Institute of Architecture of Application Systems
University of Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diploma Thesis No. 3512

Evaluation of a Methodology for Migration of the Database Layer to the Cloud based on an eScience Case Study

Xiaolei Guo

Course of Study: Computer Science

Examiner: Prof. Dr. Frank Leymann
Supervisor: Steve Strauch
Commenced: June 20, 2013
Completed: December 20, 2013

CR-Classification: C.2.4, C.4, H.2.4

Abstract

In the last years Cloud computing has become popular among IT organizations. Applications and data can be designed to be run on the Cloud, and can also be partially or completely migrated to the Cloud. The application can be separated into 3 layers: the presentation layer, the business logic layer and the data layer. The data layer can further be separated into 2 sub-layers: the data access layer and the database layer. Migration tasks are applied on these layers.

In order to benefit from the Cloud technology, migrating the data layer or the business logic layer or both of them into the Cloud is required. For the migration there are different migrating scenarios and patterns. The migrating methodologies are different cross different providers. During the migration procedure different problems may occur. The types, causes, and solutions of these problems may also be different. Identifying the properties of these problems and comparing the migrating methodologies are important to the Cloud users.

In this thesis we migrate both the data layer and business logic layer of a Scientific Workflow System to the Cloud. The migration scenario "Cloud Bursting" is considered. Several quantitative and qualitative metrics chosen from an ISO/ICE standard are introduced and are used to define the properties of the migrating methodologies. Two migrating methodologies are applied, one is the methodology of Bachmann, the other one is the methodology of Amazon. The required refactoring of the application architecture is investigated and the needed modifications are recorded and explained. Data of the metrics of the two migrating methodologies are collected, compared, and discussed.

Contents

List of Figures

List of Tables

List of Listings

1 Introduction

Cloud computing has become more and more popular in recent years. It is not only used for personal usage, but also in the enterprise area. Because a data store and computing power can be provided by the Cloud as Services, and these Services are provided with relatively lower price, migrating the data and tasks to the Cloud becomes a trend. People use applications to do the tasks. In order to construct the application efficiently, workflow was introduced. A workflow is a business process in execution in a computing environment. a scientific workflow is used in the scientific area. In order to benefit from the Cloud computing, application and data migration from local to the Cloud and vice versa is an important issue, and is also a very hot topic.

1.1 Aim and Motivation

When the scientists want to do their scientific tasks, computing power and storage space for data are always important. The decision on how much resources are required is also hard to make, because high computing power and huge storage space always imply enormous cost. Cloud computing can solve this problem, because people only need to pay for the resources that they have used. This "pay-as-you-go" way can reduce the total cost and meet the requirements of every single task. Thus how to migration the scientific workflow and data to the Cloud is the key problem to accomplish the whole task.

This thesis discusses how to migration data layer and business logic layer to the Cloud in certain scenarios such as Cloud bursting [Bac12]. The migration method and tools are also mentioned. In order to evaluate the migration process, some evaluation standards are introduced and the collected migration evaluation results are analyzed and discussed.

In order to migration data into the Cloud and between Cloud services, Bachmann [Bac12] has introduced a methodology and developed a tools for migration of database. Migration scenarios, techniques and patterns are covered in his diploma thesis. Evaluation on his methodology and tools is also one of the tasks of this thesis.

1.2 Definitions and Conventions

In the following section we list the definitions and the abbreviations used in this diploma thesis for understanding the description of the work.

Definitions

List of Abbreviations

The following list contains abbreviations which are used in this document.

ADMC	Average Delay of Milestone Completion
AMI	Amazon Machine Image
ASCE	Average Severity of Code Errors
ASDE	Average Severity of Development Errors
AWS	Amazon Web Services
Axis2	Apache eXtensible Interaction System v. 2
BPEL	Business Process Execution Language 2.0
CED	Code Error Density
CMMI	Capability Maturity Model Integration
CSI	Continual Service Improvement
CSF	Critical Success Factors
DBMS	Database Management System
DAL	Data Access Layer
DBL	Database Layer
DED	Development Error Density
DERE	Development Error Removal Effectiveness
DRE	Defect Removal Effectiveness
EBS	Elastic Block Store
ITSM	IT Service Management
JEE	Java 2 Platform, Enterprise Edition
JSP	JavaServer Pages
KPI	Key Performance Indicator
LAMP	Linux, Apache, MySQL and Perl/PHP/Python
LOC	Lines of Code
KLOC	One Thousand Lines of Code
MTTF	Mean Time To Failure

NCE	Number of Code Errors
NDE	Number of Development Errors
NFP	Number of Function Points
NIST	National Institute of Standards and Technology
NoSQL	Not only Structured Query Language
ODE	Orchestration Director Engine
ODE-PGF	ODE-Pluggable Framework
PUM	Problem per User Month
QoS	Quality of Service
QME	Quality Measurement Element
RDBMS	Relational Database Management System
RDS	Relational Database Service
REST	Representational State Transfer
SDLC	Systems Development Life Cycle
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol *(deprecated)*
SPICE	Software Process Improvement Capability Determination
SQL	Structured Query Language
SQuaRE	Software product Quality Requirements and Evaluation
TTO	Time Table Observation
WCE	number of Weighted Code Errors
WDE	number of Weighted Development Errors
WCED	Weighted Code Error Density
WDED	Weighted Development Error Density
WCEF	Weighted Code Error per Function
WDEF	Weighted Development Error per Function
WSDL	Web Services Description Language
WS-BPEL	Web Service Business Process Execution Language

1.3 Outline

The remainder of this document is structured as follows:

- **Fundamentals, Chapter 2:** concepts, principles, technologies, prototypes

- **Related Work, Chapter 3:** What has been done so far

- **Design, Chapter 4:** the design of prototype

- **Implementation, Chapter 5:** detailed description of the design

- **Evaluation, Chapter 6:** evaluation of the testing implementation

- **Results and Future Work, Chapter 7:** analysis of the results and outlook for further development

2 Fundamentals

In this chapter the fundamental concepts and technologies are discussed. These are used in this diploma thesis. Some useful tools are also introduced here.

2.1 Cloud Computing

According to the definition of National Institute of Standards and Technology (NIST) [MG11], Cloud computing is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources ", resources can be provisioned and released with minimal efforts. It has five main characteristics: on-demand self service, broad network access, resource pooling, rapid elasticity and measured service.
Three main types of *Service Models* of Cloud computing are as follows:

- Software as a Service (SaaS): With this model the consumer is able to use the provider's applications running on a Cloud infrastructure.

- Platform as a Service (PaaS) : With this model the consumer is able to create and deploy applications using programming languages, tools and libraries provided by the provider.

- Infrastructure as a Service (IaaS). With this model the consumer is able to use fundamental computing resources such as operating system, storage and network.

According to the deployment model, Cloud computing can also be divided into 4 types:

- Private Cloud: used by a single organization.

- Community Cloud: used by a specific community from organizations that share common concerns.

- Public Cloud: used by the public.

- Hybrid Cloud: a composition of two or more distinct cloud infrastructures.

To fulfill the flexibility of the Cloud applications, the n-tier architecture model is usually used so that people can easily add new layer when new functionality is needed. The mostly used 3-layer architecture is used here, which consists of a presentation layer, a business logic layer and a data layer. The presentation layer does the job to interact with users, the business logic layer expresses the business logic and the data layer provides data storage service. The data layer can also be divided into 2 sublayers: the Data Access Layer (DAL) and Database Layer (DBL). DAL provides the functionality of accessing to the database, and DBL does the

data persistence and manipulation job. In this thesis the data layer is mainly focused.
Cloud computing provides virtualized computing power to people. It is not necessary to
know where the resource resides. Customers can easily customize services as they wish.
Because of the on-demand and elasticity feature of Cloud computing, payment method such
as pay-as-you-go becomes reality, which means people only pay for services they actually
used. Thus the customers can drastically cut down the costs.

2.2 Service-Oriented Architecture

Service is a function provided at a network address. Service-Oriented Architecture (SOA) is
an specific architectural style to realize Service-Oriented Computing. It has key features such
as loose coupling and dynamic binding between services. It consists of 3 main roles:

- Requester : needs a specific service and sends specific request

- Provider : provides services and their description and publishes the description to
  broker

- Broker : provides service registration and searching service

The provider provides services, publishes information to the broker and makes a registration.
The requester gives the description of the needed service to the broker and looks up an
appropriate service. Then the broker sends the required binding information so that the
requester can connect to the proper service.
Web Services Description Language (WSDL) is usually used for service description. SOAP is
used for message exchange. Originally it means " Simple Object Access Protocol", but now
people just use SOAP.

2.3 Database

Database is widely used for storing data, it is now an embedded component for a computing
environment. The mostly used type of database is the Relational Database Management
System (RDBMS). This type of database is usually used to serve both personal and busi-
ness demands. To serve some special usage requirements, Not only Structured Query Lan-
guage (NoSQL) is also used under some specific circumstances. Not only the database on the
local host can be used, the database provided by the Cloud provider can also do the similar
job. There are also different types of Cloud databases to meet the different needs.

2.3.1 Structured Query Language Databases

The RDBMS is a Database Management System (DBMS) that is based on the relational model. The Structured Query Language (SQL) is the standard computer database language in RDBMS. SQL is a tool for organizing, managing, and retrieving data stored by a computer database [GW99]. SQL consists of data definition language and data manipulation language. The most used operations in SQL are insert, update, query, delete. Schema creation and data access control are also supported.

Microsoft SQL Server , MySQL/MariaDB, PostgreSQL and Oracle are usually used RDBMS. Oracle[1] is a successful commercial RDBMS provided by Oracle Corporation. Microsoft SQL Server[2] is provided by Microsoft Corporation. MySQL, MariaDB[3] and PostgreSQL[4] are open source software and used all over the world.

2.3.2 Databases in the Cloud

A Cloud Database is the database that runs in the Cloud environment. It can resides in a virtual machine in the Cloud , or be provided as a storage Service . According to the data model, the database can be an SQL database or a NoSQL [SF12] database. The following Table 2.1 shows some typical databases in the Cloud.

	Deployed in virtual machine	Provided as a Service
SQL	MySQL , PostgreSQL, IBM DB2	Amazon RDS, Microsoft SQL Azure
NoSQL	CouchDB/Cassandra/MongoDB on Amazon EC2	Amazon DynamoDB, Amazon SimpleDB

Table 2.1: Cloud Databases

When the databases are deployed in a virtual machine, they work identically as they do on a local host. The only difference is the location where they reside. People can easily log on the virtual machine system by using remote connecting tools such as ssh and do the job just as they do on a local machine.

To the contrary, database like Amazon RDS is a Service which provides normal database functions like CRUD. Thanks to the elasticity of Cloud, the database can be easily scaled. The deployment of the database is also different from those ones on local host. This is provided also as a corresponding service interface. Users can easily get the running status of the databases by using the corresponding Service. In a word, the selection, creation, launching,

---

[1]http://www.oracle.com
[2]http://www.microsoft.com/en-us/sqlserver/default.aspx
[3]MariaDB: http://mariadb.org
[4]http://www.postgresql.org

deployment and management for this type of databases are provided by the Cloud provider. The operations are different from those of a database on the local host or in a virtual machine.

2.4 SimTech Prototype

The SimTech prototype is a central element for the case study. It supports scientific workflows. According to the handbook of Hahn [Hah], a brief high-level description of the SimTech prototype is showed in Figure 2.1:
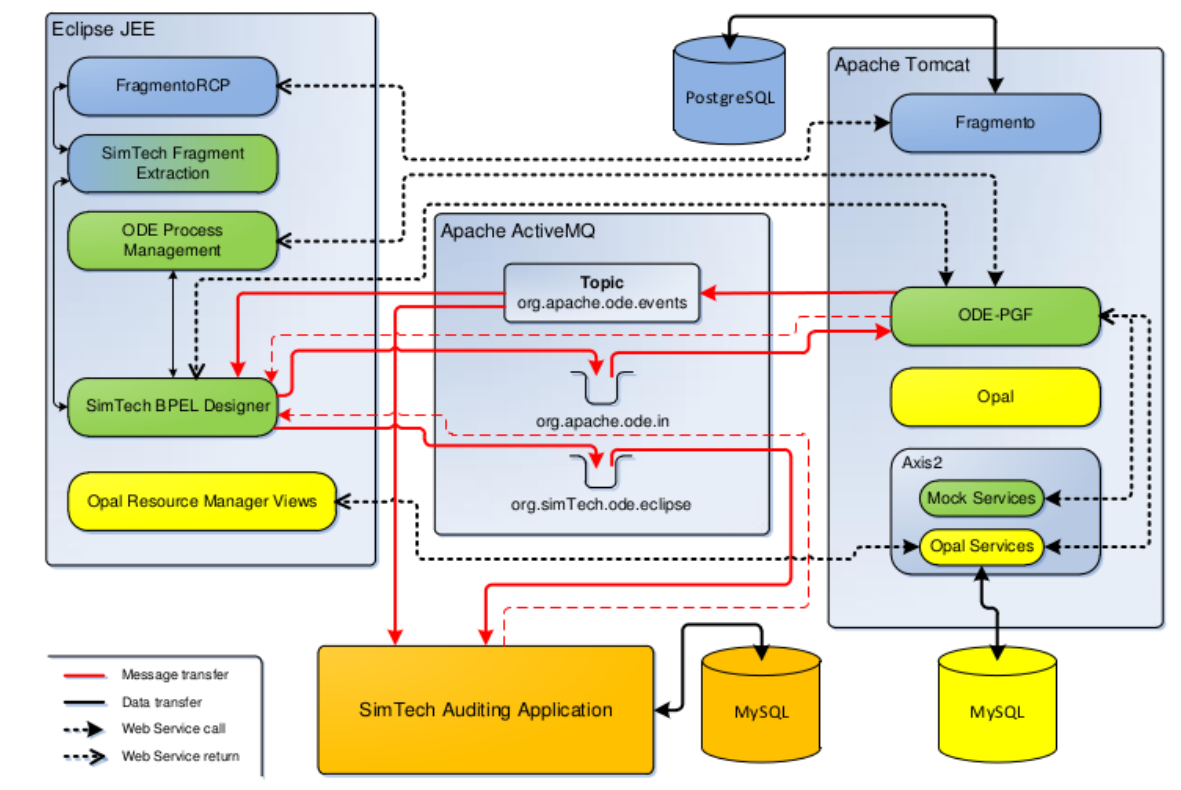
Figure 2.1: SimTech Prototype [Hah]

From this graphic we can see that the SimTech prototype has 3 main components:

- Eclipse Java 2 Platform, Enterprise Edition (JEE) with SimTech extensions: has a Business Process Execution Language 2.0 (BPEL) designer as modeling tool, a resource manager and process manager.

- Apache Tomcat Server : Apache ODE-PGF (Orchestration Director Engine (ODE) Pluggable Framework) works as a workflow engine, Apache eXtensible Interaction System v.

2 (Axis2) is an engine for Web Services/SOAP/WSDL, Opal is a workflow based solid state simulation application.

