

# **Installing, configuring, securing and validating a software development environment using Team Foundation Server**

Tarek Sadik

Bachelor's Thesis

Degree program in Business Information Technology

January 2013



## Abstract

30.5.2013

<b>Author or authors</b> Tarek Sadik	<b>Group or year of entry</b> 2010
<b>Title of report</b> Installing, configuring, securing and validating a software development environment using Team Foundation Server	<b>Number of report pages and attachment pages</b> 50 + 94
<b>Teacher or supervisor</b> Juhani Välimäki	
<p>Nowadays, software development collaborative tools and application lifecycle management (ALM) tools are widely used by IT companies for managing efficiently their software development projects. Due to this wide use, HAAGA-HELIA UAS took the initiative to make a further research about the possibility of implementing a learning environment with an ALM tool to enhance the students' professional skills.</p> <p>The goal of this thesis is to implement Team Foundation Server and produce an installation guide of the system. These will be used by HAAGA-HELIA's IT students for learning purposes to enhance their professional skills. The thesis' goal will be achieved by gathering first the required theoretical knowledge about the topic which after it will be applied for the system implementation according to a list of requirements.</p> <p>The thesis focuses on studying the requirements for TFS, selecting the suitable deployment topology, system implementation, automating user management tasks with PowerShell scripts, testing and assessing the system security.</p> <p>Upon successful test results, the system will be deployed to the internet by HAAGA-HELIA's IT department.</p>	
<b>Keywords</b> Team Foundation Server, PowerShell, Active Directory	

# Table of contents

Abbreviations.....	i
Terms.....	iii
1. Introduction.....	1
1.1 Project's purpose and goals .....	2
1.2 The scope of this thesis project.....	3
1.3 Out of scope .....	5
2. Research plan.....	6
2.1 Constraints and delimitations .....	6
3. Background knowledge .....	8
3.1 Windows Server 2008 R2.....	8
3.2 Windows PowerShell.....	14
3.3 Team Foundation Server.....	15
4. Empirical part – Building the pilot laboratory .....	30
4.1 Physical security requirements.....	30
4.2 Installing the operating systems in the servers .....	31
4.3 Installing the operating systems in the work stations.....	32
4.4 Installing TFS.....	32
4.5 Configuring and securing the software installation on the servers.....	33
4.6 PowerShell scripts - Purpose and source code.....	34
4.7 Testing the pilot laboratory.....	36
4.8 Publishing the system to the internet .....	42
5. Evaluation .....	43

6. Further development .....	45
7. Summary .....	46
8. References .....	47
9. Appendices.....	51
Appendix 1: Installation guide.....	51
Appendix 2: Tables of values .....	44
Appendix 3: Course structure.....	46
Appendix 4: Students' list .....	47
Appendix 5: Display course implementation members or teams.....	48
Appendix 6: Display students of a specific entry year .....	50
Appendix 7: Create a course structure .....	52
Appendix 8: Create users and assign them to teams .....	58
Appendix 9: Removing a course structure.....	69
Appendix 10: Scenarios performed with a teacher account.....	74
Appendix 11: Scenarios performed with a student account.....	82

## Abbreviations

Term	Definition
AD	Active Directory
ALM	Application Lifecycle Management
BIT	Business Information Technology
CGI	Common gateway interface
DNS	Domain Name System
ELMS	Electronic License Management System
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GPO	Group Policy Object
HDD	Hard Disk Drive
IDE	Integrated Development Environment
IP	Internet Protocol
IIS	Internet Information Services
MBSA	Microsoft Baseline Security Analyser
MSDN AA	Microsoft Developer Network Academic Alliance
NX	processor bit
OU	Organizational Unit

PAE	Physical Address Extension
RAM	Random Access Memory
SDK	Software Development Kit
SQL	Structured Query Language
SSE2	Streaming SIMD Extensions 2
TFS	Team Foundation Server
WSS	Windows SharePoint Services

## Terms

Term	Definition
.NET	Software framework developed by Microsoft that includes a large class library, and running mainly on Windows machines
Domain	Refers to a network made of a group of client computers and possibly servers
DreamSpark Premium	Microsoft service meant for IT education departments which provides professional developer tools and software to students
File server	Is a computer belonging to a network and responsible for providing the location of shared disk access
Forest	A forest is a logical division in AD. Trees and domains are subdivisions of a forest
processor bit, NX	Allows the processor to help guard the PC from attacks by malicious software
Physical Address Extension	Enables 32-bit processors to access more than 4 GB of physical memory on capable versions of Windows and is a prerequisite for NX
SQL Server	Microsoft's database management system product
Streaming SIMD Extensions 2	Refers to features of the processor that enhance the reliability of third-party applications and drivers running in Windows 8
Team Foundation Server,	Microsoft's application lifecycle management and col-

TFS	laborative development tool
Version control	Is the management of the changes to source code in software development
Web server	Software responsible for providing web content accessible from the internet
Windows SharePoint Services	Technology developed by Microsoft that provides collaboration tools

# 1. Introduction

Information technology has been evolving constantly. The size and complexity of the software increased substantially as well as the size of development teams which led to the proliferation of software development collaborative tools (Bird & Zimmermann 2012). Software development collaborative tools and ALM tools are widely used nowadays by IT companies for managing efficiently their software development projects. ALM Software developers are focusing on developing comprehensive ALM tools that include coordinating the planning as well as the execution of software projects. Nonetheless, every ALM tool vendor has its own strong and weak areas in what concerns the different tools and functionalities within the ALM tool. The choice of the right ALM tool depends on many factors from which are: the size of the project, type of the project, development methods, location (geographically distributed teams), etc... (Duggan & Murphy 2012.)

According to Gartner, the ALM market estimated at 1.5 billion dollar for 2011 will continue to grow at a compound annual rate of about 3.9 % (Duggan & Murphy 2012). This growth shows that ALM tools adoption by IT companies is growing and consequently, the need of highly skilled IT professionals in the use of ALM tools will grow as well.

Due to the facts mentioned above and the efforts of HAAGA-HELIA UAS to enhance its IT students' professional skills, the need of implementing an ALM system at HAAGA-HELIA for learning purposes became evident. This thesis project tries to address that need.

Because of the nature of the environment at HAAGA-HELIA, Microsoft's ALM tool called TFS was the tool chosen by the customer representative. This choice was motivated by the following reasons:

- Support for Scrum framework as it is used in BIT's application development courses

- Visual Studio IDE is the development platform used in BIT degree program
- BIT degree program focuses on Microsoft technologies
- TFS is available in MSDN AA free of charge if used for teaching purposes only
- TFS is considered to be the most complete ALM tool on the market (Duggan & Murphy 2012).

## **1.1 Project's purpose and goals**

This thesis' project development goal is to create a TFS learning environment for HAAGA-HELIA's IT students. By implementing the system HAAGA-HELIA's IT students and more concretely BIT degree program students will have the opportunity to learn how to use Microsoft's ALM tool called TFS and many of its available features.

### **1.1.1 Pedagogical goals**

The main goal of this thesis project is to provide to the IT students of HAAGA-HELIA UAS; especially BIT students; a learning environment with TFS ALM tool. The learning environment will contribute in the learning of TFS implementation, functioning and administration, as well as software development in integrated platforms, and Scrum.

Most HAAGA-HELIA's BIT degree program's software development courses are based on teams of four students that do a project of 6 to 12 weeks (3 to 4 sprints of 2 to 3 weeks of duration per sprint). That specific length of the project will permit to the students of each group to try several Scrum roles i.e. Scrum master, Product owner and developer. Consequently, the students will try as many as possible TFS features and tools available for each Scrum role.

### **1.1.2 Learning goals of this thesis**

As the thesis author and a HAAGA-HELIA UAS student, the main objective of this project is to gain knowledge in building and administering a server environment with Microsoft products as well as installing, setting up, administrating and using TFS.

System expert professional path is not available in BIT degree program; therefore my skills in that field are very basic. By performing this project successfully, I will be able to demonstrate; in possible job interviews; that I have wide and deep practical skills in the previously mentioned professional path.

## **1.2 The scope of this thesis project**

This thesis' project scope is broad as it covers many parts of a system implementation process. The method agreed with the thesis project supervisor to fit the project in the thesis' schedule (400 hours), is the use of a list of requirements (APPENDIX 12) ordered by importance. In other words, the system's critical items and features will be implemented first, then the secondary ones will be implemented later if time remains. In the case of no time remaining, the secondary items will be proposed as system's future improvement recommendations.

The thesis project focuses on the technical part of TFS implementation according to a list of requirements (APPENDIX 12). This technical part includes choosing the adequate TFS deployment model, performing the installation, securing the system and defining the different needed access rights, creating Power Shell scripts to manage users and groups, testing the system and finally deploying it to the internet. The following list depicts the main phases of this thesis project:

- Installing and securing Active Directory and TFS
- Automating user and team management with PowerShell scripts
- Assessing the system security with test scenarios and with Microsoft Baseline Security Analyser (MBSA)

- Deploying the system to the internet

### **1.2.1 Installing and securing Windows Server and TFS**

This part consists of installing the system as a whole, configuring it and securing it according to the requirements (APPENDIX 12). First, the chosen operating systems will be installed in the machines then TFS and its required components will be installed according to the chosen topology. Finally, the system will be secured by performing the needed settings and actions.

### **1.2.2 Automating user and team management with PowerShell scripts**

Due to the nature of the project and its purpose described in the section 2.1.1, teachers have the exhaustive task of managing students and teams. The goal of this thesis project's part of is to ease and speed up teachers' tasks e.g. creating student accounts, creating teams, granting each student access rights to the system, etc... This objective is reached by creating Power Shell scripts that will perform the repetitive tasks for the teacher.

### **1.2.3 Assessing the system security with test scenarios and with MBSA**

In this phase of the thesis project, the system implementation will be assessed for security breaches and misconfigurations. The first part of the assessment will be done using Microsoft Baseline security analyser (MBSA), which is a tool developed by Microsoft to assess the security state in computers running Windows (Microsoft 2013a). The second part of the security assessment will be done by running test scenarios and comparing the obtained results of each scenario with the expected ones.

### **1.2.4 Deploying the system to the internet**

This phase of the thesis project consists of deploying the system to the internet so that the students can access TFS outside of the HAAGA-HELIA premises. The goal behind deploying TFS to the internet is to give to the students the possibility to experiment how geographically distributed teams can collaborate in a project using TFS.

### **1.3 Out of scope**

As mentioned in the section 2.2, this thesis project focuses on the technical part of TFS implementation according to a list of requirements (APPENDIX 12). Thus anything not included in the requirements list is considered being out of scope. Moreover, usability issues related to TFS, technical details of the environment where our system will be deployed, back up and failover strategies and finally secure connections implementation will not be addressed in this thesis paper.

## **2. Research plan**

The research method used to get the background theory of this thesis project is the literature review. This method consists of gathering theoretical knowledge and a profound understanding of the relevant concepts related to the thesis topic. The gathered knowledge will be put into practice to choose the right TFS implementation topology, create the required Power Shell scripts and finally to understand the software's installation guides and be able to perform the system implementation according to the requirements (APPENDIX 12).

The empirical part will be carried out in the following way: The first step consists of assessing the physical security of the servers, after which will start the phase of installing the chosen operating systems to the computers (servers and workstations). The next phase is the installation of TFS and its required components according to the chosen deployment topology. When the installation phase will be done, then we will proceed with the system configuration and security phase where the access rights and privileges will be set according to the requirements (APPENDIX 12). Finally, and in order to improve and speed up the student and course management tasks, the required scripts to carry out those tasks will be created.

In order to test the system security and user access rights, a test plan will be produced and performed. Upon a successful test result, the system will be published to the internet.

### **2.1 Constraints and delimitations**

This thesis project has some delimitations and constraints that might affect the workflow in it. The constraints can be divided in to two categories: technical and personal constraints.

### **2.1.1 Technical constraints**

This thesis project's implementation is meant to be performed within HAAGA-HELIA's premises and concretely in a computer laboratory belonging to the secured school's laboratories network. Accessing that network is only possible from HAAGA-HELIA's premises which restricted the thesis work flexibility and reduced the working time at some extent.

### **2.1.2 Personal constraints**

Performing a project thesis requires a certain level of knowledge in order to be able to carry out the work. As the thesis performer, I had very narrow skills in systems, networks and Power Shell commands and scripting. Performing this thesis requires to put a lot of efforts in gathering the needed skills.

### 3. Background knowledge

This chapter of the thesis is of major importance as it is the foundation for the work to be done. The theoretical knowledge needed by the thesis author to be able to carry out the work will be described in this chapter.

#### 3.1 Windows Server 2008 R2

Windows Server is an operating system developed by Microsoft and meant to run on server machines. The first version of Windows Server called Windows NT Advanced server was released on 1993. This release was a basic network operating system server (Long 2008).

After the first release of Windows Server, there have been several releases. Each new release brought new functionality and features as well as improvements in the whole system (Long 2008).

Windows Server series are more powerful versions of the desktop operating system counterparts. They were designed to handle corporate networking, internet and intranet hosting, databases, enterprise messaging, etc...(Webopedia 2013.)

Windows Server 2008 R2 is the penultimate edition of the Microsoft Server just before Windows Server 2012 and was released on October 2009 (Webopedia 2013). According to Microsoft (2009a) Windows server 2008 R2 offers plenty of services and technologies that can be listed as follows:

- Active Directory: Defined as a special-purpose database that consists of objects and attributes and used to stores objects like users, computers, groups, etc...
- Application server: Provides an integrated environment for deploying and running custom applications built with Microsoft .NET framework
- Failover clusters: They are groups of computers that work together to increase the availability of the server services. In other words, when a service in a physi-

cal server fails, another machine automatically takes care of providing that specific missing service

- Fax server: Used for managing shared fax resources
- File services: group of technologies used in managing storage, enabling file replication, managing shared folders and enabling access for UNIX client computers
- Group policy: “Provides an infrastructure for centralized configuration management of the operating system and applications that run on the operating system” (Microsoft 2009b).
- Hyper-V: It is a virtualization environment where virtualized server environments can be created, managed and run
- Print and document services: They enable the user to share printers on a network as well as allow their centralized management
- Remote desktop services: Provides technologies that allow the access of a Windows-based computer or server from other computers
- Web server (IIS): Unified platform for web publishing that integrates different Microsoft technologies

### **3.1.1 Active Directory Domain Services**

Microsoft (2013b) defines Active Directory Domain Services as “The foundation for distributed networks built on Windows 2000 Server, Windows Server 2003 and Microsoft Windows Server 2008 operating systems that use domain controllers. Active Directory Domain Services provide secure, structured, hierarchical data storage for objects in a network such as users, computers, printers, and services. Active Directory Domain Services provide support for locating and working with these objects.”

Active Directory provides the fundamental features which every directory service should provide. The fundamental features include: Location transparency (find and object data without the object address), object data (store object data in a hierarchical tree), rich query (locate objects by querying for object properties) and high availability (directory located in a location efficient for read/write operations). (Microsoft 2013d.)

In addition to the fundamental features, Active Directory Domain Services provides some advanced features like the support for internet standards, tightly integrated and flexible security, possibility of accessing the server programmatically thanks LDAP API and some other APIs, Directory enabled system services, key application integration and finally rich and extensible schema. (Microsoft 2013d.)

### **3.1.2 Server roles**

Windows Server 2008 R2 provides multiple server roles. However, we will focus in this theory part on defining the roles that are directly related with our thesis project.

#### **3.1.2.1 Domain controller**

A domain controller can be defined as a server that is running some specific version of Windows Server and configured with the role of domain controller by installing Active Directory Domain Services in it (Microsoft 2010).

The main function of a domain controller is to store the objects for the domain in which it is installed. This means that every machine or user that belongs to the domain will have his data stored in Active Directory (Microsoft 2010).

A domain controller can be designated as a Global Catalogue Server that stores the objects of all the domains in the forest. However, a domain designated as a Global Catalogue Server will store its own full writable domain replica (all objects and attributes), but will only store partial replica of the other domains (limited set of values). (Microsoft 2010.)

A domain controller can also hold the Operation Master Role. This role consists of performing some specific tasks to ensure consistency (eliminating conflict and duplicates in AD database). (Microsoft 2010.)

### **3.1.2.2 DNS server**

Domain Name System (DNS) is the name resolution protocol for TCP/IP networks and can be defined as a hierarchical distributed database that contains host names and their corresponding IP addresses. Many computers connected to the internet host part of the DNS database and the software that allows others to access it. This type of computers is known as DNS servers. (Shuler 2012.)

Shuler (2012) states that: “no DNS server contains the entire database; they only contain a subset of it. If a DNS server does not contain the domain name requested by another computer, the DNS server re-directs the requesting computer to another DNS server.”

A DNS server’s duty is to resolve the incoming queries for the host names into IP addresses or vice versa, so that the client computer can understand the address of the host computer.

### **3.1.2.3 IIS server**

IIS Server is web server software created by Microsoft and runs on Windows operating systems. IIS server is a means to share information with users on the internet, intranet or extranet. The IIS server role includes Internet Information Services (IIS), which is a unified platform that integrates the following Microsoft technologies: ASP.NET, Windows communication foundation and Windows SharePoint services. (Microsoft 2009c.)

IIS server provides plenty of features and services depending on the Version of Windows Server it is running on. According to Microsoft (2013e), the latest version of IIS (IIS 7) provides the following features when running on Windows Server 2008 R2:

- HTTP features: Static content, Default document, Directory browsing, HTTP errors and HTTP redirection
- Application development features: ASP.NET, .NET extensibility, ASP, CGI, ISAPI extensions, ISAPI filters and Server-Side Includes (SSI)
- Health and diagnostic features: HTTP logging, logging tools, request monitor, tracing, custom logging and ODBC logging
- Security features: Basic authentication, Windows authentication, Digest authentication, client certificate mapping authentication, URL authorization, Request filtering and IP and domain restrictions
- Performance features: Static content compression and dynamic content compression
- Management tools: IIS management console, IIS management scripts and tools, management service, IIS Metabase compatibility
- Windows Process Activation Services Features: Process model, .NET environment and Configuration APIs
- File transfer protocol (FTP) publishing service features: FTP server and FTP management console
- IIS 7 provides for Windows Server 2008 R2 unlimited amount of simultaneous connections

### **3.1.3 Server security**

Server security or server hardening is the process of enhancing a server's security level by performing a number of actions and by using some means to achieve that goal. The result of the process is a more secure operating environment. ([serverhardening.com](http://serverhardening.com).)

A server has the delicate tasks to deliver data in a reliable and secure way following the concepts of confidentiality, integrity and availability. This means that every server considered as secure must guarantee the three previously mentioned concepts in a way to prevent unauthorized access to the server, unauthorized access to its resources and services and finally any disruptions or interruptions in the services. (Bryant University 2009.)

According to Jansen, Scarfone and Tracy (2008), server security principles include the following:

- Simplicity: Keeping everything as simple as possible as complexity is security mechanisms may lead to breaches in security.
- Fail-safe: Failures of the system should occur safely i.e. security mechanisms should stay operative in case of failure in the system.
- Complete mediation: The providing of the information should happen through security mediators that enforce access policies.
- Open-design: The security system should be totally independent from the implementation
- Separation of privilege: Consists of separating privileges as much as possible. E.g. a system administrator can have an administrator account and a normal user account. The goal behind this is that when he is not administering the system he can log in as normal user as it is more secure.
- Least privilege: The idea behind this principle is to give to each user the exact privileges he needs for his role without any extra privileges.

Going through every single security feature in Windows Server 2008 R2 is an exhaustive task for this thesis as there are so many security features and settings available in it. Instead, there are many recommendations and check lists available for the public from several organizations. One example of these lists is under the following link:

(<https://wikis.utexas.edu/display/ISO/Windows+2008R2+Server+Hardening+Checklist>) and it describes a Windows Server 2008 R2 server hardening checklist used by the IT department of the University of Texas at Austin. The checklist is based on the recommendations of the “Centre for Internet Security” (CIS).

### **3.1.3.1 Microsoft Security Baseline Analyser**

Microsoft Security Baseline Analyser is a free and easy-to-use tool designed by Microsoft for IT professionals to assess security state of machines running Windows operating systems. The assessment is done based on Microsoft’s security recommendations and possible solutions to the problems found are offered by the tool. (Rains 2012.)

Rains (2012) stated that “MBSA is a standalone security and vulnerability scanner designed to provide a streamlined method for identifying common security misconfigurations and missing security updates. MBSA is used by many leading third-party security vendors and security auditors and, on average, scans over 3 million computers each week”.

MBSA will scan Windows machines for Windows administrative vulnerabilities like weak passwords, presence of known IIS and SQL administrative vulnerabilities and missing security updates (Rains 2012). The scanning process is done based on the computer’s IP address, domain or computer name. At the end of the scanning, MBSA will provide a detailed security report about the found vulnerabilities and the instruction on how to convert your working environment into a more secure environment. (Rains 2012.)

## **3.2 Windows PowerShell**

The term “Shell” in IT refers to a piece of software that wraps around the operating system’s kernel and where built-in commands or external commands can be entered to run programs. (O’neill, Rosen & Stanek 2009, 57.)

According to O’neill, Rosen & Stanek (2009, 57), Microsoft has one single shell: COMMAND.COM in DOS-based operating systems, and CMD.EXE in the NT-based operating systems. Although Microsoft has been adding more functionalities and features in every new version, the language remained the same for 20 years. Basically, PowerShell is offering the same functionality that existed before in addition to new features and its aim is to give Windows a modern shell.

PowerShell has the capability to understand different kinds of commands and this is possible because its commands are aliased to their equivalents in other shells. New Commandlets (Cmdlets) can be added to PowerShell to extend its functionality and users can even define their own functions. (O’neill, Rosen & Stanek 2009, 88.)

### **3.3 Team Foundation Server**

Team foundation server is Microsoft’s ALM tool and its solution for providing core collaboration functionality for software development teams in a very tightly integrated product. TFS is specifically designed for engineering teams with developers, testers, architects, project managers, business analysts and any person contributing to software development projects (Blankenship, Holliday, Keller & Woodward, 2011, 3-4.)

The functionality facilitated by TFS includes: project management, work item tracking, version control, test case management, build automation, Reporting, and virtual lab management (Blankenship, Holliday, Keller & Woodward, 2011, 4.)

