



UPPSALA  
UNIVERSITET

UPTEC-ST512033

Examensarbete 30 hp  
2012-08-20

# Effective Internal IT-development at Nordea Portfolio and Advisory Solutions Including Offshoring

---

Gabriella Hammarin



UPPSALA  
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet  
UTH-enheten**

Besöksadress:  
Ångströmlaboratoriet  
Lägerhyddsvägen 1  
Hus 4, Plan 0

Postadress:  
Box 536  
751 21 Uppsala

Telefon:  
018 – 471 30 03

Telefax:  
018 – 471 30 00

Hemsida:  
<http://www.teknat.uu.se/student>

## Abstract

### **Effective Internal IT-development at Nordea Portfolio and Advisory Solutions Including Offshoring**

*Gabriella Hammarin*

Modern organizations within IT-developing needs to be prepared to face challenges that are not necessarily connected to the mere technological aspects of softwares. These challenges might lie within e. g. communication between stakeholders, user involvement, organizational regulations, the need for standards and maintainability of the products. This study is investigating the software development at one of the various IT-departments at Swedish bank Nordea, in order to point out the most interesting areas of improvement. Many different tools, standards, organizational processes and methodologies are available to the developers, whereof some of them might be inhibitory rather than enhancing the effectiveness. Nordea is also having an offshoring-oriented strategy, having development resources located in India. The discussion is concerned with modern methodologies such as Scrum and other agile development concepts, and their use in a geographically dispersed context and within a non-agile organization.

Handledare: Kristina Bäckman, Nordea AB  
Ämnesgranskare: Lars Oestreicher  
Examinator: Elísabeth Andrésdóttir  
ISSN: 1650-8319, UPTEC-STS12033  
Sponsor: Nordea AB

Tryckt av: Ångströmlaboratoriet, Uppsala

# Populärvetenskaplig sammanfattning

Moderna organisationer inom IT-utveckling möter utmaningar som inte nödvändigtvis är relaterade enbart till de tekniska aspekterna hos mjukvara. Dessa utmaningar kan handla om till exempel att upprätthålla kommunikation mellan intressenter, användarinvolvering, organisatoriska riktlinjer och regler, behovet av att utveckla eller använda standarder och underhåll av produkterna. Denna studie undersöker mjukvaruutveckling på en av de många IT-avdelningarna hos den svenska banken Nordea, i syfte att hitta intressanta förbättringsområden. Många olika verktyg, standarder, organisatoriska processer och metoder finns tillgängliga för utvecklare inom Nordea, varav några av dem eventuellt verkar mer försinkande snarare än ökar effektiviteten. Nordea har också en strategi för *offshoring*, med utvecklingsresurser placerade bland annat i Indien. Diskussionen handlar om moderna metoder som Scrum och andra agila utvecklingskoncept, och deras användning i geografiskt spridda sammanhang och inom en primärt icke-agil organisation.

# Contents

Populärvetenskaplig sammanfattning.....	1
Contents .....	2
Wordlist .....	5
1 Introduction .....	7
1.1 Problem statement.....	8
1.2 Limitations .....	8
1.3 Disposition .....	8
2 Background .....	9
2.1 Organization and roles .....	10
2.1.1 Savings & Nordic PB IT Solutions .....	11
2.1.2 External partners.....	12
2.1.3 Portfolio and advisory solutions .....	12
2.2 Local Development Processes .....	14
2.3 Financials .....	15
2.4 Development at other IT departments at Nordea.....	16
2.4.1 SAP and GL Production: .....	16
2.4.2 Funds Distribution Services Solutions .....	17
3 Literature review .....	18
3.1 Agile methods for software development and Scrum.....	21
3.2 Scrum methodology .....	22
3.2.1 Sprint .....	23
3.2.2 Backlog.....	23
3.2.3 Team .....	23
3.2.4 Scrum-master.....	23
3.2.5 Daily stand up-meeting.....	23
3.2.6 Storyboard .....	23
3.2.7 Retrospective .....	23
3.3 Popular concepts of other agile methods .....	24
3.3.1 Test driven development .....	24
3.3.2 Documentation .....	25
3.3.3 Pomodoro .....	25

3.3.4	Pair programming (XP) .....	25
3.3.5	RUP - Rational Unified Process .....	25
3.4	Kanban .....	26
3.5	Testing .....	27
3.6	Sociotechnical system.....	27
3.7	Process improvement and CMMI .....	27
3.8	Emergence and reasons for Offshoring.....	30
3.9	Best practice.....	31
3.9.1	Globally dispersed XP practices.....	32
3.9.2	Globally dispersed Scrum practices .....	35
3.9.3	Agile knowledge transfer.....	35
4	Methodology .....	37
4.1	Interviews.....	37
4.2	Workshops .....	38
4.2.1	Workshop 1 .....	38
4.2.2	Workshop 2 and 3.....	38
5	Empirical review .....	40
5.1	Why adapt internal development? .....	40
5.2	Why adapt offshore development? .....	41
5.3	Quality and efficiency.....	41
5.3.1	How to ensure quality in the finished product?.....	42
5.4	Offshoring and outsourcing .....	43
5.4.1	How do we work together offsite and onsite? .....	43
5.5	Information, attitudes and involvement .....	44
5.6	PAS compared to others .....	44
5.7	Feedback and evaluation.....	45
5.8	Available resources and support .....	46
5.9	Documentation, test and user involvement.....	46
5.10	Problems .....	47
5.11	Things that work well .....	50
5.12	Improvements suggested.....	51
5.13	Workshop results .....	52
5.14	Views on efficiency .....	55

6	Analysis .....	59
7	Discussion .....	64
8	Conclusions .....	66
9	Reference list.....	68
9.1	Literature.....	68
9.2	Internet .....	68
10	Appendix I: Questionnaire 1, PAS Staff .....	70
11	Appendix II: Questionnaire 2, External managers .....	71
12	Appendix III: Questionnaire 3, Upper manager .....	72
13	Appendix IV: Questionnaire 4, Change Management.....	73
14	Appendix V: Practical tips.....	74
15	Appendix VI: The action list .....	78

# Wordlist

## **Agile software development**

A philosophy for modern software development, focusing on the organization's ability to quickly make changes in direction according to new needs or changed requirements. See chapter 3.

## **Change Management**

Nordea's organization for handling changes within the bank. See chapter 2.

## **CIO (Chief Innovation Officer)**

A commonly used title for the head of the IT-department within an organization.

## **CMMI**

Integrated Company Maturity Model, a tool for describing how well developed a certain organizations' processes are. See chapter 3.7.

## **COBOL (Common Business Oriented Language)**

An old programming language.

## **Driver**

Local project leading role at Nordea Portfolio and Advisory Solutions. See chapter 2.1.3.

## **Funds Distribution Services Solutions**

An IT-department at Nordea.

## **Group Finance**

An umbrella term for all finance supporting facilities at Nordea

## **IDG**

The Improvement Decision group at PAS, having the responsibility to solve issues within the product and prioritizing change requests. See chapter 2.

## **Kanban**

A methodology for enhancing modern software development, originally developed for manufacturing processes.

## **Nordea PB (Private Banking)**

Nordea's organization for facilitating private banking services, available for customers with assets of 3MSEK or more. See Chapter 2.

## **Offsite**

The staff and site located in India

## **Offshoring**

To relocate employees (can be consultants) within the organization at a site offshore(called offsite) from the original one(called onsite). See Chapter 3.8.

**Onsite**

The staff and site located in Stockholm

**Outsourcing**

To assign in-house activities to external actors. See Chapter 5.4.

**PAS (Portfolio and Advisory Solutions)**

One of Nordea's various IT departments. See chapter 2.1.3.

**PBIT**

Private Banking IT, the various IT departments at Nordea Private Banking as a whole. See chapter 2.

**Pomodoro**

A concept used for more efficient and concentrated work. See chapter 3.3.2.

**Release manager**

A manager responsible for the development and implementation of a certain IT product. See chapter 2.

**RUP (Rational Unified Process)**

A methodology for project management. See 3.3.5.

**SAP and GL Productions**

An IT-department at Nordea.

**Scrum**

A methodology for enhancing modern software development. See chapter 3.2.

**Sociotechnical system**

Any system that includes components (hardware and software) and that is operated by humans in an organizational or societal context and therefor expected to be influenced by external forces such as organizational policies, procedures and structures.

**UML (Unified Modeling Language)**

A standardized language used in software development in order to document and explain workings by certain graphics and documents.

**XP (Extreme programming)**

A methodology for enhancing modern software development. See chapter 3.3.4 and 3.9.1.

# 1 Introduction

Nordea Private Banking IT (PBIT) is internal supplier of IT at Nordea Private Banking, with one of the objectives to deliver cost effective development and maintenance. This has led to PBIT pursuing development both locally, in Stockholm, and also offshore, located in India.

