HAAGA-HELIA
University of Applied Sciences

Windows Phone development using Scrum templates for Team Foundation Server

Pete Stockley

Bachelor's Thesis
DP in Business IT
2012 – 9

HAAGA-HELIA
University of Applied Sciences

Abstract

Date of presentation

Degree programme

Author or authors	**Group or year of entry**
Pete Stockley	2009
Title of thesis	**Number of pages and appendices**
Windows Phone development using Scrum templates for Team Foundation Server	34 + 26
Thesis advisor(s)	
Juhani Välimäki	

This thesis aims to assess the suitability of using the scrum framework within a Microsoft Team Foundation Server application development environment when developing a Windows Phone 7 application.

The goal of the thesis is to assess whether Team Foundation Server is a suitable tool for learning scrum and application development at HAAGA-HELIA University of Applied Sciences.

The thesis was conducted by researching the topics related to the thesis goal and then creating a test Windows Phone 7 application in the Team Foundation Server development environment.

The results of the the test application were then evaluated to asses the development environment's suitability.

This thesis also contains appendices of the step-by-step processes used to set up the development environment.

Keywords

Team Foundation Server, Scrum, Windows Phone 7, Application development

Table of contents

Vocabulary

Agile software development – A group of software development methods that promote adaptation, flexibility and continuous development of the application within time boxes.

Codeville – A revision control system for application development teams to assist with merging code.

Git – A revision control and source code management system.

Java – An object orientated programming language derived from C and C++.

Javascript – A primarily web-based programming language.

Mercurial – A revision control system .

Scrum – A software development method developed from agile software development.

Sharepoint – A web application development platform developed by Microsoft.

Silverlight – A framework for writing rich internet applications developed by Microsoft.

Visual Studio – An integreated development environment developed by Microsoft.

Windows Forms – The graphical application programming interface, part of Microsoft's .NET framework.

Abbreviation

ASP – Active Server Pages. A server-side script engine for creating dynamic web pages, developed by Microsoft.

BIT – Business Information Technology. In this context, the name of a Bachelor degree program at Haaga-Helia University of Applied Sciences.

CSS – Cascading Style Sheet. A programming language for defineing a web page's style.

C# – C sharp. A Microsoft software development language.

DNS – Domain Name System. A system for translating domain names into IP addresses.

DVD – Digital Versatile Disc. An optical media storage.

ELMS – Electronic License Management System. A license management system for providing Microsoft software to educational facilities that teach Microsoft application development.

FQDN – Fully Qualified Domain Name. A name that specifies a server's location within a domain hierarchy.

GPO – Group Policy Object. A feature of Microsoft operating systems that allows control over user priviledged.

HDD – Hard Disk Drive. A digital storage media.

HTML – HyperText Markup Language. A standard markup language used to define how web browser visualize or utilize web sites.

IAP – In-App Purchase. The purchase of additional content made within an application.

IIS – Internet Information Services. A Microsoft web server application.

IP – Internet Protocol. A network protocol for realying data packets around the internet.

MySQL – My Structured Query Language. An open source relatinoal database management system.

OOP – Object Orientated Programming. A language that uses objects, such as classes, which can be independent from each other and are capable of interacting with each other when necessary.

PBI – Product Backlog Item. An object in the scrum framework for defining a work item to be completed.

PDA – Personal Digital Assistant. An obsolete form of mobie device.

PHP – Hypertext Pre Processor. A server-side script language used for creating dynamic web pages.

SBI – Sprint Backlog Item. A Product Backlog Item that has been taken into a scrum sprint.

SDK – Software Development Kit. A set of development tools.

SQL – Structured Query Language. A query language for handling data in relation database management systems.

TFS – Team Foundation Server. A Microsoft software development environment.

UI – User Interface. The interface where a user interacts with a machine.

VB – Visual Basic. An event-driven programming language from Microsoft.

WPF – Windows Presentation Foundation. A graphical subsystem for rendering user interfaces.

WP7 – Windows Phone 7. The operating system for Windows Phones. This also includes version 7.5 also known as Mango.

WSS – Windows Sharepoint Services. A platform for allowing communication between web applications and other Microsoft platforms such as TFS.

XAML – Extensible Application Markup Language. An XML-based user interface language developed by Microsoft.

XML – Extensible Markup Language. A markup language that defines a set of document rules.

XNA – A set of tools for video game development and management developed by Microsoft.

.NET – A framework consisting of a large ibrary of classes and a software environment in which most Microsoft software applications run in.

1 Introduction

This thesis topic was born from a variety of needs that existed at Haaga-Helia University of Applied Sciences that became apparent during the author's practical training period at the beginning of his third years of studies.

The practical training took place at Haaga-Helia University of Applied Sciences and it was identified by the lecturer responsible for the author's training that students on the BIT degree programme could use tools to better improve their application development processes, with particular reference to the development processes used in scrum framework, which is also taught to BIT students.

Due to the development environment predominantly used by the students of the BIT degree programme being Microsoft's Visual Studio, it was decided that Microsoft's TFS development environment should be the tool to be investigated.

As mentioned, the scrum framework is taught to BIT students at Haaga-Helia University of Applied Sciences. The author and the author's lecturer both identified that the application of learning the scrum framework could be refined from its current state. In the author's experience the learning process was made poorer because students had to focus on overcoming the learning curve of all the reporting required in scrum as well as the developing their application.

TFS is a tool that has the capability to provide a certain amount of automation to the reporting processes involved in scrum (scrum will be discussed in more details in chapter 2.2). This includes the daily updates to the work items by all the team members as well as the documentation from all the meetings. For this reason, it was decided that TFS' ability to support the scrum framework should be investigated.

1.1 The purpose

The purpose of this thesis is to evaluate the suitability of using TFS as an environment for developing a WP7 application.

It was the author's opinion that in order to best test TFS support of the scrum framework, it would be effective to develop a test application in order to simulate a more realistic development process. One of the areas that BIT students at Haaga-Helia University of Applied Sciences develop applications in, is WP7. As the author was familiar with, and was in the process of updating a WP7 application, it was deemed suitable to use a WP7 application for this test.

1.2 The author

The author is a Haaga-Helia University of Applied Sciences student who has completed WP7 development courses, written an intensive course on WP7 application development, published several WP7 applications, has some initial familiarity with the TFS environment and has completed several university course projects using the scrum framework as a scrum master and team member it can be argued that the author is a suitable candidate for making this assessment.

1.3 The scope

The scope of the thesis has been narrowed down quite considerably in order to make it acheivable. As mentioned in the previous chapter, the three key areas of this thesis are TFS, scrum and WP7.

1.3.1 TFS

There are several tools for managing version control and project management such as Git, Codeville, Mercurial and many others but TFS was selected as the tool for this thesis for a variety of reasons. The primary reason for this was the accessibililty to the TFS software which was available for student use for free as long as it is used for educational purposes, specifically how to use TFS. In addition to this TFS is specifically designed for development of Microsoft software and for being used with Visual Studio which is the primarily development tool at Haaga-Helia University of Applied Sciences.

1.3.2 Scrum

Scrum is one of many forms of agile software development but of these, scrum is specifically taught at Haaga-Helia University of Applied Sciences. TFS also supports the scrum framework with an additional template available from Microsoft. For these reasons, scrum was the agile development method chosen for this thesis.

1.3.3 Windows Phone 7

WP7 is one of the many Microsoft software environments taught on the BIT degree programme at Haaga-Helia University of Applied Sciences (WP7 will be discussed in more details in chapter 2.5) including ASP and Windows Forms. Although any of the software environments taught at Haaga-Helia University of Applied Sciences would be suitable for the test application, it was decided that a WP7 application would be created because the author could focus more on the TFS development environment and less on learning new skills that may have been required to create a application.

1.4 Research Plan and chosen methods

The research plan applied to this thesis follows that of a constructive research thesis. Firstly an understanding of the key concepts relating to the thesis question is gained by studying relevant literature and other sourced materials, which are discussed in chapter 2.

Secondly, the literal understanding of these concepts is put into practice by completing a test application. The processes of this research method will be discussed in further details in chapter 3 and the details of the test application are provided in chapter 4.

2 Theory

In examining this thesis topic, a number of theory areas were identified and like any research-based thesis, a number of literature sources were required to be studied in order to completely understand the thesis topic.

The literature sourced can be found in the bibliography of this thesis and browsing through it will reveal a common theme. The majority of these sources are based on a hands-on approach to solving a problem.

The background theory for this thesis has been identified as falling under three key areas: Application development which is the key area of this thesis as it is the area where scrum and TFS meet and in fact scrum and TFS are both products to facilitate software development; Scrum which is a form of agile development, not just for software but can applied to a wide range of product developments; and TFS which is Microsoft's own development environment, designed to enable fast team development on software that uses Microsoft's C# language but can also accommodate several other software development languages such as HTML and Java.

