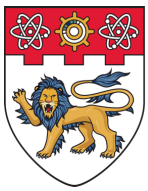

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

PROJECT COST MANAGEMENT
A study of Waterfall Model vs Agile Model

In submission for

CI6204
SOFTWARE PROJECT MANAGEMENT

AY 17/18
Sem 1

Submitted by:
CHENG Shuyun (G1701268F)
NAH Zheng Xiang, Philson (G1701513D)
XUE Fei (G1701182G)

Abstract

In today's competitive business environment, organizations require IT systems that constantly evolve to meet their ever-changing requirements. This has led many organizations to favor Agile Models over traditional Waterfall Models for software development. However, there appears to be a lack of understanding of how conventional PMBOK processes apply to these Agile Models. In this paper, we have analyzed the differences in Project Cost Management processes between Waterfall and Agile models, and found them to have fundamentally different priorities. The Waterfall Model aims to complete all the specified functionalities, while the Agile Model works on a ROI maximization approach. We then used the FBI Sentinel case study to examine the practical implications of the different approaches. In practice, the Waterfall Model's sequential approach to development lead to greater project rigidity and inflexible timelines, while the Agile Model allowed for re-prioritization of requirements to deliver the features that matter most to users. Further analysis of the case study allowed us to tease out 3 major benefits of the Agile Model for Project Cost Management, namely – allowance for change, focus on business value, and predictable costs.

1. Introduction

Software development methodologies are constantly evolving due to changing technologies and new demands from users. Today's dynamic business environment has given rise to emergent organizations that continuously adapt their structures, strategies, and policies to suit new environments (Truex, Baskerville, & Klein, 1999). Such organizations require IT systems that constantly evolve to meet their ever-changing business requirements. Traditionally, most IT projects have been developed and managed using the Waterfall Model (Boehm, 1988). However, such a plan-driven software development methodology lacks the flexibility to adapt and adjust to changes in today's competitive business environment (Osorio, Chaudron, & Heijstek, 2011). Since the early 2000s, the Agile Model of development has been steadily gaining popularity among companies (Augustine, Payne, Sencindiver, & Woodcock, 2005). In fact, a recent survey conducted by Intland revealed that 46% of companies have begun adopting the Agile Model for software development (Intland, 2017).

The project management processes as defined in Project Management Body of Knowledge (PMBOK), are structured around five process groups (initiating, planning, execution, controlling and closure) and nine knowledge areas (integration management, scope management, time management, cost management, quality management, human resource management, communication management, risk management, procurement management) (Project Management Insitute, 2000). These processes have been widely regarded as the de facto IT project management approach (Holtzman, 1999). However, while the PMBOK approach is well suited and tested for the structured Waterfall Model (Crespo-Santiago & Cosme, 2011), it is not yet well understood how these processes would apply to modern Agile Models that focus on flexibility and adaptability (Fitsilis, 2008).

Project Cost Management is one of the most important factors in The Iron Triangle that influences a project's success or failure (Morris & Hough, 1988). Studies show that approximately 45% of all large-scale Waterfall Model based projects encounter cost overrun issues even before project completion (Bloch, Blumberg, & Laartz, 2012). Yet, majority of projects adopting the Agile Model appear to able to achieve high customer satisfaction and

deliver projects on budget (Lee & Xia, 2010). Therefore, the objectives of our research would be to analyze the differences in Project Cost Management processes used in the Waterfall Model and the Agile model, examine the practical implications of the two different approaches, and to expound the benefits of the Agile Model on Project Cost Management. In the first portion of this paper, we shall be reviewing existing literature put forth in the subject domain, so as to analyze the differences in cost management processes between the Waterfall and Agile Models. Next, through our analysis of the FBI Sentinel Project case study, we will shall examine the practical implications of the two different approaches and expound on the benefits of the Agile Model that led the team to a successful project delivery.

2. Literature Review

2.1 Project Cost Management

Project Cost Management is the process of ensuring that a project is developed within a given budget limit and successfully delivered on time (Schwalbe, 2016). Project Cost Management is one of the most important factors in The Iron Triangle that influences a project's success or failure (Morris & Hough, 1988). Studies show that approximately 45% of all large-scale IT projects encounter cost overrun issues even before project completion (Bloch, Blumberg, & Laartz, 2012). Moreover, inaccurate cost estimation and inadequate use of cost estimation tools rank as among the most common reasons for software development project failures (Rajkumar & Alagarsamy, 2013). It is therefore critical for project managers to be cognizant of the four core processes that constitute Project Cost Management. They are: Panning Cost Management, Estimating Costs, Determining Budget, and Controlling Costs (Schwalbe, 2016).

Planning Cost Management

The goal of Planning Cost Management is to create a cost management plan. This entails the defining of policies, procedures, and documentation flows that are to be utilized for planning, executing, and controlling project cost (Schwalbe, 2016). The work breakdown structure of the project, scope statement, resource pool description and organizational policies form useful inputs in the planning of project cost. (Project Management Insitute, 2000). Project managers may use a combination of analytical techniques and expert judgment to develop the cost management plan (Schwalbe, 2016).

Estimating Costs

Developing the cost estimate is one of the most challenging and essential tasks in software development (Keung, Jeffery, & Kitchenham, 2004). Project managers need to use various inputs such as the project requirements, resource utilizations and other constraints to formulate accurate cost estimations (Schwalbe, 2016). By forming accurate cost estimates, project managers are then able to evaluate project processes, create reliable schedules, and monitor project success. (Lederer & Prasad, 1995). Tools and techniques such as analogous estimates (Schwalbe, 2016) and parametric estimation (Keaveney & Conboy, 2006) help aid the project manager in developing more accurate cost estimations.

Determining Budget

To determine the budget, is to allocate the total project budget to individual activities or items, and establish a cost baseline for measuring project performance (Project Management Insitute, 2000). This process involves going through the individual items listed in the Work Breakdown Structure (WBS) document, to assign costs (Schwalbe, 2016). The goal of determining the

budget is to create a cost baseline for the purpose of measuring project performance and evaluating if the project is on course (Schwalbe, 2016).

Controlling Costs

Controlling costs involves adjusting the cost baseline, informing stakeholders of authorized changes, determining the baseline that has changed, monitoring project cost performance and managing changes when they occur (Project Management Insitute, 2000). Project managers can utilize tools such as Earned Value Management (EVM) to compare actual values of project costs with cost baselines to determine project performance, and whether actions are needed to remedy the situation (Lipke, Zwikael, Henderson, & Anbari, 2009). The output of cost controlling processes include: updates to the budget, revised cost estimates, estimates of the total cost of completion and lessons learned reports. (Project Management Insitute, 2000)

2.2 Waterfall Model

The Waterfall Model is a linear sequential design approach in which progress flows in one direction downwards like a waterfall. It is believed that the model was first documented by Winston Royce in 1970. The author did so with the intention of documenting the steps taken for the successful projects that he worked in. By "successful", it was with respect to "arriving at an operational state, on-time, and within costs" (Royce, 1987). This means that for the Waterfall Model, there is a fixed budget, which implies that the project scope and the time taken for the project should also be fixed.

