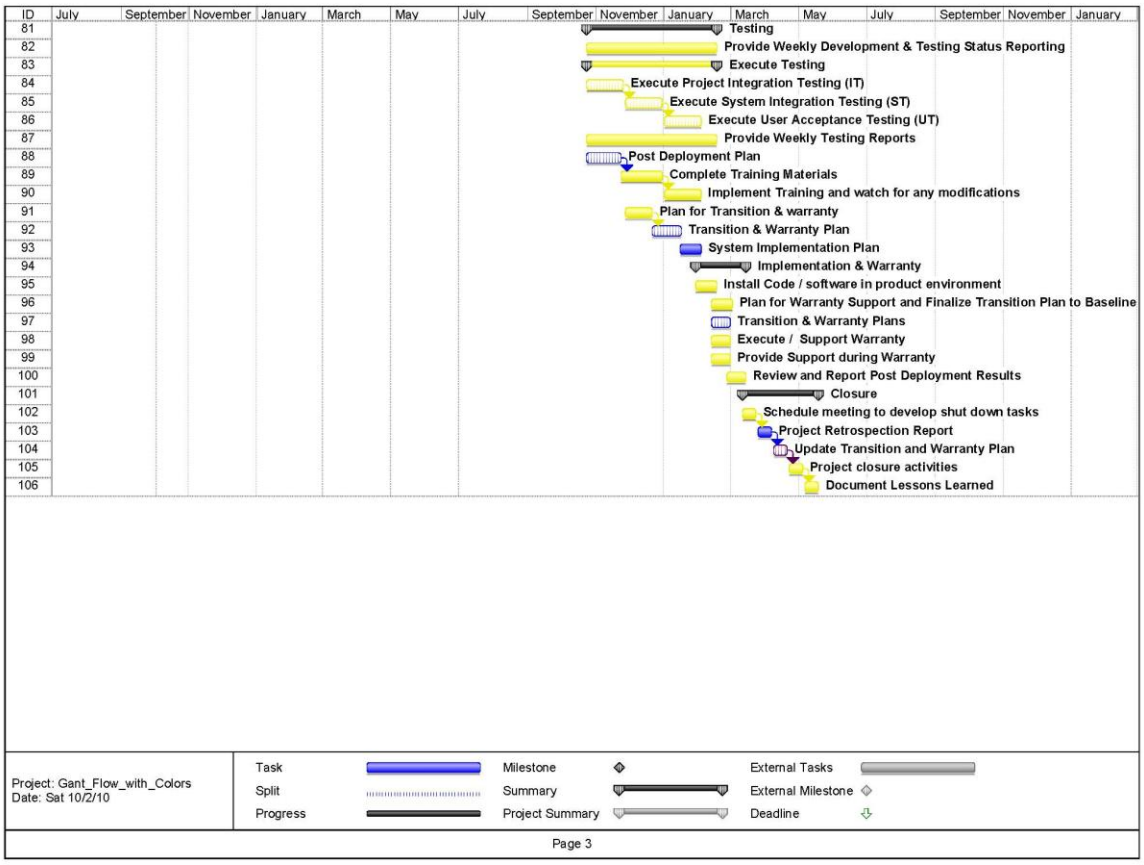

Figure 11: MS Project Representation of Current SDLC Page 3 (HTG, 2010)

4.2.4 Core Team Members

When a request for a project is made, a Program Director will request a Core PM (Project Manager) from the PM Practice Center (HTG, 2010). The Core PM will assemble the core team by requesting resources from the Practice Centers (HTG, 2010). The core team includes a Delivery Manager, QA Project Lead, Lead BA, Tech Lead, Solutions Architect, and a Data Architect (HTG, 2010). Each of these roles has its own responsibilities and documentation the role is required to produce (See Table 4).

Table 4: Core Team Roles and Responsibilities

Title	Role	Documentation Responsibilities
Project Manager	Manage the resources and the process in order to ensure completion of a project on time	Project Charter Charter Estimate Consolidated Requirements
Delivery Manager	Responsible for the front end work of a project, creating initial investment proposals and weekly status reports, working on the requirements and building out the development team	Statement of Work
QA Project Lead	Responsible for managing the quality assurance team and making sure that testing is done properly	QA Charter Estimate QA User Requirements Estimate QA Functional Requirements Estimate QA Strategy Complete Test Conditions QA Specifications and Design Estimate
Lead BA	Responsible for managing the subordinate Business Analysts and creating documentation that contains the requirements for the project	Business Requirements Document BA Charter Estimate BA Functional Requirements Estimate BA User Requirements Estimate
Tech Lead	Managing the Developers	Dev Charter Estimate Technical Specifications Development Specifications & Design Estimate
Architects	Responsible for managing the subordinate Architects for a project	Technical Approach Technical Design

4.2.5 Project Team Members

Once a project proceeds past the Charter and Value Gate, the Core PM, with assistance from the core team, requests additional resources from the BA, Architect and MIS Practice Centers (HTG, 2010). The resources requested are based on estimates created by the core team and work on the project as they are assigned. The resources required are Business Analysts, Architects, Developers, Quality Analysts/Testers, and an Implementation Manager. A summary of their roles and responsibilities can be seen in Table 5.

Table 5: Team Member Roles and Responsibilities

Title	Role	Responsibility
Business Analyst	Act as a liaison between business sponsor and HTG to ensure the business requirements are understood by the Architects and Developers	Create documents containing user requirements, functional requirements, non-functional/supplementary requirements, and business rules
Architect	Design the framework by which the developers will create the code Make high level design decisions based on requirements from BA's	Create the Technical Design document which "depicts the logical view of the application architecture, describes its logical layers and components, and identifies the associated system-level processes and the supporting infrastructure design required" (HTG, 2010)
Developer	Construct software based on the Technical Design document	"Complete all development work as requested, execute the unit test plan, complete the move memo and move all code to the production environment" (HTG, 2006).
Quality Analysts/Testers	Ensure all the proper testing is done and to create the test cases for a project	Execute testing procedures
Implementation Manager	Make sure delivered software does what it was intended to do for the business	Managing change, creating training programs and creating a Post Deployment Plan, communication, training, end user involvement, business sponsorship and lessons learned

4.2.6 Change Management Procedures

"All projects contain elements of change. Proper management of all changes in the scope of the project is essential. Therefore it is imperative that a formal change control procedure be in place and followed" (HTG, 2006). Change management must be followed once the BRD has been approved. The formalized process is done in twelve steps; however the PAB is currently in the process of assessing these procedures. The current process is depicted below (HTG, 2006):

1. Identify Change – the Core PM works with the team to identify any changes from the BRD once past the Requirements Gate
2. Estimate Impact – an estimate of the impact to resources, effort, cost, schedule, and risk
3. Document Change – the change is documented in a Change Request Form by the appropriate team member and given to the PM
4. Deliver to / Notify Approver – the Change Request Form is delivered to the appropriate entity
5. Obtain Approval – the PM obtains approval on the form
6. Log the entry – the change is documented specifying key information, dates and estimates for HTG
7. Receive from Approver – a decision is received
8. Communicate Change Status with Core Program Managers – all Core Program Managers affected by the change are notified
9. Update Weekly Project Status Reports/ Change Log – the change management section of a Weekly Project Status Report is updated
10. Update Project Folder – the change request is filed in the project folder
11. Review the Tailoring Matrix – if the change is significant enough to reclassify the project size, the documentation may change via the Tailoring Matrix
12. Execute the Change – if the change is approved, the PM assigns the appropriate team member to execute the change

APRs are completed by the PM and include the reason for the change, background information about the change, and the level of risk involved in accepting versus not accepting the change. Shown below in Table 6 are all the APRs from 2009-2010 that involved methods that can be considered agile, overlapping phases, or splitting the projects into tracks.

Table 6: APRs from 2009-2010

Number: Name	Size	Category	Phase	Year
4: PPL987 Rate Capping	Large	Phase Overlap	Tech Design/ Test Planning	2010
12: PEN080 Total Agency Compensation	Large	Phase Overlap	Initiation	2010
17: ICS040 HRIS Data Infrastructure	Large	Phase overlap	Business Systems Analysis	2010
27: PCL005 Agent User ID	Large	Phase Overlap	Tech Design/Tech Planning	2010
34: PEB025 AIX HCS	Medium	Phase Overlap	Tech Design	2010
36: PCL065 Loss Control Replacement	Large	Phase Overlap	Business Systems Analysis	2010
45: PCL067 CL Auto Vehicle type	Large	Phase Overlap	Tech Design	2010
46: PCL011 Automate Forms	Large	Phase Overlap	Business Systems Analysis	2010
6: PCL078 Claims OB Enhancements	Large	Tracks	Business Systems Analysis	2010
7: PCM010 June Claim Enhancement	Small	Tracks	Initiation	2010
21: PCL011 Tech Coverage Enhancements	Large	Tracks	Business Systems Analysis	2010
2: PCM801 Claims Wave II	Large	Phase Overlap	Test Planning	2009
9:HTG911 Plexus Replatforming Project	Large	Phase Overlap	Business Systems Analysis	2009
12: PCL936 Management Liability	Large	Phase Overlap	Business Systems Analysis	2009
25: PCL936 Management Liability Phase 2	Large	Phase Overlap	Business Systems Analysis	2009
28: PCM970- Bond Claims	Large	Phase Overlap	Business Systems Analysis	2009
7: ICS910 HRO Transition Phase II	Medium	Tracks	Initiation	2009
10: PMC945 Claims Medicare Secondary Payer	Large	Tracks	Pre-Initiation	2009
11: PCM920	Large	Tracks	Multiple	2009
36: PPL986 Geo Code for Renewals	Large	Tracks	Business Systems Analysis	2009
40: PCL979/PCL078 ACT Lighthouse Book Roll	Large	Tracks	Business Systems Analysis	2009

4.3 Analyze

The following section describes the analysis we performed on the data collected.