2.1 Team work

Team work, or the understanding of how members of a team contribute towards a common goal is an area of study all to itself, sometimes under the title of team dynamics. This is far too wide a scope for this thesis and so it has been narrowed down to look at how team work relates to software application development.

Team work in software development usually revolves around a formal structure which defines the time periods and deadlines, member roles and distribution of tasks. There are many frameworks for team application development including process-centric models such as the waterfall model and the v model and more recently, extreme programming and agile development. Even within agile development there are many frameworks of which scrum is only one. Agile development frameworks are less linear and more adaptive to change. This puts a precedence on the people working in the

team and allows the development process to change with what works best for the team. It is the scrum framework which will be discussed in the next chapter.

2.2 Scrum

In the Scrum Guide, scrum is defined as a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value (Scrum.org 2010, 3).

Scrum is not a strict set of rules or processes to follow but rather a guide to enabling good collaboration, fast and easy productivity, and constant feedback to allow for fast obstacle identification and removal. In many of the author's experiences with working within a scrum framework, it is not essential to adopt every aspect of the framework but select those parts that work best for the company and project team.

There are three areas of scrum that allow it to shape a successful project: Transparency, which allows the entire group to follow and understand the progress of the project; Inspection, which allows identification of obstacles and reporting of progress; and adaptation which helps the project change and adapt to the evolving requirements of the project product and goals (Scrum.org 2010, 4).

A key area of scrum is the structure of events. Each project is split into equal length time periods called sprints that usually last a month. Each sprint should have two planning meetings, a goal, daily meetings, a review and retrospective.

2.2.1 Sprint Planning meeting one

The first planning meeting aims to answer what will be done during this sprint. The team assesses the product backlog and prioritizes which should be taken into the sprint backlog. After this, the team defines the sprint goal to help guide themselves on what the sprint is aiming to achieve (Scrum.org 2010, 9).

2.2.2 Sprint Planning meeting two

Planning meeting two answers how the work will be completed. The definition of done is created and the items from the sprint backlog are broken down into manageable tasks, usually with a maximum time limit of one working day.

By the end of the second meeting, the team should have a clear understanding of how they will accomplish the sprint goal which they can provide to the product owner (Scrum.org 2010, 9-10).

2.2.3 Sprint Goal

The sprint goal is the final outcome to be reached by the end of the sprint. Even if the team modifies the task list or need to work around unexpected obstacles, they can still aim to achieve the goal (Scrum.org 2010, 10).

2.2.4 Daily meetings

The daily meetings also known as the daily scrum is a strongly defined fifteen minute meeting designed to allow the team to maintain communication, keep organized and ensure they understand the current progress of the sprint.

During the meeting, team members and the scrum master can discuss what has been accomplished since the latest meeting, what will be done before the next meeting and what obstacles, if any, have to be removed (Scrum.org 2010, 10-11).

2.2.5 Sprint Review

The sprint review is held at the end of the sprint to examine how close to the sprint goal the team has got to. Here the product owner confirms what PBIs have been done and with the team, they discuss changes to the product backlog if necessary (Scrum.org 2010, 11).

2.2.6 Sprint Retrospective

The retrospective meeting is meant to be used to plan the internal aspects of the project with particular attention given to how well the team works together, possible improvements for the work processes and to create a plan for implementing these improvements (Scrum.org 2010, 12).

Another key area of a scrum project is the member roles. Each member in a scrum project should have a role that defines their tasks and responsibilities to the project and the other team members. There are three defined roles:

2.2.7 Product owner

The product owner the product's value and is responsible for maximizing the value of the product and ensuring the development team can be as efficient as possible. The key role of the product owner is manage the product backlog, a 'to do' list of items for the project. By defining and ordering the product backlog as well as ensuring an understanding of all the PBIs a product owner can help the project and the team to succeed (Scrum.org 2010, 5).

2.2.8 Development team

The development team are a team of professionals aiming to complete the project. The team is self-organizing and should be empowered to work at their maximum efficiency.

Each team member should be able to complete any task created by the team. Although team members often have different strengths and weaknesses, each member should be prepared to handle each task and be accountable for it.

The ideal size of a development team is said to be seven with minimum and maximum limits of five and nine respectively. A small team helps to provide clear coordination and easy management whilst still providing a good skill pool and reducing organization complexity (Scrum.org 2010, 6).

2.2.9 Scrum master

The scrum master's role is primarily to ensure that the scum framework is understood and adhered to, at least those areas of the framework that have been implemented. Secondarily the scrum master should help the development team administrate themselves and be the intermediary between the team and outside interactions and work towards removing any identified obstacles (Scrum.org 2010, 7).

2.3 Application development

The wide scope of application development in general is far too vast be researched to a level of completeness that would suffice any literary document, let alone this thesis. For this reason, the scope has been narrowed down considerably to provide an introduction into object-oriented programming and where examples are given, these relate specifically to the C# object-oriented programming language.

Application development is the process of turning a concept of an application into a final product, with the use of development tools and a programming language. Object-oriented programming languages use objects, such as classes, which can be independent from each other and are capable of interacting with each other when necessary. C# is one such language.

C# is an OOP language from Microsoft and is used in a wide range of their programming platforms including ASP, Windows Forms, WPF and WP7. C# is a very simple and multipurpose language that runs within the .NET framework.

Applications can be developed for a wide-range of hardware including desktop computers, mobile devices and web-based services. C# can be used to develop applications on all of these platforms.

2.4 Mobile application development

Much like its encompassing container, application development, mobile application development is a very large area of study. For this purpose, the scope has been nar-

rowed down to provide an introduction to explain the key differences in mobile application development as opposed to development for desktop or web-based applications and more specifically, the differences between these environments within Microsoft's .NET umbrella and even more specifically, those environments that accommodate C#.

Mobile application development is the development of applications for what is commonly known as mobile devices. This is somewhat a misnomer because most laptops, which are mobile, certainly more than desktop computers, are not considers as mobile devices.

Mobile devices range from small mp3 players up to tablet devices and contain devices such as PDAs, phones, smartphones, digital cameras, pagers and even calculators. Common to all of these is the presence of an operating system, a method of user input such as touchscreen or keyboard and may also provide communication technologies such as wifi or Bluetooth. They are often designed to be used "on the go" and can be said to allow the user more mobility.

Mobile application development differs from the concept of application development discussed in the previous chapter by taking into the consideration factors such as shorter battery life, minimized user input controls, smaller display areas and shorter usage periods.

2.4.1 Shorter battery life

Battery life is a major factor in mobile application development and programmers should be aware that their code uses as fewer operations as possible, or rather more efficient code, in order to preserve battery life and provide a better user experience.

2.4.2 Minimized user input controls

Minimized user input controls should be considered as there are less possibilities for the user to utilize. In many cases, there is an absence of a keyboard and the user must rely on gestures such as tapping, swiping, pinching, shaking and rotating. If any virtual

controls are displayed on the screen, they should be carefully considered whether they are easily used due to size and location.

2.4.3 Smaller display areas

Mobile devices have smaller displays than traditional computing devices so consideration should be taken when designing the user interface as mentioned previously and also when designing the layout of the display areas of the application.

2.4.4 Shorter usage periods

Mobile devices are often used for shorter periods of time than conventional computing devices and applications can be opened and put on standby several times before being fully closed. It is important to consider the storing and retrieval of data in order to accommodate such usage and provide a timely, secure service.

These factors must be all taken into consideration when developing aplications for mobile devices if the developer wants the application to be usable and therefore succesfull.

2.5 Windows Phone 7 development

In the previous chapter we looked at how mobile development such as WP7 differs from other environments such as the WPF desktop environment and the ASP web-based environment. This chapter aims to focus more on the areas that make up a WP7 development environment.

WP7 applications are built using a single language or a combination of Silverlight and XNA. Silverlight is primarily used for functional, non-game applications and XNA is primarily used for games although both languages could be used to create a final product. A third option exists by using a web application languages, HTML (and HTML5), CSS and JavaScript within a Silverlight web control (Nathan 2011, 3-4).

Regardless of which language is chosen, the application is written with a combination of xml derivative xaml and .NET code which is written in C# or VB. Xaml is the primary language used to define the UI objects although they can also be defined using C# or VB. C# or VB is the language used to write the business logic (Nathan 2011, 6). WP7 applications are created within Visual Studio, Microsoft's application development software, with the necessary WP7 SDK installed. Optionally, and highly recommended by Microsoft is the Silverlight toolkit. An open source developed set of additional tools for developing WP7 applications (Nathan 2011, 5).