- SimTech Auditing Application : execution events of the workflow instances will be stored by this application.

All of the three components are loosely coupled and integrated through Web Services and messaging. Apache ActiveMQ is a message server, which connects SimTech Auditing Application with the ODE-PGF. The migration of workflow designer will not be considered in this thesis.

2.5 Migration

In this part the concepts and the phases of data migration will be in detail introduced.

When the migration takes place, we need to know the following aspects:

- In which scenario will the data migration take place.

- Which kind of challenges does a data migration process solve.

- How does a data migration process look like.

Categorized by usage characteristics the data migration can be divided into ten scenarios [SABL13]. For eScience usage the data should be easily moved between local host and Cloud, and the modification on the workflow should be easily adopted to the Cloud. Thus in this thesis only the scenario of "Cloud Bursting" will be focused. "Cloud Bursting" means that database layer can be temporarily outsourced to the Cloud, in this way the additional resources can be used to off-load the peak loads. This is a two-way scenario: the database layer can be migrated from local host to the Cloud, and can also be moved back to local host.

When we migrate data to the Cloud, some challenges such as incompatibilities come out over and over again. These challenges should be resolved. Description of these solutions on an abstract and technology independent level can be referred as patterns. Until now there are three main categories of Cloud Data Patterns [SABL13] :

- Functional Patterns: provide reusable solutions to offer functionality by Cloud data store and services. Because some functionality provided by one data store may not exist in another one. Data Store Functionality Extension and Emulator of stored Procedures are components of Functional Patterns.

- Non-Functional Patterns : ensure an acceptable Quality of Service (QoS) by scaling the data store to increase data R/W load. Local Database Proxy and Local Sharding-Based Router are parts of Non-Functional Patterns.

- Confidentiality Patterns : provide solutions to avoid disclosure of confidential data, especially handling the data moved from/to different data store with different confidentiality levels. Confidentiality Level Data Aggregation, Confidentiality Level Data Spliter, Filter of Critical Data, Pseudonymizer of Critical Data and Anonymizer of Critical Data are parts of Confidentiality Patterns.

Migration scenarios can be mapped to migration patterns [SABL13].

There are two different types of migration methodology tested in student thesis of S.Passow [Pas13].
One is from AWS. Amazon is an existing Cloud service provider. The phase driven approach from AWS [Jin10] is showed in Table 2.2:

Phases	Description
Cloud Assessment Phase	Assessment for costs, architecture and security issue.
Proof of Concept Phase	Building a pilot to validate the technology.
Data Moving Phase	Move of data when one storage solution is chosen.
Application Moving Phase	A migration strategy is selected. for each component one Amazon Machine Image (AMI) is created.
Leverage the Cloud Phase	Automating the elasticity and Systems Development Life Cycle (SDLC).
Optimization Phase	Improvement of efficiency, the on-demand usage is optimized.

Table 2.2: AWS Migration Phases

Another one is from the diploma thesis of Bachmann [Bac12], which is derived from the idea of Laszewski. The detailed information is shown in Table 2.3:

2.6 Software and Components used

In this section we will introduce some software, which will be used in the thesis.

2.6.1 Apache Tomcat

Apache Tomcat[5] is an open source software implementation of the Java Servlet and JavaServer Pages (JSP) technologies.It can be managed by configuration tools or XML configuration files. Apache Tomcat has many components, such as

- Catalina, a Servlet container.

- Coyote, the HTTP connector.

---

[5]http://tomcat.apache.org/

Phases	Operations
Select Migration Scenario	One proper scenario from the 10 scenarios described in [SABL13] will be chosen. Based on this scenario a migration strategy will be formed. In our case, only the scenario of Cloud Bursting will be considered.
Describe Desired Cloud Data Hosting Solution	Requirements for functional and non-functional properties will be specified in this phase. These properties can be divided into different categories such as availability, security .etc.
Select Cloud Data Store or Data Service	According to the specifications that gathered in last phase, a data store or data service the migration will be selected.
Describe Source Data Store or Data Service	In order to solve migration conflicts between source and target data stores, the properties of the source data store or data service should be specified.
Identify Patterns to Solve Potential Migration Conflicts	In this phase the incompatibilities between target data store and source data store will be identified.
Refactor Application Architecture	In this phase hints about how to refactor the application are introduced. The Adaption usually occurs on network, the data access layer or the business logic layer.
Migration Data	This is the final step: the migration phase.

Table 2.3: Bachmann Migration Phases

- Jasper, the JSP engine.

- Cluster, support for management for large applications.

2.6.2 Apache ActiveMQ

Apache ActiveMQ[6] is a powerful open source messaging and integration patterns server.It supports cross language clients and protocols. The enterprise integration patterns [HW03] means a number of design patterns for the use of enterprise application integration and message-oriented middleware. It can be implemented by using Spring Integration, Camel library. Apache ActiveMQ supports powerful patterns such as smart routing and transformation.

2.6.3 Apache Axis2

Apache Axis2[7] is a Web Service / SOAP / WSDL engine. It supports SOAP 1.1, SOAP 1.2, WSDL 1.1, WSDL 2.0 , Representational State Transfer (REST) style Web Services, Spring

---

[6]http://activemq.apache.org/
[7]http://axis.apache.org/

framework .etc. Various protocols such as SMTP, FTP are also supported. The main components of Apache Axis2 is Axis2-core, Rampart (providing features like reliable messaging and security) , Sandesha (an implementation for Web Service reliable messaging).

2.6.4 MySQL

MySQL[8] is an open source RDBMS which is commonly used as the core component of Linux, Apache, MySQL and Perl/PHP/Python (LAMP) web application software stack. It can run on a lot of platforms such as Linux, Windows, Mac OSX, Free-/OpenBSD. It supports SQL. A lot of websites such as Wikipedia use MySQL.
MySQL can be used in a virtual machine in the Cloud, or as a Web Service provided by the Cloud provider.
Although MySQL is open source software, it has commercial versions. Additional functionality is supported.

2.6.5 ODE-PGF

A workflow can be described by using Web Service Business Process Execution Language (WS-BPEL). Process event monitoring and adapting operations are provided by workflow engines, but most engines do this in a proprietary way. The pluggable framework for extended BPEL behavior fills this gap and provides a standardized BPEL event model and ways to change the engine's behavior upon receipt of an event. Apache ODE[9] executes business process written in WS-BPEL standard, it is a workflow engine. The ODE-PGF[10] is an implementation of the pluggable framework for Apache ODE engine.

2.6.6 Opal

Opal[11] is a workflow-based solid state simulation application. The structural changes of metallic solid state over a long period can be simulated.

2.6.7 Amazon RDS

Amazon RDS [12] is a service that enable people easily creating, operating and scaling a database in the Cloud. MySQL, Oracle or Microsoft SQL Server are provided.

---

[8]http://www.mysql.com/
[9]http://ode.apache.org/
[10]http://www.iaas.uni-stuttgart.de/forschung/projects/ODE-PGF/
[11]http://www-01.ibm.com/software/tivoli/features/opal/
[12]http://aws.amazon.com/rds/

2.6.8 Amazon EC2

Amazon EC2 [13] stands for "Amazon Elastic Compute Cloud". Amazon EC2 provides a virtual computing environment with user-friendly tools. Virtual machine images of different operating systems can be provided here. Custom machine image with specified applications can also be created. The resources of Amazon EC2 can be used on-demand.

2.6.9 Amazon EBS

Amazon EBS [14] is a storage service provided by Amazon. When a application needs file system, Amazon EBS can meet the requirement. It can be attached to an Amazon EC2 instance and be used as a virtual storage device.

---

[13]http://aws.amazon.com/ec2/
[14]http://aws.amazon.com/ebs/

3 Related Work

This chapter mainly discusses the related work that describe the evaluation of processes and evaluation of tools. Some metrics are also introduced.

3.1 Evaluation Basics

In this section some basic concepts of evaluation will be introduced. Before an evaluation is planned and applied, several core questions should be answered [TPS96, Sto04] :

- What is the purpose of the evaluation? The most common goal is to find defects and to make improvement of the software.

- What tasks should be done? This question involves the information gathering, planning, evaluation operations arrangement, etc.

- Who will use the evaluation results? How do they use these results? The relevant people can be developers, business stakeholders, etc. They can do operations like analyzing, documenting.

- What do these people want to know? One common thing that people want to know is whether the software product meets the requirements in different aspects.

In order to perform evaluation on the software product, data should be gathered. There are a lot of sources that can be used such as existing documents or relevant people. A lot of evaluation methods can also be applied, such as doing a survey, making group assessment, testing .etc. When we choose a method, things like the evaluation purpose, time, cost, type of the gathered information should also be considered.
There are two types of evaluation paradigm that are introduced in [Sto04]:

- Empirical scientific methods. These methods follow the critical rational research logic and all of known empirical research methods can be applied.

- Constructing "reality" from different perspective. When a dispute exists about the "reality" from the empirical scientific method, a social constructing method can combine the "reality" from different perspective together. Conflict can remain untouched.

When the empirical scientific methodology is used on a software process, the process should at first be well defined. Thus, operations like observation and measuring can be performed. Because there is no one-size-fits-all process, iteration definition is needed in the process definition. When a entire process is evaluated, the global lessons which affect the entire

process can be grasped. The specific lessons which affect a single step can also be obtained. The methodology for empirical validation of processes can be divided into 4 parts:

- Feasibility study: can usable results be generated? Was the time well spent? Is the process cost-effective?

- Observation study: are these steps of the process meaningful? Data can be collected so that people can tell how a particular task is accomplished. One type of the data is observational data, which can be obtained during executing the process. Another type is inquisitive data, which can be got at the completion of the process.

- Case study on using the process in real lifecycle : do the fine-tuning or tailoring of the technology.

- Case study on using the process in the industry environment to check whether it fits the industrial setting.

Case study can be applied on process to evaluate the benefits on the methods and tools of the process. It is a cost-effective way to make sure of getting the desired results. The steps of case study are [KPP95]: At first the context (objectives, baseline and constrains) of the project should be identified. Then a hypothesis should be clearly defined: supposed effect, measurable terms and what not supposed to happen. After that a typical pilot project is selected. This project can represent the type of project that organizations or companies undertake. Then the method of comparison should be identified. A sister project can be chosen to make comparison, or a method against the baseline. The effect of confounding factors should be minimized. The effect of confounding factors can not be easily distinguished from one to another. People need to learn how the evaluation method works, the effect of the steps, and the working mechanism. Then the plan of the case study should be made. It should include requirements, procedure, people involved, data to be collected, analyze. The case study is monitored against the plan. After that a report should be written, the results should be analyzed and reported.

When a evaluation method is applied, both quantitative data and qualitative data can be gathered. Quantitative data that is usually with numerical values is used to measure a particular aspect of a process, and qualitative data which uses expressions can make people understand the process better.

Metrics are a system of parameters or ways of quantitative assessment of a process that is to be measured, along with the processes to carry out such measurement. Metrics define what is to be measured [CS11]. According to Daniel Galin [Gal04] , the quality metrics are consisted of two parts: the quantitative measure to a quality attribute and the function which can translate the software data to the numerical value. The objectives of metrics are to facilitate the management control and to identify the different required situations in the development and maintenance process. Software metrics need to be relevant to the attributes, valid, reliable, applicable to different situations, mutual exclusive, easy, simple, immune to the interventions from a third party. When metrics are measured, the results could be analyzed.

3.2 Evaluation of Processes

In this section the evaluation of processes will be discussed. Metrics for process evaluation will be also introduced.

3.2.1 CMMI, SPICE and ITIL CSI Concept

Capability Maturity Model Integration (CMMI) [Gec] is a quality model for software process evaluation and improvement. The advantages and disadvantages can be located after the assessment. CMMI basically has 2 mature grades modes. One is staged mode, the other one is continuous mode.

For the staged mode there are 5 phases: the initial phase, the managed phase for dividing project and process into sub-projects and sub-processes, the defined phase about organization for development activities, the quantitatively managed phase to use metrics, and the optimizing phase to apply innovation and casual analysis.

For the continuous mode the sequence is similar, there are 6 phases: the incomplete phase, the performed phase, the managed phase, the defined phase, the quantitatively managed phase and the optimizing phase.

The problem with CMMI is that it has a large number of processes, which leads to big complexity. The maintenance and change management capabilities are also missing.

Software Process Improvement Capability Determination (SPICE) [Gec] is a generic assessment model for software development process. SPICE has 2 dimensions: process dimension and mature grade dimension. The process dimension is divided into 5 categories: the customer-supplier process, the development process, the supporting process, the management process and the organizational process. The mature grade dimension is divided into 6 levels: incomplete, executed, controlled, defined, predictable and optimized. The difficulty of the SPICE model is that the cost is relatively high and due to the strict definition, this model is not flexible.

The next to be introduced quality evaluation model is the ITIL Continual Service Improvement (CSI) model. ITIL [ITI] is the most widely accepted approach to IT Service Management (ITSM) in the world. Service management [CS11] is capable for providing value to customers in the form of services in a specialized organizational way. Services (in form of functions and processes) over a lifecycle are managed with specializations in strategy, design, transition, operation and continual improvement. CSI is part of ITSM. CSI process can learn from past successes and failures by using methods from quality management. When the services and service management are improved, the benefits of the customers will be increased.

ITIL CSI core (in ISO 20000) consists of 5 parts [CS11], which are:

- Service Strategy (SS) : The guidance in this part shows the organizational and strategic view to design, develop and implement service management.

- Service Design (SD) : The guidance here indicates the way to design and develop services and service management processes. achievement service levels, standards, regulations are included.

- Service Transition (ST) : this part gives guidance on transitioning new and modified services into operations.

- Service Operation (SO) : this part includes guidance about effectively and efficiently delivering and supporting services to ensure value for the customer and the service provider.

- Continual Service Improvement : it is based on Deming Cycle [DR52] (Plan-Do-Check-Act). SD → ST → SO → SD, where SS stays in the center, just as shown in Figure 3.1. This part provides guidance on how to create and maintain value for customers. Services should be better designed, introduced and operated.
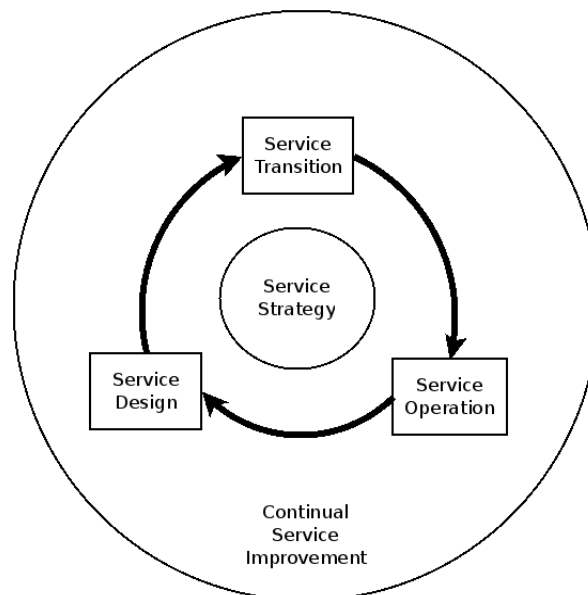
Figure 3.1: ITIL Core [CS11]

The main steps of CSI process are :

- Define what should be measured. A project can be set up and a migration strategy can be chosen.

