MASTER

Requirements engineering in a global enterprise
how can we reuse our knowledge?

Waterman, B.B.

Award date:
2007

Link to publication
Can we reuse our knowledge?

Report

Benno Waterman - June 2007
TBM
Requirements Engineering

in a Global Enterprise:

How can we reuse our knowledge?

A Design Oriented Approach

B. B. Waterman
Requirements Engineering
in a Global Enterprise:
How can we reuse our knowledge?

A Design-Oriented Approach

Master's Thesis

B. B. Waterman
Student ID: 0494077

Eindhoven University of Technology
Department of Technology Management
Program in Industrial Engineering and Management Science

Supervisors:
Dr. Ir. S.A. Angelov
Prof. Dr. R. Kusters

ABN AMRO Global IT Services
IT Strategy & Architecture
IT Strategy

Supervisors:
B. Schreiner
Ir. J. Ploum
If a problem has no solution, it may not be a problem, but a fact - not to be solved, but to be coped with over time - Shimon Peres.
Abstract

This master thesis describes the development of a model how reuse of non-functional requirements can be embedded in the current Requirements Specification process of the ABN AMRO bank. A conceptual has been constructed based on three best practices models from literature. Subsequently, the conceptual model is adjusted to the real-life situation of the ABN AMRO Bank and is partially implemented by the means of a NFR checklist.

*Keywords:* non-functional requirement, software reuse.
Preface

This report is my master thesis for the study of Industrial Engineering and Management Science at the faculty of Technology Management at Eindhoven University of Technology. The report contains a model how reuse of Non-Functional Requirements can be embedded in the Requirements Specification process for ABN AMRO's Business Units (BUs). I would like to thank several people that contributed to this thesis.

First of all I would like to acknowledge my supervisors from the subdepartment Information Systems at Eindhoven University of Technology. Their comments helped me highly to improve the quality of my thesis. I would like to express my gratitude to Samuil Angelov for providing guidance throughout my project and for being always around for supervising me. In addition, I also wish to thank Rob Kusters for this critical questions that kept me on the 'academical track' by letting me realize the importance of the 'why-question'.

During my time at ABN AMRO, I gained experience in what challenges it takes to start IT initiatives throughout the bank and how it is to work in such a large company like ABN AMRO. The rumors about a potential takeover by Barclays or by the consortium of Royal Bank of Scotland have intensified this experience increasingly. I would like to thank the employees from the Global IT Strategy & Architecture team who facilitated a nice graduation period for me: Thomas, Franciska, Ben, Ronald and Inigo, thanks for the great time. Especially, I would like to thank Jos Ploum for his mentoring role during my internship and for the assistance whenever I needed it. I would also like to thank Peter Penders for his valuable input due his widespread knowledge about Requirements Engineering related affairs at ABN AMRO. Finally I would like to thank Ben Schreiner for being a part of the Global IT Strategy team during my graduation period.

My special thanks go to Krista Valk, for her extensive feedback on my reports and for patiently and cheerfully encouraging me when things didn't work out as intended. I want to thank Thijs Grünhagen for his effort on screening my report on English and Rutger Stolker and Maarten Wismans for providing feedback on the content of my report. Further, I want to thank Michiel de Nijs for his graphical contribution. My thanks go to my parents for their unconditional support, and to all my friends, especially the Commons of the fraternity WASSADAMO, who were able letting me forget my graduation related affairs in a trice.

I have had beautiful years at the TU/e. These years were not only intellectually challenging, but first and foremost contributed to my personal development. I am grateful to everyone who has been part of that.

Le'chaim.

Benno Waterman

Utrecht, the Netherlands
June 2007
Executive Summary

The ABN AMRO Bank (AAB) is a global bank that ranks eight in Europe, covering five regional Client, two global Client, three product and two support Business Units (BUs). One of this support BUs is BU Services that supports the core banking functions along with striving for cost savings. The MSc. Graduation project, which is reported in this thesis, has been conducted in the Global Services IT department. This department looks for IT solutions that cover the bank as a whole.

Motivation
Being competitive as a global organization has recently influenced the traditional way of banking. In order to stay competitive, banks have to focus on their core business by outsourcing activities that are not supporting this core business functions. The AAB started the Harvest & Symphony offshore outsourcing initiative to gain IT synergies. This initiative means for software development projects that the actual building and testing of IT applications is executed on a different location by IT vendors that are assigned to a specific BU. However, the intended cost savings of the Harvest & Symphony initiative are not automatically reflected in the efficient use of IT in the bank. This was often caused by the fact that IT vendors did not develop IT application according to the requirements that were set up by the retained IT organization of the AAB. The problem encountered is encapsulated in the problem statement below.

In the new IT environment resulting from the Harvest and Symphony initiative, it is increasingly important to define requirements in a proper way. ABN AMRO Global IT Services wants to improve the way the specification of requirements is performed at this moment.

Objective
Regarding the current situation of the AAB, it can be concluded that BUs have little coherence between each other in developing software projects. Furthermore, there is a variation in maturity level between the different BUs regarding the Requirements Engineering domain. However, a precondition for setting the research objective is that this objective covers a problem that is present for the bank as a whole. A cross-case analysis, executed at the end of four exploratory case studies, revealed the aspect of NFR reuse. According to Kotonya (1998), NFRs include standards, regulations, and contracts to which the product must conform. These can be descriptions of external interfaces, performance requirements, design and implementation constraints and quality attributes. The research objective of this thesis is given below.

The objective of this research is to improve the way NFRs are specified at the BUs of the AAB, by reusing existing NFRs. An important part of this objective is the elaboration of a model how reuse of NFRs can be embedded in the current lifecycle for requirements specification and how this model can be operationalized for ABN AMRO.
Research
An analysis has been performed, in order to construct a model how reuse of NFRs can be embedded in the current Requirement Specification process. In short, three main findings result from the analysis.

1. NFRs differ highly in character and are dispersedly located in the organization.

2. At this moment, there is no formal process at the AAB during the specification of requirements in which reuse of NFRs is actually taken place.

3. High-quality NFRs have to be specified in specific, measurable and unambiguous terms that meet the user expectation fully.

These three main findings form the design requirements for developing the conceptual model. However, it has been concluded that no model could be derived from literature that meets all the criteria for fulfilling the objective of this research. As a consequence, the three design requirements form three complementary models, which are based on literature and serve as building blocks. Finally these building blocks are merged into the conceptual model.

Subsequently, this constructed conceptual model is fitted to a real-life situation by adjusting the three aforementioned building blocks to the AAB. Further, the conceptual model is partially implemented by gradually introducing the reusable repository by the means of a NFR Comprehensive Checklist. Eventually a case study example has been described to test the applicability of this NFR checklist.
Main conclusions
The main conclusions of the research project are given next.

- Two different stages for setting NFRs at the AAB can be identified: the setting of high-level NFRs and the setting of detailed NFRs.

- The realization of a goal-oriented approach, proposed by Chung et al. (2000), for managing conflicting NFRs at the AAB is extremely difficult to establish at this moment. This approach can be realized when a higher maturity level regarding NFR reuse is established.

- The creation of high-quality NFRs by decomposing high-level NFRs into detailed NFRs is not explicitly taking place at the AAB. Although NFRs are prioritized, there is no standard way in what way NFR decomposition takes place.

- The creation of a reusable repository is the crucial element for operationalizing the conceptual model at the AAB.

- This repository can be updated in two ways for the AAB. By abstracting NFRs from Requirements Specification documents and by adding external imposed NFRs to the repository.

- A NFRs checklist, by being a ‘repository on paper’, is the best possible way for enabling NFRs reuse due to its negligible developing cost, its relative low resistance and its high functionality.

Recommendations
Several recommendations that address ABN AMRO Global Services IT (and also other possible practitioners) are given. The most important recommendations resulting from this research project are given below.

- Define NFRs as early as possible in software development projects and make as much as NFRs explicit as possible.

- Use a standard and structured way for defining high-level business NFRs in and between BUs and use a standard and structured way for decomposing high-level NFRs into detailed NFRs.

- Realize the long-term effect of using standardized NFRs. NFR reuse can enable on the longer term reuse on system level and NFR reuse affects other facets in the organization.

- Appoint people with high mandate to initiate the embedding of NFR reuse in the Requirements Specification process.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iv</td>
</tr>
<tr>
<td>Preface</td>
<td>vi</td>
</tr>
<tr>
<td>Executive Summary</td>
<td>vii</td>
</tr>
<tr>
<td>Contents</td>
<td>x</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xiv</td>
</tr>
<tr>
<td>Abbreviations</td>
<td>xvi</td>
</tr>
<tr>
<td>1 Context of the Study</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Company description ABN AMRO Group</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Global Services IT</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Harvest and Symphony</td>
<td>3</td>
</tr>
<tr>
<td>1.4 ABN AMRO's efficiency in respect of IT</td>
<td>4</td>
</tr>
<tr>
<td>2 Research Problem</td>
<td>5</td>
</tr>
<tr>
<td>2.1 The Type of Research</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Initial Problem Statement</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Problem Analysis</td>
<td>7</td>
</tr>
<tr>
<td>2.4 Exploratory Problem Analysis by the use of literature</td>
<td>7</td>
</tr>
<tr>
<td>2.5 Exploratory Problem Analysis by the use of case studies</td>
<td>8</td>
</tr>
<tr>
<td>2.5.1 Case Study selection</td>
<td>9</td>
</tr>
<tr>
<td>2.5.2 Case Study Results</td>
<td>9</td>
</tr>
<tr>
<td>2.5.3 Cross-case Analysis</td>
<td>10</td>
</tr>
<tr>
<td>2.6 Research Problem Definition</td>
<td>10</td>
</tr>
<tr>
<td>3 Research Methodology</td>
<td>13</td>
</tr>
<tr>
<td>3.1 Exploratory Research phase</td>
<td>13</td>
</tr>
<tr>
<td>3.1.1 Exploratory Problem Analysis by the use of literature</td>
<td>14</td>
</tr>
<tr>
<td>3.1.2 Exploratory Problem Analysis by the use of case studies</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Confirmatory Research Phase</td>
<td>14</td>
</tr>
</tbody>
</table>
3.3 Exploratory and Confirmatory Data Collection Methods ..................................14
3.3.1 Desk Research .............................................................................................15
3.3.2 Physical Artifacts ..........................................................................................15
3.3.3 Participant Observations .............................................................................16
3.3.4 Interviewing .................................................................................................16
3.4 Research Model .................................................................................................16
3.5 Research Questions ............................................................................................17
3.6 Outline of the report ..........................................................................................18

4 Deriving a Model from Literature .......................................................................19
4.1 Identifying design criteria ..................................................................................19
4.2 Selecting a method .............................................................................................21
4.3 Selecting a Knowledge Management Model ....................................................22
4.3.1 What is knowledge? .......................................................................................22
4.3.2 Design Requirements towards a Knowledge Management Model ..............23
4.3.3 Selecting a Knowledge Management Model ................................................23
4.3.4 Reuse during Requirements Specification ...................................................24
4.3.5 Conclusions selected Knowledge Management Model ................................26
4.4 Selecting a Requirements Specification Process for NFRs .................................26
4.4.1 Design Requirements towards a Specification Process for NFRs .................26
4.4.2 Selecting a Specification process for NFRs ....................................................27
4.4.3 Representing NFRs in a structured approach ...............................................28
4.4.4 Conclusions NFRs during Requirements Specification .................................28
4.5 Selecting a Quality Model .................................................................................29
4.5.1 Design Requirements towards a Quality Model ............................................29
4.5.2 Selecting a Quality Model .............................................................................29
4.5.3 SQFD Framework .........................................................................................30
4.5.4 Conclusions Selected Quality Model ............................................................31
4.6 Constructing the Conceptual Model .................................................................31
4.7 Conclusions .......................................................................................................36

5 Suiiting for Practical Application .........................................................................37
5.1 Adjusting the conceptual model to ABN AMRO ..............................................37
5.1.1 Selecting an approach ....................................................................................38
5.1.2 Graphical Mapping .......................................................................................38
5.2 Realizing the designed model at ABN AMRO ..................................................42
5.2.1 Selecting an approach ....................................................................................42
5.2.2 Development of a Comprehensive Checklist ...............................................44
5.2.3 Logical Data Model .......................................................................................45
5.2.4 Responsible stakeholders ............................................................................45
5.2.5 Resistance Analysis .......................................................................................45
5.2.6 Guideline for implementation: project plan ...............................................46
5.2.7 Costs ..............................................................................................................47
5.3 Evaluating the approach for realizing the conceptual model .............................48
5.3.1 Selecting an approach ....................................................................................48
5.3.2 Case Description ...........................................................................................49
5.3.3 Approach .......................................................................................................49
5.3.4 Results ..........................................................................................................50
5.3.5 Financial Results ..........................................................................................51
5.4 Conclusions .......................................................................................................52

6 Conclusions and Recommendations .....................................................................53
6.1 Conclusions .......................................................................................................53
6.2 Recommendations .............................................................................................55
List of Figures

Figure 1.1 Organizational Structure ABN AMRO Group (ABN AMRO Group, 2007).........1
Figure 1.2 Organizational Overview ABN AMRO Services IT (adapted from ABN AMRO
Group (2007))....................................................................................................................2
Figure 1.3: IT Efficiency Performance ABN AMRO Group (adapted from ABN AMRO
Group (2007))....................................................................................................................4

Figure 2.1 Regulative Cycle (Van Strien, 1975)........................................................................5
Figure 2.2 Case study approach (Eisenhardt, 1989)...................................................................8
Figure 2.3 Different Software Development Methodologies ...................................................10

Figure 3.1 Research methodology approach (Onwuegbuzie et al., 2005).................................13
Figure 3.2 Exploratory and Confirmatory Research Data Collection Methods.......................15
Figure 3.3 Research Model......................................................................................................17

Figure 4.1 Methodology for constructing the Conceptual Model ............................................21
Figure 4.2 Explicit and Tacit Knowledge (Nonaka et al., 1995).............................................22
Figure 4.3 The knowledge value chain (Weggeman, 1997).......................................................24
Figure 4.4 Software Reuse lifecycle (Zhang et al., 2001).........................................................25
Figure 4.5 five step Requirements Specification Model (Bamford et al., 2004)..........................27
Figure 4.6 Sort Hierarchy NFRs (Chung et al., 2000).................................................................28
Figure 4.7 Basic SQFD model (Haag et al., 1996)....................................................................30
Figure 4.8 Building Block 1.....................................................................................................32
Figure 4.9 Building Block 2.....................................................................................................33
Figure 4.10 Building Block 3...................................................................................................34
Figure 4.11 Constructed conceptual model for NFR reuse in the Requirements Specification
process ..................................................................................................................................35

Figure 5.1 Successive steps for implementation ......................................................................37
Figure 5.2 Iterative NFR Specification process ........................................................................39
Figure 5.3 Proposal for implementing the conceptual model in the current Requirements
Specification Process ........................................................................................................41
Figure 5.4 plan for implementing the comprehensive checklist............................................47
Figure 5.5 Interface Internet Banking Application ..................................................................49
List of Tables

Table 1.1 Overview of assigned Harvest vendors to BUs..............................3

Table 3.1 Research Questions classified per Research Phase of Van Strien (1975) ........17

Table 4.1 Survey of existing methods for reusing NFRs during Requirements Specification. 21
Table 4.2 Survey of existing models for selecting a knowledge management model........23
Table 4.3 Survey of existing models for selecting a specification model for NFRs ........27
Table 4.4 Survey of existing models for selecting a quality model.......................30

Table 5.1 Survey for selecting approaches for reaching a successful fit..................38
Table 5.2 Survey of approaches for partial realization of the conceptual model ..........43
Table 5.3 Survey of existing approaches for evaluating implementation approaches.....48
Abbreviations

AAB | ABN AMRO Bank
AD | Application Development
AIM | ABN AMRO Instruction Manual
AM | Application Maintenance
ATR | ABN AMRO Time Registration
BAD | Business Architecture Definition
BIB | Best Internet Bank
BU | Business Unit
BUAM | Business Unit Asset Management
BULA | Business Unit Latin America
BUNA | Business Unit North America
BUNL | Business Unit Netherlands
BUPC | Business Unit Private Clients
BUTB | Business Unit Transaction Banking
CARE | Computer Aided Requirements Engineering
DSDM | Dynamic System Development Method
EPIC | Enterprise Portfolio Information Center
FMI | Functional Model Iteration
FR | Functional Requirement
FTE | Full Time Equivalent
GSD | Global Software Development
IB | Internet Banking
IS | Information Systems
IT | Information Technology
KIO | Knowledge Intensive Organization
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>Methodology for ABN AMRO Projects</td>
</tr>
<tr>
<td>MT</td>
<td>Management Team</td>
</tr>
<tr>
<td>NFR</td>
<td>Non-Functional Requirement</td>
</tr>
<tr>
<td>PDD</td>
<td>Project Definition Document</td>
</tr>
<tr>
<td>PDLC</td>
<td>Project Development LifeCycle</td>
</tr>
<tr>
<td>PMO</td>
<td>Project Management Office</td>
</tr>
<tr>
<td>PRL</td>
<td>Prioritized Requirement List</td>
</tr>
<tr>
<td>RAD</td>
<td>Requirement, Analysis and High-Level Design procedure</td>
</tr>
<tr>
<td>RDD</td>
<td>Requirement Definition Document</td>
</tr>
<tr>
<td>RE</td>
<td>Requirements Engineering</td>
</tr>
<tr>
<td>REM</td>
<td>Requirement Engineering and Management</td>
</tr>
<tr>
<td>RMP</td>
<td>Requirements Management Plan</td>
</tr>
<tr>
<td>SAD</td>
<td>System Architecture Definition</td>
</tr>
<tr>
<td>SBT</td>
<td>Systeem Basis Toegangsregeling</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SME</td>
<td>Subject Matter Expert</td>
</tr>
<tr>
<td>SOXA</td>
<td>Sarbanes-OXley Act</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirements Specification</td>
</tr>
<tr>
<td>TCS</td>
<td>Tata Consultancy Services</td>
</tr>
<tr>
<td>TNS</td>
<td>Telecommunication Network Services</td>
</tr>
</tbody>
</table>
1 Context of the Study

1.1 Company description ABN AMRO Group

The ABN AMRO Bank (AAB) is an international bank, where the history is going back to 1824. The AAB ranks eighth in Europe and 13th in the world based on total assets, with more than 4,500 branches in 53 countries, a staff of over 110,000 full-time equivalents (FTEs) and total assets of EUR 999 billion (as at 30 September 2006).