WP7 applications are distributed primarily through Microsoft's distrubition platform, the Windows Phone Marketplace. This is an online store accessable via a mobile deveice running the WP7 operating system and through this, users can download applications on to their devices. Apps can be free or cost a fee determined by the application developer. Any updates to the applications are always free of charge but IAPs can be charged for.

2.6 TFS

Team Foundation Server is Microsoft's solution to team-based project management. It supports a variety of functionalities important to project management including: Work item tracking; Version control; Test case management; Build automation and; Reporting (Blankenship et al 2011, 4).

TFS is a two tier system: The client tier which allows access to the TFS via Visual Studio client and consists of a set of Sharepoint web services which allow users to locally interact with TFS, and the data tier which is an SQL server based tier containing all the data of the TFS instance and projects built within that instance (Blankenship et al 2011, 4).

One of the major benefits of using TFS over competing software development project applications is the support when building Microsoft-based applications (Blankenship et al 2011).

Projects in TFS are based on users, work items, iterations and areas which can be used according to templates relating to application development. These templates reflect application development frameworks such as scrum (Blankenship et al 2011, 50).

More detailed descriptions of these project elements can be found in appendix 7.

3 Emperical Research

As mentioned in the introduction, the key method of research is to answer the thesis question, which was a practical examination of how suitable the scrum framework can be applied to the development of WP7 application within TFS.

In order to accomplish this, a test application was designed and an environment was set up within which the practical examination could be run. This consisted of a location, various hardware and software elements, development skills and knowledge of the research areas applicable to completing the practical examination.

The test application that was designed for the practical test was an expansion to an existing WP7 application written by the author previously in the year called Letter Link. The expansion to the application did not require having to learn many more skills than was used in the original application so therefore time could be better focused on the development of the application rather than development of the author's skill set.

Letter Link is a word game where the player has one hundred levels to solve with each level offering a line in a story, prompting the player to solve all the levels. Each level consists of six clues where the answers have been split into a selection of buttons. In order to solve the clue, the player must tap the buttons containing the parts of the clue in the right order and submit their guess. If the guess is correct, the user is shown the answer and the answer buttons are removed from the available buttons.

As the player eliminates answer buttons by guessing clues correctly, it becomes easier to find the answers for the remaining clues.

The test application, or expansion to the current application would include the following: An additional one hundred levels; a level creator and editor; the ability to upload created levels to a server; the ability to search and download levels from a server; the ability to reset individual levels; and improved help and options.

3.1 Location of the test application

Due to the hardware requirements, it was necessary to create the environment in a Haaga-Helia University of Applied Sciences server lab. The server lab allows students to create and test domains within a safe environment by restricting incoming traffic to the environment that is not a response to an initiating request made from within the server lab environment.

Positive aspects of the location included the lab providing all the hardware, peripherals and networking systems required to create the environment as well as covering the costs to run the servers.

A negative aspect of the location was the restriction of incoming traffic from the internet as mentioned above. This restriction's effect on the project made it impossible to connect to the TFS system from outside of the server lab. This detracted from the realism of the testing environment by requiring all work to be done in the server lab, and also decreased the available time for working on the test application.

3.2 Hardware requirements of the test application

The hardware requirements were made available by Haaga-Helia University of Applied Sciences within their server lab as mentioned in the previous chapter.

The hardware requirements included four desktop PCs with network capabilities, each with its own mouse, keyboard and monitor as well as the necessary hardware components such as motherboard, processor, ram and hard drive.

Three PCs would be used as a domain containing a domain controller, a development server and a build server. The forth PC would be used as a client computer.

3.3 Software requirements of the test application

The software requirement were made available via the ELMS which provides windows software to students for development and test use. The ELMS provides this software for free to students at participating educational institutes that teach their students Mi-

crosoft-based computer and network administration and / or Microsoft-based software development.
For the purpose of making the software requirement easy to read, they are presented here in list form. It is important to note that this list of software is not the most current in terms of available versions but were supported by the TFS environment at the time of writing. It is very likely that the software requirements will evolve in tandem alongside newer developments in the TFS environment.

- Microsoft Server 2008 R2 SP1 64-bit
- Network Drivers
- Microsoft SQL Server 2008 R2
- Microsoft Windows SharePoint Services 3.0 SP1
- Microsoft Team Foundation Server 2010
- Microsoft Windows 7 Pro (x64 or x86)
- Microsoft Visual Studio 2010 Ultimate SP1
- Microsoft Visual Studio Scrum Template 1.0
- Microsoft Framework 3.5

3.4 Development skills required

The development skills required, or at least recommended by the author for this project include: An understanding of the foundations of .NET, C#, Silverlight, XAML and XML for the WP7 software and an understanding of PHP and MySQL for the web pages that communicate between the WP7 web requests and the server database.

The author acquired these skills from the three years of study prior to the thesis as well as conducting personal studies and research on the subjects and developing WP7 applications.

An understanding and experience of these development skills helped to better plan for the project in terms of data structures and solutions to problems that would occur in the test project.

3.5 Knowledge of research areas required

The areas of knowledge, as mentioned earlier in this document, include: An understanding of the fundamentals of server architecture; An understanding of the fundamentals of application development; An understanding of the fundamentals of mobile application development and more specifically WP7 application development and an understanding of the scum framework.

Like the required development skills, the author acquired these skills from the three years of study prior to the thesis as well as conducting personal studies and research on the subjects and developing WP7 applications.

An understanding and an experience of these areas were very helpful during the research process as they helped to make the test project flow faster and smoother with less obstacles and when obstacles did occur, they were overcome faster.

3.6 Limitations of the test application

In order to complete this assesment, there was a requirement for large quantities of hardware including servers and a domain. Haaga-Helia University of Applied Sciences was able to provide this hardware but came with some limitations. The domain was closed meaning the environment could initiate communication to the internet (allowing for web page browsing and file downloads for example) but communication could not be initiated with the domain from outside. This meant that the research had to be done inside the same lab as the servers in the Haaga-Helia University of Applied Sciences Pasila campus building. The result of this meant that access to the environment, and therefore the possibility to conduct the research, was limited in terms of time and detracted from the realism of the test application development.
Another limitation of the asessment was the lack of assessors. The author conducted this thesis alone and therefore the scrum project was conducted alone which is a very unlikely and arguably impossible scenario for a scrum project. This limitation will decrease the validity of the thesis assesment to a certain extent even thought it was conducted with the author completing different project tasks under different

usernames to simulate different team members, each with their own tasks, goals and access rights.

The process of accomplishing this thesis goal was the common theory and research approach wherby an understanding of the key thesis concepts was gained by reading literature based on these concepts. This was followed by practical research.

4 Test application

The research process began on the sixth of June and continued for eight weeks at Haaga-Helia University of Applied Sciences server lab 5005. The first two weeks went towards setting up the environment and setting up the scrum framework in TFS.

Following this, the next six weeks were three two-week sprints for completing the test project. At the end of these six weeks, an additional week was used for final testing by persons outside of the development team.

Some areas of application development, such as testing and building were constant throughout the test application development process and are discussed in the following two chapters.

Following are the details of the stages of the development of the test application as mentioned above. In each step, the process is only briefly mentiond. That is to say what was done rather than how. A detailed step-by-step process of these stages have been listed in appendices 1 through 7.

4.1 Application testing

Throughout the duration on the development of the test application, testing was continuously done at each stage, performed primarily by the author.

After the final sprint, final testing was performed by several people who were not involved in the development process so that the view was unbiased and from the point of view similar to that of the final users, WP7 handset users who have no prior understanding of the game's engine. The testing of the application at the final stages of development were performed by Virpi Vidal Pellikka, Teemu Tapanila and Hal Clive.