PBIT has a constant struggle for quality enhancements in a context of concurrently and at increasing rate changing orders and requirements, in a large multinational organization. Besides from the systems developing departments, there's also Change Management, Nordea's organization for managing changes within the development of new technology.

The change of starting to have development resources located in India at Nordea IT seems to have been rapidly conducted. Many members of staff indicate that they're more or less dissatisfied with the current situation or the introduction of the offsite. This is interesting, since the concept of Indian offshoring itself has been very well received within other organizations similar to Nordea IT. An article published 2012-05-17 in Computer Sweden tells of Indian consultant agencies being especially successful, where five of the six most valued consultant agencies within outsourcing are Indian, valued by an customer satisfaction index.<sup>1</sup>

Nordea is one of many organizations that have chosen this strategy and there's a lot of knowledge in the IT-business on how to succeed and which pitfalls to avoid. Every situation is unique in its details, but one can assume that Nordea is struggling with about the same challenges as many other organizations, for example irrelevant activities stealing time from development, or problems with finding the right competence and to keep it. According to an article published 2012-06-04 in the magazine Ingenjörskarriär, the ten most experienced time-consumers are<sup>2</sup>:

1. Becoming interrupted often when working: 33% of 6000 Swedish employees
2. Malfunctioning technology: 33%
3. Misunderstandings and unclear communication: 26%
4. Lack of time to work with my actual assignments: 22%
5. Deficient technology/tools/software: 22%
6. Bottlenecks such as waiting for others' deliverances before being able to continue work: 20%
7. Bad management: 20%
8. Meaningless or unproductive meetings: 19%
9. Unclear guidelines for what one's supposed to work with: 17 %
10. Hard to find people or make appointments: 17%

Many of these alternatives can be seen at Nordea PAS as well, as suggested by the employees in the interview study below.

---

<sup>1</sup> Rosengren (2012-07-31)

<sup>2</sup> Karlsson, (2012-07-31)

Nordea also has old COBOL-based systems and are having a hard time finding developers with the right competence in the home market. Other Swedish banks and insurance companies, among them Skandia, If, SEB, Handelsbanken, Länsförsäkringar, AMF Pension, Swedbank and Folksam, are facing this challenge as well and have together taken an initiative to form a one year long COBOL education together with a supplier of education.<sup>3</sup> This common interest suggests that the issue of finding the right competence as discussed further below is of greater interest than of just this study.

### 1.1 Problem statement

What are the most interesting areas of improvements today at PBIT, according to best practice, other internal IT-departments at Nordea, and eventual benchmarks, to pursue effective IT-development with off-shoring?

### 1.2 Limitations

The study is limited to Portfolio and advisory solutions, one of the various IT-departments at Nordea.

### 1.3 Disposition

This report provides certain solutions to the problem statement based on relevant literature and empirical support derived from interviews with employees at Nordea. A literature review is presented below in order to work as a conceptual foundation for the discussion including a presentation of modern software development methods and outsourcing/offshoring, followed by examples of best practice of such. The empirical review is focused on certain sub-questions evaluated with the responses of the interviews to regarding the actual experiences of the staff. The collected results from the interviews were used as discussion topics on a number of workshops that were held subsequently with a subset of the staff. Conclusions drawn from this material is subsequently analyzed, discussed and presented in the Analysis Chapter, structured as a SWOT-analysis(Strengths, Weaknesses, Opportunities, Threats), concentrated on pointing out the characteristics of Nordea PAS. The Discussion Chapter further discusses the results compared to the presented literature and practices of modern software development, and the conclusions drawn thereof is presented in the Conclusions Chapter, valuating the results from the workshops as well, placing them in a context. A set of appendices are attached to the end of this report, containing the interview questionnaire, the list of actions derived from the workshops and a list of practical tips for managing offshoring suggested in the literature.

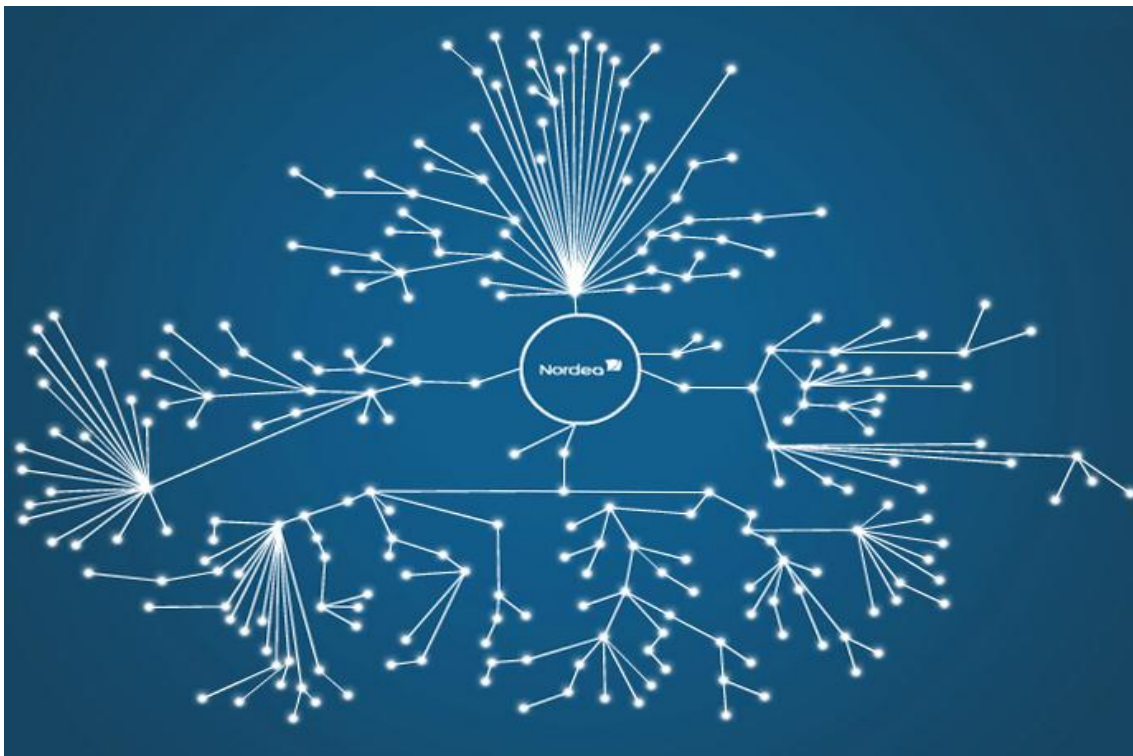
---

<sup>3</sup> Gusmán, (2012-07-31)

## 2 Background

Nordea is the biggest financial concern in northern Europe with 1400 branches, 33 000 employees, a market capitalization of roughly €700 billion and about 11 million customer whereof 9.5 are active household customers<sup>4</sup>. The home markets of Nordea are nine: Sweden, Denmark, Finland, Norway, Estonia, Latvia, Lithuania, Poland and Russia, in both retail and wholesale banking.

Many of the many branches, employees and resources have been acquired by divestments and acquisitions. Today's Nordea consists of what were over 300 different organizations merging over time until the four large banks Nordbanken (Sweden), Merita Bank (Finland), Unibank (Denmark) and Christiania Bank og Kreditkasse (Norway) united in December 2001 under the name of Nordea (see figure 1.).



**Figure 1: The mergings of Nordea 1820-2011<sup>5</sup>**

Through the mergings, the bank also acquired the former organizations' different IT-systems whereof many are today still in use or advanced or incorporated into new systems. These systems are maintained and developed by the many IT-departments of Nordea, which are organized in a structure of one IT-department per function.

Since 1999 Nordea has been offering private banking services for customers with a capital of 3MSEK or more. The vision of Nordea Private Banking is "to be the preferred

---

<sup>4</sup> Nordea AB (2012-07-25)

<sup>5</sup> Nordea AB (2012-07-25)

Private Bank across all our markets - acknowledged for our people and creating superior value for our customers”<sup>6</sup>.

The management of Nordea Private Banking has taken the decision to focus resources on quality improvements during the year 2012. The means and goals were said to concern<sup>7</sup>:

#### Clients/financial advisors

- Re-create trust in relation to the customer interfaces/facilities
- Assure that information/communication are meeting the expectations concerning timing, quality and feedback
- To establish one single support channel

#### Tools

- Adjust all stability-, quality- and velocity issues
- Enhance stability, make sure that facilities are available when needed
- Assure quality in client reports and advisor interfaces
- Assure that functionalities are fast and responsive without delay for printers etc.

#### Processes

- Document the structure of all value chains and make sure that the processes are understood by all concerned
- A process that assure the full implementation of new or revised products in all system supporting functionalities
- Well established roles and responsibilities
- Verifying competences within people in the value chain

#### Organization

- Evaluate the present organizational structure; if necessary perform changes in consideration of a well-functioning value chain.