4.3.1 Interview Analysis

From the analysis of interview data, five major causes were identified that inhibit HTG's speed to market. These causes are resources, procedures, sponsors, change requests, and APRs. In the interviews conducted, many interviewees discussed these as causes to the main issue, reduced speed to market. The issue, causes, and reasons can be seen in the fishbone diagram illustrated in Figure 12.

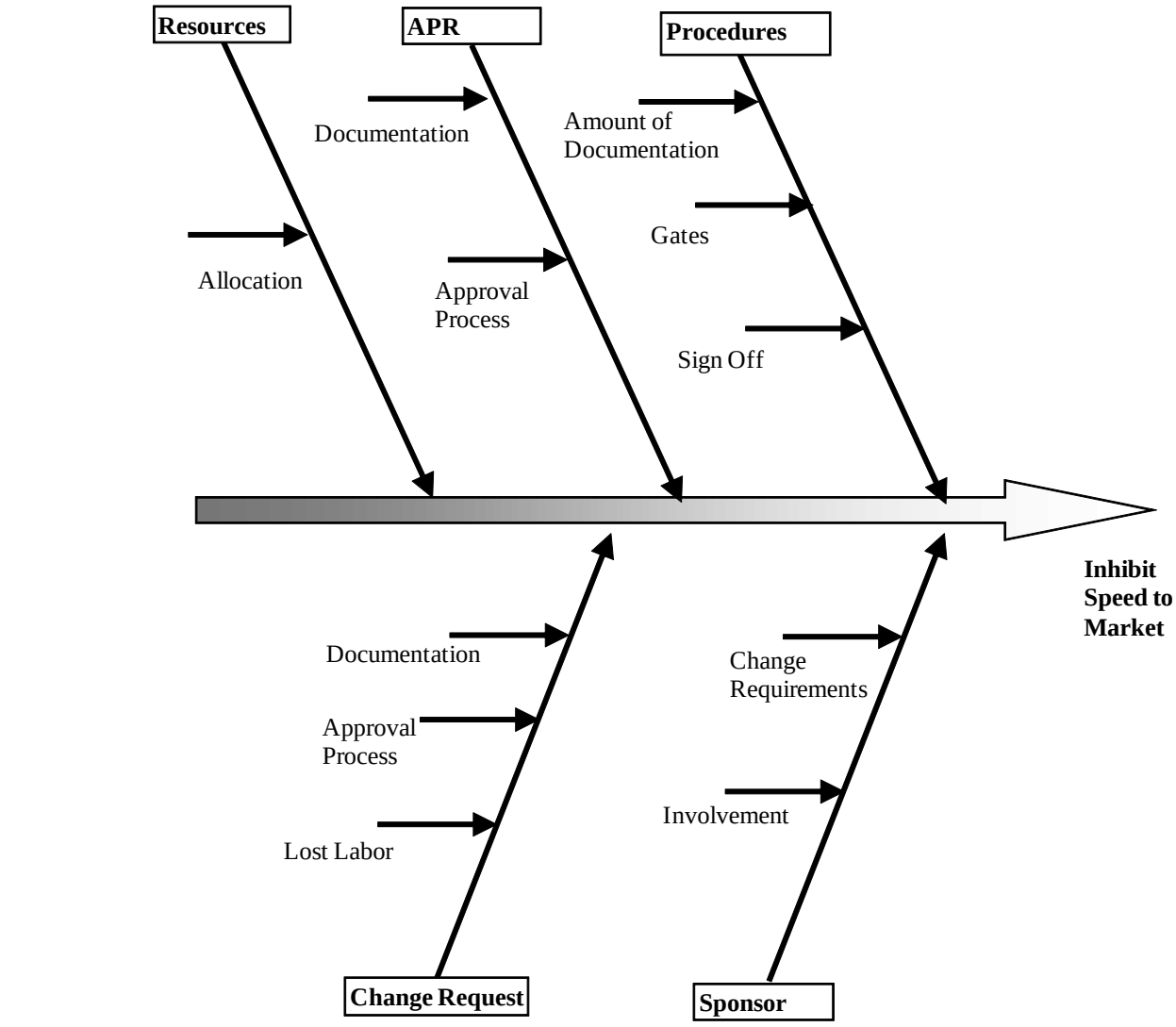

Figure 12: Fishbone Diagram

4.3.1.1 Resources

As shown in the fishbone diagram, the first cause is related to the allocation of resources. HTG Employees are assigned to work on several projects at once. This means that the employee is responsible for managing their time spent on each project and may have multiple conflicting due dates. As explained by a Lead Architect, when multiple projects are assigned all at once, it is difficult for employees to rank the importance of their projects and manage their time. For

example, an employee may feel that one project is more important than the others and allocate more time for that project and fall behind with others. This may result in other projects being delayed. Therefore, assigning a single employee to multiple projects can inhibit the speed to market.

4.3.1.2 Procedures

The second issue identified during the interview analysis was the SDLC process. The documentation, gates, and sign offs were identified by interviewees as reasons that inhibit speed to market. Some interviewees suggested that certain pieces of documentation could be combined to shorten the process. A Delivery Manager suggested that technical documents could be combined into one document to save time. Many interviewees suggested that the sign off process was time consuming and that it did not add value to the process.

4.3.1.3 Business Sponsor

Many interviewees mentioned the business sponsors as an important factor in the process. The main issues addressed were lack of business sponsor involvement in the project and changing requirements. A Lead Architect, a Senior BA and a Program Director all expressed that the business sponsor is not involved enough in the projects, which often leads to misinterpreting requirements or changing requirements after the BRD has been completed. A Lead Architect explained that requirements need to be clearer upfront. The issues with the sponsors involvement and changing requirements inhibits speed to market because project teams often have to revert back and make changes within the project when requirements change, this can take a lot of time especially if the requirements change is drastic.

4.3.1.4 Change Requests

The change request process was another issue identified in the interview analysis. Change requests are needed when there is a requirement change. Many of the interviewees felt that the change management process is very time consuming, or as put by a Senior BA, the "biggest headache". As discussed by both a DM and a PM, HTG needs a better way to manage changing requirements because requirements change can impact the project timeline and cost. Change requests reduce the project's speed to market because they are lengthy and can greatly impact the project.

4.3.1.5 APRs

APRs were a major issue identified by many interviewees, who expressed that almost every project has or should have an APR. Many interviewees voiced that initiating and getting permission to move forward with an APR takes too much time. A PM explained that it takes a lot of paper work to make a decision on APRs. An Enterprise Solutions Architect stated that the APR approval process needs to be more adaptable due to the fact that it takes too much time for an APR to get approved. The APRs create a problem for speed to market because if the project needs an APR they need to wait for approval of the APR before they can move forward with those changes.

All of the issues identified from the interview analysis are used in the Improve and Control phases. The information helped us in deciding what changes HTG would benefit from. All of these issues were kept in mind when formulating our recommendations. The suggestions table that was created during the interview analysis (see Appendix B) was also used when making decisions in the Improve and Control phases.

4.3.2 APR Analysis

A project can have more than one APR. During 2009-2010, a total of eighty-nine APRs were submitted, out of which twenty-four were for small projects, twenty-four were for medium projects and forty-one were for large projects. We focus on twenty-two APRs as they are the most relevant to the cause of reduced speed to the market. Some of the irrelevant APRs submitted requested to change the size of the project from small to medium because the amount of labor hours used was more than the estimated labor hours, or to change the documentation of the project. In addition, we focus on APRs that we categorized into request for tracks or phase overlaps categories because those are the types of APRs that affect the speed to the market. Tracks are similar to a structured parallel design methodology, where a project is divided into sub-projects and developed in parallel to each other. For phase overlaps, two or more phases in a project are performed at same time. Out of 22 relevant APRs, most APRs (19 in total) were submitted for large projects. There were two APRs for medium projects and one APR for a small project. Figure 13 shows all APRs submitted from 2009-2010 categorized by size and relevancy to our project.

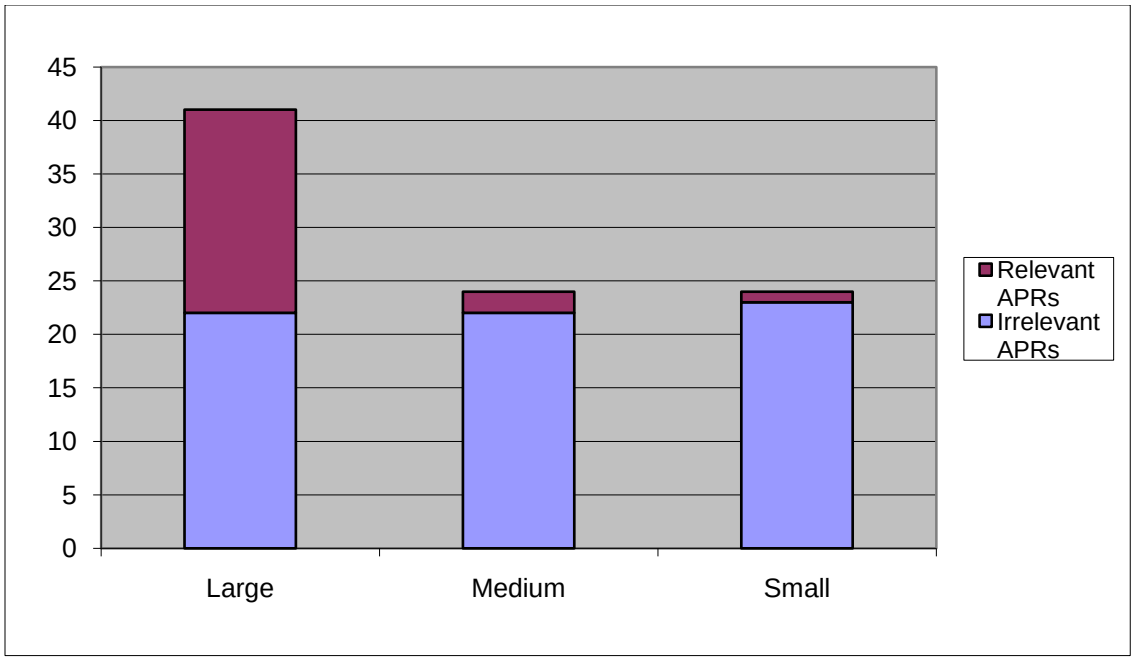

Figure 13: Number of APRs

We used a concentration diagram, which is a tool in Six Sigma used to identify defects. For this project, we define defect as changes to the process. The diagram is used to show in which phase APRs were most frequently submitted. In the concentration diagram (see Figure 14), the size of the project is represented by L, M or S for Large Medium and Small projects, and the type of request is represented by T for tracks and O for overlaps. We graphed the number of APRs against the Phases on X-axis.