The AAB implements its strategy through a number of Business Units (BUs). These units are responsible for managing a distinct region, client segment or product segment, while also sharing expertise and operational excellence across the Group. This is shown graphically in Figure 1.1:

![Organizational Structure ABN AMRO Group](image)

Figure 1.1 Organizational Structure ABN AMRO Group (ABN AMRO Group, 2007)

The AAB has five **regional Client BUs**: the Netherlands, Europe, North America, Latin America and Asia. These BUs serve about 20 million consumer clients and small to larger businesses worldwide. ABN AMRO is among the world’s leading players in these businesses.

The AAB has two **global Client BUs** to serve clients with global needs. The BU Private Clients provides private banking services to wealthy individuals and families and has EUR 136 billion in Assets under Administration (as at 30 September 2006). The BU Global Clients serves 550 multinational clients.

The AAB has **three Product BUs**:

- **Global Markets** develops products for the AAB’s commercial clients across the globe.
- **Transaction Banking** is AAB’s product organization covering all payments and trade in the bank for the retail, private client, and commercial markets.
- **Asset Management**, which is one of the world's leading asset managers, operates from over 20 locations worldwide and manages EUR 192 billion worth (as at 30 September 2006) of assets for private investors and institutional clients.

**Services**

Services was established to create cost savings through consolidation and standardization. It
focuses on further exploiting new market solutions for support services with the aim to achieve better products and services for AAB’s clients at lower costs.

**Group Functions**
Group Functions collaborates with the BUs in maximizing client and shareholder value. Its basic functions are governance (facilitating the implementation of Managing Board policy throughout the bank), standard and policy setting (setting the parameters that the BUs work within), and sharing expertise across the company.

**Consumer Client Segments**
In order to provide all the clients with even better products and services, the AAB also has a cross-BU Consumer Client Segment and a cross-BU Commercial Client Segment. These segments focus on aligning the Client BUs with the Product BUs, sharing best practices and exchanging formulas across the Group in order to deliver high-quality solutions to AAB’s client bases across the world.

### 1.2 Global Services IT

BU Services, described in the previous section, has different sub departments. One particular sub department is Services IT, covering the total IT operations of the bank. Services IT consists of approximately 2,500 employees, who serve over 60 countries, and services globally every business function in the bank. Services IT can best be considered as a matrix organization, which is depicted in Figure 1.2:

![Figure 1.2 Organizational Overview ABN AMRO Services IT (adapted from ABN AMRO Group (2007))](image)

The blue circle in Figure 1.2 highlights special departments inside Services IT: Global Services IT. Global Services IT operates within two organizational structures: regional and functional. By combining their geographic expertise together with group-wide functional expertise, Global Service IT is able to offer high-quality IT services and to generate cost savings through
economies of scale and by utilizing group synergies. In fact, Global Services IT enables the AAB to share best practices regarding IT across the group.

This research is conducted at the department Global IT Strategy & Architecture. This department aims, as part of Global Services IT, to look for IT solutions to cover the bank as a whole, instead of focusing on one particular BU. As a consequence, this research study considers the problem by looking for a solution for the bank as a whole.

1.3 Harvest and Symphony

The global strategy of the AAB is to become a top 5 European-based-bank. In order to reach this position, the AAB constantly has to strive for growing revenues and continuously has to reduce its costs. The AAB choose to attain cost reduction regarding its IT operations by starting the Harvest & Symphony program. In fact, on March 2005 the Managing Board of the AAB decided to start with the Harvest & Symphony initiative by proceeding with an IT service delivery that encompassed in-house consolidation, partial outsourcing, and multi-vendor offshore outsourcing. This meant that the majority of IT activities had to be transferred to third party service providers. Four major programmes were established involving five different BUs (i.e. Netherlands, Asset Management, Private Clients, North America and Latin America) to investigate the options for improving the group-wide service delivery models for IT Infrastructure (Symphony), Application Support and Maintenance (Harvest), and Fast-tracks including Telecommunications Network Services (TNS) and Infrastructure and Data Services. As a result, the first two programmes were bundled under the name: Harvest & Symphony.

Finally in November 2005, Global service agreements were signed with ‘best of breed’ vendors valuing approximately EUR 1.8 billion for a five-year period, covering the following vendors, which are described below. IBM was selected as IT Infrastructure (Symphony) vendor. For Application Support and Maintenance (Harvest), two different types of vendors could be distinguished: the Application Maintenance (AM) vendor and the Application Development (AD) vendor. AM vendors cover the maintenance towards software applications and can be considered as the “main vendor” for that particular BU. The other side, AD vendors cover the development of software applications and are not BU specific. However, many BUs prefer mostly the involvement of BU specific AM vendors in application development, which results in AD vendors being mostly involved as subject matter experts during software development. Infosys and Tata Consulting Services (TCS) were assigned as BU specific AM vendors. In addition, five AD vendors were assigned: Accenture, IBM, Infosys, Patni and TCS.

Table 1.1 provides an overview of the assigned AM vendors for AAB’s BUs that are affected by Harvest and Symphony.

<table>
<thead>
<tr>
<th>Business Unit</th>
<th>Assigned Application Maintenance vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netherlands</td>
<td>Tata Consultancy Service</td>
</tr>
<tr>
<td>Asset Management</td>
<td>Infosys</td>
</tr>
<tr>
<td>Private Clients</td>
<td>Tata Consultancy Service</td>
</tr>
<tr>
<td>North America</td>
<td>Infosys</td>
</tr>
<tr>
<td>Latin America</td>
<td>Tata Consultancy Service</td>
</tr>
</tbody>
</table>

Table 1.1 Overview of assigned Harvest vendors to BUs
In addition to the BU-specific AM vendors, every BU has its own test vendor. For a BU, this test vendor is not allowed to be the same as the AM vendor, in order to test in an unbiased way. For example, BU Netherlands has Infosys as test vendor.

1.4 ABN AMRO’s efficiency in respect of IT

As described in the previous section, the Harvest and Symphony initiative aims to better position technology as a strategic asset for the AAB to enable business value and to deliver a better service to their customers. Furthermore, the AAB aims by this program to create an IT organization that responds to the needs of a bank by thinking of the bank first and the IT as an enabler.

In order to attain the most value from the offshore outsourcing initiative, IT has to be used in the most efficient way.

Figure 1.3 shows an efficiency performance grid. On the y-axis, the IT cost in relation to the operating cost are reflected and on the cost/income ratio is shown on the x-axis. The AAB’s position is shown in the right upper box of the grid, where IT overspending is taking place. This grid reflects that the AAB has not been able to get an efficient IT spending yet in the first year (i.e. 2006) after the Harvest & Symphony initiative was launched. At this moment, this lack of internal efficiency towards IT forms a crucial problem for ABN AMRO Global IT Services and for the competitive advantage of the ABN AMRO Group.

Figure 1.3: IT Efficiency Performance ABN AMRO Group (adapted from ABN AMRO Group (2007)) (due to confidentiality the grid is removed)

The relative low IT efficiency performance of the AAB and its resulting IT overspending, is the focus area for this research. Chapter 2 succeeds by discussing the research problem.
Chapter 1 has provided the context and the motivation for executing the research study. This chapter elaborates on what to study. For this reason, this chapter determines the objective of this research study.

According to Van Aken (1994), first the type of research should be determined in which the research study is taken place. This research study belongs to the design sciences, in which the definition of a clear-cut problem plays a crucial role. For this reason, the initial research problem that is carried out by the AAB has to be refined by analyzing this problem in order to make the problem more specific (Kempen et al., 2000). The problem analysis leads to the setting of a research problem definition and its derived research objective. This results in four topics that are addressed in this chapter.

- The identification of the research type
- The setting of the initial research problem
- The problem analysis
- The research problem definition and the derived research objective.

The outline of this chapter has the following structure. Section 2.1 determines what type of research is executed in this research study. Subsequently, Section 2.2 states the initial problem statement that has been carried out by the organization. Section 2.3 provides a thorough problem analysis, which consists of two parts. Firstly, Section 2.4 provides an exploratory problem analysis by the use of literature, followed by Section 2.5 that discusses the conduction of four exploratory case studies. Chapter 2 ends by providing a problem-breakdown, from which the research problem is defined. Subsequently, the objective for this research study is derived from the research problem definition.

2.1 The Type of Research

Basically two types of research can be distinguished: the empirical cycle and the regulative cycle (van Aken et al., 2003). Since the main objective for this research study is to develop a method to solve a problem that is found at the AAB, this research is conducted according to the regulative cycle of Van Strien (1975), as depicted in Figure 2.1.

Van Strien (1975) acknowledges a set of phases during a design-oriented research. The first two phases of this cycle, highlighted by a blue circle, are discussed in this chapter in order to...
identify a problem-web or a problem-breakdown diagram from which the problem definition of this research is determined. These two steps are executed in the orientation phase, according to Verschuuren et al. (1999). Subsequently, the analysis and the succeeding diagnosis phase, as represented in Figure 2.1, result in research methods to gather specific knowledge regarding the specifications and context of the problem. Validity and reliability are important issues in this phase. In the design phase, a solution for the specific problem will be developed and is adjusted to the organizational context in the implementation phase. The final deliverables of the research are the object and the realization design. These will be implemented and evaluated.

It is important to hold a constant scope during the research. Therefore the initial problem must be verified by analyzing the main causes of the problem. Finally, the resulting problem definition must be detailed enough to avoid a wide scope and to make sure the diagnosis is focused. Important from this perspective is the analysis and verification of the initial problem that ABN AMRO Global Services IT has posed, in order to make sure that a solution will be developed for the real problem and not for a perception problem only.

2.2 Initial Problem Statement

The initial problem is derived from several issues posed by ABN AMRO Global Services IT during the preparation and background of the research. These issues are the following.

- ABN AMRO Global Services IT experiences problems in specifying proper requirements to its assigned Harvest & Symphony vendors for building software applications. As a result, the potential cost reductions, which form the main incentive for starting the Harvest and Symphony program, are not realized when the same requirements are to be redefined.

- The AAB’s BUs have small mutual coherence. Often, this results in situations in which “the wheel is reinvented”. A better way of knowledge sharing might lead to time and cost reductions for the AAB in order to attain a higher internal efficiency.

The initial problem that forms the basis for the research is that ABN AMRO Global Services IT faces problems with the specification of their requirements to its Harvest & Symphony vendors. In order to verify whether the initial problem statement forms a real problem, the initial problem is verified (Van Aken et al., 2003).

Initial Problem Verification

In order to verify the initial problem proposed by ABN AMRO Global Services IT, several IT department heads were interviewed during the orientation phase of this research. The complete list of interviewed persons and the interview format are represented in Appendix II.

The interviews regarding the initial problem statement gained insight into the structure of the Global Services IT department. Potential problems, issues and questions have been discussed in the interviews, and the need to a profound way of setting requirements for realizing the intended cost savings of ABN AMRO Global Services IT incurred by the Harvest and Symphony initiative.

This can be transferred into the following verified initial problem definition.

*In the new IT environment resulting from the Harvest and Symphony initiative, it is increasingly important to define requirements in a proper way. ABN AMRO Global IT Services wants to improve the way the specification of requirements is performed at this moment.*
2.3 Problem Analysis

After defining the initial problem statement by conducting orientating interviews for verification, an exploratory problem analysis took place. According to Kempen et al. (2000), the problem analysis follows after setting the initial problem statement. The problem analysis aims to determine the eventual research problem by making the initial problem statement more specific. In this research, the problem analysis is determined by the way of a diagram that breaks down the most compelling problems regarding the specification and management of software requirements at the AAB.

The realization of this problem-breakdown as result of the problem analysis consists of two parts. This first part of the problem analysis provides an introduction in the basics of the Requirements Engineering domain by providing a profound overview of the used terminology in this domain. This part acts as the ideal situation regarding the specification of software requirements. Further, the literature forms the basis for the problem breakdown, which is executed in the second part of the problem analysis. The second part of the problem analysis consists of exploratory case studies, which analyzes how the specification of software requirements is actually taken place at the AAB its BUs.

By identifying the gap between the ideal situation (i.e. the theory) and the actual situation at the AAB (the practice), the research problem definition is identified (Verschuuren et al., 1995). Section 2.4 describes the exploratory problem analysis by the use of literature and Section 2.5 conducts the analysis by the use of exploratory case studies.

2.4 Exploratory Problem Analysis by the use of literature

This section aims to provide a basic understanding and a profound overview of the terminology regarding the Requirements Engineering domain. In order to fulfill this objective, a number of questions have arisen, which are explained in the remainder of this section. The fundamentals for constructing these questions have been the basic “What”, “When” and “How” questions to determine what software requirements are (question 1), when software requirements are set (question 2) and how software requirements are set (question 3, 4 and 5).

1. What are Software requirements?
2. When are Software Requirements set?
3. What is Requirements Engineering?
4. How is the Requirements Specification process executed?
5. How is Requirements Management executed?

By answering the question about what software requirements are, a thorough overview of the decomposition of software requirements in high-level business requirements, interface, user, functional and non-functional requirements is provided. The second question provides an insight into the role of software requirements in the software development lifecycle. Answering this question provides organizational context towards the research field. Next, Requirements Engineering describes the way software requirements are handled in the organization. Requirements Engineering comprises two fields: Requirements Specification in which the actual setting of requirements takes place, and Requirements Management that aims at controlling changes made to software requirements.

Appendix III provides an extensive elaboration of the questions that are stated in this section. Section 3.1.1 discusses how this part of the problem analysis is executed.
2.5 Exploratory Problem Analysis by the use of case studies

This section forms the second part of the problem analysis that is executed in this research, and suits two objectives.

(1) Getting a profound understanding how Requirements Engineering is executed at the BUs of the AAB, and

(2) Identifying the existing problems regarding Requirements Engineering in the four selected BUs of the AAB.

When these two objectives are realized, one particular problem will be selected that exists for all the BUs, since the focus of the Global Services IT department is to look for IT solutions that cover the bank as a whole.

In order to realize these two objectives, case studies are executed at the BUs of the AAB. A case study is defined as a research strategy that investigates a phenomenon within its real-life context (Yin, 2002). This research study applies a case study research, since case study research can rely on multiple sources of evidence and can be based on any mix of quantitative and qualitative evidence (Yin, 2002). Orientating interviews regarding the verification of the initial problem statement revealed that BUs of the AAB have different maturity levels in specifying requirements. Some BUs have formal processes for handling requirements, while other BUs have less experience in this domain. Due to the ability of case study research to handle multiple sources of evidence, this research technique fits this situation.

This research enhances the approach of Eisenhardt (1989) as a methodology for analyzing case studies, which is depicted in Figure 2.2. After setting the objective of the case study, case studies are selected. Subsequently, the research methods are determined for conducting the case studies. After the data have been collected, the case study data are analyzed for each case in particular. Finally, the case data are analyzed between the selected cases during the cross-case analysis.

The methodology of Eisenhardt (1989) is a variation and addition on the methodology of Yin (1984) by the development of a cross-case analysis step. In this way, the methodology is able to realize the two aforementioned objectives in this case study by collecting data for providing elaborated case studies, followed by a case data analysis by constructing a problem-breakdown diagram for Requirements Engineering problems at all the selected BUs. The cross-case analysis enables to select from the diagram one particular problem that exists for all the BUs at the AAB in view of the focus of the Global Services IT department by looking for IT solutions that cover the bank as a whole.

![Figure 2.2 Case study approach (Eisenhardt, 1989)](image)

The outline for this section follows partially the methodology of Eisenhardt (1989), since not every phase in Figure 2.2 is represented in this chapter. Section 2.5.1 begins with the case study selection, since the case study objective has been defined in the beginning of Section 2.5. The research methods are discussed in Chapter 3, since this chapter concern the research methodology of the whole research. Subsequently, a detailed description of the content of the
selected case studies is represented in Appendix V, which is considered as the collect-data step of Figure 2.2. Section 2.5.2 presents the main results of the analyses conducted and Section 2.5.3 discusses the cross-case analysis.

2.5.1 Case Study selection

In order to select representative BUs for the conduction of case studies, the following criteria were taken into account.

- A BU has to have a certain level of experience in Requirements Engineering. Some BUs, like Business Unit Transaction Banking (BUTB) and Business Unit Antonveneta, have little or no explicit documented knowledge regarding Requirements Engineering, which explains their absence in the case studies.

- Due to language barriers, time differences and other cultural difficulties, it was difficult to get in contact with Business Unit Latin America (BULA), which has led to the absence of this BU in the case study.

This resulted in the selection of the following four BUs for case study research.

1. Business Unit Netherlands (BUNL)
2. Business Unit North America (BUNA)
3. Business Unit Asset Management (BUAM)
4. Business Unit Private Clients (BUPC)

By selecting the aforementioned BUs, conclusions are drawn of the problems at the AAB regarding the Requirements Engineering domain. This is important, since the selection of the BUs to be analyzed influences the cross-case analysis and as a result the research objective.

2.5.2 Case Study Results

In Section 2.5, the objective of the case study research has been provided by (1) getting a profound understanding of the way Requirements Engineering is executed at the BUs of the AAB, and by (2) identifying the existing problems regarding Requirements Engineering in the four selected BUs of the AAB.

