

University of Mississippi

eGrove

Honors Theses

Honors College (Sally McDonnell Barksdale
Honors College)

2013

An Exploratory Study of Automated Tools to Support SCRUM- Based Project Management in Agile Software Development Teams

Nirmal Dharmaratne

Follow this and additional works at: https://egrove.olemiss.edu/hon_thesis

Recommended Citation

Dharmaratne, Nirmal, "An Exploratory Study of Automated Tools to Support SCRUM-Based Project Management in Agile Software Development Teams" (2013). *Honors Theses*. 1982.
https://egrove.olemiss.edu/hon_thesis/1982

This Undergraduate Thesis is brought to you for free and open access by the Honors College (Sally McDonnell Barksdale Honors College) at eGrove. It has been accepted for inclusion in Honors Theses by an authorized administrator of eGrove. For more information, please contact egrove@olemiss.edu.


An Exploratory Study of Automated Tools to Support SCRUM-Based Project Management in Agile Software Development Teams

©2013

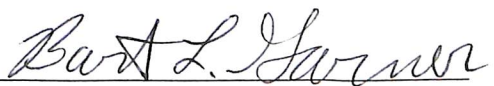
By Nirmal Hemantha Dharmaratne

A thesis presented in partial fulfillment of the requirements for completion
Of the Sally McDonnell Barksdale Honors College
The University of Mississippi
University, Mississippi
May 2013

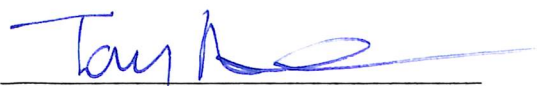
Approved by:



Advisor: Dr. Brian J. Reithel



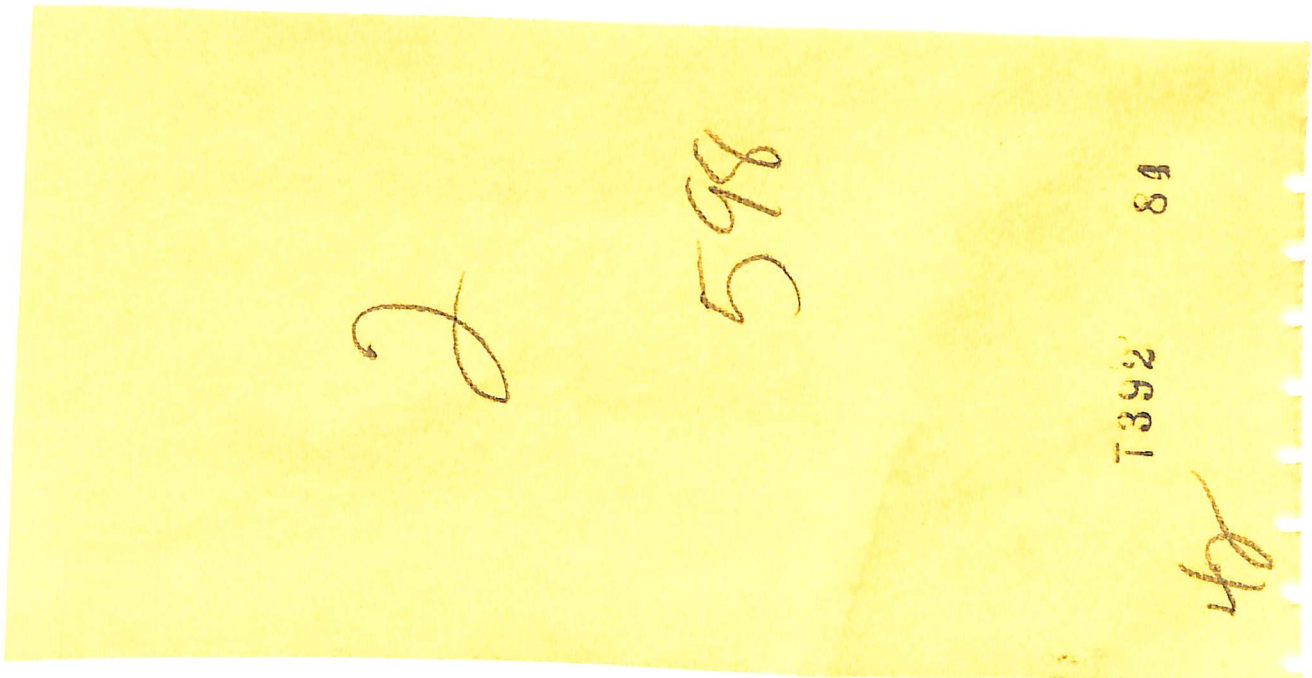
Reader: Dr. Bart L. Garner



Reader: Dr. Tony Ammeter

Abstract

The growing use of Agile Methodologies for project management has led many organizations to utilize agile-based project management tools. This thesis explores areas for improvement in project management tools for software development teams using the Scrum methodology. A qualitative research method was used to gain insight on the users' satisfaction and possible new utilities to create that would complement the OnTime Scrum project management tool. This paper further describes the methods and experiences encountered in the designing and implementation of a Work Breakdown Structure website tool that was selected as a utility to support project management.



Acknowledgements

First and foremost, I would like to thank my advisor, Dr. Brian Reithel, for agreeing to advise me throughout this research project. His advice and patience helped me immensely in making this thesis possible. Also, I am grateful for the guidance provided by the thesis readers, Drs. Bart Garner and Tony Ammeter. I would also like to thank my parents, brother and especially my sister for their support and encouragement. This thesis is dedicated for them for their unconditional support.

CONTENTS

INTRODUCTION.....	6
REVIEW OF LITERATURE	
Traditional Software Development Methodology.....	8
Agile Software Development Methodology.....	9
Scrum Software Development Method.....	10
Technology Acceptance Model.....	16
Work Breakdown Structures.....	16
OnTime.....	18
RESEARCH METHODS AND PROCEDURES	
Method.....	19
Research Instrument.....	22
Respondents.....	25
DATA ANALYSIS AND RESEARCH RESULTS	
Quantitative Analysis.....	27
Qualitative Analysis.....	31
Constraints and Suggestions.....	34
Conclusion.....	34
WBSMANAGER DOCUMENTATION	
WBS Website.....	36
Application Architecture.....	37
Screening Sequence of Events.....	38

Database Architecture.....	42
Testing and Evaluation.....	43
Suggestions for Improvement.....	45
CONCLUSIONS	
Conclusion.....	47
Limitations.....	48
Future Research.....	49
REFERENCES.....	50
APPENDICES	
Appendix A. Interview Script.....	52
Appendix B. WBSmanger code.....	56
Appendix C. WBSmanger database schema.....	62
Appendix D. WBSmanger Evaluation Questionnaire.....	63
Appendix E. Oral Consent Script.....	64

INTRODUCTION

The software development industry is changing direction by becoming more customers focused. Along with this shift in focus, there has been a rise in the use of agile methodologies for software development. These incremental and iterative agile processes focus more on satisfying the customers' needs. Among the agile methods, the Scrum methodology has established itself as a front-runner, with an increasing number of project teams switching to the Scrum methodology. Scrum uses repeatable work cycle called sprints. Sprints are the basic units of development in Scrum, with working components due at the end of each sprint. A key principle of Scrum is the recognition that customer requirements change during the project.

With the growing popularity of Scrum and other agile methodologies, tools have been created to assist in the project management process. With clients such as the U.S Army and Nike.Inc, OnTime is an agile project management tool that is amongst the most widely used. The OnTime software has a dashboard with customizable views to assist in the project management process. While the project management feature is the central feature of OnTime, it also supports other functions such as bug tracking and helps desk activities.

This study explores possible new utilities that can be used to assist in the project management process in Agile settings. An extensive literature review was done concerning the different software development methodologies. The main focus of this literature review was the Scrum methodology. Afterwards, a questionnaire was created to facilitate interviews of scrum-based project team members. The interviews were designed to gather information regarding user satisfaction and opinions on the new features they would like to see. The thesis will also present the results from the interviews. Upon reviewing the information gathered from these interviews, a new utility was selected to build. This new utility was selected based on the feasibility studies

conducted with the interview participants and thesis advisor.

Once the project was selected, it was constructed to incorporate the inputs and suggestions obtained through the interview process. After the completion of the project, an evaluation survey was conducted to predict usage. The procedures followed to build the project and the documentation for the project is presented along with suggestions for improvement. Finally, a set of conclusions, limitations, and directions for future research will be offered.

REVIEW OF LITERATURE

Traditional Software Development Methodology

During the early years of software development, most projects were small scale projects. Due to this reason, software developers did not pay much attention to planning and design. Instead, they used an ad-hoc method of writing code first and fixing bugs later, making it impossible to measure and catalog performance and the quality of the code. As time passed and the scale of projects increased, developers started encountering problems. Due to the lack of planning and design in these large-scale projects, developers were forced to spend most of their time fixing bugs instead of writing code. This led to the realization that some kind of methodology was needed to bring order to the software-development process. For this they looked toward plan-driven methods used in the engineering field. In these plan-driven methods from the engineering field, documentation of all requirements was more important than design, development, and implementation (Awad, 2005). These plan-driven software-development methods came to be known as traditional software-development methods (TSDM). One of the most widely accepted TSDM is the Waterfall model. The Waterfall model follows a sequential path between planning, analysis, design, implementation, and maintenance phases. In the Waterfall method each phase must be completed before going to the next phase, while you cannot go back to a previous phase, hence the name waterfall for this being strictly one directional. The planning stage involves identifying the scope of the system and performing technical, economic and organizational feasibility analysis. In the analysis phase, requirements for the system are gathered followed by the design phase where decisions about the system's software, hardware, and network operations are decided upon. The coding starts at the implementation phase. Finally in the maintenance phase support, training, and debugging is

performed. These TSDM worked perfectly in situations where all the requirements are well understood. Other traditional methods include the spiral model, RUP (Rational Unified Process), and object oriented systems analysis and design.

Agile Software Development

Most TSDM projects were frequently over budget, behind schedule, and were slow in adapting to the constant changes in business requirements. The solution for these limitations in TSDM was Agile Software Development Methods (ASDM). ASDM is an iterative and incremental adaptable system that requires minimal planning with fast development cycles, and is customer-centric with collaboration and many releases. ASDM focuses on providing high customer satisfaction through quick delivery of working software, active participation of concerned stakeholders, and by leveraging changing requirements (Highsmith, 2002). The active participation of stakeholders improves communication and cooperation within projects which increases the chances of fulfilling customer needs. Agile development compensates the lack of heavy documentation with continuous communication, while allowing the developers to focus on producing working software instead of documentation.