#### **3.3.1 TFS Architecture**

TFS uses several components and services. Some of these are required for the correct functioning of TFS and others are optional. Thus, understanding TFS architecture is a key factor for a successful TFS implementation. According to Microsoft:

TFS employs a logical three-tiered architecture, including client, application, and data tiers. TFS clients interact with the application tier through various Web services; the application tier is in turn supported by various databases in the data tier. (Bansode, Jones, Mackman, Meier & Taylor, 2007).

The different TFS logical tiers are depicted in the figure 1:

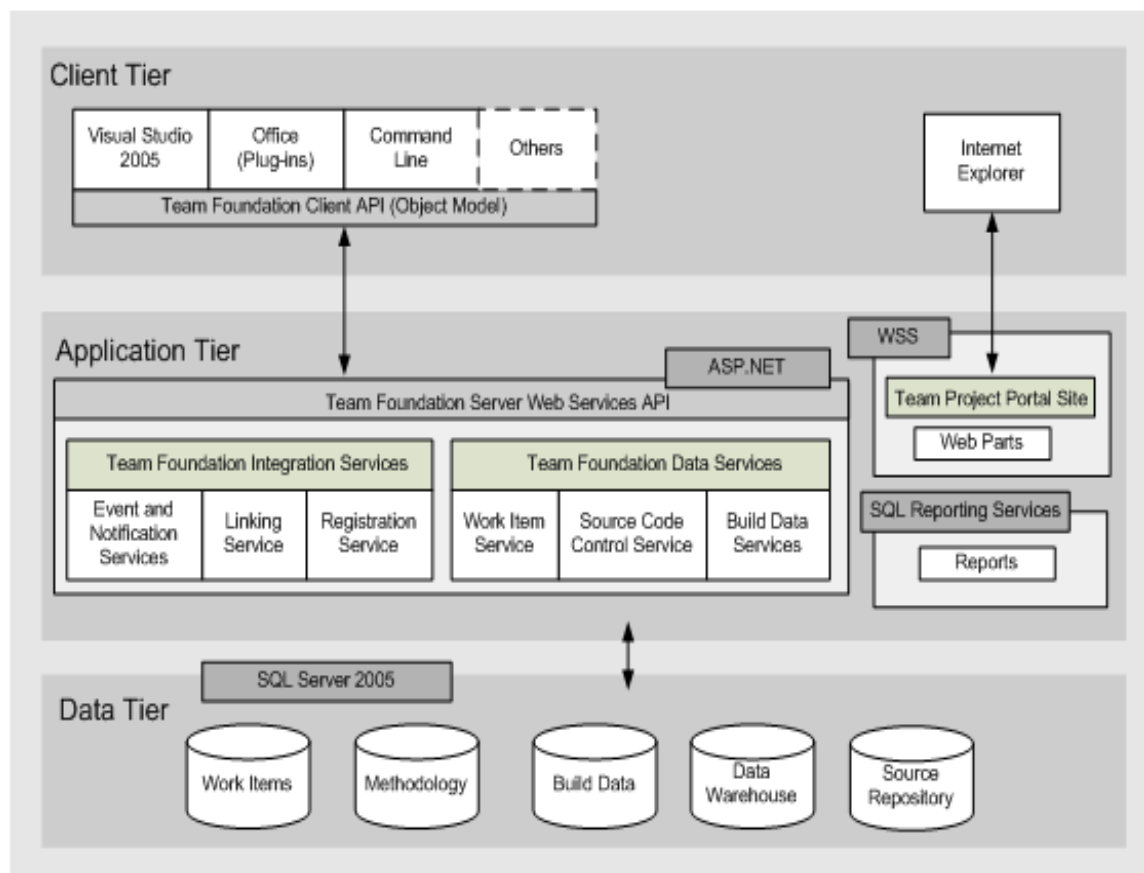


Figure 1: TFS components and tiers (Bansode, Jones, Mackman, Meier & Taylor, 2007).

**Client Tier:** According to (Bansode, Jones, Mackman, Meier & Taylor, 2007), the client logical tier is composed of the following components:

- **Team Foundation Client API (Object Model):** The public API to interact with TFS. In other words, the interaction between the different application tiers goes through the API in question
- **Visual Studio:** Microsoft’s IDE and a means to connect and interact with TFS.
- **Command line tools:** A set of tools and commands that enables the users to interact with TFS from the command line
- **Other components:** Third party tools that can be used to interact with TFS

- **Office plug-ins:** Set of add-ins that permits the interaction of some Microsoft Office tools (Microsoft Excel and Project) with TFS

**Application Tier:** This logical tier is composed of two ASP.NET web services accessed from the client tier through an API called Team Foundation Server web services API (Bansode, Jones, Mackman, Meier & Taylor, 2007).

The first service is called Team Foundation Data Services, and consists of a set of web services responsible for manipulating data in the data tier (Bansode, Jones, Mackman, Meier & Taylor, 2007):

- Version control web service: Service that provides interaction with the source control database
- Work Item Tracking Web Service: Service used by the client to create, update and query work items in the Work Item tracking database
- Team Foundation Build Web Service: Service used by the client and the MSBuild framework to execute build processes

The second service called Team Foundation Integration Services is responsible for providing integration and automation functionality, and does not interact with the data tier. According to (Bansode, Jones, Mackman, Meier & Taylor, 2007), the Team Foundation Integration Services includes the following components:

- Registration Web Service: This service is responsible for registering other TFS services and keeping their registering data in a registration database
- Security Web Service: Service consisting of a service called Group security service responsible for managing all TFS users and groups, and another service called Authorization service responsible for providing an access control system for TFS

- Linking Web Service: This service is responsible for linking tools with data elements they hold
- Eventing Web Service: This service enables users to register to events and receive notification through email or by invoking a web service

**Data tier:** Bansode, Jones, Mackman, Meier & Taylor (2007) state that this logical tier consists of various databases that can only be accessed through the web services on the application tier. Each database is storing data of some specific component of TFS:

- Work Item Tracking: Work Items' related data is stored in this database
- Version control: Source control related data is stored in this Team Foundation Build: Team build feature related data is stored in this database Reporting Warehouse: The information related to all TFS tools and features is stored in this database. The data stored in this database is used in the creation of reports with data from multiple tools

### 3.3.2 TFS deployment topologies

TFS deployment can be done using a variety of topologies ranging from simple single-server installations to complex multi-server topologies (Bansode, Jones, Mackman, Meier & Taylor, 2007).

#### 3.3.2.1 Single-Server topology with Active Directory

Single-server deployment is the simplest topology that can be used to implement TFS. In this implementation model, TFS and all its required components are located in the same physical server, i.e. the application tier and data tier components are installed on the same server and thus, the same domain. (Bansode, Jones, Mackman, Meier & Taylor, 2007.) The picture of the figure 2 describes a single-server topology.

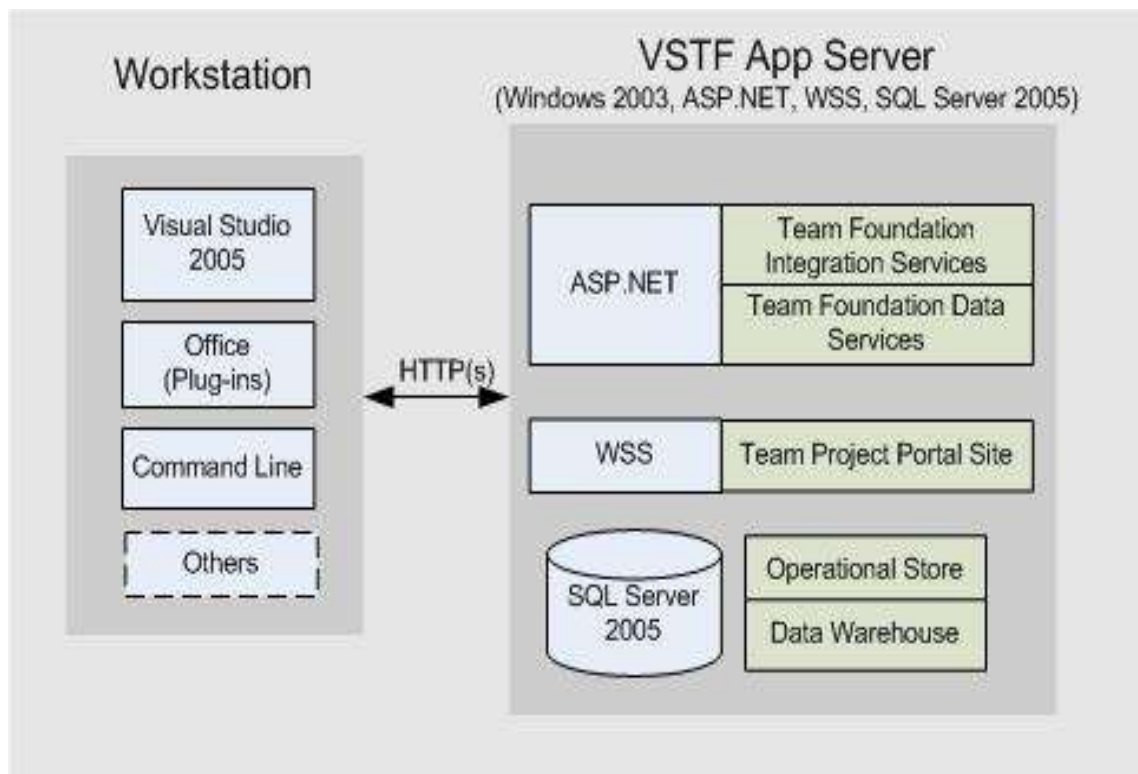


Figure 2: TFS single-server topology (Bansode, Jones, Mackman, Meier & Taylor, 2007).

According to Bansode, Jones, Meier, Taylor and Mackman (2013), single server topology offers the following benefits:

- **Simplicity:** All the aspects of TFS can managed in a single-server and the maintenance tasks are scheduled for one server.
- **Availability:** No network restrictions or network latency considerations are considered when planning the deployment, as both the application tier and data tier are on the same server.

### 3.3.2.1 Dual-server topology

Dual-server topology consists of separating the application tier and data tier by installing them on two different servers, however in the same domain. This deployment topology is suitable for large development teams in the range of 2000 users. (Bansode, Jones, Mackman, Meier & Taylor, 2007.)

In this type of deployment topology, build servers can be installed on the same server dedicated for the application tier. However, if the type of projects to be performed need performance testing, then it is recommended to implement dedicated build servers. (Bansode, Jones, Mackman, Meier & Taylor, 2007.)

The picture of the figure 3 describes a dual-server topology:

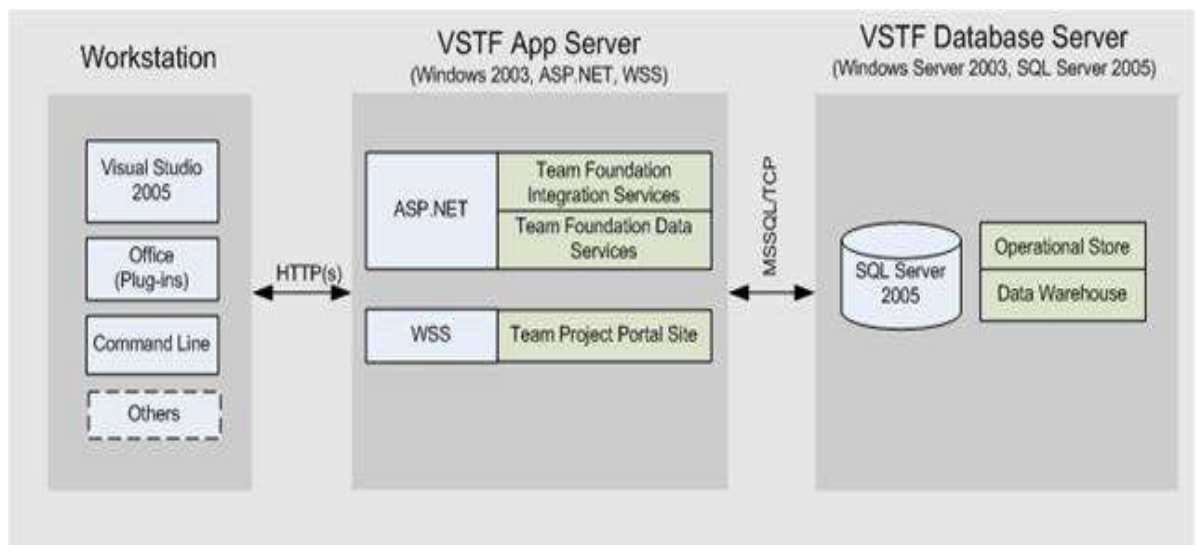


Figure 3: TFS dual-server topology (Bansode, Jones, Mackman, Meier & Taylor, 2007).

### 3.3.2.2 Hardware requirements for TFS implementation topologies

One of the key factors in making a decision on the TFS deployment topology is the number of users. This part is called the capacity planning of TFS (Bansode, Jones, Meier, Taylor and Mackman, 2013). In order to have a correctly functioning TFS implementation, the right hardware has to be chosen. The hardware choice is done based on Microsoft's specifications, the chosen topology and the number of users. The table of the figure 4 describes the minimum hardware specifications per deployment option and maximum number of users:

Deployment Option	REF <sup>17</sup>	Example specification	Physical Hardware Max users
Single Server	1	1 x ATDT: 1 Processor, 2GB RAM, 125GB Disk	250
	2	1 x ATDT: 1 Dual Core Processor , 4GB RAM, 300GB Disk	500
Scale-up Servers	3	1 x AT: 1 Dual Processor, 4GB RAM, 500GB Disk	2200
	4	1 x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	
	5	1 x AT: 1 Quad Core Processor , 8GB RAM, 500GB Disk	3600
Scale-out Servers	7	n x AT: 1 Dual Processor, 4GB RAM, 500GB Disk	3600+ <sup>18</sup>
		m x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	

Figure 4: minimum hardware specifications per deployment option and maximum number of users (Bansode, Jones, Meier, Taylor and Mackman, 2013)

### 3.3.3 Key requirements for TFS

Regardless of the chosen topology when making a TFS implementation, there are a number of key requirements that are essential for the correct functioning of TFS. Those key requirements have to imperatively be taken into consideration.

According to Bansode, Jones, Meier, Taylor and Mackman (2013), the key requirements for TFS implementation are as follows:

- The application tier and data tier must be on the same domain, although they could be implemented on different servers belonging to the same domain.
- TFS requires a server running Microsoft Windows Server 2003 service pack 1 or a later version.
- TFS application tier's web services must be installed on the same server.
- Single TFS instances must be installed on a single physical server.
- Only one instance of TFS can be installed per physical server.

- TFS databases must be in one single database server, and cannot be distributed across multiple database servers.
- An existing Microsoft SharePoint portal cannot be used to host the team project portal. Instead, a dedicated server should be used for SharePoint portals.
- TFS must not be installed on a domain controller, because this is not supported.
- For dual-server deployments, some user accounts must be created beforehand when running TFS services.

### **3.3.4 TFS Security and privileges**

Security, access and privileges are very important parts of the TFS implementation process. A well-defined security policy and granting the right privileges to the right groups and users will guarantee the confidentiality, integrity and the availability of the data (Krutz & Vines, 2004, 5).

#### **3.3.4.1 Users**

This is considered as the first security concept in TFS. According to Blankenship, Holliday, Keller & Woodward (2011, 511 - 513), TFS has two types of users:

- Domain users: They are defined as the users created and stored in the Active Directory. Domain users could be a person's account, Service account or a machine account.
- Local users: They are defined as the users that belong locally to a machine but not to a domain. Those users can have certain rights but exclusively on that specific machine.

### 3.3.4.2 Groups

According to Blankenship, Holliday, Keller & Woodward (2011, 511 - 513), groups are the second security concept that everyone willing to implement TFS should be familiar with. The groups used in TFS include:

- Domain groups: Sometimes referred to as Security groups, Domain groups are defined as groups that exist in Active Directory. These groups are used to secure resources and restrict user access to the domain.
- Distribution groups: They are defined as groups belonging to Active Directory, but are exclusively used by Microsoft Exchange to send mail to a list of recipients. Distribution groups cannot be used under any circumstance to secure resources or restrict user access.
- Local groups: These groups exist only locally in a machine, and can only restrict the access the resources of that single machine. These groups are very useful in TFS implementation because they permit to grant to domain users and groups specific privileges on a specific machine.
- Team Foundation Server groups: They are forming TFS' group structure and are used to restrict user access to certain objects in TFS. There are three different levels within TFS:
  - Server groups: These are the server default groups and cannot be changed, removed or modified. Detailed information about these groups is depicted in the table of the figure 5.

Name	Users	Description
SharePoint Web Application Services	Service accounts for SharePoint Products. If you installed SharePoint Products as part of installing Team Foundation Server, the service account that you specified during installation will be populated as a member of this group.	This group has service-level permissions that are required for interoperation with SharePoint Products. If your deployment of Team Foundation Server does not include SharePoint Products, this group should be empty. This group should only contain service accounts. Do not add user accounts to this group.
Team Foundation Administrators	Administrators for Team Foundation Server. The account that you use when you install Team Foundation Server is automatically a member of this group.	This group is the most powerful user group in Team Foundation Server. Members of this group can perform all tasks in Team Foundation Server at the server level, the collection level, and the project level. You should limit membership in this group to only those users who must administer the deployment of Team Foundation Server.
Team Foundation Service Accounts	Service accounts for Team Foundation Server. The service account that you specified for Team Foundation Server during installation will be populated as a member of this group. By default, the system account Network Service is used.	This group has service-level permissions that are required for Team Foundation Server and its services, such as the Visual Studio Team Foundation Background Job Agent. This group should contain only service accounts. Do not add user accounts to this group.
Team Foundation Valid Users	All accounts that have access to the deployment of Team Foundation Server.	You cannot directly modify membership in this group. This group automatically contains all users and groups that have been added anywhere within Team Foundation Server.
Work Item Only View Users	Users whose Web access to Team Foundation Server is restricted to viewing work items.	This group is restricted from using the full range of features that are provided when users view projects and collections in Team Web Access.

Figure 5: Team Foundation Server groups (Microsoft 2011, Server-level groups' link)

- o Team project collection groups: These groups are created as a part of the process of creating a new Team project collection and they are tightly related to their Team project collection. Consequently these groups can only control access privileges to the Team project collection they were created with. Detailed information about these groups is depicted in the table of the figure 6.

Group	Permissions
Project Collection Administrators	Members of this group can perform all operations within a team project collection.
Project Collection Valid Users	Members of this group have access to the team project collection. You cannot manually add or remove users or groups from this group.
Project Collection Service Accounts	Members of this group are the service accounts used by the team project collection. This group should not contain any user accounts.
Project Collection Build Service Accounts	Members of this group are the service accounts used for creating and running builds on build computers. This group should not contain any user accounts.
Project Collection Proxy Service Accounts	Members of this group have proxy service permissions for the collection. This group should contain only service accounts and groups that contain only service accounts.
Project Collection Test Service Accounts	Members of this group have test service permissions for the collection. This group should contain only service accounts and groups that contain only service accounts.

Figure 6: Team project collection groups (Microsoft 2011, default groups' link)

- Team project groups: These groups are created as a part of the process of creating a new Team project and they are tightly related to their Team project. Hence, these groups can only control access privileges to the Team project they were created with. Detailed information about these groups is depicted in the table of the figure 7.

Group	Permissions
Project Administrators	Members of this group can manage all aspects of the team project.
Readers	Members of this group can see information about the team project (read access).
Contributors	Members of this group can modify the team project (read and write access).
Builders	Members of this group can create and modify builds.

Figure 7: Team project groups (Microsoft 2011, default groups' link)

### 3.3.4.3 Permissions

Permissions are the actions that a user is allowed to perform in TFS. There are more than 80 different built-in permissions in TFS (Blankenship, Holliday, Keller & Woodward, 2011, 519).

The permissions in TFS are based on many levels; nonetheless the scope of this thesis project focuses exclusively on the following permissions:

- Server permissions: These permissions have the broadest scope as they are applied on the server level. Generally, they affect the members of the administrators group, and their purpose is to prevent an administrator from being locked out of the system (Blankenship, Holliday, Keller & Woodward, 2011, 519). The table of the figure 8 describes the server permissions

Permission Name	Name at command line	Description
Create team project collection	CreateCollection	Users who have this permission can create and administer team project collections in Team Foundation Server.
Delete team project collection	DeleteCollection	Users who have this permission can delete a team project collection from the deployment. <b>Note</b> Deleting a team project collection will not delete the database for the collection from SQL Server.
Edit instance-level information	GENERIC_WRITE tf: AdminConfiguration tf: AdminConnections	Users who have this permission can edit server-level permissions for users and groups in Team Foundation Server. They can add or remove server-level application groups from the collection. When set through the menus, the <b>Edit instance-level information</b> permission also implicitly allows the user to modify version control permissions. To grant all these permissions at a command prompt, you must use the <b>tf.exe Permission</b> command to grant the <b>AdminConfiguration</b> and <b>AdminConnections</b> permissions in addition to GENERIC_WRITE. <b>Note</b> Default server-level groups such as Team Foundation Administrators cannot be removed.
Make requests on behalf of others	Impersonate	Users who have this permission can perform operations on behalf of other users or services. This permission should be assigned only to service accounts.
Trigger Events	TRIGGER_EVENT	Users who have this permission can trigger alert events within Team Foundation Server. This permission should be assigned only to service accounts and members of the Team Foundation Administrators group.
Use full Web Access features	FullAccess	Users who have this permission can use all of the features of Team Web Access.
View instance-level information	GENERIC_READ	Users who have this permission can view server-level group membership and the permissions of those users.

Figure 8: Server permissions (Microsoft 2011, Team Foundation Server Permissions' link).

- Team project collection permissions: These permissions are targeting project collections and all the items under them. According to Microsoft (2011, Team

Foundation Server Permissions' link), there are three groups affected with these permissions: Collection-level groups, Project-level groups and custom groups created by the administrator. The table of the figure 9 describes the team project collection permissions.