The results of the first objective are captured in Appendix V, in which elaborated case studies are presented by using semi-structured interviews. The format of these interviews is represented in Appendix IV. Regarding the first objective, it can be concluded that although the four BUs are part of Harvest & Symphony program, there is a striking variation in maturity between BUs. One can easily say, that the retail-oriented BUs like BUNL and BUNA are the most mature in comparison with BUAM and especially BUPC. Maturity can be regarded in the broadest sense of the words implying tooling, a formal way of documenting and storing of requirements.

The second objective is also achieved by conducting semi-structured interviews. This has led to a problem-breakdown, based on the problems that were identified in the four BUs, depicted in Figure 2.4. Further, BUs showed to have different approaches towards software development methodologies. Figure 2.3 shows an example of this variety in use: BUNL uses the Dynamic System Development Method (DSDM), BUNA uses the Methodology for ABN AMRO Projects (MAP), BUAM uses the Product Development Lifecycle and BUPC uses a traditional waterfall mode. A detailed description of the fundamentals of these methodologies is presented in Appendix V.
Regarding the second objective, a problem-breakdown diagram is presented in Figure 2.4. This diagram is constructed by identifying the problems regarding Requirements Engineering in the aforementioned BUs. Subsequently, the interviewees, listed in Appendix IV, validated this diagram, together with the outcomes of the interviews that were held in particular. The problem-breakdown diagram shows that the problems regarding Requirements Engineering for the AAB can be decomposed in three categories: Requirements Specification, Requirements Management and Global Interaction problems. The first category follows as a result of the decomposition structure of Kotonya (1998) regarding Requirements Engineering, represented in Appendix III. The third category is a typical problem for global companies like the AAB, in which BUs inside one organization have small mutual coherence. A striking example is provided in Appendix VI, which shows the wide variety in terminology regarding Requirements Engineering. An extensive explanation of the elements in the problem-breakdown diagram of Figure 2.4 is provided in Appendix VII.

2.5.3 Cross-case Analysis

As pointed out in Section 2.5, during the cross-case analysis one particular problem from the problem-breakdown diagram is selected that forms an existing problem for all the BUs at the AAB. The diagram of Figure 2.4 is a collection of problems from all the BUs, although not every problem in the diagram is applicable for all the BUs. During the cross-case analysis one problem was identified that is applicable for all the BUs according to the above-mentioned criterion. This research problem is discussed in Section 2.6.

2.6 Research Problem Definition

The problem-breakdown diagram forms the basis for a structured approach for defining the research problem definition. As a consequence, the initial problem statement can be refined. Discussion with experts in the organization resulted in selecting one research area by focusing on how reuse of non-functional requirements (NFRs) can take place for all the BUs at the AAB. In Figure 2.4 a blue circle represents this specific problem.

According to Kotonya (1998), NFRs include standards, regulations, and contracts to which the product must conform. These can be descriptions of external interfaces, performance requirements, design and implementation constraints and quality attributes. For an extensive background into the Requirements Engineering theory this paper refers to appendix III.
Figure 2.4 Breakdown of Problems in Requirements Engineering at ABN AMRO
Besides its global approach, mentioned in Section 2.5.3, this specific problem is motivated by the following reasons:

- NFR reuse is a relatively new aspect regarding Requirements Engineering for the AAB. For this reason, the AAB is interested in what extent NFR reuse can increase the internal IT efficiency of the AAB;

- Reusing NFRs can be considered as a way knowledge sharing is established in BUs and especially between BUs. This potential achievement of creating knowledge synergies between BUs is very interesting for the Global Services IT department. For this reason many stakeholders are interested in this subject, although there is no clear idea how NFR reuse can actually be embedded in the Requirements Specification process;

- NFRs contain the characteristic features of being relatively easy to reuse, in comparison with functional requirements (Chung et al., 2000). NFRs do not reflect the functionality of a specific software application like functional requirements (FRs). NFRs reflect the attributes associated with the software application and the external constraints implied, which makes them highly reusable.

This has resulted in a refinement of the initial problem statement: the research problem definition.

In the new IT environment resulting from the Harvest and Symphony initiative, it is increasingly important to reduce the time and costs for specifying software requirements and to improve the quality of the requirements. ABN AMRO Global Services IT is interested in how it can improve its way of specifying one type of software requirements: NFRs, by using reuse methods.

Research Objective

The research objective in problem solving research is closely related with the research problem definition and is defined as follows.

The objective of this research is to improve the way NFRs are specified at the BUs of the AAB, by reusing existing NFRs. An important part of this objective is the elaboration of a model how reuse of NFRs can be embedded in the current lifecycle for requirements specification and how this model can be operationalized for the AAB.

From now on, this thesis will use the term ‘Requirements Specification’ for the process in which requirements are set and reuse of requirements takes place, according to the decomposition of the Requirements Engineering domain by Kotonya (1998), which is described in Appendix III.

This chapter has elaborated on what has to be researched by providing the research problem definition and its derived research objective. Chapter 3 will elaborate on how the research is executed by discussing the research methodology.
3 Research Methodology

The pathway of the research, followed to answer the research questions, is called the research methodology and describes the way the research is performed. Research methodology often distinguishes qualitative and quantitative research methods. This research could not be assigned to be qualitative or quantitative, nor could it be divided into a qualitative and quantitative part in advance. The objective of the research, provided in the last section of Chapter 2, is to construct a conceptual model how reuse of NFRs can be embedded in the Requirements Specification process, which can be used in practice at the AAB. For this reason, theory not only must be built, it must be tested as well. To achieve this objective, it is necessary to divide the research into an exploratory phase and a confirmatory research phase (Onwuegbuzie et al., 2005), as is depicted in Figure 3.1. A theory, certain hypotheses or a framework is connecting the exploration and the confirmation phase to each other (Newman et al., 1998).

Figure 3.1 Research methodology approach (Onwuegbuzie et al., 2005)

The outline of this chapter is the following. Section 3.1 and Section 3.2 discuss how the exploratory and confirmatory research phases were conducted, according to Figure 3.1. Subsequently, Section 3.3 discusses the underlying data collection methods that have been used during the exploratory and confirmatory research phase. The operationalization of the research methodology into activities that have been executed in this research is discussed in Section 3.4 by providing the research model. The corresponding research questions and the resulting outline of the research are presented in Section 3.5 and Section 3.6.

3.1 Exploratory Research phase

An exploratory research is typically executed when there is no hypothesis, assumption or, in this case, conceptual model available. Sometimes, it even has been argued that exploratory research is research into the unknown and often intends developing hypotheses. Furthermore, exploratory research may also be regarded as research that can identify all potentially available variables within a research area.

The primary goals of the exploratory phase of this research is to gain understanding in the way Requirements Engineering is executed at the AAB, and to identify the existing problems regarding Requirements Engineering in the four selected BUs of the AAB. The exploratory problem analysis, as stated in Section 2.3, has enabled a proper understanding of the Requirements Engineering field, executed at the four BUs of the AAB. Furthermore the literature has enabled the derivation of a structure for sorting the different problems regarding Requirements Engineering, as showed in Figure 2.4.
3.1.1 Exploratory Problem Analysis by the use of literature
The exploratory problem analysis by the use of literature has already been discussed extensively in the previous chapter. Concepts available in the literature were used to develop a structure of sorting out different problems regarding Requirements Engineering. The theory is mostly derived from peer-reviewed articles published in scientific journals that were found by desk research. For overview purposes and to create a broader approach towards the impact the setting of proper requirements has on the total software development process, software engineering and requirements textbooks have been used as well. Besides the application of peer-reviewed articles and textbooks in Chapter 2, the same data collection methods were used extensively in Chapter 4.

3.1.2 Exploratory Problem Analysis by the use of case studies
The use of case studies for four BU's of the AAB, has already been discussed extensively in the previous chapter. The case study research material was mostly derived by conducting semi-structured interviews with subject-matter experts (SMEs) in a face-to-face way or by the use of telephone. Further, relevant case study material was gathered by screening all relevant templates that could be applied to the situation towards Requirements Engineering for that particular BU. Finally, physical artifacts like the Requirements Management tool Caliber RM had been used to get a better understanding of the practical consequences for setting and documenting requirements.

3.2 Confirmatory Research Phase
Confirmatory research, or theory testing (Johnston et al., 1999) is a type of research that is performed to confirm a certain assumption or theory. It involves testing results against a hypothesis, specific assumption or framework. Theory testing can be done, using both quantitative and qualitative methods, depending on the chosen data collection methods of the research (Onwuegbuzie et al., 2005).

The objective of the total research is to construct a conceptual model how reuse of NFRs can be embedded in the Requirements Specification process of the AAB. The conceptual model is constructed based on literature. Subsequently, expert stakeholders validate the constructed conceptual model. In this way, the model is tailored to the current Requirements Specification process of the AAB and challenged to the issues that have to be overcome. In order to test the added value of the model for the AAB, confirmatory case study research is executed.

Chapter 5 elaborates on how the constructed conceptual model could be realized in a real-life situation for the AAB. At the end of that chapter the provided method is evaluated by the means of a case study project that has already been executed by the use of interviews. In this way, unforeseen problems are identified, which could emerge during implementation in a real-life situation. This reduces the likelihood for meeting unforeseen cost increases in the future when the method is actually implemented in the organization.

3.3 Exploratory and Confirmatory Data Collection Methods
This section will discuss the data collection methods that were used during the two aforementioned research phases. In this way, this section explains in more detail how the research study has been executed.
From literature it is derived that qualitative methods are well suited to exploratory research (Yin, 1994). However, Onwuegbuzie et al. (2005) support the view that exploratory research can be supported by both qualitative and quantitative methods. Besides, the case study research makes use of a mixed-method data collection approach (Cresswell, 2002). The most important advantage of a mixed-method data collection approach is that potential weaknesses of the methods can be balanced. Four different data collection methods have been used in this research, following the classification of Stake (1995): desk research, participant observations, physical artifacts and interviews. These methods are shown in Figure 3.2, which gives an overview of the data collection methods and their purposes.

![Figure 3.2 Exploratory and Confirmatory Research Data Collection Methods](image)

In the remaining part of this section all the used data collection methods are discussed.

### 3.3.1 Desk Research

Desk research is one of the first tasks of the researcher and concerns the research of data that are already available in the company. Often these are data that have been collected by others or other institutions and companies. The purpose of desk research is to raise the quality of the research. The main advantages of desk research are that it is relatively fast and cheap. Disadvantages of desk research are that not all relevant data are easily accessible and that the information that is found is not always suitable to solve the researched problem area.

The desk research performed in this research can be considered as purely qualitative research and is performed in the following way. First, all the sources within the AAB have been identified and used to collect information valuable for the research. The most important elements are the ABN AMRO Intranet, annual reports, available process architecture schemes, memoranda and administrative documents of the stakeholders in that particular BU.

### 3.3.2 Physical Artifacts

Physical artifacts can be tools, instruments, or some other physical evidence that may be collected during the research. The physical artifacts that have been used in this research aimed to get experience how Requirements Engineering and Management is executed at BUNL. In fact, BUNL uses the requirement management system Caliber RM, which has been described extensively in appendix X. By experiencing the capabilities and the use of this system, a better understanding is created what the critical activities are. This method has been solely used during the exploratory research phase of the research.
3.3.3 Participant Observations

Participant-observation makes the researcher into an active participant in the events being studied (Stake, 1995). This research method has been used in two ways. First, attendance in the monthly conference calls initiated by the Requirements Engineering and Management (REM) Forum, created a good idea about the most current issues in this domain for the AAB. This method was used in the exploratory research phase.

Second, several seminars were visited. By visiting seminars, a proper understanding of the research field is generated in an interactive way. Furthermore, the participant is able to get a proper understanding of the current developments in the research field. For this reason, the attendance of seminars was especially useful in the exploratory research phase. Appendix II shows a list with seminars, which have been attended.

3.3.4 Interviewing

The last data collection method that has been used in this research is also done in a qualitative manner. The goal of this method is to capture the voice of the interviewee (Onwuegbuzie et al., 2005). First, this step comprehends orientating interviewing with involved ABN AMRO IT managers in order to get a proper overview of the different problem areas. Second, BU specific Requirements Engineering SMEs were interviewed in order to understand how is dealt with Requirement Engineering in the four BUs. Each interviewed employee represents the business, the retained IT organization and the vendors for the particular BU. A semi-structured interview format was used. In fact, the structured interview is similar to a survey, and is used to gather data in cases such as neighborhood studies. The questions are detailed and developed in advance, much as they are in a survey (Emans, 1990). The interview list is shown in Appendix IV, which shows the different interviewees. Although the format was semi-structured, enough space was given to probe into aspects that were important for the interviewee, enabling the identification of factors that resulted in the biggest problems during Requirements Engineering.

The interviews formed the main source for the problem-breakdown diagram in Figure 2.4. As a conclusion, the interviewees validated this diagram, together with the outcomes of the interviews that were held in particular. This has resulted in a problem-breakdown diagram with a high validity.

During the confirmatory research phase, validation interviews were held with experts to tailor the constructed conceptual model derived from literature to the AAB. Due to their extensive knowledge of experts in specifying and capturing NFRs, together with the qualitative nature of the conceptual model that was constructed, validation interviews suited as research method.

3.4 Research Model

The research model represents the research design and forms for this research the operationalization of the methodology presented in Figure 3.1. Figure 3.1 defines the different research phases in this research study, while the research model describes the most important activities of all the phases in the research study. The research model is constructed by the use of the research design elements of Verschuuren and Doorewaard (1995), which are reflected in the relations between activities that are indicated with arrows and double arrows. As a result, the research model gives a comprehensive overview of how the research is designed and structured and is depicted in Figure 3.3.

The research model is explained in the following way. During the orientation phase of the research, the initial problem is analyzed by the means of a literature study and by the use of case studies. This resulted in the problem-breakdown diagram of Figure 2.4. Subsequently, after
through analysis and discussion with experts, the research objective was defined. In order to
meet this objective, first a model from literature is developed. After this model has been
constructed, the model is adjusted to the current Requirements Engineering process of the AAB.
Subsequently, the adjusted model is suited for practical application. This results in actions that
are undertaken how the conceptual model should be realized. The research ends by drawing the
conclusions and recommendations and by defining further research areas.

3.5 Research Questions
In this section the research questions are defined. The research questions embrace the remaining
part of this research. The research questions are answered during the remainder of this research
for which at the end conclusions are drawn and recommendations are made. The questions are
classified per phase of the regulative cycle of Van Strien (1975). This cycle has been selected to
represent the different phases during this research in Section 2.1 and is graphically represented
in Figure 2.1. This has led to the following research questions, which are represented in Table
3.1.

<table>
<thead>
<tr>
<th>Q</th>
<th>Phase</th>
<th>Question</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Diagnosis (Analysis)</td>
<td>Which method is used for designing a conceptual model where reuse of NFRs is embedded in the Requirements Specification process?</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>Q2</td>
<td>Design</td>
<td>What does the constructed conceptual model derived from literature look like?</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>Q3</td>
<td>Design</td>
<td>How should the designed model be adapted to the current situation of ABN AMRO?</td>
<td>Chapter 5</td>
</tr>
<tr>
<td>Q4</td>
<td>Implementation</td>
<td>In what way can the designed model be operationalized to suit practical application for ABN AMRO?</td>
<td>Chapter 5</td>
</tr>
<tr>
<td>Q5</td>
<td>Evaluation</td>
<td>Which further research areas can be defined?</td>
<td>Chapter 6</td>
</tr>
</tbody>
</table>

Table 3.1 Research Questions classified per Research Phase of Van Strien (1975)
Table 3.1 does not present a research question during the orientation phase of the research. The underlying reason is that the orientation phase, as depicted by the blue circle in Figure 2.1, has been described in Chapter 2. The objective of Chapter 2 is the setting of research problem definition and its derived research objective, which form the starting-point for the research to be conducted.

**Deliverables**

As deliverables of this research are considered:

- Elaborated case studies how Requirements Engineering is executed at ABN AMRO’s BUs Netherlands, North America, Asset Management and Private Clients;
- A conceptual model derived from literature how reuse of NFRs can take place;
- A graphical map how the designed model from literature is adjusted to the Requirements Specification Process of ABN AMRO;
- A structured and flexible method how the designed model could be operationalized in embracing reuse of NFRs.
- Implementation steps for guiding the AAB before the aforementioned structured and flexible method can be actually used.
- Fields for further research.

### 3.6 Outline of the report

The outline of the research is represented schematically in the last column of Table 3.1. This section discusses for every chapter the corresponding research questions.

In the next chapter, Chapter 4, the conceptual model will be derived from literature. In order to construct this model, first Research Question Q1 has to be answered. When a method has been selected, the conceptual model can be constructed that results in the answering of Research Question Q2. In Chapter 5 the conceptual model of Chapter 4 is first adjusted to the current Requirements Specification process at the AAB, in line with Research Question Q3. Subsequently, a method is developed to operationalize the conceptual model to suit practical application for the AAB, according to Research Question Q4. Finally, in Chapter 6, the conclusions and recommendations will be provided as a result of this research, together with the design of further research, listed in Research Question Q5.
Chapter 2 ended by providing the problem-breakdown diagram of Figure 2.4, which formed the basis for defining the objective of this research. The objective of this research study became to provide a structured model how reuse of NFRs can take place in the current Requirements Specification process of the AAB. Chapter 3 provided the pathway to accomplish this by first deriving a model from literature, followed by tailoring this model to the AAB. According to Van Aken et al. (2003), before a model can be derived from literature, design criteria have to be specified, which the designed model should meet. When the design criteria have been made clear, the model can be constructed.