Compared to TSDM, ASDM differentiates itself by being people-centric, facilitating instead of controlling management, and by providing an organic project management style (Sørensen, Jense, & Holm, 2008). ASDMs provide higher customer satisfaction, lower bug rate, shorter development cycles, and quicker adaptation to rapidly changing business requirements (Boehm, 2002). Organizations that use agile tend to be organic with a flat structure when compared to the bureaucratic and tall structures of organizations using traditional methods. Other characteristics of agile development include short development iterations and self-organizing cross-functional teams. The use of self-organizing teams lessen the managers' duties and

responsibilities by handing more authority and decision making power to the team (Tata & Prasad, 2004). This results in faster and more accurate problem solving. Some examples of agile methods are eXtreme Programming (XP), Scrum, Crystal, and Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), and Feature Driven Development (FDD).

In 2001, a group of 17 agile software developers met in Snowbird, Utah, and formed the Agile Alliance, an association aimed at formalizing the agile methodologies. They created the "Manifesto for Agile Software Development," which revealed the practices that were considered valuable by ASDMs (Beck, 2001). They are as follows:

“individuals and interactions over processes and tools, **working software** over comprehensive documentation, **customer collaboration** over contract negotiation, and **responding to change** over following a plan.”

Furthermore, the agile manifesto states, “while there is value in the items on the right, the items on the left are valued more.” The core value of the agile method is the continuous adaptation of the development with the needs and expectations of the customer.

Scrum Software Development Method

Scrum is an agile software development process that is used to manage multipart software and product development tasks using iterative and incremental methods. The Scrum Alliance describes scrum as *“an agile framework for completing complex projects. Scrum originally was formalized for software development projects, but works well for any complex, innovative scope of work. The possibilities are endless. The Scrum framework is deceptively*

simple.” The term “Scrum” is borrowed from the sport of rugby, where: “[A] Scrum occurs when players from each team huddle closely together... in an attempt to advance down the playing field” (Highsmith, 2002). The Scrum analogy was used because of the similarities between a rugby team and a self-organized software development team that works to achieve a common objective with the guidance of the project manager. Scrum is ideal for project management situations with rapidly-changing user requirements and environments where it will be difficult to plan ahead (Schwaber & Beedle, 2001). The short iterations (sprints), continuous feedback through daily scrum meetings, and sprint retrospectives help face these challenges head-on. The Scrum process was formalized by Dr. Jeff Sutherland and Ken Schwaber (Sutherland & Schwaber 2007).

The following are important Scrum roles and techniques-

Scrum master: The Scrum master guides the development of team members and facilitates meetings while encouraging communication and creative input. They also support the collaboration process with stakeholders. The Scrum master’s goal should be to make sure the path is clear by removing barriers and overcoming hurdles to help the team work smoothly and more efficiently. The Scrum master is more of a facilitator and mentor to the team than a manager.

Sprint: A sprint is a short development iteration that usually lasts 1-4 weeks in length. The Sprint starts by setting goals and determining the functionality to be included in the upcoming sprint-release and a hard deadline for when the sprint ends. Teams are advised to choose a sprint-

duration and stick with it to ensure consistency. The length of the sprint is fixed and it is not changed even if the work is not completed (Deemer and Benefield 2007). Sprints help add a work rhythm to teams and also help in time estimation and planning.

Scrum team: Scrum teams tend to be small cross-functional groups with 5-10 developers. They are self-managing with high levels of autonomy and accountability. Their goal is to develop shippable software products at the end of each sprint. The team is in charge of deciding which tasks will be done within each sprint and how to arrange themselves with regard to the work at hand.

Product owner: The product owner is responsible for identifying product features and converting them into a prioritized list where the top-most gets chosen to be worked on for each sprint and put in the sprint backlog. The product owner will continuously refine and re-prioritize this list. The product owner is selected jointly by the Scrum master, the customer, and the management (Moe, 2009). Hence, the product owner works as a liaison to customer.

Product backlog: After the preliminary vision for the product is articulated, it is then converted into a basic requirements document which lists all the features the system must include. These are ranked according to the priority assigned to each by the customer or the product owner. This list consists of all high-level product and business requirements, technical features, bug fixes, procedures, updates/upgrades, major patches or other enhancements for the final product, which are often written in the form of "user stories". The product backlog continuously changes as the product owner changes the priority level of different features. The product backlog can be considered the management tool that facilitates management and coordination between software

development teams regarding the requirements, features, and project task dependencies. The product backlog can change frequently to reflect changes in customer needs.

User story: The user story is a way to reference product backlog items during the sprint planning meeting. "The user story should be written as a description of the desired functionality as seen from a user or the customers' perspective" (Cohn, 2004). User stories serve as a constant reminder to developers to communicate with the customer or product owner. Roles are typically incorporated into stories which allow them to be written on the form: *As a <user>, I want to <goal> so that <benefit>* (Behrenbruch, 2012). "The goal when developing a user story is to make it independent, negotiable, valuable, possible to estimate, sized appropriately and testable" (Cohn, 2004).

Daily stand-up meeting: "Every day during the sprint, typically during the first few hours, a brief stand-up meeting is held in front of the task board" (Rising & Janoff, 2000). "This meeting should last for maximum 15 minutes and everyone attending should stand during the meeting in order to ensure that it remains short" (Schwaber & Beedle, 2001). The goal of this meeting is for each member to update everyone about what they have done, what they will do, and any hurdles they might face. This will give an idea about the status of the project. However, this meeting time is not used to solve any problems that might arrive. Instead, the Scrum master will take note of the problems and try to solve them later (Schwaber & Beedle, 2001).

Sprint planning: Each sprint starts with a short planning meeting. "At these meetings the *product owner* and the team discuss the top priority items in the product backlog and their

perspective in order to achieve a common understanding of what to include in the sprint" (Andersen, 2009). In the next step, the development team goes through the product backlog starting from the top and picks the items that they think they can complete during the forthcoming sprint. This is one of the key practices in Scrum because "rather than the managers deciding how much the team must complete, the team itself arrives at a consensus regarding the workload they will take on"(Deemer & Benefield 2007). The team sets acceptance criteria for what they would consider as a shippable product. Shippable products are working software that a customer can evaluate. The next step is to perform detailed planning on implementing the backlog items that were chosen by the team. The chosen items are then placed on story cards where the time needed to finish each story card is estimated by the team. Then, the tasks will get distributed among the team.

Sprint backlog: The sprint backlog is a detailed work-break down on what each team member should be doing during the sprint. The sprint backlog is filled with the items and requirements from the product backlog that were selected during sprint planning for completion in the current sprint. The sprint backlog should remain unchanged during the sprint, unless a sudden critical task arises (Moe & Aurum, 2008).

Burndown chart: The burndown chart is a helpful tool used in Scrum to show progress and highlight divergence from the initial time estimates that were decided during sprint planning. The burndown chart is updated after completion of tasks since the last daily stand-up meeting. Using the burndown chart, the scrum master and the team can easily determine at an early stage

whether they are on track or not. The x-axis of the burndown chart indicates time while the y-axis indicates the total remaining amount of story points.

Sprint Review: A sprint review is held at the end of each sprint to show the product owner the result from the sprint, in order to get the final approval. The attendees of this meeting are the team, the scrum master, the product owner and any other individual who takes interest in the sprint output.

Sprint retrospective: After the sprint review, a sprint retrospective meeting is held where the team reflects on the finished sprint to in order to gain insights on how to be more efficient and effective in the future. The attendees of the sprint retrospective are the team and the scrum master.

Scrum of Scrums: A weekly or biweekly inter-team meeting attended by one representative from each team (chosen by the team) and the all the scrum masters. These meetings tend to be short. The meetings are similar to stand-up meetings, but talk of the progress from a team perspective. The meetings address issues such as what your teams have done, will do, and the hurdles the teams will face in the future.

Team velocity: Team velocity is a measure used when managing Scrum projects that represent how much of the product backlog list a team is able to deliver in one sprint. It is measured in story points (or hours) per sprint. During sprint planning, the team velocity can be used calculate

the amount of work they think they can deliver. Teams usually achieve a stable velocity after several sprints.

Technology Acceptance Model

Influenced by the intention models from psychology and the research of Ajzen and Fishbein on the "Theory of Reasoned Action," the Technology Acceptance Model (TAM) was developed by Davis and Bagozzi in 1989 with revisions made by Venkatesh et al. TAM is designed specifically for modeling and predicting information technology acceptance and usage. The core constructs of TAM consist of Perceived Usefulness and Perceived Ease of Use. Perceived ease of use is the extent to which a person believes using the system will be free of effort. Perceived usefulness is the extent to which a person believes that using the system will enhance their job performance (Davis & Venkatesh, 2004). Davis theorized perceived usefulness to be a better determinant of intention of use than perceived ease of use. The reason behind this theory is that as the user gains hands-on experience on the system, the effect perceived ease of use on intension of use diminishes, thus making perceived usefulness the major determinant of technology acceptance and usage.

Work Breakdown Structures

The Work Breakdown Structure (WBS) shows the work tasks that are included in a project. This is used as a tool to help managers easily communicate with employees the work and processes involved to execute a project. This technique has been used successfully in managing portfolios, programs, and projects. The WBS is being used by many diverse but related users, such as project managers, product owners, contractors, and suppliers. This is used for

communication among managers and the workers who are performing the work required for a successful completion of a program, portfolio, or a project. There are many formats of presenting WBS, with the differences being the varying levels of detail.

With the help of the project team, the project manager uses the WBS to develop the project schedule, resource requirements and costs. In WBS, the term Level of Effort (LOE) is used to describe the amount of work required to complete a certain task. A component of a WBS which is located at any level can be a Work Package or a WBS Element, as there is no restriction on what may be considered to be a WBS Component. A WBS element is a single WBS component and its associated attributes are located anywhere within a WBS. A WBS element can contain work, or it can contain other WBS elements or work packages. A work package is a deliverable or work component at the lowest level of its WBS branch.