Testing was performed externally from the TFS environment on the testers' own devices. Each tester was comfortable in how to install a development stage application on their handsets and were given instructions on what areas to test. Virpi tested the addi-

tional 100 levels for game while Teemu and Hal tested the functionality of uploading and downloading user-made levels. Feedback gathered from the testing would be used to further improve the test application

4.2 Application builds

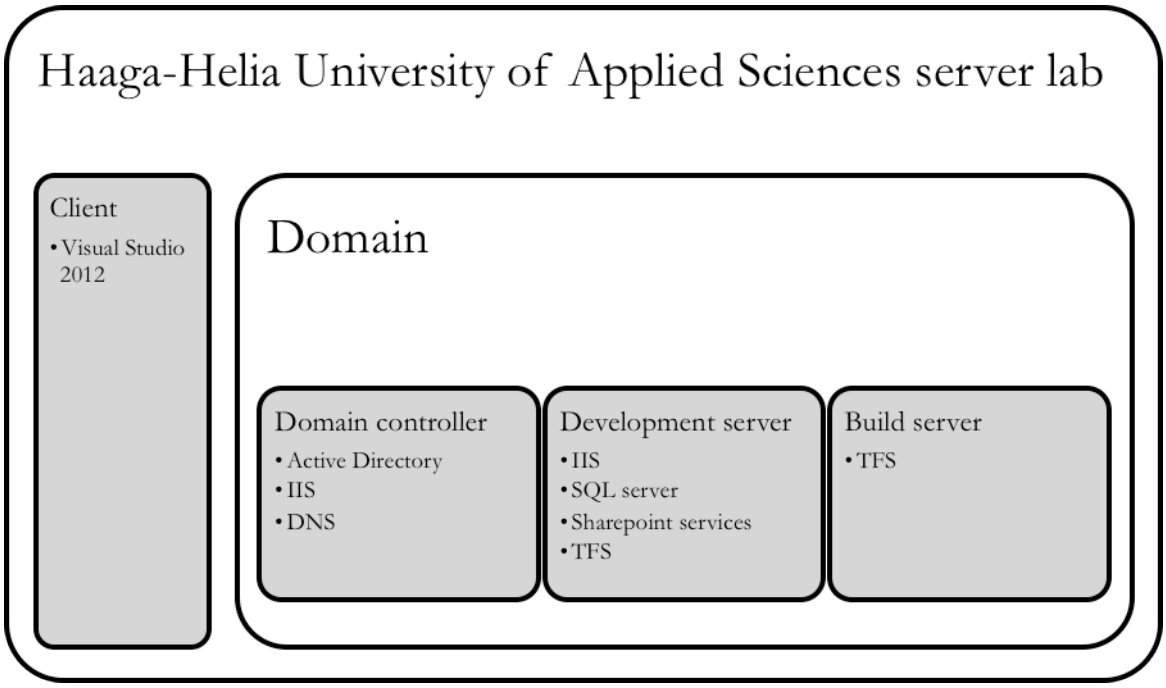
As part of the testing process, the test application was built by the build server. If the build failed, the errors were explored and handled before reattempting another build. If the build was successful manual testing would occur.

Builds of the test application were performed regularly throughout the development process, often several times in a day.

4.3 Setting up the test application environment

Before any development could begin, the environment to contain the development had to be created. Setting up the environment consisted of several steps: Setting up the domain controller, setting up the development server, setting up the build server and setting up the client.

Due to the domain existing within a Haaga-Helia University of Applied Sciences server lab, it was possible to forgo looking into server security principles as the Haaga-Helia University of Applied Sciences server lab had already been made to be a secure environment. Below is a diagram that presents the relationship between the server lab, the domain and the software installed on each server.



Setting up each of the servers started in the same way. Installation of Microsoft Server 2008 R2 SP1 64-bit and the network drivers to allow the PCs to connect to the internet via the server lab environment. In each case, the server PC was given a name that defined the server's role, a static IP address was given and any available updates we performed.

The domain controller had active directory installed in order to define the domain that would host the TFS environment as well as maintain the user profiles both for the TFS services and for the end users. Optionally, IIS and DNS was set up in order to make testing connections between the servers easier by allowing other servers to view a simple website under the domain url.

The development server required the most time to set up as it had the most required components. After the initial setup, the first step was to add the development server to the domain. In the second step, log on rights were provided to the required TFS service profiles were given the necessary permissions and the development server's firewall was turned off. Thirdly, IIS, SQL server and a report server were set up to enable reporting for the TFS projects. Fourthly, SharePoint services, a web application and a site collection were set up to allow web access to the TFS projects and reports. Finally, TFS was installed.

Setting up the build server was the fastest server to set up. Firstly Microsoft framework was installed and secondly the TFS build service was installed.

Setting up a client can be done from a clean PC or an existing client can be configured. If the client does not have windows installed yet, install the Windows 7 operating system, network drivers and define a static IP address that exists within the domain. Clean or existing clients should have Visual Studio 2010 installed and then connect to the TFS. From any client, a connection to the TFS should be made and the scrum template for TFS should be uploaded to the development server.

4.4 Setting up the scrum framework in TFS

Before the scrum framework could be setup in order to begin the project, it was necessary to connect to TFS via the client and upload the scrum template. Once the scrum template had been uploaded to the development server, it was possible to create a new project based on this template.

Once a user has logged in to TFS, they can create a project and begin to define the project members, work areas, sprints, product backlog items and build definitions.

4.5 Adding project members and defining permissions

Users in the active directory can be assigned to projects as users. Users can only view projects they are a user of and the user group type can assigned various privileges according to their scrum role. For example a product owner should not have write access to task items, only read access but may write product backlog items and have permissions to delete a project. Development team members should have access to create and edit tasks but not product backlog items. They may also be allowed to create and modify build definitions.

In this case, as there was only the author creating the test application, it was necessary to play multiple roles. Three user were created: ”Tom”, ”Dick” and ”Harry” where

Tom would be the development team member, Dick would be the scrum master and Harry would be the product owner.

4.6 Creating work areas

Work areas are non-essential notes used when creating and editing work items such as product backlog items, tasks and bugs. This enables easier searching and taking of tasks based on the work area. Work areas can be completely defined by the product owner (or development team if the permissions allow) and could be areas such as user interface, database, business logic and so on.

For this test application, the area were defined as: User interface; client logic; server logic; server database and application data.

4.7 Defining sprints

The sprints are one of the key features of the scrum framework and in TFS are known as iterations. For each sprint, a start date and end can be defined and each sprint object has place for noting the planning meeting, review meeting and retrospective meeting notes.

Once the sprints are defined, product backlog items and tasks can be assigned to a sprint thereby making them part of the sprint backlog.

As mentioned earlier in this document, this test application project had three sprints each spanning two weeks.

4.8 Adding product backlog items

Product backlog items are goals to be achieved in the project that sum up what the final project should do. They are often presented in the user story sentence: ” As a <type of user>, I want <some goal> so that <some reason>”. Later on the development team would break these down into smaller, manageable tasks.

Product backlog items are added to sprint backlogs (a list of PBIs that will be completed over a sprint) but in TFS, sprint backlogs do not exist as work items so PBIs are just defined as existing in a particular sprint.

The PBIs for this test project were made under Harry's user credentials and are listed here:

- A UI and code page that would allow users to create new levels
- A UI and code page that would allow users to edit their levels
- A UI and code page that would allow users to search, view and download user-made levels
- The client-side business logic to check users' levels for errors before allowing uploads
- The client-side business logic to allow users to edit, reset and delete user-made levels
- The client-side business logic to upload user-made levels to the server
- The client-side business logic to search user-made levels on the server
- The client-side business logic to download user-made levels from the server
- A server database to store all the user-made levels
- The server-side business logic to handle user-made level uploads
- The server-side business logic to handle search queries
- The server-side business logic to handle user-made level downloads
- Develop an additional 100 game levels.

4.9 Creating build definitions

A build definition defines the settings required to configure a build on a build server. The definition only needs to be setup once but it can be beneficial to set up multiple definitions particularly when it comes to the build trigger. Builds can be queued manually or triggered by events such as check-ins or on at specific times. It is quite common for build of complex applications to occur after the working day to allow plenty of time.

For this test application, only one build definition was configured under Dick' user credentials and was triggered manually.

4.10 Sprint one

Sprint one began on the twentieth of June and continued for two weeks. Following the scrum framework, some PBIs were selected to be undertaken over the course of the sprint. These SBIs were split into smaller manageable tasks and added as work items to the project in TFS.

The SBIs that were taken and subsequently broken down into smaller tasks under Tom's user credentials were:

- A UI and code page that would allow users to create new levels
- A UI and code page that would allow users to edit their levels
- The client-side business logic to check users' levels for errors before allowing uploads.
- The client-side business logic to upload user-made levels to the server

Due to the team size being one, the planning meetings were not done, however some time was spent planning the solution for achieving the SBI user stories.

The tasks were completed on time even though some obstacles became apparent. This was mainly due to much of the code used being recycled from other existing pages, particularly the UI areas.

The client-side logic dealing with level uploads proved a little tricky because it was a new area of communication to the author. The server used PHP MySQL-based web pages and functions to clean the user level text and upload it to the server database. On the client-side, a web client object was sufficient to pass the data to the PHP webpage. An example of how this was achieved can be found in appendix 8

Similar to the planning meetings, at the end of the sprint no review or retrospective meeting was held as having a team of one made it unnecessary.

4.11 Sprint two

Sprint two began on the fourth of July and continued for two weeks. Following the scrum framework, some PBIs were selected to be undertaken over the course of the sprint. These SBIs were split into smaller manageable tasks and added as work items to the project in TFS.

