Mika J. Mäkelä

Improving Deployment Process by Using Automated Deployments for COTS Products

Helsinki Metropolia University of Applied Sciences

Master's Degree

Information Technology

Master's Thesis

11.5.2016

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Mika J. Mäkelä Improving Deployment Process by Using Automated Deployments for COTS Products 60 pages 11.5.2016
Degree	Master of Engineering
Degree Programme	Information Technology
Instructor(s)	Ville Jääskeläinen, Principal Lecturer

Deployment process is one of the key phases of any software development project. During the deployment process the implemented software solution is deployed to the IT-environment and prepared for the customer's use. Fluency of the deployment is in a key role in the success of the delivery project and customer satisfaction. Thus, development of the deployment process has an important role when improving the conditions for the success of the whole delivery project.

The purpose of this thesis was to collect and analyse information to improve the deployment process. This was done by researching theoretical background and technical options for the automated deployments. The thesis focuses on the deployments of the COTS (Commercial off-the-shelf) software products and Esri ArcGIS Geographic Information Systems software. In this thesis technical options were tested and compared, such as Python scripting and Chef deployment framework. The research is scoped only to the Esri ArcGIS enterprise products.

The study consists of four parts. The theoretical section explores the background information related to the deployment process and presents architectural patterns of the GIS Systems. The second part defines the state of the current software development process based on interviews and practical experiences. In the third part different kind of technical options are compared to find an optimal deployment approach. The last part presents a pilot implementation of the chosen deployment process.

The findings of this research indicate that the automation of the data model and software installation deployment is a key factor in development of the deployment process. The results suggest that the automation of the Esri ArcGIS software deployment increases the effectivity and reliability of the process. The automatization of the deployment process is most beneficial to complex large environments and multiple deployment releases.

Keywords	Deployment, Installation, Software, COTS, Esri, ArcGIS, GIS

Tekijä Otsikko Sivumäärä Aika	Mika J. Mäkelä Improving Deployment Process by Using Automated Deployments for COTS Products 60 sivua 11.5.2016
Tutkinto	Master of Engineering (Insinööri YAMK)
Koulutusohjelma	Information Technology (Informaatiotekniikka)
Ohjaajat	Ville Jääskeläinen, yliopettaja

Sovelluskokonaisuuden käyttöönotto on yksi ohjelmistoprojektin keskeisistä vaiheista. Käyttöönotossa sovelluskokonaisuus asennetaan IT-ympäristöön ja se valmistellaan asiakkaan käyttöä varten. Käyttöönoton sujuvuus on keskeisessä roolissa toimitusprojektin onnistumisen ja asiakastyytyväisyyden kannalta. Käyttöönotto onkin tärkeä vaihe, jota kehittämällä voidaan olennaisesti parantaa toimitusprojektin onnistumisen edellytyksiä.

Tämän opinnäytetyön tavoitteena on antaa välineitä teknisen käyttöönottoprosessin kehittämiseen keräämällä ja analysoimalla taustatietoa sekä tarkastelemalla teknisiä vaihtoehtoja käyttöönoton automatisoinnille. Opinnäytetyössä keskitytään COTS-sovellusten (Commercial Off-The-Shelf) käyttöönottoon sekä Esri ArcGIS -paikkatietosovelluksiin. Työssä tutkitaan ja koestetaan teknisiä vaihtoehtoja, kuten Python-scriptausta sekä Chef-käyttöönottotyökalua. Työ on rajattu Esri ArcGIS -paikkatietosovelluksiin sekä -palvelinratkaisuihin.

Tutkimus jakaantuu neljään osaan. Kirjallisessa katsauksessa tarkastellaan yleisesti käyttöönottoprosesseja ja esitellään paikkatietojärjestelmien arkkitehtuuria. Toisessa osassa analysoidaan käyttöönottoprosessin nykytila perustuen haastatteluihin sekä empiirisiin havaintoihin käytännön työssä. Tutkimuksen kolmannessa osassa vertaillaan teknisiä ratkaisuja toimivimman käyttöönottoprosessin valitsemiseksi. Viimeisessä osassa esitellään tutkimuksen osana toteutettu tekninen käyttöönottokokeilu, jolla kokeiltiin tutkimuksessa valittua käyttöönottoprosessia käytännössä.

Tutkimuksen perusteella erityisesti tietomallin ja sovellusasennusten käyttöönoton automatisointia voidaan pitää käyttöönottoprosessin keskeisenä kehityskohteena. Tutkimus osoittaa, että Esri ArcGIS -sovellusten käyttöönoton automatisoinnilla voidaan parantaa käyttöönottoprosessin tehokkuutta sekä toimivuutta. Käyttöönottoprosessin automatisoinnista on eniten hyötyä kompleksisissa ja laajoissa asiakkaan IT-ympäristöissä.

Keywords	Käyttöönotto, Asennus, Sovelluskehitys, COTS, ArcGIS, GIS

Contents

List of figures

1 Introduction

The software deployment phase is one of the final steps of a software development project. It contains all the activities that make a software system available for use. The deployment process can vary considerably depending on the scenario and the complexity of the system. Mäntylä et Vanhanen [1] summarized that software deployment is a multifaceted topic, consisting of activities such as customer interaction, integrations to the other systems, product configuration, and testing.

In a COTS (Commercial off-the-shelf) retail business, such as with the case company, the deployment process has a significant role. The deployment process in generally is a key issue that a customer faces when buying a new software. A weakly managed deployment process will delay the project schedule and it will affect the project revenue. Therefore, improving the deployment process is one of the key issues when targeting a higher customer satisfaction and a better project revenue. In a COTS retail business, the deployment process is actually almost the only part that can be improved when thinking of customer expectations.

The deployment process has a clear connection to the quality of the software. Mockus et al. [2] proposed that the perceived quality can vary a lot just depending on the manner of deployment. This indicates the importance of the deployment strategy when managing customer perceived quality, especially when a customer's expectations are high. Jansen and Brinkkemper [3] suggest that smooth deployments are essential for increasing the customer base of software product companies. That is an important conclusion to take into account especially with companies that are dealers of the COTS software products.

This thesis focuses on the deployments of the COTS (Commercial off-the-shelf) software products and Esri ArcGIS Geographic Information Systems software. The thesis was done for Esri Finland, a retailer of the Esri ArcGIS software products in Finland.

1.1 Business Problem

The business challenge of this study is to find options to improve the deployment process. To develop the processes to be more efficient and formal, there is a need for the

background information and knowledge of how the procedures could be improved. The Esri platform provides a strong set of tools to organize an automated deployment process but there is a lack of knowledge. There is no formal knowledge on what tools are available, how to use the tools and what issues and benefits the tools provide. As a summary, possibilities and benefits of the automated deployments are unclear and the question when to choose an automated process rather than a manual one is open.

1.2 Case Company Background

Esri Finland, later referred to as the case company, is an official retailer and expert of Esri ArcGIS technology in Finland. The primary business of the case company is Esri software license selling for the Finnish business customers. Customer companies are from the private and public sectors. To support license selling, the case company offers services such as consulting, project delivery, product support, maintenance and training.

The head of the case company, Esri Inc. (founded in 1969 US) is the global market leader in a location and geographical information systems. Esri software products are widely used in the business sector worldwide. From the product development perspective, Esri Inc. is responsible for the core product development. The role of the local distributors, such as Esri Finland is license selling and they are not involved in core product development.

Many of the customer projects include a deployment phase. Deployment, in this study means software installations, configurations and data in the different terms. Presently, there has been plans to develop more product style packages, including services and software deployments. This kind of packaging brings up a need to get rid of the manual work and to develop more sophisticated deployment models. Automating the deployment process is one way to fulfill that need.

1.3 Objective

The objective of this thesis is to provide background information to improve the software deployment process of the customer projects in the case company. The aim is to provide theoretical information to improve the quality of the deployment process and evaluate

technical options and tools available for the automated deployments. Finally, to show the benefits of the script automation by creating a deployment package using selected Esri products.

The research question for this thesis is:

How to improve the COTS software deployment process by using automated deployments?

The result of this study consists of a recommendation for improving the deployment process in the case company.

In this study, the current best practices of the software deployments are collected to find information of the common deployment processes and available technical options. The current state analysis was performed to evaluate the current deployment processes in the case company and to find out bottlenecks and problem areas. A deployment package was created based on the findings of the current state analysis and the best practices section. The package was used to test how the selected technological options work in practice and to collect practical knowledge of the technology.

1.4 Scope and Structure

Since deployment is a wide topic, the research only covers a limited part of it. The research is mainly focused on the software installations, configurations and data related tasks. The viewpoint is more technical than commercial. The main objective is to improve the quality of the software delivery by solving challenges and streamlining the current deployment process. The technical viewpoint pushes the research to solve the challenges by improving technical knowledge and presenting options for the automating processes. The aim of this thesis is not to provide a detailed deployment process, but rather to evaluate the current state and suggest how the deployment process could be improved.

Generally, the research is focused on the Esri GIS (Geographic Information Systems) software products and especially on COTS (Commercial Off-The-Shelf) products, i.e. mainly out of the box software that has a small amount of customizations or no customizations at all.

Within the thesis, the software delivery process and deployment process are described from the deployment point of view. The aim is to clarify and understand how the deployment process is related to the whole delivery process and clarify the sub parts of the processes. From a technical point of view the scope of the thesis is the Esri enterprise solution deployments.

This thesis first details the method and material used for the study. Section 3 discusses the best practices of the software deployments, presents the architecture patterns of the GIS systems, the COTS ideology and overviews Esri products. Section 4 analyses the current state in the case company and presents typical deployment scenarios. Section 5 analyses and evaluates technological options that can be used to automate the deployment process. Section 6, the practical part of the thesis, introduces the deployment package and its technological details. Section 7 sums up the results and discusses how the study results could be used to improve a deployment process in practice.

2 Methods and Material

This section overviews the methods and material used in the present study and explains how data was collected. The case study research method was chosen as the major method of this study. In addition, the proof-of-concept method was applied in the practical part.

2.1 Case Study

Case Study is an empirical research method. Typically, it is used for an in depth study of a particular situation rather than an extensive statistical survey. The aim is to narrow down the research from a very broad field into one easily researchable topic. The method gives indications and allows future development of a subject rather than answers a question completely. The Case Study research method is useful for testing whether scientific theories and models actually work in practice. [6]

> "The area of Software Engineering involves development, operation and maintenance of software and related artefacts. Research on software engineering is to investigate how these topics are conducted by software engineers and other stakeholders under different conditions. Software development is carried out by individuals, groups and organizations, and social and political questions are of importance for this development. That is, software engineering is a multidisciplinary area involving areas where case studies normally are conducted. This means that many research questions in software engineering are suitable for case study research." [7]

As a summary, Case Study focuses on studying phenomena in their context, especially when the border between the phenomenon and its context is unclear. This is typically true in Software Engineering. Case study offers an approach which does not need a strict border between the studied object and its environment.

In this study, Case Study approach is used as the major research method. The design of the study, the steps of process and key elements of study are the prevalent guidelines of the Case Study approach.

2.2 Proof-of-Concept

Proof-of-concept (POC) is most often used to decrease technical risk when a new or unfamiliar technology is investigated. The method is designed to prove that a risky element of the proposed architecture is feasible and to highlight any problems. Proof-of-Concept is a temporary implementation that is discarded when it has served its purpose. The danger of proof-of-concept is to start to think of it as a final production ready application. In most cases it is only a light demo that has been built to just prove that proposed design is working [4].

In this study, the Proof-of-concept approach is used to test and prove how the proposed technologies work in a practice. The aim is to get practical experience of technologies and find possible problems.

2.3 Research Design

To reach the objective, the research is divided to five main sections: 1) Theory section, 2) Current state analysis, 3) Initial draft, 4) Proof-of-concept, 5) Conclusions.

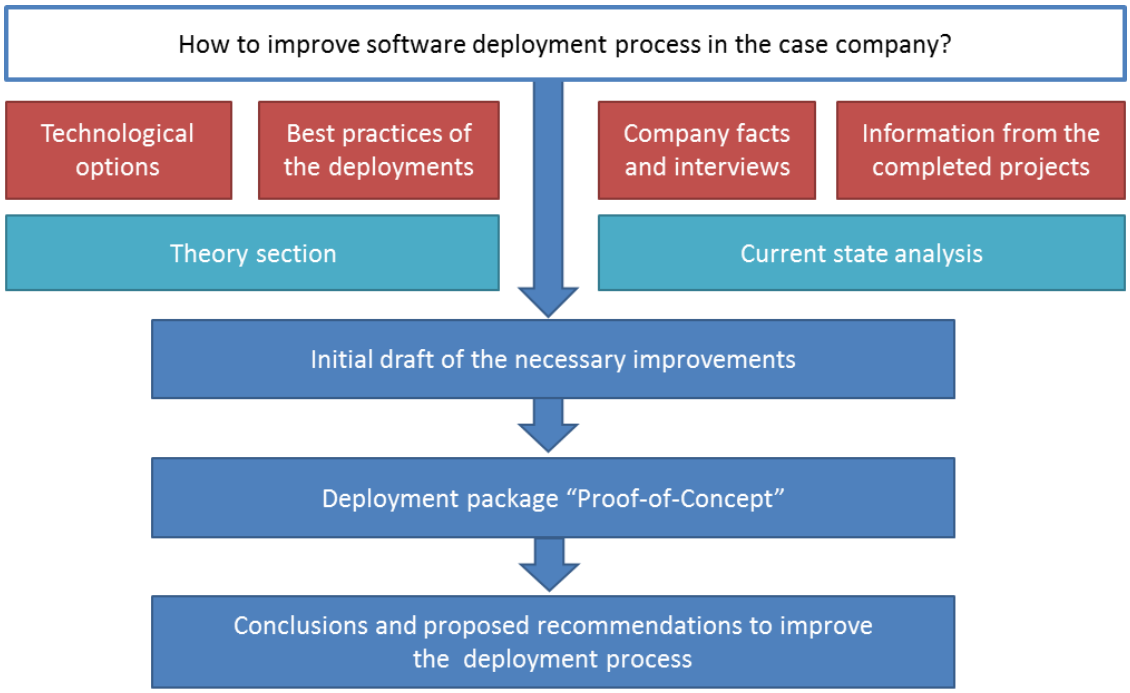

Figure 1. Research design.