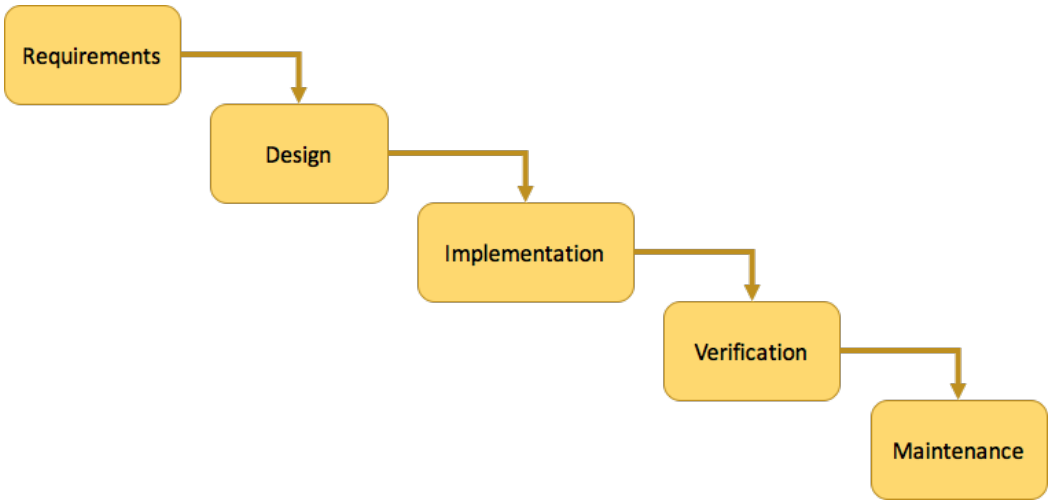

Figure 1: Phases In Waterfall Model

There are 5 phases in the Waterfall Model - Requirements, Design, Implementation, Verification, Maintenance. As seen in Figure 1, the phases of the Waterfall Model flow from one phase to the next, with no options of iteration. This implies that each phase must be fully completed before the project can move to the next phase.

At the start of the project, the scope has to be clearly defined during the Requirements phase so that requirements can be fully documented before the Design phase can begin. Like what the authors wrote in the paper Agile Project Management - Agilism versus Traditional Approaches, projects that uses the Waterfall Model are "clearly defined with well documented and understood features, functions, and requirements" (Fernandez & Fernandez, 2016, p. 15). The strict requirement of completion of each phase before moving to the next also implies that the Waterfall Model tend to emphasize on predictability, stability and compliance (Fitsilis,

2008), as you are able to predict what is going to happen next, things are stable and are of certain compliance.

When it comes to Cost Management, in order to make good estimations on cost, all requirements of the project have to be made known up-front at the beginning of the project. The Waterfall Model has it that each phase must be completed before the project can move to the next. When the stakeholders find something wrong in the Verification phase, e.g. a portion of the developed project doesn't seem to be what they were looking for, sometimes it could mean that the project has to "restart" from the Requirements phase again. When such events happen, the cost of changes could rise exponentially with time (Highsmith & Cockburn, 2001).

2.3 Agile Model

Critics of the traditional Waterfall Model often cite the fact that it is too heavily regulated, planned, and micro-managed, leading to excessively long project lead times (Osorio, Chaudron, & Heijstek, 2011). Such long lead times were deemed unacceptable by businesses, as projects often delivered systems that did not meet the business's current needs, even if the project's original objectives were met (Begel & Nagappan, 2007). During the 1990s, several lightweight software development methods emerged as alternatives to the traditional Waterfall Model (Larman & Basili, 2003). Such methods included: rapid application development (RAD), dynamic systems development method (DSDM), SCRUM, and extreme programming (XP). Given their similar characteristics, these methods were collectively referred to as agile development (Larman C. , 2004). Modern implementations of the Agile Model, often display the following three characteristics: Iterative, Adaptive, and Incremental (Miller, 2001). These characteristics are manifested in the software development lifecycle as illustrated in Figure 2.

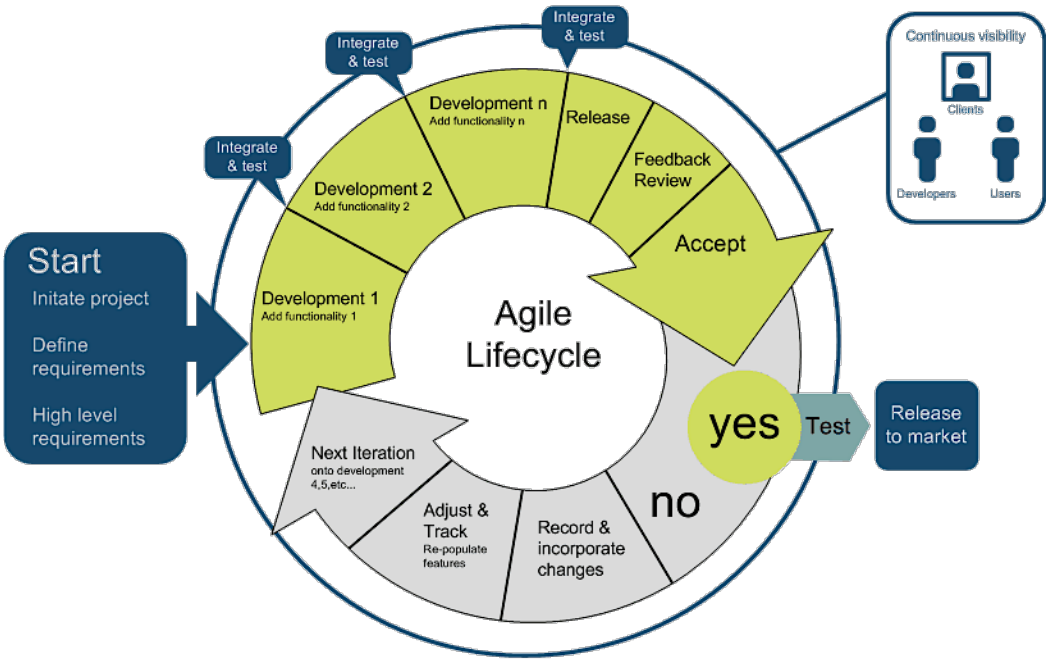

Figure 2: Software Development Lifecycle of Agile Model

Adapted from: *Agile Software Development Life Cycle* (Paul, 2016)

Software development projects following the Agile Model are executed in a series of short iterations called “Sprints”. High level requirements are gathered and defined before the start of each Sprint (Fernandez & Fernandez, 2016). Functionalities are then incrementally developed and added to the system, with each functionality tested for quality before being integrated into the main system (Jeffries, Anderson, & Hendrickson, 2001). Project stakeholders maintain visibility throughout the entire process by means of face-to-face communication (Ambler, 2002). At the end of each Sprint, the resulting system is presented to the clients for feedback review. If the product is accepted, it will be primed to be released to market. Otherwise, the remaining adjustments, along with any new requirements, will be recorded and reprioritized to be incorporated into the next Sprint (Paetsch, Eberlein, & Maurer, 2003).