The outline of this chapter is the following. First, the current situation of the AAB is identified regarding the reuse of NFRs in the Requirements Specification process, in order to derive design requirements for creating the conceptual model. This results in the identification of three design requirements to which the conceptual should meet (i.e. knowledge management, Requirements Specification process, high-quality product). In Section 4.2 it appears that no direct method could be selected from literature that suits all the three design criteria. As a consequence, the three design criteria are becoming three complementary research fields. For each field a best-practices solution is identified, according to design requirements that are imposed for each of the three research fields in particular. As a result, the three (best-practices) models are merged in Section 4.6 into the conceptual model that forms the basis for Chapter 5. Finally, Section 4.7 draws the conclusions for this chapter as a whole.

4.1 Identifying design criteria

According to Van Aken et al. (2003) the diagnosis of the problems that are present in the current situation of an organization forms the basis for the criteria to which the designed model should meet. In order to conduct this diagnosis, two successive actions were undertaken:

- First, this research gained insight into the way NFRs are set in the Requirements Specification process of the AAB. The result of this action is represented in appendix IX. Five questions were posed in order to gain insight into this process:
  - What different NFRs exist throughout the organization?
  - Can these NFRs be classified in types of similar characteristic features?
  - When are NFRs set during the Requirements Specification process?
  - How can NFRs be retrieved throughout the organization?
  - Who are the different stakeholders in specifying NFRs?

- Discussion with experts in the organization to understand how reuse of NFRs is taking place at the AAB at this moment.

The aforementioned actions have led to the following conclusions regarding the design of a NFR specification model that enhances NFR reuse:

1. NFRs differ highly in character. In line with this characteristic feature, NFRs have to be retrieved throughout the organization. It may be worthwhile to store NFRs in a less dispersed way.
2. At this moment there is no formal process at the AAB during the specification of requirements in which reuse of NFRs is actually taking place. When a formal process is developed, it should also manage the interrelationship between conflicting NFRs.

3. NFRs have to be specified in specific, measurable and unambiguous terms. Specifying NFRs is a process where concern for quality is vital. In fact, a subset NFRs has to cover fully the high-level business need in order to become a high quality NFR. For this reason, a systematical dealing with NFRs has to generate software applications that meet user expectations fully.

Based on the conclusions that are listed above, the criteria can be set up for designing a model that enhances NFR reuse. The order of the design criteria listed below relates to the conclusions that are drawn above.

1. In order to handle the dispersedly located NFRs, the model to be designed should have the form of a knowledge management process in which NFR knowledge is generated and NFR knowledge is reused. Knowledge management activities would prove very helpful for NFRs, since a NFR is form of knowledge. As mentioned before, NFRs vary highly in character between each other and are dispersed in the organization at this moment. Identification of the content (i.e. making tacit NFR knowledge explicit) and the location (i.e. a shared repository for NFRs) of NFR knowledge would be extremely beneficial to the AAB. This improved way of reusing NFR knowledge in the Requirements Specification process increases the internal productivity, since the use of knowledge management leads to decreasing software project’s development time and cost.

2. The model to be designed should fit in the current Requirements Specification process and should handle NFRs in a structured way. The objective of the Requirements Specification process is to deliver a validated set of requirements, captured in a Requirements Specification document. One special type of requirement that has to be captured in this phase is the NFR. However, relatively little attention has been paid to deal NFRs in a structured way, resulting in the fact that NFRs are often retrofitted late in the Requirements Specification process. Embedding reuse of NFRs in the early phases of the requirements specification process can lead to a better utilization of available resources, encouragement of systematic reuse across the entire software development and developmental assistance early in the lifecycle (Agresti et al., 1988).

3. The model to be designed should develop high-quality NFRs that are appropriate for reuse early in the Requirements Specification process. Product quality and product development efficiency has become an important issue in software development. According to Chung et al. (1995), the quality of a software product depends largely on the quality of the software process that leads in this situation from high-level business requirements to the detailed requirement. For this reason, the implementation of quality improvement techniques during the requirements specification process increases the quality of the product (i.e. the NFR). Besides, quality improvement techniques also lead to increased analyst and programmer productivity, fewer design changes and a reduction in the number of errors passed from one phase to the next. Further, high-quality NFRs form the basis for systematically reuse throughout the Requirements specification process. These resulting new quality software applications require less maintenance and allow departments to shift budgets from maintenance to new product development.
4.2 Selecting a method

So far, it has become clear what the design criteria are for constructing the conceptual model. A conceptual model aims to describe a phenomenon and can be constructed by the use of literature. Therefore, a method available in literature is selected that meets the design criteria that are listed in Section 4.1. Table 4.1 summarizes the methods reviewed according to the three aforementioned criteria. These methods are described more profoundly in appendix X.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Knowledge Management</th>
<th>Requirements Specification</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIREN (Toval et al., 2002)</td>
<td>Simple REuse of software RequiremNets: reuse based approach based on requirements documents hierarchy.</td>
<td>Yes</td>
<td>Only security NFRs</td>
<td>No</td>
</tr>
<tr>
<td>Cybulski classification (Cybulski, 1998)</td>
<td>Collection of requirements reuse patterns based on the structural properties of the requirements documents themselves and the processes used in their production.</td>
<td>No. Purely focus on techniques</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reuse based process Kang et al. (1992)</td>
<td>Six-phase life cycle model in which reuse activities find place for every stage.</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Cyneiros et al. (2001)</td>
<td>This method aims to integrate NFRs into conceptual models to tackle the problem of representing NFRs and to understand their impact on conceptual model design.</td>
<td>No. The intended solution favors a meta model approach</td>
<td>No</td>
<td>Yes. Focus on the setting of high-quality NFRs that not conflict.</td>
</tr>
</tbody>
</table>

Table 4.1 Survey of existing methods for reusing NFRs during Requirements Specification

From Table 4.1 it appears that the SIREN approach of Toval et al. (2002) is the only method that meets the three design criteria most. However, the aim of this chapter is to design a conceptual model that meets all the three identified criteria and represents the ideal situation derived from literature. Therefore, for each design criterion a best practice model derived from literature is selected, so each design criteria is in fact becoming a research field. These three research fields are complementary to each other, since the objective of this research is to construct a model where all the aforementioned design criteria have to be included. This procedure is schematically represented in Figure 4.1, in which the eventually constructed conceptual model comprises three research fields for which only a part of the selected fields fits to the context in which the AAB specifies its NFRs.

Figure 4.1 Methodology for constructing the Conceptual Model

The remainder of this chapter describes for each field the best practices on the basis of predefined design requirements. As mentioned above, the design criteria constructed in section 4.1 have turned into research fields for which design requirements have to be formulated. The reason for this is twofold. First, drawing design requirements narrows down the research field resulting in a structured and a focused research. Second, by drawing design requirements, the
quality of the constructed model can be assured by checking if the intended design requirements reflect the elements in the constructed conceptual model.

For clarification, this report uses the term \textit{design requirements} for selecting best practices from the aforementioned research fields. However, this must not be confused with the concept requirements, which forms the subject for this research.

Finally, these best-fitted models are merged into the conceptual model, that is constructed in section 4.6.

4.3 Selecting a Knowledge Management Model

This section selects a knowledge management model that serves as input for the conceptual model of section 4.6. First, this section makes a distinction between different types of knowledge. Secondly, the design requirements are presented and explained in Section 4.3.2. Subsequently, a selection takes place between different knowledge management models. Section 4.3.4 describes reuse activities during the Requirements Specification process to stress the most crucial activities, based on existing literature about software requirements reuse. Section 4.3 ends by drawing conclusions regarding the eventual selected knowledge management model.

4.3.1 What is knowledge?

Before going further into how knowledge management can address organizational needs, this research introduces some key knowledge management concepts:

\textbf{Knowledge levels}

The three levels of refinement to knowledge items are data, information and knowledge (Rus et al., 2002). Data consists of discrete, objective facts about events but nothing about its own importance or relevance. It can be considered as raw material for creating information. Information is data that is organized to make it useful for end users who perform tasks and make decisions. Knowledge is broader than data and information and requires understanding of information. It is not only contained in information, but also in the relationship among information items, their classification and metadata. According to Weggeman (1997) knowledge is a personnel capacity that should be seen as the product of the information, the experience, the skills and the attitude, which someone has at a certain point in time. For this reason, knowledge is more than information.

\textbf{Knowledge characteristics}

A well-known distinction in defining knowledge characteristics is the distinction between explicit and tacit knowledge by Nonaka and Takeuchi (1995).

\begin{center}
\includegraphics[width=0.6\textwidth]{figure4.2.png}
\end{center}

\textit{Figure 4.2 Explicit and Tacit Knowledge (Nonaka et al., 1995)}
Explicit knowledge, also known as codified knowledge, is expressed knowledge. It corresponds to the information and skills that employees can easily communicate and document, such as processes, templates and data.

Tacit knowledge is personal knowledge that people gain through experience. This can be hard to express and is largely influenced by their beliefs, perspectives and values.

In this respect, the knowledge that can be expressed in words and numbers is just the tip of the iceberg, like depicted in Figure 4.2.

4.3.2 Design Requirements towards a Knowledge Management Model

The first design requirement for selecting a knowledge management model is that this model should reflect knowledge management activities in a process-oriented approach. This means that the emphasis has to be laid on the process of "working smarter" by the creation of activities for reusing existing knowledge in the company. The underlying reason for this design requirement finds its way in the dispersed locations where NFR knowledge is found in the organization. For this reason, this typical characteristic feature has to be reflected in the selected model.

Furthermore, the selected knowledge management model has to be cyclic and has to embrace that knowledge is created and used in a continuous way. The reason for this is that the knowledge management model has to fit in an iterative process in which Requirements Specification is taking place at this moment for the BUs of the AAB.

Next, key in managing knowledge is the type of knowledge that has to be managed. At the AAB, NFRs are listed in Requirements documents as explicit knowledge. For this reason, the selected model has to deliver explicit reusable information as a result of the knowledge management process. This can be done in two ways: (1) by making tacit knowledge explicit and (2) by keeping explicit knowledge up-to-date.

Finally, the selected knowledge management model must have experience in the context of the total field of this research (i.e. Requirements Engineering). By choosing a model with a proven track record in Requirements Engineering, it is more likely that this model improves the knowledge management process for specifying requirements for the AAB.

4.3.3 Selecting a Knowledge Management Model

Table 4.2 shows the collection of a knowledge management models according to the aforementioned design requirements. The methods are described more profoundly in appendix X.

<table>
<thead>
<tr>
<th>Model</th>
<th>Short description</th>
<th>Process-oriented approach</th>
<th>Cyclic and continuous knowledge</th>
<th>Produce explicit knowledge</th>
<th>Experience in requirements engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>De Jarnett (1996)</td>
<td>Generic knowledge management model: knowledge creation, capture and dissemination</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Demerest (1997)</td>
<td>Emphasizes that knowledge is embodied in the organization through social interchange</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Knowledge value chain</td>
<td>Successive constituent knowledge management processes started by a knowledge gap in the organization</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Weggeman (1997)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2 Survey of existing models for selecting a knowledge management model

It turned out that the knowledge value chain of Weggeman (1997) meets the criteria best. Weggeman (1997) defines knowledge management as arranging and managing the operational processes in the knowledge value chain in such a way that realizing the collective ambition, the
targets and the strategy of the organization is being promoted. This is a continuous process. This ‘knowledge value chain’ distinguishes successive constituent processes, as depicted in Figure 4.3: specifying the knowledge needed according to the strategy of the organization, drawing up the inventory of available knowledge, developing knowledge, sharing knowledge, applying knowledge and evaluating knowledge.

According to Weggeman (1997) the first concern for effective knowledge management is to develop a strategic need and a collective ambition. This ambition stimulates an effective bond with the organization. Subsequently, the knowledge gap needs to be determined. This is the quantitative and the qualitative difference between the knowledge needed and the available knowledge in the organization. In the third place, this knowledge gap needs to be narrowed by developing new knowledge, by buying knowledge, by improving existing knowledge or by getting rid of knowledge that is out of date or has become irrelevant. The first three steps are depicted in the arrow in the first step (step 1). Thereafter, the available knowledge is disseminated in step 2 (e.g. knowledge sharing) and applied (i.e. applying knowledge) in step 3 to serve the interest of the customers and other stakeholders. Finally, the knowledge is evaluated (step 4) and will form as explicit knowledge the input for the available knowledge in the organization (i.e. determining available knowledge).

Figure 4.3 The knowledge value chain (Weggeman, 1997)

However, the only drawback of the selected knowledge management model is that this model not specifically manages software development knowledge, as being one of the design requirements. Analyzing knowledge management during software development turned out that management of software development knowledge is strongly related to software reuse. In Figure 4.3, knowledge sharing and reuse are represented as one phase in the knowledge value chain, highlighted by the blue circle. The knowledge value chain has to be adjusted to software reuse by substituting the knowledge-sharing phase by a software reuse process, in order to suit the design requirements in Table 4.2.

Generally, a software reuse process comprises four basic steps: abstraction, retrieval, specialization and integration (Krueger, 1992; Sutcliffe et al., 1992), as shown in Figure 4.4. Abstraction forms a process that standardizes components by assigning them into uniform profiles. This profile describes interfaces and specific features of each component. In this sense any reuse process implies some sort of metamodelling where uniform features and the structure of the reusable components is defined. These features can be based on syntactic features or on conceptual features where the meaning of semantics of the reusable component is utilized in a new design context.

Within the reuse process abstracted, components are stored in a repository where users can search and retrieve them. The retrieval utilizes repository concepts to the extent reuse can take place through reference (i.e. the same conceptual instance participates in all reuse contexts) or by copy (i.e. the same physical instance is copied to different parts by ‘cut and paste’). Usually, reuse by reference is regarded as more powerful and efficient, but it requires proper technical solutions, like Computer Aided Requirements Engineering (CARE) solutions.
Normally, a selected candidate component cannot satisfy all requirements of the new context. Therefore, they need to be adapted to the specific novel functions that were recognized during the specialization process. Finally, the integration step integrates revised components into a new application or service platform.

4.3.4 Reuse during Requirements Specification

There are many different ways of improving the requirements specification process. Some approaches aim at assisting requirements elicitation, while others propose the use of tools, like CARE tools, to support requirements specification and modeling (Boehm, 1984). A number of projects emphasize the issues of requirements reconciliation and viewpoint analysis or stress the importance of the verification and validation of requirements (Zhang et al., 2001).

This section looks for methods how reuse of previously analyzed and refined requirements documents and their parts can be embedded in the cycle for specifying requirements by the use of existing literature about requirements reuse. The reuse of requirements concerns two categories: domain analysis and the know-how in the control of the system analysis (Roudiès et al., 2001). In the first case, it is a question of identifying generic components to describe the concepts and the functions of the organization. In the second case the objective is to guide the analyst in order to build the method adapted to its context, by assembling fragments of methods.

Introducing reuse practice requires a change of working methods. First of all, it is necessary to have a repository of powerful and useful reusable components. Then, the analyst must systematically consider the reuse in the management of a project. It is thus a question of acquiring the ability for two activities. Roudiès et al. (2001) defines the different steps in both “design by reuse” and “design for reuse”.

- identification, rewriting and documentation of components candidates that are considered to be useful for later projects (design for reuse),
- selection, adaptation and assembly of relevant components for a project in progress (design by reuse).

During the design by reuse, the refinement of each requirement concept is tackled by a specific pattern (e.g. a context element, a goal and a specific requirement pattern). For further research this paper refers to the work of Roudiès et al. (2001).

In order to illustrate reuse activities with the development and the reuse of a requirements diagnosis, a language of patterns is needed that is dedicated to requirements elicitation.
Patterning aims to capture know-how in Requirements Engineering and to propose solutions in the form of patterns to recurrent problems of analysis. This paper refers to the work of Cybulski (1998) for an extensive elaboration for the different applications for patterns in requirements reuse.

4.3.5 Conclusions selected Knowledge Management Model

In the beginning of this section, design requirements were provided for selecting a knowledge management model (i.e. process-oriented approach, a cyclic and continuous process, produce explicit knowledge and experience in the Requirements Specification field). It turned out that the knowledge value chain of Weggeman (1997), adjusted for software reuse, would be the best-fitted model. The basis for selecting this model lies first in a process-oriented approach towards knowledge management, rather than being a people solution or a technical solution. This means that the emphasis has been laid on the process of “working smarter” by reusing existing knowledge in the company. Furthermore, the knowledge value chain embraces a total cyclic continuous process towards knowledge management. Next, the knowledge value chain produces explicit knowledge as result of the knowledge management process. In fact, only explicit knowledge can be reused resulting in two different approaches possible towards knowledge sharing: transforming tacit knowledge into explicit knowledge and keeping explicit knowledge up-to-date.

The only drawback of the knowledge value chain was that this model did not have a proven experience towards software development. Section 4.3.3 revealed that software development approaches towards knowledge management are strongly related to software reuse. As a consequence, the knowledge-sharing phase of the value chain of Weggeman (1997) had to be substituted by the software reuse process of Zhang (2001) (i.e. abstraction, integration, retrieval and specification).

This section concluded by describing how reuse could take place during requirements specification. Especially, the “design by reuse” method, described by Roudès et al. (2001) can be considered as useful methods for reusing requirements. Further, refinement of knowledge can be considered as important during requirements reuse. Refinement of reusable requirements can be established by specific patterning, usually by context element, goal and requirement patterning.

4.4 Selecting a Requirements Specification Process for NFRs

This section discusses the selection of Requirements Specification process. First, the design requirements for selecting a Requirements Specification model for NFRs are discussed. This will be followed by a selection of this model. Subsequently, a method is discussed that forms a refinement of the selected model. Finally conclusions are drawn.

4.4.1 Design Requirements towards a Specification process for NFRs

As defined in Chapter 2, Requirements Engineering can be divided in Requirements Specification and Requirements Management. In the Requirements Specification process NFRs are captured and documented.