"WBS provides a framework for specifying the objectives of the program in terms of hierarchically related product-oriented elements which provides logical summary points for assessing technical accomplishments and for measuring cost and schedule performance" (MIL-HDBK-881x). WBS also serves as a coordinating medium. Technical, schedule, and cost data are routinely generated through WBS for reporting purposes. "WBS also helps summarize data for successive levels of management and provide the appropriate information on the projected, actual, and current status of the elements for which they are responsible. The WBS keeps the program's status constantly visible so that the program manager, in cooperation with the product owner, can identify and implement changes necessary to assure desired performance"(MIL-HDBK-881x).

With regard to the Scrum methodology, WBS helps increase the pace of completion, decreases steady-state project lists, and improves team communication. Through the series of

lists and meeting used to monitor the teams tasks and progress, the WBS feeds off the product backlog list which outlines the deliveries and expectations from the customer. The purpose of the WBS within the Scrum methodology is to serve as a Sprint Backlog. This ensures that the intent of the customer drives the scope of work, which, in turn, drives the requirements and subsequent actions taken by the project team (Pries, Kim H., and Jon M. Quigley, 2011). Since the WBS is a functional decomposition of top-level deliverable elements, the WBS should change as the requirements change in the product backlog. The WBS assures that the necessary actions are taken to produce the product or service that the customer demands on time.

OnTime

OnTime is a feature-rich proprietary project management and bug tracking system designed specifically for software projects (Vogel, 2010). OnTime was developed by the software development company Axosoft, LLC. Major customers of OnTime include Nike Inc, Intel, EA Sports and the U.S. Army. By using OnTime, project managers and developers are able to visually see each task, requirement, defect and incident in the system on individual filing cards through the Scrum planning board (Serignese, 2010). The version of OnTime currently used by the project team in my case study is OnTime Scrum. OnTime Scrum dashboard supports agile project management for project teams following the Scrum methodology. The features provided by OnTime include bug tracking, requirement management, planning board, burndown charts, and help-desk incident tracking assistance, and reporting.

RESEARCH METHODS AND PROCEDURES

Method

Since the research was an exploratory study, qualitative research methods were used to study and analyze the complexity of the issues Scrum-based project teams face. The starting point of this research process was an extensive literature review on the subject of software development methodologies. A significant portion of the literature review was conducted on the topic of Agile Software Development and, specifically, the Scrum process of project management.

After gaining the required background knowledge on the subject area, the process of preparing a questionnaire to assist in the interviews was started. The questionnaire consisted of both structured and unstructured questions. The items in the survey section of the questionnaire were designed to obtain information regarding the extent to which the project team was satisfied with the ease of use and the functionality of the OnTime Scrum project management tools. OnTime software's functionality was measured by the extent to which OnTime achieved functional goals, met work requirements, satisfied user needs, and met technical requirements. OnTime software's ease of use was measured by assessing the amount of effort required to become familiar with the system. To assess the amount of effort required, questions regarding the visual interface and intuitiveness of the OnTime Scrum system were asked. More information regarding the questionnaire can be found in the Research Instrument section of this thesis.

The main objective of the interview was to gain insight in to the daily use of the OnTime software by the sample project group. The interview and survey respondents were members of a Scrum-based project team that produces FDA-regulated systems for the healthcare industry. The team members were geographically distributed. More details about the sample project team will

be provided later in the Respondents section. The interview consisted of two parts. First, the structured survey questions which were used to measure the perceived ease of use and functionality of OnTime Scrum users. The second part consisted of unstructured open-ended questions. The reason behind having the open ended questions was to gain user input regarding possible improvements, new features and the short-comings of OnTime.

Before conducting the interviews, approval of the Institutional Review Board (IRB) was obtained because this research involved human subjects. Before submitting the application to conduct research with human subject, all investigators involved in this study were required to complete the appropriate course provided by the Collaborative Institutional Training Initiative (CITI). The course followed the human research curriculum. After the completion of the course, a report stating completion of the required modules and associated quizzes was given to the researcher. The interview questions, scripts and the consent forms had to be filed along with the completed IRB application. After getting the approval of the IRB, contact was initiated with the interviewees.

The interviewees were initially contacted through email in order to schedule phone interviews. The reason phone interviews were chosen instead of a having the participants fill out a questionnaire was due to the ability to ask follow-up questions when conducting phone interviews. The script of the interview was shared with the interviewee thirty minutes prior to the interview. This was to done in order to improve communication while not giving the interviewee enough time to prepare preset answers. The first step of the phone interview was to get the interviewees consent to perform the interview. For this, an oral consent statement was used where the participant had to either agree or disagree to allow me in performing and recording this interview. The consent statement is provided on the Appendix E.

Upon the conclusion of the interview, the recordings were transcribed to help in reviewing the interviews. The data gathered from the interviews was broken into two segments. The survey results were used to analyze the satisfaction with regards to ease of use and functionality. The results from the second part of the interviews were reviewed to obtain information on possible utilities to support OnTime Scrum. With the information collected from the open ended questions, a list of possible new utilities was developed. These new utilities provided solutions to the short-comings, and the new features the users sought. The list of new utilities and the reasoning behind the suggested utilities appear in the Data Analysis and Research Results section of this thesis.

In order to pick a utility that was feasible to build, a feasibility analysis was performed on all of the possible new utilities. The feasibility of each project was assessed based on technical, operational, and schedule feasibility. The researcher's ability to build the utility was considered under technical feasibility. For this process, the project risk factors that were contingent upon the project size and structure were taken into account. For operational feasibility, the question of whether the new utility will provide solutions to the problems or the short-comings that OnTime Scrum had were assessed. As for schedule feasibility, the question of whether this project can be completed within the limited time frame was considered. Using the above mentioned criteria, a utility was selected to build with the hopes of improving the sample team's performance.

The utility that was selected to build was a work breakdown structure to help the team distribute the work requirements more efficiently. The work breakdown structure was built in the form of a web application. Both PHP and HTML programming languages were used in building this web application. After the web application was built, user acceptance testing was done in the form of a survey to evaluate and predict the use of the website.

Research Instrument

The questionnaire that was used for the interview was constructed to measure and assess user satisfaction and to gain insight into possible new utilities for OnTime Scrum to make it more user-friendly. The questionnaire consisted of two parts, in which the first part measured user satisfaction and the second part gathered information on shortcomings of OnTime Scrum, new ideas, and possible improvements to OnTime Scrum.

The questionnaire was constructed based on the extensive review of literature in the areas of customer satisfaction, technology acceptance model, SERVQUAL model, system implementation, and computer-system usability. Many survey questions were adapted from existing literature and suggestions by scholars in these areas. The questionnaire had a total of 28 questions, in which the first 14 questions were structured questions and the rest of the questions open ended. The first half of the questionnaire on user satisfaction covered the areas of ease of use, usefulness, security and dependability, and information quality of OnTime Scrum. Among the focus areas, greatest emphasis was given to the area of information quality since the main function of project management is the transfer of information. Questions regarding the usefulness of OnTime Scrum as a project-management tool largely depended on the quality of information. These questions used a five-point psychometric scale like the Likert scale where 1=strongly disagree and 5=strongly agree.

To measure the ease of use, questions were asked such as, "I find OnTime Scrum easy to use," "Pages within OnTime Scrum are easy to interpret," and other questions regarding the ability to automate processes and the level of customization. These questions assessed the intuitiveness of operations and the amount of control the user has over the data, which is a major factor in the ease of use. Meanwhile, the usefulness of the tool was measured by questions such

as " All my project management needs can be completed via OnTime Scrum," " I am satisfied with the depth of information on the OnTime Scrum dashboard," "I am satisfied with the release flexibility of OnTime Scrum," and "I would recommend OnTime Scrum to a colleague interested in project management." While the question regarding the completion of all project management needs measured the overall usefulness of the program, questions regarding the depth of information and release flexibility were more specific. The information depth covered how deep the information displayed goes or the level of detail provided by OnTime. The release flexibility looked at the ability to switch between different product releases. The question regarding recommending the program to a colleague asked the question of usefulness in a different manner.

Questions regarding the security and dependability of OnTime Scrum were also asked. The question "I trust OnTime Scrum to keep my project information safe" was intended to get insight into how much the users trust the program with their data. The question "I am satisfied with the ability to recover of OnTime Scrum" measured the user's satisfaction regarding recovering lost data. Although a simple question, the question "OnTime Scrum is dependable," was used to measure the dependability of the tool, which reflected the user's perceived consistency of performance of OnTime. The quality of information was measured using the criteria of completeness, customizability, security, depth, and dependability of the information received. For this, questions such as, "OnTime Scrum adequately meets my information needs," and "Interaction with OnTime Scrum allows me to receive tailored information about my project" were asked.

The second section of the questionnaire facilitated gathering new ideas and information on areas to improve OnTime Scrum. The questions were open ended in order to gain as much

insight as possible. To find out whether there are errors that need to be fixed to improve the quality of information, the question, "Have you encountered errors in OnTime Scrum where the dashboard displayed information you knew to be false?" was asked. In the initial review of OnTime, the researcher felt the need to improve certain aspects of the visual planning board feature of OnTime Scrum. Therefore, in order to check whether the proposed improvement was necessary, two questions were asked: "How quickly can you spot changes and/or progress in the visual planning board?" and "Does the information make sense as quickly as it would if it were a scoreboard of a game?"

To figure out the shortcomings of OnTime Scrum, questions such as, "How often do you use pen and paper in the project management process?" and "If you use pen and paper, when do you use it?" were used. The goal of the first question was to find areas OnTime Scrum falls short and makes the user rely on pen and paper to complete the task. From the information gathered through this question, the researcher was able to digitize the process where the user has to use pen and paper.

Questions such as "Which tasks in OnTime Scrum take the longest amount of time?" were asked to find areas to improve performance and efficiency of the project-management tool. The question, "What are the features in OnTime Scrum that you use the least?" was asked to identify features with room to increase usability. To gain further insight into what possible improvements there could be the question "What types of improvements would make you use those features more?" was asked. In order to find out suggestions the users have regarding possible new features, the questions "What new features would you like to see in OnTime Scrum?" and "If you could only have one new feature to support agile development, what would it be?" were asked.