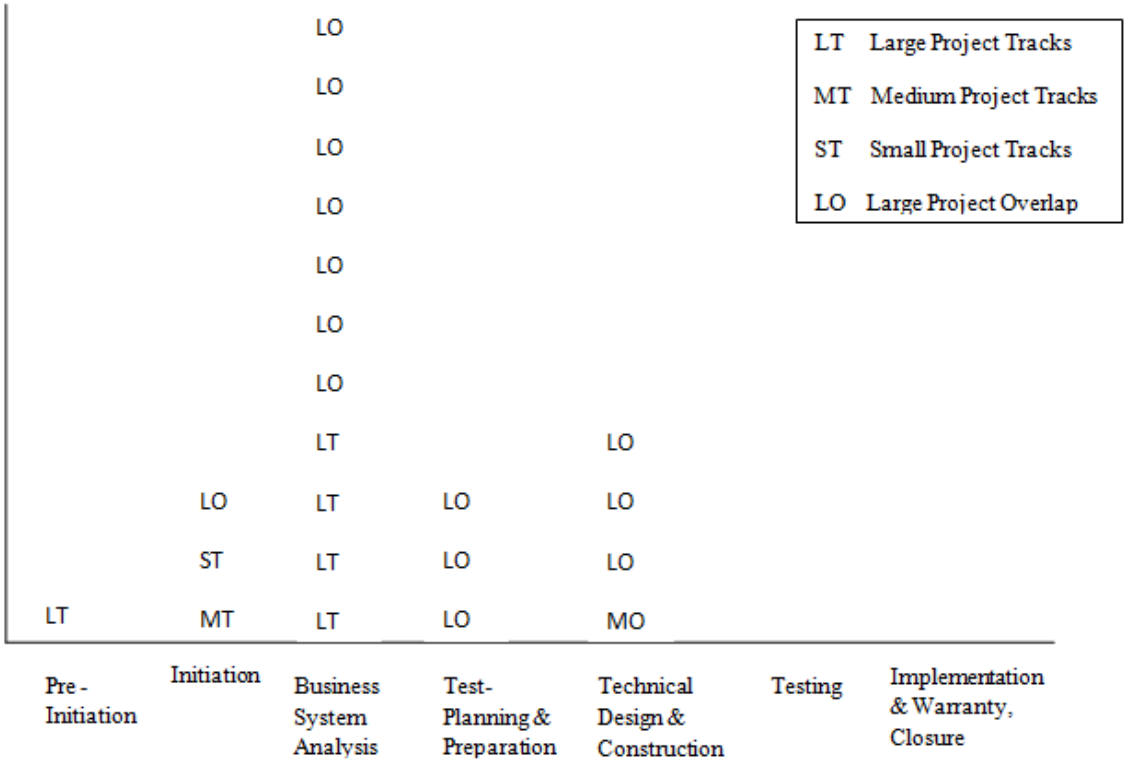

Figure 14 Concentration Diagram for APR

Looking at Figure 14, we conclude that the highest concentration of APRs occurs in the Business Systems Analysis Phase, totaling 11 APRs out of which 7 were large overlaps and 4 were large tracks. Only one APR was submitted in Pre-Initiation phase, three in the Initiation phase, three in the Test Planning & Preparation phase, four in the Technical Design & Construction phase and none were submitted in the Testing, Implementation & Warranty, and Closure phases. With this information it is clear that most APRs occur in the BSA phase, but there are also points in other phases where the decision could be made. This information was used to determine the evaluation points which will be discussed in Section 5.2.

5.0 Recommendations

The following section describes our recommendations for HTG. This includes both the Improve and Control phases for both recommendations.

5.1 Recommendation One - Scrum

Our first recommendation is for HTG to adapt a standardized agile methodology for certain projects. As described in the literature review section, agile methodologies eliminate most of the documentation and modeling. Instead, they focus on simple iterative development and allow the development team to easily adapt to any changes in requirements. There are many different agile methodologies, and various companies adapt the standardized practice to their own needs. Scrum is one of the most popular versions and easily adaptable to most organizations. For this reason we discuss Scrum and how it can be applied at HTG.

5.1.1 Improve

Scrum is an agile methodology that embraces the idea of cross-functional teams and delivering working pieces of software on a regular basis. There are three roles, three documents and three meetings in Scrum, as discussed in detail in Section 2.2.3.2. A representation of the process can be seen in Figure 15.

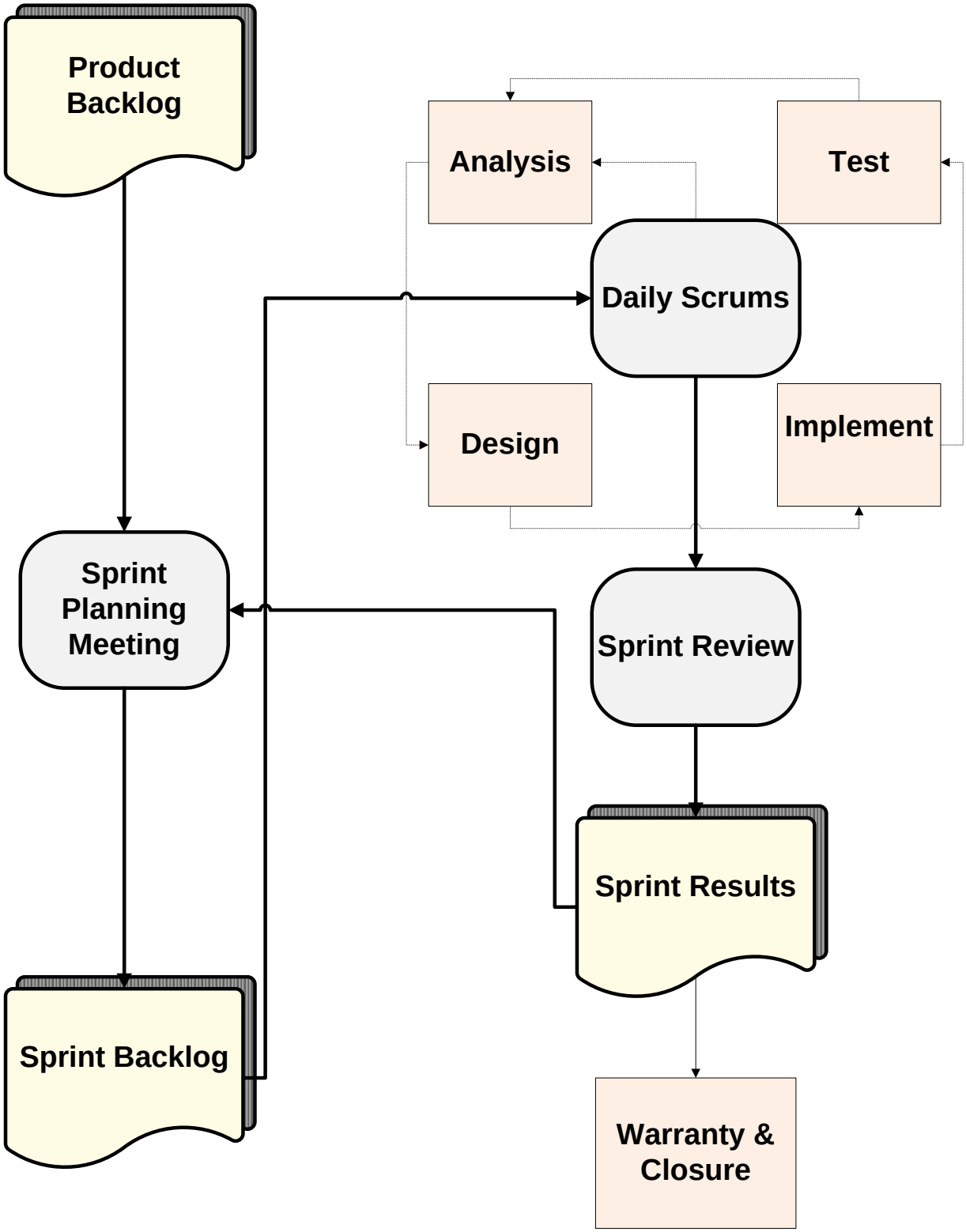

Figure 15: Scrum Process Flow and Documentation

For HTG, a team would be comprised of a Project Manager, a cross-functional team of resources and the Product Owner, a representative from the business unit. The Project Manager becomes the Scrum Master who is responsible for facilitating the Scrum process and general management activities that are similar to traditional Project Manager responsibilities. The team consists of 5-10 people with varying roles within the development process such as Business Analyst, Architect, Developer and QA Analyst. The Product Owner is responsible for identifying and delivering the requirements and ensuring that the project is what the business wants. Scrum begins by gathering the requirements and storing them in the Product Backlog and is created by the Scrum Master and Product Owner with input from the team. The work is prioritized by the Product Owner with the most important features ranking highest.