- Define what can be measured. The required qualitative and quantitative metrics can be chosen.

- Gather data.

- Process data (measurements of metrics).

- Analyze data.

- Present and use the information.

- Implement corrective action.

In this thesis the steps of the evaluation according to the steps above are:

- At first set up the evaluation project, determine what kind of a project is to be evaluated. Here the methodology of Bachmann and the tool developed by Bachmann are to be evaluated.

- Define which metrics need to be measured. Metrics such as usability should be mentioned.

- Implement the evaluation and collect the data.

- Do the calculation on the collected data to get the meaning of these data.

- Analyze the results and do further steps such as presenting the results and improving the methodology and tool.

When using the CSI process, the metrics of CSI should also be mentioned. Three core types of metrics for CSI are:

- technology metrics: performance,availability .etc.

- process metrics: Critical Success Factors (CSF) and Key Performance Indicator (KPI) are usually used. 4 key aspects that KPI can help to solve are about quality, performance, value and compliance. The relationship between CSF, KPI and metrics can be seen in Figure 3.2.

- service metrics: these are results of end-to-end services.
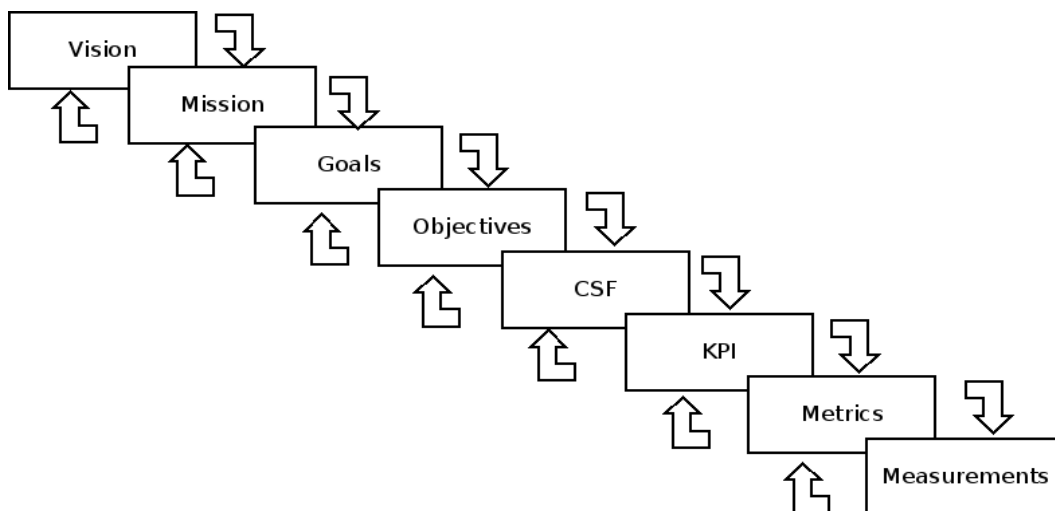
Figure 3.2: ITIL Vision to Measurements Structure [CS11]

3.2.2 Metrics for Process

There are several usually used metrics for process, which are in [Kan04] introduced. These metrics are showed in Table 3.1. The related explanation is also presented in the table.
 In this thesis we mainly focus on the qualitative metrics in the categories of usability and

Categories	Details
Defect Density During Machine Testing (after code is integrated into the system library)	Defect occurrence rate becomes lower or higher than previous period?
	Defect occurrence rate becomes lower or higher than same part of last release?
Defect Arrival Pattern During Machine Testing: different patterns of defect arrivals indicate different quality levels in the field.	How many defect arrives (reported with raw number) during the testing phase by time interval?
	What pattern are the defect arrivals? Are these true defect pattern?
	How big is the defect backlog overtime? The metric is needed because defects can not be fixed immediately.
Phase-Based Defect Removal Pattern: extension of the test defect density metric.	The tracking of defects at all phases of the development cycle is needed. The main components are the design reviews, code inspections, and formal verifications before testing. It is better to find defects earlier.
Defect Removal Effectiveness (DRE)	$$\text{DRE} = \frac{RD}{LD} \times 100\%$$ RD = the number of removed defects during the development phase, LD = the number of the latent defects in product.

Table 3.1: Process Metrics 1

functionality of the methodology and tool of Bachmann, so Defect Density During Machine Testing and Defect Removal Effectiveness will be considered.
In [Gal04] similar metrics are also introduced. In Table 3.2 these metrics are explained.

Categories	Sub-categories	Details
Software Process Quality Metrics	Error Density Metrics	The needed metrics are: One Thousand Lines of Code (KLOC), Number of Code Errors (NCE), Number of Development Errors (NDE), number of Weighted Code Errors (WCE), number of Weighted Development Errors (WDE), Number of Function Points (NFP). Then the following values can be calculated: Code Error Density (CED), Weighted Code Error Density (WCED), Development Error Density (DED), Weighted Development Error Density (WDED), Weighted Code Error per Function (WCEF) and Weighted Development Error per Function (WDEF). $$CED = NCE \div KLOC$$ $$WCED = WCE \div KLOC$$ $$DED = NDE \div KLOC$$ $$WDED = WDE \div KLOC$$ $$WCEF = WCE \div NFP$$ $$WDEF = WDE \div NFP$$
	Error Severity Metrics	Two metrics are used here: Average Severity of Code Errors (ASCE) and Average Severity of Development Errors (ASDE) $$ASCE = WCE \div NCE$$ $$ASDE = WDE \div NDE$$
	Cyclomatic Complexity Metrics	These metrics determine the maximum number of independent paths needed to achieve full line coverage of the program. The structure of the program can be treated as a graph. 3 Ways can be used for calculation: $$V(G) = N_{closed\_region}$$ $$V(G) = N_{edge} - N_{node} + 2$$ $$V(G) = N_{node\_with\_multiple\_exit} + 1$$

		Here the abbreviation of "MS" means "Milestone".
Software Process Timetable Metrics	Time Table Observation (TTO)	$TTO = MS_{on\_time} \div MS_{total}$
	Average Delay of Milestone Completion (ADMC)	$ADMC = MS_{total\_delay} \div MS_{total}$
Error Removal Effectiveness	Development Error Removal Effectiveness (DERE)	Here NYF indicates the number of failures in a year. $DERE = NDE \div NYF$
Software Productivity Metrics	4 metrics	$P_{development} = Hours_{development} \div KLOC$ $P_{function\_point\_development} = Hours_{development} \div NFP$ $P_{reuse} = KLOC_{reuse} \div KLOC_{all}$ $P_{document\_reuse} = N_{document\_reuse} \div N_{document}$

Table 3.2: Process Metrics 2

As we mainly focus on the qualitative metrics in categories of usability and functionality, Error Density Metrics will be considered. The information about how many errors occur during the evaluation is important to the topic, and how effectively these errors are solved is also important. During the evaluation the number of errors should be recorded. The types of the errors, the solutions to these errors should also be recorded. The time cost during migration is also important. These metrics indicate whether the methodology of Bachmann works and how good it works.

3.3 Evaluation Principles for Software Product

In order to make a proper assessment of a migration solution tool, evaluation standards on quality of software need to be set.

3.3.1 ISO 9126

ISO 9126 series of standards define a general-purpose quality model, quality characteristics and five examples of metrics. 4 main parts of the standards are defined as follows [ISO05] :

- ISO 9126-1 : Quality characteristics and subcharacteristics. When the software is used as a part of a computer system, the subcharacteristics are manifested externally and can be static attributes of the software product. 2-part model are described here: the internal, external software quality (6 subcharacteristics) and quality in use (4 subcharacteristics) .

- ISO 9126-2 : External metrics provide description of metrics from the external perspective when the software is in use. The behavior of the system is measured. Quantitative values such as time and effort are used as basic metrics.

- ISO 9126-3 : Internal metrics can measure internal attributes of the software related to its architecture. These measures can be used as indicators to predict the operation in a system, or as tools to forestalling downstream problems.

- ISO 9126-4 : Quality in use metrics are used to measure the ability of a product that how good it can meet the needs of specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in a specified context of use. The measures can be retrieved by simulating a working environment or observing the operation behavior of the product.

3.3.2 SQuaRE

SQuaRE means Software product Quality Requirements and Evaluation [ISO05]. The aim of its set of standards is to provide an organized, unified series covering two main aspects: software quality requirements, software quality evaluation. SQuaRE provides terms and definitions, reference models, a general guide to the standards and individual guides to each division, and also the standards for requirements specification, measurement and evaluation purposes, planning and management. Software product quality is categorized in SQuaRE quality model in characteristics, characteristic is divided into subcharacteristics, and subcharacteristic is further divided into attributes.

SQuaRE standards consist of 5 main parts, The organization of the 5 parts is as showed in Figure 3.3.
The detailed information about the SQuaRE standards is as follows:

ISO 2500n Quality Management Division

The division defines all common models, terms and definitions that will be referred by other standards in the SQuaRE series. This division also provides requirements and guidance for a supporting function. Thus the management of software product requirements specification and evaluation is possible.
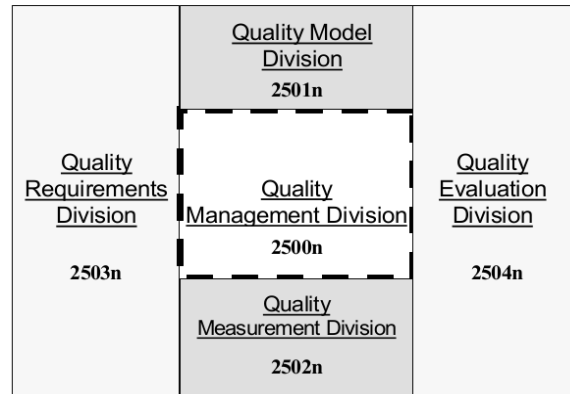
Figure 3.3: Organization of SQuaRE series of standards

ISO 2501n Quality Model Division

A detailed quality model is presented in this division. The model includes characteristics for internal, external, and quality in use. These characteristics can also be decomposed into subcharacteristics. 6 characteristics are important here: functionality, reliability, usability, efficiency, maintainability, portability. These 6 characteristics, together with quality in use, can be called inherent properties. Inherent properties with assigned properties such as price construct the software properties. From bottom to the upper layer, the views of Quality Model are defined as : internal software quality, external software quality, and software quality in use.

The standard introduces five categories of metrics in quality in use model and 8 categories of metrics in product quality model, which is showed in Table 3.3 and in Table 3.4.

Category	Metrics
Effectiveness	effectiveness.
Efficiency	efficiency
Satisfaction	usefulness, trust, pleasure, comfort.
Freedom from Risk	economic risk migration, health and safety risk migration, environment risk migration.
Context Coverage	context completeness, flexibility.

Table 3.3: Quality in Use Model

ISO 2502n Quality Measurement Division

A software product quality measurement model, mathematical definitions of quality measures are included in this division. A guidance is also included. The presented measures can be

24

Category	Metrics
Functional Suitability	functional completeness, functional correctness, functional appropriateness.
Performance Efficiency	time-behavior, resource utilization, capacity.
Compatibility	co-existence, interoperability.
Usability	appropriateness, recognisability, learnability, operability, user error protection, user interface aesthetics, accessibility.
Reliability	maturity, availability, fault tolerance, recoverability.
Security	confidentiality, integrity, non-repudiation, accountability, authenticity.
Maintainability	modularity, reusability, analyzability, modifiability, testability.
Portability	adaptability, installability, replacablity.

Table 3.4: Product Quality Model

applied on internal, external, software quality, and quality in use.

Quality Measurement Element (QME)s "measure defined in terms of an property and the measurement method for quantifying it" [ISO12]. It refers to a measure on a property, the method to measure and quantify the property. QMEs can become the quantification of characteristics and subcharacteristics. A QME should at least contain information like QME name, target name, property to be quantified, relevant quality measures, measurement method, input, unit of the measurement, numerical rules, scale type, intended use of the measurement results, software life cycle process, measurement constrains. A list of the initial set of QMEs also exists, which includes the number of accessible functions, the number of user problems, the number of records, duration, effort, number of system failures, number of failures, number of software faults, functional size of the product, number of interruptions, number of data items, number of error messages, number of incorrect action/step/result, number of messages, number of steps, task complexity, number of test cases, number of use cases, number of operations, number of fatal errors, size of database, size of memory. These QMEs can also be mapped to the characteristics defined in ISO-9126. This mapping can be found in Appendix A in [ISO12].

ISO 2503n Quality Requirements Division

Quality requirements are specified in this division [ISO06] . The requirements can be mapped to technical processes defined in ISO 15288 - Information Technology - Life Cycle Management - System Life Cycle Processes. To meet the needs of stakeholders a definition process is applied by using Quality Model (ISO 2501n). The quality requirements of stakeholders are handled in a analysis process by referring to the quality measures (ISO 2502n), Then the system quality requirements, that are defined in ISO 2503n, need to be met.

ISO 2504n Quality Evaluation Division

This division provides requirements , recommendations and guidelines for software product evaluation. Documenting a measure as an Evaluation Module is also supported [ISO10].

- 25040 Evaluation process

- 25041 : specified requirements and recommendations for developers, acquirers and evaluators.

- 25042 Evaluation modules: the evaluation modules contains information (characteristics, subcharacteristics and the corresponding internal, external, quality in use measures) about the quality model.

- 25045 Evaluation modules for recoverability: specification about the subcharacteristic of recoverability that based on the characteristic of reliability.

An overview of the evaluation process goes like this way: inputs are gathered and given to the evaluation process, the evaluation process proceeds under some constrains (user needs, resources, schedule, cost, environment, tools, reporting) and uses resources (tools/methodology/module, applicable SQuaRE documents, human resources, economical resources, information system and knowledge database) for the evaluation, after that the outcome of evaluation process exists.
The process goes as follows:

- Establish the evaluation requirements

- Specify the evaluation

- Design the evaluation

- Execute the evaluation

- Conclude the evaluation

After step 3 the evaluation plan can be obtained. The evaluation plan should contain the purpose, the organization, the budget, expected product, schedule, responsibilities for the involved parties, evaluation environment, evaluation methods and tools, decision criteria for software quality measures and assessment, adopted standards, evaluation activities.

3.3.3 Metrics for Tools

Several metrics are introduced in the book of Stephen H. Kan [Kan04]. They are showed in Table 3.5.
According to the aim of this thesis, the usability and functionality of the methodology and migration tool is mainly concerned. Thus metrics such as MTTF, Defect Density, Customer satisfaction will be considered. The related data such as migration time, the number of errors should be recorded during the migration.