Firstly, background information was collected for theory section by examining literature, other studies and technical papers. Different technical options of automated deployments were Investigate and knowledge was collected what tools and software is available and how to use them.

Secondly, current state analysis was done to establish the challenges in the current deployment process. This information was collected by interviewing colleagues (what needs and issues there are currently regarding deployments and what are the options/changes to overcome them by using script automation).

Thirdly, initial draft of the recommendations was constructed based on the interviews and theoretical background information. The available options were evaluated and those that can be useful was suggested.

Fourthly, as a practical part, the deployment package was implemented and tested how it is working in practice. Technical issues, benefits and findings was gathered.

Finally, recommendations/conclusions were summarized - whether or not it is possible to improve deployment process by using script automation and what to do next towards this goal.

3 Deployments and GIS Systems

Deployment process has an important role in a software delivery process. Deployment is the part where the customer gets the software system to use and is able for the first time to experience how the deployed system behaves in a practice. A poor managed deployment process will delay project schedule and affect the project revenue. Also, the customer will face issues and financial concerns if the ordered system is not functioning properly at the agreed time.

The deployment process has a clear connection to the quality of the software that a customer perceived. Mockus et al. [2] propose that the perceived quality can vary a lot just depending on the manner of deployment. This indicates the importance of the deployment strategy in managing customer perceived quality, especially when the customer's expectations are high. Jansen and Brinkkemper [3] suggest that smooth deployments are essential for increasing the customer base of software product companies. That is an important conclusion to take into account especially in companies that are dealers of COTS software products.

When comparing the advantages of the automated deployments to the manual approach, there are a few main aspects to mention. Talwar, V & co. [5] summarizes the benefits of the automated deployments in the following manner:

> “Automation of service deployment is beneficial for improved correctness (by reducing human errors), speed (parallelizing long-running installations), as well as for improved documentation. However, automation is achieved at an increased cost at the development time and an increase in the learning curve for administrators. This initial cost and overhead may be acceptable if the overall gains are significant and worthwhile.” [5]

Automated deployments provide many measurable benefits such as better quality, increased speed and improved documentation but there is still no clear winner among the approaches. Measurable benefits compete with the cost and time effort required to develop automated processes. This raises questions whether to use automation or not in a certain project and how many steps an automated process can save compared to the manual process?

There are scenarios where the manual approach is suitable and scenarios where the automated approach is better. The hypothesis based on the researcher's experience with actual projects is that the automated deployments are most suitable for big complex projects and the manual approach is more suitable for small simple projects.

3.1 Basics of GIS Systems

GIS (Geographic Information Systems) is computer software that links geographic information (where things are) with descriptive information (what things are). C. Dempsey [9] writes in her GIS Lounge article that GIS is a technological field that incorporates geographical features with a tabular data in order to map, analyse, and assess real-world problems. Dempsey also mentions that the real power of the GIS is the use of the spatial and statistical methods to analyse attribute and geographic information. Another overview is from Esri white paper [10] where GIS is summarized as a mature technology that began in university computer science departments in the late 1960s. The seminal idea was associating data with a geographically referenced map graphics to allow an understanding of the influence of the geography on behaviours and outcomes. As a summary, GIS is everything where location is involved.

GIS software are mostly used in the business world, private and public sectors. Typical sectors are for example defence, forestry, retailing, engineering, government, tourism etc. GIS is typically used to provide background information for decision making and give answers to business challenges where the location information can be used as a variable. Typical questions in the retailing sector are for example: where the new food market should be located? What kind of population is living in a certain area? How to choose the selection of goods for a new food market? What kind of the competitors there are near in a certain area? And maybe more complex ones such as: How much a new market affects the sales of the competitor markets in a certain area?

The above questions are examples of how GIS can be used to solve business challenges. The common denominator is the combination of the separate location information sources and other historical information. This information is then analysed and in many cases visualized on a map.

3.2 Fundamentals of Map Applications

GIS software can be roughly divided into two sub-segments based on the use: applications for the map content creation and solutions for the map content browsing. Typically map content creation is performed by using desktop applications. Map content browsing is performed by the internet based solutions and by mobile clients. This is of course not a strict division but gives some general understanding of the field.

From a technical, functional and user point of view above two segments differ quite a lot. The creation of the map content is commonly done by the GIS professionals the using applications as their main job. This requires good usability and controls in order to make the work effective. Desktop applications are typically used for that. Practically the work of creating map content could be for example drawing and digitizing features, browsing amount of the data, making spatial analyses, converting location based data from format to other, and working with different coordinate projections. This requires a lot of computing power, big data space, rich application functions and analysis capabilities.

Map content browsing on the other hand is lighter compared to content creation. Browsing is typically done by using the internet browser or the mobile client. Minimum requirement is some sort of solution that is able to present a map and allows basic controls to interact with it. Background data could be published and processed by a web server so the client is no need to handle that. [14]

3.3 Architecture Patterns of GIS Systems

The typical architecture of a GIS system varies depending on the scenario. Architecture choices can be roughly divided to two mainlines: internet based setups and local setups without internet components.

Probably the most basic local setup is a simple GIS desktop client application that uses a local file storage to store geospatial content. This is an effective setup if there is no requirement to share data outside of the local environment. The setup can be extended by adding a geodatabase to the environment where geospatial data is stored to. That will enable multiple users to connect to the same data and work simultaneously from

their desktop clients. This kind of the setup has been common especially in past decades when fast internet connections and cloud services did not yet exist.

Nowadays, requirements are commonly asking to share the information globally to get it accessible from anywhere in anytime. Organizations are big and offices are distributed geographically to different locations. The current working culture favours remote working and customers are requesting accurate data to be available without latencies or physical media. To meet these growing requirements, the architecture should be switched to the cloud services or at least to the internal network web servers.

Server and/or cloud based GIS architecture is probably the most typical approach used in modern organizations. Basically the architecture is similar compared to the traditional web application architectures but with the addition of GIS components. The principle of this architecture is that data is stored and served in a server or cloud databases and client applications access data by using network standards such as REST interfaces or HTTP. Advantages of this architecture are that architecture allows to store data in one centralized data store and easily share data to the different user groups such as internal users, extranet partners or even global internet. In addition, it is easy to allow a rich set of client applications to connect to the data by standard interfaces. This approach will significantly expand the usage of GIS data in organizations and help businesses to better decision making.

A typical server and/or cloud based architecture contains parts such as geodatabase, file storage, GIS server(s), cloud services, cloud data storage and client applications. Client applications vary from desktop clients to mobile apps or web based services. The architecture can contain only the server, only could or both of them. That is called a hybrid architecture. [25]

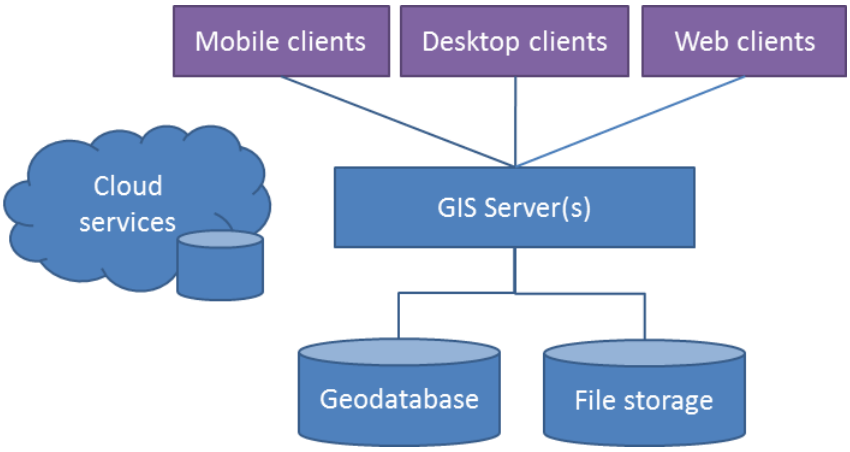

Figure 2. Typical GIS architecture.

The figure 2 present a typical server and cloud based architecture approach. Figure contains the main parts of architecture components including data store, server and client applications. From the network point of view, the components can be located ij the internal network or internet.

3.4 Commercial Off-the-Shelf Products (COTS)

The term commercial off-the-shelf (COTS) is very generic. There is a lot of discussion and several definitions exist. In general, COTS means ready packaged software products supplied by a vendor. From a customer point of view COTS software products are typically bought for a certain business need. The value of software is measured by profit.

M. Morisio et co. [23] described COTS as follows:

> "The term COTS means a software product, supplied by a vendor, that has specific functionality as part of a system – a piece of prebuilt software that is integrated into the system and must be delivered with the system to provide operational functionality or to sustain maintenance efforts". [23]

One key aspect in this description is that COTS software is part of a complete system. One individual COTS product can cover only some part of the needed functionality. To fulfil all parts, it takes multiple COTS products that are integrated to work together. This is a typical scenario in Esri deployments especially on the server side.

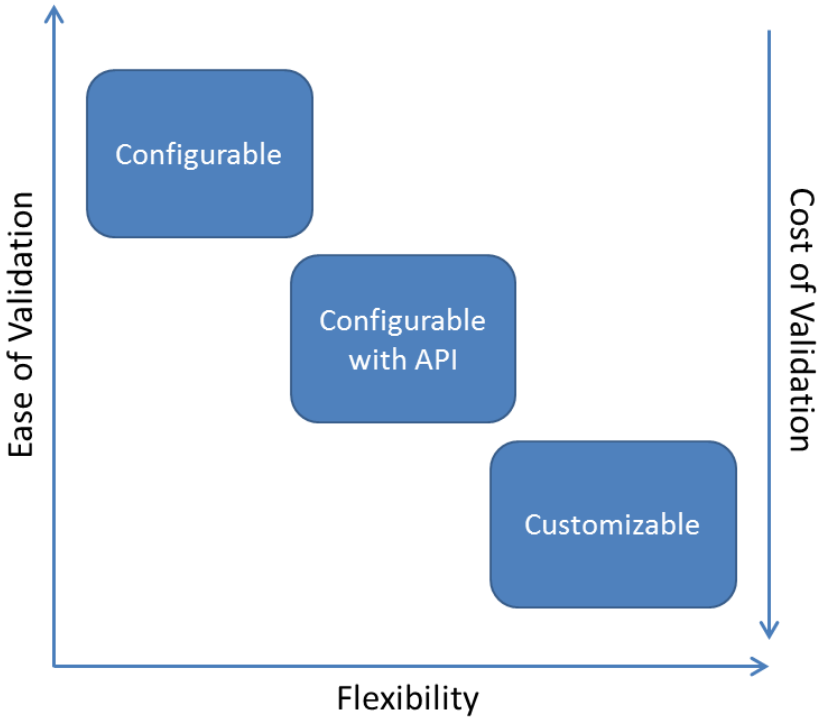

Figure 3. Description of COTS by John R. Joyce [24] in his article.

John R. Joyce [24] discusses in his article the aspects of customizations, configurations and cost of validation. He states the following,

> "Because you are altering the program code, you can make a customizable system do anything you want, where you can only select the options in a configurable system that the programmers gave you." [24]

In configurable systems there are typically limits of how widely configurations can be done. After the limit is reached, the flexibility of the system will end and going forward can be very expensive or even impossible. At this point customizable systems have no limits of a functionality since new functions can be produced when they are needed and as they are needed. Key aspect of this discussion is that these systems can be differentiated in terms of cost of validation. In generally, non-COTS custom build systems have the highest validation cost. Configurable systems have the lowest validation cost but can potentially be too limited in terms of functionality. In the middle of these two edges are configurable systems that allow implementing custom functionality via a flexible API, without need to modify their base code.

3.5 Overview of Esri ArcGIS Products

This chapter gives an overview of the Esri ArcGIS products and discusses what kind of deployment options Esri products provide.

Esri software products are typical COTS applications. They offer a wide set out of the box functionality such as configurable and customizable sections. In generally, the Esri products family contains software for the desktop, server and cloud platforms. In addition, it allows the customer to support their business processes from practical fieldwork to detailed desktop analyses. Typically, Esri users are GIS professionals who need to manage large GIS data contents and provide business value by sharing and publishing spatial data to the multiple channels.

Desktop applications

ArcGIS desktop applications are intended for GIS professionals. Applications provide a set of tools to author geospatial data such as managing, editing, creating and performing the full spectrum of analyses. The advantages of the desktop applications are the ability to manage and visualize big amount of geospatial data and run complex spatial analyses algorithms. Analyses are the key elements of the data authoring and they provide functions to turn raw data into valuable information.

The data of the desktop applications are typically stored to the geodatabase or different type of file-based data stores. Geodatabase is practically common relational database such as Oracle or MS SQL Server that has a special spatial extension installed. File-based data stores could be for example shape files, CSV formatted files or so called filegeodatabase. Data can also be stored to cloud platform. [21]

The setup of desktop applications has many difficulties and challenges from the deployment point of view. Typically, the desktop applications are installed to the end user personal computer and for that reason there are many variables to take account of such as operating systems, security settings and other local setups. It is generally known that the manual installation procedure is the most efficient way to deploy the desktop applications.