The Requirements Specification model that has to be developed has to have a continuous and iterative character in setting NFRs, as the process of specifying requirements cannot be considered as a one step process in the total software development lifecycle. This design requirement is different from the first design requirement in Section 4.3.2 by its different
aggregation levels for constructing the conceptual model, which will be discussed in Section 4.6. Furthermore, the selected model has to fit to the character of NFRs at the AAB. This is established in two ways. Since NFRs are highly dispersed throughout the organization, the selected model has to contain a step in which the different types of NFRs are brought together. Secondly, when the NFRs are eventually captured, the NFRs have to be handled in a structured way, since NFRs are likely to change during the iterations of the Requirements Specification process.

### 4.4.2 Selecting a Specification process for NFRs

Table 4.3 shows a comparison of specification models for NFRs according to the design requirements in the previous subsection. The spiral model is described more profoundly in Appendix III.

<table>
<thead>
<tr>
<th>Model</th>
<th>Short description</th>
<th>Iterative Character</th>
<th>Dispersed Requirements gathering</th>
<th>Structured NFR approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spiral Model (Kotonya, 1998)</td>
<td>The traditional four step model for specifying requirements (elicitation, analysis, representation and validation)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Five step model (Bamford et al., 2004)</td>
<td>The traditional spiral model is extended with a fifth &quot;manage&quot; step.</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 4.3 Survey of existing models for selecting a specification model for NFRs

According to Table 4.3, the "ISO 9001:2000 model for Requirements Engineering" from Bamford et al. (2004) meets the design requirements most. Although this model is not a spiral model, the model has a continuous approach for specifying requirements. Furthermore, the traditional Representation-step has been substituted by a Capture-step, this model contains an additional step in the center of the model, namely the Manage-step. By this way, the requirements during specification are managed in order to deliver validated requirements. This step can be beneficial to the properties of NFRs regarding their dispersed location in the organization and their wide variety in character. The model is represented in Figure 4.5:

![Figure 4.5 five step Requirements Specification Model (Bamford et al., 2004)](image)

However, the only drawback is that the model in Figure 4.5 does not favor a structured approach to determine conflicting NFRs, as listed in Table 4.3. For this reason, this approach has to be fitted in the Manage-step of the model of Figure 4.5 as indicated by the blue circle. The underlying reason is that a structured approach for dealing with NFRs has the most added value during a step where all the NFRs are brought together (Chung et al., 2000). This structured approach has the form of a goal-oriented approach. These goal-oriented conceptual models have the capability of representing non-functional aspects, such as confidentiality, performance, ease of use and timeliness. By this structured way of dealing with NFRs, conflicts can be determined...
early. In the next section, the model of Chung (2000) is discussed, whose work has formed the basis for capturing NFRs by using a goal-oriented approach.

### 4.4.3 Representing NFRs in a structured approach

Although relatively little has been written about methods to manage NFRs, the model of Chung is profound application for managing NFRs during the early phases of the requirements specification process.

The NFR framework of Chung et al. (2000) is a process-oriented approach aiming to involve NFR setting early in the requirements specification process. Through its structured approach, the framework aims to make the development process less ad hoc and advocates a relatively formal systematic methodology. In order to deal with NFRs in a structured way Chung et al. (2000) developed a goal-oriented approach for representing NFRs is shown in Figure 4.6. In fact, NFRs are often associated with a rich but confusing set of concepts. Chung et al. (2000) treat NFRs as goals that might conflict and are organized into a hierarchy. The NFRs must be represented as soft goals that have to be “satisficed”. Mylopoulos et al. (1992) introduced this term as it can rarely be said that a NFR can be satisfied. “Goal satisficing” suggests that the solution used is expected to satisfy a NFR within acceptable limits.

In Figure 4.6 each goal is decomposed into satisficing goals, represented by a graph structure, inspired by the AND/OR trees used in problem solving. This satisficing goal can also be decomposed using other satisficing goals. This process continues until the requirements engineer considers the goal satisficed. NFRs must follow a set of decomposition methods, which are extensively discussed in Chung’s work. Chung’s work also proposes how to represent and deal with conflicting goals. The idea here is that a sub-goal can impact positively or negatively on another goal or sub goal.

![Figure 4.6 Sort Hierarchy NFRs (Chung et al., 2000)](image)

According to Chung et al. (2000), the development knowledge about specific NFRs are to be taken from literature and industrial experience and has to be captured as methods. These methods are then presented for reuse to help the developer generate new goals and links. For example, techniques can be incorporated from security evaluation criteria, performance principles and accuracy concepts.

### 4.4.4 Conclusions NFRs during Requirements Specification

This section started by providing design criteria for selecting a requirements specification model for dealing with NFRs, as shown in Table 4.3 (i.e. iterative character, dispersed requirements capturing, structured requirements representation). As a result, a combination of the ISO 9001:200 model (iterative character and dispersed requirements capturing) and Chung’s goal-
oriented approach (structured requirements representation), were selected to fully cover the design requirements for constructing the conceptual model of Section 4.6.

4.5 Selecting a Quality Model
This section discusses the selection of a quality model. First, the design requirements for selecting a quality model are discussed. This will be followed by the actual selection of this model. Finally conclusions are drawn about the eventual selected quality model.

4.5.1 Design Requirements towards a Quality Model
The first design requirement towards selecting a quality model is that the model has to be process-oriented. This means that the model has to point out in what way the quality of a production process can be assured and improved.

During the iterative Requirements Specification process, continuous quality control is necessary to guarantee the quality of the process, since the quality of the product (i.e. validated NFRs captured in Requirements Specification documents) depends largely on the quality of the process (Chung et al., 1995). By this way, this research favors to take a process-approach towards a quality model.

Second, the quality model to be selected must not be an operational tool that constantly manages the quality performance. This model must be a mechanism to assure the quality during the specification of requirements. By selecting a quality model that forms a mechanism for assuring process quality, quality is fully integrated during the creation of requirements. In fact, a well-defined software application is only perfect when it reflects the customer requirements in a totally exhaustive way.

Third, the quality model has to ensure that quality is embedded during the early phases of the (software development) lifecycle. In fact, the actual setting of requirements takes place in the early phases of the software development process, so a quality model with this characteristic feature is likely to be more valid than a model that only ensures the quality in design phases.

Finally, the selected model has to fit in the research environment, so a quality model has to be chosen with experience in the field of respectively Software Engineering and Requirements Engineering. By choosing a quality model with a proven track record in this research area, it is more likely this model improves the quality for specifying requirements.

4.5.2 Selecting a Quality Model
Table 4.4 shows the selection of a quality model according to the aforementioned design requirements. The models that are not selected are described more profoundly in appendix X.

<table>
<thead>
<tr>
<th>Model</th>
<th>Short description</th>
<th>Process-oriented</th>
<th>No control tool</th>
<th>Early Phase involvement</th>
<th>Experience in requirements engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFD (Akao, 1990)</td>
<td>Quality Function Deployment: Technique to deploy quality according to customer needs by the use of the “four houses of Quality”</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>FMEA (Dale et al., 1988)</td>
<td>Failure Mode and Effect Analysis: Systemized group of activities intended to recognize and evaluate potential failures</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CMM (Paulk, 1992)</td>
<td>Capability Maturity Model: This model examines six dimensions, rating each on a scale of 1 to 5 to determine the process quality maturity</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SQFD</td>
<td>Software Quality Function</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
The QFD framework adjusted to software development.

Table 4.4 Survey of existing models for selecting a quality model

It turns out that the Software Quality Function Deployment (SQFD) framework developed by Haag et al. (1996) is the best-fitted quality model, according the design constraints. The SQFD framework represents a transfer of the technology of QFD from its traditional manufacturing environment to the software development environment (Haag et al., 1996). The SQFD framework is discussed in the next section.

4.5.3 SQFD Framework

According to ISO 9001:2000 (Bamford et al., 2004) the quality of setting requirements for software and system providers can be best reassured by executing a QFD. Haag et al. (1996) has refined the QFD to the SQFD framework.

The SQFD framework focuses on improving the quality of the software development process by implementing quality improvement techniques especially during the requirements elicitation phase. According to Haag et al. (1996), these quality improvement techniques lead to increased productivity, fewer design changes, a reduction in the number of errors passed from one phase to another, and quality software systems that satisfy customer requirements. Figure 4.7 represents this basic SQFD model.

Although this model has many similarities with the QFD framework, which has been described in appendix X, the SQFD is explained in the following way: In step 1, represented as sphere 1, customer requirements are elicited. The “customers” include end users, managers and software development personnel. The requirements are short statements that are defined in customers' terminology. Step 2 on the top y-axis considers the business requirements that are converted to technical and measurable statements. In step 3, the correlation matrix is constructed by identifying the strength of the relationships between the various customer requirements and the technical product specifications. Based on customer survey data, the requirement priorities for the stated customer requirements are developed and listed on the right x-axis. This is shown in step 4. The process that is shown in step 5 involves the development of the technical product specification priorities by summing the results and multiplying the customer requirement priorities by the correlation values between the customer requirements and the technical product specifications. The end result of the SQFD process contains measurable technical product specifications, their importance percentage, and targeted measures.
4.5.4 Conclusions Selected Quality Model

This section started by providing design requirements for selecting a quality model, as shown in Table 4.4 (i.e. process-oriented, no control tool, early phase involvement and adjusted to software engineering). The fundamentals of the SQFD framework, as extension of the traditional QFD, suit these design criteria most. Since the SQFD is considered as an extension of the traditional QFD, the SQFD inherits all the characteristic features of the QFD. Customer orientation and process rather than product orientation characterize the SQFD framework as being based on the concept of Company Wide Quality Control (CWQC). The philosophy of selecting a process-orientated quality model is in line with the aforementioned design requirements.

Next, the SQFD is not an operational mechanism that controls the performance on a temporary basis, but this framework aims to increase the quality of the total process. This framework enables an organization to “build” quality into the product and to enhance the development process from concept to the operations.

Furthermore, in contrast with the FMEA, the SQFD has a proven to be extensively useful in the early phases of the software development lifecycle (Dale et al., 1988). Finally, the SQFD can be considered as a well-applied QFD framework regarding software development.

4.6 Constructing the Conceptual Model

This section constructs the conceptual model that embeds reuse of NFRs in the traditional Requirements Specification process. In order to construct the conceptual model, design requirements were set up to select the best-fitted models (i.e. knowledge management model, specification model for NFRs and the quality model). These three models form the building blocks for constructing the conceptual model. For this reason, the conceptual model comprises three parts.

1. The first building block is the knowledge management part, which explain on the highest aggregation level what the total knowledge management process looks like. The value chain of Weggeman (1997) has been selected in Section 4.3, in which the knowledge-sharing phase in this chain is substituted by the traditional software reuse process.

2. The second part consists of an adaptation of the traditional Requirements Specification process, by providing a five-step requirements specification model. In order to suit the design requirements fully, the Manage-step should capture NFRs in such a way that conflicting NFRs that could possibly constrain each other are early identified. This research selected the goal-oriented approach of Chung (2000) in Section 4.4, for managing conflicting NFRs.

3. The third part of the conceptual model considers the internal quality of the process. This has resulted in the selection of the SQFD model (Haag et al., 1996). Although the influence of this part is not graphically reflected in the eventually constructed model, this part aims to create high-level NFRs. According to the SQFD model of Section 4.5, high-level NFRs have to be decomposed in the most exhaustive subset of detailed NFRs, and should have scores to reflect their criticality.

The remainder of this section is structured as follows. The first two building blocks are represented schematically in the conceptual model and the elements in these models are described. Subsequently, the effect of the third building block is described. Finally, the three building blocks are merged into the eventually constructed conceptual model.
Building Block 1: A Knowledge Management Model for NFRs

Figure 4.8 provides the first building block of the conceptual model. This building block represents the knowledge value chain of Weggeman (1997) in Figure 4.3. The four steps that are highlighted and depicted in Figure 4.3 are described in this building block.

The high-level business requirements form the input for the Requirements Specification process and determine the scope of the system to be built, and as a consequence the amount and character of the NFR knowledge needed. In this way, the high-level business requirements influence which NFRs have to be specified. For this reason the line is dashed. Subsequently, a gap analysis takes place to determine which NFR knowledge is needed and which knowledge is available in the organization, according to step 1 in Figure 4.3. This process is represented in Figure 4.8 by first gathering already existing NFRs from the reusable repository that have to be integrated in the Requirements Specification process, together with the development of NFR knowledge that is specific to the situation. This process is represented in a simplified way as one block, so the place of the arrows in Figure 4.8 does not reflect the actual input and output of NFRs.

The process continues with step 2 in Figure 4.3 in which knowledge sharing takes place. As mentioned before, this step is substituted by the generic software reuse processes of Zhang et al (2001) from Figure 4.4 that represents the reuse process of existing NFRs (i.e. abstract, retrieve, specialize and integrate) in this model. The way repository improvement is generated from validated requirements documents concerns the abstraction of reusable components. In this way, tacit knowledge is made explicit by abstracting this NFR knowledge in the repository. Further, the repository enables the retrieval of reusable NFRs. This repository is not considered...
as a final static product, but as an evolutionary product that has to be enriched with those new requirements found throughout the development of the current system and which are considered as likely to be used in forthcoming applications. In order to add value, the reusable components have to be specialized to the situation. This specialization step is represented as a selection step, since a requirement is selected from a repository with reusable components that fit to the current situation. In fact, during design by reuse it is vital to select relevant components for reuse. Finally, the selected reusable NFRs are integrated in the Requirements Specification process, which is simply represented as one block, in which the NFRs are actually used.

The white block, representing the Requirements Specification process should be considered as the application of the reused NFR knowledge in the Requirements Specification process, according to step 3 of the knowledge value chain in Figure 4.3. The content of this block is handled when Building Block 2 is discussed. As depicted in Figure 4.8, empty lists with NFRs form the input for the NFR Selection step. Reusable NFRs from the repository are captured in the NFR list, which results in filled in NFR lists. The NFR list comprises a subset of NFRs, which explains the form of the picture. When the NFR lists are validated, all the validated NFRs form the Requirements Specification document.

The knowledge value chain of Weggeman (1997), depicted in Figure 4.3, ends with the knowledge evaluation step of step 4. Normally, all the relevant stakeholders at the end of the Requirements Specification process validate the NFRs. An example of all the relevant stakeholders at the AAB during the specification of NFRs is represented in appendix IX.

Knowledge evaluation occurs by keeping explicit knowledge up-to-date by evaluating already existing NFR knowledge in the repository with the newly updated NFRs that are generated in the Repository Improvement step of Figure 4.8.

Building Block 2: A Specification Model for NFRs

The second building block discusses in more detail the process that has been represented in Figure 4.8 as a white box where application of the reused NFR knowledge takes place. Figure 4.5 has already presented the selected model of Bamford et al. (2004) for specifying NFRs. This model was further adapted by substituting the Manage-step by the goal-oriented approach of Chung et al. (2000). NFRs can change during iterations and could conflict each other. A goal-oriented approach identifies these conflicting NFRs in the early phase of the Requirements Specification process. Further, NFRs are handled in a structured way which results in a structured overview of the total subset of NFRs. Figure 4.9 schematically represents the process for specifying NFRs.
Building Block 3: Developing high-quality NFRs

The last building block has a more detailed approach than the previous two building blocks, since this block elaborates on the quality of the constructed NFRs. However, the third building block is not directly presented in the conceptual model like Building Block 1 and 2. Building Block 3 contains a methodology for delivering high quality NFRs at the beginning of the Requirements Specification process, while Building Block 1 and 2 reflect process steps. As a consequence, the methodology of Building Block 3 is used during the NFR elicitation and NFR analysis and negotiation step that are presented in Building Block 2 and will be explained in the remainder of this section.

In section 4.5 the SQFD framework was assessed as the ideal model for building in quality in the early steps of the lifecycle in an iterative and continuous way. The SQFD framework decomposes customer requirements into technical requirements and gives scores of the correlation between the customer requirements and the resulting technical requirements. In this way, all technical requirements are theoretically represented in this specification document. This structured way of decomposing and listing NFRs increases the quality of the Requirements Specification process, since the likelihood of omitting NFRs is substantially decreased (Haag et al., 1996). Furthermore, high-quality NFRs are created, which support the potential reuse of these NFRs for future projects and enables as a consequence NFR standardization. Subsequently, the gathered NFRs get scores, according to the SQFD model, that reflect the criticality of using these NFRs. This clarifies the negotiation step during the specification of NFRs, since these scores guide the negotiations by reaching consensus of NFR features quicker among stakeholders than before (Haag et al., 1996). As a result, a collectively exhaustive, validated and structured set of NFRs is constructed that are represented in the Requirements Specification document.

The SQFD model can be integrated in the Requirements Specification process of Figure 4.9 as follows. According to Figure 4.10, the decomposition of high-level business requirements (sphere 1) in NFRs (sphere 2) takes place during the NFR elicitation phase, since this phase concerns the gathering of NFRs. The red dashed lines highlight this process. Subsequently, the prioritization of the NFRs takes place in the analysis and negotiation phase, in which an actual trade-off is made which NFRs should or should not be included in the Requirements Specification document. This trade-off analysis, highlighted by the dashed blue lines, is made between the priorities of the high-level business requirements (sphere 4) and the priorities of the decomposed NFRs (sphere 5).
Constructing the building blocks in a conceptual model
As a result, Figure 4.11 represents the eventually constructed conceptual model. The components for constructing this model are already discussed during this chapter. This section discusses the sequence of steps that have to be followed for delivering validated NFRs captured in a Requirements Specification document.