The next step in the process is where the iterations begin. For every sprint, there is a planning meeting where a Sprint Backlog is created that has a certain portion of the requirements to be addressed in the upcoming sprint. Every day during a sprint, a daily meeting is held where the Scrum Master checks the progress of the team. During each sprint, all the analysis, design, testing and implementation for each piece of the project is completed. At the end of each Sprint, a Sprint Review meeting is held to show the results of the Sprint and demonstrate the features completed in the Sprint. If there are more requirements in the Product Backlog, then a new Sprint Planning meeting is held. If there are no more features to complete, the project enters the Warranty and Closure phase.

5.1.1.1 Barriers and Enablers to Scrum Adoption at HTG

Change in a work environment is always difficult, and changing the SDLC can be particularly challenging. Structured design models such as waterfall follow a set process and have large amounts of documentation to guide the work, where as agile methodologies allow for

more freedom in design and are focused on teamwork. Adopting a new set of procedures will require not only a change to the process, but a cultural shift. If HTG wants to follow Scrum, there are certain practices that must be modified in order to have the potential for success. HTG possesses certain enablers that will contribute to success of transitioning to Scrum. The barriers and enablers that HTG has were identified through analysis of the interviews we conducted. The barriers that HTG will face are: location of teams/creating an agile work environment, allocation of resources, training, business sponsor commitment, and a change in cultural style. The enablers are: having a certified Scrum Master on staff, a large on-site staff and the fact that HTG has used certain agile techniques in previous projects.

5.1.1.1.1 Location of Teams/Work Environment

One of the core concepts of Scrum is having a co-located team. During each sprint, the team works together in one area completing all the analysis, design, testing and implementation work for each piece of the project. This is done best in an agile style work environment, which would be a room that has whiteboards, computers and enough space to collaborate as a team. Currently HTG has people in various locations of the building and some developers and QA engineers working offshore. This poses a problem, but it can be overcome.

CampusSoft explains how they used offshore teams in Scrum in the conference paper (Summers, 2008) *Insights into an Agile Adventure with Offshore Partners.* CampusSoft is based in the U.K and had almost an even split of on and offshore resources and were able to successfully implement Scrum by modifying certain key aspects of the process. The onshore team created the Product Backlog and from there the work was divided. They had their Sprint Planning meetings utilizing video conferencing including the Product Owner, the Scrum Master and all team members, both on and off shore. They held the Daily Scrums using video

conferencing with everyone at their seats using a headset. The Sprint Review was done differently, all onshore personnel would gather in a conference room as would the offshore team simultaneously. The barrier of location was overcome by the use of video conferences, something HTG could easily implement.

Creating an agile room would be simple, there are a lot of conference rooms at Hanover, and one of them could be converted into an agile room. The issue of co-location is slightly more difficult. For people who work on-site in Worcester, their location could be moved and if an agile room was created, most of the work would be done there; however, between 30-50% of current development work is done offshore, and about 50-70% of testing is completed offshore.

There are two solutions to this problem: using only on-site developers and QA engineers or adapting the methodology to fit. The major benefit of using on-site resources is the team would be able to learn the process as it was designed. This would set a foundation for future projects that may include offshore resources. The key difference is that the offshore team would not participate in the meetings. The process would be the same except that explicit analysis documents would need to be created for the user stories that the offshore resources receive. At first, HTG should only select on-site resources for Scrum projects, but as they become more comfortable with the process, the inclusion of offshore resources becomes possible.

5.1.1.1.2 Allocation of Resources

A core concept in agile is having a team fully allocated to one project; however HTG currently has the need to have resources working on multiple projects at a time. This is especially important to the developers because they are specialized in one or two programming languages. Projects have the need for different developers because they may require development work in multiple programming languages. According to a Tech Lead we interviewed, this can be

overcome by training developers on new technologies or by pairing junior developers with senior personnel to teach them multiple languages. Having off-site developers is a benefit because they can be reallocated more easily than the fully allocated team on-site. For a pilot project, choosing one that only uses one or two programming languages that are common among developers at HTG would be the best choice because it would allow HTG to follow the process better initially, and then make changes if necessary.

5.1.1.1.3 Training

To be successful in agile, proper training must be given to all those involved. The Project Manager would now become a Scrum Master and has the most important role in the process. Training for this is available online or there are seminars and training courses that could be taken. The team members also require training so that they are familiar with the process and how they will be completing their work in the future. For example, the website http://agiletraining.com/agile-training/ provides webinars for training or if HTG wanted to have employees attend a live event, http://www.agileuniversity.org/ offers courses in various areas, specifically they go to Boston which would be convenient for HTG team members. It is very important to properly train all those involved before initiating a Scrum project.

5.1.1.1.4 Business Sponsor Commitment

One of the reasons agile methodologies were developed was to deliver better software to the client. This is accomplished by including a representative from the business, who has a clear understanding of what the software should do, in the process. This representative attends the initial meeting to draft the requirements and complete the product backlog, and subsequently attends all of the sprint results meetings to see what has been completed. This is important to the

process because it allows the business sponsor to ensure that the completed parts of the software are what he/she wants and allows them to make changes if necessary.

HTGs current relationship with their business sponsors varies by business unit. Some do not view HTG as a strategic business partner, rather a cost center. Having sponsor involvement is important to the agile process, and in order to have a successful project, choosing a project for a business unit that has a good relationship with HTG would be best. Once HTG establishes the benefits of involving a representative from the business more than they do now, other business units would be more likely to engage in these activities.

5.1.1.1.5 Cultural Changes

Many HTG employees have been working there for years, have used the same process and followed the formal documentation standards. A change to this might be difficult for some to adapt to. For this reason, the best candidates for a pilot project would be younger or newer employees because they are not dedicated to the current process, or people who are willing and excited to try a new SDLC.

5.1.1.1.6 Enablers

Currently HTG has a certified Scrum Master on staff and is part of the PAB. Having someone who has worked in Scrum before and is certified in managing a Scrum team provides the benefit of having an experienced individual to help facilitate the transition and make sure that the process is correct. As mentioned previously, it would be beneficial to HTG to use only on-site resources at first before including offshore resources. The current HTG staff is large enough to support this. Through analysis of the APRs, we have seen that certain requests were to use agile techniques such as iterative development. By having teams that are making these kinds of requests, it shows that the organization has potential for success with using Scrum.

5.1.1.2 Pilot Project

Not every project fits the qualifications for using Scrum and to begin a pilot project with specific criteria should be chosen. The pilot project should:

- Use only personnel located on-site at Hanover and they should be co-located on the same floor
- Have a dedicated room
- Have all resources fully allocated to the project
- Only use resources that received proper training before beginning the project
- Include people who are willing and excited to experiment with a new methodology
- Be for a business unit that is willing to be involved or an internal HTG project
- Be of small-medium size

According to a PM, the reality is that because HTG operates for Hanover Insurance, the likelihood of being able to go full Scrum is small. There are aspects of Scrum that would be helpful in reducing the life cycle time, they are: putting an emphasis on teamwork and giving the team more ownership of the project, involving the business sponsor more, and allowing for iterative development if necessary. These will be discussed in Chapter 5.2 with respect to the hybrid methodology we developed.

5.1.2 Control

For Scrum, there are two metrics that should be measured to ensure the process is successful. Agile methodologies were designed to decrease the development time, so the most important metric is the time taken to complete a project. Currently HTG does not have accurate data on the time it takes for a project to go from start to finish due to the fact that resources are allocated to multiple projects. Since Scrum requires a team to be fully allocated, it will be much easier for HTG to track the time it takes to complete a Scrum project. This should be compared with an estimate of what it may have taken to complete the project using the waterfall system currently in place. The PAB would be responsible for keeping track of this data.

The second metric to measure success is the acceptance by team members and adherence to the process. A team should be properly trained in Scrum before beginning a pilot project, but that does not mean that they will be adept in the process. Transitioning from a document-driven methodology to a team based environment will be challenging for some, but if they are satisfied with Scrum, then it can be deemed a success. To measure this we created two short surveys to be completed after the pilot project, for both the team members and the business sponsor. The surveys address the areas that were slowing down the development time while using the waterfall SDLC (Appendix C). The questions are straightforward and measured on a scale of 1-5 so they can be easily compared side-by-side in a bar chart. An example of what this may look like can be seen in Figure #16.

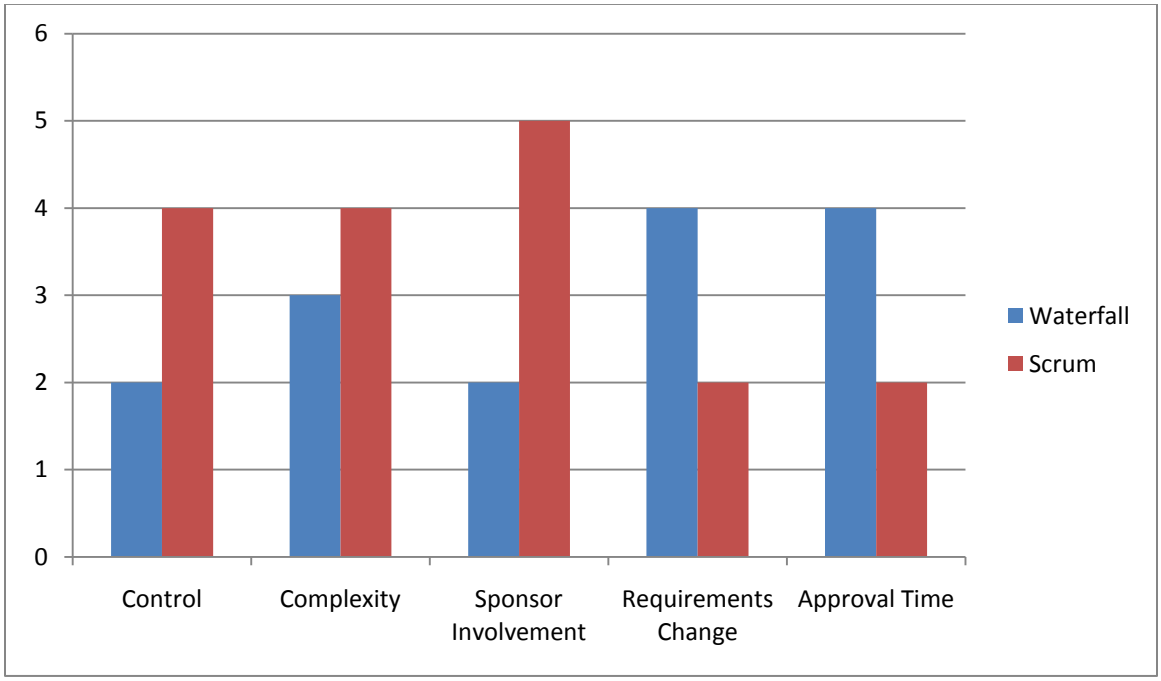

Figure 16: Survey Comparison between Waterfall and Scrum

This data can be used to see if the process was an improvement in the major areas that we were addressing by recommending Scrum.