The use of short iterations allows the Agile Model to be more adaptive as new requirements and modifications can be added and reprioritized at the end of each iteration, allowing businesses to quickly respond to changes (Fitsilis, 2008). Additionally, each iteration delivers a fully serviceable working system (Rising & Janoff, 2000). This helps to mitigate the cost of adding new requirements and changes, as the new functionalities are simply integrated into the existing system, without requiring a complete overhaul of the system design. The cost estimates are hence adjusted accordingly before the start of each iteration (Keaveney & Conboy, 2006). This iterative approach means that clients can maximize their Return on Investment (ROI) of the project as they get to determine which requirements are to be prioritized to be delivered within the remaining budget.

2.4 Waterfall Model vs Agile Model

Given the differences in software development lifecycles of the Waterfall Model and the Agile Model, the approach to Project Cost Management of the two models would understandable be different. In this section, we shall be using the Project Cost Management processes outlined in section 2.1 to analyze the differences in Project Cost Management between the Waterfall Model and Agile Model of software development. A side-by-side comparison of their differences is summarized in Table 1.

	Waterfall Model	Agile Model
Planning Cost Management	Cost Management Plan is done at start of project, to provide guidance and direction on how the project costs will be managed throughout the entire project (Project Management Insitute, 2000).	Uses incremental, multi-level planning (Sulaiman, Barton, & Blackburn, 2006). Cost Management Plan is continually refined at the beginning of each iteration.
Estimating Costs	All the requirements must be defined upfront, following which the cost estimation is done (Strike, El Emam, & Madhavji, 2001).	Cost estimation is iterative. Scope and costing will be determined before the start of each cycle (Keaveney & Conboy, 2006).
Determining Budget	Uses WBS and other inputs to allocate the budget and create Cost Baseline for every activity in the WBS (Schwalbe, 2016).	Makes use of User Stories and Story Points to determine the budget for the current sprint (Sulaiman, Barton, & Blackburn, 2006).
Controlling Costs	Uses a change control system to prevent too many changes from occurring and deviating too far from the Cost Baseline (Schwalbe, 2016).	Uses "biggest bang for the buck" approach (Stepanek, 2005). Requirements are prioritized by the users to determine what to deliver within the budget.

Table 1: Differences in Project Cost Management between Waterfall and Agile Models

Planning Cost Management

The purpose of Planning Cost Management, is to create a Cost Management Plan to provide guidance and direction on how project costs will be managed throughout the project. This involves determining the procedures, allocating resources, planning the documentation flow for managing project cost (Schwalbe, 2016). In the Waterfall Model, all project requirements are formalized upfront (Fernandez & Fernandez, 2016). This facilitates the planning of cost management procedures for the entire project. Project managers can establish the necessary policies, documentation procedures, and change control systems to manage cost across the duration of the project (Project Management Insitute, 2000). Unlike the Waterfall Model that focuses on conformance (Fitsilis, 2008), the Agile Model focuses more on being responsive and adapting to change (Dybå & Dingsøyr, 2008). Hence, it uses an incremental, multi-level planning approach (Sulaiman, Barton, & Blackburn, 2006). Broad high-level plans and policies may be outlined at the start of the project. However, as the Agile Model primarily works towards the release of each sprint (Hoda, Noble, & Marshall, 2008), the cost management plan is expected to be continually refined and adapted based on the current needs of the project, before the beginning of each sprint.

Estimating Costs

Estimating costs is often challenging, yet it is an essential step for gauging if a project would be financially viable (Keung, Jeffery, & Kitchenham, 2004). By estimating costs, project managers are able to evaluate project processes, create reliable schedules, and monitor project success. (Lederer & Prasad, 1995). In the Waterfall Model of development, all requirements must be defined upfront at the beginning of the project, following which the cost estimation is done (Strike, El Emam, & Madhavji, 2001). This assumes that there is perfect knowledge of

all the requirements of the project, with which accurate cost estimates can be prepared. Conversely, in the Agile Model, cost estimation is conducted iteratively. The development scope and costs are determined before the start of each sprint (Keaveney & Conboy, 2006). In the Agile Model, requirements are gathered in the form of User Stories, which are short descriptions of system functionalities that satisfy the users' needs (Cohn, 2004). Following which, Story Points are then assigned to each User Story (Coelho & Basu, 2012). Story Points are a relative measure of the effort required to develop a User Story (Bundschuh & Dekkers, 2008). For example, a User Story with a Story Point of 2 is assumed to take twice the effort of a User Story with a Story Point of 1.

Determining Budget

By determining the budget, cost is allocated to to individual activities and items in order to establish a cost baseline for measuring project performance (Project Management Insitute, 2000). In the Waterfall Model of development, all requirements have been clearly defined upfront. As such, determining the budget involves going through the Work Breakdown Structure (WBS) document, to assign a cost to each item (Schwalbe, 2016). The end result of this activity, would produce a Cost Baseline, which can be used to track cost deviations and hence the overall performance of the project (Vanhoucke, 2009). In the Agile Model, budget is determined at the sprint level. Prsoject managers would make use of the aforementioned User Stories and Story Points to determine budget (Sulaiman, Barton, & Blackburn, 2006). Before the start of each sprint, the team will prioritize the User Stories to be developed in the current iteration (Schwaber & Beedle, 2002). The Story Points of the User Stories for the current sprint would then facilitate in the determination of the budget of that sprint.

Controlling Costs

Controlling costs is the process of tracking, monitoring and managing costs in order to ensure that a project does not encounter a budget overrun (Project Management Insitute, 2000). In the Waterfall Model, requirements are clearly defined, and Cost Baselines have been established. As such, project managers can utilize tools such as Earned Value Management (EVM) to track and monitor costs (Lipke, Zwikael, Henderson, & Anbari, 2009). Nevertheless, changes in requirements are to be expected in any project. Hence, project managers adopting the Waterfall Model would be required to put in place a change control system to prevent too many changes from occurring and deviating too far from the Cost Baseline (Schwalbe, 2016). On the other hand, the Agile Model was designed to be adaptive and responsive to changes. As such, changes are acceptable even in the later stages in the project (Cao, Mohan, Xu, & Ramesh, 2009). Project managers monitor and control overall project performance by means of using Burndown Charts, which are visual representations of the total backlog of Story Points and progress made against that backlog over time (Karlesky, Object, & Vander Voord, 2008). The Agile Model adopts a "biggest bang for the buck" approach (Stepanek, 2005), which focuses on delivering the requirements valued most by the users within the given budget, thereby maximizing the ROI of the project.

3. Case Description

In this paper, we shall be examining the case of the FBI Sentinel Project. The Sentinel project emerged as a result of a history of failures from prior implementations in the FBI Trilogy Project. With the project consisting of four main phases, Phase 1 and 2 of the Sentinel project were initially implemented using the Waterfall Model. Given the lackluster performance of the project, and the poor quality of results delivered in Phases 1 and 2, the project team then

decided to transition to the Agile Model of development. Eventually, the team was able to produce an effective system that met the expectations of stakeholders, and delivered the project within the given budget. By analyzing the differences in approaches taken by the team across the span of the Sentinel project, we will be able to better understand the practical implications of the Waterfall Model and Agile Model on Project Cost Management, and glean insights on the qualities of the Agile Model that aided the team in the successful delivery of their project.