To check for features that one could combine to increase efficiency, participants were asked the question, "Are there features in OnTime that you always use concurrently?" To gain insight on features and characteristics of those features the users appreciate the questions "What are the features in OnTime scrum that you use the most?" and "Which characteristic of these features that you find most attractive," were asked. The question "Which feature on OnTime scrum would you consider as ground-breaking," was asked to figure out the features that increased the users' productivity the most. To find out the feature that has the lowest usability for the user, interviewees were asked the question "if you had to give up a feature from OnTime Scrum, which feature would it be?" was asked. The aim of the questions above was to use the attributes of the most-used features to improve the quality of the least-used features. The question, "Are you able to display things that are not done in a Sprint using OnTime Scrum?" was asked to find areas to improve outside communications of OnTime. The complete questionnaire can be found in Appendix A.

Respondents

Participants of the research study were members of a Scrum-based project team in a software development company in the healthcare industry that produces FDA-regulated information systems. The team members were geographically distributed throughout the United States. The individual team members represented various backgrounds and cultures. The team members were highly experienced in their respective fields with years of work experience ranging from 18 to 33 years. The roles and the years of experience of the interview respondents are as follows: senior Java developer and systems architect with 29yrs of experience, software quality assurance specialist with 33 years of experience, project manager and end-user support manager with 27 years of experience, senior Java developer with 23 years of experience, and a

Java developer with 18 years of experience. All the team members used the OnTime Scrum project managements system to varying degrees ranging from using the system slightly to using it extensively. The reason for not including at least about 10-20 participants in this research is because this is an exploratory study that aims at paving way for future research on Scrum-based project teams. Thus, the data collected was largely qualitative. Another justification for the small sample size is the assumption of high reliability of the response based on the collective years of work experience amongst the team.

DATA ANALYSIS AND RESEARCH RESULTS

Quantitative Results

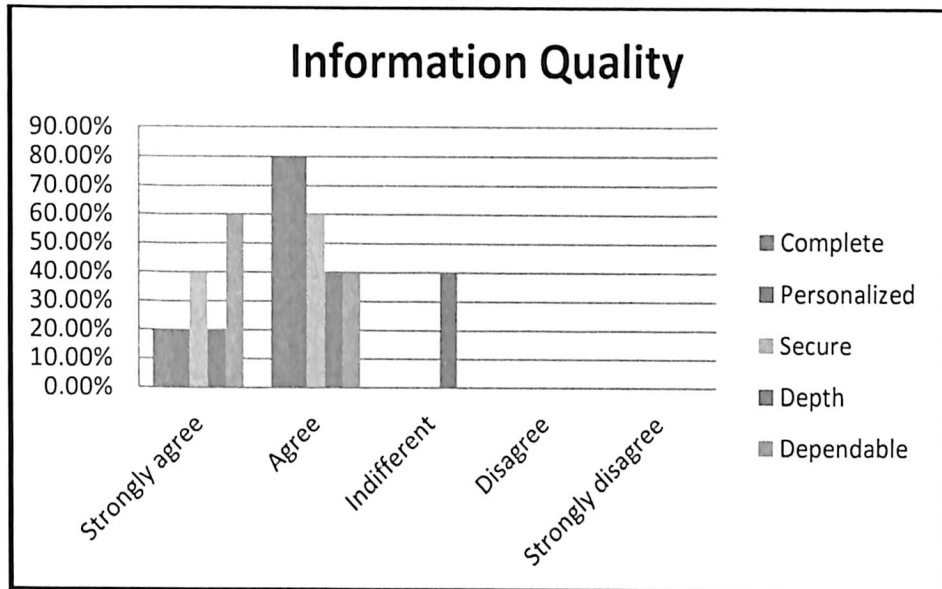


Figure 1.1

Figure 1.1 above shows the results of the information quality portion of the survey.

The users were asked whether or not OnTime Scrum adequately meets their information needs to measure satisfaction with regard regarding the completeness of information. As indicated by the figure above, 20% of the users responded “strongly agree” while the rest responded “agree.” For the personalized information criteria, when asked whether they could receive tailored information from OnTime, 20% of users responded “strongly agree” while the rest responded “agree.” To measure the satisfaction regarding information security, the users were asked if they trust OnTime Scrum to keep their information safe, to which 40% of the users strongly agreed with the rest just agreeing. When asked whether the users were satisfied with the depth of information of OnTime, 20% of the responded strongly agreed while 40% each responded

“agree” and “indifferent”. When asked about the dependability of OnTime Scrum, 60% of the users strongly agreed with the rest agreeing. When looking at **Figure 1.1**, we can see that the overall user satisfaction regarding information quality is high with a strong sense of security and dependability toward the product.

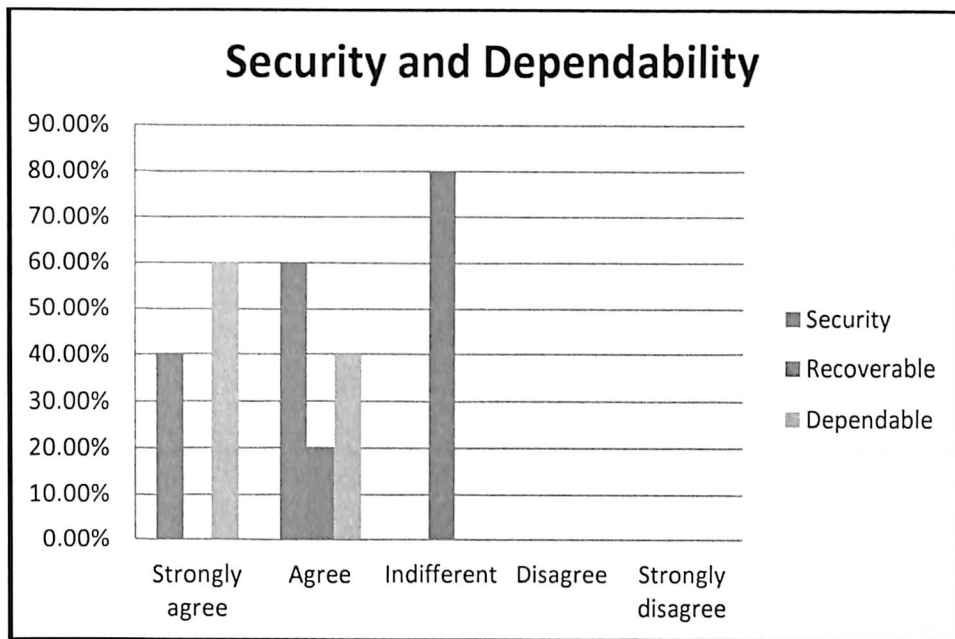


Figure 1.2

Figure 1.2 above shows the results of the security and dependability portion of the survey. To measure the satisfaction regarding security, the users were asked if they trust OnTime Scrum to keep their information safe, to which 40% of the users strongly agreed with the rest just agreeing. The users' responses to the level of satisfaction in the recoverability of OnTime were 80% responding indifferent while 20% agreed. The answers to the follow-up questions indicated that many of the users never encountered any situation where they had to recover lost data, thus explaining the significant amount of indifferent responses. When asked about the dependability of OnTime Scrum, 60% of the users strongly agreed with the rest agreeing. When looking at

Figure 1.2 we can conclude that the overall sense of security and dependability of users are high with recoverability being a non-issue.

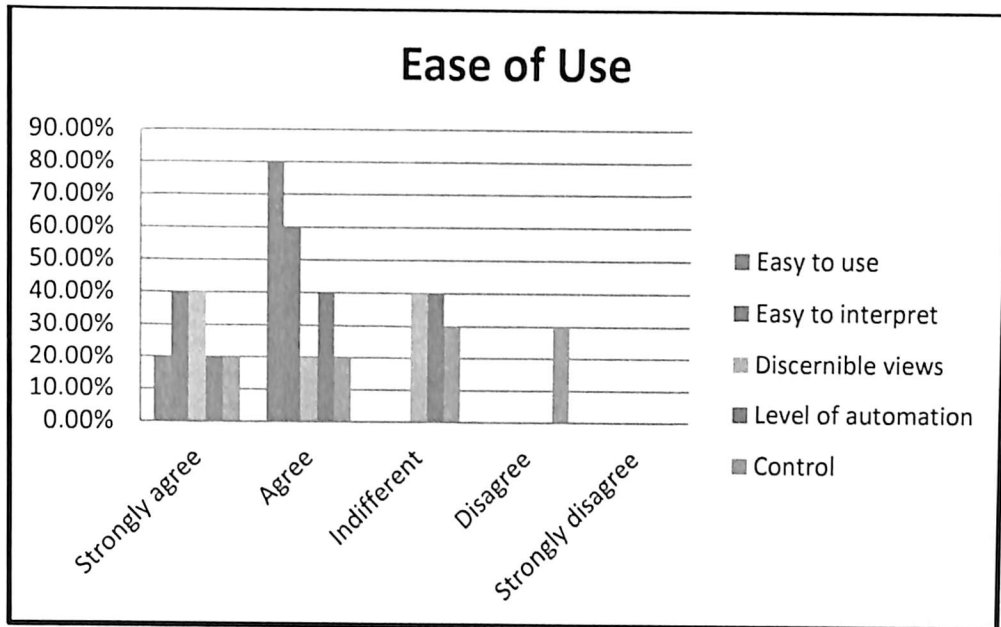


Figure 1.3

Figure 1.3 above shows the results of the ease of use portion of the survey. When asked about their overall satisfaction with the ease of use of OnTime, the users responded with 80% agreeing and the rest strongly agreeing. To measure the ease of interpretation, the users were asked whether the pages within OnTime Scrum were easy to interpret. To this question, 60% of the users agreed and 40% strongly agreed. Under the criteria of discernible views, respondents were asked if they find OnTime Scrum visually appealing. To this question, 40% responded with “strongly agree,” 40% with “indifferent,” while the rest responded with “agree.” When asked to assess the level of satisfaction regarding automation, 40% responded “agree,” 40% “indifferent,” and the rest said “strongly agree.” To measure the amount of control the user has on the product, the users were asked whether they find it easy to use OnTime Scrum to do exactly what they want it to do. To this question, 30% responded with “indifferent, another 30% responded with

“disagree,” 20% with “strongly agree,” and the remaining 20% with “agree.” By looking at **Figure 1.3** we can see that the overall user satisfaction regarding the ease of use varied among the users.