5.2 Recommendation Two - Hybrid Methodology

This methodology incorporates features from Scrum into the current methodology in practice at HTG.

5.2.1 Improve

This solution was created by incorporating the APRs and interview analyses. The main changes made to the original SDLC were the removal of two gates, the addition of evaluation points, and transitioning to the use of evolving documentation.

5.2.1.1 Evaluation Points

The major change in the playbook is the addition of evaluation points. The evaluation points will work as checkpoints and as a point to split the project into tracks or to make the decision to overlap phases. At each evaluation point, the core team and business sponsor will meet to discuss the progress of the project. This allows the core team to take ownership over the project. The core team may want to involve members of the PAB at these meetings as well, this will allow the PAB to be able to see the progress of the project and how the team plans to complete the project.

At these meetings, the team must check the scope of the project, and decide if the project can be broken into tracks or if certain phases can be overlapped. The addition of these evaluation points eliminates two gates. This should reduce 25% of the APRs that we found were submitted asking to change the current SDLC at the points where we have placed the evaluation points. If the project team decides to move forward with tracks or phase overlap, they will only complete evolving documentation for each track. The documentation will be complete when the document is updated to include information about the last track. The flow for the hybrid methodology can be seen in Figures 17-19. In these figures the white documents represent the original process, the

yellow triangles represent the evaluation points, the green shading indicates an evolving document and the red triangles indicate sign offs.