3.1 Background

In 2000, FBI's IT system and hardware was not upgraded more than ten years. More than 13,000 FBI computers could not run the modern software and this caused the network components to malfunction. For instance, 12 computers were employed to upload one document into the case management system, etc (USDOJ/OIG Audit Division, 2005).

After the 2011 September 11 attacks, FBI realized the IT case management system and framework were outdated and needed to be improved. From September 2000, FBI started a technology upgrade project called "FBI Information Technology Upgrade Project", later renamed as "Trilogy" (USDOJ/OIG Audit Division, 2005). The objective of Trilogy was to improve the IT infrastructure, redesign the case management system, upgrade IT applications for FBI officers and department. The requirements of Trilogy Project were the Information Presentation Component (IPC), Transportation Network Component (TNC), and User Application Component (UAC) (USDOJ/OIG Audit Division, 2005). The key component of UAC is called the Virtual Case File (VCF) which was to become the new case management system for FBI - replace all the paper case files, and create a new operation support environment for FBI agents.

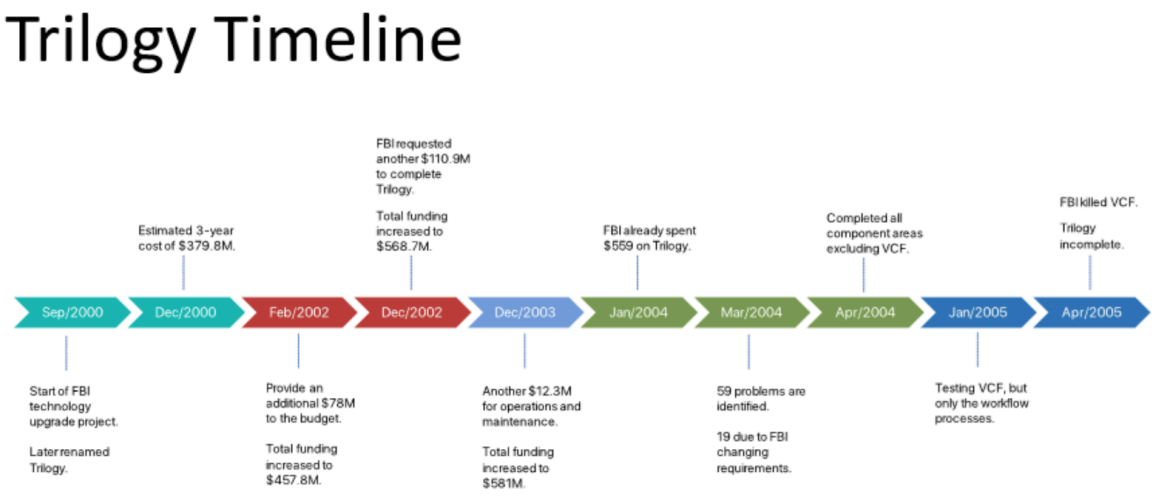

Figure 3. Trilogy Timeline

Adapted from (USDOJ/OIG Audit Division, 2005) (Schmitz, Tada, & Hess, 2014)

At the end of 2000, Trilogy started with a budget of $379.8 million over a period of three years using the Waterfall Model (Marchewka, 2010). Between 2002 to 2003, FBI spent $201.3 million on Trilogy. At the same time, the total funding increased to $581.1 million. By the end of January 2004, FBI spend $559.6 million on Trilogy which was $179.8 million more than the original amount (USDOJ/OIG Audit Division, 2005). In April 2004, the infrastructure components were completed by FBI. However, 59 issues and sub-issues were identified and

19 of them were due to requirements changes (Marchewka, 2010). In April 2005, FBI announced the failure of Trilogy as the key system VCF was unaccomplished (Schmitz, Tada, & Hess, 2014). The Trilogy timeline is summarized in Figure 3.

Sentinel

In 2005, after the Trilogy project and VCF have failed, FBI decided to start another IT project called Sentinel (USDOJ/OIG Audit Division, 2006). Sentinel was similar to the VCF, it provided a new computer-based case management system, converted paper-based documents into soft copies, and allow FBI agents to easily maintain the investigation data (Figure 4). Moreover, due to the functionalities of Sentinel, the project is required to change software components and to maintain it efficiently. Therefore, FBI decided to use a more flexible software development model for Sentinel (USDOJ/OIG Audit Division, 2006).

While Sentinel continue to adopt the Waterfall Model which is the same as Trilogy, FBI decided to separate the entire Sentinel project into four overlapping phases whereby each phase will take about 12-18 months. Therefore, Sentinel was scheduled to complete in 2009 with a total budget $425 million (USDOJ/OIG, 2006).

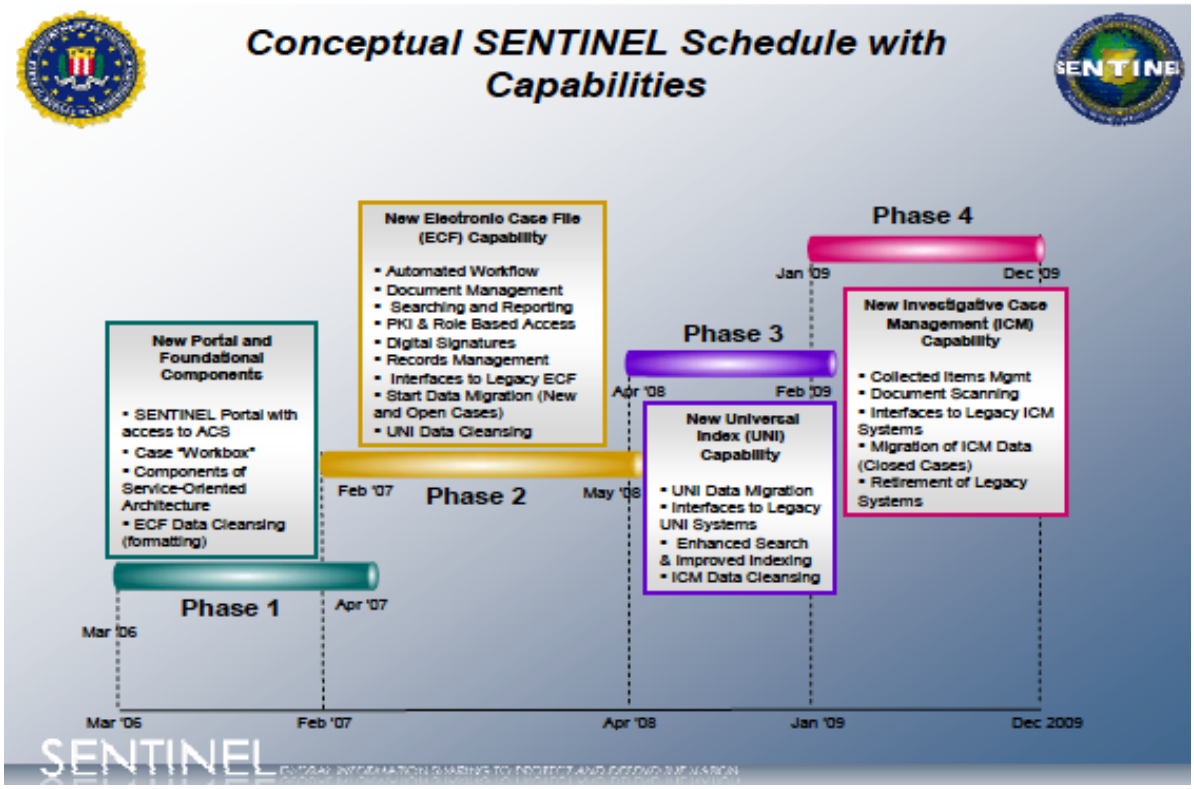

Figure 4: Initial Project Plan for FBI Sentinel

Adapted from (USDOJ/OIG Audit Division, 2007)