Permission Name	Name at Command Line	Description
Administer shelved changes	tf: AdminShelveSets	Users who have this permission can delete shelveSets created by other users.
Administer warehouse	ADMINISTER_WAREHOUSE	Users who have this permission can change warehouse settings by using the ChangeSetting Web method of the WarehouseController.asmx Web service. For example, you could allow users to set the update interval for calculating the OLAP cubes.
Administer workspaces	tf: AdminWorkspaces	Users who have this permission can create workspaces for other users and delete workspaces created by other users.
Alter trace settings	DIAGNOSTIC_TRACE	Users who have this permission can change the trace settings for gathering more detailed diagnostic information about Web services for Team Foundation Server.
Create a workspace	tf: CreateWorkspace	Users who have this permission can create a version control workspace.
Create new projects	CREATE_PROJECTS	Users who have this permission can create projects in the team project collection.  <div style="border: 1px solid #ccc; padding: 5px;"> <p> <b>Note</b> You must not only have this permission but also run Visual Studio as an administrator to successfully complete the Create a New Team Project Wizard. For more information, see <b>Create a Team Project</b>.</p> </div>
Delete team project	Delete	Users who have this permission can delete team projects in the team project collection.  <div style="border: 1px solid #ccc; padding: 5px;"> <p> <b>Important</b> Deleting a team project will delete all data that is associated with the project. You cannot undo the deletion of a team project except by restoring the collection to a point before the project was deleted.</p> </div>

Figure (part 1) 9: Team project collection permissions (Microsoft 2011, Team Foundation Server Permissions' link).

Delete team project collection	DeleteCollection	Users who have this permission can remove a team project collection.
Edit collection-level information	GENERIC_WRITE tf: AdminConfiguration tf: AdminConnections	Users who have this permission can edit collection-level permissions for users and groups in the team project collection. They can add or remove collection-level Team Foundation Server application groups from the collection. When set through the menus, the <b>Edit collection-level information</b> permission also implicitly allows the user to modify version control permissions. To grant all these permissions at a command prompt, you must use the <b>tf.exe Permission</b> command to grant the <b>AdminConfiguration</b> and <b>AdminConnections</b> permissions, in addition to GENERIC_WRITE.  <b>Note</b> Default collection-level groups such as Project Collection Administrators cannot be removed.
Make requests on behalf of others	Impersonate	Users who have this permission can perform operations on behalf of other users or services. This permission should be assigned only to service accounts.
Manage build resources	ManageBuildResources	Users who have this permission can manage the build computers, build agents, and build controllers for the team project collection. These users can also grant or deny the <b>View build resources</b> and <b>Use build resources</b> permissions for other users.
Manage process template	MANAGE_TEMPLATE	Users who have this permission can download, create, edit, and upload process templates to the team project collection.
Manage test controllers	MANAGE_TEST_CONTROLLERS	Users who have this permission can register and de-register test controllers for the team project collection.
Manage work item link types	WORK_ITEM_WRITE	Users who have this permission can add, remove, and change the types of links for work items.
Trigger Events	TRIGGER_EVENT	Users who have this permission can trigger project alert events within the team project collection. This permission should be assigned only to service accounts.
Use build resources	UseBuildResources	Users who have this permission can reserve and allocate build agents. This permission should be assigned only to service accounts for build services.
View build resources	ViewBuildResources	Users who have this permission can view build controllers and build agents that are configured for the collection. To use those resources, you need additional permissions.
View collection-level information	GENERIC_READ	Users who have this permission can view collection-level group membership and the permissions of those users.
View system synchronization information	SYNCHRONIZE_READ	Users who have this permission can call the synchronization application programming interfaces. This permission should be assigned only to service accounts.
View test runs	VIEW_TEST_RESULTS	Users who have this permission can view test plans in this node.

Figure (continued) 9: Team project collection permissions (Microsoft 2011, Team Foundation Server Permissions' link).

- Team project permissions: These permissions are specific to one project’s users and groups (Microsoft 2011, Team Foundation Server Permissions’ link). The table of the figure 10 describes the team project collection permissions.

Permission Name	Name at Command Line	Description
Create test runs	PUBLISH_TEST_RESULTS	Users who have this permission can add and remove test results and add or modify test runs for the team project.
Delete team project	DELETE	Users who have this permission can delete the project for which they have this permission from Team Foundation Server.
Delete test runs	DELETE_TEST_RESULTS	Users who have this permission can delete a scheduled test for this team project.
Edit project-level information	GENERIC_WRITE	Users who have this permission can edit project-level permissions for users and groups on Team Foundation Server.
Manage test configurations	MANAGE_TEST_CONFIGURATIONS	Users who have this permission can create and delete test configurations for this team project.
Manage test environments	MANAGE_TEST_ENVIRONMENTS	Users who have this permission can create and delete test environments for this team project.
View project-level information	GENERIC_READ	Users who have this permission can view project-level group membership and the permissions of those project users.
View test runs	VIEW_TEST_RESULTS	Users who have this permission can view test plans in this node.

Figure 10: Team project permissions (Microsoft 2011, Team Foundation Server Permissions’ link)

## 4. Empirical part – Building the pilot laboratory

This chapter addresses the details of the system implementation phase. In this part are described the detailed steps of the implementation of each project part as well as the motivations for the choices done during this phase.

### 4.1 Physical security requirements

The goal of this phase of the thesis project is to set a list of requirements for the physical security of the servers. Applying those requirements correctly will guarantee a certain level of physical protection to the servers. The physical requirements include the following:

The servers have to be kept in an access-restricted site which only authorized persons have access to. The goal behind this action is to protect the servers from vandalism, stealing, accidental damaging, switching off, malicious software inserting with the help of some removable media like USB keys, DVD, etc...

According to Krutz & Vines (2004, 455 - 456), security requirements for the servers' site include the following considerations:

- Walls: They must be from the floor to the ceiling with an acceptable fire rating
- Windows: Windows are not accepted in a server site except if they are translucent and shatterproof
- Doors: Must resist forcible entry and must have an acceptable fire rating
- Sprinkler system: There must be a fire suppression system which location and type must be known
- Air conditioning: The server site must be cooled and well aerated

- Electrical requirements: The server site should have backup and alternate power sources, e.g. uninterruptible power supply (ups) systems. The entry to the electrical distribution panels must be restricted to authorized personal

HAAGA-HELIA has available several server rooms in its premises at Pasila which fulfil the previously mentioned physical security requirements.

## **4.2 Installing the operating systems on the servers**

This part of the thesis project consists of installing Windows Server 2008 R2 Data Centre edition on the servers. The motivations behind choosing this specific version of Windows Server by the thesis advisor are the following:

- The free of charge availability of Microsoft's software from MSDN-AA and DreamSpark if used for learning purposes only.
- Windows Server 2008 R2 Data Centre is meeting TFS requirements (Microsoft 2013c )
- At the time of elaborating the project idea, Windows Server 2008 R2 was the latest available version of Windows Server
- Data Centre version of Windows Server 2008 R2 offers the widest variety of features amongst all the versions as well as the highest number of processors and amount of RAM memory (Windows Admins 2012)

The detailed installation steps of Windows server 2008 R2 Data centre are partly based on the ones described in the Windows Phone development using Scrum templates for Team Foundation Server thesis document (Stockley 2012, 35 - 37). The servers' operating system installation steps are described in the chapter 1 and subchapters 2.1 -2.5 of the Installation guide (APPENDIX 1).

### **4.3 Installing the operating systems on the workstations**

In this part of the thesis project we will prepare the workstations that will be part of the domain. The chosen operating systems by the customer representative were Windows 7 with Visual Studio 2010, and Windows 8 with Visual Studio 2012.

The motivation behind choosing Windows 7 and Visual Studio 2010 is that they are the software used by default right now at HAAGA-HELIA UAS.

The reason behind installing Windows 8 with Visual Studio 2012 in dual boot along with Windows 7 is that Windows 8 and Visual Studio 2012 are required for Windows Phone 8 and Windows 8 development.

The detailed installation steps of Windows 7 and 8 are described in the chapter 3 of the installation guide (APPENDIX 1).

### **4.4 Installing TFS**

This part of the thesis project will describe the installation of TFS as well as all its needed components. The motivations behind choosing TFS were described in the introduction of this thesis document.

#### **4.4.1 Chosen implementation topology for TFS**

There are several ways to deploy TFS from which the most common are: single-server installation, incorporating TFS to an existing infrastructure, and installing a TFS server farm where TFS components are implemented on different physical servers.

The chosen topology for this thesis project is single-server topology with Active Directory, as it is the one that arguably suits HAAGA-HELIA's needs and requirements (APPENDIX 12). The motivations behind this choice are the following:

- The nature of the environment at HAAGA-HELIA UAS being a learning environment. Thus, high performance is not a critical parameter in the system to be implemented

- Single-server topology has lower costs in matter of hardware
- Due to the nature of the projects being for learning purposes (no business projects), no high availability is required, and thus no redundancy is needed. Nonetheless, students are required to keep back-ups of their projects regularly.
- According to Blankenship, Holliday, Keller & Woodward (2011a, 28), single-server topology implemented in a physical server fitted with a dual core processor of 2.13 GHz and 4 Gb or RAM can support up to 500 users. This number covers largely HAAGA-HELIA’s needs
- According to Microsoft (2013d), single-server topology is the simplest topology available for TFS, therefore learning TFS implementation and operation will be easier with the previously mentioned topology
- Single-server topology makes TFS maintenance easier (Microsoft 2013e)
- Microsoft recommends: “Deploy the simplest topology that meets your business needs” (Microsoft 2013d).

The detailed installation steps of TFS and its needed components on the development server are described in the sub-chapters 2.6 – 2.19 of the installation guide (APPENDIX 1).

#### **4.5 Configuring and securing the software installation on the servers**

This part of the thesis project focuses on securing the TFS environment and setting the different user access levels according to the requirements (APPENDIX 12). A detailed description of the required group structure and user access levels as well as the system’s security implementation steps are described in the chapter 4 of the installation guide (APPENDIX 1).

## 4.6 PowerShell scripts - Purpose and source code

The purpose of creating Power Shell scripts is to support the tasks of the teachers: creating and managing students' accounts, course structures and teams. At the beginning of every course, teachers have to create several users, assign them to the course implementations they belong to and create teams. These tasks take a considerable time when done with user interface especially when there are many course implementations. Power Shell scripts will reduce the needed time to do the previously mentioned tasks considerably and will reduce the risk of errors e.g. typos, missing users, etc...

### 4.6.1 Display course implementation members or team members

The purpose of this script is to display the members (students) belonging to a course implementation or a team. The script takes as input the course implementation name or team name (e.g. **eba-10-01**), to finally display a list of users and groups where are the student number, student name and the object class (user or group).

This script's source code is available in the Appendix 5 (Display course implementation members or teams).

### 4.6.2 Display the students of a specific entry year

The purpose of this script is to display the students belonging to a degree programme at some specific entry year. The script takes as input the degree programme name in the following format (**BIT** or **TIKO**), the year of entry and the term of study e.g. **BIT** students of fall 2010 would be in the format: **bite10s**. The output of the script is a user list of the specified entry year.

This script's source code is available in the Appendix 6 (Display students of a specific entry year).

### 4.6.3 Create a course structure

The purpose of this script is to create a group structure as the one described in the sub-chapter 4.2 of the installation guide (APPENDIX 1). This script takes as input a

comma separated file (CSV) where the course structure is described. The comma separated file is the result of converting the Excel file of the appendix 3 by selecting the format: “CSV(comma delimited)(\* .csv)”.

After providing the CSV file’s path when asked in the script, a group structure that represents the course structure will be created. As an output of the successful execution of the script, the user will get a message confirming that the structure was created successfully.

This script’s source code is available in the appendix 7(Create a course structure).

#### **4.6.3.1 Create users and assign them to teams**

The purpose of this script is to create user accounts for the students, add each student to a team, and finally create a file containing the user accounts and passwords. This script takes as input a comma separated file (CSV) that has the student information and the team the student belongs to. The comma separated file is the result of converting the HAAGA-HELIA’s students’ list Excel file model of the appendix 4 by selecting the format: “CSV(comma delimited)(\* .csv)”.

After providing the CSV file’s path and the degree program to which the student belongs to when asked in the script, the students’ accounts will be created and placed to the organizational unit where they belong to (**BITE** or **TIKO**). A successful execution of the script will produce a message confirming that the structure was created successfully. A CSV file called “**UserPasswList.csv**” containing the created users Ids and passwords will be created too and placed in the Shared folder described in the section 5 of the installation guide (APPENDIX 1). The path of the shared folder is the following:  
**C:\ScriptShared\output\UserPass-wList.csv**

This script’s source code is available in the appendix 8 (Create users and assign them to teams).

#### **4.6.3.2 Removing a course structure**

The purpose of this script is to remove a group and all its member groups. The script takes the name of the group to be removed as an input, and then all the group members will be found and finally removed including the input group itself. The output of this script is a message confirming that the removal was successful.

This script's source code is available in the appendix 9 (Removing a course structure).

### **4.7 Testing the pilot laboratory**

This part of the project consists of assessing the system's security and user access privileges. To reach this goal, first we will perform a system security scan with Microsoft Baseline Security Analyse (MBSA). After scanning the system for security issues with MBSA, we will run a list of test scenarios to check that the users have the right access privileges according to the requirements (APPENDIX 12).

Other objectives of the test scenarios is to provide step by step instructions on how to create: team project collections, team projects, assign teams to team projects, and grant to the teams the right privileges to perform the specific tasks in the team projects according to the requirements (APPENDIX 12). The test scenarios include also the steps to create a new teacher and grant him the needed access rights.

#### **4.7.1 Test methodology and results validation**

The first test will be based on ready scenarios performed by one person to assess the user rights and privileges. Every scenario has an Id, name, purpose, input data, before the test (refers to the preparations before performing the test scenario), steps, expected result, and test result. The person performing the tests will do every scenario and compare the obtained results to the expected results.

A test is considered passed if the test results obtained are identical to the expected results.

Microsoft baseline security analyser will be used to assess the system security (looking for accounts without password, checking for latest system patches...)

#### **4.7.2 Test prerequisites and preparation**

In order to perform the tests we will need to have the domain controller, the development server and a workstation belonging to the domain turned on. The following list describes the data needed to perform the testing part:

Domain controller:

- Host address: `NetDevDc.lumitulib.com`
- IP address: 172.28.150.1

Development server:

- Host address: `DevServer.lumitulib.com/tfs.lumitulib.com`
- IP address: 172.28.150.2

Workstation:

- Computer name: `PA01K209W07`
- IP address: 172.28.150.3

Note: the workstation has to be booted with windows 8.

Before starting the tests, we will need to have the following accounts:

- A user account with administrator privileges
- A user account belonging to the TFSUsers\_OU\_Admins group (teacher account)
- Comma separated file with the students data of the appendix 4 (Students' list)

- Comma separated file with the group structure of the appendix 3 (Course structure)
- Power Shell scripts for creating the users and groups

Power shell scripts folder's mapping path is: \\lunitulib.com\ScriptShared

### 4.7.3 Creating a teacher user account

Performing the tests requires having a teacher account available. The following steps describe the creation of a teacher account:

In the domain controller log on as administrator: Start > administrative tools > server manager > roles > active directory domain services > active directory users and computer > domain name (e.g. lunitulib.com) > TFSUsers > right click Teachers OU > new > user > First name: test, last name: teacher, user log on name: teacher > teacher > password: \*\*\*\*\* > check password never expires and uncheck user must change password at next logon > next > finish.

Now we need to add the account created to the group TFSUsers\_OU\_Admins: Right click the user we created in the above steps > properties > member of > add... > advanced... > find now > select TFSUsers\_OU\_Admins > ok > ok > ok.

### 4.7.4 Performing the test scenarios

In this part, the test scenarios will be done to check that the system configuration was done successfully and according to the requirements (APPENDIX 12).

#### 4.7.4.1 Scenarios performed with a teacher account

The test scenarios performed with a teacher account are listed in the appendix 10 (Scenarios performed with a teacher account).

#### **4.7.4.2 Scenarios performed with a student account**

The test scenarios performed with a student account are listed in the appendix 11 (Scenarios performed with a teacher account).

#### **4.7.5 Assessing the system with MBSA**

In this part of the thesis we will scan the computers (servers and workstations) with a security assessment tool called Microsoft Security Baseline analyser.

##### **4.7.5.1 MBSA installation steps**

Download the latest version of MBSA for 64-bit systems from Microsoft's site and save it to the desktop: <http://www.microsoft.com/en-us/download/details.aspx?id=7558>

Run the downloaded file > press next > accept the terms of license > keep the default installation location and press next > install > ok.

##### **4.7.5.2 Scanning a computer with MBSA**


Launch MBSA from the shortcut in the desktop > select Scan a computer > in the field Computer name, enter the name of the computer in the format (domain name\computer name), or in the IP address field enter the computer's IP address. In the Security report name field, leave the default values. Finally in the options field, select the following: Check for Windows administrative vulnerabilities, Check for weak passwords, Check for IIS administrative vulnerabilities, Check for SQL administrative vulnerabilities and Check for security updates > press Start scan.

The procedure described above should be done to every computer, especially to the servers.

The picture of the figure 11 shows a report generated by MBSA.

Microsoft Baseline Security Analyzer 2.2


Microsoft  
Baseline Security Analyzer

**Security assessment:**  
 **Potential Risk (One or more non-critical checks failed.)**




---

**Computer name:** LUMITULIB\NETDEVDC  
**IP address:** 172.28.150.1  
**Security report name:** LUMITULIB - NETDEVDC (7.5.2013 9-47)  
**Scan date:** 7.5.2013 9:47  
**Scanned with MBSA version:** 2.2.2170.0  
**Catalog synchronization date:**  
**Security update catalog:** Microsoft Update

---

Sort Order:  

**Security Update Scan Results**

Score	Issue	Result
	SQL Server Security Updates	No security updates are missing. <a href="#">What was scanned</a> <a href="#">Result details</a>
	Silverlight Security Updates	No security updates are missing. <a href="#">What was scanned</a> <a href="#">Result details</a>
	Windows Security Updates	No security updates are missing. <a href="#">What was scanned</a> <a href="#">Result details</a>

**Windows Scan Results**

**Administrative Vulnerabilities**











Score	Issue	Result
	Password Expiration	Some user accounts (9 of 27) have non-expiring passwords. <a href="#">What was scanned</a> <a href="#">Result details</a> <a href="#">How to correct this</a>
	Incomplete Updates	No incomplete software update installations were found. <a href="#">What was scanned</a>
	Windows Firewall	Windows Firewall is enabled and has exceptions configured. Windows Firewall is enabled on all network connections. <a href="#">What was scanned</a> <a href="#">Result details</a> <a href="#">How to correct this</a>
	Automatic Updates	Updates are automatically downloaded and installed on this computer. <a href="#">What was scanned</a>
	File System	All hard drives (1) are using the NTFS file system. <a href="#">What was scanned</a> <a href="#">Result details</a>
	Autologon	Autologon is not configured on this computer. <a href="#">What was scanned</a>
	Guest Account	The Guest account is disabled on this computer. <a href="#">What was scanned</a>
	Restrict Anonymous	Computer is properly restricting anonymous access. <a href="#">What was scanned</a>
	Administrators	No more than 2 Administrators were found on this computer. <a href="#">What was scanned</a> <a href="#">Result details</a>
	Local Account Password Test	Password checks are not performed on a domain controller. <a href="#">What was scanned</a>

Figure 11: MBSA generated report after scanning the domain controller (Beginning of the figure)

Microsoft Baseline Security Analyzer 2.2

Microsoft Baseline Security Analyzer

- Restrict Anonymous: Computer is properly restricting anonymous access. [What was scanned](#)
- Administrators: No more than 2 Administrators were found on this computer. [What was scanned](#) [Result details](#)
- Local Account Password Test: Password checks are not performed on a domain controller. [What was scanned](#)

**Additional System Information**

Score	Issue	Result
	Auditing	Neither Logon Success nor Logon Failure auditing are enabled. Enable auditing and turn on auditing <a href="#">What was scanned</a> <a href="#">How to correct this</a>
	Services	Some potentially unnecessary services are installed. <a href="#">What was scanned</a> <a href="#">Result details</a> <a href="#">How to correct this</a>
	Shares	6 share(s) are present on your computer. <a href="#">What was scanned</a> <a href="#">Result details</a> <a href="#">How to correct this</a>
	Windows Version	Computer is running Microsoft Windows 7. <a href="#">What was scanned</a>

**Internet Information Services (IIS) Scan Results**

**Administrative Vulnerabilities**

Score	Issue	Result
	IIS Lockdown Tool	The IIS Lockdown tool was developed for IIS 4.0, 5.0, and 5.1, and is not needed for new Windows. <a href="#">What was scanned</a>
	Sample Applications	IIS sample applications are not installed. <a href="#">What was scanned</a>
	IISAdmin Virtual Directory	IISADMPWD virtual directory is not present. <a href="#">What was scanned</a>
	Parent Paths	Parent paths are not enabled. <a href="#">What was scanned</a>
	MSADC and Scripts Virtual Directories	The MSADC and Scripts virtual directories are not present. <a href="#">What was scanned</a>

**Additional System Information**

Score	Issue	Result
	Domain Controller Test	IIS is running on a primary or backup domain controller. <a href="#">What was scanned</a> <a href="#">How to correct this</a>
	IIS Logging Enabled	All web and FTP sites are using the recommended logging options. <a href="#">What was scanned</a>

**SQL Server Scan Results**

Score	Issue	Result
	SQL Server/MSDE Status	SQL Server and/or MSDE is not installed on this computer.

[Print this report](#) [Copy to clipboard](#) [Previous security report](#)

Figure 11 (Continued): MBSA generated report after scanning the domain controller

Based on the results of the report, the needed Actions can be taken to set the security to the level recommended by Microsoft. MBSA will list in the report the issues encountered, the severity of the issues and the way how to fix them.

#### **4.8 Publishing the system to the internet**

This part of the thesis consists of publishing the successfully tested system to the internet. Publishing the system requires a public IP address provided by an internet service provider (ISP). Moreover, the domain name we are using (e.g. lumitulib.com) has to be available in the “.com” domain. If these conditions are met, i.e. available public IP address and available domain name, in addition to meeting HAAGA-HELIA’s security principles, then publishing the server to the internet is possible.

For some administrative and security constraints at HAAGA-HELIA, publishing the server to the internet will be handled by HAAGA-HELIA’s IT department staff.

## 5. Evaluation

This thesis project's goal was to implement, secure and deploy a learning environment with TFS to be used by HAAGA-HELIA's IT students. One of the main outputs of the thesis was a detailed installation guide where the different elements needed for the installation and the steps to be taken are described. The goals aimed at in this thesis were reached as the system was implemented according to the requirements (APPENDIX 12), tested and the installation guide is available.