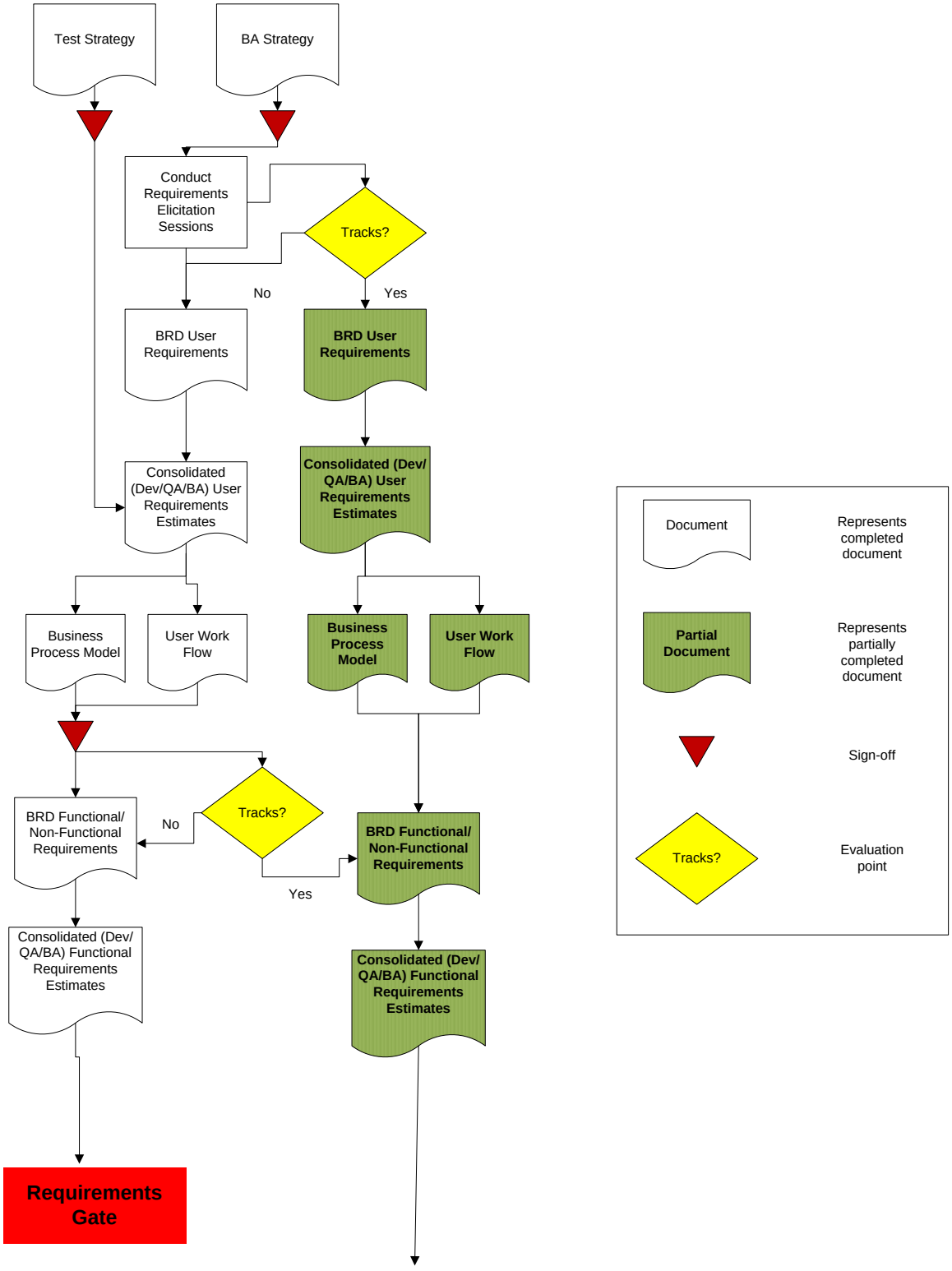

Figure 17: Hybrid Methodology Page 1

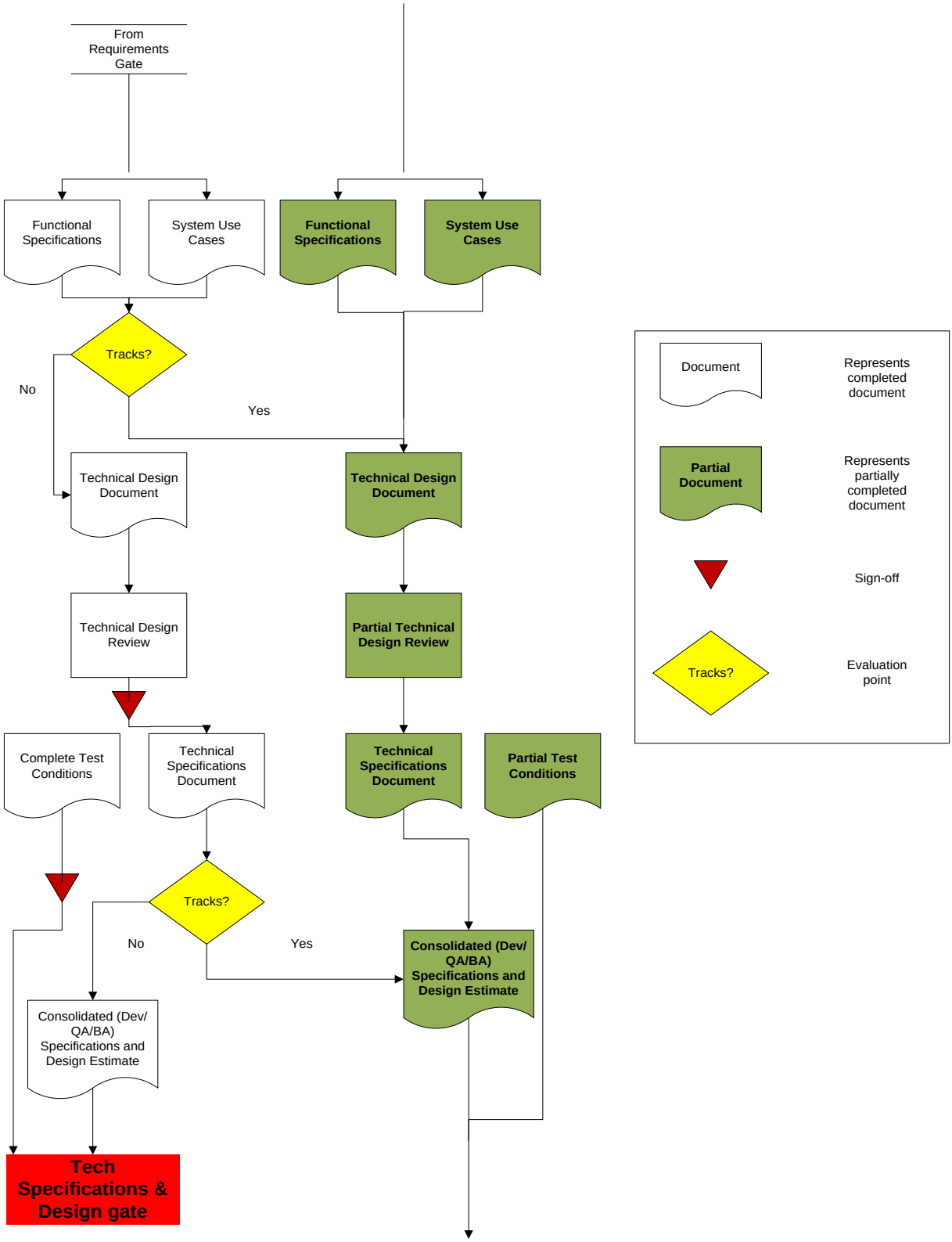

Figure 18: Hybrid Methodology Page 2

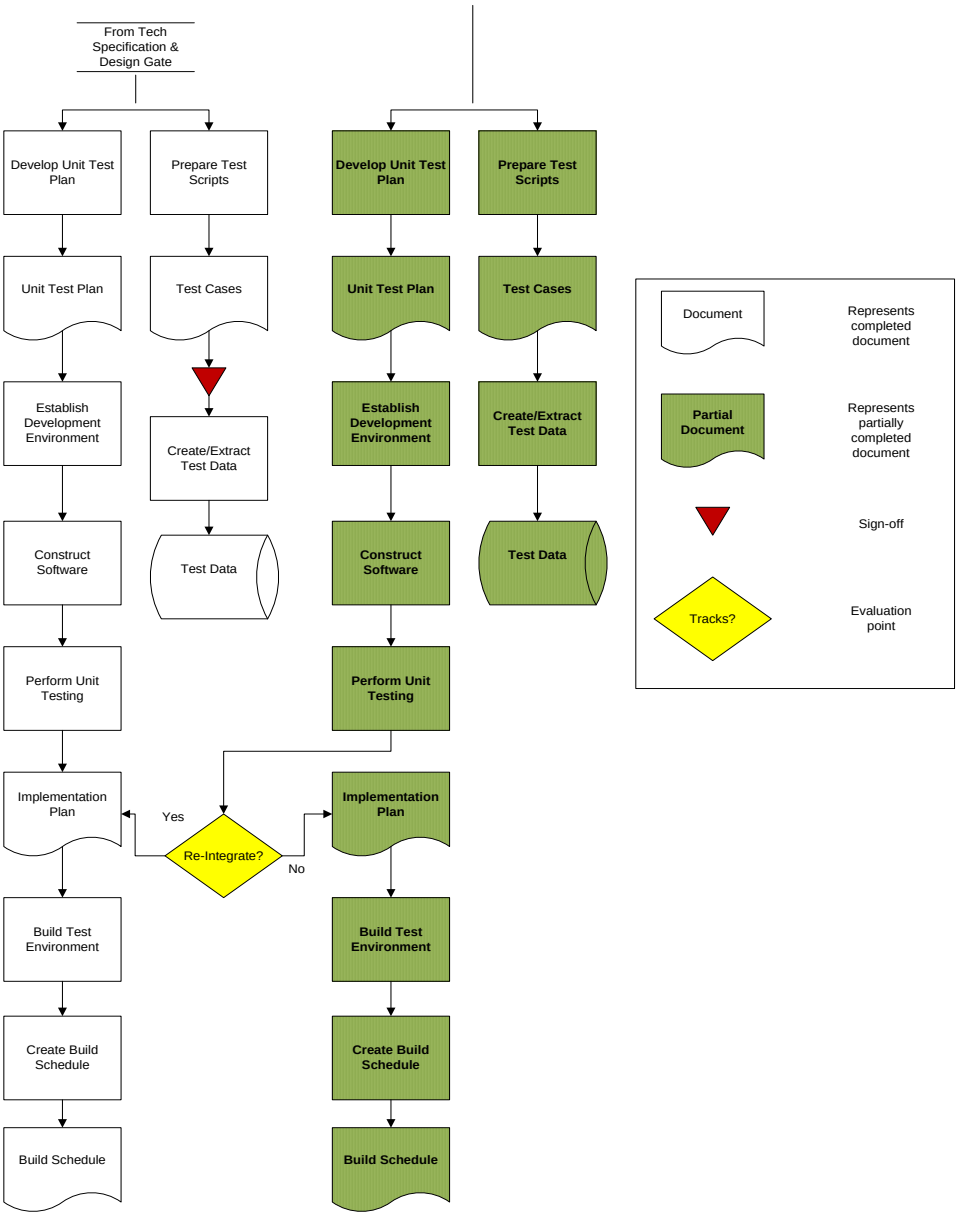

Figure 19: Hybrid Methodology Page 3

5.2.1.1.1 Evaluation Point 1

The first evaluation point is located after the requirements elicitation session. This location was chosen because the core team may feel that they have enough requirements to proceed with one part of the project, one track, while they may need to work more with the sponsor with other parts of the project. If the team decides to move on with the track, they will create the BRD User Requirements for this track and then move forward with other project documentation. When the requirements for other tracks are set those tracks will then be added to the BRD User Requirements and will continue to move through the project as its own track.

5.2.1.1.2 Evaluation Point 2

The second evaluation point is located before the BRD Functional/Non-Functional requirements document is created. This location was selected because at this point in the project process, the functional and non-functional requirements of one track of the project may be defined while they may be unknown for others. The track with the defined functional and non-functional requirements will continue with the project process while other tracks will continue on with the process after the functional and non-functional requirements are established.

5.2.1.1.3 Evaluation Point 3

When the Functional Specifications and System Use Cases are being drafted there is a third evaluation point. The third evaluation point is located here because at this point one track may have all its specifications and use cases while another may need more work. This allows for part of the project to move along to Technical Design while the specifications and use cases are developed for another track of the project.

5.2.1.1.4 Evaluation Point 4

The fourth evaluation point is located after the Technical Specifications document. This location was chosen because at this point one track may be able to move into development while other parts of the project may need more work in technical design.

5.2.1.1.5 Integration Point

The final evaluation point is different from the others because the core team is not deciding if the project should be broken into tracks, instead the core team is deciding if the project will be integrated as one project or if the tracks should be released separately. The location for this point is right before the integration plan because the decision must be made in order to proceed with the integration plans.

5.2.1.2 Flow with Evaluations Points

The flow of the new playbook is different from the current waterfall SDLC. With the evaluation points there will be loops within the playbook. Where the loops begin will depend on which evaluation point split the project into tracks. Once the project is ready to begin with a new track, the track must begin the project process at the evaluation point where the project was broken into tracks and then continue on with the rest of the project process.

5.2.1.3 Evolving Documentation

As mentioned in the Improve section, the evolving documents would be built for each track and would be updated as the project moves on to next track. To make it easy to record, document and retrieve them at any point, all the evolving documents should be uploaded on SharePoint, and should be updated as the project builds. Auditing procedure for the projects would not change, as we are recommending use of all current documentation. Through our

interviews, certain documents were deemed unnecessary by some, but critical by others. As a result, we are not recommending the removal of any current documentation.

All the documents should have a standard naming convention, so it is easy for people referring/updating the document to know what information each document contains. The recommended naming convention for each document is *DocumentName_Track# (BRD_Track1)* or *DocumentName_Version1.X(Y%) (BRD_Version1.1 (30%).*

5.2.1.4 Implementation Plan

As discussed in the Chapter 3.0 we were not able to implement our recommendations due to time constrains. In this section we will provide suggestions for HTG on how to implement the hybrid methodology.

First HTG should pick a few projects and project teams to run a pilot project using the hybrid methodology. When choosing the project team, HTG should pick employees that are willing to try something new. When picking the pilot project, HTG should select a project that would work well if broken into tracks.

The next step HTG should take in implementing the hybrid methodology is to train the project teams on how to use the new playbook. The training should include an explanation of the playbook, the evaluation points and their purpose, as well as information about how to use and name the evolving documentation.

The PAB should be closely involved with the pilot projects, and be included in the evaluation point meetings. This will let them see how well the team is adjusting to the new process and if the new process is making development more efficient. It will also allow the PAB to make sure that the new process is meeting their goals.

As with all pilot projects, the first one may not be as successful as hoped, however there will be lessons learned from each pilot project. These lessons should be applied to subsequent projects using the hybrid methodology to continuously improve the process. Section 5.2.2 will discuss how to measure the success of these and future projects using this hybrid methodology.

5.2.2 Control

Some controls have been listed below that can be put into place to sustain the gain. Once the recommended SDLC has been implemented, we measure the process for its success.

It is important to monitor the new process to measure the success and modify it if the results are not up to specifications. The first metric to monitor the success is to count the number of APRs submitted while following it. The number of APRs submitted can be compared to the number of APRs submitted for projects that strictly follow the waterfall methodology. The second metric to measure success would be time it takes to build a project. Currently HTG does not have any accurate data on time it takes to build a project. To overcome this, HTG can measure the time it takes to build a project using their current system and compare it with recorded time of the project following the recommended hybrid methodology. The PAB would be responsible for recording these data. The third metric would be the satisfaction with the new process, which can be done by using the recommended short survey form for the team members and the business sponsors (Appendix C). These surveys address the major points of change measured on a scale of 1-5. The data collected from the survey can be analyzed side-by-side using a simple bar chart, an example of this is shown below in Figure 20. We have not mentioned anything about the documentation in the process monitoring plan, as we did not change or eliminate any current SDLC documents. The audit process would be same as the one for current waterfall methodology.