### 2.1 Organization and roles

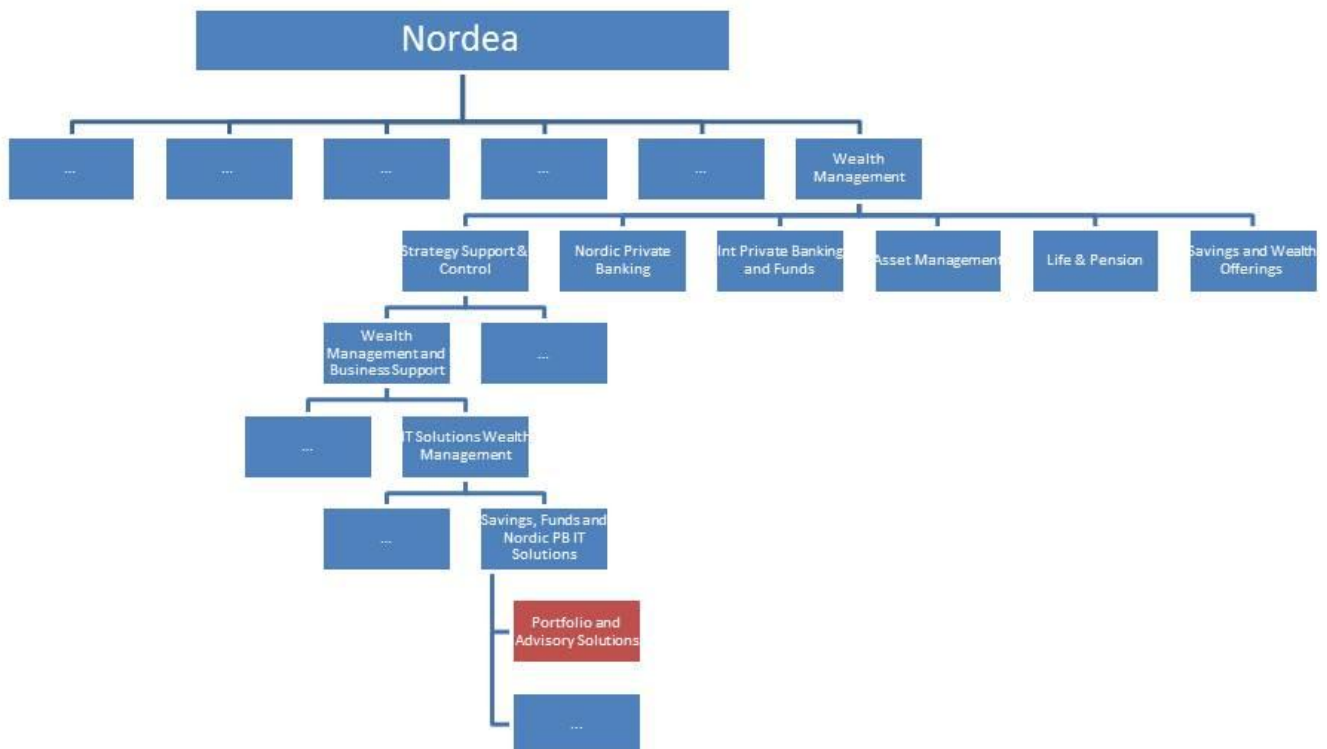
Nordea AB is divided into six divisions and two supportive functions. The six divisions are Retail Banking, Wholesale Banking, Group Operations and Other Lines of Business, Group Corporate Centre, Group Risk Management and Wealth Management. The division Wealth Management is further divided into six subdivisions: Strategy Support &Control, Nordic Private Banking, International Private Banking and Funds, Asset Management, Life &Pension and Savings and Wealth Offerings. Beneath Strategy Support & Control are further subdivisions, there among Wealth Management Business Support beneath which IT Solutions Wealth Management is located. One of its

---

<sup>6</sup> Nordea's Intranet for employees

<sup>7</sup> Nordea's Intranet for employees

subdivisions is Savings, Funds & Nordic PB IT Solutions, beneath which Portfolio and Advisory Solutions is to be found, see figure 2.



**Figure 2: Organizational map (abbreviated)**

### 2.1.1 Savings & Nordic PB IT Solutions

According to the internal webpages of Nordea, Savings and PB IT Solutions is “aiming at being a proactive and trustworthy IT Partner of Excellence, and is continuously investing in our ability to be more efficient.”

It’s also said that the means for reaching this goal is to:

- Fulfill the business area's need for daily deliveries from IT, and enable them to execute their long-term business IT strategies
- Help the business to reach their goals and objectives
- Enable us to attract and retain motivated, competent and empowered employees who provide superior IT solutions

Nordea has about 3500 different internal IT systems, many of these developed in-house<sup>8</sup>. The development and maintenance of these systems is the responsibility of Savings and PB IT Solutions who works in all the Nordic countries.

### **2.1.2 External partners**

All handling of hardware is outsourced to Nordic Processor by HP. Nordea has settlements with several partners within software development and maintenance for offshore resources, there among Accenture, CapGemini and L&T.

### **2.1.3 Portfolio and advisory solutions**

The purpose of subdivision Portfolio and advisory solutions is to build and maintain tailor made systems aiming to help clients to achieve a position as market leaders and achieve their respective goals and visions.

At the office there are posters stating PAS's visions and values set by the staff:

Portfolio and Advisory Solutions builds and maintain tailor made systems which helps our clients to become market leaders and to achieve their goals and visions.

We are a professional partner that values teamwork.

We who work in the Portfolio Management Solutions are proud of what we do.

We, together with our partners,  
makes it possible!

#### **1. Teamwork.**

We work together toward common goals by using each other's expertise and experience.

The team's success is more important than the individual.

Spread and share knowledge

Dare to ask for help and willingness to provide help

Be aware of the team's goals and values

Encourage and support each other both socially and professionally

#### **2. Professionalism**

We maintain the group's reputation as a good supplier of IT services. We work efficiently with good quality, meet client's requirements and act professionally and in contact with our partners.

Notify when agreements cannot be fulfilled

Be prepared for meetings

Exhibit behaviors that are in line with the team's values

Ensure knowledge for tasks and duties

---

<sup>8</sup> Interview with IT Manager

Listen to customer needs  
Advise and act pro-actively

3. Commitment

We hold ourselves accountable for achieving team goals and comply with the group valuations.

- Be engaged and motivated
- Take responsibility

4. Respect

We see and respect the strengths and diversity of our colleagues and partners.

- Foreclose and inform all
- You do not talk behind someone's back
- Be humble
- Engage in a constructive dialogue
- Stand for your opinion
- Listen to other's opinions and ideas

PAS is a fairly new unit which has been working under this name only for a short while, since the most recent reorganization of Nordea Private banking in May 2012. From May 2012, Portfolio and Advisory Solutions consists of a staff of about 30 people, about half of them consultants from many different agencies. In addition to this, a flexible number of consultants located in Bangalore in India employed by Indian consultant agency L&T are available as project resources.

The different roles at PAS are team leader, head of department, application manager, developer, driver, and release manager. In addition to this, there are the offshore resources located in Bangalore, scalable to need in projects. All roles at PAS except for managers can be filled by consultants.

**2.1.3.1 Developers**

The 14 software developers at PAS are mainly focused on either Java or database development, whereof two within databases and the rest in Java. In addition to these, PAS has one UX developer. Since the beginning of 2012, Nordea has no longer any testing resources, only test managers. Instead tests are to be performed by offshore consultants.

**2.1.3.2 Managers**

The local management of PAS consists of one Team leader and one manager.

**2.1.3.3 Onsite Indian consultants**

Currently, two Indian consultants are sited in Stockholm in order to work as managers for the offsite staff.

#### **2.1.3.4 Drivers**

The title “Driver” is a description of a project leader. The driver is responsible for estimating the budget for each project, follow it up and inform upwards. The estimate is including making time estimates, allocating resources for travelling costs, staff (offshore and onsite), Nordic Processing costs, test resources, business and hardware costs. It’s also up to the driver to locate and engage external resources, if needed and set up work and system accesses for those.

If (or when?) the plan is not being followed, the driver reports to Change Management, Steering Group and Team leader.

The driver is also supposed to stay informed on her project and other ongoing projects and keeping all stakeholders informed about the progress, possible impact on other systems and values of PAS and to confirm and validate participants’ knowledge and skills. Cooperating with other groups is a continual activity: consulting the expert group, synchronize release dates and code base etc. with Application Managers and Change Release managers and performing handovers for Application managers and offshore resources.

Testing is also a responsibility of the drivers’, both to make sure that it’s budgeted for and that it’s actually performed as well. In bigger projects, drivers should facilitate for one hour long system tests where all resources within PAS participate.

#### **2.1.3.5 The expert group**

Since the beginning of 2012 a specific group consisting of four senior developers has been established, named the expert group. The members work with their ordinary tasks as well, aside of counseling projects in the group. The expert group is supposed to support in larger change requests or projects, mainly concerned on IT architecture but sometimes on business as well. The members of the expert group are to keep each other updated on solutions on ongoing projects and to mediate their knowledge among each other and other project participants to enhance the competence in the group and make the representatives grow in their roles. The idea is to allocate one expert group resource for each project. The expert group should also have mandate to dismiss bad solution ideas.