While gathering information for this thesis project, I noticed that there was a huge amount of information related to TFS and its implementation, however the information was scattered around and getting an insight of the topic was not straightforward. I believe that this thesis paper will give to its reader a good understanding of TFS and its implementation's different steps as well as a general insight of the topic and this from one single document: this thesis paper. The topology model used (single server topology); which is the easiest and simplest implementation model as stated in the sub-chapter 3.3.2.1 of this thesis paper; would make the understanding of TFS implementation and operation easier for the thesis reader.

The research method followed in this thesis project (literature review) was the adequate choice since I had no previous knowledge about the topic. By reading the relevant literature related to the thesis topic I was able to understand how TFS works and thus implement the system according to the requirements

In what concerns the project planning part, the gathered project management skills in some previously done BIT courses; where I learned that planning is a very important part of an IT project; helped me to plan and make a good scheduling for the thesis project. The planning part took a considerable time (more than one week), but the resulting project plan was of quality. Consequently, the project went smoothly and within the planned timeline.

The method used in the system implementation part of this thesis helped me to stay in the planned schedule. Indeed, the requirements list (APPENDIX 12) ordered by im-

portance helped me to focus on the primary tasks and if time was left the secondary tasks were addressed. As a result of this method, all the important features were implemented and only the automated deployment to IIS server with Visual Studio was left out from the thesis project.

Since this thesis project's scope is quite broad and the time for performing the thesis work is limited, the IIS server configuring for automated project deployment with Visual Studio was left out. Nonetheless, this part did not affect the goals of this thesis. According to the requirements (APPENDIX 12), the automated project deployment in IIS server would bring extra functionality which is not of high importance for this project at the time being.

This thesis project gave me a new insight about IT and I am very satisfied with the knowledge I acquired. Now in addition to the software development skills I gathered all along the degree program (web developer path), I can afford to say that I have also systems and system security knowledge and thus I can apply to jobs where such skills are required.

## 6. Further development

While doing this thesis project, the thesis author became aware that there are some features that would bring additional functionality to the system. In addition to the parts that could not be implemented due to the time limitations, the following features could be considered for further research:

- Automating project deployment to the IIS server with Visual Studio: Implementing this part will allow the students to deploy, access and test their web applications in a real environment. This is beneficial from an educational point of view, as the students will have the chance to learn the deployment process of a web application.
- Automating TFS teacher's routine tasks with Scripts: Creating project collections and team projects for every team is an exhaustive task for the teachers. Automating the previously mentioned tasks would ease the teacher's tasks, and bring additional value to the system's usability from the teacher's point of view.
- Implementing a build server for every course: In TFS, every team project collection has one dedicated build controller which cannot have more than one collection attached to it. According to Microsoft (2013h), there can't be more than one build server instance per physical server. Moreover, on a build server we can run only one controller, and thus we would need for every team project collection one build server. This part is based on the HAAGA-HELIA's needs where every course is represented with a team project collection and thus, every course needs a dedicated build server.

## 7. Summary

This thesis project started with the will of HAAGA-HELIA UAS to provide to its IT students a learning environment with an application lifecycle management tool to enhance their professional skills. As mentioned in the introduction, Team Foundation Server was the chosen tool by the school representative for its support for Scrum framework, compatibility with Visual Studio, free of charge availability as long as used for learning purposes, for being considered one of the most complete application management tools available on the market, and finally because Business Information Technology degree program focuses on Microsoft technologies.

The implementation of the system was based on a list of requirements (APPENDIX 12) ordered by importance in order to fit the thesis project in the planned schedule and get a functional system with the main features listed in the requirements implemented. The system's security assessment was based on a test plan where some specific test scenarios were performed to test the users' access and privileges. Computers' security assessment was done also by means of a security-scanning tool called Microsoft Baseline Security Analyser. Finally when the system was successfully tested, it was ready to be deployed to the internet by the HAAGA-HELIA's IT department staff.

The goals set for this thesis project were reached since the system was successfully implemented, tested and the system installation guide intended to be used by HAAGA-HELIA's IT students for learning Team Foundation Server was provided. Moreover, the thesis project was carried out according to what was agreed in the project plan and within the planned schedule.

## 8. References

Bansode, P., Jones, K., Mackman, A., Meier, J. D., Taylor, J. 2007. Team Development With Visual Studio Team Foundation Server. URL: <http://msdn.microsoft.com/en-us/library/bb668952.aspx>. Accessed: 29 Apr 2013.

Bansode, P., Jones, K., Meier, J. D., Taylor, J. & Mackman A. 2013. Team Development with Visual Studio Team Foundation Server. URL: <http://tfsguide.codeplex.com/wikipage?title=Chapter%2016%20-%20Installation%20and%20Deployment>. Accessed: 1 May 2013.

Blankenship, E., Holliday, G., Keller, B. & Woodward, M. 2011. Professional Team Foundation Server 2010. Wiley Publishing Inc, Indianapolis.

Bryant University 2009. Server Hardening Policy. URL: <http://web.bryant.edu/~commtech/downloads/ServerHardening.pdf>. Accessed: 9 May 2013

Duggan, J & Murphy, E 2012. Magic quadrant for application life cycle management. URL: <http://www.gartner.com/technology/reprints.do?id=1-1ASCXON&ct=120606&st=sb>. Accessed: 24 Apr 2013.

Jansen, W., Scarfone, K. & Tracy, M. 2008. Guide to general server security. URL: <http://csrc.nist.gov/publications/nistpubs/800-123/SP800-123.pdf>. Accessed: 9 May 2013

Krutz, R. L. & Vines, R. D. 2004. The CISSp prep guide second edition. Wiley publishing, Inc. Indianapolis, Indiana.

Long, M. 2008. Microsoft Windows Server 2008. URL: <http://www.vtc.com/products/MicrosoftWindowsServer2008/CourseOverview/75466>. Accessed: 7 May 2013.

- Microsoft 2009a. Browse Windows Server technologies. URL:  
<http://technet.microsoft.com/en-US/library/dd283012>. Accessed: 8 May 2013.
- Microsoft 2009b. Active Directory Domain Services. URL:  
<http://msdn.microsoft.com/en-us/library/windows/desktop/aa362244%28v=vs.85%29.aspx>. Accessed: 8 May 2013.
- Microsoft 2009c. Web Server (IIS). URL: <http://technet.microsoft.com/en-US/library/cc753433>. Accessed: 8 May 2013
- Microsoft 2010. Domain Controller roles. URL: <http://technet.microsoft.com/en-us/library/cc786438%28v=ws.10%29.aspx>. Accessed: 8 May 2013.
- Microsoft 2011. TFS Admin guide. URL: <http://www.microsoft.com/en-us/download/details.aspx?id=8175>. Accessed: 2 May 2013.
- Microsoft 2013a. Microsoft Baseline Security Analyzer 2.2 (for IT Professionals). URL:  
<http://www.microsoft.com/en-us/download/details.aspx?id=7558>. Accessed: 26 Apr 2013.
- Microsoft 2013b. About Active Directory Domain Services. URL:  
<http://msdn.microsoft.com/en-us/library/windows/desktop/aa772142%28v=vs.85%29.aspx>. Accessed 8 May 2013.
- Microsoft 2013c. System requirements for Team Foundation Server. URL:  
<http://msdn.microsoft.com/en-us/library/dd578592%28v=vs.100%29.aspx>. Accessed: 5 May 2013.
- Microsoft 2013d. MSDN, Team Foundation Server architecture. URL:  
<http://msdn.microsoft.com/en-us/library/bb668952.aspx>. Accessed: 26 Apr 2013.
- Microsoft 2013e. A simple Team Foundation Server topology. URL:  
<http://msdn.microsoft.com/en-us/library/ms400693%28v=vs.90%29.aspx>. Accessed: 1 May 2013.

- Microsoft 2013g. Windows 8 system requirements. URL:  
<http://windows.microsoft.com/en-us/windows-8/system-requirements>. Accessed: 10 May 2013
- Microsoft 2013h. Deploy and configure a build server. URL:  
<http://msdn.microsoft.com/en-us/library/vstudio/ms181712.aspx>. Accessed: 13 May 2013
- O'neill, J., Rosen, J. & Stanek, W., R. 2009. Microsoft PowerShell, VBScript and JScript. . Wiley publishing, Inc. Indianapolis, Indiana.
- Rains, T. 2012. Microsoft free security tools – Microsoft Baseline Security Analyzer. URL: <http://blogs.technet.com/b/security/archive/2012/10/22/microsoft-free-security-tools-microsoft-baseline-security-analyzer.aspx>. Accessed: 6 May 2012
- Server Hardening. URL: <http://www.serverhardening.com/>. Accessed: 9 May 2013
- Shuler, R. 2005. How does internet work?. URL:  
[http://www.theshulers.com/whitepapers/internet\\_whitepaper/index.html#dns](http://www.theshulers.com/whitepapers/internet_whitepaper/index.html#dns). Accessed: 8 May 2013
- Stockley, P. 2012, Windows Phone development using Scrum templates for Team Foundation Server. URL:  
[https://publications.theseus.fi/bitstream/handle/10024/48387/Pete\\_Stockley\\_THESIS.pdf?sequence=1](https://publications.theseus.fi/bitstream/handle/10024/48387/Pete_Stockley_THESIS.pdf?sequence=1), Accessed: 15 February 2013
- Webopedia, 2013. Windows Server. URL:  
[http://www.webopedia.com/TERM/W/Windows\\_Server.html](http://www.webopedia.com/TERM/W/Windows_Server.html). Accessed: 7 May 2013.
- Windows Admins 2013. Windows Server 2008 R2 Edition comparison. URL:  
<http://winadmins.wordpress.com/2012/01/18/windows-server-2008-r2-edition-comparison/> . Accessed: 5 May 2013.

Zimmermann, T & Bird, C 2012. Collaborative Software Development in Ten Years: Diversity, Tools, and Remix Culture. URL: <http://thomas-zimmermann.com/publications/list/Short>. Accessed: 24 Apr 2013.

## 9. Appendices

### Appendix 1: Installation guide

## Appendix 1

### Installation guide

Tarek Sadik

Bachelor thesis

DP in Business

Information Technology

January 2012



## Table of contents

1	Setting up the domain controller .....	1
1.1	Installing and setting Windows Server 2008 R2 .....	1
1.2	Setting up Active Directory .....	3
1.3	Setting up the DNS server .....	3
2	Setting up the development server .....	5
2.1	Installing and setting Windows Server 2008 R2 .....	5
2.2	Setting up Internet Information Services (IIS) webserver on the development server.....	7
2.3	Joining the development server to the domain controller.....	7
2.4	Moving the development server from “Computers” container to “Domain Controllers” OU.....	7
2.5	Testing the DNS .....	8
2.6	Setting up TFS on the development server .....	8
2.7	Setting AD service account access levels and adding them to the relevant GPO.....	10
2.8	Disabling Windows Firewall and user account control settings (UAC) and performing the updates: .....	11
2.9	Installing IIS on the development server.....	11
2.10	Installing SQL Server 2008 R2 on the development server .....	11
2.11	Configuring the Report Server .....	13
2.12	Installing SharePoint services 3.0 SP3.....	14
2.13	Verifying WSS installation.....	14
2.14	Creating a web application and a site collection .....	15
2.15	Installing and configuring TFS.....	16
2.16	Setting up the build server .....	17
2.17	Setting a build controller and build agents for a specific team project collection .....	19
3	Setting the client(s).....	20
3.1	Windows 7 .....	20
3.2	Installing Windows 8 along with Windows 7 (dual boot) .....	22

3.3	Installing Windows 8 .....	23
4	Securing Active Directory .....	27
4.1	Creating the business structure .....	27
4.2	Creating the custom AD groups .....	29
4.3	Creating the teachers group .....	32
4.4	Enabling and setting Remote desktop.....	33
4.5	Granting administrative privileges of TFS and its components to the teachers group (TFSUsers_OU_Admins) .....	35
4.6	Granting TFS users reading rights in the Reporting Server .....	38
4.7	Security settings using GPOs .....	38
	In the domain controller do the following: .....	38
5	Creating a shared folder for the scripts.....	39
5.1	Purpose.....	39
5.2	Creating and setting the shared folder for PowerShell scripts .....	39

# 1. Setting up the domain controller

The first chapter's content is partly based on the installation steps described in the Windows Phone development using Scrum templates for Team Foundation Server thesis document (Stockley 2012, 35, 36).

## 1.1 Installing and setting Windows Server 2008 R2

1. Disconnect the computer from network.
2. Turn on the computer and press ESC for the boot menu.
3. Put Windows Server 2008 R2 DVD into the DVD tray.
4. Choose boot menu (F9) > choose the boot device: HP DVD-RAM GH80N > Press any key to start the setting up menu > Choose Windows setup [EMS enabled].
5. Choose the language to install (English), time format and currency (Finnish), and the keyboard input method (Finnish). Press NEXT and then finally press INSTALL NOW.
6. Choose the Windows Server 2008 R2 Data Center (Full installation) as it offer the maximum features in software and hardware (Future expansion), in addition to the entire user interface. After selecting press NEXT.
7. Accept the license terms and press NEXT > Choose Custom installation to install everything from scratch.
  - a. Choose Drive options (advanced) and delete all the existing partitions. After Partitioning the HDD and selecting the partition where to install Windows Server, press NEXT to start the installation.
8. After the installation has completed and the computer restarted, provide a new administrator password and log in by pressing OK.
9. Now we will proceed with the computer naming:
  - a. Renaming the computer, it can be done as follows: start > control panel > system > Change settings > Change... > Write the new name in the text box under computer name > press OK.
  - b. Restart your pc when prompted.
10. Set Windows firewall (both home or work (private) networks and public networks) as active and set the notification state to notify when Windows Firewall blocks a new program: Start > Control panel > Windows Firewall > Change notification settings > turn on Windows Firewall (for both) and select "Notify me when Windows Firewall blocks a new program" > press ok.

11. Change folder and file settings: Open file explorer > click Organize > Folder and search options > View > Select Show hidden files, unselect Hide extensions for known file types, unselect hide protected OS files.
12. You will need another computer to download the drivers, and a clean memory stick where to save them for later installing on the server.
13. Download the network drivers from Intel's webpages and install them:  
<http://downloadcenter.intel.com/confirm.aspx?httpDown=http://downloadmirror.intel.com/18725/eng/PROWinx64.exe&lang=eng&Dwnldid=18725>
14. After the installation, restart the computer, connect the computer to the network, and make sure that your drivers work by checking that you can access the internet and by checking the network adapter device status in the device manager
15. Perform computer updates: Start > All programs > Windows update > (Select the updates to be installed if needed)> install updates.
  - a. Select don't install Internet Explorer 9 when prompted.
  - b. After finishing the updates, restart the computer > go back to the update window and check if there are more updates to be installed > in case of yes, select them and press install updates > restart your computer if needed.
16. Download graphic card drivers and audio drivers: Search in Google with computer model. E.g. "HP Compaq 8200 Elite Convertible minitower drivers", choose ONLY the results that have as source the device or the computer brand's website (trusted sources).
  - a. Choose your device with the right OS (Remember that sometimes a driver can be used also for many OS: check the possibilities you have).
  - b. Run the graphic card drivers and unselect "Automatically run WinSAT and enable Windows Aero".
  - c. Run the audio drivers.
  - d. Finally restart the computer to apply the changes.
17. Start the Audio service: Start > Control panel > Administrative tools > Services > Right click Windows Audio > properties > Set start up type to Automatic and click ok > right click Windows Audio again and select start to start the service.
18. Install Firefox: browse [www.mozilla.org](http://www.mozilla.org) and press download Firefox. (Select standard installation and select Firefox to be the default browser)
  - a. Install Silverlight plugin for Firefox: Press Firefox > Add-ons > plugins > type in the search bar Microsoft Silverlight > choose install plugin (Enable automatic updates for Silverlight)
19. Configure the IP Settings:

- a. Start > Network > Network Sharing center > Change adapter settings > Right click local area connection > Properties
- b. Select TCP/IPv4 > Click properties > Check use the following IP address > Enter the domain controller's IP address, subnet mask, default gateway and preferred DNS server E.g. Hirmuinen's IP address. (Check the table of values of the Appendix 2 for this information)

## 1.2 Setting up Active Directory

The setting up of the AD DS (Active Directory Domain Services) can be done with the Add Roles wizard. The Add Roles wizard installs the files needed for installing and configuring AD DS on a server, but does not start the actual AD DS installation. In order to start the AD DS installation we will need to run the "dcpromo" application:

1. Click Start > type dcpromo in the search bar > click enter to run the dcpromo.exe > click next > click next.
2. Choose create a new domain in a new forest
3. Give the fully qualified domain name (FQDN) to your domain > press next.
4. Select a forest functional level (functional levels determine the features of AD DS that are enabled in a domain or forest, and they restrict which Windows Server OSs can run on domain controllers in the domain or forest) > press next
5. Make sure DNS server option is checked (as we want to install on our domain controller the DNS server) > press next > yes > press next (We'll keep the default locations provided by the AD DS installation wizard as we have only one hard disk, and server performance is not taken into consideration this time)
6. Set a Directory services Restore Mode Administrator Password (It has to be different than the password for the domain administrator account. The DSRM password I used for logon to a domain when AD DS is not running either because it is stopped or because the domain controller has been started in Directory Service Restore Mode DSRM) > press next
7. Review your selection and check that everything is correct > click next
8. When the AD DS installation is ready > click finish and reboot.

## 1.3 Setting up the DNS server

We have been already adding a DNS server role to our domain controller at the time of setting up AD DS with dcpromo. When adding a DNS server, a

forward lookup zone is created automatically having as a name our domain controller. Now we will proceed with more configuring in the DNS server, by adding a host resource record to the forward lookup zone (and aliases), and creating the reverse lookup zone where DNS resolves IP addresses to domain names with its pointers resource records:

1. Open the DNS Manager: Start > Administrative tools > DNS
2. Expand the applicable DNS server name
3. Creating a reverse lookup zone:
  - a. Right click Reverse Lookup Zones > select new zone > press next
  - b. Select: primary zone and check that Store the zone in Active Directory is checked > press next
  - c. Select: To all DNS servers running on domain controllers in this domain > press next
  - d. Select: IPv4 Reverse lookup zone > press next
  - e. Select: Network ID and enter the network Identifier of your network (remark that the network ID is obtained from the domain controller's IP address and the subnet mask: check the table in the appendix 1 to view an example on how to obtain the network ID) > press next
  - f. Select: Do not allow dynamic updates > press next (This can be changed later on if needed as we have our zone integrated in the AD so we can have dynamic updates from DNS clients securely by using AD security features)
4. Creating a host resource record with and an alias to the forward lookup zone:
  - a. Expand forward lookup zone folder > right click your FQDN folder > select New Host (A or AAAA)...
  - b. Enter the host name (e.g. www) and its associated IP address and make sure that create associated pointer is checked because like this the system will create for us the reverse lookup resource record for our host which is a pointer that points to the host name > press add host > press ok
5. Test the www URL to ensure the access to the website works.
  - a. Test the URL in the web browser: Open a session of Internet Explorer e.g. and enter in the address field www.lumitilib.com. The IIS server default page should be displayed.

## 2. Setting up the development server

### 2.1 Installing and setting Windows Server 2008 R2

The subchapter 2.1 content is partly based on the installation steps described in the Windows Phone development using Scrum templates for Team Foundation Server thesis document (Stockley 2012, 37)

1. Disconnect the computer from network.
2. Turn on the computer and press ESC for the boot menu.
3. Put Windows Server 2008 R2 DVD into the DVD tray.
4. Choose boot menu (F9) > choose the boot device: HP DVD-RAM GH80N > Press any key to start the setting up menu > Choose Windows setup [EMS enabled].
5. Choose the language to install (English), time format and currency (Finnish), and the keyboard input method (Finnish). Press NEXT and then finally press INSTALL NOW.
6. Choose the Windows Server 2008 R2 Data Center (Full installation) as it offers the maximum features in software and hardware (Future expansion), in addition to the entire user interface. After selecting press NEXT.
7. Accept the license terms and press NEXT > Choose Custom installation to install everything from scratch.
  - a. Choose Drive options (advanced) and delete all the existing partitions.
8. After Partitioning the HDD and selecting the partition where to install Windows Server, press NEXT to start the installation.
9. After installation has completed and the computer restarted, provide a new administrator password and log in by pressing OK.
10. Now we will proceed with the computer naming:
  - a. Renaming the computer, it can be done as follows: start > control panel > system > Change settings > Change... > Write the new name in the text box under computer name > press OK.
  - b. Restart your pc when prompted.
11. Set Windows firewall (both home or work (private) networks and public networks) as active and set the notification state to notify when Windows Firewall blocks a new program: Start > Control panel > Windows Firewall > Change notification settings > turn on Windows Firewall (for both) and select "Notify me when Windows Firewall blocks a new program" > press ok.

12. Change folder and file settings: Open file explorer > click Organize > Folder and search options > View > Select Show hidden files, unselect Hide extensions for known file types, unselect hide protected OS files.
13. You will need another computer to download the drivers, and a clean memory stick where to save them for later installing on the server.
14. Download the network drivers from Intel's web pages and install them:  
<http://downloadcenter.intel.com/confirm.aspx?httpDown=http://downloadcenter.intel.com/18725/eng/PROWinx64.exe&lang=eng&Dwnldid=18725>
15. After the installation, restart the computer, connect the computer to the network, and make sure that your drivers work by checking that you can access the internet and by checking the network adapter device status in the device manager.
16. Perform computer updates: Start > All programs > Windows update > (Select the updates to be installed if needed)> install updates.
  - a. Select don't install Internet Explorer 9 when prompted.
  - b. After finishing the updates, restart the computer > go back to the update window and check if there are more updates to be installed > in case of yes, select them and press install updates > restart your computer if needed.
17. Download graphic card drivers and audio drivers: Search in Google with computer model. E.g. "HP Compaq 8200 Elite Convertible minitower drivers", choose ONLY the results that have as source the device or the computer brand's website (trusted sources).
  - a. Choose your device with the right OS (Remember that sometimes a driver can be used also for many OS: check the possibilities you have).
  - b. Run the graphic card drivers and unselect "Automatically run WinSAT and enable Windows Aero".
  - c. Run the audio drivers.
  - d. Finally restart the computer to apply the changes.
18. Start the Audio service: Start > Control panel > Administrative tools > Services > Right click Windows Audio > properties > Set start up type to Automatic and click ok > right click Windows Audio again and select start to start the service.
19. Install Firefox: browse [www.mozilla.org](http://www.mozilla.org) and press download Firefox. (Select standard installation and select Firefox to be the default browser)
  - a. Install Silverlight plugin for Firefox: Press Firefox > Add-ons > plugins > type in the search bar Microsoft Silverlight > choose install plugin (Enable automatic updates for Silverlight)

20. Configure the IP Settings:

- a. Start > Network > Network Sharing Center > Change adapter settings > Right click local area connection > Properties
- b. Select TCP/IPv4 > Click properties > Check use the following IP address > Enter the development server's IP address, subnet mask, default gateway and preferred DNS server that is your domain controller. (Check the table of values of the Appendix 2 for this information)
- c. Finally check that internet connection still works.