3.2 Sentinel Phase 1 and 2 (Waterfall Model)

Sentinel Phase 1 started in March 2005 with an estimated budget $57.2 million and it was planned to complete within 12 months (USDOJ/OIG Audit Division, 2006). The main scope of Phase 1 was to develop a web based portal of existing case management system, create a summary of cases which involved, and to complete the fundamental components of Sentinel project architecture (USDOJ/OIG Audit Division, 2006). Nonetheless, Phase 1 with a two

months delay was finally delivered on June 2007. Furthermore, the budget of Phase 1 raised from $57.2 million to $59.7 million due to the impractical schedule and increasing workload (USDOJ/OIG Audit Division, 2007). When Phase 1 was delivered, the team were not able to get the users' buy-in to use the system due to its limited functions (USDOJ/OIG Audit Division, 2008).

After Phase 1 was completed, FBI decided to split the rest of phases into smaller segments with the aim for greater project efficiency (USDOJ/OIG Audit Division, 2008). Sentinel Phase 2 was divided into four segments with an estimated budget $137 million. Phase 2 started in October 2007, was planned to finish in 16 months (USDOJ/OIG Audit Division, 2008). The main scope of Phase 2 was to transfer all the paper case documents to electronic records with a supported workflow tool and build a new security framework for the system (USDOJ/OIG Audit Division, 2007). Throughout this phase, the team had a system that monitored and controlled project costs for the project. They also had a methodology that allowed the program managers to measure the project against baselines and identify possible issue (USDOJ/OIG Audit Division, 2006).

In January 2008, because of the strategic planning, the estimated budget of Sentinel increased from $425 million to $451 million (USDOJ/OIG Audit Division, 2009). Even though Sentinel Phase 2 delivered all of the segments on 1st July 2009, it failed to fulfil the requirements for the key functions during the testing stage. In addition, FBI reconsidered the remaining parts of Phase 2 (USDOJ/OIG Audit Division, 2009). As a consequence, the cost of Phase 2 increased to $155 million and delivered in December 2009 (USDOJ/OIG, 2010).

On 3rd March 2010, FBI announced to terminate all the ongoing tasks of Phase 3 and 4 due to the catastrophic quality issues of Phase 2 Segment 4 during testing stage. Phase 2 Segment 4 had many loopholes in terms of usability, performance, and productivity. At the same time, 26 critical function-related issues were discovered (USDOJ/OIG, 2010). Due to the change in schedule and cost, FBI had difficulty in estimating the specific cost and time-line for the rest of the project. In August 2010, FBI had already spent $405 million on Sentinel which was originally estimated at a budget of $451 million (USDOJ/OIG, 2010). FBI defended that the project might cost more than the estimated $451 million, and could not be completed on time (USDOJ/OIG, 2010). Phase 2 was conditionally accepted by FBI in Apr 2010, which was 2 years behind the original schedule (USDOJ/OIG, 2010).

3.3 Sentinel Phase 3 and 4 (Agile Model)

In October 2010, both Phase 1 and Phase 2 were delivered, but nearly half of the functions didn't meet the requirements (USDOJ/OIG, 2010). Since Sentinel started in 2006, FBI had already spent six years on this project. Sentinel absolutely brought significant leading technology and modern work processes to the FBI after completed Phase 1 and 2. However, at that time, Sentinel was $100 million over budget with only $45 million remaining budget left, and two years delayed according to the original schedule (USDOJ/OIG, 2010). As a result, FBI officially decided to use a new approach to accomplish efficiencies with lower cost to finish the rest of Sentinel, which is the Agile Model. FBI established $32.6 million estimated budget to complete the rest of Sentinel, also reduced staff number from 250 to 52 (USDOJ/OIG, 2011).

Agile Model is not only a set of the tool but also an approach for Sentinel project that allows collaboration between users and requirements and focuses on the high priority functions. FBI divided all the delivery work in Phase 3 and 4 into increments called sprints. (Figure 5) Each sprint lasted for two weeks, it contained User Stories and Architecture Stories which referred

to users performed functions and software configuration architecture (USDOJ/OIG, 2011). This approach helped measure the progress and schedule of the rest of Sentinel by using a burndown chart as seen in Figure 5 (USDOJ/OIG, 2011). Initially, the primary scope of Phase 3 and 4 was to create the new universal index, increase the number of attributes, and combine different components into one case management system. But because the FBI decided to use the Agile Model, plus the fact that there were some unfinished parts for Phase 1 and 2, FBI changed the scope of Phase 3 and 4. They choose to replace and improve the functionalities delivered to Sentinel Phase 1 and 2. The new scope mainly included the development of System Of Record (SOR), create a Full Operating Capability (FOC), and Sentinel Advisory Group testing (SAG) (USDOJ/OIG, 2011). FBI also conducted surveys to get users' feedback on how they can improve the system (USDOJ/OIG, 2011).

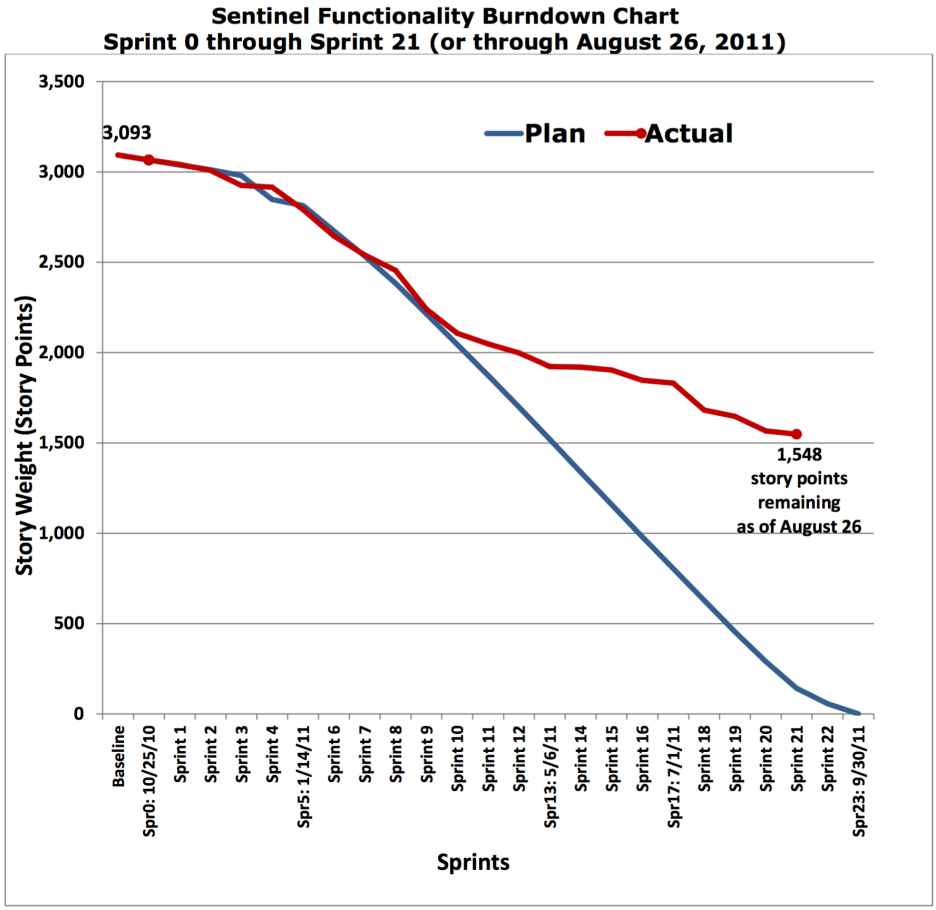

Figure 5: Sentinel Functionality Burndown Chart

Adapted from (USDOJ/OIG, 2011)