Category	Metrics and Explanation
Mean Time To Failure (MTTF)	How long the software can last before a failure occurs.
Defect Density (opportunity for error)	Lines of Code (LOC),the size of the software. Defect number is also tracked. The current LOC should be so calculated: $$LOC_{current} = LOC_{original} - LOC_{delete} - LOC_{changed} + LOC_{new}$$ Then the LOC defect rate R can be so calculated: $$R_{total} = \frac{Num_{total\_defects}}{LOC_{current}}$$ $$R_{release\_original} = \frac{Num_{release\_original\_defects}}{LOC_{changed}}$$ $$R_{field} = \frac{Num_{field\_defects}}{LOC_{current}}$$ $$R_{release\_original\_field} = \frac{Num_{release\_original\_field\_defects}}{LOC_{changed}}$$ Function point calculation : $$FC = Num_{external\_input} \times 4 + Num_{external\_output} \times 5$$ $$+Num_{internal\_file} \times 10 + Num_{external\_interface} \times 7 + Num_{external\_inquiry} \times 4$$ then 14 characteristics are assigned with weight value w_i from 0 to 5: Data communications, Distributed functions, Performance, Heavily used configuration, Transaction rate, Online data entry, End-user efficiency, Online update, Complex processing, Reusability, Installation ease, Operational ease, Multiple sites, Facilitation of change . $$VAF = 0.65 + 0.01 \times \sum_{i=1}^{14} w_i$$ $$FP = FC \times VAF$$
Customer Problems (not defect is also possible)	$$Problem per User Month(\text{PUM}) = \frac{Num_{problems}}{Num_{lic\_month}}$$ $Num_{problems}$ is the number of problems that customers reported (true defects and non-defect-oriented problems) for a time period. Num_{lic_month} is the number of license-months of the software during the period(license copy number x months).
Customer satisfaction	5 degrees: Very satisfied, Satisfied, Neutral, Dissatisfied, Very dissatisfied.

Table 3.5: Metrics for Tools

4 Specification and Design of the Evaluation

So far the basic knowledge about the migration : fundamental knowledge in Chapter 2 and related evaluation principles in Chapter 3 have been introduced. Based on those knowledge the evaluation can be designed and applied. The evaluation process should consist of several steps. These steps can be mapped to the 7 steps of ITIL CSI processes, which were introduced in Section 3.2. The 7 steps can't be directly used, adaption on the usage environment is also required. Concrete assignment of chapters and sections are shown in Table 4.1:

Steps	Chapter	Section(s)
Define what should be measured.	4	4.1
Define what can be measured.	4	4.2, 4.3
Gather data.	6	6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.1.5, 6.1.6, 6.1.7, 6.1.8, 6.2.1, 6.2.2, 6.2.3, 6.2.4, 6.2.5,
Process data (measurements of metrics).	6	6.1.3, 6.1.6, 6.1.7, 6.2.3, 6.3
Analyze data.	6	6.1.3, 6.1.6, 6.1.7, 6.2.3, 6.3
Present and use the information.	5,6	5.2.3, 5.2.6, 5.2.7, 5.3.3, 6.1.3, 6.1.6, 6.1.7, 6.2.3, 6.3
Implement corrective action.	6	6.1.3, 6.1.6, 6.1.7, 6.2.3

Table 4.1: Research Steps that mapped to ITIL Steps

Then the evaluation can be performed according to steps.

4.1 Evaluation Components

In the first step a migration project should be set up. In this project the methodology and tools of Bachmann and Amazon AWS should be used. Also a migration strategy should be chosen, and the decision about which components of the local system need to be migrated should also be chosen.

During the migration both the DBL and the business logic layer should be migrated, because the aim of the migration process is benefiting from Cloud computing for scientific usage. Scientific usage usually means a large amount of data. Due to the limitation of network, frequently transferring data between local host and the Cloud is not acceptable. In our case the Auditing Application part and Opel part of the SimTech prototype should be migrated to the Cloud.

The ITIL CSI introduces the concept of continual software improvement. In order to improve

the old methodology and tools of migration, problems or errors should be gathered during the migrating process. Thus, analysis and further development can be later applied.

For the migration to the Cloud the AWS services will be used.

4.2 Structure of the Evaluation

Generally there are 2 iterations of evaluation which should be made.

In the first iteration the methodology of Bachmann will be used. The services of AWS will be used. The specific migration tools provided by Amazon will not be used. The installation and configuration of required software will be installed manually. As introduced in Table 2.3, the phases needed is shown as follows:

- Select Migration Scenario

- Describe Desired Cloud Data Hosting Solution

- Select Cloud Data Store or Data Service

- Describe Source Data Store or Data Service

- Identify Patterns to Solve Potential Migration Conflicts

- Refactor Application Architecture

- Migration Data

After these steps are applied, a test phase is required. Through this test phase people can determine whether this migration process is success or not.

The second one uses the methodology of AWS together with the migration tools of Bachmann. Not only the services of AWS are used, but also the tools provided by Amazon are used. Through this methodology the data especially error data about baseline for migration can be collected. The corresponding phases of AWS can be found in Table 2.2. The Leverage the Cloud phase is not critical in our condition, this phase will only be considered when the simulation workflow doesn't work well. And the Optimization phase demands high workload, which is not usually occurs on the prototype, so this phase is also ignored. Thus 4 migration phases will be applied:

- Cloud Assessment Phase

- Proof of Concept Phase

- Data Moving Phase

- Application Moving Phase

A test phase is also required here. When it is needed, additional analysis and conclusion will be added.

Additional analysis and conclusion can also be added in this iteration. The mainly difference for the two iterations is the difference in methodology. The used tools can be the same.

For the concrete implementation of the migration, 2 AWS services should be used: Amazon RDS and Amazon EC2. Base on the idea of Mr. Passow, the general structure of the SimTech prototype in the Cloud is showed in Figure 4.1.

The Amazon EC2 instance that holds Apache ActiveMQ is used for message exchanging. It
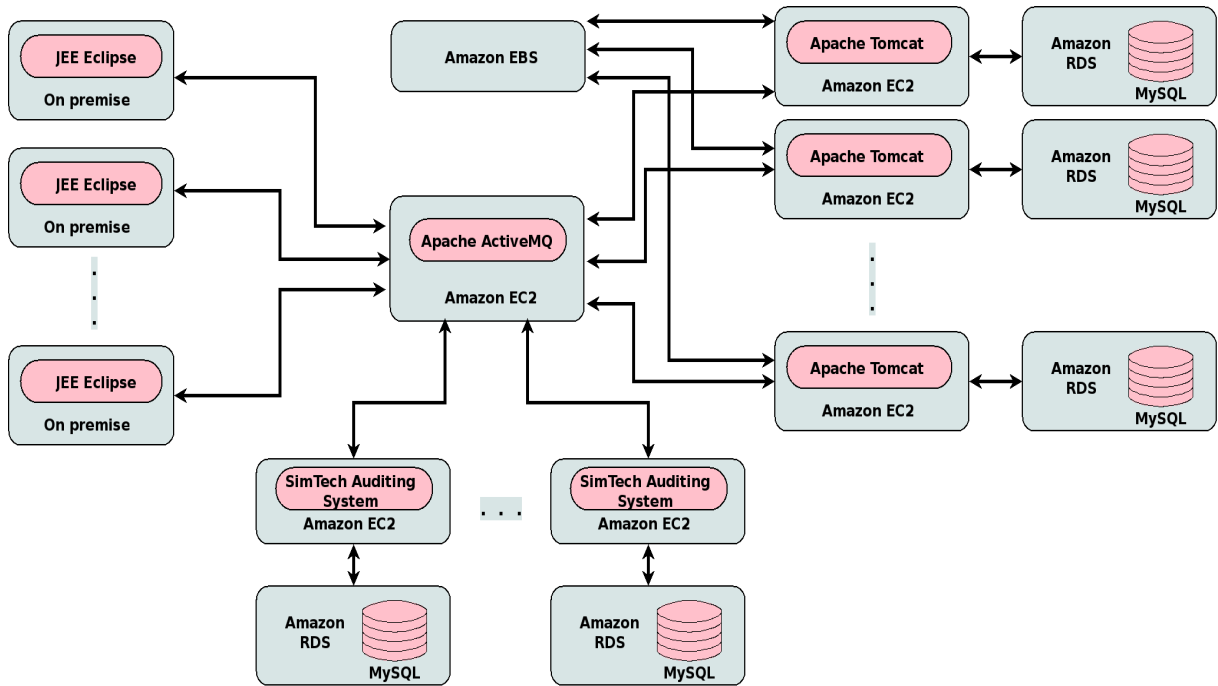
Figure 4.1: SimTech Prototype in Cloud [Pas13]

can connect to multiple Amazon EC2 instances. These instances can install tools like Apache Tomcat, Opal, gfortran and other relevant software. Thus, scientific workflow simulation functionality can be provided. Each of these instances can have an Amazon RDS as its database. The auditing system can also reside on an Amazon EC2 instance. It has its corresponding Amazon RDS database and can connect to the Apache ActiveMQ. Connection from multiple auditing systems are also possible. When the scientific workflow runs, temporary or final data is generated. In order to store these data, the service that provides the functionality of file system should be used. Here the Amazon EBS service is chosen.

4.3 Data Collection Approach

The data collected during the migration phase can be used for the improvement of both migration methodology and migration tools. Which class of data should be collected is important to the evaluation, and depends on the aim of the evaluation. In Section 3.2 and Section 3.3 the data classes in evaluation were introduced. Both the qualitative and quantitative data are collected and used.

In order to record the data, the format of the record should also be considered. In this thesis the format that is shown in Table 4.2 is used.

ID is the unique identifier for an event (an error). It consists of letters and numbers. In this thesis the IDs that start with "A" mean that the errors occur with the AWS methodology with tools of Bachmann, and IDs that start with "B" mean that the errors show up in the Bachmann methodology. For the second letter: "A" means error in migration process for application, "D" means error in migration process for DBL.

Name is the additional information to the ID for better understanding.

Class specifies what kind of problem occurs. As mentioned in Chapter 3, it can be value for process, for tools or something else. For this thesis the software quality metrics which are described in Software product Quality Requirements and Evaluation (SQuaRE) in Section 3.3.2 will be used.

The Severity property has 4 levels: low, middle, high or critical. The level depends on the error impact on the migration result.

The Priority property describes the necessity to fix a problem. It has levels of low, middle, high.

The Reproducible property describes whether this problem can be invoked again under some circumstances. This property can be used for deciding whether it is the fault of the methodology or tools, or the problem is just caused by other reasons.

The Error Handling property describes a way to find a solution to the problem, and the Solution property describes how the problem can be solved. Sometimes further improvement on the tools or methodology might be needed. Extra information about the record can also be added to the Comment property.

As mentioned in Section 3.2 and Section 3.3, qualitative metrics will be mainly considered in this thesis. Metrics such as number of errors, error density (derived from the number of errors), MTTF, error revoke effectiveness, solutions to these errors, time cost, learnability, operability, user error protection, functional completeness, functional correctness should be recorded.

In both iterations that mentioned in Section 4.2, metrics that related to errors (and solutions to the errors) need to be recorded. In the evaluation of methodology and tools of Bachmann, the metrics related to functionality, learnability and operability should also be mentioned. Thus related metrics can be recorded in a table. The structure of the result table is showed in Table 4.3.

Property	Value
ID	BD - 01
Name	Connection fails with RDS data store
Description	AMI instances can not access RDS instances. Connection fails always.
Class	tool (operability)
Severity	low
Priority	high
Reproducible	yes
Error Handling	Configure the AWS security groups.
Solution	Firstly go to the "RDS dashboard, show all instances, click on the corresponding instance and modify its security group. Add this instance to the same security group with its related EC2 instance. Then go to the EC2 dashboard, choose "Security Groups" in category "Networks & Security", click on the corresponding security group, modify the "inbound" property at the bottom of the page, add a new rule to allow outside connection at port 3306 and apply the new rule.
Adaption	No further adaption needed.
Comment	Amazon should put some notice during the creation process of RDS instances.

Table 4.2: Example: Error BD - 01

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes No
Functional correctness	Does everything work correctly? Yes No
Numbers of Error	_____
Total time consumed	_____
Error density	_____
Error protection	Is there mechanism for error protection? Yes No
Error recoverability	Can the process be recovered when encountered an error? Yes No
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: _____
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: _____
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: _____
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: _____

Table 4.3: Metrics Table for each step
.

5 Implementation

In this chapter the two iterations of migration process and problems during the migration process will be described. As discussed in Chapter 4, the 2 iterations represent 2 different migration methodologies: one uses methodology of Bachmann, the other uses methodology of Amazon combined with the migration tools of Bachmann. Both of them are combined with the migration tools of Bachmann. The SimTech prototype will be migrated into the Cloud environment. Both the database layer and the business logic layer will be migrated.

5.1 Structure and Software

The SimTech prototype is assembled with several components. According to the installation instruction of SimTech [HV] and description in Section 2.4 , these components are:

- ActiveMQ 5.6.0

- Apache Axis2 1.5.6

- ODE-PGF 1.3.5

- SimTech Auditing System (can be auditing service or auditing tools)

- Fragmento

- SimTech Test Web Services (MockServices)

- Opal

In order to run the prototype, the following software and files are also required:

- Java Runtime Framework 1.6.38

- Tomcat 7.0.34

- SimTech Eclipse Plug-Ins

- MySQL 5.1.67

- Postgresql 9.2.5

- Extra libraries for Tomcat to support Fragmento

- SoapUI and FragmentoSimTechFragmentsInitialFillingDUNE-soapui-project.xml

- MySQLConnector/J driver (for Opal) 5.1.26

Software and components that are without version in the list above can be obtained directly on the website of SimTech.

The structure of all of the components in Tomcat can be showed as When these software or

```
Tomcat
    ├──bin
    ├──conf
    ├──lib
    │    └──$(Fragmento_libs)
    ├──opal
    ├──resstorage
    └──webapps
            ├──axis2
            │    └──WEB-INF
            │          ├──lib
            │          │    └──$(MySQLConnecter/J)
            │          └──services
            │                └──$(MockServices)
            ├──ode
            ├──Repository
            └──Auditing Service
```
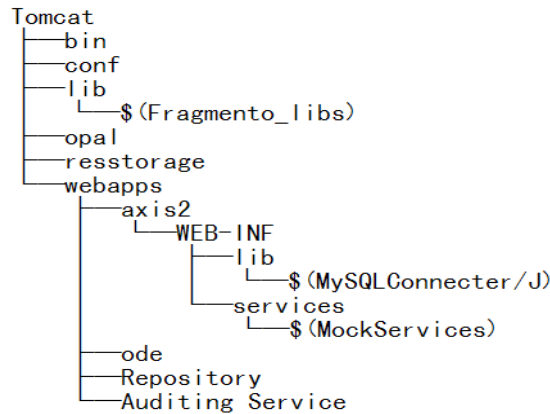
Figure 5.1: Tomcat components

files are not provided by AWS, these software and files have to be uploaded to the AMI and installed. Further configuration should also be done manually.

5.2 Methodology of Bachmann

In the first iteration of evaluation the methodology of Bachmann will be used. The data layer (databases) will be migrated by using the methodology and migration tools of Bachmann. The business logic layer will be migrated manually. Services of AWS will be used.