MXD document

MXD document format is a core file format used by the ArcGIS Desktop. The MXD document contains all the required information to build certain map including map data sources, map visual outlook and map layers. MXD document is created and managed by the ArcMap desktop application. MXD document can be published to the ArcGIS Server as a service.

Server products

ArcGIS for Server product family is a core element of the modern GIS platform. The server provides tools to distribute and share geospatial data using commonly known interface technologies. Web applications are the most typical scenarios for the server using. With this kind of applications, the user interface runs on web browser and server works as a backend. Other scenario to mention are mobile solutions where server works as a backend and the user interface is native application or responsive web based solution.

The server provides many kinds of core functions. The most important functions to mention are full access to the data published through the server, ability to execute spatial analysis, run address locators and provide rich set of the geometry operations. All functions are published and available via REST based interfaces. Data is transferred between the server and client as JSON formatted strings. The server is capable of managing a rich set of services such as many kind of map services to provide map content for the clients and customizable analysis services to perform geospatial analysis tasks.

In addition, the core server can be extended with different kind of extensions. The most important extensions are WebAdaptor reverse proxy, GeoEvent Extension for real time data and Portal for ArcGIS for user friendly interface for server. All extensions typically provide their own REST based interfaces.

Server products are containing rich administration interfaces where configurations can be done remotely. Admin interface contains graphical user interface running on the web browser and REST based interface that can used programmatically. Especially REST

interface is powerful option since it provides tools to perform installations and configurations remotely using scripts.
Admin and configuration REST interfaces are one of the key elements of the server software when discussing automated deployments. The REST interface provides standardized access to data published to the server and gives standard way to read, modify and create spatial content published to server. The REST interface offers several functions to manipulate and investigate spatial data such as geometry operations (join, union, cut, length, area), address locators (geocoding), network analysis (road network) etc. In addition, the REST interface offers an admin interface that can be used for several administrative operations without the need to open a graphical user interface. The administrative REST interface can be used programmatically that offers effective way to create automatic tasks and deployment scripts.

The server application stack fits well to the automated deployment process. Typically, each server deployment contains installation and configuration of multiple server products and publishing of set of services. The products provide remotely used interfaces for administrative operations that enable to run script based deployment and maintenance processes.

Geodatabase products

Typically, a GIS platform contains data store that is often relational SQL database. Esri technology can take direct connections to the various commonly known database management systems such as Oracle, MS SQL Server, IBM DB2, PostgreSQL etc. ArcGIS geodatabase is in generally a layer over the relational database that provides required functions and formats to store and manage spatial data content. The geodatabase has the native ArcGIS specific data structure and it is the primary data format used for editing and data management.

From the deployment point of view there are three separate aspects to take into account: set up geodatabase, initialize data model and load data content. Esri provides an installation package to set up geodatabase. The package is wizard based and it will install the necessary functions, procedures and other libraries to enable spatial functionality. ArcGIS Python Arcpy sitepackage provides tools to connect to the geodatabase schema

and content. That enables script based management of geodatabase. Sitepackage contains functions such as create data objects (tables, relationship classes) and modify data objects and database schema (add attributes).

4 Current State Analysis

This chapter covers the current state analysis of the software deployments in the case company. The current state analysis was based on interviews with the researcher's colleagues and the researcher's own experience as an employee in the case company. The researcher has worked five years in the case company at the time when this thesis was written.

4.1 Deployment Process as Part of Delivery Process

The deployment process is one part of the whole software delivery process in the case company. Typically, a complete delivery process contains sub processes such as sales, development, deployment, testing and support process. The complete process carries out the software solution from customer requirements to final release that is ready for production use. The deployment process is part of the delivery process. The role of the deployment process is to carry the implemented software solution to customer use including environment setups, service setups and necessary data loadings to the data stores.

The objective of this thesis is not to model or develop new business processes as described in Chapter 1. Processes presented in this chapter are modelled by describing how daily work is currently managed in the case company. Processes are described based on discussions and interviews with colleagues.

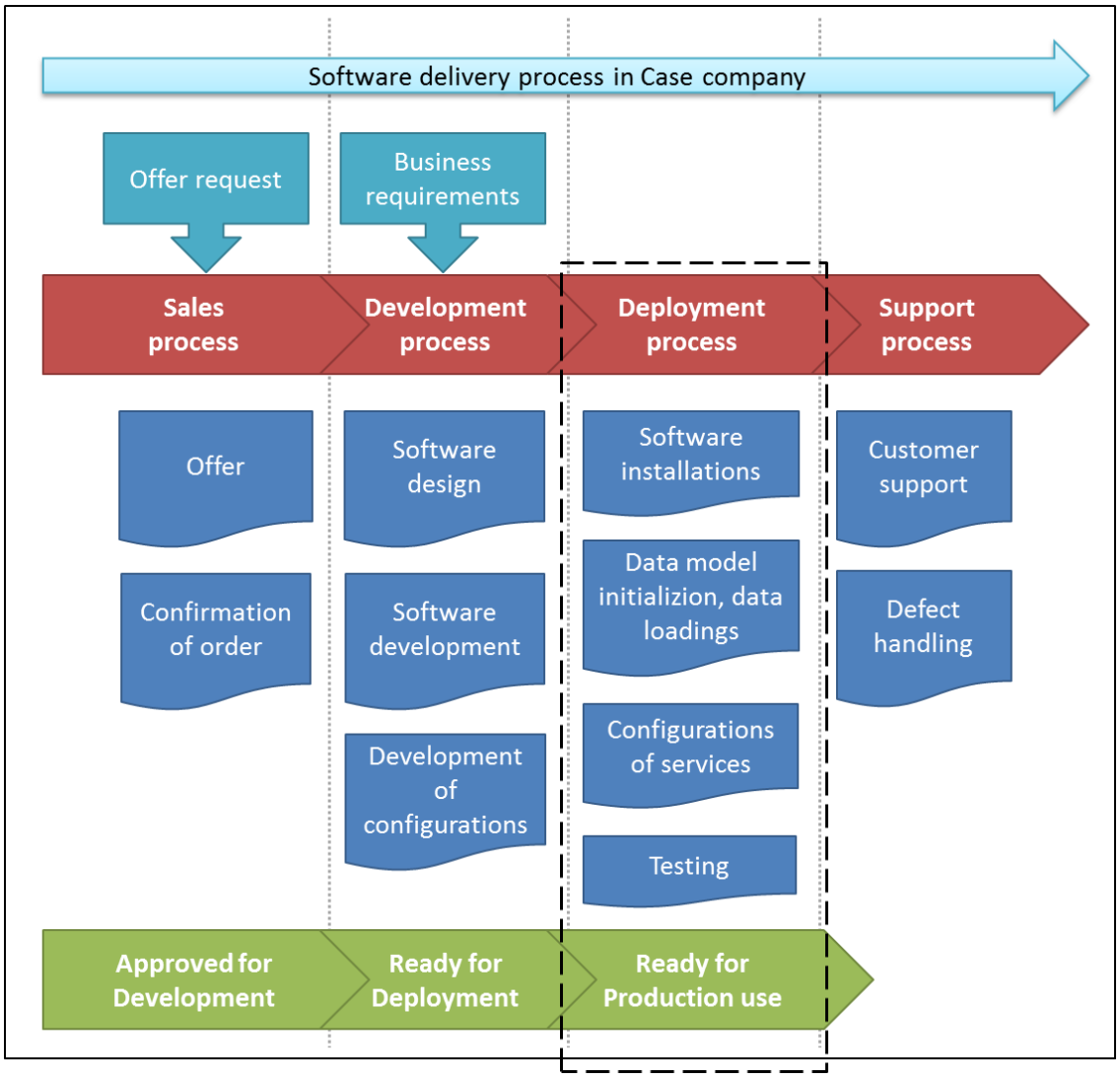

Figure 4. Software delivery process in case company.

Figure 4 describes a typical software delivery process. It includes common sub pars of the delivery process and its deployment part. The main steps of the process are represented in red. The outcome of each main step is represented in green. The subtasks of the main steps are represented in blue. The deployment process (scope of this thesis) is bordered by the dotted line.

4.1.1 Deployment Process in Details

The primary role of the deployment process is to carry out the implemented software solution from development environment to production environment. Deployment process in a technical context contains typically three separate phases. These phases are included in a usual deployment project in the case company. The process is divided to the

phases to have a better understanding of the details, challenges and technical specialities. The deployment phases are: 1) installation of applications, 2) data in different roles and 3) configuration and publishing services. Naturally there is variation between the projects and for example the installation phase is not included or is only minor when deploying a solution to the cloud environment.

Figure 5 presents the deployment process phases with the most common subparts in details. The parts are placed on the timeline to give a sense how they are linked together.

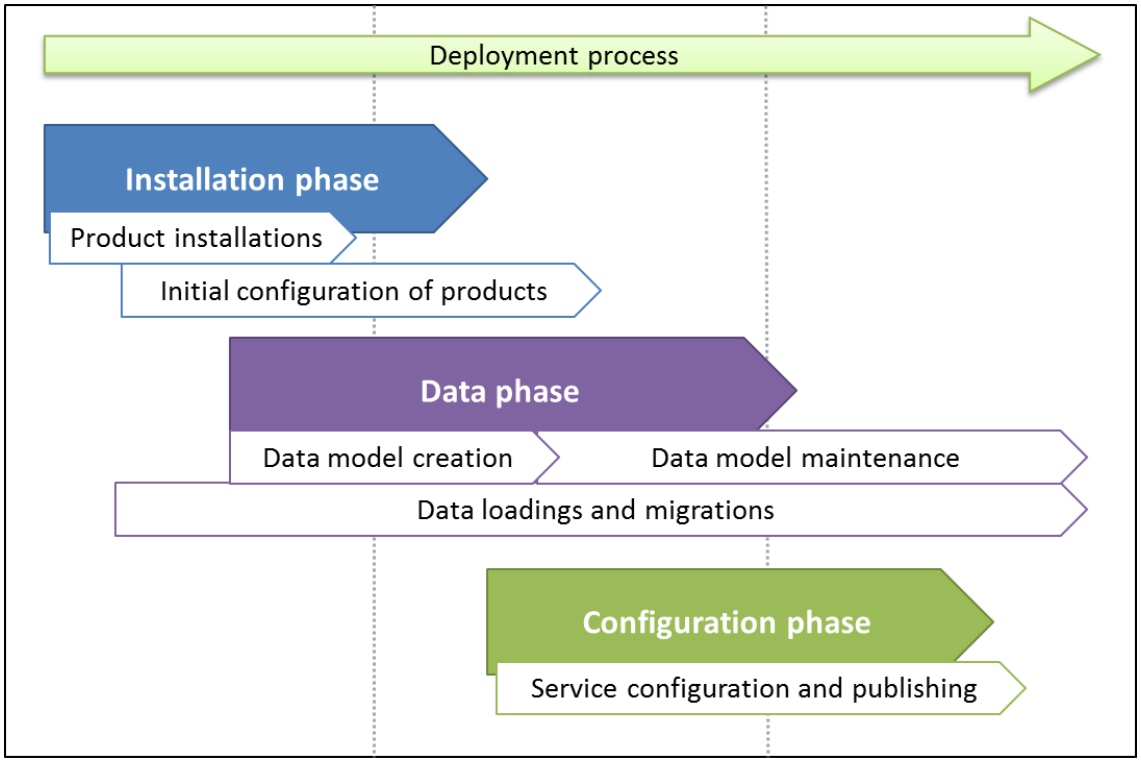

Figure 5. Deployment process in details.

The deployment process starts from the software installations and continues to the data model creation, data loadings, service publishing and service configurations. The installation phase is usually the first step where the deployments are started. Software installation is required before the other phases can be started so it has to be done first. Typically, rest of the phases and the subparts are run more or less simultaneously and iteratively. This may happen especially when dealing with the long projects where sub releases are published multiple times to the customer environment before the final release.

Installation phase

The installation phase is the first part of the deployment process. The role of this phase is to firstly install a certain set of the software applications to the desired environment and secondly configure the applications to work together. Usually when talking about the Esri enterprise software, it is required to install multiple applications to the same environment at once.

The first initial configuration step of the installations is including initial setups to get the installed applications to talk together. That is required to have a full working setup. For example, a typical deployment case is to setup ArcGIS Server, Portal for ArcGIS and WebAdaptor to the one environment. After the installation is finished, it is required to join applications together by setting up a certain set of the configuration settings. That is the second step of the process. The configuration includes security settings, environment based settings such as IP addresses, computer names and other necessary setups. Applications are visible to each of the other after the configuration is ready. In addition, when the requirement is to use relational database as a geodatabase, database must be "geo enabled" by using the ArcGIS tools. This setup will deploy the necessary functions and data types to get the database work with the spatial content. The applications contain wizard style of the user interface for the installation and the configuration that is run manually. Also the "silent" installation method is possible for the script driven installations.

Data phase

The data phase is the second phase of the deployment process. Data is divided to the two general sub steps: 1) creation and maintenance of the data model and 2) loading data to the data stores.

The data structure for the solution is produced during the first sub step. The step contains required setups to have the data model created, maintained and updated. In this context the data model means database structure including tables and attributes. Technically the database can be a relational database or some other file based data store solution.

Typically, the design and initial implementation of the data model schema is done during the software development phase. In the deployment phase the data model is deployed

to a customer environment. That type of projects where the requirement is to manage multiple releases, the data model has also to be deployed multiple times. Usually the deployment between the separate releases is more about updating earlier deployed data model, not creating it from the scratch every time.

The second sub part is divided in two scenarios: data loading for test purposes and data loading of the customer specific content (data migration).