In July 2012, FBI announced that Sentinel 1.0 was launched, and available to all the users (USDOJ/OIG Audit Division, 2014). For the cost, FBI spent a total of $441 million on Sentinel system development out of $451 million budget, but this is not including operation, maintenance and employee cost. For two years of operations and maintenance cost, the total estimated cost was $60 million, and another approximately $30 million for employee costs (USDOJ/OIG, 2012). With Sentinel 1.0, it completed about 97% functionalities out of 1047 Sentinel's System Requirements Specification (USDOJ/OIG, 2012) (Table 2). After delivered Sentinel 1.0, FBI still worked on the unfinished functionalities and improved on it. Until September 2014, FBI released final version Sentinel 1.5, and the total budget increased to $551.4 million (USDOJ/OIG Audit Division, 2014).

Sentinel Functional Area	Functional Task or Ability	Stage Involved
Case Management	Oversight of Investigative and administrative activities associated with a case	Phase 1, Phase 2, Agile
Collected Items Management	Activities associated with documenting the collection, storage, and tracking of physical items related to FBI cases.	Phase 2, Agile
Indexing	Collection and maintenance of investigative and administrative information about persons, organizations, locations, incidents, property, and communication accounts.	Phase 2, Agile
Records Management	Functions required to manage the records entered in to official FBI case files.	Agile
Search	Ability to locate different types of information connected within Sentinel.	Phase 2, Agile
Work Item Authoring	Memorialization of the work that has been accomplished or is in the process of being accomplished and association of that work with specific cases.	Phase 2, Agile
Work Flow	Integrated tools that allow FBI personnel to create, read, update, and delete documents and other work items. Also allows users to author or co-author work items.	Phase 1, Phase 2, Agile

Table 2: FBI Sentinel - Summary Of Functional Areas

Adapted from (USDOJ/OIG Audit Division, 2014)

4. Discussion

4.1 Differences in Cost Management

We have seen how Sentinel progressed from the initial days to the launch of Sentinel 1.5. Using our understanding of the differences in Waterfall Model and Agile Model as highlighted in section 2.4 Waterfall Model vs Agile Model, we will now compare how Sentinel Phase 1 and 2 differs from Sentinel Phase 3 and 4, given that the former used the Waterfall Model, and the latter used the Agile Model.

Planning Cost Management

At the beginning of Sentinel, decisions were made to split Sentinel into phases, with each phase spanning a duration of 12-18 months. This remained the case for Sentinel Phase 1 and 2 as both were still using the Waterfall Model. However, when FBI decided to use Agile Model for Sentinel Phase 3 and 4, as part of the planning process, they decided to "enhance" what had been done in Sentinel Phase 1 and 2, on top of what was originally planned for Sentinel Phase 3 and 4. Even though the requirement list was determined right at the beginning of Sentinel Phase 3 and 4, the requirements were prioritized and were split into sprints. What was required

to be done for each sprint was decided only at the start of the sprint. This made it easier to do cost management planning as the sprints were only of a 2 weeks duration.

Estimating Costs
Cost estimation is done when the requirements are confirmed. In the case of Sentinel Phase 1 and 2, the deliverables for each phase were defined at the start of the project. For Sentinel Phase 3 and 4, the deliverables were decided at the start of each sprint.

Determining Budget
For Sentinel Phase 1 and 2, it was mentioned that the team tracked the project progress using a methodology that was undisclosed. In Sentinel Phase 3 and 4, with the usage of the burndown chart, it was implied that the chart was used to determine the deliverables for the immediate next sprint. As the requirements were re-prioritized upon the completion of each sprint, it made sure that the more important requirements get completed first.

Controlling Costs
EVM was used to measure the project progress for Sentinel Phase 1 and 2. Change requests that came in were also logged and there was no mention that they were acted upon immediately. For Sentinel Phase 3 and 4, the team ensured that important requirements get completed first. From the steepness of the red line in Figure 5, we can deduce that requirements with higher Story Weights get completed in the earlier sprints. Such requirements are usually the more important ones due to the re-prioritization of requirements before the start of each sprint.

A summary of the differences is tabulated in Table 3.

	Sentinel (Phase 1, 2)	Sentinel (Phase 3, 4)
Planning Cost Management	Planning was done at the start of the project – by splitting the project into phases and each of their deliverables	Overall requirements planned at the start Scope was decided at the start of each iteration
Estimating Costs	Deliverables for each phase were defined at the start of the project Cost estimation for each phase was done by contractor	Scope was decided at the start of each iteration
Determining Budget	Used a methodology that allowed the program managers to measure the project against baselines	Requirements were re-prioritized before each iteration started
Controlling Costs	Change requests were logged Used EVM to track project progress	Used burndown charts to determine requirements prioritization

Table 3: Sentinel - Differences in Cost Management

We can see that both models have their own ways of "adhering" to the Cost Management processes. While the Waterfall Model concentrates on completing all the deliverables at one shot, the Agile Model aims at maximizing the ROI of the project by concentrating on the deliverables at each sprint.

4.2 Practical Implications

Waterfall Model
From the case study of Sentinel Phase 1 and 2, we saw that the Waterfall Model was inflexible. The planned phases were overlapping and the team had to start the next phase despite not completing the current phase (Phase 3 and 4 started when they were still in the midst of doing Phase 2). Because of this, when they faced issues (e.g. project delay) during Phase 2, they had to redeploy resources from Phase 3 and 4 to work on Phase 2.

In the case of Sentinel Phase 1 and 2, each planned phase had at least 4 functions to be delivered together. Even though the functions were delivered at the end of the phases, the team faced issues. They could not get the users' buy-in due to the limited available functions. The delivered functions also didn't meet the initial requirements.

Project delays can be an issue for all kinds of projects. Sentinel Phase 1 was delivered 4 months behind schedule. Because there was no concept of incremental deployment, deployment for the phase was late due to delays during development.

Agile Model
In the case of using Agile Model, burndown charts are used to re-prioritize requirements before the start of each sprint – which was exactly what happened in Sentinel Phase 3 and 4. When you have short sprints as compared to long timeframe like the Waterfall Model, requirements can be easily re-prioritized depending on the priority. When budget is a constraint, you can give priority on features that are the most important and can fit into the budget for the sprint.