## 2.2 Setting up Internet Information Services (IIS) webserver on the development server

1. Start > Administrative tools > Server manager > right click roles > Add roles
2. click next to skip the Before you begin page > Select Web Server (IIS) > next > next
3. In the role services view you will select all the features available except: Dynamic Content Compression and IIS Hostable Web Core.
4. Click install and at the end of the installation press finish.

## 2.3 Joining the development server to the domain controller

1. Start > Control panel > System > change settings > click change... > in the section: member of, enter the FQDN of the forest root domain > press ok
2. When prompted, enter the User ID and Password of the domain controller's administrator
3. When prompted, restart the PC.

## 2.4 Moving the development server from "Computers" container to "Domain Controllers" OU

Start > administrative tools > server manager > expand: roles > expand active directory domain services > expand active directory users and computers > expand the domain name e.g. lumitilib.com > computers > right click the development server and select move... > select domain controllers OU > ok

## 2.5 Testing the DNS

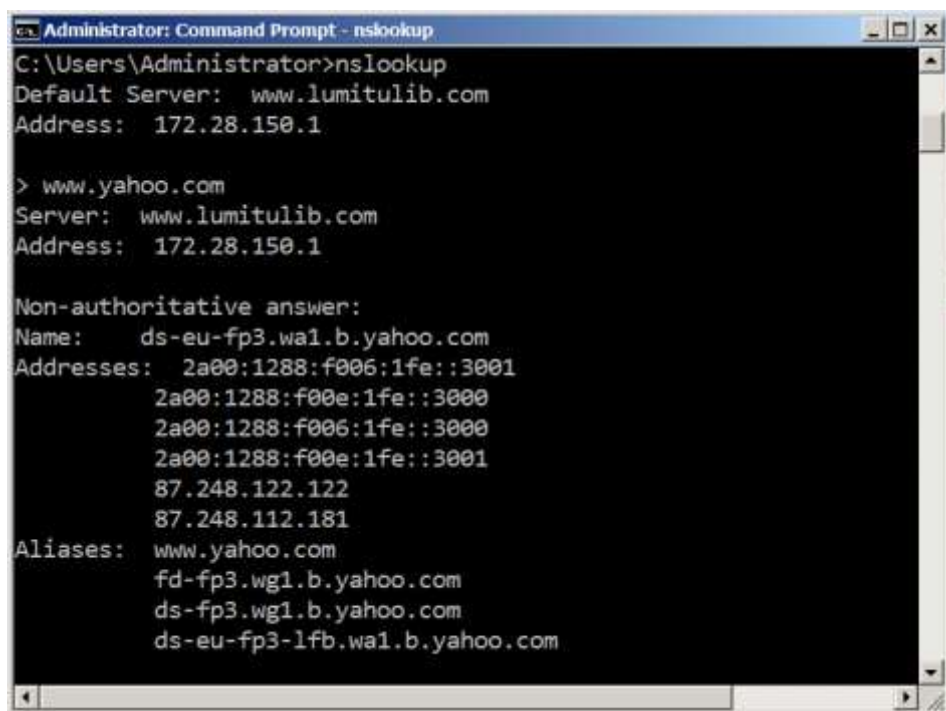
After setting up our DNS server in the domain controller and the IIS server in the development server, we will proceed with testing them:

1. Testing the DNS server from the development server

Open a Cmd session:

- a. Execute the following command to flush the DNS cache: `ipconfig/flushdns`
- b. Test the DNS server to make sure it is resolving:

Check that the DNS resolution is working e.g. execute the command `nslookup www.yahoo.com` (the picture 1 describes the results of a successful execution)



```
Administrator: Command Prompt - nslookup
C:\Users\Administrator>nslookup
Default Server:  www.lumitulib.com
Address:  172.28.150.1

> www.yahoo.com
Server:  www.lumitulib.com
Address:  172.28.150.1

Non-authoritative answer:
Name:    ds-eu-fp3.wa1.b.yahoo.com
Addresses:  2a00:1288:f006:1fe::3001
            2a00:1288:f00e:1fe::3000
            2a00:1288:f006:1fe::3000
            2a00:1288:f00e:1fe::3001
            87.248.122.122
            87.248.112.181
Aliases:  www.yahoo.com
          fd-fp3.wg1.b.yahoo.com
          ds-fp3.wg1.b.yahoo.com
          ds-eu-fp3-lfb.wa1.b.yahoo.com
```

Picture 1: nslookup command execution

2. IIS server: Test that you can access your default web page by typing the www URL in a web browser E.g. `www.lumitulib.com`

## 2.6 Setting up TFS on the development server

Before proceeding with the installation of TFS and its required software, we need to create the needed user accounts for the installation and grant them the right permissions.

### 2.6.1 Setting password security policies before creating the AD accounts:

1. First, we will set password policies for the domain controller: Start > Administrative tools > group policy management > extend Forest e.g. lumitilib.com > extend Domains > extend The wanted domain e.g. lumitilib.com > extend Group policy objects > right click Default Domain Controllers Policy > Edit > select computer configuration > Policies > Windows settings > Security settings > Account Policies > Password policy
2. Right click maximum password age > properties > select Define this policy setting > set to zero > click apply > ok
3. Right click minimum password age > properties > set to zero > click apply > ok
4. Right click maximum password length > properties > check Define this policy setting > set to six > apply > ok
5. Right click password must meet complexity requirements > properties > check Define this policy > Set to disabled > apply > ok

Notice: these settings are unsecure as there are no strict password complexity requirements or password age enforced, and thus, it will be the responsibility of the user to set a sufficiently secure password for his account.

### 2.6.2 Creating Active Directory service accounts used in TFS, SQL Server, and SharePoint services

1. **In the domain controller:** click Start > Administrative tools > Active Directory Users and Computers
2. Expand the domain controller and right click the OU folder called Domain controllers > new > Organizational Unit: call the new OU: Users > press ok.
3. Right click the Users OU > new > Organizational Unit: call the new OU: TFS > press ok.
4. Right click the TFS OU > new > User
5. Create for TFS the needed user accounts: (make sure that user must change password at next logon is unchecked and password never expires is checked)
  - a. tfssetup:
    - i. First name: tfs
    - ii. Last name: setup
    - iii. User Logon name: tfssetup
    - iv. password: tfssetup

- b. tfsservice:
  - i. First name: tfs
  - ii. Last name: service
  - iii. User Logon name: tfsservice
  - iv. password: tfsservice
- c. tfsreport:
  - i. First name: tfs
  - ii. Last name: report
  - iii. User Logon name: tfsreport
  - iv. password: tfsreport
- d. wssservice:
  - i. First name: wss
  - ii. Last name: service
  - iii. User Logon name: wssservice
  - iv. password: wssservice
- e. tfsbuild:
  - i. First name: tfs
  - ii. Last name: build
  - iii. User Logon: tfsbuild
  - iv. password: tfsbuild

Remark: the previous accounts have very predictable passwords. When making the implementation some more complex passwords should be used.

## 2.7 Setting AD service account access levels and adding them to the relevant GPO

1. Adding the tfssetup user account to the local administrators in the **development server**:  
 Start > Administrative tools > Server manager > configuration > Local Users and Groups > Groups > Administrators > Right click and choose add to group > Add > Advanced > enter User name and Password of the domain controller administrator > Find Now > Select tfssetup user from the list > Ok > Apply > Ok
2. Add the created accounts to the relevant GPO groups:  
**In the development server:** press start > in the search bar type: gpedit.msc to display the “Local group policy editor” window > computer configuration > windows settings > Security settings > local policies > User right assignment > Log on as a service > Add user or Group > advanced > enter

User name and Password of the domain controller > Find Now > Select: tfsreport, tfsservice, wssservice, and tfsbuild > ok > ok > Apply

## **2.8 Disabling Windows Firewall and user account control settings (UAC) and performing the updates:**

1. Disable Windows Firewall: Start > Control panel > Windows Firewall > Turn Windows Firewall on or off > Select Turn off Windows Firewall for “Domain network location settings”, “Home or work (private)network location settings” and “public network location settings” > click Ok
2. Disable UAC: Start > in the search bar type: UAC > click on Change User Account Control Settings > drag the slider to Never notify > click Ok
3. Restart the computer (In the Shutdown event tracker window, select in the option “Other (planned)” and in the comment text box type “p” standing for planned) > click Ok
4. Perform Windows updates: Start > All programs > Windows update > Click on install updates
5. Restart the computer if needed by clicking “Restart now”

## **2.9 Installing IIS on the development server**

1. Start > administrative tools > Server manager > Roles > Add roles > In the add roles wizard click next > select Web Server (IIS) and click next > next > and select the following role services by checking their boxes:
  - a. Application development and its sub-features
  - b. Windows authentication
  - c. IIS 6 Management Compatibility and all its sub-features
2. Click next > install

## **2.10 Installing SQL Server 2008 R2 on the development server**

1. Begin the installation of SQL server from the SQL server 2008 R2 DVD
2. Start > computer > Double click the DVD reader icon where the SQL server DVD is inserted to start the installation > in the Microsoft SQL server 2008 R2 Setup window select Ok
3. Before launching the installation:
  - a. In the planning menu, perform all the checks:

- i. Check your machine for the minimum requirements of the hardware
  - ii. Launch the System Configuration Checker
  - iii. Launch the installation of the Upgrade Advisor: Launch > next > next > Finish
- b. In the Installation menu:
- i. Select the option: New installation or add features to an existing installation
  - ii. In the “pop up” Setup support rules windows select Ok after making sure that no failures or warnings are detected
  - iii. In the product Key window, enter the product key in the corresponding text box and then press next
  - iv. Check I accept the license terms and uncheck send feature usage data to Microsoft > press next > then press install
  - v. Press next > Select in the setup role window “SQL Server Feature Installation” > press next
  - vi. In the feature selection window, select the following features:
    - 1. Database Engine Services
    - 2. SQL Server Replication
    - 3. Full Text Search
    - 4. Analysis Services
    - 5. Reporting Services
    - 6. Client Tools Backward compatibility
    - 7. Client Tools SDK
    - 8. Client Tools Connectivity
    - 9. Integration Services
    - 10. SQL Server Books Online
    - 11. Management Tools – Basic
    - 12. Management Tools – Complete
  - vii. Press Next > make sure the installation rules check was successful > press Next > Next > Check that the disk requirements are fine and press Next
  - viii. In the server configuration window, click on “Use the same account for all SQL Server services” > In the account name select “NT AUTHORITY\NETWORK SERVICE” > press Ok > Change the SQL Server Agent to “Automatic”
  - ix. In the Collation Tab: press Customize... and check that the Database Engine collation is SQL\_Latin1\_General\_CP1\_CI\_AS under SQL collation, used for backwards compatibility. In the Analysis Services press Customize... and check that the collation

is Latin1\_General and that Accent-sensitive is selected > press Ok > Next

- x. Specify the Database engine Authentication mode and the administrators: Select the Authentication Mode: “Windows authentication mode” > press Add Current User and check that the Server’s administrator is visible in the list e.g. DEVSERVER\Administrator (Administrator) > press Next (the domain administrator will be added later)
- xi. Specify the Analysis Services administrators: Press Add Current User, then press Add... > advanced > enter the Domain controller’s Username and Password >press Find Now > select from the list “tfs setup” > press Ok > Next
- xii. In the “Specify the Reporting Services configuration mode” make sure the option ”Install the native mode default configuration” is selected > press next > uncheck the “Send windows and SQL Server Error report to...” > press Next > check that the Installation Configuration rules has passed successfully > press Next
- xiii. Check the summary of the SQL Server 2008 R2 features to be installed to make sure that everything is correct > click Install to start the installation > After the installation has completed successfully, press Close.

## 2.11 Configuring the Report Server

1. **In the development server:** Start > All programs > Microsoft SQL Server 2008 R2 > Configuration Tools > Reporting Services Configuration Manager
2. Enter the server name (e.g. DevServer) and the report server instance (e.g. MSSQLSERVER) > press Connect
3. Go to Web Service URL menu Item > in IP Address dropdown list, select the development server’s IP address > press Apply
4. Go to Database menu item > press Change Database > Select Create a new report server database > press Next > check that Authentication type is set to Current user (local administrator) and that the server name is the name of the development server (e.g. DevServer) > press Test Connection to check that the connection is available (you should get a confirmation dialog box that confirms that the test connection was successful) > press Next > create a name for the report server e.g. TFSReportServer, select the language (e.g. English (United States)) to be used for running SQL scripts and set the Re-

port Server Mode as Native Mode > press Next > in the credentials view, check that the Authentication type dropdown list is set to Service credentials > click Next > Next > Finish

5. Go to Report Manager URL menu item > in URLs click the link (the browser should start and a web page called SQL Server Reporting Services should be visible)

## 2.12 Installing SharePoint services 3.0 SP3

1. Download Microsoft SharePoint services 3.0 form the following link: <http://www.microsoft.com/en-us/download/confirmation.aspx?id=5709> (According to the instructions in the Microsoft's TFS installation <http://www.microsoft.com/en-us/download/confirmation.aspx?id=27828> <http://www.microsoft.com/en-us/download/confirmation.aspx?id=27828> guide)
2. Execute the downloaded file > accept the terms of the agreement and press continue > press Advanced > select Web Front End > press Install now > check "Run the SharePoint Products and Technologies Configuration Wizard now" > press close
3. At the SharePoint Products and Technologies Configuration Wizard window press Next > select yes > Select "No, I want to create a new server farm" > press Next
4. In the Database server field, enter the development server name e.g. DevServer. In the Username field enter the user name in the following format (DomainController\User account) as the configuration database is hosted in a different server e.g. (LUMITULIB\wssservice). In the Password field, enter the password for the specific user account > press Next
5. Specify the port number 17012 (as it is the port number user by TFS for SharePoint Products administration site) > set the authentication provider to be NTLM > press Next > Next > Finish
6. As a confirmation of successful installation, a web page called Central Administration should appear after you enter the user Id (tfssetup) and its password.

## 2.13 Verifying WSS installation

1. Go to SharePoint Central Administration page: Start > Administrative tools > SharePoint 3.0 Central Administration > Application Management > Authentication providers > from web application drop down list select: Change Web Application > select in the pop up window: SharePoint Cen-

- tral Administration v3 > Click on default (make sure the integrated Windows authentication is NTLM and that client integration is enabled)
2. Go to Application Management > Web Application List > click on SharePoint Central Administration v3 (You have to be redirected to the Application Management page)

## 2.14 Creating a web application and a site collection

1. Creating a web application:
  - a. Go to SharePoint Central Administration page: Start > Administrative tools > SharePoint 3.0 Central Administration > application Management > Create or extend Web application > Create a new web application
  - b. In create a new Web application web page perform the needed actions:
    - i. Select Create a new IIS web site description and give to the web site a new name e.g. TFS – 80
    - ii. Make sure the Authentication provider is set to NTLM
    - iii. In “Select a security account for this application pool” field set the security account to predefined and enter in the User Name and password fields the ID of the SharePoint service account and its password (e.g. LUMITULIB\wssservice and password)
    - iv. Press Ok (You should get a confirmation message: “Application created”)
2. Creating a site collection:
  - a. Go to Application Management > Create site collection
    - i. Under Title and Description section in the title field give a name to the site collection e.g. Sites
    - ii. Under the Web Site Address in the URL field select in the drop down list the name proposed by the system (e.g. http://devserver/sites) and in the adjacently appearing box add a name of your choice (e.g. collection) to create the site at this specific path
    - iii. Under the section Template select in the list box “Team Site”
    - iv. Under the section Primary Site Collection Administrator, in the user name field press the book icon (Browse) and in the appearing box write the first letters of the SharePoint service account (e.g. wss) and click search > select the account of the SharePoint service > press Ok
    - v. Under the section Secondary Site Collection Administrator, in the user name field press the book icon (Browse) and in the appear-

ing box write the first letters of the TFS setup account (e.g. tfs) and click search > select the account of the TFS setup > press Ok

- b. Press Ok to create the site collection (You should get a confirmation message: "Top-Level Site Successfully Created")
- c. Finally, click on the link provided by the system in the feedback message screen. You should be able to see a web page called Sites
- d. If the access is denied to you, then check the account you are signed in with. If it is not the SharePoint service account or the TFS setup account (Primary sites administrator and Secondary sites administrator), then you have to click the link "Sign in with different user" and enter in the Authentication window one of the previously mentioned accounts (e.g. LUMITULIB\wssservice and password)

## 2.15 Installing and configuring TFS

1. Begin the installation of Team Foundation Server 2010 from the TFS 2010 DVD
2. Browse the Team Foundation Server 2010 DVD for the folder "TFS-x64" for the 64 bit version of TFS > open the folder and execute the file called "setup.exe"
3. At the Microsoft Team Foundation Server 2010 Setup window, uncheck the field "Yes, send information about my setup experiences to Microsoft Corp" > press Next > Accept the license terms > press Next
4. In the TFS 2010 Setup option page select the features to install:
  - a. Select Team Foundation Server
  - b. Don't select Extensions for SharePoint Products and Technologies (installed previously manually and available on the same physical server as the one where we will install TFS 2010)
  - c. Don't select Team Foundation Server Proxy (used typically to cache files locally of a remote location, so it reduces bandwidth use and thus increases the performance)
  - d. Don't Select Team Foundation Build service
  - e. Press Install (you should get a message to confirm that the installation was successful)
5. Configuring TFS 2010:
  - a. At the end of the installation press Configure > In the configuration Center window select Standard Single Server > press Start Wizard
  - b. In the standard Configuration press next (Uncheck yes send information...) > enter the SharePoint service username and password (e.g.

- LUMITULIB\wssservice and password) > press test to check that the account works > press Next > Next > After that the readiness check has passed successfully press configure > press Next
- c. Test the links provided (enter the user ID and password of the tfs setup account when prompted e.g. LUMITULIB\tfssetup and password) > press Close > Close

## 2.16 Setting up the build server

NOTE: According to Microsoft's TFS 2010 installation guide, the best practice is to install the build server on an independent physical server. The purpose of that is to increase the performance, because when the build server is located at the same physical machine as the TFS application tier, there is a heavy demand on the resources (processor and memory, especially when a build controller is managing many builds at the same time). As a result of this, the TFS application tier may slow down because of the resources used by the build service. However, and according to the installation guide, if there are few teams and the builds are performed nightly or there are not so many parallel builds, then installing TFS build server and the TFS application tier on the same physical server may not have an adverse effect on performance.

### 2.16.1 Checking the requirements for installing Team Foundation Server Build Service

1. According to Microsoft's TFS 2010 installation guide, there are some minimum requirements that have to be met in order to be able to install TFS build service:
  - a. Checking that the hardware and software (OS) are supported from TFS 2010 installation guide
  - b. Check for the required permissions and user accounts:
    - i. The account used to install TFS build service (e.g. tfssetup) is a member of the administrators: Start > Administrative tools > Server Manager > Configuration > Local Users and Groups > Groups > right click administrators > properties > check that the account is in the list of members

- ii. If the account used to install TFS build service is not a member of the Project Collection Administrators group then extra configuration steps will be needed.

### **2.16.2 Install TFS Build Service**

1. Begin the installation of the TFS build service from the Team Foundation Server 2010 DVD
2. Browse the Team Foundation Server 2010 DVD for the folder “TFS-x64” for the 64 bit version of TFS > open the folder and execute the file called “setup.exe”
3. At the Microsoft Team Foundation Server 2010 Setup window, uncheck the field “Yes, send information about my setup experiences to Microsoft Corp” > press Next > Accept the license terms > press Next > Select Add or Remove Features
4. In the TFS 2010 Setup option page select the features to be installed (in addition to Team Foundation Server):
  - a. Select Team Foundation Build Service
5. Press Update
6. Select the option ”Configure Team Foundation Build Service” > press Start Wizard > Next >
7. In Select a team project collection press browse > in the window “Connect to a team project collection” select the Development server and under Team project collections select DefaultCollection > press connect > press next
8. Select Use the default setting and under Number of build agents to run on this build machine, select: 2 (recommendation is 1 build agent per CPU core. In our case we have a dual core CPU) > press next
9. Select use a system account and set the account to NT AUTHORITY\SYSTEM and select the default port 9191 > next
10. Press verify to check that everything works properly (all the checks has to pass successfully) > press next
11. Click configure to start the configuration and when completed successfully > press next (you should get a message that the installation was successful) > close > close

### **2.16.3 Creating a shared folder for storing build outputs**

1. Create a new folder in the C drive (start > computer > double click LocalDisk (C:) > right click > new > folder) and call it tfsBuildOutput

2. Right click the folder and select properties
3. Click the sharing tab and select: Advanced Sharing...
4. Check Share this folder > press Permissions
5. Press add > press Advanced > enter Domain's administrator Id and password > press Find Now > from the list select: tfsbuild > press ok
6. Set the permission of the tfsbuild account to full control > press apply > press Ok > Ok > Close.

## **2.17 Setting a build controller and build agents for a specific team project collection**

NOTE: this section is optional and describes the needed steps to create a new build controller for a team project collection in the build server.