Typically, the test data is needed almost in every deployment case to ensure that the system is working properly. The test data usually contains a predefined set of the data objects to have the installed setup tested. In the current deployment model test data is typically collected as a manual process or by using some other non-automatic data source.

In many projects customer data is loaded to the environment as a data migration task. The data migration task can be included in the deployment project or performed as a separate migration project depending on the project requirements. The challenge related to the data migration is that the customer data is typically provided in a such format that is not possible to directly load into the GIS system. In a data migration process data content is processed to such a format that it is possible to load into the system. The migration process is challenging to automate since the data content can vary a lot between the different scenarios and the deployment setups. For that reason, the migration project can be separated to a different project to not disturb the deployment project itself.

Configuration phase

The configuration phase is the third phase of the deployment process. Configurations are done after the installations are successfully finished. The role of the configuration phase is to setup project specific services and settings. The aim is to have a full working application setup with the required interfaces and look and feel.

The configuration part is typically the most time consuming part of the installation process. It contains a lot of small steps to manage and it is sensitive for the human errors when run as a manual process. The types of configuration tasks vary depending on the deployment setup and the installed products. The aim of the configurations phase is to

deploy the custom configuration setup to the final environment including services and the other required parts. The configuration setup is initially built up as a result of the development process. Publishing different kind of the services is the most important part of the configuration setup. The services provided by the setup are for instance mapping, geoprocessing, analysis and other services. In addition, the configuration contains steps related to the published services such as creation of the user accounts, setting up security rights, setup authentication settings for services etc.

4.2 Typical Deployment Scenarios

The case company is an Esri software retailer and its main business is software licence selling. Therefore, the typical projects contain more or less software deployments, installations and configurations. From the business process perspective, the deployment step is one subpart of the whole software delivery process. It contains phases such as sales, development, deployment and support process.

Based on the current state analysis, made by the researcher, there have been identified six typical deployment scenarios that are divided into two parts: role definitions and complexity definitions. The scenarios described later are based on the actual projects in which the researcher has been involved. In addition, the researcher has interviewed his colleagues related to the software deployment processes.

4.2.1 Deployment Scenario Definitions

The following deployment case definitions describe different deployment scenarios. In the scenarios described below the roles of the supplier and customer are different in each case. These cases are common in the actual software deployment processes.

Deployment scenario (a) "Partial supplier"*: The supplier has the primary role of the deployment.*

In this scenario the deployment is done to the customer's internal environment by the supplier product experts. This is probably the most common deployment case. In this scenario product experts install the software components to the customer environment and if needed, the customer's ICT provides technical advice. Advice is typically related

to the components that are maintained by the ICT, such as firewall settings, user accounts and IT infrastructure.

In this scenario the challenge related to the automated deployments is that typically the deployment environment contains parts that are fully managed by the customer ICT. In that case the deployment process can be automated only partially.

This scenario is well suitable for the automated deployments. The supplier has the main responsibility of the deployments and access to the deployment environment. That allows the use of the automated deployment process by the supplier as long as the required IT infra related configurations are done correctly.

***Deployment scenario (b) "Customer ICT"**: Customer's ICT provider has the primary role of the deployment.*

In this scenario the deployment is fully performed by the ICT provider of the customer. This is a common scenario especially in large organisations where the ICT services are outsourced to an external provider. It is typical for this kind of the scenario that the ICT does not allow access to the deployment environment for the external providers. Detailed installation instructions are required from the supplier so that the ICT is able to run installations independently.

The challenge is that the environment contains parts that are not reached by the supplier. The manual deployment process requires detailed installation instructions that should be provided to the customer ICT so that the ITC provider is able to run deployment without the supplier advice. It is required that the deployment setup is well packaged and documented.

***Deployment scenario (c) "Full supplier"**: The deployment is done fully by the supplier.*

In this case installations and configurations are done fully by the supplier without the customer's ICT. This scenario is common in the following two cases: if the deployment is done to the cloud environment or if the customer has a fully dedicated machine for the

GIS solutions. In these two cases a full access and maintenance rights to the deployment environment are required for the supplier.

This scenario is well suitable for the automated deployments. The supplier has full control of the deployment environment that lower the barrier of use of the script based deployment process.

4.2.2 Deployment Complexity Definitions

Complexity definitions represent different kind of deployment scenarios. The following three scenarios are examples of those reflecting the actual deployment projects. The scenarios are used to describe challenges of the deployment cases. In the actual projects scenarios are naturally mixed to each of other. In the following definitions, the "Server" object represents a physical or virtual machine that contains a set of the Esri products such as ArcGIS Server, WebAdaptor and configurations of them. This kind of a case is the most typical one with the Esri server side deployments when the aim is to install server software with the utility programs, data store software, server extensions and so on.

Complexity definition (a) Simple: *one deployment during the project lifecycle to single machine environment.*

In this case the deployment is done once during the project lifecycle to the single non-clustered environment. In other words, the solution is deployed once to the deployment environment without steps. This is a typical case in small projects where there is no requirements to provide sub releases or use separate environments.

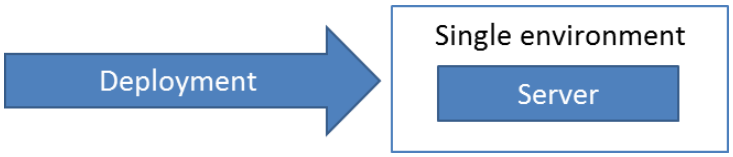

Figure 6. Complexity definition (a) Simple: one deployment during the project lifecycle to the single machine environment.

Complexity definition (b) Medium: *one deployment during the project lifecycle to the multiple environments and/or clustered environment.*

In this case deployment is done once to the multiple environments and/or to the clustered environment. The requirement is to deploy the solution separately to the several environments such as the test environment, the user acceptance testing environment (UAT) and the production environment. In the clustered cases, the deployment environment contains multiple parallel machines where the solution is deployed. The aim of the clustering is typically to get the request load divided to multiple machines to get more performance and minimize down time of the services. This kind of a deployment model requires that the same identical installation and configuration process have to be run separately multiple times. As a result, all the machines must contain an identical setup of the applications and the published services.

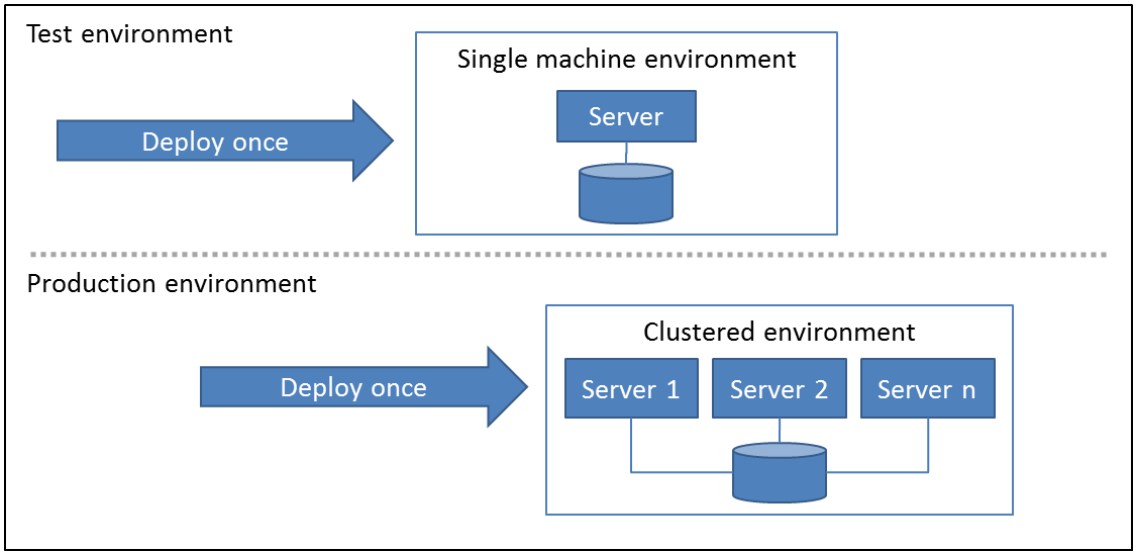

Figure 7. Complexity definition (b) Medium: one deployment to the multiple environments and/or clustered environment.

Complexity definition (c) Complex: *multiple deployments during the project lifecycle to the multiple environments and/or clustered environment.*

In this case the deployment is done multiple times during the project lifecycle to the multiple environments and/or the clustered environment. Compared to the complexity definition (b), deployment is done multiple times. This kind of a deployment case can occur for example in large projects where it is needed to provide sub releases to the complex

customer environment and in Scrum projects where is required to deploy solution to the customer environment after every sprint.

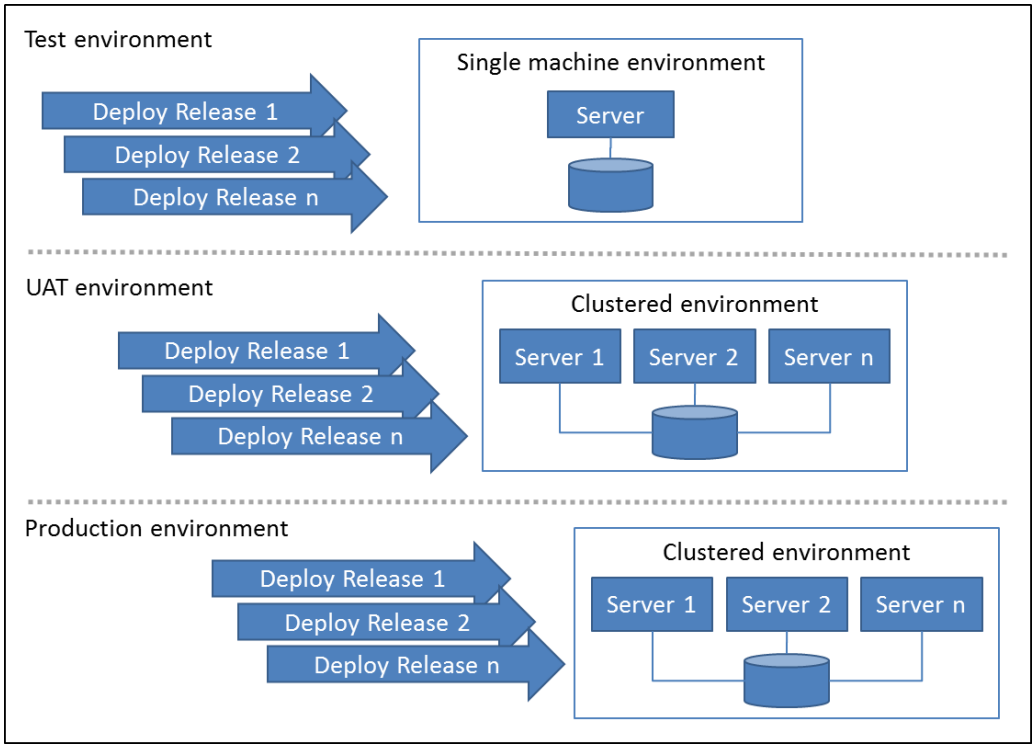

Figure 8. Complexity definition (c) Complex: multiple deployment releases to the multiple environments.

From the technical point of view the same identical installation process is done to all the machines in all the environments. In this case it is important to take care that the environments are fully identical from the installation and the configuration point of view. It is also possible that there will be needs to install batches, version updates or some other configuration changes to all the machines included in deployment. Maintaining this kind of an environment could be demanding since it could include a lot of separate machines.

5 Analysis of Technical Alternatives of Deployment Framework

This chapter outlines requirements and presents alternative technical options for the deployment framework. Requirements are based on the deployment process and deployment scenarios presented in Chapter 4. This chapter is divided into three separate parts in order to cover the whole deployment process from beginning to end. These parts face to the phases of the deployment process. This separation is needed in order to identify individual requirements of each part of the deployment process.

5.1 Requirements for Deployment Framework

The deployment framework has to fill three separate main requirements based on the current state analysis. These main requirements are divided into a set of sub requirements. The main requirements are: 1) installations of the software products, 2) data model management and data loadings, 3) service configurations.

5.1.1 Installation Phase of Deployments

The installation phase contains following two requirements: 1) install desired software products to the desired environment and 2) setup initial configurations for the installed software. Typically, the requirement is to install multiple applications to the same environment and get applications integrated together. In addition, the deployment environment can contain different kind of the setups introduced in Chapter 4.

5.1.2 Data Phase of Deployments

The data phase contains the following two requirements: 1) create and maintenance the data model and 2) load content to the data model. The data model is typically stored to the relation database or to the file based data store. Data store solutions are introduced in Chapter 3.

The first requirement, the data model creation and maintenance, contains steps to initialize and update the data model schema. Typically, that includes tasks to create the database tables, views, relations and the other necessary objects. The data model can

be created from the scratch when the environment is initially setup or existing data model can be updated in the maintenance phase of project.

The second requirement, data loading, contains steps to load existing data to the data store and in some cases load test data for the testing purposes. Migration of the data is not in the scope since it is too large a topic to include to this thesis. Commonly data migration means conversion of the data content from the unknown formats to the GIS format.

5.1.3 Configurations Phase of Deployments

The configuration phase contains requirement to publish GIS services to the deployed setup. This contains different type of services that are supported by the ArcGIS Server products. The most typical services are the basic map services that provide geospatial information through the REST interfaces and the analysis services that are used for the geospatial analysis operations. The services contain a wide set of the settings and data sources configurations and they should be handled by the deployment framework.

5.2 Alternative Technical Solutions

This chapter describes available technical options for the deployment process automation.