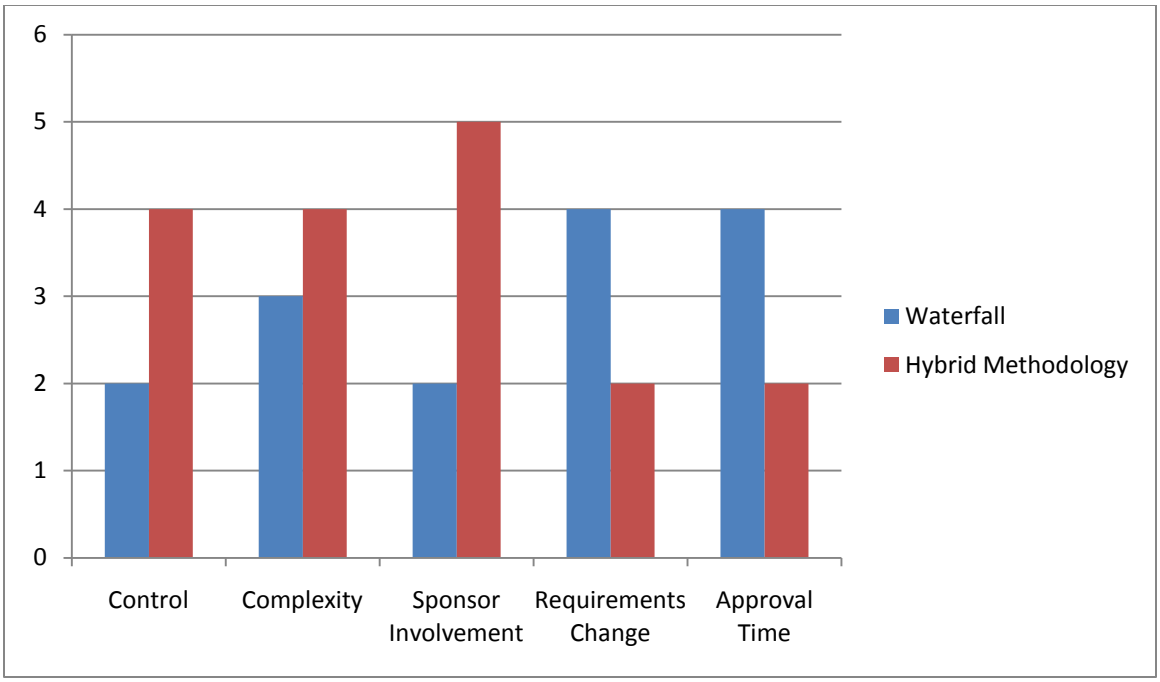

Figure 20 Survey chart

Satisfaction would be achieved if there is decrease in the time it takes to complete the project and zero APRs are submitted for the tracks and phase overlaps in project (as defined in the paper). The possible red flags in process would be avoiding the evaluation points. It is the PM's job to make sure they do not mix evaluation points with current gates, and ensure that all meetings are held to the standards previously described. The business sponsor involvement is one of the critical success factors for the new SDLC. The PM is responsible for ensuring that the business sponsors are involved in the project. It is important to maintain standard naming conventions for the documentation so the content of each document is clearly defined.

Finally, since further change in the process environment is inevitable, the project team should develop a process for updating the new procedure when required. HTG currently has a process for that called Continuous Process Improvement (CPI) and should continue to use this system to request any changes.

6.0 Reflection

6.1 DMAIC

We used DMAIC (Define, Measure, Analyze, Implement and Control), a Six Sigma tool, to approach this project. We defined our problem statement as follows: To increase HTG's speed to market through making the process more flexible. To measure the current process at HTG, we collected the data from personal interviews with HTG personnel and documentation from within HTG. In this project we designed process flow diagrams for both of our recommendations as well as metrics to measure the success of implementing the recommendations.

We used Six Sigma tools, a fishbone diagram and a concentration diagram, to analyze the data. The fishbone diagram was used to analyze the interviews and identified five major causes that inhibit the HTG's speed to market. These causes were resources, procedures, sponsors, change request, and APRs. The APR analysis was done using the concentration diagram, a Six Sigma tool to identify the defects in process. We defined defects as a change in process. We summarized all of the APRs in excel sheets and selected the APRs that were most relevant to the project. After that, a concentration chart was made to determine which phase APRs were most frequently submitted.

The information gathered from the analysis was used to make recommendations. For our improve section, we proposed two sets of recommendations for HTG, implementing Scrum under specified criteria and using a hybrid methodology. Normally in DMAIC, the improve phase involves implementing the new systems, however due to time constraints we were unable to implement our recommendations. Ultimately for the control section, we developed metrics to measure the effectiveness of the implementation. Some areas that were measured are time, team members/sponsors satisfaction with new SDLC and the number of APRs that are submitted.

6.2 Constraints

From our measure and analysis phases, we found several constraints. The constraints identified, as described more in-depth in Section 5.1.1.1, were location of employees and work environment, allocation of resources, sponsor commitment, and cultural challenges. For the first recommendation of implementing Scrum to be successful, HTG would have to overcome all of these constraints. With the difficultly of overcoming all of these constraints at once, we decided to make a second recommendation, a hybrid methodology, that would only require HTG to overcome the constraints of sponsor involvement and some of their cultural challenges.

6.3 Design Discussion

For the design component, we created two sets of recommendations that HTG can use to increase their speed to market.

There were two sets of recommendations which HTG can implement. Our first recommendation for HTG was to adopt Scrum under certain criteria. As discussed in Section 2.2.3.2 Scrum follows the rule of threes, it has three documents, three meetings and three roles in projects. These roles, documents and meeting for Scrum are defined in detail in Section 1.7. HTG can implement Scrum under certain conditions, the barriers and enablers to Scrum adoption at HTG are discussed in detail in Section 5.1. To select a pilot project for Scrum, HTG should follow the criteria of selection discussed in Section 5.1.1.2. The results from the first project may not be as expected, but HTG can use the metrics to measure success from Section 5.1.2 and update their process as required.

The second recommendation for HTG is to implement the hybrid methodology, an adapted version of the current HTG playbook that includes traits from the Scrum and addresses the issues from Section 5.1.1.1. The changes made in the new playbook are discussed in detail in

Section 5.2. Section 5.2.2 discusses metrics to measure the success of the recommended playbook. The implementation plan for the hybrid methodology can be found in Section 5.2.1.4. The criteria to measure successful implementation are reduction in time required to develop software, reduction in the number of APRs submitted and the satisfaction of team members/sponsor using the recommended version of playbook. The recommended process should be updated constantly when required.

For each recommendation, we have suggested an implementation plan and provided metrics to measure the success of each recommendation. Both implementation plans include ways to select and train project teams. These implementation plans can be found in Sections 5.1.1 and 5.2.1.4.

7.0 Conclusion

In order to reduce HTGs speed to market, we developed two process redesign recommendations; a set of criteria for selecting a pilot Scrum project and how to implement Scrum, and an adapted version of the current playbook we define as a hybrid methodology. HTG has the ability to adopt Scrum, but the project should be selected carefully as some projects may not be appropriate for Scrum. By implementing the hybrid methodology HTG can reduce their APRs by 25% based on previous years APRs. This will also allow the core team to take ownership of the project.

These are not the only potential options for HTG to move forward. They can also look into a more RAD oriented SDLC with prototyping, or different type of agile methodology as this is not something we researched in depth. HTG can refer to the standard SDLC practices in the literature review (Chapter 2) and explore the potential areas for improvement in future projects to broaden their options of development to suit the business needs allowing more flexibility to support the business.

8.0 Bibliography

Alexander, L. C., & Davis, A. M. (1991). Criteria for selecting software process models. Paper presented at the *Computer Software and Applications Conference, 1991. COMPSAC '91., Proceedings of the Fifteenth Annual International*, 521-528.

Alexandrou, M. (2010). System develpment life cycle(SDLC) Retrieved from .http://www.mariosalexandrou.com/methodologies/systems-development-life-cycle.asp

Andersson, D. (2006). Software development methods. Retrieved from http://www.microansa.com/it/software_development_methods_by_daniel_andersson.pdf

Bergstrom, A. (2008). *Sofware Confugration Management in Scrum Projects*. Computer Science, Lund Institute of Technology. Retrieved from http://www.cs.lth.se/EDA920/2008/Rapport_2008-33.pdf

Chow, T., & Cao, D. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software, 81*(6), 961-971. doi:DOI: 10.1016/j.jss.2007.08.020

Dennis, A., Wixom, B., & Roth, R. (2006). *System analysis and design* (3rd ed.) Hoboken, NJ: John Wiley & sons, Inc.

Drew Procaccino, J., Verner, J. M., Overmyer, S. P., & Darter, M. E. (2002). Case study: Factors for early prediction of software development success. *Information and Software Technology, 44*(1), 53-62. doi:DOI: 10.1016/S0950-5849(01)00217-8

HTG (2005). *Introduction to BA v1-0.* (Technical Document Hanover Insurance)

HTG (2006). *Change management procedures v2-1.* (Technical Document Hanover Insurance)

HTG (2006). *Developer project organizer v2-0.* (Technical Document Hanover Insurance)

HTG (2006). *IM post deployment assessment procedures v2-0.* (Technical Document Hanover Insurance)

HTG (2006). *Technical approach v2-0.* (Technical Document Hanover Insurance)

HTG (2007). *Gate processes and structure v4-0.* (Technical Document Hanover Insurance)

HTG (2007). *Statement of work v4-1.* (Technical Document Hanover Insurance)

HTG (2010). *Project playbook v5-2.* (Technical Document Hanover Insurance)

HTG (2010). *Tailoring matrix v7-1.* (Technical Document Hanover Insurance)

HTG (2010). *Technical design v2-0.* (Technical Document Hanover Insurance)

III (2010). Insurance 101. Retrieved from http://www.iii.org/article/insurance-101/

Ilieva, S., Ivanov, P., & Stefanova, E. (2004). Analyses of an agile methodology implementation. Paper presented at the *Euromicro Conference, 2004. Proceedings. 30th*, 326-333.

Jacobs, F. R., Chase, R.B.,& Aquliano, N.J. (2006). *Operations & supply chain* management (12th ed.). New York, NY: The McGraw-Hill Companies, Inc.

Juric, R. (2000). Extreme programming and its development practices. Paper presented at the *Information Technology Interfaces, 2000. ITI 2000. Proceedings of the 22nd International Conference on*, 97-104.

Kasser, J. (2002). The cataract methodology for system and software acquisition. Retrieved from http://www.seecforum.unisa.edu.au/Sete2002/ProceedingsDocs/03P-Kasser.pdf

Kimmond, R. M. (1995). Survey into the acceptance of prototyping in software development. Paper presented at the *Rapid System Prototyping, 1995. Proceedings., Sixth IEEE International Workshop on,* 147-152.