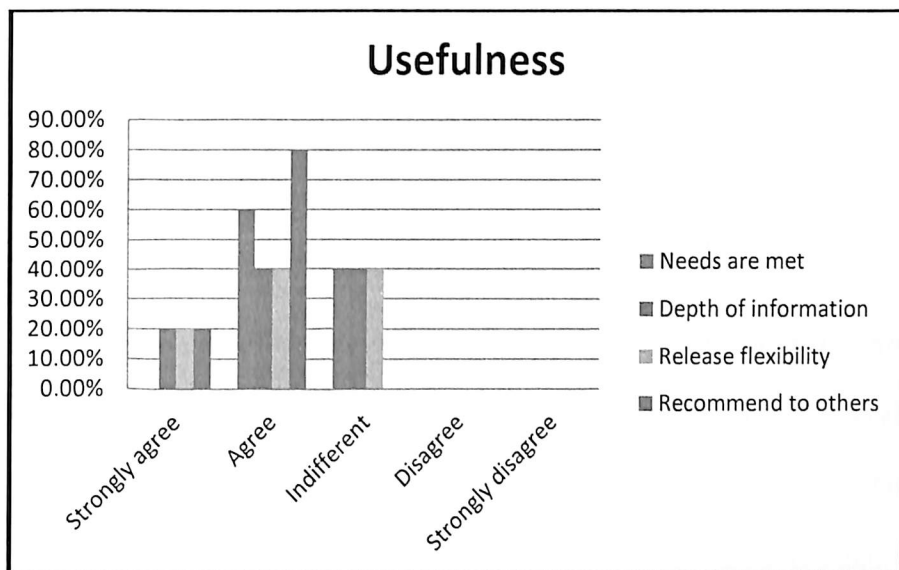


Figure 1.4

Figure 1.4 above shows the results of the usefulness of OnTime Scrum. For the criteria of needs being met, the users were asked if all their project-management needs can be completed via OnTime Scrum. For this question, 60% of the users agreed with the rest responding with “indifferent.” Some users mentioned their primary role was not project management, which could explain the high number of the response “indifferent.” To measure the user satisfaction regarding the depth of information, the users were asked if they are satisfied with the depth of information available on OnTime Scrum. The users responded with 40% each selecting “agree” and “indifferent” while the remainder selected “strongly agree.” When asked about the satisfaction regarding the flexibility of OnTime Scrum, 40% each responded “agree” and “indifferent” while the rest strongly agreed. To see whether the users were satisfied enough to

recommend the product to others; the users were asked if they would recommend OnTime Scrum to a colleague interested in this project management. To this question, 80% of the respondents agreed while the remainder strongly agreed. When looking at **Figure 1.4** we can see that the overall user satisfaction regarding the usefulness of OnTime Scrum is average with a high likelihood of users recommending the products to others.

Qualitative and subjective responses

The second section of the interview had 14 open-ended questions. Each question could result in follow-up questions when necessary. The responses obtained indicated varying degrees of the ease of use when using OnTime Scrum amongst team members. When asked about their level of use, the interviewees responded with responses such as "I don't use many of the features of OnTime," and "I don't use OnTime really in a project management sense as a tool, it's more of a helpdesk that I use to respond to our customers' needs." One respondent further stated that the while OnTime is a project management tool, they do not "operate it that way." Instead, they "use a work breakdown methodology that is separate from OnTime."

When asked about the quality of information, one responded, "You get out of it what you put in it." The user further mentioned that "if you don't categorize, and organize the data properly, then you are not going to get the right information." Regarding the ability put in and get out data from OnTime, a user stated "It's easy to put information in, but it's hard sometimes to get the information out."

When asked questions regarding the visual planning board which can be considered the central feature of OnTime Scrum, many responses indicated that the users rarely used the visual planning board. "I've never used the visual planning board. I think it looks like an electronic white board," "Our planning board is an Excel spread sheet with vertical swim lanes, where we

have each resource across the x-axis and is setup where we can see who's doing what this week," and "I think OnTime tries to do the same, but we really don't use it," were some of the responses. The above comment suggest the users avoid using OnTime and rely on other tools, due to the difficulty of learning to use new features of OnTime or to their own team management practices.

When asked about the overall ease of use, the respondents stated that they have had some difficulty learning to use the program, "but once you get how it works, it's easy." Another user said that "there is a little bit of a learning curve when it comes to OnTime," and that "it would have been nice if OnTime were intuitive." One user mentioned that in order for them to use different features, they "had to have a particular need, and then find how to do it." The user also indicated that "it's a little bit tricky to set up the appropriate views. There are lots of different things you need to get right in order to view the data."

The features used the most by users depended on the user's job with "time recording" being the most common answer. When asked why, the respondents stated that they "have a strict contractual requirement to record time on a granular level." Other features that the users mentioned include the help-desk, bug reporting, and the wish list. When asked about the attributes they like about those features, the respondents provided responses like, "The attribute I like most about the helpdesk is that with a push of a button, I can know who's waiting for what from me," "The wish-list is really just the new features," And "I like the tabular data views. The way that we use it, we can add, remove, reorder, and sort columns."

The tasks that took the longest amount of time depended on what the user's job was. It was suggested that "moving items from list to list takes time, because it is a very manual process," and that "it would be nice if it can be automated." Among the responses regarding the tasks that take the longest amount of time, the following answers give insight on how the project

teams operate:"All workflow requires a lot of steps to wrap-up an item that you have finish working on. After that, we pass the item around to add information that can be used in the next level. The purpose behind this is to easily complete FDA documentation since we are a regulated industry, "and "We use OnTime predominately for client tracking."

The features of OnTime that were used the least by the user vastly differed from what was initially expected. The most common feature the users did not use was the planning board."We really don't use the dashboard," said one user. "We have a modified version of Scrum; we do have backlogs, iterations, sprint planning, and the weekly deliverables," and " So we organize things like the relationship between things to be implemented and the things that were completed, but using Excel instead of OnTime," were two other responses.

One user stated that they use "pen and paper for keeping track of items in their help desk." A possible new feature that could be built base on the comment above would be a feature where the user can put a tracker on help-desk items. Another user mentioned that pen and paper was used "for fishing out rough ideas and brainstorming." When asked for features used concurrently, it was mentioned that "it would be nice to have bug tracking, time recording, and the wish list in one page view." Another user said that they "use OnTime for a lot of their correspondence" and that it would be nice to "read emails associated with the project while editing and changing the project the same time." The users had conflicting opinions regarding reporting, with one use saying "reporting is terrible;" while another said "I think the reports are excellent and easy to organize with not a lot of chunk data in it."

Thoughts on possible new features and improvement that the respondents shared are as follows. From a developer's perspective, a user suggested, "Linking a bug with our code to show where in the code there is a bug." Another user also suggested, "Integration with source code

control, like subversion." It was also suggested that, "Automating the procedures for moving items from list to list," while keeping the items connected would immensely help their productivity. Another user mentioned that "it would be nice to have a visual breadth calendar where I can see color coded entries like green for complete." It was further suggested to simplify the dashboard to one view. The same user stated that while the current dashboard is customizable, you "had to go line by line to see which buckets have been entered to the workbox.

Constraints and Suggestions

As with any research, this study was not immune to limitations. The first limitation was the study being focused on a single project management tool. This constrained the audience of the research instrument. This analysis of this research was narrowed down to the Scrum software development process. Hence, the finding of this research cannot be generalized to the greater population using other methods. Due to the limitations in terms of the small sample size of respondents, less weight was given to quantitative data. Further research should focus beyond project management into other areas of Scrum, such as requirements gathering and quality testing. The questionnaire focused only on the project management aspects of OnTime Scrum. For future research, it is suggested that focusing on all aspects of OnTime Scrum while interviewing a larger sample size might be productive. Future research shouldn't be solely limited to OnTime.

Conclusion

It can be inferred from results and analysis that OnTime Scrum is used as a group work support tool, rather than a project management tool by the sample project team. Due to the lack of intuitiveness of OnTime or the team's own preferred practices the users focus only on the

product features that they are required to use. Instead of using OnTime, the project team uses a Work Breakdown methodology with the help of Microsoft Excel to support their project management needs.

Based on the results and suggestions from the interview, the following list of possible utilities was created.

1. Source code integration feature
2. Work Breakdown Structure website
3. Tracker for helpdesk items

The Work Breakdown Structure website was selected as the utility to build for part two of this thesis. The reasoning behind selecting of the Work Breakdown Structure website was due to the feasibility of the project being high. This website also had the potential to reach a much larger audience when compared to the other two options which were limited only to OnTime users. The reasons for not choosing source code integration and a tracker for helpdesk items was due to the scope of these projects being beyond the researcher's capabilities. These other projects also would have required access to the source code of OnTime. Being a commercial product, the likelihood of AxoSoft, who are the developers of OnTime, sharing the source code was slim. It was estimated that both these projects will not fit into the study's time constraints. It was further determined that the researcher did not have the level of expertise required for the completion of these two complex projects.

WBSMANAGER DOCUMENTATION

WBS Website

The purpose of the WBS website is to facilitate the project manager of the sample project team in the efficient distribution of Work Breakdown Structure (WBS) between the geographically distributed project team members. The main advantage that the website offers is the instant distribution of the WBS. In the previous method used by the team, the WBS was saved in a Microsoft Excel spreadsheet that was distributed among the team members through e-mail. Through the use of the website that was created for this study, the users will have access to the WBS anywhere and anytime using a web browser.

Requirements for this website were gathered from initial interviews of the project team members. The open-ended questions from the second section of the interviews, along with the follow up questions provided insight on how to improve the WBS distribution. Following the Agile approach of incremental and iterative development, a prototype was developed with constant feedback from the Chief Information Officer (CIO) of the sample project team's organization. First, a rough design of how the WBS table should look was developed. Next, the rough design was shared with the CIO. Using his feedback, the table format was redesigned to be more suitable for the sample project team. The first prototype was created using the PHP coding language. This prototype included only the basic functions such the ability to send and retrieve data. More visual styling was added to the second prototype. The final product incorporated easy navigation and intuitive design, along with the required functions. During this whole process, the researcher maintained communication with the CIO to obtain constant feedback.

Application Architecture

In the Application Architecture section, the various components of this website and the interaction of web pages in the context of the whole website will be defined. This web application follows a three-tier architecture. These three tiers are presentation, application and data.

Presentation Layer: For the presentation layer, the standard Internet Browser is the primary client. From this layer, HTML pages are displayed to the client browser upon a user's request. These web pages also included JavaScript and CSS functions that were invoked using the web-designing tool Twitter Bootstrap. Twitter Bootstrap contains HTML and CSS-based design templates for typography, forms, buttons, charts, navigation, and other interface components, as well as optional JavaScript extensions based on the jQuery JavaScript library. This layer also served as the front-end information layer. User input is collected from the various forms and posted to the application layer. Information from the application layer is taken is also displayed in this layer.