In general, there are two commonly known technical options for the deployment process automation available: Python scripting and Chef Cookbooks. These two options are so called *best practices* which are officially documented and supported by the Esri. In addition to the above, cloud environments provide many benefits for the deployment automation.

Python scripting and Chef Cookbooks differ significantly from each other in technically. Python is a traditional scripting language that can be used in a many kind of software projects including deployments. Chef is a third part installation platform that is extended to work with the Esri software by cookbooks. Both are capable to run basic software installations but Python provides much wider functionality for the interface and data ma-

nipulation. On the other hand, Chef offers an easy and straightforward platform to produce simple deployment processes and Python provides functionality to connect to the REST interfaces, the data stores and the other Esri specific services.

5.2.1 Cloud Environments

In modern software projects it is a very common scenario that the development and the deployments are done by using cloud environments. A cloud environment can be a private cloud where virtual machines are maintained or some other cloud system that contains all the functionality out of the box. One example of an out of the box cloud service is the web based Esri ArcGIS Online that contains a rich set of functions to the maintenance of the geospatial content. Cloud environment provides powerful tools to optimize the solution development and the deployment workflows.

A cloud environment sets up special requirements and challenges compared to traditional physical machine environments. On the other hand, a cloud provides powerful new options to get software project workflows more streamlined. A cloud is flexible in many ways. It provides a flexible way to adapt to the situations where the load of the system varies in a big scale. Hardware resources can be scaled based on the load that allows to optimize expenses of the system. This gives flexibility to expand the system with the new services and it provides services for the new user groups.

A cloud provides tools to streamline processes from the deployment point of view. Cloud machines can be copied, duplicated and moved from one place to another with minor configuration. This provides options to organize projects in completely new ways. The development of the solution could be done in the cloud environment and then just copied to the production side without need for the heavy deployment process. This kind of workflow can be suitable especially for a well productized solution package that contains a strictly limited set of applications.

Typically, common cloud environments such as Amazon and Azure contain out of box application templates. New machines can be deployed easily from these templates. The templates include an initially configured set of the applications. This provides a good start point for the development and the deployment projects. Both Amazon and Azure contain Esri ArcGIS templates.

5.2.2 Python Scripting

Python is a traditional scripting language that is widely used and well integrated with the Esri platform. Python is a fundamental tool for the GIS professionals to extend the functionality of the ArcGIS and to help automate workflows. Python is used in different roles in the Esri desktop applications and the server solutions. The role of the Python scripting varies from the desktop software geoprocessing tasks to the server side scripting. Python is a powerful tool to streamline the daily workflows without a need of the advance level software development skills.

The most common application where Python is used is the ArcGIS Desktop. Python contains interfaces to read, execute and extend the ArcGIS Desktop functionality. Python is used in the ArcGIS Desktop as a base for the geoprocessing models. The geoprocessing models are automated tasks where multiple small subtasks are chained together to have a desired output. Python can be also used to manipulate data that is loaded to the ArcGIS Desktop application. This is a powerful option when there is a need to execute identical procedures for a large group of the data objects.

Another strength of Python in the Esri environment is its ability to connect with the ArcGIS Server and the ArcGIS Online REST interfaces (Representational State Transfer). The REST interfaces are one of the core elements of the Esri Server platform. The REST provides functions such as server configuration, data publishing and data manipulating. When creating deployment scripts the REST interfaces are used as a key element to have a configuration initialized and data published.

Python is extended by the ArcPy site package to get it integrated with the Esri platform. The ArcPy site package is like a library of functions that adds functionality to the core Python. The ArcPy package acts as the central role and contains all the necessary functions to interact with the Esri programs and the application interfaces. [17]

5.2.3 Chef Deployment Framework

Chef is an automation framework and a configuration management tool that is made to help the server and the application deployments. Chef can be used in different kind of

environments such as physical, virtual and cloud environments. Chef uses system configuration “recipes” that describe how the deployed applications and the utilities are managed. [19] [20]

Chef uses so called cookbooks that are fundamental unit of the configuration and policy distribution. The Cookbook contains everything that is required to support different kind of deployment scenarios such as: recipes that specify the resources to use, attribute values, file distributions, templates and extensions. Ruby is the reference programming language for create cookbooks and recipes. [19]. The main elements of Chef are described in Table 1.

Recipe	A recipe is a base element of the Chef. Typically, one individual recipe handles installation process for one individual application. A recipe is for example, “Install ArcGIS Server”, “Install Portal”, “Federate Server to Portal”. Recipes are built by the Ruby programming language and can be modified if needed. In a general scenario there is no reason to modify recipes.
Cookbook	A cookbook is basically a set of recipes for different purposes. Cookbooks are for example “Arcgis”, “Java”, “Tomcat”, “Windows” etc. The “Arcgis” Cookbook includes recipes for the individual ArcGIS application installations. Cookbook is a simple folder that contains the recipe files in the Chef file hierarchy.
Installation scripts JSON	Installations JSON scripts are the main elements of the Chef deployment process. Scripts contain a list of applications to install (recipes) and application specific installation parameters. For example, the “Arcgis” setup script contains initial user accounts and installation paths that have to be setup before installation can be started. Scripts are built by JSON language and they can be easily modified by using standard text editor.

Table 1. Main elements of Chef.

Esri officially provides the ArcGIS cookbook for the Chef to support deployments for the ArcGIS software. The Cookbook contains several recipes for the different ArcGIS application deployment scenarios. The recipes are focused on the enterprise software deployments but the desktop applications are not supported. One example of the ArcGIS

recipe is so called “Webgis” environment setup. The “Webgis” recipe contains deployment scripts for the applications such as ArcGIS Server, Portal, WebAdaptor and DataStore. The ArcGIS Cookbooks are available beginning from the ArcGIS software version 10.3.1.

5.3 Service Publishing Workflows

This chapter presents the service publishing workflows related to the ArcGIS Server software. The workflow includes a set of steps to have certain services published to the server. The ArcGIS Server services are for example different type of the map services and the analysis services. All services typically contain an REST interface for the client requests.

In general, the ArcGIS system provides several service publish workflows such as manual and script driven. The most common workflow in the case company is manual workflow. In manual workflow the service source document is first created with the ArcGIS Desktop application and then published to the server. The ArcGIS Desktop application contains all the necessary tools for the service publishing. In many cases this is a well working process especially when the service is published only once without the need for the repeatable process. A script driven process is more efficient if the requirement is to publish service multiple times.

Two service publish workflows are described below. Both of the workflows contain user triggered parts and script driven parts. In addition, a fully manual process is supported but that is left outside of these workflows.

Service publishing workflow (a): “User-triggered”

In a user-triggered workflow option, the first service publish steps are done manually by using the ArcGIS Desktop application. The final step is executed by the script. The final step contains the actual publishing process installed to the server. This workflow option is suitable when the source MXD document changes rapidly. In this case it is required to ensure that the modified source document is still valid for the server publishing. It is typically reasonable to manage the validation as manual work in the ArcGIS Desktop application. The outcome of the manual steps is a so called Service definition file that is

published to the server by using the script. Figure 9 shows the ‘User-Triggered’ service publishing workflow.

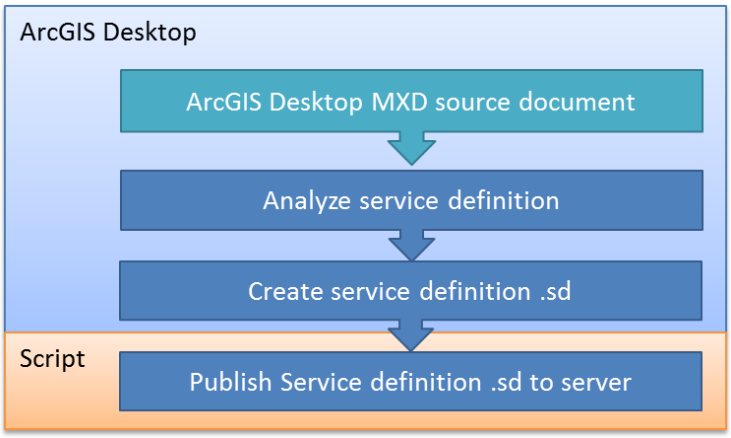

Figure 9. Service publishing workflow (a): “User-triggered”.

Service publishing workflow (b): “Script-triggered”

In a script triggered workflow option the source MXD document is managed manually in the ArcGIS Desktop software. All the other steps are handled automatically by the script. Figure 10 shows the ‘Script-Triggered’ service publishing workflow.

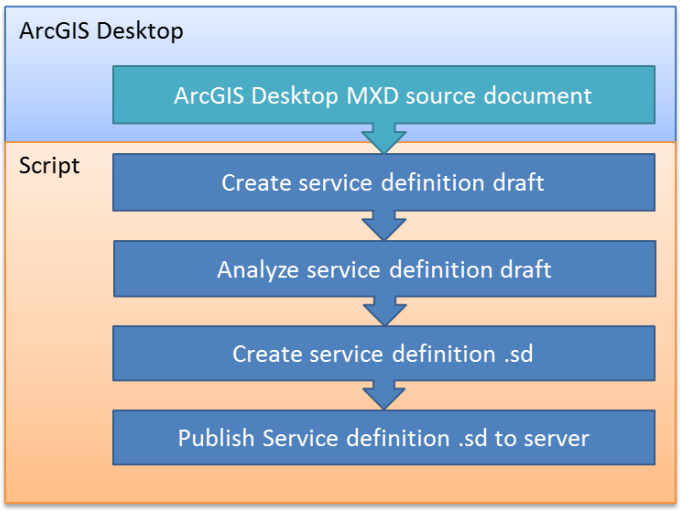

Figure 10. Service publishing workflow (b): “Script-triggered”.

This workflow is suitable when the source MXD document is not changing and the requirement is to publish the service multiple times to different environments. In addition,

the script can be extended to read multiple MXD documents at once that allows to use one process to publish multiple services.

5.4 Proposed Deployment Process Options

As a conclusion, Python and Chef are both suitable technological options for the automated deployments. Both contain required functionality to install and configure the Esri products. Figure11 illustrates the proposed technical options for the deployment process.

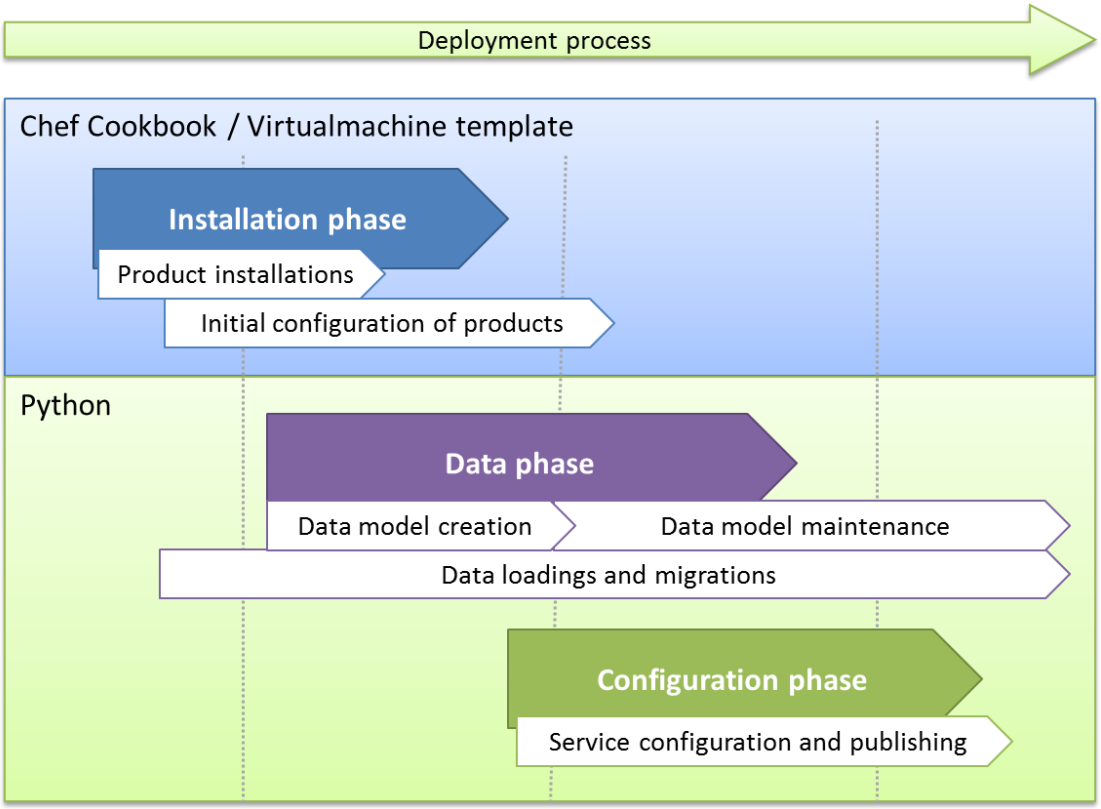

Figure 11. Proposed technical options for the deployment process.

Chef is the most suitable option for the installation phase of the deployment process. Esri provides a ready Cookbook configuration for the typical deployment scenarios. Chef is capable of installing applications and handle the required initial configuration. The environment is ready to run after the installation script has finished work.

Python is more suitable for the data phase and the configurations phase of the deployment process. In fact, Chef is not able to make required configurations or data model tasks. Python contains functionality to connect the server resources from the script level.

That allows to use Python for configuring the server and publish services. Python also contains functions to manage geodatabase that enables possibility to create database management scripts.

6 Development of Deployment Package POC

This chapter introduces the practical part of this thesis. The practical part contains the implementation of the deployment package by using the selected technologies. The deployment package includes a selected set of the applications and their configurations.