The result of each step of the methodology will be collected and showed in next chapter.

The methodology of Bachmann consists of 7 steps. The following sections will show the detail steps of this methodology. Because the methodology of Bachmann doesn't refer to the migration of business logic layer, it is not 100% suitable for our situation. A little modification is needed. The main structure of this methodology remains the same, the migration steps of business logic layer are shown together with the steps of methodology of Bachmann.

5.2.1 Select Migration Scenario

This step only works for the migration of data layer. As mentioned in section 2.5 and [SABL13], there are totally 10 migration scenarios for Cloud Data Migration. Because the data should be moved between local host and the Cloud, and the usage of SimTech prototype demands extra resource to meet the requirement of the peak loads. That means the data should be moved back and forth, the resource in the Cloud should be used on demand. According to the definition of the categories of the migration scenarios, the scenario "Cloud Bursting" should be considered.

Several options in tools of Bachmann for this step are shown in Table 5.1:

Migration Strategy	live
Local DBL	two way synchronization
Migration Degree	complete
Source Data Store type	RDBMS
Target Data Store type	RDBMS
Product Version	same product, different version
Direction of Date Movement	from local datacenter to Cloud

Table 5.1: Chosen Options in the Migration Tool (part 1)

Nothing should be done for the business logic layer in this step.

5.2.2 Describe Desired Cloud Data Hosting Solution

For the data layer:
There are totally 13 parts of options to specify the data store in Cloud. The major chosen options are shown in Table 5.2:

Scalability	Vertical(scale up)
Availability	Replication: No (in our case)
Security	Storage Encryption: Yes Transfer Encryption: Yes Firewall: Yes Authentication: Yes Confidentiality: Yes Integrity: Yes Authorization: Yes
Location	single location
Interoperability	Import&Export&Two-Way-Synchronization Data Exchange Format: Proprietary Data Access: SQL&Proprietary
Compatibility	MySQL
Storage	Storage Type: RDBMS
CAP	Consistency Model: strong Availability in case of Partitioning: available
Cloud Computing	Deploy Model: Public Cloud

Table 5.2: Chosen Options in the Migration Tool (part 2)

For the business logic layer:
As described in section 4.2, the original SimTech prototype should be divided into 3 systems, which are the SimTech Auditing System, the ActiveMQ System and the SimTech running

environment. 2 databases should be used: one for the SimTech Auditing System, another is used for SimTech running environment. 1 file system should be used as a temporary partition for storing temporary data. These data can be synchronized by using certain tools between Cloud hosts and local hosts. The corresponding AWS services are shown in Table 5.3:

Components	Cloud Host Solution
SimTech Auditing System	EC2 + AMI + JDK + SimTech Auditing service Size of required software : 140 MB
ActiveMQ System	EC2 + AMI + JDK + ActiveMQ software Size of required software : 117 MB
SimTech running environment	EC2 + AMI + Axis2 + ODE + Fragmento + Mock Service + Opal Size of required software : 541 MB
database for SimTech Auditing System	RDS
database for SimTech running environment	RDS
file system	EBS

Table 5.3: Cloud Host Solution Description

The required software should be uploaded to the AMI images. The installation can be done by following the instructions in the document of SimTech.

5.2.3 Select Cloud Data Store or Data Service

For the data layer:
As we have discussed in last section, the Amazon RDS service will be used. The ideal situation is that the type and version of databases of RDS are identical to that of the local data store. On the local host, the databases that the SimTech prototype uses are MySQL, so the databases used by the RDS should also be MySQL. When using the Amazon RDS service, man should follow the instructions of the Amazon Service: pick "DB instances", then choose "launch instances", then choose a proper relational database, follow the 6 steps of RDS to create the required instance.
For MySQL instances there are no version of 5.1.67. So the nearest version 5.1.69 is chosen.
In the tools of Bachmann in step 3 "Select Cloud Data Store": the "Amazon Relational Database Service (RDS) - MySQL Engine" is chosen.
For the business logic layer:
The properties of the AMI and RDS instances can be seen in Table 5.4:
The required software can be easily transferred to the AMI instances by using remote desktop connection. Listing 5.1 shows an example with rdesktop under Linux environment [Mat]:

```
1  rdesktop -r disk:sharename=/tmp/da_soft_new/ -u administrator -g 1000x700
       ec2-54-194-78-35.eu-west-1.compute.amazonaws.com
```

Listing 5.1: Usage of rdesktop

AMI instance for SimTech ActiveMQ System	Windows 2008 32 bit, 1.7 GB ram, 30 GB storage space
AMI instance for SimTech Auditing System	Windows 2008 32 bit, 1.7 GB ram, 30 GB storage space
AMI instance for SimTech running environment	Windows 2008 32 bit, 3.7 GB ram, 30 GB storage space + 10 GB EBS storage space
RDS instance for SimTech Auditing System	MySQL 5.1.69, 5 GB
RDS instance for SimTech Opal component	MySQL 5.1.69, 5 GB

Table 5.4: Properties of the used AWS Services

With option "-r disk:sharename" we can bind the local directory to the remote server. Then it is easy to copy and paste the required software to the remote server.

Because the RDS instance is not part of the AMI instance, the initialization script "init_-database.sh" for the database of Opal should be modified. An example for the modification is shown in Listing 5.2:

```
1  echo ''Enter MySQL root password:''
2  read password
3  mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < opalmanager_dist_opal_bcclat.sql
4  mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < opalmanager_dist_opal_energies.sql
5  mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < opalmanager_dist_opal_simulations.sql
6  mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < opalmanager_dist_opal_procinstances.sql
7  mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < resmanager_dist_rm_managementcontexts.sql
8  mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < resmanager_dist_rm_servers.sql
9  mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < resmanager_dist_rm_serviceendpoints.sql
10 mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < resmanager_dist_rm_servicetickets.sql
11 mysql -u opalroot -h opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com --
       password=$password < createUsers.sql
12 echo ''Datenbank initalisiert :)''
```

Listing 5.2: Modified init_database.sh

Although there are micro type of AMI instance, this type has too little memory (600 MB). Thus, problem of running ActiveMQ could exist. That is not enough for running services on Tomcat. This problem is showed in Table 6.5. So the larger ones are chosen.

For running the test process of SimTech, the service "OpalSnapProc" is needed. This service should be manually deployed into folder: "%Tomcat%\webapps\ode\WEB-INF\processes\". Otherwise the test process could not run. This problem is shown in Table 6.10.

5.2.4 Describe Source Data Store or Data Service

For the data layer:
The source data stores are two MySQL databases: one is for SimTech Auditing System, the other one is for Opal. Both of them use MySQL with version 5.1.67.
The major chosen options in the tools of Bachmann are shown in Table 5.5:

	Location of the source system	off-premise
Characterization	Cloud Computing deployment model for source system	none
	Cloud Computing service model for source system	none
	Data Store type of source system	RDBMS

Table 5.5: Chosen Options in the Migration Tool (part 4)

The data that should be migrated to the Cloud are separated into 3 parts:

- Data that stored in the MySQL database for SimTech Auditing system. This is the running log of the simulation process.

- Data that stored in the MySQL database for Opal. The amount of this part of data is very small. Information about which simulation result is stored on the disk is provided here.

- Simulation results that stored on the hard drive. More specifically, this part of data is stored in the directory "resstorage", which is sub-directory of Tomcat.

For business logic layer:
The tools of Bachmann doesn't provide a method to migrate data from local disk to remote hard drive. So the synchronization tools should be used. In this thesis the simple synchronization tool "rsync" is chosen. On the AMI instance of SimTech running environment the rsync server should be installed. Two types of rsync server can be used: one is cygwin environment[1] + rsync server, the other one is cwRsync[2] package. In this thesis the cwRsync is chosen, because it is smaller, less than 20 MB. The cygwin environment needs at least 600 MB storage space. The rsync client tool should be installed on the local machine. the cwRsync should be well configured: ports, access privilege, user name, password for the user. An example of the rsync.conf file is showed in Listing 5.3 :

```
1  # uid and gid are required, otherwise user can not log in.
2  uid = 0
3  gid = 0
```

[1]http://www.cygwin.com/
[2]https://www.itefix.no/i2/cwrsync

```
4  # password for the user. Format:  user:password.
5  secrets file = passwd.txt
6  # open port to outside world.
7  port = 8010
8  [resstorage]
9  path = /cygdrive/c/dev/Tomcat/
10 read only = false
11 transfer logging = no
12 auth users = opalroot
```

Listing 5.3: Modified rsync.conf

Listing 5.4 provides an example of rsync_fstab. It enables the rsync service to access the corresponding directory:

```
1  none /cygdrive cygdrive binary,posix=0,user,noacl 0 0
```

Listing 5.4: Modified rsync_fstab

5.2.5 Identify Patterns to Solve Potential Migration Conflicts

For the data layer:
The information of conflicts are automatically generated by the tools of Bachmann. These conflicts should be read and considered. Alternative options in the first 4 steps should also be reconsidered.
For the business logic layer:
Check again whether all of the SimTech components on the local host are identical to that in the Cloud.

5.2.6 Refactor Application Architecture

For both layers:

- Open ports on firewall on the local host and instances in the Cloud.

- Record the IP address or domain names of local host and instances in the Cloud. Make sure that these machines can be connected.

For the data layer:
The SimTech Auditing system and Opal component need to modify its configuration files to connect to the RDS databases. Listing 5.5, Listing 5.6 and Listing 5.7 show examples of the to be modified files:
%Tomcat%\webapps\SimTechAuditing\WEB-INF\spring\database\spring-jms.xml

```
1  <property name=''brokerURL'' value=''tcp://ec2-54-194-27-238.eu-west-1.
       compute.amazonaws.com:61616''/>
```

Listing 5.5: Modified spring-jms.xml

%Tomcat%webapps\axis2\WEB-INF\classes\opalmgr.properties

```
1  db.url=jdbc\:mysql\://opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com\:3306/
       opalmanager_dist
```

Listing 5.6: Modified opalmgr.properties

%Tomcat%webapps\axis2\WEB-INF\classes\resmgr.properties

```
1  db.url=jdbc\:mysql\://opal.cpqczvfuurm2.eu-west-1.rds.amazonaws.com\:3306/
       resmanager_dist
```

Listing 5.7: Modified resmgr.properties

For the business logic layer:
The service endpoints of the test process should be modified to those on the remote server. The SimTech Auditing system and the ODE engine should connect to the ActiveMQ system with the modification on their configuration files.
%Tomcat%\webapps\SimTechAuditing\WEB-INF\spring\database\spring-datasource.xml

```
1  <property name=''url'' value=''jdbc:mysql://auditing.cpqczvfuurm2.eu-west-1.
       rds.amazonaws.com:3306/auditing''/>
```

Listing 5.8: Modified spring-datasource.xml

%Tomcat%\webapps\ode\WEB-INF\conf\ode-axis2.properties

```
1  ode-axis2.activemq.url=tcp://ec2-54-194-27-238.eu-west-1.compute.amazonaws.
       com:61616
```

Listing 5.9: Modified ode-axis2.properties

5.2.7 Migrate Data

For data layer:
Choose the source and target data store, fill in the information about the local data store and Cloud data store. Username, password, host name, port, table name are required. Then

simply click the "Start Migration" button at the bottom of the migration page of the tools of Bachmann. Wait until the record prints the success log.

Two data stores needs to be migrated here.

For business logic layer:

The command in Listing 5.10 can synchronize the data from local host to the instance in the Cloud.

```
1    rsync -avR --progress --delete --password-file=/tmp/passwd * rsync://
        administrator@ec2-54-194-87-181.eu-west-1.compute.amazonaws.com
        :8010/resstorage/
```

Listing 5.10: Rsync usage from local host to remote server

This example command specifies the position of the password, the data to be migrated, the position, port and directory of the remote server.

Command in Listing 5.11 can synchronize the data from the instance in the Cloud to local host:

```
1    rsync -avR --progress --delete --password-file=/tmp/passwd rsync://
        administrator@ec2-54-194-87-181.eu-west-1.compute.amazonaws.com
        :8010/resstorage/ .
```

Listing 5.11: Rsync usage from remote server to local host

By using rsync we can easily migrate the simulation results between to machines.

5.2.8 Test

In this step we should test whether the SimTech prototype in the Cloud works properly. The following issues should be checked:

- Can the BPEL designer on local host connect to the ActiveMQ engine?

- Can the BPEL designer on local host get the information of the Fragmento library?

- Can the BPEL designer on local host connect to the Axis2 service and ODE service?

- Can the BPEL designer on local host get the information of the deployed processes from ODE service?

- Can the BPEL designer on local host connect to the Auditing system?

- Can the BPEL designer on local host get the information of the Opal services?

- Can the test process runs?

- Is there feedback from the server in the Cloud to BPEL designer on local host?

- Is there information of the running status of the test process in the Auditing system?

- Is there information about the finished test process in the database of Opal?

- Does the simulation result resides in the right directory?

5.3 Methodology of Amazon

In the second evaluation iteration the methodology of Amazon will be used. The methodology of Amazon can also be applied on migration of the business logic layer, further modification on the methodology is not needed. The database layer will also be migrated by using the migration tools of Bachmann. The business logic layer will be migrated by using the tools provided by Amazon when the corresponding tools exist. If there's no tools of Amazon for migrating certain component of the business logic layer, the component will be migrated manually.

According to the description in Section 4.2, 4 steps are required.

5.3.1 Cloud Assessment Phase

As we have discussed in Chapter 3, the metrics about functionality and user interaction will be considered. Other aspects like financial metrics, security metrics will not be included. This means that the assessment about cost, security are less important part of this step. Technical assessment is important here. For the license assessment, the license of the used software are showed in Table 5.6. The architecture of the prototype should be assessed. The related

License	Software
Apache license[a]	ActiveMQ, Axis2, ODE, Fragmento, Tomcat
BSD license[b]	Opal
Oracle Technology Network Developer License terms[c]	JDK
GPL[d]	MySQL, MySQLConnector/J driver,
PostgreSQL License[e]	PostgreSQL

[a]http://www.apache.org/licenses/LICENSE-2.0
[b]http://opensource.org/licenses/BSD-3-Clause
[c]http://www.oracle.com/technetwork/licenses/standard-license-152015.html
[d]http://www.gnu.org/licenses/
[e]http://www.postgresql.org/about/licence/

Table 5.6: Software License

software should also be considered whether these software are compatible to the Cloud environment, which type of modification should be applied.

For the preparation of the migration, related software should be prepared, in case these are not provided by Amazon. For a concrete migration plan the following steps are listed:

- Prepare the software of JDK, Apache ActiveMQ, Apache Tomcat, Apache ODE, Axis2, eclipse, Fragmento library, SoapUI, Opal, SimTech Auditing System.

- Design the target architecture, which is already showed in Section 4.2.

- Create instances by using AWS.

  – Create an EC2 instance and choose an AMI for SimTech Auditing System.

  – Create an EC2 instance and choose an AMI for Apache ActiveMQ.