### **2.17.1 Registering a build service, creating a build controller and creating build agents**

In this part we will create a build controller and create agents for it. But before that, we have to register a build service to be used by a specific team project collection:

Start the Team Foundation administration console > expand the name of your server e.g. DevServer > select build configuration > register > under Connect to team project collection (outgoing) click browse > select the name of the development server where TFS is installed > select the collection for which you want to create the build service, from the list of team project collections > press connect > under local build service endpoint (incoming), press change and under host name and path write the development server's fully qualified computer name (e.g. tfs.lumitilib.com) > under endpoint details set port number to 9191 and the protocol to http > ok > under run the build service as select Windows service > and under credentials keep the default options (this account selected: NT Authority\System ) > Start

After a successful registering, a message will be displayed: Build service configured for "name of the server\name of the collection" e.g. 172.28.150.2\eba\_collection

### **Creating a build controller:**

Start the Team Foundation administration console > expand the the name of your server e.g. DevServer > select build configuration > new controller > under display name enter a name for the controller > optionally you can give a description for the controller under description > keep version control path to custom path empty > set maximum number of concurrently running builds to default to number of agents > make sure that option “build controller service is enabled” is checked > ok.

A feedback text and sign will be displayed: the controller icon has a green sign on it and the word ready.

### **Creating a build agent:**

According to Microsoft’s recommendations, the number of build agents that should be installed on a machine (server) has to be equal to the number of cores in its processor. E.g. a server with a quad core processor (4 cores) can have 4 build agents installed on it.

Start the Team Foundation administration console > expand the name of your server e.g. DevServer > select build configuration > new agent > under display name, assign a name to the agent > you can optionally give a description to the agent > in the section controller, select a controller from the drop down list > in working directory keep the default value > make sure the option: build agent service is enabled > ok.

A feedback text and sign will be displayed: the controller icon has a green sign on it and the word ready.

## **3. Setting the client**

### **3.1 Windows 7**

#### **3.1.1 Installing Windows 7 and Visual Studio 2010**

In this project, the computers had a ready image of windows 7 with visual studio 2010 installed on them. The image is provided by HAAGA-HELIA’s IT department.

### 3.1.2 Creating a local user in the client

Turn on the work station (Windows 7 will boot) > when the log in page is prompted log in as a **local administrator** (user Id and password provided by the IT department)  
Start > control panel > User and Accounts > Manage User Accounts > select advanced tab > under Advanced user management press Advanced button > right click the folder called Users and select New User... > Enter the new user name and its password > uncheck: user must change password at next logon > check: User cannot change password and check: password never expires > press create  
Right click the user you created in the Users folder and select properties > Select the tab: Member of and make sure your user belongs to the group Users.

### 3.1.3 Configuring the client IP settings

Start > Control panel > network and sharing centre > Change adapter settings > right click the adapter (Local Area Network) and select properties > uncheck Internet protocol Version 6 (TCP/IPv6) > select Internet protocol Version 4 (TCP/IPv4) > press properties > select the option Use the following IP address and the option use the following DNS server addresses > in IP address enter a unique address to identify each work station > enter the subnet mask > enter the default gateway > and in the preferred DNS server enter the domain controller's IP address  
Press Advanced button > in the tab Wins Select the option: disable NetBIOS over TCP/IP > press Ok > Ok > Ok

### 3.1.4 Adding the Client to the domain controller

Start > Control panel > System > Change settings > press the change button > in the field member of select the option domain, and then enter the domain controller's name (e.g. lumitulib.com) > press Ok > an authentication window will be prompted: enter the domain controller's administrator's user Id and password > press Ok > restart the computer.

Check that internet is available (e.g. browse to some URL)

### **3.1.5 Connecting the client to TFS**

The connection to TFS is done through Visual Studio (in this case 2010). The step of installing Visual Studio is skipped as it is available in the image we installed on the work station(s).

Start Visual Studio 2010 > Team > Connect to the Team Foundation Server > press servers > press Add > In the field: Name or URL of the Team Foundation Server enter the URL of the server in the following format: “DevelopmentServer-Name.DomainName.com” (e.g. DevServer.lumitulib.com) > in the path field enter: tfs > in the Port number enter: 8080 > select the Protocol: HTTP > press Ok > an authentication window will be prompted: enter the domain controller administrator’s user Id and password > press Ok > press close > Select the added server from the drop down list and select the default collection > press Connect

### **3.1.6 Uploading SCRUM process template to TFS**

Download the Microsoft Visual Studio Scrum Template from <http://visualstudiogallery.msdn.microsoft.com/59ac03e3-df99-4776-be39-1917cbfc5d8e>

Install Microsoft Visual Studio Scrum Template 1.0 by executing the downloaded file. Start Visual Studio > Team Explorer > Right-click the server > Team Project Collection Settings > Process Template manager...

Click upload and locate the folder on which you installed the Scrum template > select the folder > Ok

## **3.2 Installing Windows 8 along with Windows 7 (dual boot)**

### **3.2.1 Minimum requirements for W8 installation**

According to Microsoft (2013g), the minimum hardware requirements for Windows 8 are the following:

- **Processor:** 1 Gigahertz (GHz) or faster with support for PAE, NX, and SSE2
- **RAM:** 1 GB for 32-bit systems or 2 GB for 64-bit systems
- **Hard disk space:** 16 GB for 32-bit or 20 Gb for 64-bit
- **Graphics card:** Microsoft DirectX 9 graphics device with WDDM driver

### 3.2.2 Operations performed in the BIOS:

In order to be able to run Windows phone emulator, the processor needs to have the Virtualization technology available:

Access the BIOS (e.g. by pressing F10) → check your system's BIOS access key.

In the BIOS, under the security tab, enable the virtualization features:

Virtualization Technology (VTx)

Virtualization Technology Directed I/O (VTd) (if available)

In the Boot order option, set the boot order to start from the DVD reader for the Windows 8 installing

NOTE:

In some computers, disabling the EFI system might be mandatory to allow the dual boot of Windows 7 and Windows 8: The Windows 7 image used at HAAGA-HELIA is installed based on the MBR system. When Windows 8 detects that the computer where the installation will be performed supports the EFI system, it does not allow the installation on the old MBR system based HDDs.

The solution: in the BIOS under the security menu > disable the EFI boot sources option and under the legacy boot sources option, set at the first position the drive from where we will perform the Windows 8 installation (e.g. DVD drive)

Save the changes and exit.

## 3.3 Installing Windows 8

### 3.3.1 Pre-installing phase for dual boot installation

Before proceeding with installing, we need to create a partition where we will install Windows 8:

Start the computer > Start > Control panel > Administrative tools > Computer management > Storage > Disk management (local) > right click the C drive and select Shrink volume > calculate the right amount of space to shrink (depends on what will be installed along with Windows 8) > input the amount to shrink in the text box with field: Enter the amount of space to shrink in MB > press Shrink.

As a result, there will be a new partition with the following information: Its size and “Unallocated”

### 3.3.2 Starting the installation

Start the installation, insert the DVD of Windows 8 professional 64 bit and restart the computer.

Press any key when asked to for stating the boot from the DVD

Select the language to install (English united states), time and currency (Finnish), keyboard or input method (Finnish) > next > press install now

Enter the product key > press next > accept the terms and conditions > Select **Custom: install windows only (advanced)** > Select from the list of available partitions the one you created previously > Press install

Select the colour theme you want, and in PC name field enter the name of you PC > press Next > Press Customize > select the following options:

No, don't turn on sharing or connect to devices

In drop down list select: Automatically install important and recommended updates

Turn on: Automatically get device drivers, apps, and info for new devices

Turn on: Turn on Windows SmartScreen to check files and apps with Microsoft

Turn on: Turn on Internet Explorer SmartScreen Filter to check URLs and downloads with Microsoft

Turn on: Send a do not track request to websites you visit in Internet explorer

Press Next > Select the following options:

Turn off: Help improve Windows store by sending URLs for web content that apps use

Turn on: Help Microsoft to respond to malicious apps and malware by joining Microsoft Active Protection Service

Turn off: Help improve Microsoft services by sending some location data when location-aware apps are used

Turn off: Participate in the customer Experience improvements program...

Turn off: Help improve Windows Help content...

Press next > Select the following options:

Turn on: Use windows error reporting to check for solutions to problems

Turn on: Use internet Explorer compatibility lists to help improve my experience on some sites

Turn off: let apps use my name and account picture

Turn off: Turn on windows location platform so apps can ask users for their location

Press next > Select Sign in without a Microsoft email address > Press Local account  
Create a local Administrator account (Username/password) and the password hint >  
press Finish

### **3.3.3 Create a local user account**

In the tile screen, press Desktop > move the mouse to one of the right corners > press settings > Control panel > User accounts > manage another account > Add new user in PC settings > Add a user > Sign in without a Microsoft account > Local account > Input the user name and password, and a password hint > press next > Finish

### **3.3.4 Set up the IP configuration**

In the tile screen, press Desktop > move the mouse to one of the right corners > press settings > Control panel > Network and sharing center > change adapter settings > right click Ethernet and select properties > uncheck Internet Protocol Version 6 (TCP/IPv6) > select Internet Protocol Version 4 (TCP/IPv4) and press properties > select the options: use the following IP address automatically and use the following DNS server addresses:

In the IP address enter a unique IP address (same as the one used for windows 7 in the same computer)

Enter the subnet mask

Enter the Default gateway IP address

Enter the DNS server address which is the address of the domain controller

Press advanced > in wins tab select the option Disable NetBIOS over TCP/IP > ok > ok > close

Test that internet connection works.

### **3.3.5 Adding the computer to the domain controller**

In the tile screen, press Desktop > move the mouse to one of the right corners > press settings > Control panel > system > press change > select in the field member of the option domain > enter the domain URL (e.g. lumitilib.com) > when prompted enter the domain's administrator user id and password to join the pc to the domain > ok > restart the computer

### **3.3.6 Changing the default operating system to be Windows 7**

When restarting the computer, you will be prompted to choose between booting Windows 7 or windows 8.

Select change defaults or choose other options > choose default operating system > Select Windows 7 > press the back arrow > select the OS you want to launch

### **3.3.7 Installing Visual Studio 2012**

Start the computer and log in as administrator > Insert the DVD of Visual Studio 2012 Ultimate > in the tile screen right click and select all apps > select Computer > in the computer window right click the icon of Visual Studio 2012 and press open > double click the file called VS\_Ultimate > select the installation location > agree the term and conditions (don't select: Join the customer improvement program to help improve...)

> next > press Install > in the user account control (if prompted) press yes > when the installation is ready press Launch

In the Choose default Environment settings select General development settings and select in the Local help documentation drop down list: Advanced > press Start visual studio

In the Manage content tab select all the documentation to be installed in order to have it available locally in the C drive: press the add link in every row of the grid > press update > yes.

### **3.3.8 Installing Windows Phone SDK 8.0**

Start Visual Studio 2012 > File > New > Project > install Windows phone SDK 8.0 > Ok

Press Download Windows Phone SDK 8.0 > Press Download WPexpress\_full.exe > Run > Run

Follow the instruction of the installing wizard: Close visual studio > press continue (in the windows phone SDK 8.0 window) > agree the license terms and conditions (Don't select Join the customer experience improvement program to help improve...) > press install.

## **4. Securing Active Directory**

### **4.1 Creating the business structure**

The business structure is a structure that will help us managing the users and groups we will be creating.

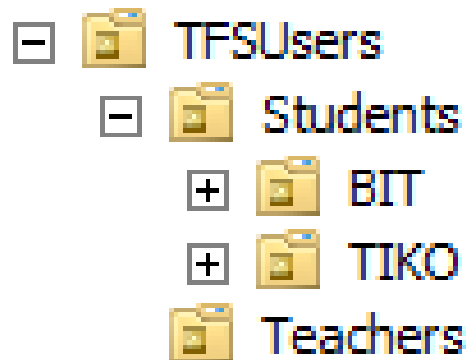
#### **4.1.1 Creating the OU structure**

Click start > Administrative Tools > Server Manager > expand Roles > Active Directory Domains Services > Active Directory Users and Computers > expand the name of your domain (e.g. lumitulib.com) > right click you domain name > new > Organiza-

tional Unit (OU) > give a name to the OU and keep “protect container from accidental deletion” checked > click ok.

Proceed with the creating of the rest of the structure as shown in the picture 1. Follow the same procedure as described above.

The picture 2 describes the business structure with OUs you should get at the end of this process.

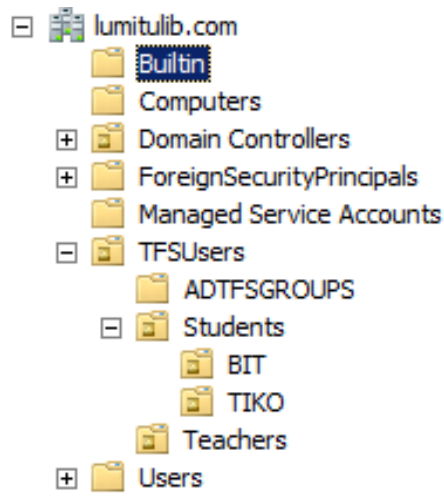


Picture 2: The business structure with OUs

#### 4.1.2 Creating a container for the custom AD groups

Click start > Administrative Tools > ADSI edit > More Actions > Connect to... > keep the default name and path provided, under connection point in select a well-known naming context select Default naming context and under computer select default (domain or server you that you are logged in to) > ok > double click your server name (e.g. NetDevDC.lumitulib.com) > double click you domain name (e.g. DC=lumitulib, DC=com) > right click the OU we created (TFSusers) > new > Object > select “container” > next > write a name for the container in the field “value” (e.g. ADTFSGROUPS) > next > finish.

The picture 3 depicts the final structure you should get.

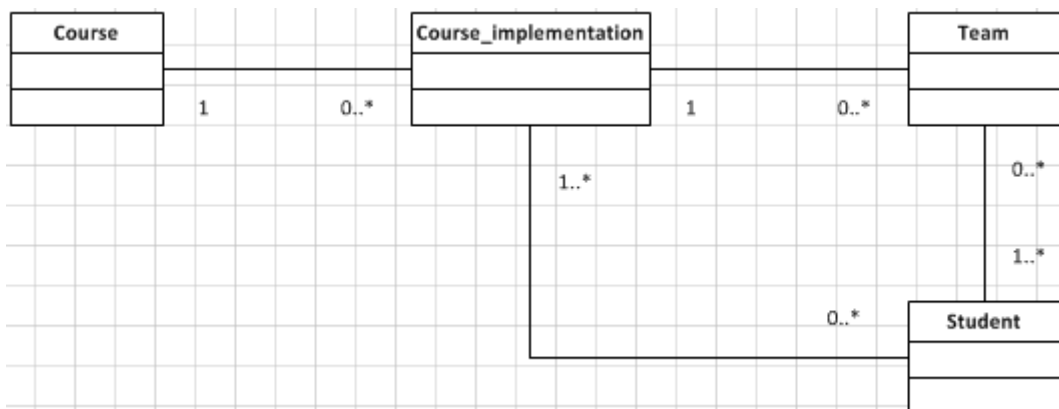


Picture 3: The final state of the business structure

## 4.2 Creating the custom AD groups

The reason behind creating the custom AD groups is to classify and give filtering possibility of the users (e.g. students) based on course implementation and teams they belong to. A student might belong to different course implementations, and for this reason the business structure cannot be implemented using OUs because a user cannot be in more than one OU at a time.

The ER diagram in the picture 4 depicts the relationships between the courses, course implementations, teams and students.

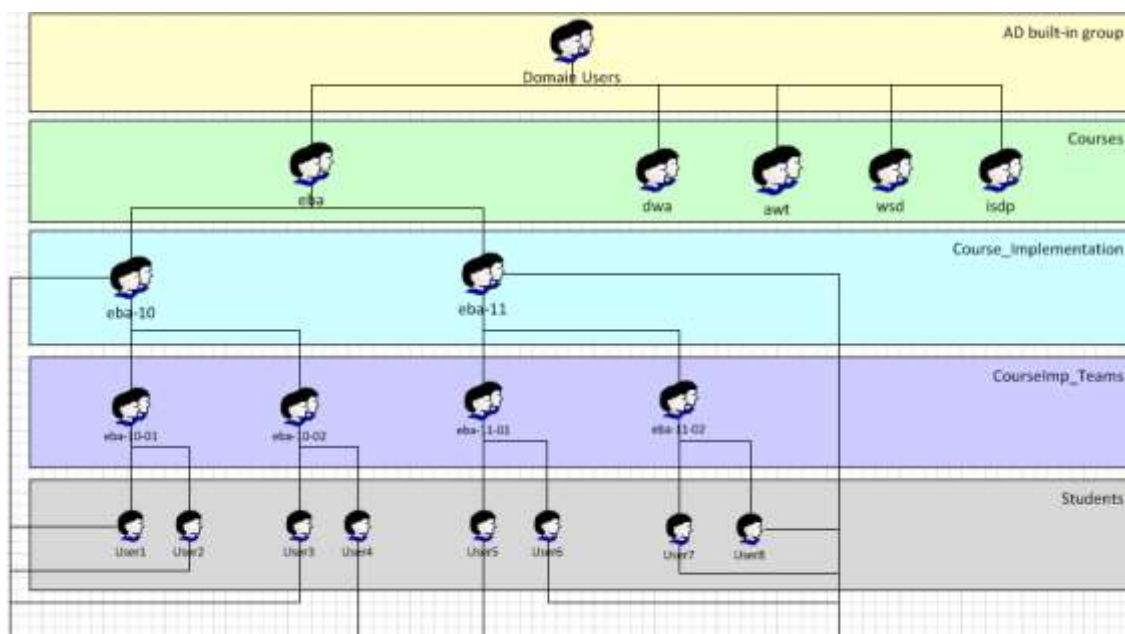


Picture 4: ER modelling of the business AD groups

The picture 5 describes an example of a custom AD group structure for a course called “eba”:

Eba course has 2 implementations (eba-10 and eba-11), each implementation has 2 teams:

- eba-10 includes the teams: eba-10-01 and team eba-10-02.
- eba-11 includes the teams: eba-11-01 and team eba-11-02



Picture 5: Custom AD group structure for “eba”

In order to enhance the maintainability of the system, its security and differentiate between the security AD groups and groups used to classify and filter users, the custom AD groups will be kept in the container we created inside of the OU TFSUsers called ADTFSGROUPS. In the other hand, the AD security groups we will create will be kept in their AD built-in container called Users.

Below are listed the steps to create a custom AD group structure. The example of the picture 5 will be taken as a model:

### **Creating a course group and setting its group membership:**

Click start > Administrative Tools > Server Manager > expand Roles > Active Directory Domains Services > Active Directory Users and Computers > expand the name of your domain (e.g. lumitulib.com) > expand the TFSUsers OU > in view check advanced features in order to be able to see hidden objects (our ADTFSGROUPS created container) > right click ADTFSGROUPS > new > Group > in Group name write: eba, under Group scope select Global and under Group type select Security > Ok. Right click the group created (eba) > properties > member of tab > check that eba group is member of Domain users group.

### **Creating a course implementation group and setting its group membership:**

Click start > Administrative Tools > Server Manager > expand Roles > Active Directory Domains Services > Active Directory Users and Computers > expand the name of your domain (e.g. lumitulib.com) > expand the TFSUsers OU > in view check advanced features in order to be able to see hidden objects (our ADTFSGROUPS created container) > right click ADTFSGROUPS > new > Group > in Group name write: eba-10, under Group scope select Global and under Group type select Security > Ok. Right click the group created (eba-10) > properties > member of tab > Add > Advanced > Find now > Select eba group > Ok > Ok > in the member of tab, check that our created group (eba-10) is member of the eba group > Ok.

### **Creating a team group and setting its group membership:**

Click start > Administrative Tools > Server Manager > expand Roles > Active Directory Domains Services > Active Directory Users and Computers > expand the name of your domain (e.g. lumitulib.com) > expand the TFSUsers OU > in view check advanced features in order to be able to see hidden objects (our ADTFSGROUPS created container) > right click ADTFSGROUPS > new > Group > in Group name write: eba-10-01, under Group scope select Global and under Group type select Security > Ok.

Right click the group created (eba-10-01) > properties > member of tab > Add > Advanced > Find now > Select eba-10 group > Ok > Ok > in the member of tab, check that our created group (eba-10-01) is member of the eba-10 group > Ok.

### **Creating a student (user) and setting its group membership:**

Click start > Administrative Tools > Server Manager > expand Roles > Active Directory Domains Services > Active Directory Users and Computers > expand the name of your domain (e.g. lumitulib.com) > expand the TFSUsers OU > Expand the Students OU > right click the BIT organizational unit > new > User > fill the first name and last name fields, assign a user logon ID (User1) > next > write a default password for the user and check the option: User must change password at next logon > next > finish.

Right click the user you created (User1) > properties > member of tab > Add > Advanced > Find now > Select eba-10 and eba-10-01 groups > Ok > Ok > in the member of tab, check that our created user (User1) is member of the eba-10 and eba-10-01 groups > Ok.

### **4.3 Creating the teachers group**

The teachers need to have administrative rights over the business structure we created, but not over the whole domain. Moreover, the teachers need to have administrative rights over TFS and its components (SQL, SharePoint in order to be able to create team project collections, team projects, granting user rights for projects, etc...). Hence, the need of a group that grants these particular privileges to its members.

The group to be created for the teachers is a security group. It will be used to grant a certain access level and thus, it will be placed in the built-in container called Users.

The following steps are the needed to create a new security group for the teachers called "TFSUsers\_OU\_Admins":

Start > Server manager > roles > active directory domain services > Active directory users and computers > the domain's name (e.g. lumitulib.com) > right click users > new > group > under group name: name of the group e.g. TFSUsers\_OU\_Admins > select group scope: global and group type: security > ok.

### **4.3.1 Delegating the business structure's administration (TFSUsers OU) to the TFSUsers\_OU\_Admins (the group to whom teachers belong)**

Start > Server manager > roles > active directory domain services > Active directory users and computers > the domain's name (e.g. lunitulib.com) > right click the OU which administration will be delegated to the TFSUSer\_OU\_Admins > delegate control... > next > Add > Advanced > Find now > select the group TFSUSer\_OU\_Admins from the list > next > select the option: delegate the following common tasks and select all the options > Finish.

## **4.4 Enabling and setting Remote desktop**

### **4.4.1 Enabling remote desktop service**

In the domain controller as well as in the development server do the following:

Start > Control panel > System > Remote Settings > Select allow connections only from computers running remote desktop with network level authentication (more secure).

### **4.4.2 Adding the teachers user group (TFSUsers\_OU\_Admins) to the remote desktop users group**

#### **4.4.2.1 In the Domain controller**

The domain administrator user is by default a member of the remote desktop users group, thus we don't need to add it. Nonetheless, we would like to grant the privilege of using remote desktop also to future domain administrators in addition to the teacher group (TFSUsers\_OU\_Admins). So we will add both groups and for that we will proceed as follows:

Start > control panel > system > remote settings > press select users > add... > advanced > find now > enter the domain administrator credentials if prompted > select the teachers group (TFSUSer\_OU\_Admins) and Domain Admins group > ok > ok > ok.