The deployment package is created as a Proof-of-concept (POC). The goal of the POC is to examine the alternative technical options and show how the options are suitable for the case company business needs in a practice. The package is not meant to be production ready but it can be used as a base for the later development.

In general, the deployment package POC covers the key elements of the fully functioning ArcGIS enterprise setup. The deployment package contains software installations, creation of the data model and publishing map services to the server. The aim of the deployment package is to design and test the script based concept to manage the key parts of the deployment tasks. The deployment package covers Esri software products. Other elements such as the base operating system are not included in the package.

The deployment package contains three separate parts based of the deployment process that is presented on the current state analysis. The parts are: 1) software installations, 2) data model management and 3) service configurations. Technically these tree parts are implemented as a separate script. First part is implemented by using Chef, second and third part is implemented by using the Python scripts.

6.1 Deployment Project Setup and Tools

Esri software products for the deployment package were selected based on the current state analysis. Selected Esri enterprise products are used in the actual projects. The products are well suitable for the automated deployments. The deployment package itself does not contain the installation of the base operating system.

6.1.1 Base Operating System selection

The base operating system for the deployment package was selected from the two main options: Windows and Linux. Both options are supported by the Esri and both options

provide similar functionality from the Esri software point of view. In addition, Chef and Python are both supported by Windows and Linux. As a conclusion there are no technical aspects to choose the one over the other.

The Windows operating system was selected over Linux based on the following aspects: 1) most of the actual projects on the case company are based to Windows and 2) researcher has more experienced with Windows based environments. The Windows 7 was selected as an environment over the Windows Server. The Windows 7 is a light weight solution and it provides almost similar deployment experience than the Windows Server. The whole deployment setup was implemented to the Oracle Virtual Box virtualizing environment. The virtualized environment was selected since it provides flexibility to run the deployment scripts multiple times from the scratch.

6.1.2 Architecture of Deployment Setup

The selected architecture approach is commonly used in the actual projects and the architecture contains a typical set of the enterprise GIS applications. The architecture contains key parts such as the server solution, the data stores and the reverse proxy. The end result of the deployment setup is fully functioning server solution that includes data stores and server interfaces services. Figure 12 presents the architecture of the deployment setup.

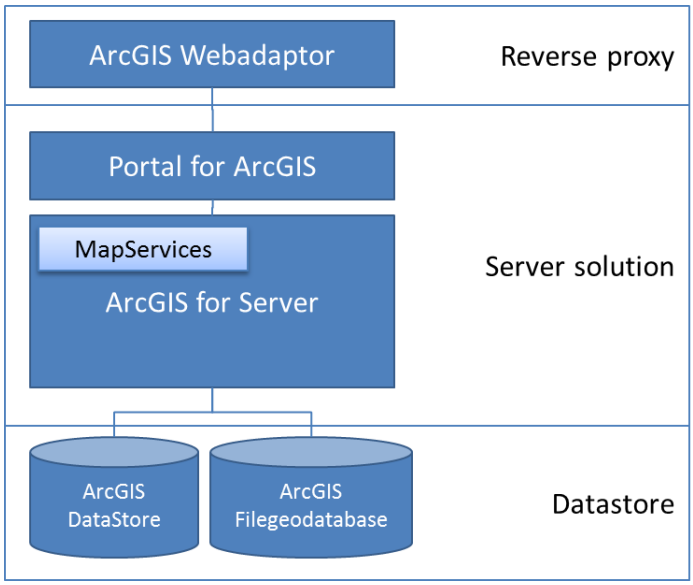

Figure 12. Architecture of the deployment setup.

The setup was built inside of the Oracle VirtualBox environment. All the required core components were installed at once. The deployment scripts were developed and executed inside of the environment. The end result was tested by using the mock data. Table 2 summaries the software components used in the deployment setup.

Oracle VirtualBox 5	Virtualization environment for the deployment setup
Windows 7	Base operating system
Chef client 12.4.1.1	Software installation framework
ArcGIS Cookbooks 1.1.3	Esri extension for Chef
Python 2.7	Python scripting with Esri ArcPy sitepackage
ArcGIS products	Presented more detail in the Table 3

Table 2. Software components in the deployment setup.

Chef and the Chef cookbook version was selected based of the Esri recommendations. Python was installed within the ArcGIS software and version was related to the ArcGIS version.

6.2 Installation Phase

The installation phase of the deployment setup contains script to handle the actual installation of the ArcGIS applications. Supporting applications such as the operating system are not in the scope. Table 3 presents the ArcGIS applications included to the installation setup.

ArcGIS for Server*	ArcGIS for Server is an Esri core server product. The role of the server is to publish geospatial content as a REST interface for the client applications.
Portal for ArcGIS*	Portal for ArcGIS is an extension product for ArcGIS for Server. The role of Portal is to provide user friendly interface to browser, manage and create geospatial information. Typically, Portal is deployed to an organization private network to provide services for-intranet use.

ArcGIS WebAdaptor*	The role of ArcGIS WebAdaptor is to work as a top layer for ArcGIS Server from a network perspective. It is so called reverse proxy that manages network traffic to a server environment. Typically, WebAdaptor is used to enable single-sign-on login experience for GIS services. On Windows environment WebAdaptor is deployed as an individual IIS site that is managed by the standard IIS admin console.
ArcGIS DataStore*	The role of ArcGIS DataStore is work as a light weight data base solution for ArcGIS Server environment. Typically, ArcGIS DataStore is taken to use when there is no requirement to use full functional relational database. DataStore is primary data base solution for Portal for ArcGIS.
ArcGIS Filegeodatabase*	ArcGIS Filegeodatabase (FGDB) is an ArcGIS specific file system based data base solution. Filegeodatabase is a collection of v GIS datasets held in a file system folder. Filegeodatabase is maintained primarily by ArcGIS Desktop applications and additionally with Python scripts. From ArcGIS functional perspective filegeodatabase is nearly equal with a real relational database solution. It means that same maintenance script logic is used to ArcGIS filegeodatabase and the relational database.

Table 3. ArcGIS applications included to the deployment setup.

** Esri product version 10.3.1.*

The ArcGIS installation setup was selected based to the actual projects. Selected set of the applications is commonly used in the modern GIS software environments. The applications are integrated to work together and purpose of the installation script is to perform the installations and execute the required configurations.

6.2.1 Chef as Installation Framework

Chef is used to perform the ArcGIS software installations. Chef contains ready to use cookbooks, recipes and scripts for the ArcGIS installations. That allows straight forward installation experience.

To get Chef installation process started, the following requirements have to be fulfilled:

1. Installation medias for ArcGIS software

The all required installation medias for the planned setup have to be stored to the certain folder where Chef can found them. Typically, Esri software is provided as an ISO-image file or a setup package. In addition, the installation medias should be extracted so that every application is store to the separate folder.

2. Esri authorization license files

Typically, every Esri product must be authorized before use. The authorization files have to be provided for the installation script when using Chef for the installation. This is required to allow the installation script to finalize the software configurations. The authorization files have to be stored to the known folder where Chef is able to read the files.

3. Domain name for Web site

A fully qualified domain name is required for an ArcGIS Portal deployment. This is mandatory especially when deployment is run to a production environment. A server simple name can be used as well for development purposes but in that case all features may not work.

4. SSL certificate issued to Domain

A trusted SSL certificate is required for deployment of ArcGIS for Portal. ArcGIS for Portal uses a secure connection for example when log in and for a server federation. If the trusted certificate is not defined, deployment tool will generate a self-signed certificate and deploy it. That allows ArcGIS for Portal to be initially tested and quickly verified that installation was successful.

6.2.2 Chef Deployment Process in Brief

The Chef deployment process is presented in Figure 13.

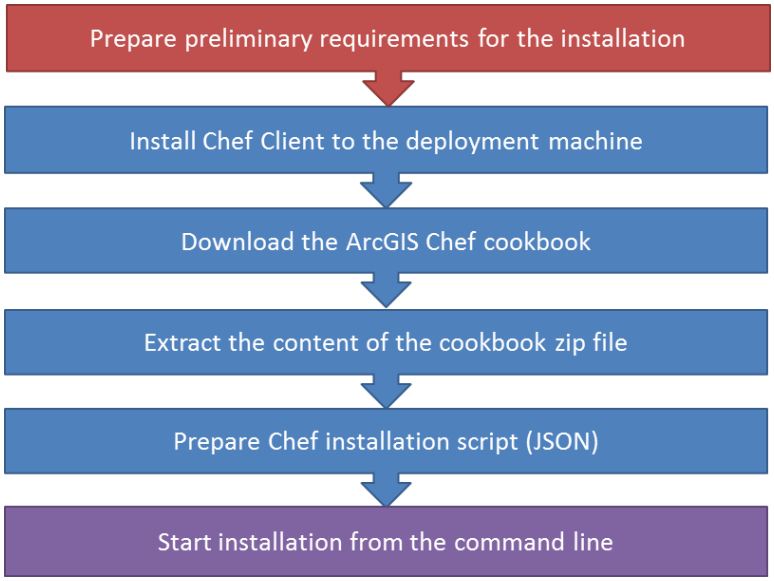

Figure 13. Chef install process steps.

The Chef deployment process starts by checking the preliminary requirements such as the installation medias and the required license. The process continues by installing the Chef Client to the target environment. The Chef Client is a component that is required to run the deployment process. After that, the “Arcgis” cookbook is downloaded and extracted. The cookbook contains the ArcGIS software specific installation scripts. The installation JSON script is choose from the available scripts and modified based to the requirements. The cookbook contains the several JSON script files that can be used for the different scenarios. Finally, the deployment process is started by the command line function "`chef-solo -j C:\chef\xxxx.json`”.

6.2.3 Cookbook Installation JSON Script

The Chef cookbook contains several installation script JSON files for the different scenarios. The each JSON script contains the configuration setup of the installed applications, key attributes such as paths, user account settings and the license file locations. By editing the JSON scripts the Chef deployment setup it is possible to fit certain purposes and scenarios. The JSON script is a plain text file and it can be edited by using an

ordinary text editor. The JSON scripts are located on the root folder of the Chef cookbook.

The cookbook JSON script “webgis-windows” was selected as a base for the deployment setup. The selected script contains a suitable set of ArcGIS enterprise applications such as ArcGIS for Server and Portal for ArcGIS. The script was slightly modified based for the requirements of the deployment setup. The most important variables to modify was user names and passwords, installation media paths and deployment paths. See Figure 14 for the Chef JSON script file.

```
{
 "arcgis":{
   "run_as_password":"arcgispw",
   "run_as_user":"arcgis"
 },
 "web_adaptor":{
   "setup":"C:\\ArcGIS10.3.1\\WebAdaptorIIS\\Setup.exe"
 },
 "data_store":{
   "setup":"C:\\ArcGIS10.3.1\\DataStore\\Setup.exe",
   "data_dir":"C:\\arcgisdatastore\\data"
 },
 "server":{
   "admin_username":"admin",
   "admin_password":"adminpw",
   "directories_root":"C:\\arcgisserver",
   "setup":"C:\\ArcGIS10.3.1\\Server\\Setup.exe",
   "authorization_file":"C:\\ArcGIS10.3.1\\Server.prvc"
 },
 "portal":{
   "admin_username":"admin",
   "admin_password":"adminpw",
   "security_question":"Your favorite ice cream flavor?",
   "security_question_answer":"vanilla",
   "content_dir":"C:\\arcgisportal\\content",
   "setup":"C:\\ArcGIS10.3.1\\Portal\\Setup.exe",
   "authorization_file":"C:\\ArcGIS10.3.1\\Portal.prvc"
 },
 "run_list":[
   "recipe[arcgis::system]",
   "recipe[arcgis::iis]",
   "recipe[arcgis::server]",
   "recipe[arcgis::server_wa]",
   "recipe[arcgis::datastore]",
   "recipe[arcgis::portal]",
   "recipe[arcgis::portal_wa]",
   "recipe[arcgis::federation]"
 ]
}
```

Figure 14. Chef Cookbook JSON script file.

Figure 14 presents the Chef JSON script file. Every application included to the setup has a separate block for the variables. The variables vary depending on the application requirements. The block named as “run_list” contains a definition of the applications to be deployed. The “run_list” refers to the cookbook recipes.

6.2.4 Summary

The key benefits of the script based installations are the ability to install and configure multiple applications at once by using the predefined installation script., The recommended solution for the software installations is the Chef cookbook based deployment model. This is based to the technical analysis and the POC. Chef is able to fulfil the requirements of installations and it is supported by Esri. The Chef installation JSON script is setup once against to the destination environment and used with the multiple machines with only minor changes. This reduces the risk of human errors and decreases time spent for the installation process. In addition, the script based installations are simplifying the installation process when installing an identical software set to the multiple environments.

The key disadvantages of the script based installations are the work effort to fulfil the Chef requirements before the installation process can be started. The requirements contain installation of the Chef Client software to the desired machines, collect the needed installation medias and set up the installation Chef JSON script. This requires noticeable work effort depending the complexity of scenario.

Technically the Chef cookbook based installation process was running smoothly. The most time consuming part was to fulfil the Chef requirements. After the requirements where fulfilled, the installation process itself was running relatively quick and reliable. The installation process does not ask for anything during the execution so the process can be left running without need for the controlling.

6.3 Data Phase

The data phase of the deployment setup covers the script functionality to maintain the database model. The role of the data phase is to implement a script that can be used to create and maintain a database schema in several environments. In addition, the script works as a document for the data model. The script contains functionality such as the

creation of database tables, attributes, relationship classes and the other necessary set-ups. In addition, the script contains test data load to the database.