## **2.2 Local Development Processes**

Nordea is following a trademark project management standard called PM4U, it’s mandatory for all departments.<sup>9</sup>

A process is state to not be allowed to start unless

1. An analysis is made
2. The design is approved by the Expert group
3. There is a (big enough) budget

---

<sup>9</sup> Interview with Change Management representative

#### 4. An agreement is signed by both Business and IT

The analysis may start as soon as the project has been assigned to the driver by the team leader or the CIO. During the analysis phase, some steps ought to be performed: a workshop with customers, assurance that the requirements document has sufficient quality, a Quality and Risk Analysis should be performed and an agreement needs to be set and documented on what the project includes and not.

The actual process follows the project management lifecycle consisting of four decision points where mandatory documents are approved by the steering group and between these points three phases called Prepare Project, Run Project and Close Project (see figure 3).



**Figure 3: The Project Management Lifecycle of Nordea<sup>10</sup>**

These phases are set by Nordea's customization of the project management standard based on the international standard described by PMI in the PMBOK Guide. The system development happens in the Run Project phase which can be divided into several development phases.

The decision points should include a review of results and documentation and self-assessment; recommendations for decisions; and decisions for the next phase: whether the project is allowed to continue to the next phase or not.

The system development process that runs in the Run Project phase is common for all IT areas within Nordea and is based on RUP but tailored to fit Nordea standards, altogether by Nordea called SDP (Software development process). The names of the phases in SDP are aligned to PM4U: Prepare Project, Run Project/Establish, Run Project/Achieve and Run Project/Transition.

### 2.3 Financials<sup>11</sup>

Developers that are Swedish Nordea employees cost 96€/h. The costs for salaries are allocated to the projects or maintenance tasks that the specific resources is assigned to and accounted for by hour.

<sup>10</sup> Nordea's Intranet for employees

<sup>11</sup> Interview with Application manager at PAS

Indian consultants range in costs depending on their specific role, competences and placement, varying from 15€/h plus tax (less experienced developers and testers sited in India) to 25€/h plus tax (senior developers sited in India). Indian consultants sited in India having in manager roles cost up to 30€/h plus tax. When Indian resources are sited in Stockholm the costs are between 50 and 60€/h plus tax depending on competence and roles.

Efficiency and productivity are not currently being measured by PAS. Attempts to use the number of Story points worked on in maintenance projects as a measurement has been made but it appeared to be obtuse since only efficiency of teams could be measured, and the teams were changing for every project.

## 2.4 Development at other IT departments at Nordea

Two other of Nordea's IT departments are SAP and GL Production and FDS Solutions. The managers of these departments were interviewed to provide a short description of their work and how they've chose to tackle the same challenges and opportunities as PAS are facing.

### 2.4.1 SAP and GL Production<sup>12</sup>:

At SAP and GL Production, there's no internal development, everything is located in India. The department consists of 100 people in India and 40-45 people in the retail organization in Scandinavia. The reasons are many but the primary one is the lower price. Another reason is that SAP and GL Production's applications are old, running on COBOL. New developers are not educated in COBOL and the staff is getting older so when new developers are needed they're to be found them in India. The scalability is also a benefit; it's easy to adjust the number of consultants.

The differences between the tasks SAP and GL Production does compared to PAS at the different sites are that in Sweden they're focused on design, architecture and business analysts translating business requirements to IT requirements, since both business knowledge and IT knowledge are needed. The offsite work is outsourced and not offshored, since offsite has the full responsibility for the areas that are assigned to them.

Communication is pursued by phone calls, emails and video conferences and the Indian consultants rotate every three months between onsite and offsite. The challenge is that two different cultures meet, and it takes a long time to get to know each other especially since the two groups speak different kinds of English. That's why the most important issues are communicated by email, so they're documented to avoid misunderstandings. The Indian consultants work at the same time as the Scandinavian staff, CET.

---

<sup>12</sup> Interview with IT manager at SAP and GL Production

### **2.4.2 Funds Distribution Services Solutions<sup>13</sup>**

At Funds Distribution Services (FDS) Solutions, the development is pursued by teams that are both cross border and cross functional, all adopting the methodologies of Scrum. Before this configuration, IT used to work as a proxy without benefit since the costs went down but without increasing the deliverances. Three Indian people cost as much as one swede but the swede would do the same amount of work as the three Indians together. Now all teams except one are cross border, the last one is entirely Nordic to be used as a reference for measuring and comparing efficiency. The Scrum masters are usually located in India.

The challenges lie within the distance and the communication. Staff at FDS Solutions travels to India a lot to make sure the development is running fine and it does when one's present but regresses when one leaves.

To increase the proximity FDS Solutions has a directly connected TV to enhance communication by being able to see each other's faces. The daily Scrum meetings are held by conference phones or by web cameras. There's a lot to gain from increasing proximity instead of just sending documents to India since they can learn about Nordea and Nordea's activities.

FDS Solutions measures efficiency by story points, e.g. if an Indian raises a third of the cost of a Swedish employee but on the other hand produces less story points, the gain is less. Costs for travelling and communication have to be accounted for as well.

All of the maintenance assignments are handled with Kanban in order to lose commitment, the staff works on these assignments when one has the time, and that has been a good practice. One can trust people that they're handling the assignments when they have the possibility.

---

<sup>13</sup> Interview with IT manager at Funds Distribution Services Solutions

### 3 Literature review

Professor Emeritus Ken Eason tells in *Information Technology and Organisational Change*<sup>14</sup> of this kind of new roles emerging within user organization in order to cope with implementation and operation of information systems. People in these positions are to smooth the path to effective implementation and therefore the roles are often filled by people from different backgrounds: users, technical specialists with UX skills or people from the outside. Eason further claims that these roles create a need for new methods since the normal methods for software development are void since the people filling these roles haven't the traditional technological knowledge. Eason therefor develops techniques for handling these aspects of software developments. Eason's book might be considered as obsolete being published over 20 years ago but in fact many of Easons reflections upon IT development are highly relevant yet today since Eason manages to predict some of the challenges that organizations today are still struggling with.

As a context for these techniques, Eason defines the organization; as a collection of resources deployed to handle a specified workload<sup>15</sup>. Information technologies are means for either increasing the work undertaken or to replace or reduce the needed resources. He groups the benefits of IT into four types: cost reduction, improved productivity, improved support and organizational enhancement. These benefits are more or less tacit, and also maps to either the resource deployment or the work enhancing, as mentioned above.

The many human and organizational change issues that occur in order to make benefits of IT need to be respected in the development process. For this purpose, Eason suggests considering these aspects during the system design for systems to be implemented within the organization, underlining other forces having impact on the implementation such as politics anxieties or user frustrations. Eason presents a list to illustrate users' reactions as a list of "counter implementation strategies" impeding the development as follows<sup>16</sup>:

1. Lay low, keep out and do not give help or encouragement which increases the likelihood of failure
2. Rely on inertia and make sure to be too busy when asked in order to delay the process
3. Keep the project complex, hard to coordinate and vaguely defined, in order to consume unnecessary amounts of energy.
4. Minimize the implementers legitimacy and influence: keep designers alienated to make other users not willing to allow them to work effectively
5. Exploit their lack of inside knowledge, making the system more inadequate when implemented.

---

<sup>14</sup> Eason, (1988)

<sup>15</sup> Eason, (1988), page 13

<sup>16</sup> Eason, (1988), page 33

The design process is stated as a political process where many forces and interests will affect the implementation. If this is not managed, it will probably be too late to resolve implementation problems when all the forces have worked during the design phase making the system mismatched to the actual organizational needs. By this, it's concluded that organizational change issues must be addressed early in the design process so that "technical and organizational work can proceed in parallel rather than in sequence".<sup>17</sup>

Eason describes problems that might occur when working with the modern and by the time of his book's origin unexplored, development methodologies concerned with user involvement. The suggested problems are within the users' lack of technical knowledge, not being able to understand or evaluate design proposals, communicating with specialists, defining the users' needs, seeing opportunities, choosing the user representatives, resolving conflicts concerning different interests of different groups, achieving technical not organizational design and the matter of working with the users at the right time. The users are to contribute to their benefit of the systems but they are not well prepared to do so. When the user representatives experience this, it might affect their willingness to work with the system development and implementation. In the conclusions, Eason suggests that methods similar to 21st century agile development are in favor to the users, but the essence is to tackle the two problems of users either choose the wrong system or devote too much time to the development or that one is developing a system fit for the narrow needs of the user but not to the rest of the organization.

Eason comes down to two conclusions<sup>18</sup>:

1. "We are unlikely to achieve real benefits from information technology unless we find ways of designing for the human and organizational changes that are needed as well as designing the technical changes."
2. "Examination of systems design processes suggests that current methods do not systematically address these issues. The implication is that we must seek ways of compensating for the current emphasis upon the technical side of the change process."