The SBIs that were taken and subsequently broken down into smaller tasks under Tom's user credentials were:

- A server database to store all the user-made levels
- The server-side business logic to handle user-made level uploads
- The server-side business logic to handle search queries
- The server-side business logic to handle user-made level downloads

Similar to sprint one, the planning meetings were not done, however some time was spent planning the solution for achieving the SBI user stories.

These tasks were the fastest to complete, in part due to the simplicity of the solution when it was thought to be more difficult and also in part to the assistance from outside sources and in particular a former student, colleague and continued friend of the author, Manuel Bacso who had experience in finding solutions for communication between WP7 applications and PHP MySQL-based web sites.

Because the SBIs were completed ahead of schedule, another PBI was broken into smaller tasks and taken into the sprint:

- Develop an additional 100 game levels.

In order to make this fast and easy to complete, Microsoft Excel was used to create a dynamic code-generating worksheet that only required clues and answers. The final product of the Excel worksheet was the C# code that could be copy-pasted into the project files. An example of this Excel sheet can be found in appendix 9.

This SBI was not completed before the end of the sprint as was returned incomplete to the product backlog

Also similar to sprint one, at the end of the sprint no review or retrospective meeting was held as having a team of one made it unnecessary.

4.12 Sprint three

In the same form as the previous sprints, sprint three began immediately after the previous sprint on the eighteenth of July and continued for two weeks. The remaining PBIs were taken into the third sprint backlog:

- A UI and code page that would allow users to search, view and download user-made levels
- The client-side business logic to allow users to edit, reset and delete user-made levels
- The client-side business logic to search user-made levels on the server
- The client-side business logic to download user-made levels from the server
- Develop an additional 100 game levels.

Similar to the previous sprints, the planning meetings were not done, however some time was spent planning the solution for achieving the SBI user stories.

The final tasks were completed in time without any major obstacles. Having a well-defined level class structure made transferring data from the web server to the phone easy. When a web client object makes a request to the server, an xml file is returned to the device and parsed into a level object. This object is then stored in the applications isolated storage for use later.

In this sprint the additional 100 levels were completed and added to the application without any issues.

Similar to the previous sprints, at the end of the third sprint no review or retrospective meeting was held as having a team of one made it unnecessary.

4.13 Final testing

As mentioned in the introduction to this chapter, time was designated after the third and final sprint for testing. This testing could be considered as a closed beta testing, as it was performed only by specific persons and they were already familiar with the product from testing during the alpha stage.

The primary purpose for this was to discover if any bugs existed that may have been overlooked by or predicted by the developer. The testing proved useful as several level bugs were found, primarily those concerning answer button performance (or lack thereof). These bugs were promptly fixed and retested to ensure they no longer existed.

5 Results and evaluation

At the end of the test application development process, the test application had been completed and was ready for release to the Windows Phone Marketplace.

A server-based database and related web pages exist to support uploading levels from a WP7 device to the server and also to support querying the server and downloading levels from the server.

An excel file was created to support the creation of creating built-in levels for the application.

Each scrum sprint was completed on time and with one exception, all PBIs, were completed within the sprint they were taken into.
The scrum framework had been followed, although not completely strictly. Some liberties were adapted in relation to the development team, meetings and meeting documentation.

The author was able to acertain the suitability of using TFS for WP7 application development by Haaga-Helia University of Applied Sciences BIT students.

5.1 Evaluation of results

The results of the test application are somewhat unclear and can be interpreted in various ways. If the determining factor is the completion of the project in compliance with strict scrum guidelines, the result could be argued to be incomplete due to the inconsistencies with a realistic scrum project. This can be counter argued in that a strict scrum project does not have to exist. As mentioned in chapter 2.2, scrum can be flexible and should be adapted to suit the development team.

If the understanding of the scrum framework is that it can be adjusted to fit the goals and work methods of the project members and the key goal is to provide a complete,

functioning release within the time frame allotted, it could be argued that the test application was successful.

In either case, the results provided were the product of a completed WP7 application alongside the other necessary server –based database and related pages.

5.2 Evaluation of test application

The test application was, as mentioned earlier, completed successfully and was completed with relative ease within the TFS environment. Because of the author's experience with all the areas included in the scope of this thesis, and therefore the test application, the test application was rather easy to complete with only a few areas that required learning new code classes or new methods of implementing classes that were known already.

The suitability of the test case can be argued both suitable and leaving room for improvement. The recommendations for future improvement are discussed in chapter 6 but briefly we could argue that because the test application did not reflect a true scrum project, it was not a suitable test for determining the suitability of scrum on TFS.
On the other hand however, it is not an impossible scenario that a BIT student may wish to complete a WP7 application by themselves and by using TFS to manage their software code and files and using the scrum template as a project plan. In this scenario, it can be argued that the test application was suitable and the results are valid.

Taking into account the limitations discussed in chapter 3.6, it can be argued that no matter the outcome of the test application, the test was inheritly flawed from the beginning and should have been dismissed outright which would have been considered if a "true" scrum project was essential for the testing which it wasn't. In this case, and indeed the case of many BIT student software application projects, there is almost always a learning aspect to each project which affects the scrum framework negatively. These negative aspects are, in the author's opinion, enough of a detour from a "true" scrum project that the detours caused by the limitation of the test application can be compared to be equal in severity and therefore supporting its validity.

5.3 Evaluation of learning

The author's learning during the test application was focused on discovering areas where scrum and TFS worked especially well as well as areas that they did not work well together at all. Secondarily, the author had the opportunity to learn some new areas of web application programming in regards to supporting communications between a WP7 phone and a MySQL server database via PHP.

Following are the areas of software development within the TFS environment that were discovered to be of particular importance to the test application's success.

It is important that any project work is completed within the TFS Environment. A key performance factor of the project is that everyone is working with up to date materials, especially in the early stages of the project. By completing your work within the TFS environment, you can ensure that all work is as up to date as possible for each member.

Each team member should create their own views of the project. TFS uses SQL queries to find PBIs, tasks and other work items in the project. Each member should set up their own views using the Query builders so that they can filter all the project information down to the parts that concern them. These can be a little work to start off with but they will save time in the long run.

There are often multiple ways of accomplishing the same task. The steps described during chapter 4 are just one examples of how to accomplish certain TFS tasks. Often there is more than one way so the team should be allowed to feel free to explore other methods and use them if they are preferable.

Checking out, checking in and comparing files is essential. Each file used should be checked out manually. Automatic check out may not select the correct lock type. A manual lock is recommended is to prevent other users using the file in order to keep

consistency but allowing other to check out and force a comparison check when the last file version is checked in.
When a file is checked in and another team member has checked in a version of that same file earlier, the last user will be prompted to do a comparison before overwriting or cancelling the check in. It is recommended that previous team mate's changes are carried through and if conflicts occur, to make changes to the last user's code whenever possible.

5.4 Summary and conclusions

As mentioned in earlier chapters, the purpose of the test application was primarily to determine whether TFS supports the scrum framework when developing a WP7 application for BIT students at Haaga-Helia University of Applied Sciences.

It is the author's conclusion that the TFS development environment could be a suitable tool for project development for BIT students at Haaga-Helia University of Applied Sciences however there are many issues that must be addressed or explored further in order to better determine its suitability.

The areas that require further research will be discussed in the following chapter. The issues that should be addressed are: Enabling access to the TFS server via the internet; The possibility of branching one project into several projects with the same structure; Testing with multiple persons; Testing reporting services; and Gaining familiarity with the TFS environment to enable more focus on the project development and less on the TFS administration.

6 Recommendations for future improvement

These areas have been identified as benefiting from further research and are: Testing with multiple persons involved; Testing reporting services; and

6.1 TFS server access via internet

As mentioned earlier in this paper, one major drawback was the necessity to be in the local environment to the servers, the Haaga-Helia University of Applied Sciences server lab. If an environment can be created that would also allow access from external connections, this would allow student more freedom in terms of the time and location the work on their development projects and could ultimately make the projects more successful.

6.2 Project branching for supporting multiple groups

In order for the TFS development environment to be a suitable development tool, it should be easy for the teaching lecturers to prepare the scrum structure such as sprint dates, work areas and PBIs for each group in a course. This may be possible through branching but was not researched in the test application.

6.3 Testing with multiple persons

One of the limitations of the thesis and and area that detracted from the authenticity of the test application was the completion of the test aplication by one person, who at time, would access the develeopment environment under differenct usernames in order to simulate different users. Further to this, the development work was planned and completed by one person which is not a valid scrum situation.

Therefore testing with multiple users, and more specifically multiple development team members would create a more realistic scenario and provide more reliable results.