Application Layer: The application layer was written using PHP exclusively. Included in this layer were various PHP scripts that were designed to do tasks such as user authentication for the login page.

Data Layer: The database is located in this layer. By keeping the data layer independent from the application layer, performance and scalability was improve. Data Access Objects using MySQL managed the interface to the database. Direct access to the database for administrative purposes was gained using phpMyAdmin. According to the phpMyAdmin website, phpMyAdmin is a tool written in PHP, intended to handle the administration of MySQL over the internet.

Screening Sequence of Events

The user could be any authenticated and authorized user of the WBS management website. There are two types of users in this website. The user could be either an admin or an employee. From the login page, the username and password entered by the user are posted to the PHP script that handles user authentication.



Starting from this PHP page, session variables are used to keep track of the user until the user logs out. The user authentication PHP script connects to the database to check whether the posted username and password match any of the usernames and passwords stored in the MySQL database. For protection against MySQL injection attacks, each non numeric user supplied value that is passed is quoted with the following database-specific string escape function: `mysql_real_escape_string()`. This user authentication script will retrieve from the database table rows from the Users table where the Username and Password equal the username and password supplied by the form. Then the amount of table rows retrieved is counted using the function `"mysql_num_rows."` This built in PHP function counts the numbers of table rows retrieved from the database. Next, an "if" statement checks whether the count equals to 1 or not. If the count equals to 1, it means that the username and password submitted by the users exists in the Database, and therefore the user is authenticated.

```

1 <?php
2 session_start();
3 $hoste="localhost"; // Host name
4 $username="root"; // Mysql username
5 $password="root"; // Mysql password
6 $db_name="testdatabase"; // Database name
7 $tbl_name="users"; // Table name
8 // Connect to server and select database.
9 mysql_connect("$hoste", "$username", "$password")or die("cannot connect");
10 mysql_select_db("$db_name")or die("cannot select DB");
11 // username and password sent from form
12 $myusername=$_POST['myusername'];
13 $mypassword=$_POST['mypassword'];
14 $_SESSION['myusername']=$myusername;
15 // To protect MySQL injection
16 $myusername = stripslashes($myusername);
17 $mypassword = stripslashes($mypassword);
18 $myusername = mysql_real_escape_string($myusername);
19 $mypassword = mysql_real_escape_string($mypassword);
20
21 $sql="SELECT * FROM $tbl_name WHERE Username='$myusername' and Password='$mypassword'";
22 $result=mysql_query($sql);
23 // Mysql_num_row is counting table row
24 $count=mysql_num_rows($result);
25 // If result matched $myusername and $mypassword, table row must be 1 row
26 if($count== 1){
27 // Register $myusername, $mypassword and redirect to file "login_success.php"
28 header("Location:login_success.php");
29 }
30 else {
31 header("Location:WBS1og.php");
32 }
33 ?>

```

If the row count doesn't equal to one, the user does not get authenticated. The unauthenticated user will get redirected back to the login page. If the user authentication succeeds, the user is redirected to another PHP page where the script will redirect the user based on the user's access privileges. In this page, the username which is stored as a session variable gets compared with the string "admin". If the username equals "admin", the user gets redirected to the webpage with administrator privileges. If not, the user gets redirected to the regular user pages, where the users have limited access.


```
<?php
session_start();

$myvalue=$_SESSION['myvar'];
if ($myvalue=="admin"){
header("Location: /addviewwbs.php");
}
else {
header("Location: /wbsuserview.php");
}
?>
```

With the administrator privilege, the user has the ability to enter tasks to the work breakdown structure, to view WBS of all users, and to add new users. To enter new tasks, the user must select an owner from the Owner dropdown list. The dropdown list is populated with unique owners retrieved from the database through an inline PHP script. Then the user must select the dates, enter the tasks and comments, and click save changes. At this instance, the values from the Enter Tasks form will get posted to a PHP page that will store the data into the database. The script for this activity is found in Appendix B.

WBS Manager		
Enter Tasks	View WBS	Logout
Owner Name	kyle	
Date	Task	Comments
mm/dd/yyyy		
mm/dd/yyyy		
mm/dd/yyyy		
mm/dd/yyyy		
mm/dd/yyyy		
<input type="button" value="Save changes"/> <input type="button" value="Cancel"/>		

The administrator also has the ability to add new users. In this page, the administrator enters new owners along with a user name and password. Upon clicking the "save changes" button, the form inputs will be posted to another PHP script where the data will be saved in the database. The admin can view the new user's credentials instantly through the User info table. Using this table, the administrator can share the new user's username and password. Once saved, the new user's owner names can be immediately seen in the owner dropdown lists.

User info table

Owner Name	User Name	Password
	admin	admin123
Eric	ecartman	ckman123
Kyle	kbroflowski	greenhat9
Megan	meguns	mg6432
Andrea	atombs	atom567
Test	Test	password123

Both the administrator and regular users have access to the View WBS webpage. However the regular users are limited to viewing only their own WBS. To view the WBS table, the user must select a start-date and an end-date along with the owner. As the administrator, the user will have access to view individual WBS tables of all users and well as WBS of all the users together. To view the WBS of all users, the admin user must select the "select all" item on the

Select Owner dropdown list. When this option is selected, the administrator can view the Date, Task, and Comments with the name of the owner on the side of the table. As a regular user, the user can view only the table with an owner name that matches the owner name retrieved from the database table row that corresponds to the username session variable. This allows regular users to view their WBS table only. The PHP script of this these actions is shared on the Appendix B.

WBS Manager Enter Tasks View WBS Add User Logout

Start Date
04/01/2013 : ▼

End Date
04/05/2013 : ▼

Select Owner Name
Eric ▼

Date	Task	Comments
Mon-1-April	Work Breakdown Structure (WBS) will dramatically increase your possibilities of delivering a successful project and it i	Sharing work load with Andrea
Tue-2-April	t techniques/tools for you as a Project Manager and the importance of creating a WBS should not and can not be u	
Wed-3-April	make you to sleep well at night. A well defined WBS will provide you, your project team and your stakeholders with a	Crucial part of the project

Date	Task	Comments	
Mon-1-April	Work Breakdown Structure (WBS) will dramatically increase your possibilities of delivering a successful project and it i	Sharing work load with Andrea	Eric
Mon-1-April	ConocoPhillips announced on Tuesday that it was suspending its plans to drill in Alaskan Arctic waters in 2014 because of uncertainties over federal		Kyle
Mon-1-April	When Chase Carey, Rupert Murdoch's top deputy at News Corporation, told broadcasters on		Megan
Mon-1-April	SNL Kagan estimates that station owners took in \$2.36 billion in retransmission fees	Shared with Eric	Andrea

Database Architecture

The database for the WBS management website consists of the following two tables.

1. wbsusers- (ID, Username, Password, Oname) this table stores the user data.
2. WBSV- (ID, Date, Task, Owner, Comments) this table stores data from the WBS

Detailed Schema is found in the Appendix C.

Testing and Evaluation

The Technology Acceptance Model (TAM) was used to evaluate the website. This model was used in order to predict the user's acceptance and usage of the website. Perceived ease of use is the extent to which the user believes using the system will be free of effort. Perceived usefulness is the extent to which the user believes that using the system will enhance their job performance (Davis Venkatesh, 2004). Perceived usefulness is measured by comparing what the software is actually doing with what the user thought it would do. To use this model, item analyses was performed by its creator to arrive at the 4-item scales. These were usefulness and ease of use items that were measured with a 7-point Likert-type scale having likely-unlikely endpoints along with the anchor points extremely, quite, slightly, and neither.

The four ease of use items were:

- Learning to operate WBSManager would be easy for me.
- I would find it easy to get WBSManager to do what I want it to do.
- It would be easy for me to become skillful at using WBSManager.
- I would find WBSManager easy to use.

The four usefulness items were:

- Using WBSManager would improve my performance in work scheduling.
- Using WBSManager in work scheduling would increase my productivity.
- Using WBSManager would enhance my effectiveness in work scheduling.
- I would find WBSManager useful in work scheduling.

The survey with the TAM-based questions was given to a single member of the sample project team who had been previously interviewed during the first study in this thesis. This

respondent is a project manager with 27 years of experience, who also specializes in end-user support. After scheduling a meeting time, the respondent was contacted through a web-hosted online meeting and desktop sharing service called GoToMeeting. The reason behind using this service over a phone interview was to use the desktop sharing feature. This feature allowed both the researcher and the subject to share their desktop screens while using the WBSManager website, thus improving communication. Along with the eight questions from the questionnaire, immediate follow up questions were asked in order to gain insight to the reasons behind the ratings and possible future improvements to the website. This interview was recorded and later transcribed for better review.

The ratings given by the respondent and the average ease of use and perceived usefulness ratings are given in the two tables below.

Table 1.1 Perceived ease of use

Learning to operate WBSManager would be easy for me	7
I would find it easy to get WBSManager to do what I want it to do	7
It would be easy for me to become skillful at using WBSManager	7
I would find WBSManager easy to use	7
Average ease of use	7

Table 1.2 Perceived usefulness

Using WBSManager would improve my performance in work scheduling	3
Using WBSManager in work scheduling would increase my productivity	2
Using WBSManager would enhance my effectiveness in work scheduling	2
would find WBSManager useful in work scheduling	3
Average usefulness	2.5

As indicated by table 1.1, the average perceived ease of use rating is 7 out of 7 which stands for "extremely likely." The average perceived usefulness rating indicated by table 1.2, is 2.5 out of 7. This is between "quite unlikely" and "slightly unlikely." When asked for the reasoning behind the ease of use ratings, the respondent indicated that the design of the website was intuitive with clear controls and navigation. The reasons for the respondent's perceived usefulness rating was that the current "excel work sheets provided more details in regards to the work breakdown." It was also mentioned that apart from the inputs of owner, date, task, and comments, the current excel work sheet included further details, and duration of tasks with the ability to "graphically show you when and what can be done." It was further pointed out that the current WBS also allowed the user to view the relationships between different work items.