#### 4.4.2.2 In the development server

The local administrator of the development server (DevServer) is a member of the remote desktop users by default but the domain administrators group is not. So we will add the domain administrators group along with the teacher group (TFSUser\_OU\_Admins) as users of the remote desktop services in the development server. For the following task we will proceed as follows:

Start > control panel > system > remote settings > press select users > add...> advanced > find now > enter the domain administrator credentials if prompted > select the teachers group (TFSUser\_OU\_Admins) and Domain Admins group > ok > ok > ok.

#### 4.4.3 Modifying the relevant GPO to allow remote access to the domain for TFSUsers\_OU\_Admins group

In the domain controller do the following: Start > Server manager > features > group policy management > Forest: Lunitulib.com > domains > lunitulib.com > Group policy objects > right click default domain controllers policy > computer configuration > policies > Windows settings > security settings > local policies > user rights assignment > allow log on through remote desktop services > add user or group > browse > advanced > Find now > select the group TFSUsers\_OU\_Admins and the Domain admins group > ok > ok > ok > ok.

## **4.5 Granting administrative privileges of TFS and its components to the teachers group (TFSUsers\_OU\_Admins)**

### **4.5.1 Setting the rights to access the development server remotely**

#### **4.5.1.1 Enabling remote desktop service:**

Done in the subchapter 4.4.1

#### **4.5.1.2 Adding the teachers user group (TFSUsers\_OU\_Admins) to the remote desktop users group**

Done in the subchapter 4.4.2

### **4.5.2 TFS console user requirements**

According to Microsoft's documentation (TFS administrator guide), a user or a group can have full administrator privileges of TFS if:

- he belongs the server's local admins group (server where TFS is installed)
- he belong to the TFS global group called: Team foundation administrators
- is granted full control over the SharePoint services 3.0 (If SharePoint sites are used)
- belongs to the Team Foundation Administration console users list
- he belongs to the SQL Server 2008 R2 serveradmin group or sysadmin group

#### **4.5.2.1 Adding the teachers group (TFSUsers\_OU\_Admins) to the local administrators group in the development server**

Start > control panel > administrative tools > computer management > local users and groups > groups > right click administrators > Add to group > add... > advanced > enter the domain controller's administrator credentials when prompted > Find now > select the TFSUsers\_OU\_Admins group > ok > ok > ok.

#### **4.5.2.2 Adding the teachers group to the Team Foundation Administrators group**

Start the Team Foundation administration console > expand the development server (DevServer) > expand application tier > expand the application tier summary > group membership > select [Team Foundation]\Team Foundation Administrators group > properties > under add a member select Windows user or group > add... > advanced > enter the domain controller's administrator credentials when prompted > Find now > select the TFSUsers\_OU\_Admins group > ok > ok > ok.

#### **4.5.2.3 Adding a user belonging to the teachers group (TFSUsers\_OU\_Admins) to the Team Foundation console users**

The team foundation console only allows adding users and thus we have to add every single user (teacher) separately. The procedure is as follows:

Start the Team Foundation administration console in the development server > expand the development server (DevServer) > expand the application tier > under Administration console users select add > expand advanced features and make sure that Add required permissions to create collections is selected then press search > advanced > find now > enter the domain administrator credentials if prompted > select the user you want to add (teacher user) > ok > ok.

#### **4.5.2.4 Adding the teachers group (TFSUsers\_OU\_Admins) and the domain administrator to the SQL server's serveradmin group**

In order to be able create a team project collection; a user belonging to the TFSUsers\_OU\_Admins has to belong to SQL serveradmin or sysadmin groups.

A new feature in Microsoft SQL Server 2008 R2 is that neither users belonging to the domain administrators nor users belonging to local admins groups have any high privileges in SQL server 2008 R2. Therefore, the privileges have to be added manually and explicitly for each user or group.

### **Granting the domain administrators group the SQL sysadmin rights:**

Log in to the development server (where SQL server 2008 R2 is installed) with local administrator account > start > in search write: ssms.exe to open the SQL server management studio > in the server type select: Database engine, in server name: DevServer (name of the server where the SQL server is installed), authentication type: Windows authentication > click connect > expand security > right click logins > right click BUILTIN\Administrators (domain administrators group) > properties > under server role select sysadmin > under user mapping select all the available databases except master, model, msdb, and tempdb > Under Status select grant for permission to connect to database engine, and select enabled for login > press ok to add privileges to the login account for the domain administrators.

### **Granting the TFSUsers\_OU\_Admins SQL serveradmin rights:**

Log in to the development server (where SQL server 2008 R2 is installed) with local administrator account > start > in search write: ssms.exe to open the SQL server management studio > in the server type select: Database engine, in server name: DevServer (name of the server where the SQL server is installed), authentication type: Windows authentication > click connect > expand security > right click logins > new login > under login name press search... > advanced > enter the domain controller's administrator credentials when prompted > press object types > make sure the groups options is selected > ok > press locations > enter the domain controller's administrator credentials when prompted > select entire directory > ok > press find now > select the TFSUsers\_OU\_Admins group (teachers group) > ok > ok > make sure that Windows authentication is selected > under default database section select master > under default language select English.

Under server roles select public and make sure serveradmin is selected too.

Under User mapping, select all the available databases except master, model, msdb, and tempdb.

Under Status select grant for permission to connect to database engine, and select enabled for login > press ok to create the new login.

Note: Another log in could be created for the group Domain Admins. But it is not necessary as long as all the domain administrators are member of the built in group Administrators (BUILTIN\Administrators).

#### 4.6 Granting TFS users reading rights in the Reporting Server

In order to be able to view reports generated by TFS, a user that belongs to a team project or a team project collection needs some specific rights in the Reporting server.

Open internet explorer > type the following address in address bar:

http://**ReportServer**/Reports/Pages/Folder.aspx and replace ReportServer with the name of your server (e.g. DevServer) > in the home page press folder settings tab > under security press new role assignment tab > under Group or use name section write the domain users group name (e.g. LUMITULIB\Domain Users) and be careful with spelling mistakes as the site does not provide browsing options to retrieve Active Directory user and group names > make sure “Browser” is checked > press ok.

#### 4.7 Security settings using GPOs

In the domain controller do the following:

Start > administrative tools > server manager > expand features > group policy management > select your forest (e.g. forest: lumitulib.com) > domains > select your domain (e.g. lumitulib.com) > group policy objects > right click default domain controller policy > edit... > computer configuration > policies > Windows settings > security settings > local policies > security options:

- Accounts: guest account status > set to disabled
- Interactive log on: display user name when the session is locked > set user display name only.
- Interactive log on: do not display last user name > set to enabled
- Interactive log on: require domain controller authentication to unlock workstation > set to enabled.

Start > administrative tools > server manager > expand features > group policy management > select your forest (e.g. forest: lumitilib.com) > domains > select your domain (e.g. lumitilib.com) > group policy objects > right click default domain policy > edit... > computer configuration > policies > Windows settings > security settings > Account policies > Password policy:

- Right click maximum password age > properties > select Define this policy setting > set to zero > click apply > ok
- Right click minimum password age > properties > set to zero > click apply > ok
- Right click maximum password length > properties > check Define this policy setting > set to six > apply > ok
- Right click password must meet complexity requirements > properties > check Define this policy > Set to disabled > apply > ok

Local policies > Security options:

- Accounts: guest account status > set to disabled
- Interactive log on: display user name when the session is locked > set user display name only.
- Interactive log on: do not display last user name > set to enabled
- Interactive log on: require domain controller authentication to unlock workstation > set to enabled.

## 5. Creating a shared folder for the scripts

### 5.1 Purpose

The purpose of creating a shared folder for the scripts is to make the PowerShell scripts available for some specific users locally as well as remotely.

The shared folder should have the delete permission disabled to avoid the accidental deletion of the scripts.

### 5.2 Creating and setting the shared folder for PowerShell scripts

In the domain controller do the following:

Start > Computer > local disk (C:) > right click and select new > folder > give you folder a name (e.g. ScriptsShared) > right click the folder > properties > sharing > advanced sharing > check share this folder > under share name, write the name used

for mapping the folder (e.g. ScriptsShared) > ok > select security tab > advanced > permissions > change permissions > uncheck the option “Include inheritable permissions from this object’s parent” > click add > in the permissions entries select both domain users (e.g. lumitulib\users) and uncheck and click remove. With this we will not allow normal users to map this folder. >click add > advanced > find now > select the teachers group (e.g. TFSUsers\_OU\_Admins) > ok > ok > from the list of permissions select: Traverse folder / execute file, list folder / read data, read attributes, read extended attributes, create files / write data, create folders / append data, write attributes, write extended attributes, read permissions > ok > ok > ok > ok.

Important: some PowerShell scripts need a folder called “output”. Create a folder within the shared folder you created and call it “output”.

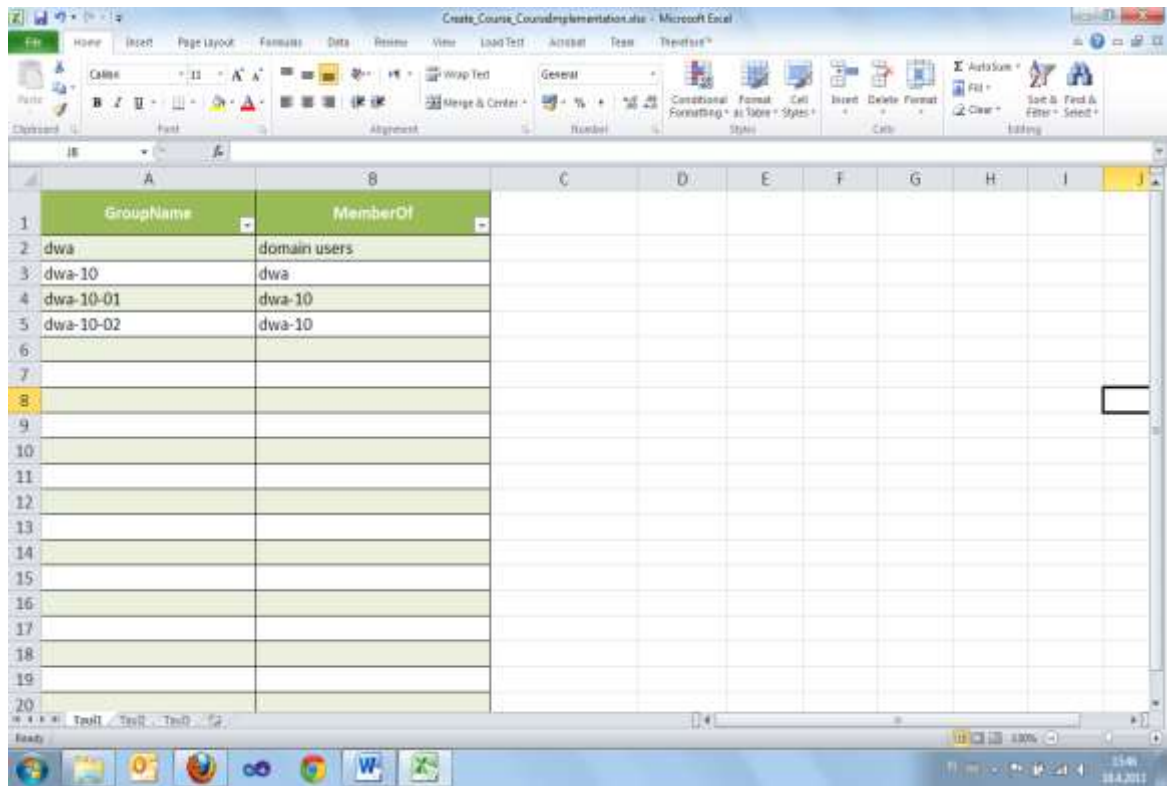
## Appendix 2: Tables of values

<b>Domain Controller</b>	
<b>Name</b>	<b>Value</b>
<b>Domain Controller Computer Name</b>	NetDevDc
<b>Domain Controller IP address</b>	172.28.150.1
<b>Domain Controller subnet mask</b>	255.255.0.0
<b>DNS server</b>	172.28.11.67
<b>Domain Controller Default gateway</b>	172.28.1.254
<b>FQDN of the forest root domain</b>	lunitulib.com
<b>Forest functional level</b>	Windows server 2008 R2
<b>Network ID</b>	IP address: 172.28.150.1  subnet mask: 255.255.0.0  172.28.0.0  Network ID: 172.28.
<b>host name (alias)</b>	www

<b>Development Server</b>	
<b>Name</b>	<b>Value</b>

<b>Computer Name</b>	DevServer
<b>IP address</b>	172.28.150.2
<b>Domain Controller subnet mask</b>	255.255.0.0
<b>DNS server</b>	172.28.150.1
<b>Domain Controller Default gateway</b>	172.28.1.254
<b>Host name</b>	DevServer
<b>Host name (alias)</b>	tfs

### Appendix 3: Course structure



The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J
1	GroupName	MemberOf								
2	dwa	domain users								
3	dwa-10	dwa								
4	dwa-10-01	dwa-10								
5	dwa-10-02	dwa-10								
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										

## Appendix 4: Students' list

The screenshot shows a Microsoft Excel spreadsheet with the following data:

Row	On Student	Name	Type Attendance	Basic eduz Situation	Completik Checked	Used term	Departme	Programr	Group	Option
2			P LA	ULK H	25.1.1900 21.1.2013	2	PASILA	BITE	BITE125	
3			P LA	ULK H	20.1.1900 21.1.2013	8	PASILA	BITE	BITE095	
4			P LA	ULK H	4.4.1900 21.1.2013	13	PASILA	BITE	BITE05K	
5			P LA	H	7.2.1900 21.1.2013	6	PASILA	BITE	BITE105	
6			P LA	SWO H	27.1.1900 21.1.2013	2	PASILA	BITE	BITE125	
7			P LA	SWO H	19.1.1900 21.1.2013	2	PASILA	BITE	BITE125	
8			P LA	SWO H	19.1.1900 21.1.2013	2	PASILA	BITE	BITE125	
9			P LA	SWO H	20.1.1900 21.1.2013	2	PASILA	BITE	BITE125	
10			P LA	SWO H	28.1.1900 21.1.2013	2	PASILA	BITE	BITE125	
11			P LA	ULK H	17.2.1900 21.1.2013	2	PASILA	BITE	BITE125	
12			P LA	H	15.1.1900 21.1.2013	1	PASILA	BITE	BITE13K	
13			P LA	SWO H	19.1.1900 21.1.2013	2	PASILA	BITE	BITE125	
14			P LA	SWO I	21.1.1900 21.1.2013	2	PASILA	BITE	BITE125	
15			P LA	H	10.2.1900 21.1.2013	6	PASILA	BITE	BITE105	
16			P LA	H	10.2.1900 21.1.2013	6	PASILA	BITE	BITE105	
17			S LA	H	17.1.1900 21.1.2013	0	AVO	AVO	AVO13K120	
18			P LA	SWO H	8.2.1900 21.1.2013	4	PASILA	BITE	BITE115	

## Appendix 5: Display course implementation members or teams

```
import-module ActiveDirectory

echo "======"

echo "*"

echo "* Enter a course implementation code to display the users  *"

echo "* belonging to it, or enter a team name to view its members"

echo "* Enter empty string to quit                               *"

echo "*"

echo "======"

write-host

do

{

    $group=Read-Host "Please enter the course implementation code"

    write-host

    if($group -ne "")

    {

        $results = Get-ADGroupMember $group | sort objectclass, name | ft

        samaccountname, name, objectclass -a

    }

    if($results)

    {

        write-host "Users for the course code: " $group

    }

}
```

```
        write-host
        $results
    }
    else
    {
        write-host "No user available for the given course code"
    }
    write-host
    "=====";
    write-host
}
}
while ($group -ne "")
```

## Appendix 6: Display students of a specific entry year

```
import-module ActiveDirectory

echo "===== "

echo "*"                                     "*"

echo "* Enter an entry year (e.g. bite10k)   *"

echo "* Enter empty string to quit          *"

echo "*"                                     "*"

echo "===== "

write-host

do

{

    $description=Read-Host "Please enter the entry year"

    write-host

    if($description -ne "")

    {

        $results = get-aduser -filter 'description -eq $description' -
searchbase "OU=TFSUsers,DC=lumitulib,DC=com" | sort-object | ft samaccountname,
name

        if($results)

        {

            write-host "Users for the course code: " $description

            write host
```

```
        $results
    }
    else
    {
        write-host "No user matching the given course code"
    }
    write-host
    "=====";
    write-host
}
}
while ($description -ne "")
```

## Appendix 7: Create a course structure

```
import-module ActiveDirectory

write-host "===== "

write-host "*"

write-host "This PS command will create an AD group structure (groups "
write-host "and their relationships) from a CSV file. "

write-host ""

write-host "*"

write-host "===== "

write-host ""

$OU = "CN=ADTFSGROUPS,OU=TFSUsers,DC=lumitulib,DC=com"

$CSVPaTH=""

do
{
try
{
        $CSVPaTH = read-host "Please enter the CSV file path"

        $checkPath=Import-csv $CSVPaTH
}

catch [exception]

{
```

```

        write-host "The path you provided is not valid. Please enter a valid
path" foregroundcolor red

        $CSVPath=""

    }

}

while($CSVPath -eq "")

write-host "Importing the CSV file data..." -foregroundcolor yellow

$DataSheet = Import-CSV $CSVPath -delimiter ';'

if($DataSheet -eq $null)

{

        write-host "Error. check the CSV file path, or check that the file
is not empty. Processs terminated"

        $end = read-host "press enter to quit"

        exit

}

        $duplicate = $false

        Write-host "Checking for the groups in active directory database..."
-foregroundcolor yellow

foreach($row in $DataSheet)

{

try

{

if($row.GroupName -ne "" -and $row.GroupName -ne $null)

```

```

{
    $name=$row.GroupName

    $group = dsquery group -name $name

    if($group -ne $null)
    {
        $duplicate=$true

        write-host "The group " $name " already exists in Active Directory" -foregroundcolor green
    }
}

}

catch [exception]
{
    write-host "Checking failure" -ForegroundColor red
}

}

if($duplicate -eq $false)
{
    write-host "Group check successfull. Proceeding with creating the groups..." -foregroundcolor yellow

    foreach ($row in $DataSheet)
    {
        try
        {

```

```

$name=$row.groupName

$samAccount=$row.groupName

$displayName=$row.groupName

$path=$OU

$memberOf=$row.memberOf

if($name -ne "" -and $name -ne $null)
{
    write-host "Creating the group " $name "..." -
foregroundcolor blue

    New-ADGroup -name $name -SamAccountName $samAccount -
GroupCategory "Security" -GroupScope "Global" -Path $path

    Add-AdgroupMember $memberOf $name

    write-host "Group " $name " created"

}

}

catch [exception]
{
    write-host"Group creating failure" -foregroundcolor red
}

}

else
{

```

```
        write-host "The groups or some of the groups you are trying to create already exist in AD! The command will be terminated. Check the list above for more information"
```

```
        $end2 = read-host "Press enter to quit"
```

```
        exit
```

```
    }
```

```
    $array=Import-CSV $CSVPath -delimiter ';' 
```

```
    $group=$array[0].groupName
```

```
    write-host
```

```
        write-host "The members (groups) of the group "$group" are:" - foregroundcolor yellow
```

```
function get-Hierarchy($group)
```

```
{
```

```
try
```

```
{
```

```
$groups=get-adgroupmember $group | sort-object objectclass -descending
```

```
    foreach($item in $groups)
```

```
    {
```

```
        if($item.objectclass -eq "group")
```

```
        {
```

```
            write-host $item.objectclass,":",$item.name;
```

```
            get-hierarchy $item.name
```

```
        }
```

```
    }  
  }  
  catch [exception]  
  {  
    write-host "Error retrieving the groups" -foregroundcolor red  
  }  
}  
  
get-Hierarchy $group  
  
$done=read-host "Done"
```

## Appendix 8: Create users and assign them to teams

```
import-module ActiveDirectory

echo "===== "

echo "*"

echo "This PS command is used to create AD users,"

echo "assign them to a course implementation and"

echo "assign them to their respective teams (if applies)"

echo "*"

echo "===== "

write-host

do

{

$OU=read-host "Select a degree program for the new users (bit / tiko)"

    if($OU -ne "bit" -and $OU -ne "tiko")

    {

        write-host "you should select bit or tiko" -foregroundcolor red

    }

}

while($OU -ne "bit" -and $OU -ne "tiko")

write-host

do

{
```

```

try
{
$implementation=read-host "Enter a course implementation (e.g. eba-10)"

    if($implementation -eq "" -or $implementation -eq $null)
    {
        write-host "Course implementation is a mandatory field and cannot be
empty" -foregroundcolor red
    }
    else
    {
        write-host "Checking that the implementation exists in AD..." -
foregroundcolor yellow

        $exist=dsquery group -name $implementation

        if($exist -eq $null)
        {
            write-host "Non existant implementation (Check for
spelling mistakes or that the implementation exists in AD)" -foregroundcolor
red
        }
        else
        {
            write-host "Implementation "$implementation" available
in AD" -foregroundcolor green
        }
    }
}

```

```

        }
    }
    catch [exception]
    {
        write-host "Implementation check failed" -foregroundcolor red
    }
}

while($implementation -eq $null -or $exist -eq $null)

write-host

$CSVPaTH=""

do

{

try

{

        $CSVPaTH = read-host "Please enter the CSV file path"

        $checkPath=Import-csv $CSVPaTH

    }

catch [exception]

{

        write-host "The path you provided is not valid. Please enter a valid
path" -foregroundcolor red

        $CSVPaTH=""

    }
}

```

```

}

while($CSVPath -eq "")

write-host "Importing the CSV file data..." -foregroundcolor yellow

$DataSheet = Import-CSV $CSVPath -delimiter ';'

$duplicate = $false

write-host "Checking the users in Active Directory database for duplicates..."
-foregroundcolor yellow

foreach($row in $DataSheet)

{

try

{

        $studentNumber="a"+$row.Student

        $name=$row.Name

        $user=get-aduser -filter {samaccountname -eq $studentNumber} -
searchbase "DC=lumitulib,DC=com"

        if($user -ne $null)

        {

                $duplicate=$true

                write-host "User "$studentnumber" with name: "$name",
already exists in AD" -foregroundcolor cyan

        }

}

catch [exception]

{