When the Agile Model is concerned, at the completion of each sprint, the planned requirements are deployed to production for usage. From the burndown chart in Figure 5, requirements were delivered each sprint. Even though these deliverables are not huge functions on their own (since the FBI announced the launch of Sentinel 1.0 only in July 2012), the incremental update of the deployed version made it easier for the users to adapt and provide feedback in a timely manner.

It is unclear if the scope for Sentinel Phase 1 and 2 and Sentinel Phase 3 and 4 are the same, from the shifting of requirements Sentinel Phase 1 and 2 to the originally-planned Sentinel Phase 3 and 4 (before they adopted using the Agile Model), we can see a rough gauge that the scopes are roughly similar as they decided to replace and improve the features that were already delivered in Sentinel Phase 1 and 2, in Sentinel Phase 3 and 4. While Sentinel Phase 1 and 2 took 5 years to complete, Sentinel Phase 3 and 4 was delivered within 2 years, half of the time of the former.

4.3 Benefits of Agile Model

Allowance for Change
One of the features of the Agile Model is the usage of sprints. Because each sprint is short, there is just this much of requirements that can be done within such a short timeframe. This made it easier to control cost.

Another feature is the usage of burndown charts. Burndown charts allow the team to re-prioritize the requirements before the start of each sprint. We saw from the Sentinel Phase 3 and 4 that because of sprints, they were able to incorporate new (and re-prioritized) requirements in a swift manner.

Focus on Business Value
From the burndown chart (Figure 5), it can be implied that the team completed requirements that were more important. Just like the Agile Model, requirements that are of higher importance are given higher weightage. This allowed the team to give priority to requirements were more important.

Because the more important requirements were given priority to be completed earlier, this left the less important ones to the last. Should there be not enough budget to continue another sprint, these requirements will be left uncompleted until more budget is available. This helps to maximize the ROI of the project. We saw from Sentinel Phase 3 and 4 that at the launch of Sentinel 1.0, only 97% of the requirements were completed. They then requested for more budget to complete the rest.

Predictable Costs
When the team re-prioritized the requirements at the start of each sprint, they were able to easily estimate the cost as the sprints were only 2 weeks long. A short sprint allows the cost to be more predictable as there is greater clarity on the possible-requirements that can be completed within the 2 weeks.

4.4 Recommendations

There are no hard and fast rules on when it is good to use the Waterfall Model, or when it is good to use the Agile Model.

Given that the Waterfall Model is based on fixed time, cost and scope, it may be a good model to use when it is a long-term project and is not urgent to be deployed. Or when the project is small and can be managed easily.

But in the modern society, where things are required to be done fast and furious, it may be a good idea to adopt the Agile Model as it is adaptable to changes.

When it comes to startups, budget is usually a constraint. Using the Agile Model might be more ideal as compared to the Waterfall Model. The Agile Model allows prioritization of requirements so the more important requirements get deployed first. When the product after a certain sprint gets rejected, the amount of cost that has been dumped in is also minimized.

5. Conclusion

Today's competitive environment require companies to continually innovate and adapt their processes. Such business landscapes force organizations to rethink and reimagine age old practices that have been deeply rooted in project management culture. Traditionally, most IT projects have been developed and managed using the Waterfall Model. However, many organizations are starting to discover that such structured methodologies are inadequate to help them cope with today's ever evolving business environment. This has pushed many companies to look to the Agile Model for a more contemporary approach to software development that provides the flexibility and adaptability which they seek. The recent rise in popularity of the Agile Model has created a knowledge gap as there appears to be a lack of consensus in understanding how conventional PMBOK process apply to the Agile Model of software development.

In this paper, we have analyzed the differences in Project Cost Management processes between Waterfall and Agile models, and found them to have fundamentally different priorities. The Waterfall Model focuses on conformance, and aims to complete all the documented functionalities and requirements, while the Agile Model works on a ROI maximization approach, by prioritizing features to develop within a given budget. We then used the FBI Sentinel Project case study to examine the practical implications of the different approaches. In practice, the Waterfall Model's sequential approach to development lead to greater project rigidity and inflexible timelines, while the Agile Model allowed for re-prioritization of requirements to deliver the features that matter most to users. Upon further analysis of the case study, three major benefits of the Agile Model for Project Cost Management were expounded, namely – allowance for change, focus on business value, and predictable costs.

While there are no definitive rules on when to adopt the Agile Model as compared to the Waterfall Model, we believe that we have put forth a comprehensive comparison that project managers can leverage on to make a more informed decision on which approach is suitable for them, given the nature of their project. Given the restricted scope of this research, we acknowledge the limitations of our comparison of the Waterfall and Agile Models. Future research may wish to expand on our comparisons to examine how the two models differ in the other aspects of project management, such as – project scope management, project time management or project quality management. A more holistic comparison of the differences between the two models would likely yield useful insights that future scholars and project managers can utilize to further their endeavors.

References

1. Ambler, S. (2002). *Agile modeling: effective practices for extreme programming and the unified process.* John Wiley & Sons.

2. Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005). Agile project management: steering from the edges. *Communications of the ACM, 48*(12), 85-89.

3. Begel, A., & Nagappan, N. (2007). Usage and perceptions of agile software development in an industrial context: An exploratory study. *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on Empirical Software Engineering and Measurement*, 255-264.

4. Bloch, M., Blumberg, S., & Laartz, J. (2012, 10 1). *Delivering large-scale IT projects on time, on budget, and on value*. Retrieved from McKinsey: https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value

5. Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer, 21*(5), 61-72.

6. Bundschuh, M., & Dekkers, C. (2008). Functional Size Measurement Methods (FSMMs). *The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement*, 365-395.

7. Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems, 18*(4), 332-343.

8. Coelho, E., & Basu, A. (2012). Effort estimation in agile software development using story points. *International Journal of Applied Information Systems (IJAIS), 3*(7).

9. Cohn, M. (2004). *User stories applied: For agile software development.* Addison-Wesley Professional.

10. Crespo-Santiago, C. A., & Cosme, S. D. (2011). Waterfall method: a necessary tool for implementing library projects. *HETS Online Journal, 1*(2), 86-89.

11. Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology,, 50*(9), 833-859.

12. Fernandez, D. J., & Fernandez, J. D. (2016). Agile Project Management - Agilism versus Traditional Approaches. *Journal of Computer Information Systems, 49*(2), 10-17.

13. Fitsilis, P. (2008). Comparing PMBOK and Agile Project Management Software Development Processes. *Advances in Computer and Information Sciences and Engineering* (pp. 378-383). Dordrecht: Springer.

14. Highsmith, J., & Cockburn, A. (2001, September). Agile software development: the business of innovation. *Computer, 34*(9), 120 - 127.

15. Hoda, R., Noble, J., & Marshall, S. (2008, April). Agile project management. *New Zealand Computer Science Research Student Conference , 6*, 218-221.

16. Holtzman, J. (1999). Getting Up To Standard: PMI reaches a new plateau of global recognition for the project management profession. *PM NETWORK, 13*, 44-48.