Davis theorized perceived usefulness to be a more important determinant of intention than perceived ease of use. This is also due to the fact that, as the user gains hands-on experience, the effect of perceived ease of use on intension diminishes (Davis Venkatesh, 2004). This makes perceived usefulness the major determinant of intention. Based on the results from the survey, it can be hypothesized that the likelihood of this website being used is low.

Suggestions for Improvement

The WBS system described in this thesis supports only the basic WBS management functions and activities. Further developments are possible for this website. Based on the feedback from the website evaluation survey respondent, the inclusion of more input options such as the duration of task to the WBSManager website is recommended. It is also highly desirable to incorporate charts and graphs, thereby allowing the user to gain context on specific work items. To increase usefulness, another recommendation would be to define work items better along with the display of relationships between work items. The first step in increasing the

information depth of this website should be to allow the inclusion of more than one work item per day. This would allow the website to display the relationship between the different work items better. Since the website currently allows one task per day, the individual work items are not stored separately. Finally, it is recommended that the system support saving the individual work items separately inside the database to allow the users to specify which category an item belongs within, while also including the relationships that item has with other work items. While the WBSManager website provides the basic support for WBS management, there is plenty of room for improvement in the area of usefulness.

CONCLUSION

OnTime Scrum is a project management tool developed to help project teams follow the Agile Scrum methodology. The central feature of OnTime is the Visual Planning Board. This feature displays all of the tasks and work items that are needed to be complete. Findings from this research suggest that the project teams interviewed in this exploratory study does not utilize the OnTime scrum software to its fullest extent. "We are not fully using all the features of OnTime for project management" said one user.

The OnTime tool was used instead as a group work support system to keep track of bugs and customer requests. The supporting functions which were used by the project team were the helpdesk, bug reporting, bug tracking, time tracking, activity log, and wish list

In examining user satisfaction in regards to the features that were actually used, the overall level of satisfaction was high. The user satisfaction was high specifically for the quality of information, and security and dependability criteria. Ease of use and usefulness yielded varying results. The findings indicated that OnTime Scrum lacked intuitive features. The users were required to spend a significant amount of time to get acquainted with the various functions of OnTime they were supposed to use.

Based on the finding of this study, it has been suggested that the Work Breakdown Structure can be used to support agile development. The Work Breakdown Structure is designed to break projects down into smaller pieces. The sample project team utilized the Work Breakdown Structure for their project management purposes as an alternative to the features provided by OnTime Scrum. The teams used the work break down structure as a replacement to both the sprint backlog and product backlog components in the Scrum methodology. As reported

earlier in this study, the product backlog comprises of all the items to be completed in the project as a whole, while the sprint backlog specifies only the items related to that specific sprint. Using the WBS as a backlog item-tracking framework ensures that the intent of the customer drives the scope of work, which in turn drives the requirements and subsequent actions taken by the project.

Limitations

This study is not immune to limitations. The first limitation of this thesis was the small sample size. Only five members of a single project team were interviewed in order to obtain information needed for the creation of a new utility. While the members of the sample project team have many years of work experience, the small sample size limited the amount of data that was collected. Due to this reason, less weight was given to the quantitative data. While the survey results gave a general overview of the sample project team's satisfaction in regards to OnTime, the results may not be equally representative of user satisfaction on other agile development tools in the broader demographic.

The questionnaire for this thesis were created with the Assumption that users of OnTime used the tool to its full extent. Due to this reason, a significant portion of the questionnaire concerned project management. This also limited the amount of information gathered on features of OnTime that were not related project management. The questionnaire being tailored to OnTime Scrum limits the audience of this survey. Due to this reason, the findings of this research may not apply to other agile project management tools.

The findings of this research are also limited to the Scrum methodology. The results may vary among different agile methods. Finally, the information was gathered from a project team that worked in the health care industry. Therefore the results from this research may not apply to

project teams working in other industries. Due to the sample project team being in a regulated industry, their work environment may differ significantly from that of a typical project team.

Future Research

For future research, the scope of the research can be expanded to include other agile tools. Furthermore the scope of the research can explore beyond the Scrum methodology. Since this study was confined to a project team in the healthcare Industry, it is suggested that information be gathered from project teams in other industries. Finally, the possibility of using WBS as a transitioning agent into the Scrum methodology should be explored. Also, future studies may need to explore the use of WBS as project management tool in other methodologies as well. Finally future researchers can build upon the current WBSManager architecture developed in this thesis and incorporate functions that go beyond the Scrum methodology and even beyond the standard Agile methods.

REFERENCES

- Andersen, J. H. (2009). *Project Management in Agile Software Development: An empirical investigation of the use of Scrum in mature teams* (Doctoral dissertation, Norwegian University of Science and Technology).
- Ajzen, I., & Fishbein, M. (1977). Attitude-behavior relations: A theoretical analysis and review of empirical research. *Psychological bulletin*, 84(5), 888.
- Awad M.A (2005). A Comparison between Agile and Traditional Software Development Methodologies, School of Computer Science and software Engineering, The University of Western Australia,
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M & Thomas, D. (2001). Manifesto for agile software development.
- Behrenbruch, K., Atzmüller, M., Evers, C., Schmidt, L., Stumme, G., & Geihs, K. (2012). A personality based design approach using subgroup discovery. In *Human-Centered Software Engineering* (pp. 259-266). Springer Berlin Heidelberg.
- Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64-69.
- Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Boston, MA, USA: Addison-Wesley Professional.
- Davis, F. D., & Venkatesh, V. (2004). Toward preprototype user acceptance testing of new information systems: implications for software project management. *Engineering Management, IEEE Transactions on*, 51(1), 31-46.
- Deemer P, Benefield G (2007). The Scrum Primer: An Introduction to Agile Project Management with Scrum, <http://www.rallydev.com/documents/scrumprimer.pdf>
- Highsmith, J. (2002). *Agile Software Development Ecosystems*. Boston, MA, USA: Addison Wesley.
- Jensen, H. M., Sørensen, M. R., & Holm, M. (2008). Agile knowledge management with Scrum. MIL-STD-881 (1968). Work Breakdown Structures for Defense Materiel Items
- Moe, N. B., & Aurum, A. (2008). *Understanding Decision-Making in Agile Software Development: a Case-study*. Paper presented at the 34th Euromicro Conference Software Engineering and Advanced Applications, SEAA '08.
- Moe, N. B., Dingsoyr, T., & Dyba, T. (2009). Overcoming barriers to self-management in software teams. *Software, IEEE*, 26(6), 20-26.

- Pries, K. H., & Quigley, J. M. (2010). *Scrum project management*. CRC Press/ Llc.
- Rising, L., & Janoff, N. S. (2000). The Scrum Software Development Process for Small Teams. *IEEE Software*, 17(4), 26-32.
- Schwaber, K., & Beedle, M. (2001). *Agile Software Development with Scrum*. Upper Saddle River, NJ, USA: Prentice Hall PTR.
- Serignese, Katie (March 19, 2010). "Scrum Planning Board adds automation to Axosoft OnTime". Software Development Times.
- Sutherland Jeff, Schwaber Ken (2007). The Scrum Papers: Nuts, Bolts and Origins of an Agile Process, <http://www.crisp.se/scrum/books/ScrumPapers20070424.pdf>
- Tata, J., & Prasad, S. (2004). Team Self-management, Organizational Structure, and Judgements of Team Effectiveness. *Journal of Managerial Issues*, 16(2), 248-265.
- Vogel, Peter (April 1, 2010). "Managing Projects with Axosoft OnTime". Visual Studio Magazine.

APPENDICES

Appendix A. Interview Script

1. OnTime Scrum adequately meets my information needs.
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

2. Interaction with OnTime Scrum allows me to receive tailored information about my project
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

3. I trust OnTime Scrum to keep my project information safe and secure.
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

4. I find OnTime Scrum easy to use.
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

5. OnTime Scrum display pages of are easy to interpret.
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

6. OnTime Scrum is visually appealing.
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

7. All my project-management needs can be completed via OnTime Scrum
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

8. I am satisfied with the depth of information available on the OnTime Scrum dashboard
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

9. I am satisfied with the level of automation of OnTime Scrum
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

10. I am satisfied with the release flexibility of OnTime Scrum
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

11. I am satisfied with OnTime Scrum's ability to recover
 - Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree

12. I would recommend OnTime Scrum to a colleague interested in project management
- Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree
13. OnTime Scrum does exactly what I want it to do.
- Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree
14. OnTime Scrum is dependable
- Strongly agree
 - Agree
 - Indifferent
 - Disagree
 - Strongly disagree
15. Have you encountered errors in OnTime Scrum where the dashboard displayed information you knew to be false?
16. How often do you use pen and paper in project management?
17. If you do use pen and paper, when do you use it? (Is it to figure out alternative plans?)
18. How quickly can you spot changes or progress in the visual planning board? Does the information make sense as quickly as it would if it were a scoreboard of a game?
19. Which tasks in OnTime Scrum take the longest amount of time?
20. What are the features in OnTime Scrum that you use the least?
21. What types of improvements in the said features would make you use them more often?
22. What new features would you like to see in OnTime Scrum? If you could have only one new feature to support agile development, what would it be?
23. Are there features in OnTime that you always use concurrently?
24. Which features in OnTime scrum do you use the most?

25. Which characteristic of these features that you find most attractive?
26. Which feature in OnTime Scrum would you consider as ground-breaking?
27. If you had to give up a feature from OnTime Scrum, which feature would it be?
28. Are you able to display things that are not done in a Sprint (such as testing done by separate teams) using OnTime Scrum?