```

```

        Write-host "User checking failure" -foregroundcolor -red
    }
}

if($duplicate -eq $false)
{
    write-host "User check successfull. No user duplicates in AD. Proceeding with team checking..." -foregroundcolor yellow
}

else
{
    write-host "Some user(s) already exist in AD! the existing users will be added to their teams if applicable" -foregroundcolor yellow

    $end=read-host "Press enter to continue"
}

write-host "Checking for the team existance in AD and team per user..." -foregroundcolor yellow

foreach($row in $DataSheet)
{
    try
    {
        $team=$row.Option

        $name=$row.Name

        $snumber=$row.Student
    }
}

```

```

        if($snumber -ne "" -and $snumber -ne $null -and $name -ne "" -and
$name -ne $null)
        {
            if($team -eq "" -or $team -eq $null)
            {
                write-host "The user "$snumber" with name
"$name" does not have a team" -foregroundcolor Cyan
            }
            if($team -ne "" -and $team -ne $null)
            {
                $group = dsquery group -name $team
                if($group -eq $null -or $group -eq "")
                {
                    $end=read-host "The team "$team"
does not exist. Press enter to continue anyway or enter (n) to stop the com-
mand"
                }
            }
            if($end -eq "n")
            {
                exit
            }
        }
    }
}
catch [exception]

```

```

{
    Write-host "User checking failure" -foregroundcolor red
}

}

write-host "Team checking in AD done. Proceeding with creating users..." -
foregroundcolor yellow

write-host

$global:ExportCsv = @( )

foreach ($row in $DataSheet)
{
    try
    {
        $random=new-object system.random

        $samaccount="a"+$row.Student

        $name=$row.Name

        $description=$row.Group

        $password=$samaccount+($random.next(1000,9999))

        $dprogram=$row.Programme

        $path="OU=$OU,OU=Students,OU=TFSUsers,DC=lumitulib,DC=com"

        $DomainGroup="Domain users"

        $teamGroup=$row.option

        $courseImpGroup=$implementation

        if($row.Student -ne "" -and $row.Student -ne $null -and $name -ne "" -and $name
        -ne $null)

```

```

{

    $getUser= get-aduser -filter 'samaccountname -eq $samaccount' -
searchbase "dc=lumitulib,dc=com"

    if($getUser -eq "" -or $getUser -eq $null)

    {

        write-host "Creating the user "$samaccount"..." -
foregroundcolor yellow

        New-ADUser -name $name -samaccountname $samaccount -
Displayname $name -Description $description -path $path -enabled $true -
accountpassword (ConvertTo-secureString $password -asplaintext -force) -
Changepasswordatlogon $true

        Add-AdgroupMember $implementation -member $samaccount

    if($teamGroup -ne "" -and $teamGroup -ne $null)

    {

        Add-adgroupmember $teamGroup -member $samaccount

    }

    else

    {

        write-host "The user "$samaccount" with name: "$name"
has no team"

        write-host "You can assign later a team for the user
"$samaccount" in the CSV file and run this PS command again"

```

```

    }

    write-host "User " $samaccount " with name " $name " created suc-
cessfully" -foregroundcolor green

    $hash = @{Password = $password ; Id = $samaccount}

    $obj = New-Object psobject -property $hash

    $global:ExportCsv += $obj
}

else
{

    Write-host "The user "$samaccount" already exists in AD"
-foregroundcolor yellow

    write-host "Adding the user to the course implementation
and team if applicable..."

    try
    {

        Add-AdgroupMember $implementation -member $samaccount
    }

    catch [exception]
    {

        write-host "Adding user "$samaccount" to the implementa-
tion "$implementation" failed. Check that the user is not already a member of
the implementation: "$implementation -foregroundcolor -red
    }
}

```

```

        if($teamGroup -ne "" -and $teamGroup -ne $null)
        {
            try
            {
                Add-adgroupmember $teamGroup -member $samaccount
count
            }
            catch [exception]
            {
                write-host "Adding user "$samaccount" to the
team "$teamGroup" failed. Check that the user is not already a member of the
team: "$teamGroup -foregroundcolor -red
            }
        }
    else
    {
        write-host "The user "$samaccount" with name: "$name"
has no team"
        write-host "You can assign later a team for the user
"$samaccount" in the CSV file and run this PS command again"
    }
}
}
}

```

```

}

catch [exception]

{

    write-host "User creating failure" -foregroundcolor red

    write-host "Operation failed at user" $samaccount". User might exist
already" -foregroundcolor red

}

}

Write-host

write-host "Importing the usersIDs and passwords to a text file" -
foregroundcolor yellow

try

{

    Write-host "List of users and passwords will be created and saved in
'C:\ScriptShared\Output' under the name UsersPassList.csv "

    $global:ExportCsv | Export-Csv -path
"C:\ScriptShared\Output\UsersPasswList.csv" -Encoding "unicode" -
NoTypeInfoInformation -force

    Write-host "File created" -foregroundcolor green

}

catch [exception]

{

    write-host "Users list file creation failed" -foregroundcolor red

}

$done=read-host "Done"

```

## Appendix 9: Removing a course structure

```
import-module ActiveDirectory

echo "======"

echo "*"

echo "This command will remove a group (user input) and all"

echo "its member groups"

echo ""

echo "*"

echo "======"

write-host

do

{

try

{

    $group=read-host "Enter the group to be removed"

    if($group -eq $null)

    {

        write-host "Group name input is mandatory" -

foregroundcolor red

    }

    else

    {
```

```

        write-host "Checking that the group exists in AD..." -
foregroundcolor yellow

        $exist=dsquery group -name $group

        if($exist -eq $null)
        {
            write-host "Non existant group (Check for
spelling mistakes or that the group exists in AD)" -foregroundcolor red
        }
        else
        {
            write-host "Group " $group " available in AD"
        }
    }
}

catch [exception]
{
    write-host "Group check failed" -foregroundcolor red
}

}

while($group -eq $null -or $exist -eq $null)

write-host "Proceeding with removing the group "$group" and its member
groups..." -foregroundcolor yellow

$global:arrayOfGroups = @()

```

```

function get-hierarchy ($group)
{
    try
    {
        $gs = get-adgroupmember $group | sort-object objectclass

        if($gs -ne $null)
        {
            foreach($item in $gs)
            {
                if($item.objectclass -eq "group")
                {
                    $global:arrayOfGroups += $item.name

                    get-hierarchy $item.name
                }
            }
        }

        catch [exception]
        {
            write-host "Error retrieving the groups" -
foregroundcolor red
        }
    }
}

```

```

get-hierarchy ($group)

write-host "The following groups will be removed form AD in addition to "$group
-foregroundcolor yellow

if($global:arrayOfGroups -ne $null)
{
    $global:arrayOfGroups
}
else
{
    write-host "There are no groups available. Press enter to end the
process" -foregroundcolor red

    $stop=read-host

    exit
}

$remove = read-host "press enter continue the operation or enter (n) to cancel"
if($remove -ne "n")
{
    write-host "removing the group "$group"..."

    remove-adgroup $group -confirm:$false

    write-host "$group" successfully removed" -foregroundcolor green

    for($i=0 ; $i -lt $global:arrayOfGroups.length ; $i++ )
    {
        write-host "removing the group "$global:arrayOfGroups[$i]"..."
    }
}

```

```
        remove-adgroup $global:arrayOfGroups[$i] -confirm:$false

        write-host $global:arrayOfGroups[$i]" successfully removed" -
foregroundcolor green

    }

}

else

{

    $stop=read-host "Opration cancelled by the user. Press enter to stop
the process"

    exit

}

$end = read-host "done"
```

## Appendix 10: Scenarios performed with a teacher account

<b>Scenario Id: S01</b>	
<b>Scenario name: access a domain workstation</b>	
<b>Purpose</b>	Checking that a teacher can access a work station belonging to the domain
<b>Input data</b>	Teacher account's user Id and password (to be provided at the time of the test)
<b>Before the test</b>	N/A
<b>Steps</b>	In the work station: at the user log in view enter the teacher's user Id and password and then press the arrow to log in
<b>Expected result</b>	Successful login
<b>Test result</b>	

<b>Scenario Id: S02</b>	
<b>Scenario name: access the domain controller locally and access server manager</b>	
<b>Purpose</b>	Checking that a teacher has access rights to the domain controller locally, but cannot access the server manager
<b>Input data</b>	Teacher account's user Id and password (to be provided at the time

	of the test)
<b>Before the test</b>	N/A
<b>Steps</b>	In the domain controller: at the user log in view enter the teacher's user Id and password and then press the arrow to log in > start > administrative tools > server manager
<b>Expected result</b>	<ul style="list-style-type: none"> <li>• Successful log in</li> <li>• Access to the server manager denied</li> </ul>
<b>Test result</b>	

<b>Scenario Id: S03</b>	
<b>Scenario name: access the domain controller with remote desktop and creating course structure and users with Power Shell scripts</b>	
<b>Purpose</b>	Checking that a teacher can access the domain controller remotely from a domain work station, access the shared folder for scripts, run scripts to create users and groups, and access the TFSUsers OU with Active Directory Users and Computers
<b>Input data</b>	<ul style="list-style-type: none"> <li>• Student data CSV file (to be provided at the time of the test)</li> <li>• Course structure CSV file (to be provided at the time of the test)</li> <li>• Teacher account's user Id and password (to be provided at the time of the test)</li> </ul>

	<ul style="list-style-type: none"> <li>• Domain controller host address</li> </ul>
<b>Before the test</b>	<ul style="list-style-type: none"> <li>• Student data CSV file path: C:\ScriptShared\students.csv</li> <li>• Course structure CSV file path: C:\ScriptShared\courses.csv</li> <li>• Power shell scripts folder mapping path: \\lunitulib.com\ScriptShared</li> </ul>
<b>Steps</b>	<p>In the work station: at the user log in view enter the teacher's user Id and password and then press the arrow to log in &gt; right click in start view &gt; All apps &gt; remote desktop connection &gt; in computer field enter the domain controller host name (NetDevDc.lunitulib.com) &gt; in password enter the teacher's password &gt; ok &gt; start &gt; computer &gt; local disk (C) &gt; Script shared &gt; ReadyScripts &gt; Create_Groups_AdUsers_FromCSV</p> <ul style="list-style-type: none"> <li>• Run with Power Shell: Create_Group_Structure.ps1 (follow the instructions in the script)</li> <li>• Run with Power Shell Create_Users_AssignToTeams.ps1 (follow the instructions in the script)</li> </ul> <p>Check that the scripts were successfully run: Start &gt; Administrative tools &gt; active directory users and computers &gt; lunitulib.com &gt; TFSUsers &gt;</p> <ul style="list-style-type: none"> <li>• ADTFSGROUPS</li> <li>• Students &gt; BIT or Students &gt; BIT</li> </ul>

<b>Expected result</b>	<ul style="list-style-type: none"> <li>• The group structure created in the ADTFSGROUPS container</li> <li>• The Students' accounts created in the selected OU (BIT or TIKO)</li> <li>• A CSV file with student user Ids and passwords created at C:\ScriptShared\Output\UsersPasswList.csv</li> </ul>
<b>Test result</b>	

<b>Scenario Id: S04</b>	
<b>Scenario name: access the development server locally</b>	
<b>Purpose</b>	Checking that a teacher can access the development server locally
<b>Input data</b>	Teacher account's user Id and password (to be provided at the time of the test)
<b>Before the test</b>	N/A
<b>Steps</b>	In the development server: at the user log in view enter the teacher's user Id and password and then press the arrow to log in
<b>Expected result</b>	Successful log in
<b>Test result</b>	

<b>Scenario Id: S05</b>	
<b>Scenario name: access the development server with remote desktop and Create a team project collection with TFS console</b>	
<b>Purpose</b>	Checking that a teacher user has the needed access rights and privileges to access the development server remotely, using the TFS console, creating team project collections
<b>Input data</b>	<ul style="list-style-type: none"> <li>• Teacher account's user Id and password (to be provided at the time of the test)</li> <li>• Team project collection name: eba_10</li> <li>• Development server host address</li> </ul>
<b>Before the test</b>	N/A
<b>Steps</b>	In the work station: In the work station: at the user log in view enter the teacher's user Id and password and then press the arrow to log in > right click in start view > All apps > remote desktop connection > in computer field enter the domain controller host name (tfs.lumitulib.com) > in password enter the teacher's password > ok > start > all programs > Microsoft Team Foundation Server 2010 > Team Foundation Administration console > expand DevServer > Application tier > team project collections > create collection > in name field write: eba_10 > next > Sql server instance: DevServer and select Create a new database for this collection > next > next > next > verify (all the test have to be passed) > create > next > close.

<b>Expected result</b>	<ul style="list-style-type: none"> <li>• Login to the development server successful</li> <li>• Access to the TFS console successful</li> <li>• Team project collection created successfully</li> </ul>
<b>Test result</b>	

<b>Scenario Id: S06</b>	
<b>Scenario name: connect to TFS, create a team project, assign a team to it and give the team the needed rights over the team project</b>	
<b>Purpose</b>	Checking that a teacher can create team projects, assign users or groups to them and grant them permissions upon the team project
<b>Input data</b>	<ul style="list-style-type: none"> <li>• Teacher account user Id and password (to be provided at the time of the test)</li> <li>• Development server host address</li> <li>• Team project name: Scrum application_team1</li> <li>• Group name (team): eba-10-01</li> <li>• TFS group name: Contributors</li> </ul>
<b>Before the test</b>	Scenario Id S05 performed successfully
<b>Steps</b>	<ul style="list-style-type: none"> <li>• In the work station: In the work station: at the user log in view</li> </ul>

enter the teacher's user Id and password and then press the arrow to log in > right click in start view > All apps > remote desktop connection > in computer field enter the domain controller host name (tfs.lumitulib.com) > in password enter the teacher's password > ok

- Connecting to TFS: Start > all programs > Microsoft Visual Studio 2010 > team > connect to Team Foundation Server > servers > add > in name or URL of Team Foundation Server: tfs.lumitulib.com and don't make changes the other fields > ok > close > select eba\_10 > connect.
- Upload Scrum template: right click tfs.lumitulib.com\eba\_test > Team project collection settings > process template manager > upload > C:\Program Files (x86)\Microsoft\Microsoft Visual Studio Scrum 1.0\Process Template > select folder > ok > close.
- Creating the team project: right click tfs.lumitulib.com\eba\_test > new team project > under what is the name of the project? Enter eba\_10\_01 > next > select Microsoft visual studio Scrum 1.0 > next > select create an empty source control folder > next > finish > close.
- Adding a team to the team project: right click the team project (eba\_10\_01) > team project setting > group membership > select Contributors > properties > Select windows user or group > Add > Advanced > Find now > Select the team you want to add (eba\_10\_01) > ok > ok > ok > close.

<b>Expected result</b>	<ul style="list-style-type: none"> <li>• Team project created successfully</li> <li>• Successful connection to TFS</li> <li>• Team project created successfully</li> <li>• The group (team) assigned to the team project successfully</li> <li>• The access right and privileges upon the team project set successfully (team added to the team project contributors successfully)</li> </ul>
<b>Test result</b>	

## Appendix 11: Scenarios performed with a student account

<b>Scenario Id: S07</b>	
<b>Scenario name: access the domain controller locally</b>	
<b>Purpose</b>	Checking that a student cannot access the domain controller
<b>Input data</b>	Student account's user Id and password (to be provided at the time of the test)
<b>Before the test</b>	N/A
<b>Steps</b>	In the domain controller: at the user log in view, enter the student's user Id and password and then press the arrow to log in
<b>Expected result</b>	Unsuccessful log in
<b>Test result</b>	

<b>Scenario Id: S08</b>	
<b>Scenario name: access the development server locally</b>	
<b>Purpose</b>	Checking that a student cannot access the development server
<b>Input data</b>	Student account's user Id and password (to be provided at the time of the test)
<b>Before the</b>	N/A

<b>test</b>	
<b>Steps</b>	In the development server: at the user log in view, enter the student's user Id and password and then press the arrow to log in
<b>Expected result</b>	Unsuccessful log in
<b>Test result</b>	

<b>Scenario Id: S09</b>	
<b>Scenario name: access the domain controller with remote desktop</b>	
<b>Purpose</b>	Checking that a student can access a work station belonging to the domain but cannot access the domain controller remotely
<b>Input data</b>	<ul style="list-style-type: none"> <li>• Student account's user Id and password (to be provided at the time of the test)</li> <li>• Domain controller host address</li> </ul>
<b>Before the test</b>	N/A
<b>Steps</b>	In the work station: In the work station: at the user log in view enter the student's user Id and password and then press the arrow to log in > right click in start view > All apps > remote desktop connection > in computer field enter the domain controller's host name (NetDevDc.lumitilib.com) > in password enter the student's password > ok.

<b>Expected result</b>	<ul style="list-style-type: none"> <li>• Successful log in to the work station</li> <li>• Unsuccessful log in to the domain controller through remote desktop</li> </ul>
<b>Test result</b>	

<b>Scenario Id: S010</b>	
<b>Scenario name: access the development server with remote desktop</b>	
<b>Purpose</b>	Checking that a student can access a work station belonging to the domain but cannot access the domain controller remotely
<b>Input data</b>	<ul style="list-style-type: none"> <li>• Student account's user Id and password (to be provided at the time of the test)</li> <li>• Domain controller host address</li> </ul>
<b>Before the test</b>	N/A
<b>Steps</b>	In the work station: at the user log in view enter the student's user Id and password and then press the arrow to log in > right click in start view > All apps > remote desktop connection > in computer field enter the development server's host name (tfs.lumitulib.com) > in password enter the student's password > ok.
<b>Expected result</b>	<ul style="list-style-type: none"> <li>• Successful log in to the work station</li> <li>• Unsuccessful log in to the development server through remote desktop</li> </ul>

<b>Test result</b>	
--------------------	--

<b>Scenario Id: S011</b>	
<b>Scenario name: access a domain work station, connect to TFS, and see own team projects</b>	
<b>Purpose</b>	Checking that a student can connect to TFS and see the projects where he is a member.
<b>Input data</b>	<ul style="list-style-type: none"> <li>• Student account's user Id and password (to be provided at the time of the test)</li> <li>• Development server host address</li> </ul>
<b>Before the test</b>	Scenarios S05 and S06 performed successfully
<b>Steps</b>	In the work station: at the user log in view enter the student's user Id and password and then press the arrow to log in > right click in start view > All apps > Visual Studio 2012 > Team > connect to Team Foundation Server > Servers >Add > development server's host name: tfs.lumitilib.com (don't change the other default values )> ok > select one team project > connect
<b>Expected result</b>	<ul style="list-style-type: none"> <li>• Successful connection to TFS</li> <li>• Team projects where user is member are displayed</li> </ul>
<b>Test result</b>	

<b>Scenario Id: S012</b>	
<b>Scenario name: map “ScriptShared” folder</b>	
<b>Purpose</b>	Checking that students cannot access the scripts shared folder
<b>Input data</b>	<ul style="list-style-type: none"> <li>• Student account’s user Id and password (to be provided at the time of the test)</li> <li>• Power shell scripts folder mapping path: \\lunitulib.com\ScriptShared</li> </ul>
<b>Before the test</b>	N/A
<b>Steps</b>	In the work station: at the user log in view enter the student’s user Id and password and then press the arrow to log in > right click computer > Map network drive > in folder enter: \\lunitulib.com\ScriptShared
<b>Expected result</b>	Unsuccessful mapping
<b>Test result</b>	

## Appendix 12: Requirements list

Task Id	Priority	Task name
1	1000	Installing the domain controller
2	1010	Checking the hardware requirements
3	1020	Installing Windows Server 2008 R2
4	1030	Installing Active Directory Domain Services server role
5	1040	Installing DNS server role
6	1050	Installing Group Policy Manager feature
7	1060	Installing .NET framework feature
8	1070	Setting the firewall
9	1100	Installing the development server
10	1110	Installing Windows Server 2008 R2
11	1120	Checking the hardware requirements and software requirements for TFS 2010
12	1130	Installing the required server roles for TFS 2010
13	1140	Installing the required features for TFS 2010
14	1150	Installing SQL server 2008 R2
15	1160	Installing Windows SharePoint Services WSS 3.0
16	1170	Creating and setting the access level of the required user ac-

		counts for TFS 2010
17	1180	Installing TFS 2010
18	1190	Installing the build server
19	1200	Installing antivirus software
20	1300	Installing the clients
21	1310	Checking the hardware requirements for Windows 8, Visual Studio 2012 and Windows Phone 8 SDK
22	1320	Installing Windows 8 along with Windows 7
23	1330	Installing Visual Studio for Windows 8
24	1340	Installing Windows Phone 8 SDK
25	1350	Installing the Scrum process template for VS 2010 and VS 2012
26	1400	creating the business structure in the domain controller
27	1410	Creating the needed OU structure
28	1420	Creating the custom AD group structure
29	1430	Creating the needed security AD groups
30	1440	Delegating the administration tasks of the business structure to the relevant AD group
31	1450	Granting the needed rights and privileges to the business structure administrators in order to be able to perform all the needed administration tasks

32	1500	Setting the rights and privileges for the domain users
33	1510	Accessing and using securely the domain computers
34	1520	No local access to the domain controller or to the development server by normal users
35	1530	Restricted remote access to the development server
36	1540	TFS users can view the reporting services database of the team projects they belong to
37	1600	Creating Scripts for maintenance and to automate user management tasks
38	1610	Create users from a .CSV file type and assign them to a course implementation and one team
39	1620	Create a course structure
40	1630	View users per course implementation / per team
41	1640	View users per entry year
42	1650	View teachers per course
43	1660	Import a user list with their passwords to a text file
44	1661	Remove outdated user accounts
45	1670	Create a shared folder for the scripts and set the access privileges to the file
46	1680	Setting remote desktop
47	1690	Remote desktop security setting

48	1700	Testing the system
49	1710	Assessing the security with security scanning tool
50	1720	Testing the user access in the domain
51	1730	Testing the user access in TFS
52	1800	Deploying the system to internet
53	1900	Implementing automated project deployment with Visual Studio to IIS server
54	1910	Creating a content provider in IIS with one account to be used by all
55	1920	Securing and isolating the other parts of IIS from the created sand box (content provider)