To provide an overall strategy for this, a set of propositions to establish the wished objectives are presented, together with a set of conditions to be met for achieving these:

Propositions of the objectives:

The successful exploitation of information technology depends upon the ability and willingness of the employees of an organization to use the appropriate technology to engage in worthwhile tasks.

1. The design target must be to create a socio-technical system capable of serving organizational goals, not to create a technical system capable of delivering a technical service.

---

<sup>17</sup> Ibid.

<sup>18</sup> Eason, (1988), pages 44-49

2. The effective exploitation of social-technical systems depends upon the adoption of a planned process of change that meets the needs of people who are coping with major changes in their working lives

Propositions of the achievements of the objectives:

3. The design of effective socio-technical systems will depend upon the participation of all relevant 'stakeholders' in the design process.
4. Major benefits will only result if the socio-technical developments are directed at major organizational purposes where there are opportunities to be taken or problems to be resolved.
5. The specification for a new socio-technical system must include the definition of a social system which enable people in work roles to co-operate effectively in seeking organizational purposes and provides jobs which incumbents perceive as worthwhile
6. Information technology systems must be designed to serve the functional needs of the organization by serving the functional needs of individual users in a usable and acceptable way.
7. The effective exploitation of information technology requires a major form of organizational and individual learning
8. The exploitation of the capabilities of information technology can only be achieved by a progressive planned form of evolutionary growth.
9. To be successful, socio-technical design concepts must as far as possible complement existing design procedures and organization change practices.

The contributors of the design processes are divided into experts and customers, the latter including "not only those who are actively seeking the change but all who will be affected by it". Stakeholders are defined as by Mitroff's concept considering "all who have a stake in the change being considered, those who stand to gain from it and those who stand to lose".<sup>19</sup>

One might think of the relations within a design process as one between contractors holding the expertise and the customers acting as stake holders but Eason indicates that there's more to it. Both customers and experts contribute to both the stake-holding and the knowledge as the customers are experts on some issues such as knowing the way the organization functions and the experts are "not neutral suppliers of a service but have a stake holding in what it supplied".<sup>20</sup> The technical experts have likely a wish to have the system contributing to advance their own design skills for example and conversely the customers' stake holding consists of having to live with the consequences for example for their tasks and jobs.

---

<sup>19</sup> Eason, (1988), page 60

<sup>20</sup> Eason, (1988), page 61

### 3.1 Agile methods for software development and Scrum

Important characteristics of businesses today are that they operate in global and volatile contexts. This has augmented the need for rapid reforms and changes of tactics. New approaches on software development in contexts like these were suggested from early 1990s<sup>21</sup>, aiming to be better suited for the new conditions. A widely spread approach deriving from these approaches is the one of agile development, characterized primarily by work being carried out in an iterative and incremental manner. These work processes are meant to enhance rapid system development and changing requirements, to produce useful software quickly<sup>22</sup> and to allow the developers to focus on the software itself rather than on its design and documentation.<sup>23</sup>

Sommerville<sup>24</sup> suggest some fundamental characteristics, adapted by most approaches:

- Enlacement of the processes of specification, design and implementation.
- The system is developed in a series of versions.
- System user interfaces are often developed using an interactive development system that allows the interface design to be quickly created by drawing and placing icons on the interface

A manifesto describing the essence has been formed:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”<sup>25</sup>

Agile methods are considered successful at least for some types of software development<sup>26</sup>:

1. “Product development where a company is developing a small or medium-sized product for sale.
2. Custom system development within an organization, where there is a clear commitment from the customer to become involved in the development process

---

<sup>21</sup> Sommerville, (2011), page 59

<sup>22</sup> Sommerville, (2011), page 57

<sup>23</sup> Sommerville, (2011), pages 58-59

<sup>24</sup> Ibid.

<sup>25</sup> Ibid.

<sup>26</sup> Sommerville, (2011), page 59

and where there are not a lot of external rules and regulations that affect the software.”

The common principles of agile methods are<sup>27</sup>:

### **Customer involvement**

Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.

### **Incremental delivery**

The software is developed in increments with the customer specifying the requirements to be included in each increment.

### **People not processes**

The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.

### **Embrace change**

Expect the system requirements to change and so design the system to accommodate these changes.

### **Maintain simplicity**

Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

Two questions should be considered when considering agile methods and maintenance<sup>28</sup>:

1. Are systems that are developed using an agile approach maintainable, given the emphasis in the development process of minimizing formal documentation?
2. Can agile methods be used effectively for evolving a system in response to customer change requests?

## **3.2 Scrum methodology**

Scrum is one of the commonly practiced agile methods, focusing on managing iterative development rather than specific technical approaches<sup>29</sup>. Below follows some of the key concepts:

---

<sup>27</sup> Sommerville, (2011), page 60

<sup>28</sup> Sommerville, (2011), page 61

<sup>29</sup> Sommerville, (2011), page 72

### **3.2.1 Sprint**

All projects are split up into a series of shorter time periods of a fixed length of 2-4 weeks, called sprints. Each sprint is resulting in an increment of the system. In each sprint, work is assessed, requirements are set, work is reviewed and the software is implemented, demonstrated and delivered to customers.

### **3.2.2 Backlog**

The backlog is the list of work to be done. The elements on the list are set with priorities in the beginning of each sprint, in cooperation with the customer and all members of the team.

### **3.2.3 Team**

The team organizes themselves on who should perform which of the set tasks. All team members are involved in selecting the features and functionality of the product.

### **3.2.4 Scrum-master**

The term project manager is avoided in Scrum methodology, since the whole team are supposed to make decisions. The role of the Scrum master is to be the only channel for communication between the customer and organization and to protect the team from external distractions. The scrum master is also the one who arranges the daily meetings and keeps track of the backlog.

### **3.2.5 Daily stand up-meeting**

Every day the team has a meeting where they share information and update each other on progress, problems and plans for the following day. To make sure the meeting is short and valuable; the participants ought to stand up.

### **3.2.6 Storyboard**

The concept of user stories and story cards as measurable chunks of development is originally a concept of another agile methodology: XP, short for *Extreme programming*. Before the development starts, the team decides on what stories should be developed, break them down into tasks and estimate the time and resources needed for each. During the sprint, team members pick up tasks one at the time. In Scrum, one has often a specific board where the stories and tasks are visible to illustrate the progress and what are left to do. Usually, the storyboard is presented at the daily meeting to help plan the day's work.

### **3.2.7 Retrospective**

At the end of the sprint, a retrospective is held to review the work that has been done. The aim is to improve the work for the next sprint or project.

Whether to use agile or plan-driven development depends on several characteristics of the project and of the organization. To fully achieve the benefits of agile development there has to be for example realistic possibilities to achieve rapid customer feedback, a fairly small and co-located team, systems that won't require a lot of analysis before implementation, specific development tools and no demands for specific documentation. Pursuing plan-driven development might be a better and more efficient choice if the project and organization have that culture, is geographically widespread or work on large systems. The important thing is not whether labeling a project as plan-driven or agile but if the customers have a system that meets their needs and does useful things for the individual user or the organization. Many companies that claim to have adopted agile development are actually having plan-driven development processes integrated with some agile practices<sup>30</sup>.

### 3.3 Popular concepts of other agile methods

The use of methodologies such as Scrum is widely adapted within organizations working with software development, but not necessary to 100% but suited for the specific organization, for example by mixing in elements from other methodologies. Below, some other concepts of agile software development that are used by Nordea PAS are introduced.

#### 3.3.1 Test driven development

The view on testing is considered as one of the major differences between incremental and plan-driven development. By writing tests, one implicitly defines both an interface and a specification of behavior for the functionality, reducing misunderstandings. With XP some key features of testing were developed, namely:

- Test-first development to first write test then code, in order to discover problems during development running the tests as the code is being developed.
- Incremental test development from scenarios, since the system requirements are always linked to the implemented code.
- User involvement in the test development and validation, to help develop acceptance test for the stories and to make sure that the system meets the user's real needs.
- Use of automated testing frameworks. All new code should be tested to meet the requirements

Some of the problems that rise with test-driven development are that the programmers might prioritize programming before testing, some tests are hard to write incrementally and it's difficult to judge the completeness of a set of tests. A large set of tests doesn't necessarily mean that the system is complete and correct; the tests must have good quality themselves.<sup>31</sup>

---

<sup>30</sup> Sommerville, (2011), page 64

<sup>31</sup> Sommerville, (2011), pages 69-70

### 3.3.2 Documentation

The point with formal documentation is both to describe the system and to make it easier for people in the future to maintain the system. Agile methods enthusiasts points out that documentation is in reality often not updated and maintained so it loses its meaning and the energy invested in writing it is wasted.<sup>32</sup>

### 3.3.3 Pomodoro

The Pomodoro technique is aiming to effectively use time, basically through cut out all possible distractions. In practice, one isolates oneself, turning off phones and email to work concentrated with a task for 25 minutes. The productivity is improved by being able to focus, cutting down on interruptions and keeping up the motivation. By setting up this ideal environment one ascertains high productivity for 25 minutes, quality time as to say. The concept is easy to understand and no specific tools are needed, even though the use of a egg timer is suggested, preferably a tomato-shaped one<sup>33</sup>.