  – Create an EC2 instance and choose an AMI for SimTech running environment.

  – Create an RDS instance for SimTech Auditing System.

  – Create an RDS instance for SimTech running environment.

  – Create an EBS instance for simulation result.

  – Do the configuration of the databases. Make sure that they are of compatible versions.

  – If possible, verify whether the SimTech prototype is runnable on one of the EC2 instance (e.g. on the EC2 instance for SimTech running environment).

- Construct the prototype.

  – Verify which software can be installed by using the tools provided by Amazon, which are not.

  – Install the software that can be provided by Amazon.

  – Install the rest software manually.

  – Check whether each component works.

  – Reconfig the databases to guarantee that they can work together with the SimTech prototype on the EC2 instances.

  – If further adjustments are required, alter the corresponding parts.

    * The configuration of IP address in eclipse should be modified.

    * The configuration of IP address of SimTech component in eclipse should be modified.

- Validate that this SimTech prototype works.

The related result can be seen in Section 6.2.1.

5.3.2 Proof of Concept Phase

In this step a pilot should be built and tested. The EC2 instances and RDS instances should be prepared. Specifications of these instances can be seen in Table 5.4. All of the required software and components should also be prepared. Specifications of these software and components can be seen in Table 5.3. The upload job will be done in step 4 of this iteration. The concrete sequence of the tasks can follow the plan in step 1 of this iteration. The evaluation result can be seen in Section 6.2.2.

5.3.3 Data Migration Phase

In this phase the content of the local data stores and the simulation result should be migrated. As we have mentioned in Section 5.2.4, 3 parts of data need to be migrated: the content of databases for SimTech Auditing system, for Opal system, and the simulation result on file system. The migration job was done by using the tools of Bachmann and the rsync program. The evaluation result can be seen in Section 6.2.3.

5.3.4 Application Migration Phase

In this step, all of the required software and components for the SimTech prototype should be uploaded to the server at first.
Then these software and components should be installed and configured. Things that need to be modified are ode-axis2.properties, opalmgr.properties, resmgr.properties, spring-datasource.xml, spring-jms.xml, rsyncd_conf, rsync_fstab, connection properties in eclipse for Fragmento, ODE and SimTech. The related result can be seen in Section 6.2.4.

5.3.5 Test

When the migration finished, we need to check whether the SimTech prototype works properly. The to be checked issues have already been listed in Section 5.2.8.

6 Evaluation

In this chapter all the data that were collected during the migration steps will be showed and analyzed. There are three main sections in this chapter. The first one will show the evaluation data of the migration steps of methodology Bachmann, the second section will present the evaluation data of the methodology Amazon. The third one will do some comparison between the two and make a preliminary analysis.

The evaluation environment: The AWS services locate in Ireland. 3 different virtual machine instances locate on local machine. For the internet connection: eduroam is used. The migrating speed depends on the speed of eduroam.

For the time evaluation: the time that is spent on problem solving will not be considered as the required time for each methodology. For each iteration only the time that is spent on tools of Bachmann and on the migration of business logic layer is calculated. The time that is spent on problem solving will also be recorded, but will not be calculated as part of the total time for each iteration.

6.1 Methodology of Bachmann

In the first evaluation iteration the methodology of Bachmann will be used.
The related data was collected and represented by using metrics which were discussed in Chapter 3.

6.1.1 Select Migration Scenario

In this thesis only the scenario "Cloud Bursting" will be considered. If users are not familiar with the related migration scenarios and patterns, at least 1 hour is needed to prepare for the basic knowledge. In our case this prerequisite is already fulfilled.
No problem occurred. Time spent in this step:

- For data layer: 4 minutes.

- For business logic layer: 0 minutes.

People need some time to learn the differences between the different scenarios. They are not hard to learn. The tools of Bachmann are easy to operate, the information that was displayed by the tools is also easy to understand. We were satisfied with this migration step.

The evaluation result is showed in Table 6.1.

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	0
Total time consumed	4 minutes
Error density	0
Error protection	Is there mechanism for error protection? Yes
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 2
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 1
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 2
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 1

Table 6.1: Metrics Table for step 1 of methodology of Bachmann
.

6.1.2 Describe Desired Cloud Data Hosting Solution

In this step users need to go through with the information about the corresponding Cloud Service. In our case the information about Amazon EC2, RDS and EBS and related services is required. With this information users can choose the appropriate services.

The separation of functionality of the business logic layer was also proceeded in this step. The components of SimTech are categorized by its functionality into 3 main groups. The size of each group was also calculated.

No problem occurred in this step. Time spent in this step:

- For data layer: 8 minutes.

- For business logic layer: 5 minutes.

For learnability: a lot of things about the services in the Cloud should be learned. People also need to know the meanings of the options in the tools of Bachmann. Although these are not hard to understand, but time need to be spent. The tools of Bachmann were not hard to operate in our case. There were several hints about conflicts in this step, they were not so easy to understand and the conflicts were not easy to eliminate. Thus, the satisfaction degree went down.

The evaluation result is showed in Table 6.2.

6.1.3 Select Cloud Data Store or Data Service

For the data layer:

No problem occurred by using the tools of Bachmann.

When the instance of RDS was created, the following connection problem that is showed in Table 6.3 existed.

A problem about generating new account in RDS also existed, which is showed in Table 6.4:

For the business logic layer:

The required EC2 instances should be created.

When the required software/components were installed, some problems occurred.

The first one was with the ActiveMQ engine, which is showed in Table 6.5:

Then several problems occurred with the SimTech prototype. These problems are showed in Table 6.6,Table 6.7,Table 6.8,Table 6.9,Table 6.10:

Total time spent on step 3:

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	0
Total time consumed	13 minutes
Error density	0
Error protection	Is there mechanism for error protection? Yes
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 3
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 2
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 3
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 3

Table 6.2: Metrics Table for step 2 of methodology of Bachmann
.

Property	Value
ID	BD - 01
Name	Connection failed
Description	AMI instances could not access RDS instances. Connection failed always.
Class	Tool (operability)
Severity	Low
Priority	High
Reproducible	Yes
Error Handling	Configure the AWS security groups. [Ama]
Solution	Firstly go to the "RDS dashboard, show all instances, click on the corresponding instance and modify its security group. Add this instance to the same security group with its related EC2 instance. Then go to the EC2 dashboard, choose "Security Groups" in category "Networks & Security", click on the corresponding security group, modify the "inbound" property at the bottom of the page, add a new rule to allow outside connection at port 3306 and apply the new rule.
Adaption	No further adaption needed.
Comment	Amazon should put some notice during the creation process of RDS instances.

Table 6.3: Error BD - 01

Property	Value
ID	BD - 02
Name	Privilege granting error
Description	The user "test" in RDS instance for SimTech Auditing system could not be granted with "All" privilege.
Class	Tool (operability)
Severity	Low
Priority	Low
Reproducible	Yes
Error Handling	Ignore this problem
Solution	No further operation needed. A normal user can also operate the database "auditing". This database can be initialized by the account in the administrative group of MySQL in this RDS instance.
Adaption	No further adaption needed.
Comment	

Table 6.4: Error BD - 02

Property	Value
ID	BA - 01
Name	ActiveMQ didn't run.
Description	ActiveMQ didn't run because the RAM was not enough.
Class	Tool (operability)
Severity	Low
Priority	High
Reproducible	Yes
Error Handling	Change to another EC2 AMI instance which has larger RAM.
Solution	Create another EC2 AMI instance that has 1.7 GB RAM. Delete the old one.
Adaption	No further adaption needed.
Comment	

Table 6.5: Error BA - 01

Property	Value
ID	BA - 02
Name	Out of memory error for JVM
Description	When we started Tomcat in SimTech running environment, it got error message as "JVM:MaxPermSize error, out of memory".
Class	Tool (operability)
Severity	Middle
Priority	High
Reproducible	Yes
Error Handling	Open Tomcat configuration tool and add "-XX:MaxPermSize=512m" at the end of the parameters of Tomcat.
Solution	Sometimes Tomcat doesn't read the options from the system environment variables. So the environment variable "JAVA_OPTS" sometimes fails to work. These parameter should be directly assigned to Tomcat.
Adaption	No further adaption needed.
Comment	

Table 6.6: Error BA - 02

Property	Value
ID	BA - 03
Name	Eclipse hanged
Description	Multiple entries in the database of Fragmento made eclipse crash. The eclipse hanged, and the Tomcat server crashed.
Class	Tool (operability)
Severity	Middle
Priority	High
Reproducible	Yes
Error Handling	Cascading delete all entries in database "repository" and re-initialize this database.
Solution	Open pgadmin III to edit the Fragmento database "repository": go through "server", "PostgreSQL", "Databases", "repository", "Schemas", "public" to "Tables". Choose all the tables on the right side, right click and choose the command "truncate cascaded". After that we can initialize this database again by using SoapUI.
Adaption	No further adaption needed.
Comment	This problem is still not recorded in the document of SimTech. Perhaps the developer may add this part to the document.

Table 6.7: Error BA - 03

Property	Value
ID	BA - 04
Name	Test process stops at step "CreateNewMgmtCtx".
Description	The test process didn't even run. It stopped always at the same phase.
Class	Tool (operability)
Severity	Low
Priority	High
Reproducible	Yes
Error Handling	Select another context name.
Solution	For each simulation a different context name is needed. All of them should not be the same. The context name is used for distinguishing one simulation instance from another.
Adaption	No further adaption needed.
Comment	A introduction for the test process is needed.

Table 6.8: Error BA - 04

Property	Value
ID	BA - 05
Name	Test process stopped at step "CreateNewMgmtCtx".
Description	The test process didn't run, although we had solved the last problem.
Class	Tool (operability)
Severity	Low
Priority	High
Reproducible	Yes
Error Handling	Reconnect to the server and activate the service.
Solution	In eclipse, choose "Window", "Open Perspective", "SimTech BPEL". Then follow "Window", "Show View", "SimTech Resource Manager Views", choose "Server Management". At the bottom of the eclipse, choose the server and edit its property, activate its service. Then in the view of "Context Management" we can get the content of all contexts.
Adaption	No further adaption needed.
Comment	

Table 6.9: Error BA - 05

Property	Value
ID	BA - 06
Name	Test process stopped at step "PickAcquireOpalMCServiceCallback".
Description	The test process didn't run.
Class	Tool (operability)
Severity	Middle
Priority	High
Reproducible	Yes
Error Handling	Copy process "OpalSnapProc" to ODE directory.
Solution	Copy "OpalSnapProc" from the test process project to the following directory: %Tomcat%\webapps\ode\WEB-INF\processes .
Adaption	No further adaption needed.
Comment	This instruction should be precisely described in the document of the test process.

Table 6.10: Error BA - 06

- For data layer: 2 minutes for operation with tools of Bachmann. 20 minutes for RDS creation. 0.5 day for problem solving.

- For business logic layer: 1 hour for AMI creation. 35 minutes for uploading the required software and components. 1 hour for installation of these software and components. A duration of 14 days were spent for problem solving, because the developers of SimTech needed time to modify the corresponding components.

A lot of things about SimTech prototype and services of Amazon should be learned and reviewed. The tools of Bachmann were not hard to operate. Several errors occurred in this step. The way to solve the errors were also not hard to operate, to the contrary, finding solutions to the errors were not so easy, because we could not infer the concrete reason from the error message. Sometimes the information in the document of SimTech is not enough to solve the problem, the instructions from the developers of SimTech were clearer and much more understandable. So the degree of understandability went down to "not so easy to understand". Due to the errors, we could not grant this step with a higher satisfaction degree. The evaluation result is showed in Table 6.11:

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	8
Total time consumed (without problem solving)	3 hours
Error density	2.67 error/hour
Error protection	Is there mechanism for error protection? Yes
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 3
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 3
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 4
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 3

Table 6.11: Metrics Table for step 3 of methodology of Bachmann
.

6.1.4 Describe Source Data Store or Data Service

In this step the specifications of the data layer and business logic layer of the source data stores and source system were determined.

In the tools of Bachmann we only need to choose which type is the source data store. Thus it took only 1 minute.

For the migration we have generated the following data by using the test process of SimTech. The data generation took 5 hours. This time will not be part of the migration time consumption.

- Data that stored in MySQL database in SimTech Auditing system: all of these are running status of the test process. The amount of this part of data is 250 MB. This part was migrated by using the tools of Bachmann.

- Data that stored in MySQL database of Opal: the information about the context names are stored here. The amount of this part of data is very small. It is less than 1 MB. Only content of table "resmanager_dist.rm_managementcontexts" are modified. This part was migrated by using the tools of Bachmann.

- The simulation data of the test process is stored in %Tomcat%\resstorage with the context name as folder name. The amount of simulation data is 1.1 GB. This part was migrated by using rsync.

For the business logic layer: The configuration of the cwRsync server should be done on the AMI instance.

No problem occurred in this step.

Total time spent on step 4:

- For data layer: 1 minute for operation with tools of Bachmann.

- For business logic layer: 1 minute to install the cwRsync server. 3 minutes to modify rsyncd_conf, rsync_fstab and passwd.txt and open port on the system firewall. The way to modify these files has been discussed in Section 5.2.4.

Things in this step were easy to learn and to operate. The only extra effort was given to the cwRsync software. The information was also understandable. We granted degree 2 of satisfaction to this step.

The evaluation result is showed in Table 6.12:

6.1.5 Identify Patterns to Solve Potential Migration Conflicts

The tools of Bachmann has generated several warnings when the first 4 steps have been finished. The main content about the warnings was about the differences of version and security settings between local data store and those of RDS. For business logic layer: all of the software and components are of the identical version.

Total time spent in this step:

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	0
Total time consumed	5 minutes
Error density	0
Error protection	Is there mechanism for error protection? Yes
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 2
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 1
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 1
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 2

Table 6.12: Metrics Table for step 4 of methodology of Bachmann
.

- For data layer: 1 minute for operation with tools of Bachmann.

- For business logic layer: 1 minute to recheck the components of the SimTech prototype.

All things in this step were easy to learn and to operate. The conflicts messages were hard to understand. But we were still satisfied with this step.
The evaluation result is showed in Table 6.13:

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	0
Total time consumed	2 minutes
Error density	0
Error protection	Is there mechanism for error protection? Yes
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 1
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 1
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 4
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 1

Table 6.13: Metrics Table for step 5 of methodology of Bachmann
.

6.1.6 Refactor Application Architecture

Problems occurred in this step were mainly on the business logic layer. They are showed in Table 6.14 and Table 6.15:

Total time spent in this step:

- For data layer: 1 minute for operation with tools of Bachmann. 5 minutes for modification of the configuration files of Auditing system and Opal system.

- For business logic layer: 3 minutes for modification of the configuration files of Auditing system and Opal system. 2 minutes to open all ports that were needed. 14 days for problem solving.