The high-level business requirements form the input for the NFR elicitation phase and determine the scope of the system to be built.
Firstly, during NFR elicitation, a gap analysis is made to discover which NFR knowledge is available in the NFR repository (i.e. available NFRs that can be reused) and which knowledge is not (i.e. specific NFRs). Subsequently, the available NFRs are selected by the Requirements Selection step from the reusable repository. The specific NFR knowledge contains the specific knowledge about the system that is going to be developed. These NFRs are not in the repository, since these are specific for the situation.
The gathering of NFRs during the elicitation phase, as mentioned above, occurs in a structured way by decomposing the high-level business requirements in NFRs as exhaustive as possible. This results in the fact that the empty NFR list evolves into a filled NFR list.

Subsequently, stakeholders negotiate in the analysis and negotiation phase which NFRs have to be implemented eventually in the system and which not. This process is enhanced by the use of correlation scores that are imposed during the decomposition of high-level business requirements into NFRs.

Next, during the NFR representation phase, the NFRs in the NFR list are documented in the Requirements Specification document and are eventually validated by all the stakeholders involved.

Further, the NFRs are continuously approached in a goal-oriented way to determine if the gathered NFRs are not conflicting each other after several iterations have taken place.

The result of the process represented by the conceptual model is a Requirements Specification document that captures a validated subset of NFRs. This document together with all the information that is relevant for the IT vendor for developing the system is sent to the vendor. Finally, the document forms the input for the Repository Improvement step by transforming tacit NFR knowledge into explicit NFR knowledge and by keeping already existing explicit knowledge up-to-date. This last step can be considered as a knowledge evaluation step.

4.7 Conclusions

This chapter has provided a conceptual model how reuse of NFRs could be embedded in the Requirements Specification process. First design criteria were established to select a conceptual model. However, no model available in literature fully met the design criteria. Subsequently, the design criteria became fields for research by selecting a best practice for every field of research. This resulted in three best-practices models that were designed according to imposed design requirements. Eventually, the three selected models formed building blocks that were merged into the conceptual model of Figure 4.11.

The model of Figure 4.11 contains three key elements: the knowledge management element, the five-step requirements specification element and the NFR quality element. The first element enables a formal way of embedding reuse in the Requirements Specification process. The second element aims to early detect conflicting NFRs and the third element aims to improve the quality of NFRs by the use of a structured decomposition of high-level business requirements into NFRs.

The constructed conceptual model forms the basis for the next chapter in which this model is fitted to the situation of the AAB and is suited to practical application.
5 Suiting for Practical Application

The conceptual model, constructed in the previous chapter, represents the content of the organizational change to be made. This chapter represents the plan for realizing the intended organizational change. This plan is a design for the change process; the way to reach the new situation from the old situation.

In order to reach a proper suitability for the constructed conceptual model towards the current Requirements Specification process, the following objectives must be met, adapted from the work of Van Aken (2000):

- The designed change process has to be adjusted to the situation in which the change has to take place (i.e. the Requirements Specification process of the AAB).
- The designed change process has to result in a realization of the designed model. The tangible and intangible costs and risks of the change process have to be as low as possible. Especially the development of resistance against the imposed change has to be taken into consideration. This can be realized by focusing not solely on technical-economical aspects, as well as to political-social aspects.
- The intended change process has to be tested in order to find out whether the aforementioned change process meets the objectives for which it has been created.

The objectives for implementation result in successive steps, which has to be realized. These steps are depicted in Figure 5.1.

5.1 Adjusting the conceptual model to ABN AMRO

According to Figure 5.1, the first step of a successful implementation is to adjust the constructed conceptual model (i.e. the ideal situation), depicted in Figure 4.11, to the current Requirements Specification process at the AAB. In order to realize this, first design criteria are presented and explained. Subsequently, the limitations for each building block of the constructed conceptual model to the AAB are identified. Finally, the adjusted building blocks form the renewed conceptual model.
5.1.1 Selecting an approach

Before discussing the different approaches in order to successfully fit the model of Figure 4.11 to the AAB, first design criteria are established to evaluate the applicability of these approaches. These criteria are derived from the desires of the stakeholders at the AAB and the context of the Requirements Specification process at the AAB. Key in selecting an approach is that stakeholders have to know how this model would (possibly) conflict with the current Requirements Specification process. An early identification of problems, limitations and challenges regarding the fit of the designed model to the existing process reduces the likelihood of meeting unexpected costs during real-life implementation. Further stakeholders were interested in getting a high-level understanding what the impact of the model is on the total process. In this way, stakeholders can easily discover the consequences of this model to their current way of working. Next, the approach selected has to enhance that the quality of the “fitness of the model” is continuously refined during the validation by stakeholders, since an improved quality of the refined model imposes a higher relevance for the AAB. This has resulted in the following selection criteria:

- Early identification of potential problems and limitations;
- Creation of a high-level understanding;
- Iterative quality improvement.

Table 5.1 summarizes the reviewed approaches according to these three design criteria:

<table>
<thead>
<tr>
<th>Approach</th>
<th>Description</th>
<th>Early Problem Identification</th>
<th>High-level understanding</th>
<th>Iterative Quality Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delta Analysis (Van Aken, 2002)</td>
<td>Conduction of a gap-analysis to discover the current situation and the ideal situation</td>
<td>Yes</td>
<td>Difficult to see the total impact</td>
<td>None</td>
</tr>
<tr>
<td>Graphical Mapping</td>
<td>Embedding of the model in the current process by representing the adjusted conceptual model as a graphical explanation</td>
<td>Yes</td>
<td>Easy to generate a total overview</td>
<td>Easy to improve the fitted model by validation with stakeholders</td>
</tr>
</tbody>
</table>

Table 5.1 Survey for selecting approaches for reaching a successful fit

Although Graphical Mapping is not an approach derived from literature, but an approach based on the desires of the AAB, Graphical Mapping is the best-fitted approach according the aforementioned selection criteria of Table 5.1. The Delta Analysis approach aims to conduct a gap analysis of the current situation and the new situation. In line with the analysis of design criteria in Section 4.1, reuse activities are not explicitly represented in the formal Requirements Specification process of the AAB, which is shown in the next section in Figure 5.2. Further, it is difficult to identify the total impact of the conceptual model by using a Delta Analysis, since the Delta Analysis approach does not use any methods for creating a high-level understanding (Van Aken, 2002).

5.1.2 Graphical Mapping

The conceptual model of Figure 4.11 is still generic by being suitable for multiple environments. Graphical Mapping, selected as the best-fitted approach, will fit the conceptual model of Figure 4.11 to the current Requirements Specification process at the AAB by adjusting...
each building block from Section 4.6 to the AAB. At the end of this section, the adjusted building blocks form the new conceptual model.

Building block 1: A Knowledge Management Model
This building block describes on the highest aggregation level how the total knowledge management process looks like. The first aspect describes the two iterative processes in which NFRs are defined. The second aspect elaborates on the different ways the repository in the conceptual model is updated.

For each BU at the AAB, the Requirements Specification process comprises two iterative processes, as depicted in Figure 5.2. During iterative process 1, high-level business needs are defined, which define the scope of the system to build. One particular type of high-level business requirements forms the high-level NFR, which describe the high-level demands the different NFR types should meet. During iterative process 2, high-level NFRs are decomposed into the actual subset of NFRs in order to be finally captured in a Requirements Specification document. An example of this document is provided in Appendix VIII. The actual process in which the actual NFRs are set is highlighted in Figure 5.2 by a blue circle. In this way, iterative process 2 is the process in which reuse of NFRs actually takes place. As depicted in Figure 5.2, the Requirements Specification process comprises five stages (i.e. elicitation, analysis and negotiation, representation, validation and change management). This last step, in the center of iterative process 2, manages the changes during the specification of requirements. Building block 2 elaborates on this aspect.

Figure 5.2 Iterative NFR Specification process (adapted from ABN AMRO Group, 2007)

The repository of reusable NFRs can be updated in two ways. The first way occurs by the Requirements Specification process itself, as represented in Figure 4.8. Validated product NFRs are abstracted from the Requirements Specification document and added to the repository. Secondly, the repository can be updated with NFRs outside the project containing process or external NFRs. Examples are: Legal, Audit and Compliance NFRs, Interoperability constraints between the AAB and its vendors and NFRs based on Service Level Agreements (SLAs). In this way, not only already discovered NFRs are stored in the repository, also NFRs that can be used in future are represented. A list of all the different NFR types is provided in appendix IX.

Building Block 2: A Specification Model for NFRs
This part discusses in more detail the process for specifying NFRs at the AAB.
One of the elements of the constructed specification model for NFRs is that it uses a goal-oriented approach for identifying conflicting NFRs and conflicts between FRs and NFRs. At the AAB, different types of requirements (i.e. user, interface, functional and non-functional requirements) are defined during iterative process 2. However, the AAB does not execute any form of a goal-oriented approach to identify interdependencies between NFRs and other requirements types. According to Chung et al. (1995) a goal-oriented approach can only be executed by tooling, like the NFR assistant presented by Chung et al. (1995). The existing CARE tool Caliber RM that is used at BUNL and BULA, can make traces between different NFRs, although this tool cannot detect conflicting NFRs.

The use of a goal-oriented approach can only be realized when NFR reuse has reached a higher maturity level (Chung et al., 2000). This can be established by the use of a unified terminology of the different NFR types that are currently used throughout the bank and by decomposing NFRs in structured way. At this moment, this is not the case for the AAB. The conceptual model to be developed represents the ideal situation for the AAB. Although this situation is not realistic to achieve in the short run due to the aforementioned arguments, the AAB can manage NFRs in a goal-oriented way, when a higher maturity level is reached regarding NFR reuse.

Further, the adjusted model will not consider the interrelationship of NFRs with FRs. At this moment, it is hardly possible to identify the interrelationship between NFRs, so the interdependence between FRs and NFRs is even more difficult.

Building Block 3: Developing high-quality NFRs
The last building block discusses the development of high-quality NFRs during the early stages of the Requirements Specification process.

In Chapter 4 the SQFD model was discussed how high-quality NFRs could be generated by the decomposition of high-level NFRs into detailed NFRs and by using correlation scores. Discussion with stakeholders revealed that some steps in the SQFD are already used in the current way of working of the AAB. Decomposition of high-level NFRs into detailed NFRs takes place at this moment for the AAB. However, NFR lists that could serve as a sample for developing NFRs are often lacking during the specification of NFRs. For the AAB, the decomposition of high-level NFRs in detailed NFRs takes place during the NFR elicitation phase.

Further, the step for prioritizing requirements that is represented in the SQFD model is also used for developing NFRs at the AAB. This has been described in Appendix V, in which prioritization takes place at BUNL by the use of the MoSCoW rules. External imposed NFRs are always critical and have to be included in the system at any time. These NFRs constrain the capacity of the technical infrastructure or impose governmental regulations. On the other hand, product NFRs are not always critical and can be subject to discussion and negotiation. For the AAB, the prioritization of NFRs occurs at this moment during the analysis and negotiation phase.

In conclusion, the result of building block 3 is in the same way represented in the conceptual model, as in Figure 4.10. The decomposition of high-level NFRs in detailed NFRs takes place in the NFR elicitation phase, while the prioritization of NFRs takes place in the Analysis & Negotiation phase.

Constructing the adjusted building blocks in the conceptual model
As a result, Figure 5.3 represents the constructed conceptual model, adjusted to the AAB. The adjustments are already discussed in this section. This part discusses the order of steps to be followed.
The high-level business NFRs, that are the output of the validation step in iterative process 1, form the input for the NFR elicitation phase in iterative process 2, where the actual detailing of NFRs is taking place. The high-level business NFRs define the scope of the system to be built. Subsequently, the NFR specification process starts by the use of reusable components. This process follows the description presented in Section 4.6.

Figure 5.3 Proposal for implementing the conceptual model in the current Requirements Specification Process

Although Figure 5.3 represents the goal-oriented approach of Chung et al. (2000), this step can be realized when a higher maturity level has been reached regarding the reuse of NFRs during the Requirements Specification process. The iterative NFR validation steps result eventually in the constructed Requirements Specification document, since the NFRs are not captured in a goal-oriented way. Subsequently, the validated Requirements Specification document forms the input for the reusable repository. The quality of the content in the repository is maintained in two ways. First, the repository is improved by abstracting and refining validated NFRs from already identified Requirements Specification documents. In this ways, NFRs are designed by reuse, according to Section 4.3.4. Secondly, NFRs are designed for reuse by improving the repository by adding NFRs that are external imposed.
5.2 Realizing the designed model at ABN AMRO

This section discusses the realization of the conceptual model, which is depicted in Figure 5.3. First, an approach is selected to operationalize the model, followed by describing how this approach works. Subsequently, an analysis takes place how the approach should be implemented and how potential resistance should be handled.

5.2.1 Selecting an approach

The second step in Figure 5.1 reflects the selection of a method for realizing the fitted conceptual model in the organization. Before discussing the different approaches for a successful realization of the designed model, first criteria are established to evaluate the applicability of these methods. These criteria are derived from the desires of the stakeholders at the AAB in order to fit the constructed conceptual model in the Requirements Specification process, together with methods provided in literature (Van Aken, 2002).

A successful implementation has to embrace the practical consequences (WHAT and WHO) for this model in a real-life situation. Employees have to know exactly which (critical) activities have to be executed by whom, so a thorough estimation can be made what the impact of this model is for the current situation. When the critical challenges to overcome are identified, the likelihood for meeting unexpected increases in project costs during real-life implementation is reduced. Further, the selected approach has to deliver a guideline to implement NFR reuse in the organization. After it has been made clear to employees what has to be done, it is important that employees keep using this approach. Next, the selected solution has to rise as little resistance as possible by not forcing to change their current way of working immediately. Finally, the costs of the selected approach have to be as minimal as possible. This has resulted in the following selection criteria.

- The selected approach has to make clear WHAT has to be done;
- The selected approach has to make clear WHO has to execute this;
- The approach has to guide users;
- The approach has to create minimal resistance by having a fit to the current situation;
- The approach has to cost as little as possible.

However, several obstacles were faced for fully realize NFR reuse in the Requirements Specification process and are listed below.

- NFR reuse covers several research fields, which makes it impossible in the given time to develop a profound implementation for the full conceptual model of Figure 5.3.
- Toval et al. (2002) state that tools are necessary both for managing the requirements and for supporting reuse.
- Validating the conceptual model of Figure 5.3 with stakeholders revealed that the crucial element in realizing the constructed conceptual model in the current process is the reusable repository. This reusable repository enables the reuse of NFRs. Without the repository, the consequences for the other steps in the model are difficult to estimate.

For these reasons, this research implements only partially the conceptual model of Figure 5.3 by realizing the reusable repository. Although the implementation occurs partially, the objective for realizing NFR reuse does not change, since the reusable repository is the enabler of NFR.
reuse, according to the aforementioned arguments. Table 5.2 summarizes a set of approaches for realizing this reusable repository. Some approaches (Sommerville, 1992) are derived from literature, while other approaches originate from stakeholders in the organization.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Description</th>
<th>WHAT</th>
<th>WHO</th>
<th>Guideline</th>
<th>Resistance</th>
<th>Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building a new NFR tool</td>
<td>The building of a documentary database that imposes stakeholders to reuse requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Low</td>
<td>High costs for developing a tool</td>
</tr>
<tr>
<td>(Sommerville, 1992)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Embedding NFR reuse in Requirements Management tool</td>
<td>BU's should use globally the same Requirements Management tool so reuse of NFRs can take place by the means of this tool</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>High; every BU wants to use their own tool</td>
<td>Low</td>
</tr>
<tr>
<td>Gradual Introducing NFR reuse by a checklist</td>
<td>Development of a checklist to gradually introduce NFR reuse in the specification process.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 5.2 Survey of approaches for partial realization of the conceptual model

According to Sommerville (1992), the easiest way to provide reuse is by means of a documentary database, so that the requirements could be searched by key words and parameterized, and exclusive requirements could be easily implemented. However, the costs are relative high to build such a tool.

NFR reuse can also be embedded in a currently used Requirements Management tool Caliber RM. Although the costs are relatively low, since the AAB has paid the license for using Caliber RM in the whole organization, it will be a time-consuming task that arouses much resistance. In fact, BUTB uses not Caliber RM as Requirements Management tool (i.e. Requisite Pro), while other BUs do not see the need for using a Requirements Management tool at all due its complicated use.

However, gradually introducing NFR reuse by a comprehensive NFR checklist meets the criteria most. This checklist can be considered as the reusable repository 'on paper' of Figure 5.3. This checklist covers all the different high-level NFRs together with its derived detailed NFRs in a checklist that is used throughout the bank and that can be reused throughout the bank. The development of the NFR checklist is explained in Section 5.2.2. When the NFR checklist has evolved into a success, a real-life NFR repository can be developed. Besides the selection criteria of Table 5.2, the following arguments underlie the development of a NFR checklist.

- The NFR checklist enables the realization of building block 3 of the conceptual model, discussed in Section 5.1.2, by enabling a structure for decomposing high-level NFRs into detailed NFRs during the NFR elicitation phase. The NFR checklist covers a representative subset of high-level NFRs, in order to align the use of the same NFR typologies in BUs and between BUs. Furthermore, the NFR checklist covers a representative subset of detailed NFRs, which serves as exemplary samples for decomposing high-level NFRs into detailed NFRs.

- At this moment no explicit process for NFR reuse takes place at the AAB. This makes it logical to start NFR reuse at the lowest aggregation level of the conceptual model of Figure 5.3, by realizing building block 3. As a consequence, other aspects of the conceptual model are enabled when a higher maturity level regarding NFR reuse is reached, for example the use of a goal-oriented approach, proposed by Chung et al. (2000). This is in line with the approach of Table 5.2 for gradually introducing NFR reuse.

- Enabling NFR reuse for all the BUs at the AAB should succeed gradually. According to Requirements Engineers in the organization, the development of a tool does not solve
the underlying problems for not reusing NFR knowledge. The costs for creating a checklist are relatively low and the underlying basic obstacles for realizing NFR reuse, like the (mis)alignment of NFR typologies between BUs, can be faced by the means of a checklist. When BUs are using the same typologies of high-level NFRs, reuse can actually take place.