### 3.3.4 Pair programming (XP)

Pair programming is a concept of XP. Two programmers sit together for a time at the same computer when developing. The advantages of this concept are collective ownership, direct code review, support for refactoring and most important: knowledge sharing. Studies show that the productivity is not negatively affected, eventually because of fewer errors<sup>34</sup>.

### 3.3.5 RUP - Rational Unified Process<sup>35</sup>

RUP is a generic process model that has been derived from UML and the associated Unified Software Development Process. It's normally described from three perspectives:

1. The dynamic – shows the phases of the model over time
2. The static – shows the process activities that are enacted
3. The Practice – suggests good practices to be used during the process.

It defines four different phases of the software development process, all closely related to business rather than technical concerns. These are *inception* – to establish a business case, *elaboration* – to develop an understanding and establish an architectural framework, project plan and risks, *construction* – including system design, programming and testing and *transition* – moves the system from development to implementation. Each phase may be iterated developing results incrementally, or the whole process can be iterated.

In RUP there are also so called workflows, which are the activities during the development phase, seen from the static view, in total six, all oriented around associated

---

<sup>32</sup> Sommerville, (2011), pages 61

<sup>33</sup> FC Garage, (2012-07-25)

<sup>34</sup> Sommerville, (2011), page 71

<sup>35</sup> Sommerville, (2011), pages 50-53

UML. When presenting the dynamic and static views there's no need to associate the phases of development with specific workflows, leading to several or all workflows could be active parallel. All phases are dynamic and have goals but workflows are static and are technical activities not associated with single phase but available during the whole development process to achieve the goals of each phase.

Six fundamental best practices of RUP are recommended:

1. Develop software iteratively
2. Manage requirements
3. Use component-based architectures
4. Visually model software
5. Verify software quality
6. Control changes to software

### 3.4 Kanban<sup>36</sup>

Kanban is a methodology developed in the 1940s by Toyota in order to control their Just-In-Time processes by the idea of minimizing work and reduce the cost of holding inventory. The methodology developed during the years and began to spread worldwide during the 1970s. Today, Kanban is popularly used to manage costs and flows within production systems and to identify bottlenecks. The basic idea of Kanban is to visualize the flows and by that make it clear to everyone what is produced, where, when and localize demands for more resources. This is visualized by *kanbans* (Japanese for signposts) that the workers set up symbolizing the status of their working station. This is to make sure that one doesn't produce more than necessary of anything and also that the production is never lagging because of dependence of the supply chain. The need for inventory is also reduced and the progress is considered agile since it's easy to increase or decrease production following the external demands.

In software development modern methodology has been developed inspired by Toyoya's *kanbans* but considering the production of code or functions. At Toyota, the actual *kanban* was symbolizing a certain number of parts to be manufactured and later available. When the *kanban* was used it was sent up the supply chain working as an order for new parts. Since no new cards enter the circulation, one has always the number of parts needed and new ones are coming in when needed from the upper level at the supply chain. When Kanban methods are used for software development the *kanbans* symbolizes certain tasks, as promised of work. There's also a *kanban board* which visualizes the teams and which tasks (*kanbans*) that each station is currently working on. Each team has a set number limiting how many tasks it can take on at the same time and can only be assigned a new task if the limit is not reached. Further, the team can only pass on a finished task to the next team if their limit allows it. If the process gets stuck at any point, the process needs improvement<sup>37</sup>.

---

<sup>36</sup> Gross&McInnis, (2003)

<sup>37</sup> Björkholm, (2009-06-04)

### 3.5 Testing

The purpose of testing is to assure that the software does what it is intended to do – fulfill the users' needs, meets the requirements and behave correctly, and that it doesn't have errors – what it's intended not to do even though testing can only show the presence of errors and never their absence. Testing is a part of a broader process within the development of software namely validation (Are we building the right product?) and verification (Are we building the product right?). This is essential because requirements specifications do not always reflect the real users' needs. The system may also have other purposes such as integration with other systems or having a certain level of security.

In modern software development, testing occurs both during and after writing the actual code, in various ways such as component testing, full system testing, acceptance tests by customers, automated testing, user testing, performance testing and more. All these concepts contribute to the common goals and different kinds of tests are needed since no one of them can test the system out of all aspects. Requirements may be functional and non-functional demanding different kinds of review and some aspects such as human behavior when real users are to interact with the system are impossible to predict.<sup>38</sup>

### 3.6 Sociotechnical system

A sociotechnical system is any system that includes components (hardware and software) and that is operated by humans in an organizational or societal context and therefor expected to be influenced by external forces such as organizational policies, procedures and structures<sup>39 40</sup>.

### 3.7 Process improvement and CMMI<sup>41</sup>

Process improvement is said to mean understanding existing processes and changing these to increase product quality and/or reduce costs and development time. Two different approaches are used:

- Process maturity, with focus on improving process and project management and introducing good practice into an organization with the goal of improved product quality and process predictability. This approach has its roots in plan-driven development and might lead to increased overhead because of the activities not being directly relevant to programming.
- Agility, with focus on iterative development and the reduction of overheads in the process by rapid delivery of functionality and responsiveness to changing customer requirements.

---

<sup>38</sup> Sommerville, (2011), Chapter 8

<sup>39</sup> Eason(1988), Chapter 4, pages 44-59

<sup>40</sup> Sommerville, (2011), page 745

<sup>41</sup> Sommerville, (2011), Chapter 26

For software products four factors influence the product quality, namely: *process quality, development technology, people quality* and *cost, time and schedule*. How important each of these factors is for a development process depends on its size. According to Sommerville, the real cause of low software quality is often not poor management, inadequate processes or poor quality training but the fact that organization must compete to survive thereby accepting unrealistic schedules.

There's no such thing as an ideal development process fit to every organization, each company has to develop and refine their own depending to its size, background, skills, demands and company culture. Therefore is process improvement not about adopting particular methodologies but considering certain aspects of the process and how to influence them with courtesy to the local context. Often one has to improve one at the cost of another, making it about leveling.

The process improvement process is a cyclic one moving from measuring to analyzing to changing to measuring. Process measuring can be used to assess whether or where efficiency has improved but the measurements cannot on their own be used to determine if product quality has improved, product quality data must also be collected and related to the activities. The three types of process metrics that can be collected are

1. Time taken for a process to be completed
2. Resources required
3. The number of occurrences of a particular event, for example defaults discovered, number of requirements changes requested or the average number of lines of code modified in response to requirements changes.

Measurements of time and resources can be used to find out if the process changes have improved the efficiency while measurements of type 3 have more to say about product quality. Measurements generate evidence about processes and process changes but it's essential that this evidence is interpreted along with other information about the process, qualitative assessment of changes.

Process analysis is concerned with understanding the key characteristics of processes and how they are performed in practice by the people involved and it's often entwined with process measurement in reality. Objectives are to understand the involved activities and the relations between them, to relate them to the measurements and to relate the specific process to processes elsewhere in the organization or idealized processes. Some interesting aspects of process analysis could be about the process' adoption, standardization or documentation, the use or lack of software engineering practices, organizational constraints, how communications are managed, if there are introspection about the process, how learning is pursued, what tool support that is available and how the process handles exception such as absence of staff or artifacts.

Process change involves making modifications to the existing process. Changes should be driven by specific goals and after the change is made it should be measured to assess the effectiveness of it. The five key stages in the process change process are

*improvement identification, improvement prioritization, process change introduction, process training and change tuning.* Two major difficulties with process changes are occurrence of resistance to change by team members or project managers and change persistence – innovations being discarded or processing being reverted. Common reasons for this might be risk-aversion; the processes being seen as a threat, not valuing the staff's skills and experience; fear of losing the job; no wish to learn new skills, tools or ways of working. To avoid this, management has to be sensitive and involve the team all the way through the change process and understand their doubts. It is much more likely that the staff will want to make the new process work if they are made stakeholders in the process change. Changes are also unlikely to result in immediate benefit, rather on long term. It's also important that the manager of the change has commitment to it, or the risk of the change reverting is high.

The Software Capability Maturity Model (CMM) strongly argues for the institutionalization of process change. The CMMI model is intended to be a framework for process improvement. The actual model is very complex but simplified its principal components are:

- A set of process areas related to software process activities.
- A number of goals that should be attained. These could be specific for the process areas or generic.
- A set of good practices working as descriptions for achieving the goals.

The generic goals and practices are not technical but associated with the institutionalization of good practice depending on the maturity of the organization. The CMM assessment involves examining the processes and rating them on a six-point scale related to the level of maturity where a more mature process is considered the better. The scale is as follows: incomplete, performed, managed, defined, quantitatively managed and optimizing. To improve its processes an organization should aim to increase the maturity level of the areas that are relevant to its business.