Creating and maintaining the data model on an ArcGIS geodatabase varies from an ordinary SQL based database solutions. The ArcGIS Geodatabase holds a collection of special geographic entities for spatial content such as FeatureClasses and RelationshipClasses. ArcGIS specific entities are based to a standard SQL tables in a database level but they include special attribute types for a spatial data. In addition, an ArcGIS geodatabase contains additional system level information such as a versioning information.

The ArcGIS geodatabase is maintained by an ArcGIS software or through a native database management system. In general, Esri's best practise is that all actions related to a database schema must be managed through the ArcGIS software. This includes operations such as add and remove tables or modifying table attributes. The actions related to a data content itself can be managed by using a native database management system and SQL scripts. This is an accepted approach if the geodatabase versioning is not enabled. In addition, the geodatabase contains an ArcGIS specific SQL level spatial functions that can be used as well.

6.3.1 Python as Scripting Technology

The Python scripting was selected as a base technology for the data phase. The Python scripting is an Esri best practice to manage geodatabases. An ArcGIS sitepackage "ArcPy" is included to the Python that enables access to an ArcGIS specific functions such as a geodatabase content and schema. An ArcGIS Filegeodatabase was selected for a database technology. It provides nearly same functionality compared to a relational database but it is much lightweight. For that reason, it is better suitable for the deployment setup.

The final data maintenance script can be used with the ArcGIS Filegeodatabase and with the relational database solutions. Database tools provided by the ArcPy sitepackage are capable to manage different kind of database systems without need to refactor the Python script itself. That is the one of the biggest advantages of the Python technology. It is possible to write a single script that can be used it a several environments even they contain different database engines. For example, a single Python script can be used

against an ArcGIS Filegeodatabase, Oracle and Postgresql. This provides a great flexibility and reusability for the scripts.

6.3.2 Designed Data Model

The data model designed for the deployment setup is a relatively simple. The data model contains database key elements that are required to ensure the script functionality. The data model contains entities such as tables, relations between of tables and attribute domain lists. Context of the example data model was selected based on the actual projects. The Data model represents a simple scenario where aim is to store, manage and present electrical network components. The sample network contains data objects such as towers, lines and cables.

The data model is designed so that it can be used as a base for the ArcGIS Server map services. The map services publish in a practice is described in the configurations phase of the deployment setup. Figure 15 presents the elements of the data model.

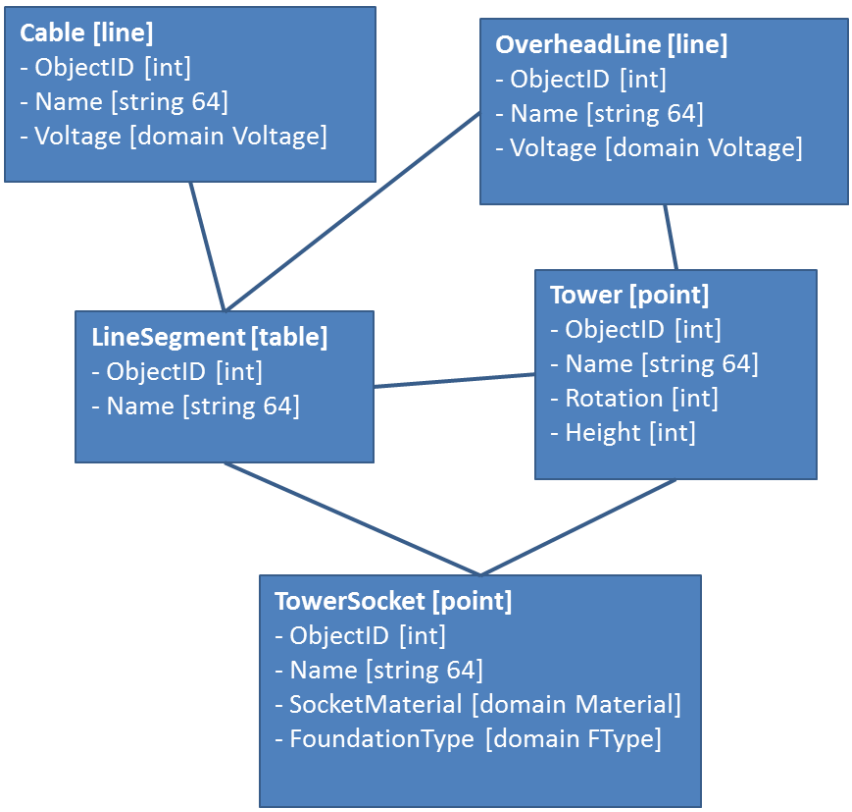

Figure 15. Elements of the data model.

As can be seen in Figure 15, the data model contains five tables with a few attribute fields and three domain lists.

6.3.3 Python Maintenance Script

The data model maintenance script is used to initially build the data model to the destination database. The script creates the data model including all the necessary database objects. Aim of the script is to create a full data model without need of the manual work.

An ArcGIS “ArcPy” Python site package was used as a key library in the script. The ArcPy site package contains the full set of functions to access to the ArcGIS geodatabase. The functions are relatively straightforward to use and there is a good documentation available. For example, a table creation requires one line of a program code. In the script there was used ArcPy functions such as AddField_management (to create the new attribute fields), CreateRelationshipClass_management (to create the relations between the tables) and CreateFeatureclass_management (to create the feature classes). Figure 16 shows an example piece of the data model deployment script.

```
# Create domain lists
createDomain("FTYPE", "TEXT", { "1":"Sand", "2":"Clay", "3":"Rock", "4":"Soil" } )
createDomain("MATERIAL", "TEXT", { "1":"Concrete", "2":"Rock" } )
createDomain("VOLTAGE", "TEXT", { "1":"110", "2":"220", "3":"400" } )

# Create tables and add attribute fields
createTable(sys.argv[1], "LINESEGMENT")
addField("LINESEGMENT","NAME", "TEXT", "", "", "64", "", "NULLABLE",
"NON_REQUIRED", "", "")

createFeatureclass(sys.argv[1], "TOWERSOCKET", "POINT")
addField("TOWERSOCKET","NAME", "TEXT", "", "", "64", "", "NULLABLE",
"NON_REQUIRED", "", "")
addField("TOWERSOCKET","SOCKETMATERIAL", "TEXT", "", "", "64", "", "NULLABLE",
"NON_REQUIRED", "MATERIAL", "")
addField("TOWERSOCKET","FOUNDATIONTYPE", "TEXT", "", "", "64", "", "NULLABLE",
"NON_REQUIRED", "FTYPE", "")

createFeatureclass(sys.argv[1], "TOWER", "POINT")
addField("TOWER","NAME", "TEXT", "", "", "64", "", "NULLABLE", "NON_REQUIRED", ""
, "")
addField("TOWER","ROTATION", "SHORT", "5", "", "", "", "NULLABLE", "NON_REQUIRED"
, "", "")
addField("TOWER","HEIGHT", "SHORT", "5", "", "", "", "NULLABLE", "NON_REQUIRED",
"", "")

createFeatureclass(sys.argv[1], "OVERHEADLINE", "POLYLINE")
addField("OVERHEADLINE", "NAME", "TEXT", "", "", "64", "", "NULLABLE",
"NON_REQUIRED", "", "")
addField("OVERHEADLINE", "VOLTAGE", "TEXT", "", "", "64", "", "NULLABLE",
"NON_REQUIRED", "VOLTAGE", "")

createFeatureclass(sys.argv[1], "CABLE", "POLYLINE")
addField("CABLE", "NAME", "TEXT", "", "", "64", "", "NULLABLE", "NON_REQUIRED",
"", "")
addField("CABLE", "VOLTAGE", "TEXT", "", "", "64", "", "NULLABLE", "NON_REQUIRED"
, "VOLTAGE", "")

# Create relations between tables
createRelationshipClass("TOWERSOCKET", "TOWER", "TOWERSOCKET_TOWER", "SIMPLE",
"TOWERSOCKET", "TOWER", "NONE", "ONE_TO_MANY", "NONE", "OBJECTID", "OBJECTID")
```

Figure 16. Example piece of the data model deployment script.

One of the key advantages of the script based data model handling is that the script itself works as a document for the data model. Code example in Figure 16 demonstrates the creation of the database objects by the script. This example is taken from the deployment setup script. By reading the script, it is possible to see structure of the database schema including tables, attributes, relations and domain lists.

6.3.4 Summary

The key benefits of the script based data store management are the possibility to use the same structured script for several environments and to have the database schema documented by the script. The script automation solves especially the following challenges: the requirement is to deploy multiple releases and the deployment is done to the multiple environments. By using the data management script, it is possible to initialize

the data model multiple times, update the existing data model by controlled way and efficiently load data to the database. The properly developed script is self-documenting the database schema and all the database objects are clearly visible from the script. This is benefit from the project documentation point of view since the script itself is working as a document.

The data management script can be used to manage the data model in different database systems such as an ArcGIS filegeodatabase and a relational SQL database. In practice that provides an option to use the single script for the several database environments without modifications. This is the major benefit especially when the requirement is to deploy solution to the multiple environments.

The disadvantage of the automated data management is the time and work effort required for the script implementation. The work effort could be a key issue when the desired solution is deployed to a single machine environment. Impact to the work effort decreases when the number of the deployment environments increases.

Technically Python provides easy to use tools to implement the GIS database management script. The Python ArcPy sitepackage contains all the commonly needed functions to initialize the data model and load data to the database. The data management script is relatively simple, fast to implement and the script can be managed with a basic Python technical knowledge.

The certain helper functions should be developed in order to get the script to run reliable. The helper functions include a logic such as exception handling and logging. It is also good to include a logic to detect the existing database schema objects when creating new tables and attribute fields. This will give a possibility to use the same script for the data model creation and also for the data model updating.

6.4 Configurations Phase

The configuration phase of the deployment setup covers the script functionality to publish the ArcGIS server Map services. The role of the configuration phase is to implement a script that can be used to publish predefined set of the map services to the several kind ArcGIS Server environments.

The technology selected for the configuration phase is Python scripting. The ArcGIS ArcPy sitepackage contains necessary functionality to connect to an ArcGIS Server management interface. The management interface contains functionality to publish and modify map services. The service publishing process can be automated by using the ArcPy package and its ability to connect to the management interface.

6.4.1 Designed Map Service Model

The service model designed for the deployment setup is based on the data model behind of it. The service model contains key elements to validate functionality of the deployment script. The service model has a three separate ArcGIS Map Services that includes few map layers. The role of map services are to provide datasets for the different purposes such as cables only or the full model. Figure 17 shows the map service model of the deployment package POC.

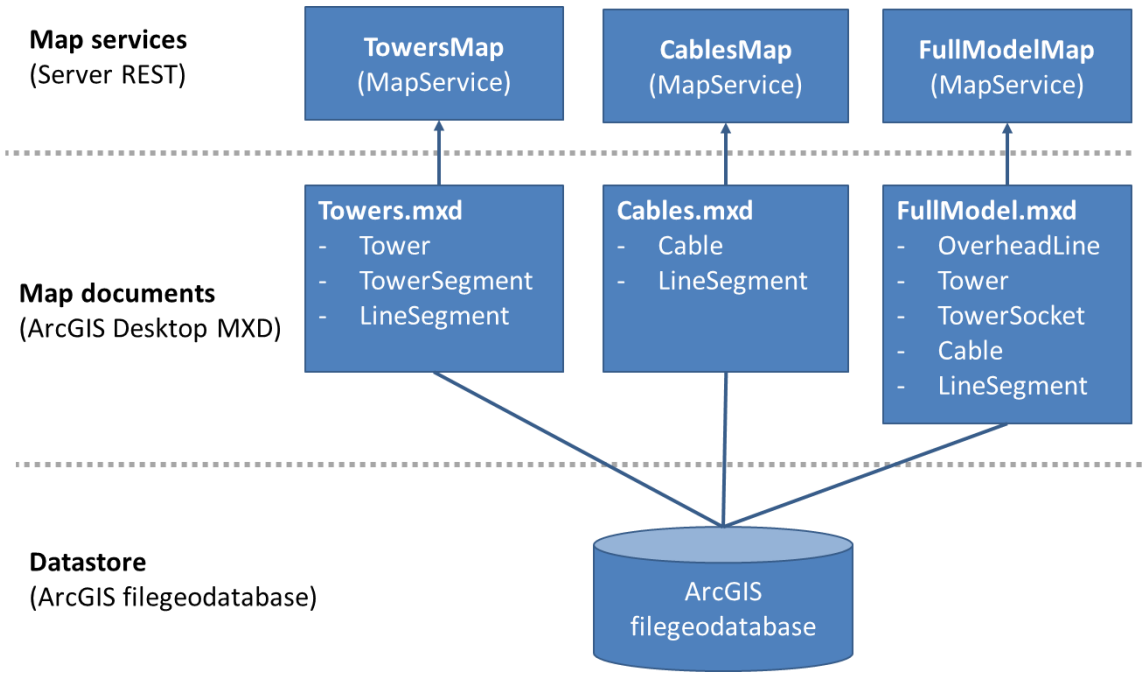

Figure 17. Map service model.

Behind the service model is the ArcGIS file geodatabase that contains all the required tables. The Map services are built by using the ArcGIS Desktop application and then published to the ArcGIS Server. The service model contains a three separate Map services that provides separate views to the data content.

6.4.2 Python Configuration Script in Details

The Python script used to publish the Map services where implemented based on the Esri best practices. The implemented script is capable of publishing multiple map services at once from the source MXD documents. The script reads source documents from the specified folder and runs the required publishing workflow for each source document. Two parameters are passed to the script: 1) the source document folder and 2) the ArcGIS Server management interface address. The script uses the source document file name as a name of the map service. The service publishing process follows the “Service publishing workflow (b): Script-oriented” described in the section five “Service Publishing Workflows”.