Lewis, J., & Neher, K. (2007). *Over the waterfall in a barrel - MSIT adventures in scrum* Retrieved from http://xml.engineeringvillage2.org/controller/servlet/Controller?CID=expertSearchDetailedFormat&EISESSION=1_d1e23312b63b1619761bcses3&SYSTEM_USE_SESSION_PARAM=true&SEARCHID=29d29412b5921c4205c4fprod4data1&DOCINDEX=1&PAGEINDEX=1&RESULTSCOUNT=1&database=1&format=expertSearchDetailedFormat

Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869-1890. doi:DOI: 10.1016/j.jss.2009.05.052

Munassar, N., & Govardhan, A. (2010). A comparison between five models of software engineering. *International Journal of Computer Science , 7* (5). p. 94-101

Mun-Wai Chung, & Drummond, B. (2009). *agile at yahoo! from the trenches* Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5261097

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM, 48*(5), 73-78. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=16915874&site=ehost-live

Stober, T., & Hansmann, U. (2009). *agile software development best practices for large software development projects* Springer Heidelberg Dordrecht London New York.

Summers, M. (2008). Insights into an agile adventure with offshore partners. Paper presented at the *agile, 2008. AGILE '08. Conference*, 333-338.

Tengshe, A., & Noble, S. (2007). *Establishing the agile PMO: Managing variability across projects and portfolios* Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4293594

The Hanover.*Hertiage.* Retrieved Septermber 25, 2010, from http://www.hanover.com/thg/about/heritage.htm

The Hanover.*Overview.* Retrieved September 26, 2010, from http://www.hanover.com/thg/about/index.htm

Tripp, S., & Bichelmeyer, B. (1990). Rapid prototyping: An alternative instructional design stratagy. *ETR&D 38* (1), p 31-44.

Wailgum, T. (2007, From here to agility. *CIO Magazine,* Retrieved from http://www.cio.com.au/article/193933/from_here_agility/?

What is rapid application development? (1997). CASEMaker. Retrieved from http://www.casemaker.com/download/products/totem/rad_wp.pdf;

Appendix A- Interview Questions

General

- What is your role at this company?
- Can you describe your average day at Hanover?
- How many projects do you typically work on at a time?
- Do you ever have any time management issues?

Current SDLC

- Briefly describe your role in the current waterfall methodology.
- Are there any steps or deliverables in the current SDLC that you feel take to much time or are non-value adding?
- How often does the core project team meet? What goes on during these meetings?
- What concerns do you have about the current SDLC?
- What are some differences you see in the methodology for developing smaller projects versus larger projects?
- Have you ever cut corners to meet a deadline?
- Are there any Project Managers that do a better job than others? If so, why?
- How often do the requirements get changed for a project?
- What is you idea on overlapping phases, can you give us instance when it worked well for a project and instance when all the efforts went to waste?
- How are people in team arranged? Are they scattered or they are concentrated in one part of the building?

AGILE

- What does agile mean to you?
- What would agile mean to Hanover?
- Do you have experience working with agile methodologies from a previous employer? If so what types and were they successful?
- What problems do you see in moving towards an agile methodology?
- What type of agile methodology do you feel would work best with Hanover?
- Do you think Hanover would benefit from having a few SDLCs in place so that PMs can decide which one fits their project best?

APR

- Do you think the APR system is an acceptable workaround?
- Have you worked on any APRs? If so what was different and how successful was it?
- Was there anytime when you felt alternative practice was needed, but no APR was submitted as the playbook was followed?
- Do you think the alternative practice you have used in this particular project should be part of the playbook and it would benefit more projects?

Business Sponsor

- When sponsoring a project what kind of documentation do you find most important?
- How often do you meet with the project manager and/or core project team?
- What are your biggest concerns when developing a project?
- Do you feel that project often take to long and should be able to be completed quicker?
- Do you think the gate meetings could be turned into a weekly meeting to update you on the progress of the meeting and ask for your approval to proceed?

Developer/ Tech Lead

- What language(s) do you program in?
- What other languages are used in developing software?
- Do you ever feel rushed in programming due to fast approaching release date?
- How do you think this effects the quality of the project?
- In a more agile approach do you think project quality would increase?

Business Analyst

- How often do requirements change?
- As the customer needs keeps changing over the time, is there any instance when the project passed the business system analysis phase and the customer wanted a change in project?
- How did you deal with that situation and what was its impact on overall project? How often that happens?

Quality Assurance

- Do you feel pressure when developing test cases because this is done close to the release date?

Implementation Manager

- Explain the Change assessment survey.
- What value does this add?
- What is the information from this survey used for?
- How does this process work?
- On average, how many changes get made to the system each release?
- What is the biggest issue you face during releases?
- If a project is scheduled for release but does not make the deadline, how does that affect you?

Security Risk Advisor

- What are the most common risks among projects of different sizes?
- What additional risks do you think an agile approach would add?
- What types of risks are sponsors willing and not willing to take?

Project Manager that strictly follows the playbook

- How do you decide how many hours of each person you need for a project?
- When do you bring the different types of team members into the project?
- Do you follow the playbook word for word, or use it as a guideline for development?

Project Manager that submits frequent APRs

- Do you follow the playbook word for word, or use it as a guideline for development?
- What types of methodologies have you used in APRs? Were they successful?
- What types of projects did you submit APRs for? Why?
- How do you decide how many hours of each person you need for a project?
- When do you bring the different types of team members into the project?

Appendix B – Interview Tables

Table 7: Resource Cause Quotes

Category	Position	Thoughts
Allocation	Lead Architect	Problems arise mean people are allocated to several projects
Allocation	Lead Architect	Problems happen when 3 projects are assigned all at once, people need to rank projects and manage time. This can be difficult and it is hard to track peoples time
Allocation	Delivery Manager	Can't share developers, need to develop a project team and have them not be so spread out, but if they are spread out they need to have Skype like meetings
Collocated	Program Director	Teams are not collocated

Table 8: Procedure Cause Quotes

Category	Position	Thoughts
Length of process	Lead Architect	If the process is followed directly, length of process is too long
Documentation	Lead Architect	Process is mostly about deliverables
Length of process	Lead Architect	Clarity of requirements elongated the time of the process
Sign offs	Lead Architect	Should let people start working on stuff if previous documentation has not been signed off
Documentation	Lead BA	Documentation should not be changed
Documentation	Delivery Manager	Make one technical documents to speed up the process
Doc and sign offs	Senior BA	Too many deliverables and sign offs

Table 9: APR Cause Quotes

Category	Position	Thoughts
frequency	Delivery Manager	Almost every project has an APR
Timely	PM	It takes a lot of paper work to make a decision on APRs
Timely	Enterprise Solutions Architect	APR system needs to be more adaptable, because it takes too much time for an APR to get approved
Timely	Program Director	APR is overdone, as long as Project director, PM DM, and sponsors are aware of changes and risks involved APR shouldn't have to go through the long process of approval

Table 10: Sponsor Cause Quotes

Category	Position	Thoughts
Requirements	Lead Architect	The clarity of requirements up front needs to be better and more thought through
Involvement & Requirements	Program Director	Requirements aren't fixed, business comes in to late
Involvement & Requirements	Program Director	Involve business earlier so they see the requirements
Involvement	Program Director	Example " for one project they spent 15 minutes with the project sponsor, he loved it, when delivered they had better results- this model should be utilized."
Involvement	Senior BA	Involve sponsor more, all people at the same table would be great to answer questions

Table 11: Change Request Cause Quotes

Category	Position	Thoughts
Change requirements	Delivery Manager	Need a way to overcome changes to requirements
Change requirements	PM	Customers change requirements after BRD is complete. This is a problem
Lost Labor	Program Director	Sometimes run the risk of rework.
Change requirements	PM	Requirements change can impact the team, the project time and cost, and the end user
Process	Senior BA	Change control work = biggest headache

Table 12: Suggestions Table

Category	**Position**	**Thoughts**
Resource involvement	Lead Architect	Involve architects/ developers early
Playbook	Lead Architect	Build a backdoor into the process for pahse overlap
Playbook	QA project Lead	Playbook should be more of a guideline/ more flexibile
Track	PM	Componentize projects into tracks which can also be broken into smaller tracks later in the project if needed
Involvement	Lead BA	Involve sponsors more and get everyone at the same table to answer questions

Appendix C – Surveys

Satisfaction with the New Methodology Survey – team

Answer all questions in sections 1 and 2 on a scale of 1-5, with 5 being the highest and 1 being the lowest

1. Answer these questions based upon the SDLC as of 2010

1. How much control you felt you had over the process ____.
2. Rate the complexity of the process ____.
3. How often was the business sponsor involved ____.
4. How often did a change in requirements affect the development time ____.
5. How long did it take to get approval for documents and move forward ____.

2. Answer these questions based upon the Hybrid Methodology involving evaluation points

1. How much control you felt you had over the process ____.
2. Rate the complexity of the process ____.
3. How often was the business sponsor involved ____.
4. How often did a change in requirements affect the development time ____.
5. How long did it take to get approval for documents and move forward ____.

3. Answer the following open ended questions about the new SDLC

1. Do you feel that is process an improvement from the previous SDLC?

2. Did you feel that the naming conventions for documents were useful?

3. Do you have any suggestions for improvement to this SDLC?

Satisfaction with the Hybrid Methodology Survey – business sponsor

Answer all questions in sections 1 and 2 on a scale of 1-5, with 5 being the highest and 1 being the lowest

1. Answer these questions based upon the SDLC as of 2010

6. Rate the amount of time you spent with the team on a typical project ____.
7. How often was development slowed due to a change in requirements ____.
8. Overall Satisfaction with this process____.

2. Answer these questions based upon the Hybrid Methodology involving evaluation points

1. Rate the amount of time you spent with the team on a typical project ____.
2. How often was development slowed due to a change in requirements ____.
3. Overall Satisfaction with this process____.

Is there anything you would change about this process?