Property	Value
ID	BA - 07
Name	No feedback from the ODE engine.
Description	Although all of the configuration files were correctly configured, the eclipse BPEL designer still got nothing.
Class	Tool (operability)
Severity	High
Priority	High
Reproducible	Yes
Error Handling	Update the ODE component.
Solution	The ODE components that are created after 03/12/2013 don't have this problem. This is variable assignment error on message queue.
Adaption	No further adaption needed.
Comment	

Table 6.14: Error BA - 07

Property	Value
ID	BA - 08
Name	Eclipse got no services information.
Description	Eclipse could not connect to the remote SimTech prototype.
Class	Tool (operability)
Severity	High
Priority	High
Reproducible	Yes
Error Handling	Edit the configuration of the components of eclipse.
Solution	The following IP addresses should be modified: • Fragmento component of eclipse. • Server View at the bottom of SimTech-BPEL Perspective. • "Window"to"Preferences"to"BPEL"to"ODE" • "Window"to"Preferences"to"BPEL"to"SimTech"
Adaption	No further adaption needed.
Comment	

Table 6.15: Error BA - 08

The modification of the configuration files are not easy to learn and sometimes hard to locate. One of the errors went with the component itself. So the modification of the source code was required. Only with the help of the developers of SimTech some of these errors were solved at the end. Without the help of the developers this step was not operable. The error message was also not understandable for us. We were not satisfied with this step.

The evaluation result is showed in Table 6.16:

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	2
Total time consumed	11 minutes
Error density	10.9 errors/hour
Error protection	Is there mechanism for error protection? Yes
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 5
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 4
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 4
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 4

Table 6.16: Metrics Table for step 6 of methodology of Bachmann
.

6.1.7 Migrate Data

For the data of SimTech Auditing system: Filling the information of local data store and RDS took 1 minute. The tools of Bachmann took 17 minutes to export the content of local data store and import it to the RDS.

For the data of Opal system: Filling the information of local data store and RDS also took 1 minute. The migration process consumed less than 1 minute.

For the data of the simulation result: Moving 1.1 GB data from local file system to remote file system took about 47 minutes.

Problems that occurred during the migration are showed in Table 6.17 and Table 6.18:

Property	Value
ID	BD - 03
Name	Connection failed on local data store.
Description	The tools of Bachmann could not access to the local data store with the given administrative account's name and password.
Class	Privilege (operability)
Severity	Low
Priority	High
Reproducible	Yes
Error Handling	Enable to access the local data store with the given account remotely.
Solution	Enter the MySQL workbench, change root accessing host from "localhost" to "%" [Ora].
Adaption	No further adaption needed.
Comment	

Table 6.17: Error BD - 03

One problem occurred after the migration, which is showed in Table 6.19:
Total time spent in this step:

- For data layer: 2 minutes for operation with tools of Bachmann. 18 minutes for migrating the content of databases. 47 minutes for synchronizing the files. 2 days for problem solving.

- For business logic layer: 0 minutes needed.

It was not hard to learn the required knowledge for this step. Operating was also without difficulty. The only thing was with the errors in this step. Some of the error messages were not easy to understand. The middle level the satisfaction degree was granted to this step. The evaluation result is showed in Table 6.20:

6.1.8 Test

This section shows whether the migrated SimTech prototype can work properly.
The list to be checked has already been shown in Section 5.2.8. The results of these questions are shown in the following:

- Can the BPEL designer on local host connect to the ActiveMQ engine?
  Yes. It can successfully connect to the port 61616 of the ActiveMQ engine on the remote server. This step took less than 1 minute.

- Can the BPEL designer on local host get the information of the Fragmento library?
  Yes. After modifying the IP address of the Fragmento repository, the BPEL designer

Property	Value
ID	BD - 04
Name	Writing failed on remote server.
Description	When we tried to synchronize the file from local host to the remote server, the rsync program could not write files into the aim directory.
Class	Tool (operability)
Severity	Middle
Priority	High
Reproducible	Yes
Error Handling	Assign the Amazon EBS with a drive letter.
Solution	At first, this EBS should be mounted as directory "resstorage" in the directory of Tomcat. But rsync can not write a linked directory. Thus, the Amazon EBS should be assigned with a drive letter. The path of the new drive should be added to rsync_fstab.
Adaption	The content of rsync_fstab should be modified.
Comment	This problem should be solved by the developers of rsync/cwRsync.

Table 6.18: Error BD - 04

Property	Value
ID	BD - 05
Name	Tomcat didn't response for certain requests.
Description	Eclipse got no information about services and contexts on remote server.
Class	Tool (operability)
Severity	Low
Priority	Low
Reproducible	Sometimes
Error Handling	Restart Tomcat on remote server.
Solution	Sometimes this happens. The simplest way to solve this problem is restarting Tomcat.
Adaption	No further adaption needed.
Comment	The SimTech prototype is still not stable.

Table 6.19: Error BD - 05

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	3
Total time consumed	67 minutes
Error density	2.7 errors/hour
Error protection	Is there mechanism for error protection? Yes
Error recoverability	Can the process be recovered when encountered an error? No
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 2
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 1
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 3
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 3

Table 6.20: Metrics Table for step 7 of methodology of Bachmann
.

can connect to the Fragmento service on remote server successfully. We don't need to modify the IP address again when we restart the eclipse BPEL designer. This step needs about 1 minute, which depends on the speed of the internet connection.

- Can the BPEL designer on local host connect to the Axis2 service and ODE service?
  Yes. The only thing need to do is modifying the IP addresses of these services in the BPEL designer. This can be seen directly at the bottom of the BPEL designer.

- Can the BPEL designer on local host get the information of the deployed processes from ODE service?
  Yes. The default situation is that there is only one process running on the remote server: the "OpalSnapProc" process.

- Can the BPEL designer on local host connect to the Auditing system?
  Yes.

- Can the BPEL designer on local host get the information of the Opal services?
  Yes. This can be seen directly at the bottom of the BPEL designer.

- Can the test process runs?
  Yes. Five iterations of the test process takes about two minutes to run.

- Is there feedback from the server in the Cloud to BPEL designer on local host?
  Yes. This problem has been solved during the migrating procedure.

- Is there information of the running status of the test process in the Auditing system?
  Yes. When the simulation began, the running status was shown in the window of the Auditing system in eclipse BPEL designer.

- Is there information about the finished test process in the database of Opal?
  Yes. When the test process has successfully finished, the dialog window of the database should be refreshed. Then the information of the newest running result will be there.

- Does the simulation result resides in the right directory?
  Yes. All of the generated results are in the directory "resstorage". 1 minute was needed to log in into the remote server.

The test phase didn't take much time. All steps were done in less than 10 minutes.

6.2 Methodology of Amazon

In the second evaluation iteration the methodology of Amazon will be used. According to the description in Section 4.2, 4 steps are required. The result of each step will be showed in the following subsections.

6.2.1 Cloud Assessment Phase

We didn't have problem in this phase.
Time spent in this step:

- Assessment about software license: 15 minutes.

- Assessment about architecture of SimTech prototype: 10 minutes.

- Preparation of the related software: 5 minutes.

The detailed requirements of this step need to be learned. It was not so easy. But we had no difficulties with the operating. The information of this step were understandable. The satisfaction degree was also good.

The evaluation result is showed in Table 6.21.

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	0
Total time consumed	30 minutes
Error density	0
Error protection	Is there mechanism for error protection? No
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 3
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 1
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 2
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 2

Table 6.21: Metrics Table for step 1 of methodology of Amazon
.

6.2.2 Proof of Concept Phase

3 AMI instances of Windows 8, 2 RDS instances with MySQL and 1 EBS instance were created. Time spent in this step:

- Creation of AMI instances: 1 hour.

- Creation of RDS instances: 20 minutes.

- Creation of EBS instance: 2 minutes.

These instances should be in one security group. The usually used ports were also opened. The modification on the security group took 5 minutes.
No error occurred in this step.
Similar to the situation in the first iteration, the usage of Amazon services were not simple. Document of these services should be read. So the degree "2" was assigned to the operability, understandability and satisfaction.
The evaluation result is showed in Table 6.22.

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	0
Total time consumed	87 minutes
Error density	0
Error protection	Is there mechanism for error protection? No
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 3
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 2
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 2
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 2

Table 6.22: Metrics Table for step 2 of methodology of Amazon
.

6.2.3 Data Migration Phase

As mentioned in Section 6.1.4, the amount of data are following:

- 250 MB data in MySQL for SimTech Auditing system.

- 1 MB data in MySQL for Opal system.

- 1.1 GB simulation result.

The first 2 were migrated by using the tools of Bachmann, the last one was migrated by using rsync.

Errors occurred when we migrated the data in MySQL. These errors are showed in Table 6.23 and Table 6.24:

Property	Value
ID	AD - 01
Name	Table already existed.
Description	There were already databases and tables with the same name in the RDS instances.
Class	Tool (operability)
Severity	Low
Priority	Low
Reproducible	Yes
Error Handling	Drop the existing databases.
Solution	Log in the databases in RDS and drop the databases or tables.
Adaption	No
Comment	

Table 6.23: Error AD - 01

Property	Value
ID	AD - 02
Name	Connection lost
Description	The wireless connection to eduroam was lost.
Class	Tool (operability)
Severity	Low
Priority	Low
Reproducible	Yes
Error Handling	Reconnect to eduroam.
Solution	Choose "eduroam" in the wireless connection setting and type the password.
Adaption	No
Comment	

Table 6.24: Error AD - 02

Time spent in this step:

- For configuring the tools of Bachmann: 20 minutes.

- For data migration of SimTech Auditing system: 18 minutes.

- For data migration of Opal system: 1 minute.

- For migration of simulation result: 48 minutes.

- For error solving: half a day.

Considering the errors we have encountered in the first iteration, the migration tools of Bachmann and the usage of rsync were not so easy, so the middle level was granted to learnability, operability and understandability. Two new errors occurred in this iteration, so the satisfaction degree became lower.

The evaluation result is showed in Table 6.25.

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	2
Total time consumed	87 minutes
Error density	1.38 errors/hour
Error protection	Is there mechanism for error protection? Yes
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 3
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 3
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 3
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 4

Table 6.25: Metrics Table for step 3 of methodology of Amazon
.

6.2.4 Application Migration Phase

Time spent in this step:

- For uploading software and components of SimTech prototype into the AMI instances: 35 minutes.

- For installation of these software: 1 hour.

- For configuration of the services: 12 minutes.

No problem happened in this step.

Considering the errors we have met in the first iteration, the migration of business logic layer was very hard to learn and to operate. Only with the help of the developers of SimTech the problems were finally solved. So "4" was granted to learnability, operability and understandability, and "5" was given to the satisfaction degree.

The evaluation result is showed in Table 6.26.

Metrics	Details
Functional completeness	Does it in the end have the required functionality? Yes
Functional correctness	Does everything work correctly? Yes
Numbers of Error	0
Total time consumed	107 minutes
Error density	0
Error protection	Is there mechanism for error protection? No
Error recoverability	Can the process be recovered when encountered an error? Yes
Learnability	How easily this part of methodology or tools can be learned? 1: easiest, 5: hardest: 4
Operability	Can this part of methodology or tools be easily operated? 1: easiest, 5: hardest: 4
Understandability	Can the information of this part of methodology or tools be easily understood? 1: easiest, 5: hardest: 4
Satisfaction degree	If this part of methodology or tools is used by a customer, how satisfied he/she can be? 1: most satisfied, 5: least satisfied: 5

Table 6.26: Metrics Table for step 4 of methodology of Amazon
.

6.2.5 Test

The Simtech prototype can run properly on the remote server. The result is shown in following:

- Can the BPEL designer on local host connect to the ActiveMQ engine?
  Yes. When the BPEL designer reconnected to the ActiveMQ engine, it got successful result.

- Can the BPEL designer on local host get the information of the Fragmento library?
  Yes. Nothing unusual happened.

- Can the BPEL designer on local host connect to the Axis2 service and ODE service?
  Yes.

- Can the BPEL designer on local host get the information of the deployed processes from ODE service?
  Yes. The situation is the same as that in the first iteration.

- Can the BPEL designer on local host connect to the Auditing system?
  Yes.

- Can the BPEL designer on local host get the information of the Opal services?
  Yes.

- Can the test process runs?
  Yes.

- Is there feedback from the server in the Cloud to BPEL designer on local host?
  Yes.

- Is there information of the running status of the test process in the Auditing system?
  Yes.

- Is there information about the finished test process in the database of Opal?
  Yes.

- Does the simulation result resides in the right directory?
  Yes.

As in the first iteration, the test phase for the second iteration took also less than 10 minutes to finish all the checking tasks.

6.3 Comparison and Analysis

In this section the metrics of both methodologies are compared and analyzed.
The basic information of the comparison is in Table 6.27:
We can use the result of the methodology of Bachmann as the baseline for the comparison.
Both of the methodologies fulfill the requirement of functional completeness and functional correctness.
Five errors that related to the data layer and eight errors that with the business logic layer have occurred during the first migration iteration. Two errors that related to the data layer have occurred during the second migration iteration. The methodology of Bachmann has more errors (13 : 2) , and a much higher error density (2.67 : 0.37). The reason is that we

| Metrics	Bachmann	Amazon
Functional completeness	Yes	Yes
Functional correctness	Yes	Yes
Total Numbers of Error (data layer/business logic layer)	5/8	2/0
Total time consumed	292 minutes	321 minutes
Total Error density	2.67 errors/hour	0.37 errors/hour
Average Learnability (How easily this part of methodology or tools can be learned?) 1: easiest, 5: hardest.	18/7=2.57	13/4=3.25
Average Operability (Can this part of methodology or tools be easily operated?) 1: easiest, 5: hardest.	13/7=1.86	10/4=2.5
Average Understandability (Can the information of this part of methodology or tools be easily understood?) 1: easiest, 5: hardest.	21/7=3	11/4=2.75
Average Satisfaction degree (If this part of methodology or tools is used by a customer, how satisfied he/she can be?) 1: most satisfied, 5: least satisfied.	17/7=2.43	13/4=3.25

Table 6.27: Basic Comparison

have done this iteration first. If we did the iteration of methodology of Amazon first, the same errors would come up with the methodology of Amazon. These errors are methodology irrelevant.

Among these errors: 2 are connection problems, 2 are privilege problems, 1 is hardware inefficiency problem, 2 are configuration problems, 7 are problems with SimTech prototype. These are drawn in Figure 6.1:

The time consumption on the test phase were less than 10 minutes, which is 3.4% of the time consumption for the first iteration, and 3.1% of the time consumption for the second iteration. The time consumption of methodology of Bachmann is smaller (292 minutes : 321 minutes). The methodology of Amazon combined with the tools of Bachmann took about 10% more time to do the same job. This is because the first step of methodology of Amazon has the tasks of assessment, which the methodology of Bachmann doesn't have. The three main time consumption parts of the migration are: time for using the tools of Bachmann (about 20 minutes), time for migrating the data layer (more than 1 hour), time for migrating the business logic layer(more than 1 hour). The tasks are the same, the migration of business logic layer was done manually, the data layer was done by using the tools of Bachmann, so there's no big differences in time consumption between the 2 methodologies.