The successor of CMM is CMMI abbreviated as Capability Maturity Model Integration, aiming to evaluate how the organization is following its development process rather than the quality of the development process itself.

The five level of maturity of a company defined by CMMI indicating the level of quality and what mechanisms there are in the organization to ensure quality<sup>42</sup>:

- Initial level – The organization have no standardized process for software development and no follow-up projects that could help make better estimates of the costs and terms for future projects. The goal of this step is to get basic control over the timing and costs.

---

<sup>42</sup> Wikipedia, (2012-07-25)

- Repeatable level - The organization has processes for configuration management, but all projects run according to the project manager's discretion. Basic statistical tracking is done by the obligations, costs, schedules and changes.
- Defined level - The organization has defined a set of processes and standards across the organization and applies them as well.
- Managed level - The organization performs extensive measurements and analysis on the effectiveness and quality of their processes in different projects.
- Optimized level - The organization monitors trends in the efficiency and quality through time and continuously improves their processes and organization.

### 3.8 Emergence and reasons for Offshoring<sup>43</sup>

Peter Dicken mentions in *Global shift* that the “higher order” financial and service functions within banks are heavily concentrated in the major global financial centers (New York, London, Tokyo...) but the so-called “front-office” functions must be close to the customer. This implies the large branch networks of retail banks and other financial services supplying final demand. Dicken further claims that it is the advent of Internet and telephone banking that has allowed retail banks to rethink their strategies within location.

Since the essence of financial service activities is the transformation of massive volumes of information, there's a great need for clerical workers that can process these routine activities. The access to large pool of appropriate, often female, labor was a key requirement. Such activities can be separated from the front-office functions to be performed anywhere where it's beneficial for the organization. The banks were early adopters of large-scale computing which led to many of them setting up large centralized data processing units and to escape the high costs of these centers, both land and labor, the units were often relocated to the suburbs.

When the introduction of dispersed computer networks made the centralized processing units unnecessary the tendency changed to decentralization of back-office functions at the same time as the distinction between back-office and front-office became less clear. It's not just the routine back-office activities that have been decentralized but some of the higher-skilled function as well becoming relocated away from the head-office into dispersed locations, nationally and transnationally – offshoring.

At 2003 approximately 25% of the companies within financial services utilized offshoring to some degree, at 2004 more than two thirds did. It's seen as the beginning of a paradigm shift within competitive dynamics for financial service companies operating globally, changing the rules of the game for every player. As the pace of offshoring accelerates, the companies' strategies are changing; relocating a broader set of function and processes, operations conductor offshore to include more back-office and customer-facing services.

---

<sup>43</sup> Dicken, (2010), pages 403-405

Dicken suggest three models of utilization of offshoring, all with their specific benefits and risks:

*The Vendor direct model* – to place contract with specialist firm in specific country

Potential benefits: Cost reduction, use of specialist expertise, speed

Potential costs: Loss of control, security risks

*The Captive direct model* – to set up own directly-owned operation(s) overseas

Potential benefits: Increased control, reduced risk, greater security

Potential costs: High cost of establishment and control

*The Vendor indirect model* – to place contract with specialist firm with operations in several countries

Potential benefits: Lower costs, use of specialist expertise, vendor reputation

Potential costs: Control issues, security risks

These different utilizations of offshoring have been distinguished as the extent of offshoring has increased, according to Dicken:

Initially, most offshoring by financial service firms was vendor direct outsourcing to a foreign firm located overseas. By far the most popular location was India, where IT-related outsourcing has grown at a phenomenal rate. As a result, the region around Bangalore has become one of the biggest geographical concentrations of IT-related activity. As major financial service firms became increasingly involved in offshoring they began to establish systems of active direct offshoring: setting up their own subsidiary operations in other countries. The third an most recent arrangement, vendor indirect offshoring, reflects the tendency for specialist outsourcing companies to establish their own transnational networks to serve more diverse customers.<sup>44</sup>

### 3.9 Best practice

Editors Smite, Moe and Ågerfalk provides in *Agility across time and space*<sup>45</sup> a collection of empirically gained knowledge of the implementation of agile methods in global software projects, working as a guide for best practice, lessons learned and a set of practical tips. Smite et al claims that “despite the progress in the field of software engineering, software projects are still being late, are over budget, and do not deliver the expected quality<sup>46</sup>”. To manage this, the authors explain the two major trends that have emerged in response to these: global sourcing and the application of agile methods, aiming for cheaper and faster development of high quality software, highly interesting as many companies today are merging these two approaches<sup>47</sup>.

The book consisting of a number of text written by different experts experienced in these strategies aiming to conclude what to expect from merging agile and distributed

---

<sup>44</sup> Dicken, (2010), page 405.

<sup>45</sup> Smite et al,(2010)

<sup>46</sup> Smite et al,(2010), Preface, page vii

<sup>47</sup> Ibid.

strategies; what challenges might occur and to recognize unfeasible strategies and unfavorable circumstances; survival and efficiency.

### **3.9.1 Globally dispersed XP practices<sup>48</sup>**

One of the most adapted methodologies in agile development is XP and whether it can be successfully used in global software project teams are discussed by experts, some of them pointing out the specific benefits that are to gain from certain XP concepts, others underlining the challenges posed. It's important to be aware on how the team, the organization and the concepts chosen might affect the suitability and thereby results.

Different strategies of team distributions can be characterized in terms of spatial, temporal and configurational dimensions. By the understanding of these terms, managers are in a better position to understand what the challenges and opportunities are for successfully implement XP practices. Spatial dispersion represents the distance between physical locations of the team and ranges between team members being located in the same building at the same floor to a dispersion of different countries with various ways of traversing the distance in order to facilitate communication. Temporal dispersion reflects the extent to which there's an overlap in normal work hours for team members where increasing levels leads to challenges for teams to engage in effective work coordination. At last, configurational dispersion is concerned with having team members across physical locations as in the number of distinct building or cities where members are located and can be understood independently of spatial and temporal dispersion.

In addition to this, there are a number of implementation strategies and pitfalls to avoid when pursuing XP in distributed software project teams, there among collective ownership, coding standards, the use of metaphors, simplicity of design, sustainable pacing, the use of pair programming, continuous integration and unit testing, refactoring, customer involvement, and developing small functional releases.

#### **3.9.1.1 Collective ownership**

Collective ownership aims to provide a clearer understanding of roles and responsibilities of all members with respect to the software. This is an important practice because it has been found to enhance the quality of the produced software by giving all team members a shared responsibility enhancing practices such as refactoring. Teams implementing collective ownership might only need to meet once at the beginning of the projects to discuss roles and responsibilities. Collective ownership also implies that team members may make changes to any part of the software if needed. Action-oriented practices such as unit testing, acceptance testing and continuous integration serve as a safety net, allowing the team to identify defects and deviations from the requirements that may have been introduced by another team member's

---

<sup>48</sup> Maruping, published in Smite et al(eds), (2010)

changes. The implementation of collective ownership is generally not a challenge for dispersed teams.

#### **3.9.1.2 Coding standards**

The benefits of coding standards highly affect the project efficiency and quality, especially for projects with distributed teams. The agreement upon certain standards (variable naming, conventions, data types or coding structures) gives the team a shared understanding for each other's work, making communication and coordination easier. As long as all team members are involved in the initial meetings setting up the standards, dispersed teams generally won't face any challenges with the implementation of coding standards practices.

#### **3.9.1.3 Use of metaphors**

The use of metaphor is concerned with team members developing a mental map of the system, resulting in an understanding on how one's input fits into the big picture, altogether working as a useful guide on how the system as a whole should operate. Temporally dispersed teams might face some challenges with the use of metaphors, largely linked to cultural differences since some team members might have other cultural references. This can be solved via synchronous communication, investing time in explaining the metaphors for each other to assure everyone has the same understanding of them.

#### **3.9.1.4 Simplicity of design**

Simple design means developing software functionality using the simplest coding structure possible. Among the benefits are reduced potential for defects, reducing the amount of effort needed to understand the code and facilitating for future changes to the code. These practices are critical in distributed projects since development efforts need to be coordinated between team members lacking shared context, making the management of communication and coordination much easier. Simple design practices can be utilized in different ways: by standards, minimizing the number of classes or functions etc.

#### **3.9.1.5 Sustainable pacing**

The point with sustainable pacing is to ensure a comfortable schedule for team members that can be maintained for the duration of the project including managing the timing of code development, testing and deliverables. Working overtime to finish before tight or impossible deadlines creates stress, exhaustion and burnout and increases the likelihood of defects being introduced or overlooked. Sustainable pacing is instead allowing developers to bring more energy to their work resulting in higher quality of the software being developed.