- In parallel with this research, a global initiative has started by the Requirements Engineering and Management (REM) Forum. This forum consists of fifteen requirements engineers from all the BUs of the AAB and aims to align Requirements Engineering activities by striving to use one global approach across the AAB. One of their approaches concerns the handling and reuse of NFRs. Discussing this approach with responsible stakeholders, has resulted that interested parties are highly interested in the development of a checklist.

- Due to time constraints, the construction of a comprehensive NFR checklist is realizable, rather than the creation of a tool that enhances NFR reuse, like the NFR Assistant of Chung et al. (1995).

- When this checklist will become a success, a reusable repository could be created. The creation of tool according to Sommerville (1992) is considered in this research as the second best initiative that is listed in Table 5.2, since the Requirements Management tool Caliber RM is not supporting parameterized requirements, so that the NFRs could not be searched by key words.

Section 5.2.2 describes the development of the NFR checklist in more detail. Subsequently, the form of the reusable repository is described briefly in Section 5.2.3 and more profoundly in Appendix XII. The remainder of this section will elaborate on the NFR checklist. Section 5.2.4 mentions the responsible stakeholders, Section 5.2.5 provides a resistance analysis, Section 5.2.6 provides a guideline for implementation and Section 5.2.7 discusses the involved costs.

5.2.2 Development of the NFR Comprehensive Checklist

Generally, the BUs of the AAB work independently in setting NFRs, which has resulted in a uncoordinated setting NFRs across BUs. The developed NFR checklist provides a subset of 31 types of high-level NFRs. For each high-level NFR a subset of detailed NFRs is abstracted from Requirements Specification Documents that have been used by several BUs in particular (i.e. BUNL, BUTB and BUNA). These detailed NFRs are abstracted in such a generic way that they are suitable to use by multiple BUs. After validation with experts, this NFR checklist is distributed to all the BUs of the AAB. Then, BUs can complete this checklist with NFRs that are applicable for their specific BU. In this way, BUs are becoming familiar with how NFRs are handled across the bank and will pass over this knowledge for setting NFRs to the particular BU.

The remainder of this subsection discusses the design constraints for constructing the comprehensive checklist:

1. The NFRs in the checklist have to be represented in a consistent and unambiguous way. This means that the use of ambiguous terms as “correctly” and “high-available” should be avoided.

2. The checklist has to deliver NFRs that are useful for all the BUs of the AAB. For this reason, the checklist should not contain NFRs that are based on BU specific Service Level Agreements (SLAs), and measurements.
3. The checklist should decompose as much as possible high-level NFRs. At this moment every BU has its own classification of NFRs, so reuse cross-BU reuse of NFRs is hardly possible. By classifying all the NFR types that are used across BUs in a structured way, ambiguities in NFR types are prevented and the checklist will be as extensive as possible.

4. The NFRs that are listed in the checklist have to contain a priority level (i.e. critical, high, medium and low). In this way, the NFR checklist can be used actively during the NFR analysis & negotiation phase.

5. The NFR checklist should be used as a dynamic working document in which refinements can take place.

This checklist is represented in the "Non-Functional Requirements Comprehensive Checklist" document. This document is provided in the last part of the appendices.

5.2.3 Logical Data Model

Section 5.2.1 stated that in case the NFR checklist has evolved into a success, a real-life repository, in the form of a tool, is developed that enables NFR reuse. This research will not elaborate on the potential challenges to overcome in order to realize this tool, but will provide the logical data model of this tool.

The objective of this tool is to construct a repository where all the BUs of the AAB can find and place their NFRs. This tool enables the reuse of NFRs across BUs together with storing the NFRs for the BUs in particular. This tool is updated by (1) abstracting validated NFRs from projects into the repository and by (2) adding external imposed NFRs to the reusable repository.

This method of first implementing a comprehensive NFR checklist followed by a real tool when the initiative has become a success, is line with the argument for choosing for a gradual introduction as represented in Table 5.2. The logical data model is elaborated in Appendix XII.

5.2.4 Responsible stakeholders

Appendix IX provides an overview of the different stakeholders for setting NFRs in the current situation. The realization of the conceptual model of Figure 5.3 occurs in two ways: the first way occurs by the checklist. When this has become a success, a real repository can be constructed, as described in Appendix XII.

The involved stakeholders for implementing the comprehensive checklist are described in section 5.2.6, together with the plan how to implement the model. The involved stakeholders regarding the real reusable repository are listed together with the design of the UML model in Appendix XII.

5.2.5 Resistance Analysis

Besides the technical – economical argument, there also political-social arguments aspects that have to be taken in consideration for implementation (Van Aken, 2002). This subsection provides an analysis of potential sources that could lead to resistance for using the NFR checklist in the Requirements Specification process of all the BUs at the AAB, and in what way this resistance can be overcome.

- BUs have a different maturity level in specifying requirements. There are "experienced" BUs like BUNL and BUNA versus "inexperienced" BUs like BUAM, BUPC and BUTB. This makes it difficult to use one approach.
Especially the "inexperienced" BUs are likely to profit from this NFR checklist to reuse NFRs, since there is no similar approach available at this moment for these BUs. For this reason, it will be very likely not to bear any resistance from these BUs to implement the NFR checklist, unless the information is developed according to the criteria in section 5.2.2. At first sight, the NFR checklist adds less value to the "experienced" BUs like BUNL, BUNA and BULA, than for the "inexperienced" BUs. BUNL and BULA use the Requirements Management tool Caliber RM in which NFRs are stored and captured, described in appendix XI. However, the NFR checklist forms a basis of discussion for these BUs by adjusting existing NFR typologies to typologies that are used by the NFR checklist. This can be established by exchanging best practices between BUs.

- BUs are using different typologies for the same NFR (i.e. Speed NFR at BUNL and Performance NFR at BUNA and BUTB) or BUs use typologies that are only used in their particular BU (i.e. Conversion NFR at BUNL). This can be a problem for capturing NFRs in the NFR checklist.

The NFR checklist aims to align the use of NFRs between the BUs by decomposing as much NFR typologies as there could be retrieved. In this way the most exhaustive overview of NFR typologies is generated that enhances a structured approach for dealing with NFRs. Furthermore, BUs are likely to accept more the use of the NFR checklist, when they recognize their own NFR typologies in the checklist. After the NFR checklist has been validated (see section 5.2.6), BUs can add their BU-specific NFRs to this checklist. In this way, the NFR checklist does not rigorously change the current way of NFR setting. Instead it forms a basis for (a discussion about) alignment in a global use and reuse of NFRs.

5.2.6 Guideline for implementation: project plan

This subsection provides the plan how the checklist will be operationalized for the BUs. At this moment, the checklist is the first initiative to coordinate the (re) use of NFRs in a global ABN AMRO initiative. This section defines the project plan from the first draft of the checklist to the acceptance by the BUs. This project plan is constructed in cooperation with two stakeholders in the REM forum.

According to the demands of the stakeholders in the REM forum, the NFR checklist has to be completed in the shortest possible time. Further, the NFR checklist has to contain a high quality content by being totally validated by the REM stakeholders. When the checklist has been validated, each particular BU is able to use it and fill it up with their BU related NFRs, like SLAs.

This approach, together with its assigned stakeholders, is represented in Figure 5.4. The remainder of this subsection describes in more detail the process of Figure 5.4, and the estimated period before the NFR checklist can be used.

Process before achieving completeness

First, a draft of the checklist has been made in this research by selecting generic components in Requirements Specification documents of the different BUs of the AAB and by capturing them in the NFR checklist. A Requirements Engineer with substantial background knowledge in setting NFRs executes the second step. This person will review the checklist on aspects that are not fully covered in the
initial draft and will refine the checklist in a way that it will be appropriate to send to all the stakeholders of the REM forum.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Assigned stakeholder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft Checklist</td>
<td>Benno Waterman</td>
</tr>
<tr>
<td>Review Document</td>
<td>Bert Dubbelman</td>
</tr>
<tr>
<td>Check in REM forum</td>
<td>Chris de Jong</td>
</tr>
<tr>
<td>Processing changes</td>
<td>all members of the REM forum</td>
</tr>
<tr>
<td>Check by REM initiators</td>
<td>Chris de Jong</td>
</tr>
<tr>
<td></td>
<td>Peter Penders</td>
</tr>
<tr>
<td></td>
<td>Ramesh Devare</td>
</tr>
</tbody>
</table>

Figure 5.4 plan for implementing the comprehensive checklist

In addition, the checklist is sent to the Requirements Engineers of the particular BUs, who are a representative in the REM forum. Subsequently, the checklist will be extensively discussed with all the members of the REM forum by the means of a conference call. BUs give their inputs to the proposed checklist in order to deliver a checklist as complete and unambiguous as possible. This way of discussing the checklist in a plenary way will (1) work as a deadline for each BU to complete their feedback and (2) shorten the total processing time.

When all the feedback is handled, the responsible Requirements Engineer for delivering the NFR checklist will process the feedback. The REM forum appoints one representative who will be responsible for this task.

When the changes are processed, the checklist is sent to the REM initiators for a final check. Due to their extensive knowledge and experience in the total REM field, these persons are likely to execute this task.

Processing time
In respect to time, the checklist needs to be discussed extensively during the monthly REM conference call in the end of July 2007. Before this date, the stakeholders have to carefully read the NFR checklist in order to give well-considered feedback. After validation by REM members, a responsible Requirements Engineer needs to process the results, followed by a final check by the REM initiators. The time for launching the initiative so the checklist can be used in the beginning of Quarter 3 (i.e. end of August 2007).

5.2.7 Costs
The comprehensive checklist can be summarized as an approachable way for realizing a global coordinated setting of NFRs. Besides, the method of gradual introduction by the use of a NFR checklist incurs relatively low costs and is likely to deliver high results. The next section conducts a confirmatory case study in which it is tried to estimate the potential cost reductions of using the NFR checklist.
5.3 Evaluating the approach for realizing the conceptual model

According to Figure 5.1, the last step of a successful implementation is to evaluate whether the selected approach suits the objective for which it was selected. For this reason, this section aims to evaluate the use of the NFR checklist, which has been described in Section 5.2. First, selection criteria are established, which has resulted in the conduction of a confirmatory case study in Section 5.3.1. A representative case is selected in Section 5.3.2, followed by an approach how to conduct the case study in Section 5.3.3. The actual conduction leads to a set of organizational results (Section 5.3.4) and financial results (Section 5.3.5) in order to estimate the impact of the NFR checklist.

5.3.1 Selecting an approach

Before discussing the different approaches for a successful evaluation, first criteria have to be established to evaluate the applicability of these methods.

- The selected approach has to exemplify how reuse of NFRs is taking place
- The approach has to test how it suits the criteria for which it was selected
- The approach has to gain insight in the potential benefits

These criteria are derived from the desires of the stakeholders at the AAB in order to fit the constructed conceptual model in the Requirements Specification process. Exemplifying how reuse of NFR can take place leads to a profound understanding of the practical consequences of the conceptual. This will enhance its relevance for the AAB. Secondly, testing the quality of the approach reveals aspects (i.e. problems) that were not covered beforehand and could prevent emerging problems for the AAB during real-life implementation. Finally, insight in the potential benefits is created, for example by cost savings, which is important for the AAB, since this model will only be used in a real-life situation when substantial cost reductions can be obtained for the AAB.

Table 5.3 summarizes two evaluation approaches according to the criteria that are mentioned above.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Description</th>
<th>Exemplification</th>
<th>Test quality of the approach</th>
<th>Gain insight in benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus Group Discussion</td>
<td>Group of people brought together to participate in the discussion about an area of interest</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Confirmatory Case Study</td>
<td>A research strategy that investigates a phenomenon within its real-life context</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 5.3 Survey of existing approaches for evaluating implementation approaches

The only approach that fits to three selection criteria from Table 5.3 is the confirmatory case study, since the focus group discussion lacks the practical insight how this approach would suit to a real-life situation.

In order to exemplify the execution of the comprehensive NFR checklist in practice, a case study example is presented in this section. The case to be studied should satisfy some criteria, in order to be a representative reflection of a project executed at the AAB.

- The project should contain a substantial subset of NFRs
The project is executed in a multi-vendor environment

The project team should be willing to cooperate in the application of the constructed conceptual model.

A project satisfying these criteria is suitable for both serving as an example to the organization as well as for preliminary evaluation. This resulted in the selection of the BIB2 project, which is described in the next section.

5.3.2 Case Description

Since the year 2000 the AAB has an Internet Banking (IB) platform in order to realize banking by the Internet. The Internet Banking application has become a major Business Asset for the AAB by offering banking activities to customers like online brokerage, a digital vault and electronic transactions. The customer interface of this application is represented in Figure 5.5. After the launch of this IB platform, continuation projects followed by starting the Best Internet Bank (BIB) projects. The BIB project has delivered a rebuilt Internet Banking Application with the functionality of the existing Internet Applications. After completion of BIB1 with some specific additional functional enhancements, the BIB2 application is based on both new functional and new technical architecture. The objective of the BIB2 project is to decompose the existing application in manageable and maintainable fragments.

The objective of the NFRs document in the BIB2 project is to gather all NFRs that are relevant for the BIB project and the projects that will use the IB platform.

5.3.3 Approach

The approach for the confirmatory case study is based on how the constructed NFR checklist would have influenced the way NFRs are set in the BIB2 project. Due to constraints in time it
was not possible to conduct a case study on the complete course of the BIB2 project. The case study was executed by means of a semi-structured interview, based on the existing NFR document of the BIB2 project. The underlying reason for executing the case study by using semi-structured interviews is that the case study concerns purely qualitative data, which makes interviewing a useful method. The aspects that were covered in the case study concerned the method how NFRs were collected during the BIB2 project and the potential ways the NFR checklist could improve the current way of working.

5.3.4 Results
The objective of the case study approach is threefold: exemplification, gaining insight in the potential benefits of the solution and testing the quality of the approach, as represented in Table 5.3. While the first objective can be considered as a characteristic feature of the case study approach, the other two objectives are considered in the BIB2 project.

Gaining insight in the potential benefits of the approach
This part will list the potential benefits of the NFR checklist during the executing of the BIB 2 project.

- During the construction of the NFRs in the BIB2 project, the NFRs were gathered in a relatively unstructured way. In iterative sessions with stakeholders, NFRs were gathered and abstracted from vendor NFRs and existing documentation based on the Business Area Definition (BAD) and the System Architecture Definition (SAD). By using the checklist, the NFR elicitation process could be handled on a more structured and standard way and would result in a reduction of the total processing time.

- Since the NFR checklist enhances a standardized way of NFR setting, discussions and negotiations for capturing the NFRs can effectively be shut down, which reduces the processing time.

- During the BIB2 project, stakeholders experienced many problems assigning NFRs to high-level business NFRs. At this point, there was no clear example that guided the setting and classification of NFRs. By using a standardized typology as provided in the NFR checklist, not only the processing time of the Requirements Specification process could be decreased, also related parties like test vendors could take advantage of this standardized way of NFR setting. In fact, test vendors have to transfer the NFRs in the Requirements Specification Document into a Test Document.

- In addition to the previous listed issue, all NFRs are eventually technical NFRs. In fact, after decomposition it has made clear that lower level NFRs enable the realization of the higher level NFRs. This causes some problems how to categorize NFRs after decomposition. For example, an availability NFR like 24/7 365 days a year availability is enabled by Degradation, Resilience and Capacity NFRs.

- NFRs are often considered as less tangible than FRs. For this reason the setting of NFRs is often postponed. The NFR checklist provides a sufficient amount of examples to make NFRs more tangible, which might avoid that NFRs are retrofitted late in the Requirements Specification process.

- The use of a NFR checklist will contribute the most to projects that are developed from scratch, rather than for continuation projects. In fact, continuation projects, like maintenance projects, rely more on the NFR document of the previous project. On the other hand, for these maintenance projects it will be increasingly important to know as
fast as possible the NFRs since these projects are often concerned with infrastructure, which is mainly a matter of NFRs.

- A potential risk in using the NFR checklist is that users think that the NFR checklist is exhaustive. However, the NFR checklist is a working document that aims to cover as much as NFRs as possible. As a consequence, users might forget some NFRs that are not covered in the checklist.

**Testing the quality of the approach**

After gaining insight in the potential contribution of the NFR checklist for the BIB2 project, the quality of the approach can be considered. As mentioned in the previous subsection, the added value of the NFR checklist will be higher in projects developed from scratch than for maintenance projects. In these situations it would make more sense to develop the new NFR list based on the previous one. However, the potential benefit of the checklist lies in its relatively easily acceptable use. For example, the Requirements Engineers of the BIB2 project did not make use of the Requirements Management tool Caliber RM, since the engineers were not familiar with this tool and seemed to be too complicated for use by active members of a particular project. The second potential benefit of the NFR checklist lies in the intended structure of the document, which guidelines the user in capturing NFRs by forming a sample. This would have been extremely beneficial in the BIB2 project, since the BIB2 project did not make use of a standard format for capturing NFRs. The NFR checklist is constructed simple, but not simplistic.

Although the confirmatory case study example has evaluated the NFR checklist by selecting a representative case, this cannot be regarded as the validation of the NFR checklist. Chapter 6 elaborates further on this topic, in which the conclusions and recommendations from this thesis are provided.