6.4 Testing reporting services

The reporting services were not utilised in this test application as with a total member count of one, there was no need to create the reports as the the single member was well aware of the completion progress and burndown status of the sprint and project.

In order to improve the test application, the reporting services should be included and presented to a person who is not involved in the application development such as the product owner for example.

6.5 TFS familiarity

In order for the BIT students to get the most out of the TFS development environment, there is a need to know so of the key functionalities such as: Creating custom queries to find work items; The process of checking in and out files in order to reduce conflicts due to multiple versions of the same file; and if multiple versions of the same file are checked in, how to resolve conflicts and ensure that if the file contains updated lines of code, that they function correctly together.

Bibliography

Schah 2010, Object-Oriented and Classical Software Engineering, 8th Edition, McGraw-Hill Science/Engineering/Math.

Horstmann 2005, Object-Oriented Design and Patterns, 2nd Edition, Wiley.

Blankenship et al 2011, Professional Team Foundation Server 2010, Wiley.

Scrum Alliance, Scrum Guide, 19.08.2012.

Scrum.org 2011, Scrum Guide, http://scrum.org/Scrum-Guides, 19.08.2012.

Pfellger & Atlee 2009, Software Engineering: Theory and Practice, 4th Edition, Prentice Hall.

Microsoft 2011, Team Foundation Installation Guide for Visual Studio 2010, http://www.microsoft.com/en-us/download/details.aspx?id=24337, 19.08.2012.

Nathan 2011,101 Windows Phone 7 Apps Volume 1, Sams.

Appendices

Appendix 1. Setting up the domain controller

Installation and Configuration

1. Boot the Windows Server 2008 R2 64-bit installation DVD
2. Select language and choose Enterprise Edition full
3. Select custom Install and remove any existing partitions
4. Create the 1st partition of 66% of your HDD storage, create a 2nd with the remaining 33%
5. After Installation has completed, provide an Administrator Password and log in.
6. Change the computer name
7. Right-click Computer from the start menu -> properties -> change settings -> change… ->
8. restart when prompted
9. Install network drivers and check that you can access the internet
10. Configure IP settings
11. From the Network and Sharing center -> change adapter settings -> right-click local area connection -> properties
12. Uncheck IPv6 -> click IPv4 and click properties -> check Use the following IP address ->
13. Define an IP address, subnet mask, gateway and preferred DNS server and check your internet connection still works
14. Perform System updates

Setting up Active Directory

1. From the Start Menu run dcpromo -> Next -> Next
2. Check create a new domain in a new forest -> provide a FQDN for your domain -> Next
3. Select a forest functional level -> Next -> Ensure DNS server is checked -> Next -> Yes -> Next

Setting up IIS

- From the Server Manager -> Add Roles -> Check web server (IIS) -> Next -> Next
- Check the following options:
- Application Development
- Windows Authentication
- Click Next -> Install -> Close
- Check you can see your default web page on your Domain Controller IP address and FQDN

Setting up DNS

1. Create a reverse lookup zone
2. From the Start menu, type DNS to open the DNS Manager
3. Expand your server until you can see the Forward and Reverse look up zone folders
4. Right-click reverse lookup zones and select New Zone
5. Check Primary zone -> uncheck Store the Zone… -> Next -> Check IPv4 -> Next
6. Provide the network ID (The first three numbers of your Domain Controller IP address) -> Next
7. Check Create a new file… -> Next -> Check Do not allow… -> Next -> Finish
8. Right-click your FQDN from the Forward lookup zones -> New Host
9. Provide WWW for the name and your Domain Controller IP address for the IP address
10. Check the Create associated pointer -> Add Host -> Done
11. Test this www url on your development server after setting it up to ensure you access the website

Appendix 2. Setting up the development server

Installation and Configuration

1. Boot the Windows Server 2008 R2 64-bit installation DVD
2. Select language and choose Enterprise Edition full
3. Select custom Install and remove any existing partitions
4. Create the 1st partition of 66% of your HDD storage, create a 2nd with the remaining 33%
5. After Installation has completed, provide an Administrator Password and log in.
6. Change the computer name
7. Right-click Computer from the start menu -> properties -> change settings -> change… ->
8. restart when prompted
9. Install network drivers and check that you can access the internet
10. Configure IP settings
11. From the Network and Sharing center -> change adapter settings -> right-click local area connection -> properties
12. Uncheck IPv6 -> click IPv4 and click properties -> check Use the following IP address ->
13. Define an IP address, subnet mask and gateway
14. Define the preferred DNS server as your domain controller and check your internet connection still works
15. Try to access the www url of your domain controller
16. Perform System updates

Join the development server to the domain controller

1. Right-click Computer from the start menu -> properties -> change settings -> change… ->
2. Check Domain in the Member of area and enter the FQDN of the forest root domain

3. Enter the domain controller Administrator username and password
4. Restart the computer when prompted

Appendix 3. Setting up TFS on the development server

Create the necessary active directory accounts on the domain controller

1. Open the Group Policy Management Editor -> Access Policies -> Windows Settings -> Security Settings -> Account Settings -> Password Policy
2. Change Password must meet complexity requirements to Disabled
3. Change Minimum password length to 0
4. Change Minimum password age to 0
5. Open the Active Directory Administration Center and select the tree view
6. Right-click the domain and select New -> Organizational Unit ->TFS -> OK
7. Right-click the Users folder of this Organizational Unit and select New -> User
8. Create accounts for the following: Ensure that the password never expires
9. tfssetup
10. tfsbuild
11. tfsservice
12. tfsreports
13. wssservice

Add the tfssetup user account to local administrators

1. Open the server manager -> Configuration -> Local Users and Groups -> Groups -> Administrators
2. Click Add… -> type or search for tfssetup and click OK

Add accounts to the relevant GPO groups

1. Open the Group Policy Editor -> Windows Settings -> Security Settings -> Local policies -> User Rights Assignment -> Allow log on locally
2. Add User or Group -> Add tfsreports -> OK -> OK
3. Navigate to User Rights Assignment -> Log on as a Service -> Add tfsreports, tfsbuild, tfsservice and wssservice

Disable Windows Firewall and User Account Control settings

Appendix 3. Setting up TFS on the development server

1. Open Network and Sharing center -> Windows Firewall -> Turn Windows Firewall on or off
2. Turn off windows firewall in all 3 locations -> Click OK
3. Open the Change User Account Control Settings -> Change notify to Never notify -> Click OK
4. Restart the computer

Install IIS

1. Open Server Manager -> Roles -> Add Role -> Web Server IIS -> Next -> Next
2. Check the following boxes:
3. Application Development
4. Windows Authentication
5. IIS 6 Management compatibility
6. Click Next -> Click install -> Close

Install SQL Server 2008 R2

1. Begin SQL server installation from the SQL Server 2008 R2 DVD
2. Click Installation -> New Installation -> Check Setup Rules are all Passed -> Click OK
3. Click Install -> Check all Setup Rules are passed -> Click Next
4. Accept license terms -> Click Next -> Select SQL Server Feature Installation -> Click Next
5. Check the following:
6. Database Engine services,
7. Full Text Search,
8. Analysis Services,
9. Reporting Services,
10. Client Tools Connectivity,
11. Integration Services,
12. Management Tools – Basic,
13. Management Tools – Complete,

14. Client Tools Backwards Compatibility,
15. Client Tools SDK
16. Click Next
17. Check all Installation Rules are passed -> Click Next
18. Click Next -> Click Next
19. Click Use the same accounts for all SQL Server services -> Change Account Name to NT AUTHORITY\NETWORK SERVICE -> Click OK
20. Change SQL Server agent to Automatic -> Click collation -> Click Database Engine Customize -> Change Collation designator to Latin1_General _100 -> Check Accent Sensitive -> Click OK -> Click Analysis Service Customize -> Change Collation designator to Latin1_General _100 -> Click Next
21. Add current user -> Click Next
22. Add Current User -> Click Add… -> find name tfssetup -> Click OK -> Click Next -> Click Next
23. Check Installation Configuration Rules are passed -> Click Next -> Click Install -> Click Close

Configure Report server

1. From the Start menu -> Microsoft SQL Server 2008 R2 -> Configuration Tools -> Reporting Services Configuration Manager
2. Enter MSSQL Report Server instance name -> Click Connect (Notice the start button is disabled)
3. Click Web Service URL -> Change IP Address to Development Server IP address -> Apply
4. Click Database -> Click Change Database -> Select Create a new report database -> Next -> Click Test Connection -> Next
5. Provide a report server database -> Next -> next -> Next -> Finish
6. Click Report Manager URL -> Click on URLs value -> You should be able to view a webpage called SQL Server Reporting Services