Appendix B. WBSmanger code

```
<?php
session_start();
$Owner=$_POST["Oname"];
$StartDate1=$_POST["sday1"];
$TaskDes1=$_POST["td1"];
$Comments1=$_POST["cmt1"];
$StartDate2=$_POST["sday2"];
$TaskDes2=$_POST["td2"];
$Comments2=$_POST["cmt2"];
$StartDate3=$_POST["sday3"];
$TaskDes3=$_POST["td3"];
$Comments3=$_POST["cmt3"];
$StartDate4=$_POST["sday4"];
$TaskDes4=$_POST["td4"];
$Comments4=$_POST["cmt4"];
$StartDate5=$_POST["sday5"];
$TaskDes5=$_POST["td5"];
$Comments5=$_POST["cmt5"];

// connect to the "tests" database
$conn = new mysqli('nirmaldata.db.10377872.hostedresource.com', 'nirmaldata', 'Lamrin!5',
'nirmaldata');
// check connection
if (mysqli_connect_errno()) {
    exit('Connect failed: '. mysqli_connect_error());
}
mysqli_query($conn,"INSERT INTO `WBSV` (`Date`, `Task`, `Owner`, `Comments`)
VALUES ('$StartDate1','$TaskDes1','$Owner','$Comments1')");
mysqli_query($conn,"INSERT INTO `WBSV` (`Date`, `Task`, `Owner`, `Comments`)
VALUES ('$StartDate2','$TaskDes2','$Owner','$Comments2')");
mysqli_query($conn,"INSERT INTO `WBSV` (`Date`, `Task`, `Owner`, `Comments`)
VALUES ('$StartDate3','$TaskDes3','$Owner','$Comments3')");
mysqli_query($conn,"INSERT INTO `WBSV` (`Date`, `Task`, `Owner`, `Comments`)
VALUES ('$StartDate4','$TaskDes4','$Owner','$Comments4')");
mysqli_query($conn,"INSERT INTO `WBSV` (`Date`, `Task`, `Owner`, `Comments`)
VALUES ('$StartDate5','$TaskDes5','$Owner','$Comments5')");
$conn->close();

header("Location: /addviewwbs.php");

?>
```

```
<?php
session_start();

$myvalue=$_SESSION['myvar'];
if($myvalue=="admin"){
header("Location: http://ndharma.com/addviewwbs.php");

}
else {
header("Location: http://ndharma.com/wbsuserview.php");
}

?>
```

```
<?php
$Owner=$_POST["oname"];
$Usern=$_POST["uname"];
$Pword=$_POST["pword"];

$conn = new mysqli('nirmaldata.db.10377872.hostedresource.com', 'nirmaldata', 'Lamrin!5',
'nirmaldata');

if (mysqli_connect_errno()) {
    exit('Connect failed: '. mysqli_connect_error());
}
mysqli_query($conn,"INSERT INTO `WBSV` ( `Owner`)
VALUES ('$Owner')");
mysqli_query($conn,"INSERT INTO `wbsusers` (`Oname`,`Password`,`Username`)
VALUES ('$Owner','$Pword','$Usern')");

$conn->close();
header("Location: http://ndharma.com/addnewuserwbs.php");
?>
```

```
<?php
session_start();
$host="nirmaldata.db.10377872.hostedresource.com"; // Host name
$username="nirmaldata"; // Mysql username
$password="Lamrin!5"; // Mysql password
```

```

$db_name="nirmaldata"; // Database name
$table_name="wbsusers"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// username and password sent from form
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];
$_SESSION['myvar']=$myusername;
// To protect MySQL injection (more detail about MySQL injection)
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);
$mypassword = mysql_real_escape_string($mypassword);

$sql="SELECT * FROM $table_name WHERE Username='$myusername' and
Password='$mypassword'";
$result=mysql_query($sql);

// Mysql_num_row is counting table row
$count=mysql_num_rows($result);

// If result matched $myusername and $mypassword, table row must be 1 row

if($count==1){

// Register $myusername, $mypassword and redirect to file "login_success.php"

header("location:login_success.php");
}
else {
header("location:WBSLog.php");
}

?>


---


<!DOCTYPE html>
<html >

```

```

<head>
  <meta charset="utf-8">
  <title>Work Breakdown Structure</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="">
  <meta name="author" content="">

  <link href="css/bootstrap.css" rel="stylesheet">
  <script src="js/jquery.js"></script>
  <script src="js/bootstrap.js"></script>
  <script>
    $(function(){

      $('.dropdown-toggle').dropdown()

    });
  </script>
  <!--[if lt IE 9]>
    <script src="../assets/js/html5shiv.js"></script>
  <![endif]-->

```

```
</head>
```

```
<body>
```

```

<div class="navbar navbar-fixed-top navbar-inner" style="background-image: -webkit-linear-
gradient(top, rgb(255, 127, 0), rgb(255, 127, 0));">
<div class="container" style="width: auto;">

```

```
</a>
```

```
<a class="brand" href="#" style="color: rgb(2, 0, 12); font-size: 25px;">WBS Manager</a>
```

```
<ul class="nav">
```

```
  <li><a href="addviewwbs.php" style="color: rgb(15, 0, 0); font-size: 19px;"> Enter
Tasks</a></li>
```

```
  <li class="divider-vertical" style="border-right-width: 1px; border-right-style: solid;
border-right-color: rgb(237, 18, 18); background-color: rgb(237, 18, 18);"></li>
```

```
  <li><a href="viewtable.php" style="color: rgb(10, 0, 0); font-size: 19px;">View
WBS</a></li>
```

```

    <li class="divider-vertical" style="border-right-width: 1px; border-right-style: solid;
border-right-color: rgb(237, 18, 18); background-color: rgb(237, 18, 18);"></li>
    <li><a href="addnewuserwbs.php" style="color: rgb(10, 0, 0); font-size: 19px;">Add
User</a></li>
    <li class="divider-vertical" style="border-right-width: 1px; border-right-style: solid;
border-right-color: rgb(237, 18, 18); background-color: rgb(237, 18, 18);"></li>
    <li><a href="WBSLog.php" style="color: rgb(10, 0, 0); font-size:
19px;">Logout</a></li>
    <li class="divider-vertical" style="border-right-width: 1px; border-right-style: solid;
border-right-color: rgb(237, 18, 18); background-color: rgb(237, 18, 18);"></li>

</ul>

</div>
</div>

```

```

<form method="Post" action="newuserenter.php">
<div class="hero-unit" >
    <div><div > Owner Name</div>
    <div ><input type="text" name="oname" size="15"> </div>        </div>
    <div><div > User Name</div>
    <div ><input type="text" name="uname" size="15"> </div>        </div>
    <div><div > Password</div>
    <div ><input type="text" name="pword" size="15"> </div>        </div>

    <button type="submit" class="btn btn-primary" style="background-image: -webkit-
linear-gradient(top, rgb(0, 136, 204), rgb(0, 68, 204)); color: rgb(255, 255, 255); background-
color: rgb(0, 68, 204);">Save changes</button>
    <button type="Reset" class="btn" style="background-image: -webkit-linear-gradient(top,
rgb(255, 255, 255), rgb(230, 230, 230)); color: rgb(51, 51, 51); background-color: rgb(230, 230,
230);">Cancel</button>
</div>
</form>
<div>
<h1> User info table </h1>
</br></br></br>
<table border="1"><thead><tr >
<th data-column="0" ><div ><p> Owner Name </p></div></th>
<th data-column="1" ><div ><p> User Name </p></div></th>
<th data-column="1" ><div ><p> Password </p></div></th>

```

```

</tr></thead>
<tbody >
<?php
$conn = mysql_connect('nirmaldata.db.10377872.hostedresource.com', 'nirmaldata', 'Lamrin!5',
'nirmaldata');
if (!$conn)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("nirmaldata", $conn);

$result = mysql_query("SELECT `Oname`, `Username`, `Password` FROM `wbsusers`");
while($row = mysql_fetch_array($result))
{
echo "<tr>";
echo "<td >";
echo $row['Oname'];
echo "</td>";
echo "<td >";
echo $row['Username'];
echo "</td>";
echo "<td >";
echo $row['Password'];
echo "</td>";
echo "</tr> ";
}
mysql_close($conn);
?>
</tbody></table>
</div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js">
</script>
<script src="js/bootstrap.js">
</script>
</body>

</html>



```

Appendix C. WBSmanger database schema

Field	Type	Collation	Attributes	Null	Default	Extra
<u>ID</u>	int(11)			No		auto_increment
Date	date			No		
Task	text	utf8_general_ci		No		
Owner	char(16)	utf8_general_ci		No		
Comments	text	utf8_general_ci		No		

WBSV table

Indexes: ②



Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	50	 	ID

WBSV Primary Key

Field	Type	Collation	Attributes	Null	Default	Extra
<u>ID</u>	int(16)			No		auto_increment
Username	varchar(16)	utf8_general_ci		No		
Password	varchar(16)	utf8_general_ci		No		
Oname	varchar(16)	utf8_general_ci		No		

wbsusers table

Indexes: ②

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	6	 	ID

wbsusers Primary Key

Appendix D. WBSmanger Evaluation Questionnaire

1. Learning to operate WBSManager would be easy for me -

1-extremely unlikely 2-quite unlikely 3-slightly unlikely 4-neither
5- slightly likely 6- quite likely 7-extremely likely

2. I would find it easy to get WBSManager to do what I want it to do-

1-extremely unlikely 2-quite unlikely 3-slightly unlikely 4-neither
5- slightly likely 6- quite likely 7-extremely likely

3. It would be easy for me to become skillful at using WBSManager-

1-extremely unlikely 2-quite unlikely 3-slightly unlikely 4-neither
5- slightly likely 6- quite likely 7-extremely likely

4. I would find WBSManager easy to use -

1-extremely unlikely 2-quite unlikely 3-slightly unlikely 4-neither
5- slightly likely 6- quite likely 7-extremely likely

1. Using WBSManager would improve my performance in work scheduling-

1-extremely unlikely 2-quite unlikely 3-slightly unlikely 4-neither
5- slightly likely 6- quite likely 7-extremely likely

2. "Using WBSManager in work scheduling would increase my productivity-

1-extremely unlikely 2-quite unlikely 3-slightly unlikely 4-neither
5- slightly likely 6- quite likely 7-extremely likely

3. "Using WBSManager would enhance my effectiveness in work scheduling-

1-extremely unlikely 2-quite unlikely 3-slightly unlikely 4-neither
5- slightly likely 6- quite likely 7-extremely likely

4. "I would find WBSManager useful in work scheduling-

1-extremely unlikely 2-quite unlikely 3-slightly unlikely 4-neither
5- slightly likely 6- quite likely 7-extremely likely

Appendix E. Oral Consent Script

Hello, my name is Nirmal Dharmaratne and I am an undergraduate student at the University of Mississippi. With the help of Dr. Brian Reithel, a professor of MIS here at Ole Miss, I am conducting a survey research study to come up with a new utility for OnTime Scrum. If you agree to participate, I will ask you questions about your level of satisfaction with OnTime Scrum, how you interact with it, and on what you think will improve your experience with OnTime Scrum. This will take about 15-30 minutes of your time. The study involves filling out a questionnaire and a few open ended questions. To ensure confidentiality, no information will be collected that can link you back to your interview. Do you have any questions about me, my research, or our interview before we begin?