**5.3.5 Financial Results**

Next to identifying the potential organizational benefits for using the NFR checklist in the BIB 2 project, this subsection aims to identify the potential financial benefit. However, identification of the financial impact of NFRs is a difficult task, since the time spent for the capturing NFRs during software development is hardly recorded. This has also been the case for the BIB 2 project, where a proper time recording was lacking. For this reason, a similar project was selected for which the amount of effort for capturing NFRs had been recorded: the OCP SEPA project. This project aims to embed the payment standard SEPA in the current software architecture of the bank and can be considered as a common large project in the AAB. The amount of effort for capturing the NFRs in the Business Study (see appendix V BUNL) was set to 40 hours to a total elapsed time in the Business Study of 1300 hours. As a consequence, for common large projects the capturing of NFRs covers three per cent of the Business Study. In addition the following assumptions can be drawn:

- According to the interviewee, the use of the NFR checklist can realize a NFR reuse of 50%. However, comparison of the BIB2 NFR list and the constructed NFR checklist revealed that 30% of the NFRs of the BIB2 NFRs are listed in the checklist.

- These numbers only consider one phase in the Requirements Specification process (i.e. the Business Study). However, the setting of NFRs is not limited to only one particular phase, but covers multiple phases in the software development process (see appendix IX).
The effect of using the NFR checklist has also consequences to other parties involved during software development. A standardized way of capturing and representing NFRs can positively affect the way test vendors are constructing test documents. Further, the use of the NFRs checklist reduces the likelihood for forgetting NFRs. Next, the use of the NFR checklist can result in standardization on a system level, since a standardization on NFR level results in a standardization of system level. In the ideal situation, besides the reuse of NFRs, also complete operating platforms are likely to be reused. This is AAB’s ideal scenario for dealing with their Symphony vendors.

Taking the premise that the positive and the negative effects that are listed above neutralize each other, reusing NFRs by the use of the NFR checklist in the short run is likely to result in a cost decrease of 1 to 1.5% of the total software development costs.

5.4 Conclusions

The implementation of the constructed conceptual model has been described in this chapter. First, the model has been suited to the Requirements Specification process of the AAB by adjusting the three building blocks to the AAB. In addition, the model is suited to practical application by first trying to find a method to implement the total conceptual model fully. However, due to time constraints and due to a misfit between the possible approaches and the selection criteria, the conceptual model is partially implemented by one aspect of the conceptual model. As a consequence, the reusable repository of Figure 5.3 is gradually implemented by the means of a NFR checklist. This checklist contains an extensive subset of NFRs that can be used by all the BUs of the AAB. In this way, this checklist aims to align the NFR typologies used throughout the bank by forming a "repository on paper". Further, the checklist aims to structure the decomposition of high-level NFRs in detailed NFRs, since the NFRs in the checklist will have a sample function. The NFR checklist can be considered as the operationalization of building block 3, since this building block aims to increase the quality of setting NFRs in the early phases of the Requirements Specification process by using a structured method for NFR decomposition.

When the NFR checklist has been discussed in the REM forum, the particular BUs have to use this checklist without forcing these BUs in changing their current way of specifying requirements immediately. In this way, the implementation of NFR reuse is established gradually, which increases the likelihood for realizing a uniform use of NFR typologies throughout the bank and a standard and structured way of NFR decomposition, according to building block 3 of Figure 5.3. When the use of the NFR checklist has evolved into a success, a real-life repository is built, for which the UML design is presented in Appendix XII. Furthermore, a project plan is provided how to implement the NFR checklist in cooperation with the REM forum at the AAB.

A case study example in Section 5.3 pointed out what the potential benefits would have been when the NFR checklist was used in the BIB2 project. This confirmatory case study example has resulted in a set of organizational benefits and an estimate of the financial benefits of using the NFR checklist.
6 Conclusions and Recommendations

Non-functional Requirements (NFRs) are becoming an increasingly important issue for the AAB. Not only because their characteristic feature of being general applicable, but also for the relatively high effort for constructing NFRs due to their dispersed location in the organization. This thesis has described the research that has been carried out addressing the effect of NFR reuse during the requirements specification process.

This research started with a problem stating that the way requirements are specified in the new IT environment resulting from the Harvest & Symphony initiative, has to be improved. An exploratory research was performed covering both theory and practice. On the basis of exploratory literature and by the use of case studies, the initial problem statement was decomposed into a comprehensive problem-breakdown diagram. This diagram covered all problem fields in the AAB regarding the specification and management of requirements. Based on a cross-case analysis of the aforementioned diagram, the research objective evolved to provide a model how reuse of NFRs can be embedded in the traditional lifecycle for specifying requirements (Chapter 2). It was found that at the moment of research the AAB did not have a formal process in which reuse of NFRs could take place during the Requirements Specification process. Furthermore, it was concluded that no model could be derived from literature that meets the criteria for fulfilling the objective of the research. Therefore, a conceptual model was constructed based on the best practices of the earlier imposed design criteria, which had been used for trying to select a model that fulfilled the objective of the research. The approach of using three complementary research fields has resulted in the eventually constructed conceptual model and reflects the ideal situation towards NFR reuse during the requirements specification based on literature (Chapter 4). Further, the constructed model was tailored to the situation of the AAB, suited for practical application, and decided to gradually introduce the model by the means of a comprehensive NFRs checklist. Eventually a case study example was described of what the affect should be for using this checklist in a real-life situation (Chapter 5).

This chapter ends this thesis by presenting conclusions and recommendations. First, Section 6.1 provides the main conclusions drawn from the research. Next, recommendations towards the AAB are formulated in Section 6.2. Finally, Section 6.3 provides direction for further research.

6.1 Conclusions

A number of conclusions can be drawn upon the findings from this research project. First the main conclusions from the orientating case studies are provided. Subsequently, conclusions derived from the development and implementation of the constructed conceptual model are provided, according to the Research Questions that are drawn in Chapter 3.

The following conclusions are drawn upon the exploratory phase of this thesis:

- With respect to the current situation of the AAB, it can be concluded that every BU has its own software development methodology. As a consequence BUs show to have little coherence between each other.

- There is a striking variation in maturity levels of the different BUs of the AAB. As a consequence, it is difficult to develop one approach for all the BUs of the AAB. At the other hand, differences in maturity levels are likely to enhance the possibility of reusing knowledge between high-maturity BUs and BUs with a relative low maturity level.
Regarding the development and implementation of the conceptual model in this thesis, the following conclusions are formulated.

- **Q1: Which method is used for designing a conceptual model where reuse of NFRs is embedded in the Requirements Specification process?**

  In order to construct a conceptual model in which reuse of NFRs is embedded in the Requirements Specification process, first the current situation of the AAB towards NFR reuse was identified. This resulted in setting up three design criteria, which the designed model should meet. However on the basis of these design criteria, no model could be directly derived from literature. As a consequence, the design criteria were transformed in three complementary research fields. Subsequently, for each research field, design requirements were set up to construct three building blocks that formed after merging the eventual constructed conceptual model.

- **Q2: What does the constructed conceptual model derived from literature look like?**

  The eventually constructed conceptual model is the result of the methodology, described in Research Question 1. The constructed conceptual model is based on the following best-practice models, derived from the imposed design requirements.

  1. Knowledge Management Model. This model is a combination of the Knowledge Value Chain of Weggerman (1997), in which the Knowledge Sharing part of the model is substituted by the Software Reuse Lifecycle (Zhang et al., 2001).

  2. Specification Model for NFRs. This model is based on the ISO 9001:2000 model of Bamford et al. (2001), in which the Manage step is substituted by the goal-oriented approach of Chung et al. (2000).

  3. Quality Model. For assuring the quality of NFRs in the early phases of the Requirements Specification process, the Software Quality Function Deployment (SQFD) framework of Haag et al. (1996) has been selected.

  The constructed conceptual model is depicted in Figure 4.10 and reflects the ideal situation how reuse of NFRs can be embedded in the Requirements Specification process.

- **Q3: How should the designed model be adjusted to the current situation of ABN AMRO?**

  The conceptual model derived from literature is constructed by merging three building blocks of the aforementioned research fields. In the same way, the model is adjusted to the AAB by adjusting every building block to the situation of the AAB.

  1. Two different stages of the Requirements Specification process at the AAB can be identified: setting of high-level NFRs and the setting of detailed NFRs. Furthermore, the repository can be updated in two ways: by abstracting product NFRs from Requirements Specification documents in the repository and by adding external imposed NFRs to the repository.

  2. Since no formal process is described at the AAB for NFR reuse during the Requirements Specification process, a goal-oriented approach is extremely difficult to establish in order to identify conflicting NFRs proactively. In case other aspects are covered (i.e. unified typologies for NFRs and a structured way of NFR decomposition) and NFR reuse has reached a higher maturity level at the AAB, a goal-oriented approach for managing NFRs is realized.
3. The creation of high-quality NFRs by decomposing high-level NFRs into detailed NFRs is not explicitly taking place for NFRs. Although NFRs are prioritized, there is no standard way how NFR decomposition takes place.

The conceptual model, adjusted to the NFR specification process of the AAB is depicted in Figure 5.3.

- **Q4**: *In what way can the designed model be operationalized to suit practical application for ABN AMRO?*

Validation of the adjusted model of Research Question 4 revealed that the crucial element in operationalizing the conceptual model of Figure 5.3 is the reusable repository. This research selected to gradually introduce this reusable repository by the means of a "Non-Functional Requirements Comprehensive Checklist", which is provided in the last appendix. This checklist aims to align the terminology of NFR types across BUs and forms a "repository on paper" that enables a structured decomposition of high-level NFRs in detailed NFRs, according to building block 3 of the conceptual model. When the use of this NFR checklist has evolved into a success, a real repository is constructed for which the UML design and the consequences regarding the responsible stakeholders are presented in appendix XII.

- **Q5**: *Which further research areas can be designed after this research?*

This question is answered in section 6.3

In short, it can be concluded that the objective of the research has been fulfilled. A conceptual model is provided how reuse of NFRs can be embedded in the Requirements Specification process, followed by an adjustment of this model to the AAB and a proposed strategy to operationalize the model. This research has gained insight in the characteristic features of NFRs and has provided ways to handle them in a structured way.

### 6.2 Recommendations

From the results and the conclusions of the research project, recommendations that address ABN AMRO Global Services IT are included in this section. Both issues that resulted directly from the model presented, and issues that stood out by the way but are worthwhile to capture in the light of this research project, are mentioned here.

- **Define NFRs as early as possible.** Frequently, NFRs are identified in the relatively late stages in the Requirements Specification process. The underlying reason is that NFRs are often considered as intangible in comparison with FRs. However, most NFRs constrain the realization of a system by the means of security, capacity, or compliance. Absence or ambiguities of these NFRs has an enormous impact on the actual realization of the system to be built. For this reason, early identification of these NFRs is key, since this will reduce the likelihood for meeting unexpected problems in the latter phases of the Requirements Specification process.

- **Make as much as NFRs explicit as possible.** The identification of the different NFR types that are used throughout the AAB for constructing the "Non-Functional Requirements Comprehensive Checklist", revealed that some NFR types showed hardly any examples while other types were highly documented. As a consequence, it was not possible to reuse these NFRs since the NFR knowledge was tacit. For this reason,
making as much as tacit knowledge explicit, forms the basis for establishing reuse of NFRs, since solely explicit knowledge can be reused.

- **Use a standard and structured format for defining high-level NFRs in and between BUs.** Comparison of different high-level NFRs (NFR types) that are used in and across BUs, revealed an uncoordinated setting of NFR types. In fact, some BUs used different NFR types for the same phenomenon (Performance and Speed NFR), while other NFR types were only used in particular BUs. For this reason, alignment of all the NFR types that are used inside and between BUs is needed to establish NFR reuse. In fact, reuse of NFRs, whether it takes place inside or between BUs, can only take place when a the same NFR types is used.

- **Use a standard and structured way for decomposing high-level NFRs into detailed NFRs.** NFRs are often developed without using a standard format for decomposing high-level NFRs into detailed NFRs. This has resulted in an uncoordinated subset of NFR types across BUs, which have limited the possibility for reuse and increased the processing time for capturing NFRs. By using a standard structure for decomposing the total set of high-level NFRs, uniformity is created. Uniformity in structuring NFRs reduces the likelihood for forgetting NFRs, decrease the amount of time for gathering NFRs and forms the basis for reuse of NFRs.

- **Realize the long-term effect of using standardized and coordinated NFRs.** Reuse of NFRs can only be established when standardization of NFR types takes place. However, this form of standardization does not only improve the efficiency for developing a Requirements Specification Document, but has also effect on other facets during software development. In fact, when all NFRs are listed in a uniform way, test vendors can easily transfer NFRs from the Requirements Specification Document into a Test Document. On the longer term not only NFRs can be reused but also reuse on system level can be established due to the standardization of its components.

- **Prioritize as much as NFRs as possible.** Although many NFRs can be characterized as critical for the system to be built, which results in automatically including these NFRs in the system, prioritization of non-critical NFRs is fruitful. At this moment, non-critical NFRs are hardly prioritized, which leads to long discussions if the NFR should or should not be included in the Requirements Specification document. For this reason, prioritization turns down the discussion by providing unambiguous prioritization scores.

- **Appoint people with high mandate to initiate the embedding of NFR reuse in the Requirements Specification process.** NFR reuse is realized in two ways: by realizing the preconditions for NFR reuse, like the development of a tool, and by social-political aspects. For this reason, how good a solution might be, support from people with high mandate in the organization is key for successfully embedding NFR reuse during the specification of requirements, and to challenge resistance from people who want to work the same as they are doing now.

- **Appoint responsible people for enabling NFR reuse.** In order to realize that NFR reuse keeps executed, three critical activities should be executed, whether a tool is used as repository or a NFR checklist.
  
  Firstly, key in realizing NFR reuse is abstracting NFRs from validated Requirements Documents and refining these for reuse. Special resources have to be appointed that refine NFRs in specific, unambiguous and generic NFRs, which have to be listed in the repository. Second, special assigned resources have to upgrade the repository with external imposed NFRs in order to establish design for reuse, as discussed in Section 5.3.4. Finally, special assigned resources have to maintain the repository in order to preserve the repository of high-quality NFRs.
- **Evaluate the NFR checklists after a good length of time.** In order to address deficiencies both intrinsic to the NFR checklist as well as induced by the way the NFR checklist applied. As a result, opportunities for NFR reuse improvement may arise.

- **Develop a tool that is user-friendly.** In recent years, many comprehensive tools have developed, that didn’t bring the intended results due to its complicated use. For this reason, when the NFR checklist has evolved into a success, basic tooling should be used consisting of high-quality knowledge. In this way, the tool will be used during the Requirements Specification process due to its simplicity, and can bring the intended results.

- **Use a tool to identify conflicting NFRs.** Decomposing high-level NFRs in detailed NFRs often results in conflicts between NFRs of the same high-level NFR and of other high-level NFRs. In fact, a capacity restriction imposed by one NFR can affect the performance level imposed by another NFR. Together with the iterative character in which NFRs are set, it is key to identify conflicting NFRs as early as possible. At this moment, it is not possible to identify conflicting NFRs proactively. It may be worthwhile to handle conflicts proactively, as recommended by Chung et al. (1995) by using a NFR Assistant.

### 6.3 Further research

This final section presents some directions for further research. As already indicated in Chapter 5, despite the fact that the NFR checklist is applied in one confirmatory case study, the NFR checklist has not been validated yet. Further research should demonstrate that this application for operationalizing the conceptual model, adjusted to the AAB, substantially contributes to NFR reuse.

The research project has been conducted in a single company and therefore it would be invalid to simply generalize its outcome. However, based on structural similarities between companies some forms of generalization can apply. Specifically, the constructed conceptual model may be suitable for other data-intensive organizations in which software development takes place for supporting the business-wide company objectives. It needs further research to enquire the deployment of the conceptual model more widely considering other companies, other industries and other software development methodologies.

As been identified in the Section 6.2, the effect of standardization in NFR types does not only enable the reuse of NFRs but also impact other aspects during software development. For this reason, it might be interesting for further research to determine more precisely what the company wide effect is of standardizing NFR types.

This research aims to gradually introduce NFR reuse by the means of a NFR checklist, followed by a real tool when the checklist seems to work. When the tool has been created, the quality of the tool should be maintained. It might be interesting to look at the possibilities for creating an open source repository and to determine what the impact of ownership is for preserving the quality of the repository.

Finally, it might be interesting to determine more precisely what the impact is on the use of correlation scores between high-level NFR and detailed NFRs, and on the quality of the eventually delivered NFRs. It will be a challenge to determine what the quality of a NFR be made up of and how the quality can be measured.
These last words are intended to provide some reflection on my MSc. graduation project. Looking back on the past ten months I can only conclude that it has been a very instructive time. Of course there were some struggles, especially in the attempts to come up with something meaningful for such a very complex organization. However, the project also brought the joy of the all the efforts I put in this project, with illuminating insights.

Despite ten months is a long period of time, in the end there is always so much more to do. Inevitably, in the beginning much time was spent to familiarize with the business environment and the quest for a feasible scope of the project. For me the setting of a proper objective is one of the key learnings, all the more since it is overlooked so easily. In addition, I wondered about the amount of effort it took for me to convey my thoughts comprehensively, understandably and well-considered on paper (hopefully this succeeded to some extent). All in all, I am glad the project has this result. I am particularly pleased that the objectives of both university and ABN AMRO could be reconciled in this project. I think the combination between academic theories and business practice has been one of the most valuable learning experiences for me.

Although this thesis has approached the problem that is faced conceptually, I do not feel that the outcome of this research is purely theoretical. The development of a NFR checklist helps to get the new approach to work. However, the proposed method is not plug-and-play solution that brings improvements without any further effort. The solution provides guidelines and facilitates a new attitude to take into account during the Requirements Specification process. This implies that NFR reuse is only realized when stakeholders in the organization share the same belief about the importance of NFR reuse.

So, this was it there. The project has been finished, my studies have been completed and a new era in my life will start shortly. Again, I would like to express my gratitude to everybody who contributed in whatever way. The show must go on!

Benno B. Waterman
Bibliography