Install SharePoint services 3.0

1. Start installation of WSS 3.0 SP2 from the downloaded file-> Accept terms -> Continue -> Advanced
2. Select Web Front End -> Install now -> After installation ensure Run now is checked and click Close
3. Click Next -> Click yes if prompted to restart service -> Select no, create a new server farm and Click Next.
4. Type in the Development server Computer Name in the Database Server field, wssservice domain\ username and password -> click Next
5. Change port number to 17012 -> Click Next -> Click Next -> Click Finish
6. Save bookmark of the WSS Central Administration page that opens. Rename shortcut as WSS – Application Management

Creating a web application

1. Open the WSS Central Administration page -> Application Management -> Authentication providers -> Web Application: -> Change Web Application -> SharePoint Central Administration v3
2. Select the new default zone -> ensure the Integrated Windows authentication is NTLM

Creating a site collection

1. Open the WSS Central Administration page -> Application Management -> Create or extend web application -> Create new web application
2. Create a new IIS website -> provide a description -> ensure authentication provider is NLTM
3. Application pool configuration user name and password should be domain\wssservice -> OK
4. You should see the application created page after a few minutes
5. Go to the Create Site Collection page provided by the Application created page
6. Change the Web Site Address to /sites/collection
7. Provide a Site collection title -> assign wssservice as the Primary Site Collection Administrator
8. assign tfssetup as the Secondary Site Collection Administrator

TFS Installation and configuration

1. Start installation of Team Foundation Server 2010 from the Team Foundation Server 2010 DVD -> Accept terms -> Click Next
2. Select only Team Foundation Components and Team Foundation Server -> Install
3. Click Configure -> Select Standard Single server -> Click Start Wizard
4. Click Next -> Enter wssservice name and password -> Click test -> Click Next
5. Click Next -> Click Configure -> Click Next -> Click the links to test -> Click Close -> Click Close

Appendix 4. Setting up the build server

Installation and Configuration

1. Boot the Windows Server 2008 R2 64-bit installation DVD
2. Select language and choose Enterprise Edition full
3. Select custom Install and remove any existing partitions
4. Create a partition using all of the available HDD space.
5. After Installation has completed, provide an Administrator Password and log in.
6. Change the computer name
7. Right-click Computer from the start menu -> properties -> change settings -> change… ->
8. restart when prompted
9. Install network drivers and check that you can access the internet
10. Configure IP settings
11. From the Network and Sharing center -> change adapter settings -> right-click local area connection -> properties
12. Uncheck IPv6 -> click IPv4 and click properties -> check Use the following IP address ->
13. Define an IP address, subnet mask and gateway
14. Define the preferred DNS server as your domain controller and check your internet connection still works
15. Try to access the www url of your domain controller
16. Perform System updates.

Join the development server to the domain controller

1. Right-click Computer from the start menu -> properties -> change settings -> change… ->
2. Check Domain in the Member of area and enter the FQDN of the forest root domain
3. Enter the domain controller Administrator username and password

4. Restart the computer when prompted.

Install Microsoft Framework 3.5

1. From the Server manager, select Roles, add role and check application server.
2. Click Next, click Next, click Next and click Install.

Install Team Foundation Server Build Service

1. Insert the Team Foundation Server installation DVD.
2. Open the x64 / x86 folder depending on the version of Windows Server you are running then locate and execute the Setup.exe file.
3. Accept the license terms and click Next.
4. Select only the Team Foundation Server Build Service and click install.
5. Once installation has completed successfully, click Configure to open the TFS configuration centre and click Start wizard to begin.
6. Click next to choose a project collection and click Browse to find a project collection.
7. Click Servers, click Add and type the url of your development server. Click OK, click Close, select the collection you wish to provide the build service for and click connect. You will be told how many build agents are currently assigned to the collection. Click Next.
8. Define how many build agents you wish to assign to the collection and click Next. Click Next to use the default system account.
9. Click Verify to run the readiness check and if all have passed, click Configure.
10. Once the configuration is complete, click Next and click Close.
11. Click Close to close the configuration wizard.

Creating a shared folder to store build outputs

1. Create a folder on the C: drive
2. Right click the folder and click properties
3. Click the sharing tab and click advanced sharing
4. Check share this folder and click permissions
5. Click Add, enter tfsbuild in the textbox and click check names

6. Once the user is resolved, click OK
7. Change the tfsbuild's permissions to Full control and click OK, OK and Close

Appendix 5. Setting up a new client

This process will allow you to set up a new client if the client has not been created and will exist as part of the domain. It is important to note that the existing client user must have been added to a project before they will be able to access any projects on the development server. This can be done by gaining access to TFS using the administrator credentials.

Installation and Configuration

1. Install Windows 7 Pro x64 from the Windows 7 Pro DVD
2. Install network drivers and check that you can access the internet
3. Change network settings so that the domain controller controls the DNS
4. Network Settings -> Uncheck IPv6 -> Double click IP v4 -> Check Use the following IP address
5. Give an IP address and Subnet Mask, the domain controller Default Gateway
6. Check Use the following DNS server addresses and give the domain controller IP address -> OK
7. Try to access the www url of your domain controller, if you succeed, you are able to access the domain.
8. Add the client to the domain
9. Open System properties -> Change Settings -> Change... -> Check Domain
10. Give FQDN of the forest root domain -> Confirm with domain\Administrator and password -> Restart
11. Perform updates

Setting up access to the Team Foundation Server

1. Install Visual Studio 2010 from the Visual Studio 2010 DVD.
2. Once installed, From Visual Studio -> Team Explorer -> Click Connect to Team Project -> Servers -> Add -> enter the url of the server (development server name + . + domain name) -> OK
3. Enter a valid username and password when prompted and click OK.

4. Choose the available projects if any and click Connect

Uploading a Scrum template to the Team Foundation Server

1. Install Microsoft Visual Studio Scrum Template 1.0 from the downloaded file
2. From Visual Studio -> Team Explorer -> Right-click the server -> Team Project Collection Settings -> Project Template settings
3. Click upload and locate the folder in which you installed the Scrum template

Appendix 6. Enabling access for an existing client

If the client computer is already setup and has visual studio installed, it is possible to provide them access to the TFS environment without adding them to the domain. It is important to note that the existing client user must have been added to a project before they will be able to access any projects on the development server. This can be done by gaining access to TFS using the administrator credentials.

Setting up access to the domain

1. Change network settings so that the domain controller controls the client DNS
2. Network Settings -> Uncheck IPv6 -> Double click IP v4 -> Check Use the following IP address
3. Give an IP address and Subnet Mask, the domain controller Default Gateway
4. Check Use the following DNS server addresses and give the domain controller IP address -> OK
5. Try to access the www url of your domain controller, if you succeed, you are able to access the domain.
6. Setting up access to the Team Foundation Server
7. From Visual Studio -> Team Explorer -> Click Connect to Team Project -> Servers -> Add -> enter the url of the server (development server name + . + domain name) -> OK.
8. Enter a valid username and password, indicating that the user exists on the domain controller of the domain domain\user -> OK
9. Select the projects available to you and click Connect.

Appendix 7. Creating and managing a scrum project

Within Visual Studio 2010, it is possible to create and manage projects. Some functionality is available to do this via the web access to Team Foundation Server but Visual Studio makes the process much easier to complete and provides a more comprehensive overview of the project as a whole.

The project undertaken by the author was to create, manage and complete development of a Windows Phone 7.5 application using Scrum methodologies. Because it is very likely that the reader could be using these instructions for alternative projects, instead of giving a step-by-step guide on how this specific project was managed, each step will be documented in a general sense so that the reader can use this document as a reference guide and apply the same steps to their project as and when needed.

How to connect to Team Foundation Server

The first step in any project is to connect to TFS and become part of the project.

1. Open Visual Studio 2012, click on the Team drop down menu and select Connect to Team Foundation Server.
2. Click on Servers…, click Add and give the url of your development server. The path and port will be added automatically.
3. Give the administration password of the development server to continue.
4. Once the server appears on your server list, click close.
5. Choose the collections and projects you wish to manage and click Connect.

How to create a new project

If you are responsible for creating the project, here is how you get started.

1. In Visual Studio, click on the View drop down menu and select Team Explorer.
2. Right-click on the collection you wish to add a project to and select New Team Project.
3. Provide a name and description of the project and click Next.
4. Choose a process template (Scrum 1.0) and click Next.

5. Choose whether to start a new source control folder or branch from an existing one and click Next.
6. Confirm the settings and click Finish. After the confirmation appears, click Close.

How to add users to a project

You will need to add users to a project. You may need to add the users on the domain controller before you can do the next steps.