The time consumption of each step in the first iteration was not even. Step 3 and step 7 took more than 80% of the time consumption. To the contrary, tasks in the second iteration were assigned more reasonable. The pie charts about time consumption of the 2 migrating iterations are shown in Figure 6.2 and Figure 6.3:
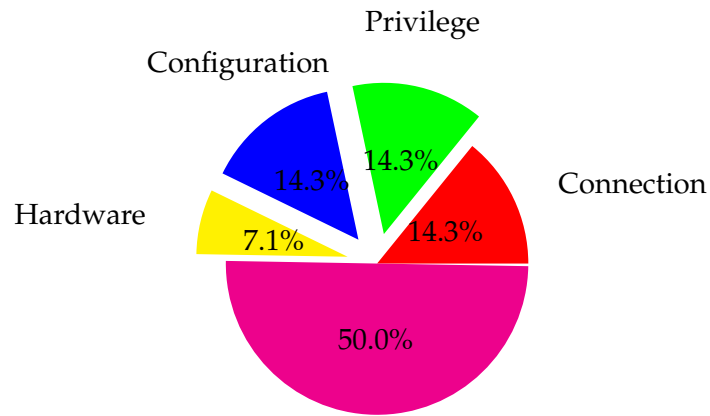
Problems with SimTech prototype
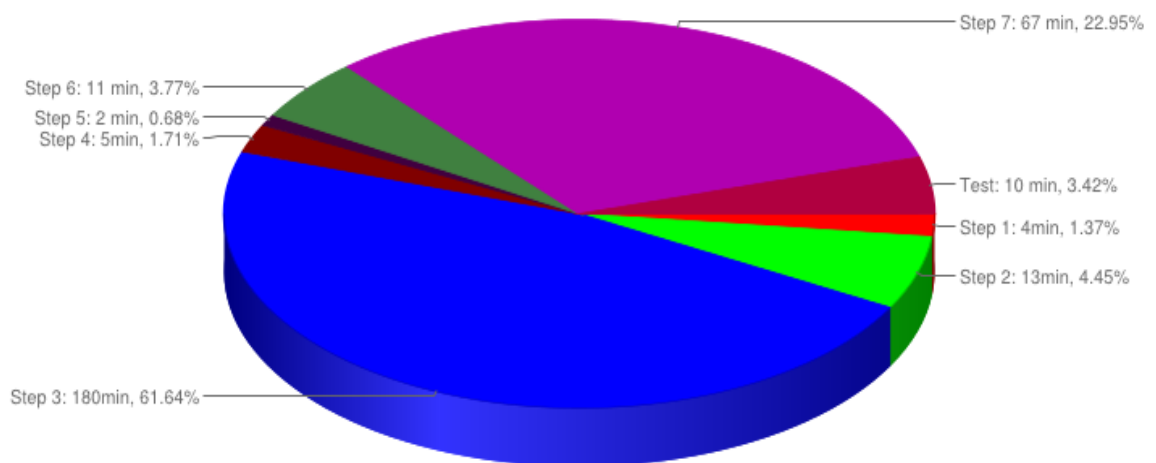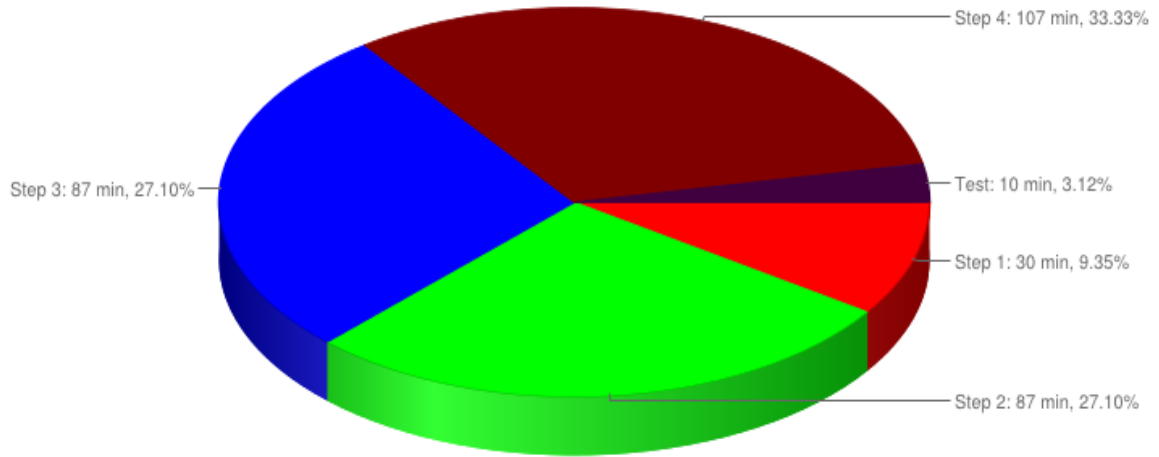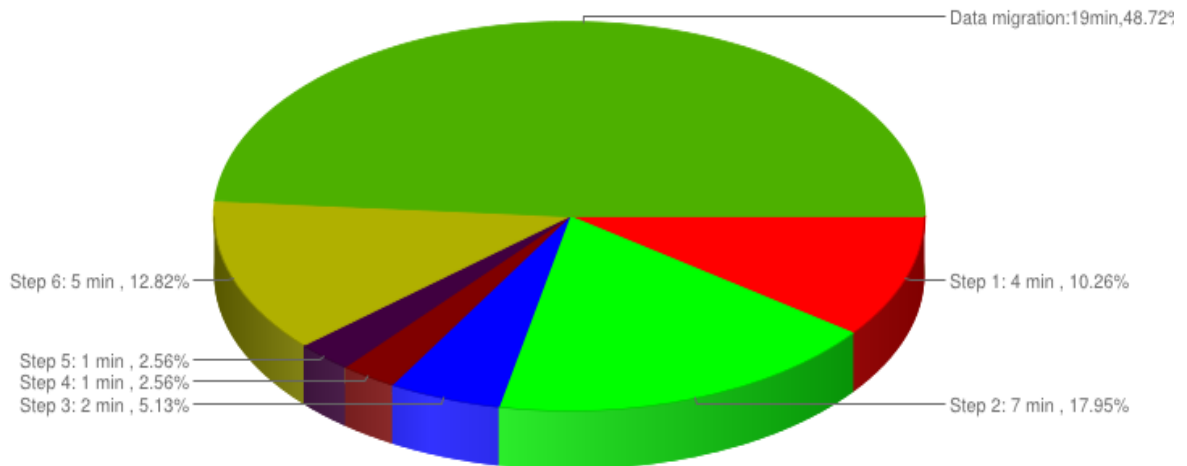
Figure 6.1: Pie chart of errors

Figure 6.2: Pie chart of time consumption of the first iteration

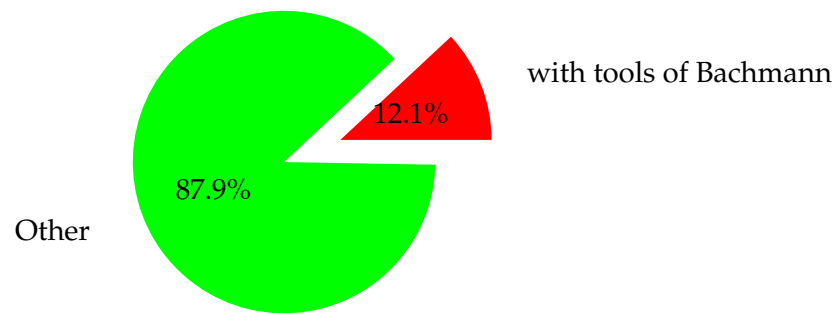**Figure 6.3:** Pie chart of time consumption of the second iteration

The time spent purely with the tools of Bachmann in the second iteration (using methodology of Amazon) is shown in Figure 6.4:

Figure 6.4: Pie chart for time consumption of tools of Bachmann in the second iteration

In the second iteration we have spent 20 minutes to configure the tools of Bachmann and 19 minutes for the migration. That's 39 minutes in the end. The proportion of the time consumption with the tools of Bachmann is shown in Figure 6.5:

For the 4 subjective aspects: The methodology of Bachmann has better average learnability and operability, and higher satisfaction degree. The methodology of Amazon is more understandable.

Figure 6.5: Proportion of time consumption of tools of Bachmann in the second iteration

7 Conclusion and Future Work

In this topic we have evaluated the migration methodology and tools of Bachmann (the business logic layer was manually migrated.) and the methodology of Amazon combined with the tools of Bachmann. We have firstly introduced the basic knowledge about Cloud Computing and migration scenarios and patterns. Then the metrics for evaluating data and processes were introduced. After that the fundamental design of migrated SimTech prototype was presented in Chapter 4. When all the preparation work has been done, the migration tasks began. In the first migration iteration the methodology and tools of Bachmann was used. In the second iteration the methodology of Amazon combined with the tools of Bachmann was used. The tasks were detailed specified in each step of these two methodologies. Both of the data layer and the business logic layer were migrated. The functional metrics such as functionality correctness and time consumption were measured. Errors during the migration were recorded and analysed.

For further improvement:

- We have divided the SimTech prototype into 3 parts. 1 SimTech Auditing system, 1 SimTech running environment and 1 ActiveMQ system can now successfully work together. The original idea was connecting multiple SimTech Auditing systems, multiple SimTech running environment with only one ActiveMQ system. This situation was not tested.

- The SimTech prototype is still not stable. For example, the ODE engine requires a restart of Tomcat when a process is undeployed. The is the job of the developers of SimTech.

- Only EC2 and RDS services were used. Other services from Amazon or services from other providers are not tested.

- When we migrate the business logic layer, the following things can be done:

  - The modification on the configuration files of the components can be done with batch file or shell script.

  - The uploading process can also be done automatically.

  - The required software and components can be packed up into one self-extracting package.

  Effort can be saved by using these methods.

  Under windows it is not easy to install the required software automatically, because interaction is required during the installation and preparation. But under Linux this job is easier to accomplish.

JDK, Tomcat, rsync server, fortran and database server can be installed by using the package manager. The EBS can be formatted and mounted by using commands "fdisk/g-parted" and "mount". The components of SimTech can be simply deployed to the webapps directory of Tomcat. The modifications on configuration files can be done through tools such as "sed" or "awk".

One problem still needs to be solved: the initialization of Fragmento database. Exporting the content from the PostgreSQL database on a windows system and importing it to the Linux server could be a solution. The modification on eclipse should be done manually.

Bibliography

[Ama]   Amazon Web Services, Inc.   Amazon Elastic Compute Cloud User Guide. `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html`.

[Bac12]   T. Bachmann. Entwicklung einer Methodik für die Migration der Datenbankschicht in die Cloud. Diplomarbeit, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Germany, September 2012.

[CS11]   G. Case and G. Spalding. *ITIL Version 3 - Continual Service Improvement*. TSO, The Stationery Office, 2011.

[DR52]   W. Deming and N. K. G. Renmei. *Elementary Principles of the Statistical Control of Quality: A Series of Lectures*. Nippon Kagaku Gijutsu Remmei, 1952.

[Gal04]   D. Galin. *Software Quality Assurance: From Theory to Implementation*. Alternative Etext Formats. Pearson/Addison Wesley, 2004.

[Gec]   B. Geck. *Übersicht der Prozessmodelle CMMI, SPICE, ITIL*. gebeCom GmbH.

[GW99]   J. R. Groff and P. N. Weinberg. *SQL, the Complete Reference*. Osborne, MacGraw-Hill, 1999.

[Hah]   M. Hahn. *Entwicklerhandbuch Simtech-Prototyp*. `http://kaaroda.informatik.uni-stuttgart.de:88/svn/simtech/documents/Entwicklerhandbuch/Entwicklerhandbuch.pdf`.

[HV]   M. Hahn and K. Vukojevic. *Installationsanleitung: Simtech-Prototyp*. `http://kaaroda.informatik.uni-stuttgart.de:88/svn/simtech/documents/Handbuch/InstallationsanleitungSimTechPrototyp.pdf`.

[HW03]   G. Hohpe and B. Woolf. *Migrating to the Cloud: Oracle Legacy Client/Server Applications*. Addison-Wesley Professional, 2003.

[ISO05]   FCD 25000 - Software Engineering – Software product quality requirements and evaluation (SQuaRE) - Guide to SQuaRE, 2005.

[ISO06]   FDIS 25030 - Software Engineering – Software quality requirements and evaluation (SQuaRE) - Quality requirements, 2006.

[ISO10]   FDIS 25040 - Systems and software engineering – Systems and Software quality requirements and evaluation (SQuaRE) - Evaluation process, 2010.

[ISO12]   Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Quality Measure Elements, 2012.

[ITI]   ITIL Official Website. `http://www.itil-officialsite.com/`.

[Jin10]   Jinesh Varia. Migrating Your Existing Application to the AWS Cloud. A Phase-Driven Approach to Cloud Migration, 2010. `http://media.amazonwebservices.com/CloudMigration-main.pdf`.

[Kan04]   S. Kan. *Metrics and Models in Software Quality Engineering*. Addison-Wesley, 2nd edition, 2004.

[KPP95]   B. Kitchenham, L. Pickard, and S. L. Pfleeger. Case studies for method and tool evaluation. *Software, IEEE*, 12(4):52–62, 1995.

[Mat]   Matt Chapman. Man pages of rdesktop. `http://linux.die.net/man/1/rdesktop`.

[MG11]   P. Mell and T. Grance. National Institute of Standards and Technology. The NIST Definition of Cloud Computing, 2011. `http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf`.

[Ora]   Oracle, Inc. MySQL Workbench Manual. `http://dev.mysql.com/doc/workbench/en/wb-mysql-connections-navigator-management-users-and-privileges.html`.

[Pas13]   S. Passow. Evaluation of a Methodology for Migration of the Database Layer to the Cloud based on a Research Case Study. Studienarbeit, April 2013.

[SABL13] S. Strauch, V. Andrikopoulos, T. Bachmann, and F. Leymann. Migrating Application Data to the Cloud Using Cloud Data Patterns. In *Proceedings of the 3rd International Conference on Cloud Computing and Service Science (CLOSER'13)*, pages 0–11. SciTePress, Mai 2013.

[SF12]   P. Sadalage and M. Fowler. *NoSQL Distilled, A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley, 2012.

[Sto04]   R. Stockmann. Was ist eine gute Evaluation? *CEVAL, Arbeitspapier*, 2004.

[TPS96]   E. Taylor-Powell and S. Steele. Collecting Evaluation Data: An Overview of Sources and Methods. *University of Wisconsin Cooperative Extension Service. Cir. G3658-4*, 1996.

All links were last followed on December 18, 2013.

Acknowledgement

I am heartily thankful to my supervisor Steve Strauch from the University of Stuttgart for his encouragement, guidance and support in all the phases of this diploma thesis. I am also grateful to Mr. Michael Hahn for his advices and useful tips. Special thanks to my family and friends for their moral support.

Xiaolei Guo

Declaration

I hereby declare that the work presented in this thesis is entirely my own. I did not use any sources and references other than those listed. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

Stuttgart, 20 December 2013   _____

(Xiaolei Guo)