17. Intland. (2017). *A 2017 Analysis of Agile Adoption Trends*. Retrieved from Intland Software: https://intland.com/blog/agile/a-2017-analysis-of-agile-adoption-trends/

18. Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme programming installed.* Addison-Wesley Professional.

19. Karlesky, M., Object, A., & Vander Voord, M. (2008). Agile project management. *ESC, 247*(267), 4.

20. Keaveney, S., & Conboy, K. (2006). Cost estimation in agile development projects. *ECIS 2006 Proceedings* (pp. 183-197). Gothenburg, Sweden: ECIS 2006 Proceedings.

21. Keung, J., Jeffery, R., & Kitchenham, B. (2004). The challenge of introducing a new software cost estimation technology into a small software organisation. *Software Engineering Conference, 2004. Proceedings. 2004 Australian.* Melbourne: IEEE.

22. Larman, C. (2004). *Agile and iterative development: a manager's guide.* Addison-Wesley Professional.

23. Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer, 36*(6), 47-56.

24. Lederer, A. L., & Prasad, J. (1995). Perceptual congruence and information systems cost estimating. *ACM SIGCPR conference on Supporting teams, groups, and learning inside and outside the IS function reinventing IS*, 50-59.

25. Lee, G., & Xia, W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *Mis Quarterly, 34*(1), 87-114.

26. Lipke, W., Zwikael, O., Henderson, K., & Anbari, F. (2009). Prediction of project outcome: The application of statistical methods to earned value management and earned schedule performance indexes. *International journal of project management, 27*(4), 400-407.

27. Marchewka, J. T. (2010). The FBI Virtual Case File: A Case Study. *Communications of the IIMA, 10*(2), 1-15.

28. Miller, G. G. (2001, July). The characteristics of agile software processes. *Tools*, 0385.

29. Morris, P., & Hough, G. H. (1988). *The Anatomy of Major Projects: A Study of the Reality of Project Management.* New Jersey: Wiley.

30. Osorio, J. A., Chaudron, M. R., & Heijstek, W. (2011). Moving from waterfall to iterative development: An empirical evaluation of advantages, disadvantages and risks of RUP. *Software Engineering and Advanced Applications (SEAA), 2011 37th*

EUROMICRO Conference on Software Engineering and Advanced Applications, 453-460.

31. Paetsch, F., Eberlein, A., & Maurer, F. (2003, June). Requirements engineering and agile software development, 2003. *Enabling Technologies: Infrastructure for Collaborative Enterprises*, 308-313.

32. Paul. (2016, June 28). *Agile Software Development Life Cycle.* Retrieved from Brain Technosys: http://www.braintechnosys.com/blog/agile-software-development-life-cycle/

33. Project Management Insitute. (2000). Chapter 7 Project Cost Management. In P. M. Insitute, *A guide to the project management body of knowledge (PMBOK guide)* (pp. 83-90). Newtown Square, Pennsylvania: Project Management Insitute, Inc.

34. Rajkumar, G., & Alagarsamy, D. K. (2013, 01). THE MOST COMMON FACTORS FOR THE FAILURE OF SOFTWARE DEVELOPMENT PROJECT. *The International Journal of Computer Science & Applications (TIJCSA), 1*(11), 74-77.

35. Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software, 17*(4), 26-32.

36. Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques. *Proceedings of the 9th international conference on Software Engineering* (pp. 328-338). California, USA: IEEE Computer Society Press Los Alamitos.

37. Schmitz, A., Tada, B., & Hess, R. (2014). *IT System Failures: The FBI's Virtual Case File Case Study.* Yonsei University, Graduate School of International Studies. Seoul: Yonsei University.

38. Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum (Vol. 1).* Upper Saddle River: Prentice Hall.

39. Schwalbe, K. (2016). Project Cost Management. In K. Schwalbe, *Information Technology Project Management 8e* (pp. 264-298). Boston: Cengage Learning.

40. Stepanek, G. (2005). *Software project secrets.* Springer.

41. Strike, K., El Emam, K., & Madhavji, N. (2001). Software cost estimation with incomplete data. . *IEEE Transactions on Software Engineering, 27*(10), 890-908.

42. Sulaiman, T., Barton, B., & Blackburn, T. (2006, July). AgileEVM-earned value management in Scrum Projects. *Agile Conference, 2006* , 10.

43. Truex, D. P., Baskerville, R., & Klein, H. (1999). Growing systems in emergent organizations. *Communications of the ACM, 42*(8), 117-123.

44. USDOJ/OIG. (2006). *Statement of Glenn A. Fine Inspector General, U.S. Department of Justice before the House Appropriations Committee Subcommittee on Science, the Departments of State, Justice, and Commerce, and Related Agencies concerning Oversight of the Federal Bureau of Investigation.* Washington D.C.: Office of the Inspector General United States Department of Justice.

45. USDOJ/OIG. (2010). *Status of The Federal Bureau of Investigation's Implementation of The Sentinel Project.* Washington D.C.: U.S. Department of Justice Office of the Inspector General.

46. USDOJ/OIG. (2010). *Status of The Federal Bureau of Investigation's Implementation of The Sentinel Project.* Washington D.C.: U.S. Department of Justice Office of the Inspector General.

47. USDOJ/OIG. (2011). *Status of The Federal Bureau of Investigation's Implementation of The Sentinel Project.* Washington D.C.: U.S. Department of Justice Office of the Inspector General.

48. USDOJ/OIG. (2012). *Interim Report on The Federal Bureau of Investigation's Implementation of The Sentinel Project.* Washington D.C.: U.S. Department of Justice Office of the Inspector General.

49. USDOJ/OIG Audit Division. (2005). *The Federal Bureau of Inverstigation's Management of the Trilogy Information Technology Modernization Project.* Washington D.C.: U.S. Department of Justice Office of the Inspector General Audit Division.

50. USDOJ/OIG Audit Division. (2006). *Sentinel Audit II: Status of The Federal Bureau of Investigation's Case Management System.* Washington D.C.: U.S. Department of Justice Office of the Inspector General Audit Division.

51. USDOJ/OIG Audit Division. (2006). *The Federal Bureau of Investigation's Pre-Acquisition Planning For and Controls Over The Sentinel Case Management System.* Washington D.C.: U.S. Department of Justice Office of the Inspector General Audit Division.

52. USDOJ/OIG Audit Division. (2007). *Sentinel Audit III: Status of The Federal Bureau of Investigation's Case Management System.* Washington D.C.: U.S. Department of Justice Office of the Inspector General Audit Division.

53. USDOJ/OIG Audit Division. (2008). *Sentinel Audit IV: Status of The Federal Bureau of Investigation's Case Management System.* Washington D.C.: U.S. Department of Justice Office of the Inspector General Audit Division.

54. USDOJ/OIG Audit Division. (2009). *Sentinel Audit v: Status of The Federal Bureau of Investigation's Case Management System.* Washington D.C.: U.S. Department of Justice Office of the Inspector General Audit Division.

55. USDOJ/OIG Audit Division. (2014). *Audit of The Status of The Federal Bureau of Investigation's Sentinel Program.* Washington D.C.: U.S. Department of Justice Office of the Inspector General Audit Division.

56. Vanhoucke, M. (2009). *Measuring time: Improving project performance using earned value management* (Vol. 136). Springer Science & Business Media.