6.4.3 Summary

The key benefits of the script based configurations are the option to publish the multiple ArcGIS Server services at once by using the single script. This gives most value in multiple machine environments where the same services are published multiple times to the separate machines.

The key disadvantages are the work and time effort to implement the script. Typically, an ArcGIS Server service publish process contains a lot of changing variables. That generates challenges for the script development. Especially the most challenging case is when the deployment environment contains several machines with the different service data sources. In that case the service MXD source documents have to be modified to point for each data source in order to run the script.

7 Conclusions and Recommendations

The objective of this thesis was to improve the software deployment process. This was done by researching options to improve the current approach to a more automatic and sophisticated process. The aim was to decrease the deployment time, improve the quality of the deployments and make the deployment process more streamlined. The focus and scope of the thesis was limited to the Esri COTS products and their deployments. Deployments of the fully customized solutions and components were not included in the study. To reach the objective, the current deployment process was analysed and described, literature on various kinds of deployments where investigated and a proof-of-concept was implemented.

When discussing the automated deployments with the project management the key concern was: “When the automated deployments should be used and what kind of projects have benefits from the automated deployments?”. This concern was based on the fact that almost all kinds of automation require work efforts to setup. So there is a barrier that needs to be exceeded. The challenge was to clarify whether the project receives benefits from the automated deployment process or is it better to use the manual approach?

The project complexity and requirements are the key concerns to take into an account when talking about the benefits of the automated deployments. As a general guideline one can state that big projects gain more benefits compared to small ones. The benefits of the automated deployment process increases when the requirement is to deploy a solution to multiple environments, multiple machines and/or multiple development releases. This is based on the required number of deployment steps between the different deployment approaches. In small projects a typical issue is that there is no time for the extra hours without measurable benefits. Setting up the deployment framework can take too much time compared to the manual approach. Figure 18 illustrates the benefits of automated deployments.

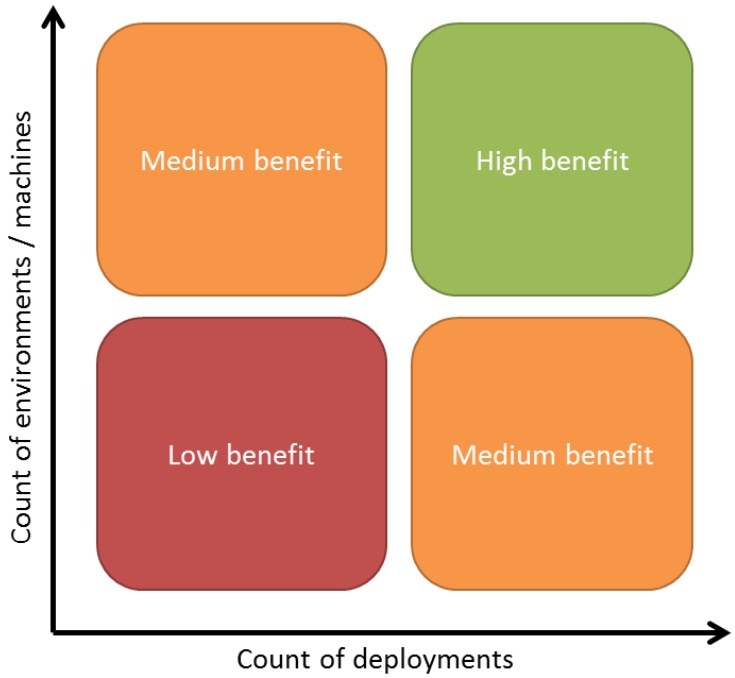

Figure 18. Overall picture of the benefits of automated deployments.

Figure 18 presents the overall picture of the benefits compared to the complexity of the deployment. The benefits of the automation are get higher when the number of deployments and environments increases.

As an outcome of this thesis the deployment process can be divided into the three separate phases: 1) product installations, 2) data in different terms and 3) configurations. A typical deployment project in the case company includes all these main phases. The aim of these phases is to help project management to make a decision how widely automated deployments should be used and to investigate what parts of the deployment process should be automated. Table 4 shows the recommended deployment methods for the deployment scenarios and phases.

Complexity definition	*Installations*	*Data*	*Configurations*
Complexity definition (a) Simple	Manual	Script	Manual
Complexity definition (b) Medium	Script	Script	Manual
Complexity definition (c) Complex	Script	Script	Script

Table 4.Recommended methods for the deployment scenarios and phases.

Table 4 presents the deployment scenarios and summaries the recommendation whether to use a manual or a script based deployment model. The deployment scenarios were described in more detailed level in Chapter 4.

Based on this research it can be stated that generally the most beneficial deployment phase to automate is the data phase. By automating the data phase gives the benefits almost to all kind and size of projects. A work effort to automate the data phase is relatively low since the required script is simple. In addition, the script work as a data model document that can be included to the project documentation. This was based to the POC implementation, see Chapter 6 for details.

The automated installation process is most beneficial in clustered environments that contain multiple identical machines. In a single machine environment, it is probably better to run installation by using a manual approach. Setting up the Chef deployment framework for installation is taking a noticeable work effort. In addition, the setup is not easy to replicate since the Chef installation framework must be installed separately to every machine in the deployment setup.

Automating the configuration phase of the deployment process provides benefits if the requirement is to deploy a large amount of services and/or deploy multiple releases. The most noticeable challenge of the automated configurations is a scenario where a deployment setup contains multiple environments with separate data sources (databases). In this scenario every environment requires individual source MXD documents where services will be published. That noticeable complicates the automation of the deployment. Figure 19 illustrates the benefits of the deployment phases related to the project complexity.

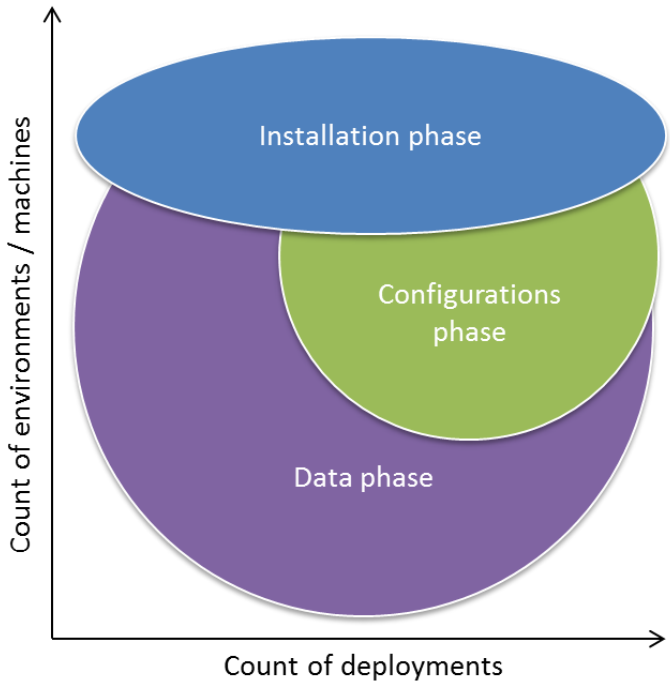

Figure 19. Benefits of the deployment phases related to the project complexity.

The benefits of the installations are high when the deployment setup contains multiple machines. The benefits of the configurations are high when the deployment setup contains multiple releases to deploy. The data phase provides high benefits to projects of all sizes.

7.1 Validity and Reliability

As a conclusion, this thesis provides valuable background information for the future work. This information can be used for example to the decision making, to improve the deployment process and to develop better productized deployment services. This thesis also provides practical information of the suitable technologies and a base for the deployment script implementation. This will help future development projects to implement deployment scripts. The thesis was focused on the Esri enterprise GIS products and it provides information related to the challenges on the COTS deployments.

A current state analysis became one of the central topic of this thesis. At the beginning, there was very little documented information related to the deployment cases and the workflows. In the current state analysis, a typical deployment process workflow was analysed and divided into parts. The analysis was done by facing to the actual deployment

projects and the interviews and discussions with the colleagues. Since deployment is a wide topic the analysis was strictly scoped into the enterprise GIS products and especially to the COTS products. The outcome of the current state analysis was a high level model of a typical deployment process including key parts of it and typical deployment scenarios. Because of the wideness of the deployment topic, the result of the current state analysis was naturally not covering everything or every project. It gives a rough understanding about the most typical scenarios of the deployments and documents key aspects of the deployment process. These results can be used in future to optimize deployment workflows, work estimates and productized service packages.

Technical options of the deployments were investigated and analysed in order to have an understanding about the suitable technologies for Esri deployments. Analysis were focused to the best practices recommended and supported by Esri. Third part deployment frameworks were scoped out from this thesis. In future, technical analysis can be continued by analysing also suitability of the third part deployment frameworks. The aim of the technical analysis was to increase knowledge of the technical options and to investigate suitability of options for the COTS software deployments in the case company business. The result of the technical analysis gives information on what different options are suitable for the Esri product deployments in the case company and what benefits and disadvantages they have.

A deployment package Proof-of-Concept was implemented to verify the practical functionality of the selected technical options. The POC covered deployment parts identified in the current state analysis section. The POC was implemented by using two separate technologies: Chef cookbooks and Python scripting. By using these technologies an identified deployment setup was studied. This deployment setup included key enterprise GIS products used in the case company deployment projects.

The result of the deployment package POC provides technological base for coming productized deployment scripts. The POC gives also practical information related to the usability and suitability of available technological options. The POC was implemented and run on a virtual machine environment over Windows 7 operating system. That is not exactly the same environment that is typically used in the practical projects. For that reason, it is possible that in the actual projects there will come issues and challenges that were not identified on the POC. For example, Linux based environments were not covered by this POC. This concerns especially a software installation part. Results of the

database management part were used in an actual project by the researcher. Experience of using a script based database management was encouraging and the recommendation is to use it as much as possible.

References

[1] Mika V. Mäntylä and Jari Vanhanen, Software Deployment Activities and Challenges – A Case Study of Four Software Product Companies, Software Maintenance and Reengineering (CSMR), 2011 15th European Conference, pp. 131 – 140.

[2] A. Mockus, P. Zhang, P.L. Li and A. Res, Predictors of customer perceived software quality, Proceedings. 27th International Conference on Software Engineering (ICSE), 2005. 2005, pp. 225- 233.

[3] S. Jansen and S. Brinkkemper, Definition and Validation of the Key process of Release, Delivery and Deployment for Product Software Vendors: turning the ugly duckling into a swan, 22nd IEEE International Conference on Software Maintenance, 2006. ICSM'06, 2006, pp. 166-175.

[4] N. Rozanski and E. Woods, Software Systems Architecture, Second Edition, Pearson Education, Inc. 2012, pp. 224.

[5] Talwar, V., Qinyi Wu ; Pu, C. Wenchang Yan, Gueyoung Jung, Milojicic, D. , Comparison of Approaches to Service Deployment, Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference, June 2005, pp. 543 - 552

[6] Martyn Shuttleworth (Apr 1, 2008). Case Study Research Design. Retrieved Apr 26, 2015 from Explorable.com: https://explorable.com/case-study-research-design, accessed 26.4.2015

[7] Per Runeson & Martin Höst, Guidelines for conducting and reporting case study research in software engineering, 19 December 2008

[8] TIMOTHY C. LETHBRIDGE, SUSAN ELLIOTT SIM, JANICE SINGER, Studying Software Engineers: Data Collection Techniques for Software Field Studies, Empirical Software Engineering, 10, 311–341, 2005

[9] CAITLIN DEMPSEY MORAIS, What is GIS?, 2012, http://www.gislounge.com/what-is-gis, accessed 26.4.2015

[10] GIS and Business Intelligence: The Geographic Advantage An ESRI White Paper, September 2006

[11] Martin Nöllenburg, Geographic Visualization, Karlsruhe Institute of Technology (KIT)

[12] Alan M. MacEachren and Menno-Jan Kraak, Research Challenges in Geovisualization, Cartography and Geographic Information Science (draft, Nov. 20, 2000)

[13] CAITLIN DEMPSEY MORAIS, History of GIS, http://www.gislounge.com/history-of-gis, accessed 12.10.2015

[14] P. Fu and J. Sun, Web GIS Principles and applications, 2011 Esri Press

[15] Roger Tomlinson, Thinking About GIS, Geographic Information system Planning for Managers, 2007 Esri Press

[16] Michael Zeiler and Jonathan Murphy, Modeling Our World, The Esri Guide to Geodatabase Concepts, Second Edition, 2010 Esri Press

[17] Paul A. Zandbergen, Python Scripting for ArcGIS, 2013 Esri Press

[18] David W. Allen, GIS Tutorial for Python Scripting, 2014 Esri Press

[19] All about Chef, https://docs.chef.io, accessed 14.9.2015

[20] Announcing Chef, https://www.chef.io/blog/2009/01/15/announcing-chef, accessed 14.9.2015

[22] http://www.esri.com/software/arcgis, accessed 17.9.2015

[23] M. Morisiod,*, C.B. Seamanb,c, V.R. Basili a,c, A.T. Parrae, S.E. Kraftf, S.E. Condone, COTS-based software development: Processes and open issues, 1 September 2001, The Journal of Systems and Software 61 (2002) 189–199

[24] Just What, Precisely, is COTS?, http://www.scientificcomputing.com/articles/2011/01/just-what-precisely-cots, John R. Joyce, Ph.D., accessed 18.9.2015

[25] P Fu, J Sun, Web GIS Principles and applications, 2011 Esri Press, pp. 33-48

[26] Cloud Deployment Models – Read the important differences, http://cloudtweaks.com/2012/07/4-primary-cloud-deployment-models, accessed 7.2.2016