1. Right-click on your project name, hover over Team Project Settings and click Group Membership.
2. Select the group you wish to add users to and click Properties.
3. In the Add member area, select Windows User or Group and click Add…
4. Enter part or the whole name of the user you want to add to the project and click Check Names. You may be asked for the domain controller administrator password. Enter the password if needed, select the user from the active directory and click OK.
5. Click OK to exit the current user group. Repeat for other groups as necessary and finally click Close to exit project groups.

How to modify user group permissions

Not every member should have full access to all facets of the project. This is how you define what areas they can work with.

1. Right-click on your project name, hover over Team Project Settings and click Security.
2. Here you can see the user groups for all projects as well as specific projects.
3. Select a group and check/uncheck the options below as required.
4. Click Close to exit the project security.

How to define the work areas

This is not a necessary step but it does help you to keep organized. By defining key areas that you are working in, it is then easier for members to browse and take tasks that belong I those respective areas.

1. Right-click on your project name, hover over Team Project Settings and click Areas and Iterations.
2. Areas help define what categories certain product backlog items, tasks, bugs and other work items relate to. For example: UI, Business logic, data logic, database, testing, etc. Click on Area and the Add a child node icon to add areas.
3. You can also remove and edit nodes by clicking on the appropriate icons.

How to define the amount of sprints

This is a necessary of Scrum and is fairly well supported by TFS. Defining the sprints before adding work items, helps the team to understand what the shorter term (sprint) goals are so they can better organize themselves.

1. Right-click on your project name, hover over Team Project Settings and click Areas and Iterations.
2. Iterations relate to the sprints of the project that may exist within a release. Click any release or sprint and you can use the icons above to remove or edit that node's location relative to its siblings or parents.
3. You can also add nodes by clicking on the Add a child node icon.

How to define the sprint properties

1. Expand your project in team explorer and expand the team queries. Double click All Sprints.
2. Using the remaining sprints you defined, double click a sprint to open its properties.
3. Edit the dates and other information and click Save Work Item.

How to add product backlog items

This is one the key facets to a Scrum project, without which the project would never get off the ground.

1. Expand your project in team explorer and right-click on Work Items.
2. Hover over New Work Item and select Product Backlog Item.
3. Define the title, iteration (release level at this point), status information, details information and any other information you require. Then click Save Work Item.

How to define a build definition

In order to queue builds of the project, a build definition must be defined to manage the builds. This tells the build server how to handle the project

1. Right click Build in team explorer
2. Click New Build Definition and provide a build definition name and description
3. Choose the trigger. I recommend you make one definition for manual builds and one scheduled build for each evening.
4. In the workspace section ensure that only relevant source control and build agent folders are included
5. Select your build controller from build defaults
6. Check build copies to already created drop folder on the build server and select the folder defined on your build server for storing the output files then click Save.

How to setup project alerts

A key factor in keeping up to date with group projects is to know things like when a team member's work items have been changed by others or as a scrum master, you might need to track all checked in items or build phases.

1. Right click on the project in Team Explorer and click Project Alerts
2. Check which events you wish to be alerted to and to which email address the alert should be sent to and then click OK.

How to add task items to product backlog items

In TFS, PBIs are not worked on or completed, only the tasks assigned to them. This helps the team to break down the PBI into the required tasks which, of course, should last no longer than 1 work day.

1. Search for the product backlog item by expanding your project in team explorer, expanding team queries and double-clicking Product Backlog.
2. Click Edit Query and delete all clauses except for Team project = @Project and Work Item Type = Product Backlog Item.
3. Click Save Query and click Run to see all the Product Backlog items below.
4. Double click the necessary product backlog item and click on the Tasks tab.
5. Click on New Linked Work Item, provide the title and comment if necessary and click OK.
6. Define the iteration, status information, details information and any other information you require. Then click Save Work Item.

How to define a project as the source control

The management of a project is a major part of the project as a whole but it is useless unless you have an actual project solution that the project is building. Here is how to set it up.

1. If you haven't already, create a new C#, WPF, ASP, etc project.
2. Expand your project in team explorer and double click Source Control to open the source control explorer.
3. Right-click your project and select Map to Local Folder
4. Browse with … button to the folder that contains your project files and click OK.
5. Click Map and if prompted, click the Get button.

How to add new files to the project

1. Open the project using the solution file from the Source Control
2. Right Click on the project and hover over Add and select Add New…
3. Choose the type of file you want to add.

4. After creating the file you would like to add, you will see the file in the solution explorer with a yellow plus symbol. This means that is it not yet added to the source control.
5. Right click the file and select Check In. You will be prompted to confirm the check in. Once this is completed you can see the file in the Source Control explorer and the file is now available for all users.

How to remove files from the project

1. Delete the file from the Solution explorer
2. You should see that file in the Source Control explorer with a red x symbol next to it.
3. Right click this file and select Check in pending changes
4. You will be asked to confirm that you want to delete the file. Select check in to do so.

How to retrieve files accidently deleted from the Solution Explorer

1. If you have accidently deleted a file from the solution explorer, you should see that file in the Source Control explorer with a red x symbol next to it.
2. Right click this file and click Undo pending changes.
3. Click Undo changes to return the file to the solution explorer.

How to associate project files with Tasks in the Scrum project

It helps to know which files relate directly to which task. In TFS, you can define this by associating file to tasks. This makes it easy for team members and testers to check the exact files that relate to a task goal.

1. Open the task you which to associate a file with.
2. In the Links tab, click Link to and select versioned item
3. Click browse and locate the file and select Latest Version from the Link to… options.
4. Add a comment if desired and click OK.
5. Click Save to save this association.

6. The associated file will appear in the Links tab and double clicking will open the file directly if there are no dependencies. If there are dependencies then you can open the file in the solution explorer instead.

How to announce an impediment

An essential part of scrum is identifying impediments and notifying the Scrum master of these so that they can removed. Here is how to do this:

1. In Team explorer, right click on Work items, hover over New Work item and select Impediment
2. Provide a title and other necessary information, ensuring that the assigned to field is your scrum master.
3. Provide a description and save the impediment. If you scrum master has set up a query to find all work items assigned to them, they should find it quickly.

How to report a bug

Any and every project will eventually find a bug. Reporting it early and solving it quickly are the key tasks to ensure that it doesn't disrupt the progress of the project.

1. In Team explorer, right click on Work items, hover over New Work item and select Bug
2. Provide a title and other necessary information, ensuring that the assigned to field is the person who has been working on the task or file.
3. Make sure that you link to the files where the bug exists if you are able to narrow it down. This will make it much easier for the team member to track down the area of code that needs to be fixed.
4. Save the file and if the team member has set up a query to find all work items assigned to them, they should find it quickly.

How to change the status of a work item to done or other status

Once you begin working on a task, you should be able to update its progress easily. Each time you check in an item, you should make sure that you update its status immediately after.

1. Locate and open the work item associated with your task.
2. Change the state to reflect its current status. You should also check the acceptance criteria (definition of done criteria) to ensure that these have been met.
3. Add some information in the description field if there is any important information about other team members may need to know when they need to use the file associated with this task (such as input criteria and output types on classes).
4. Ensure the task is linked to the file if not already and save.

How to queue a build

If you have a build definition set up you can then start to queue builds. This is how it can be done.

1. Right click on Builds in your project in Team Explorer and select Queue New Build.
2. Select the correct build definition and changed any other required option is necessary (I recommend to use the default settings) and click Queue.

The build explorer will automatically open to give you an updating status of the build progress.

Appendix 8. Server-side script for querying a PHP MySQl server

```
<?php
if(!$dbconnect = mysql_connect('localhost', 'username',
'password')) {
   echo "Connection failed to the host 'localhost'.";
   exit;
} // if
if (!mysql_select_db('database')) {
   echo "Cannot connect to database 'test'";
   exit;
} // if

$table_id = 'table';
$query = "SELECT * FROM $table";
$dbresult = mysql_query($query, $dbconnect);

// create a new XML document
$doc = new DomDocument('1.0');

// create root node
$root = $doc->createElement('root');
$root = $doc->appendChild($root);

// process one row at a time
while($row = mysql_fetch_assoc($dbresult))
{
          // add node for each row
        $occ = $doc->createElement($table_id);
        $occ = $root->appendChild($occ);
```

```
          // add a child node for each field
        foreach ($row as $fieldname => $fieldvalue)
        {
                $child = $doc-
>createElement($fieldname);
                $child = $occ->appendChild($child);
                $value = $doc-
>createTextNode($fieldvalue);
                $value = $child->appendChild($value);
        } // foreach
} // while

// get completed xml document
$xml_string = $doc->saveXML();

echo $xml_string;
?>
```