#### **3.9.1.6 Pair Programming**

Pair programming yields a number of benefits, including the ability to produce more high quality software code. Spatially or temporarily dispersed project teams can utilize

this practice by co-locating and then coordinating their work across sites. The major limitation is when a team is having an isolated member at a certain site, which makes it practically impossible to pair program if there aren't exceptionally good communication technologies available.

### **3.9.1.7 Continuous Integration and Unit Testing**

The two concepts of continuous integration and unit testing are often tightly intertwined. Integration yields several benefits by project teams becoming able to produce and maintain applications with a minimum of defects because changes are not committed to the production code until new code passes all functional tests. Discovered defects are traced to the changes they're linked to. It's also easier to incorporate changes or add functionality with minimal effort because of the iterative approach. Continuous integration is possible to implement across geographic and temporal boundaries as long as the responsibilities of each site are clear to make changes traceable and to avoid double-work.

### **3.9.1.8 Refactoring**

Refactoring reinforces the principle of simple design enabling development of efficient code that is easy to comprehend and reducing the potential for defects by eliminating duplicate code. These efforts to simplify the structure of the code also enable more efficient incorporation of changes resulting in reduced costs of making changes in later stages of the project life cycle. Also in this practice, testing and continuous integration might serve as a safety net against the introduction of defects. Refactoring should not present challenges for spatial, temporal or configurationally dispersed teams as long as the communication facilities are good.

### **3.9.1.9 Customer Involvement**

The benefits of having dedicated attention of a member of the customer organization are numerous, there among getting a better understanding of project requirements beyond the specification, better understanding of the business environment achieved, possibility to quickly resolve ambiguities and get immediate feedback on design issues, altogether resulting in increased likelihood that the software actually meets customer needs. Customer involvement shouldn't pose challenges for dispersed team as long as synchronous media is available.

### **3.9.1.10 Small Functional Releases**

This practice enables teams to focus on delivering the most important functionality first, other important components can then be added iteratively in subsequent releases. There's also flexibility to incorporate user feedback resulting in functional software yielding high customer satisfaction. For dispersed teams, small functional releases provide an opportunity to coordinate development of components since the project is decomposed into features and functionality. Critical features can be distributed among developers at different sites working in parallel, then become tested and integrated. As

long as one of the team sites are in proximity to the customer organization, this practice shouldn't present any major challenges to deploy small releases.

### **3.9.2 Globally dispersed Scrum practices<sup>49</sup>**

The daily Scrum meetings were meant to be face-to-face gatherings but in distributed projects, this might not be possible. Instead, the project team should aim for arranging circumstances as close to such situation as possible, for example by using virtual presence solutions, good quality videoconference connections or web cameras to make it possible to recognize each other and see each other's facial expressions. This is to make a more natural meeting situation, creating joint understanding and building team spirit. In case of video- or teleconferencing not being possible, for example due to technological issues or problems with spoken languages, using chat might be an option but not to recommend as much information is lost compared to videoconferencing.

If the team is dispersed over different countries, cultural differences may also have impact on the daily Scrum meetings, since people may find different things to be appropriate to report, for example revealing and discussing impediments is less natural in Asian cultures than in Scandinavia.

A good infrastructure for daily Scrums should be provided, letting meetings and communication to be easily set up. Asynchronous meetings and text-only meetings should be avoided. The team should also be actively practicing and discussing to find the optimal amount and type of information to report. To avoid the risk that team members try to hide problems leading to more problems in the future, it's important to create an open atmosphere to make it easy to raise problems and issues without fear. Discussion in small groups should also be encouraged.

If there is more than one Scrum team in a project, it's essential that they share information with each other, for example by having one team member from a team participating in the other team's daily Scrum meetings. If there's more than two or three teams, arranging Scrum-of-Scrum is a must to share information regarding what's happening in the teams, what challenges the team is facing and what kind of interconnection the teams' work has.

The use of sprints does not differ much in a distributed project than in a collocated project. Different holidays at the project sites might affect the project planning though.

### **3.9.3 Agile knowledge transfer<sup>50</sup>**

Many organizations have relied heavy on formal written communication and formal hand-overs in the belief that this would overcome the communication challenges but for most of these organizations this proved not to be sufficient. It rarely helps to add even more rigor and formality since documents, emails and contracts can only get you so far,

---

<sup>49</sup> Eckstein, published in Smite et al(eds), (2010)

<sup>50</sup> Kock&Sauer, published in Smite et al(eds), (2010)

instead the organization has to acknowledge the investment of effort that is needed to get collaboration to really work across distances. Co-location of development resources in agile practices is not a prerequisite but a luxury that rarely is accommodated in reality. Traditionally co-location has been used as a technique to improve communication in projects, but distributed project needs to find other ways of coping and *ignorance is not an option*.

To establish a correct and complete mix of skills at each project location, an extra emphasis on skill transfer and building is necessary. It's common to experience resistance from the onsite teams when facilitating such a skill transfer, likely to be caused by fear of downsizing locally.

Cultural diversity has to be coped with as well, the importance of this is often underestimated and most organizations are surprised by miscommunication caused by not understanding different cultures.

The cost of communication has to be weighed against the value added for the organization. Some strategies or tools for communication are in order of expensiveness (most cheap first): Instant messaging, low-end video-conferencing, wiki, discussion boards, pair programming, large conference calls, high-end video-conferencing and physical co-located meetings. In another dimension, the same tools sorted by assumed value for the project are(least to highest): Large conference calls, High-end video-conferencing, wiki, discussion boards, low-end video conferencing, instant messaging, pair programming and physical co-located meetings.

## 4 Methodology

To reach an adequate apprehension of the situation at Nordea PBIT, a series of steps has been followed. These steps are:

1. Performing interviews, at PBIT, and other internal and external stakeholders
2. Literature study: how do modern, effective IT-oriented organizations with off-shoring work?
3. Benchmarking: what is practiced at other IT-departments at Nordea?
4. Presence at the workplace, taking part of daily (Scrum) meetings, meetings and workshops with management, after work sessions etc.
5. Facilitating workshops with a subset of the staff to develop solution oriented and feasible actions in order to improve the found problem areas.

Interviews and workshop methodology is presented below, results of literature and contextual studies as well as benchmarking are to be found in the Literature review and Empirical review chapters.

### 4.1 Interviews

Within a period of six weeks altogether 23 interviews were performed: with 2 local managers, 9 project leaders, 8 developers 2 external IT managers at other departments at PBIT at Nordea, one upper manager and one representative for Change Management.

All interviews took place at Nordea's offices at Regeringsgatan 59 and Malmskillnadsgatan 23 in Stockholm except for one at the Nordea office in Taastrup, Copenhagen, Denmark. All interviews lasted for 40-60 minutes. All interviewees were presented a short description of the aim of the study and were encouraged to speak of anything feeling relevant to themselves in their roles as answering the set of questions. The answers provided were written down during the interview and immediately processed into digital form after finishing the meeting.

The interviewees were chosen to represent Portfolio and Advisory Solutions (further referred to as PAS) in best possible way, all employees available were interviewed, except for a few very recently employed. The external managers are representing Change Management, Group IT (upper management of PAS), Fund distribution Services and SAP and GL Production within Nordea IT. The aim is to collect a representative description of how goals, strategies etc. affect the product, relations and the development work, where both the collective and people matter.

The questionnaire was the same for all of the PAS Staff and slightly modified for the external managers, see appendices.

## 4.2 Workshops

Three solution- and improvement oriented workshops were held with a subset of the staff at Portfolio and advisory solutions. Together, these three workshops aimed to produce an action list, consisting of practical improvements of the processes. All three workshops had the same invited attendees, with exceptions of some absences for other meetings and similar. All workshops were held at Nordea at Regeringsgatan 59, Stockholm.

### 4.2.1 Workshop 1

The first of the three workshops was a whole-day event (10am - 4pm, with 1h break for lunch). The attendees were provided with an on beforehand prepared material consisting of four large posters taped to the walls, each with a different topic, namely:

1. Why does Nordea pursue internal IT-development with offshoring?
2. What is quality vs. effective development?
3. How do we work together (including offshore-personnel)?
4. How do we work with our users/customers?

Each of the blank posters had the most significant answers from the interviews presented on the left side and the “right”, official answers from Nordea managers on the right side, leaving a large empty area. After a short warm-up puzzle exercise, the attendees were introduced to the topics, rules, tasks and the results withdrawn from the interviews. The workshop was divided into four blocks, one for each topic, but with the same structure and tasks. The attendees were assigned the task to fill the empty area of the poster with comments and ideas regarding the distance between the both sides: why there was a difference, how it affects their work and what could be done about it. Tools provided were post it-notes, pens, pencils, crayons, various clipart-pictures and a noticeable amount of fika, fruit and snacks. After a certain amount of time, the post it-notes were worked through, letting everyone present and explain their contents. After the last block a short summary of the day was made.

Before the workshop, a short document with brief instructions and rules was sent out explaining the methodology and the schedule for the workshop. The rules were three:

1. Leave all cellphones unattended
2. Do not remove any post-it written by someone else
3. Never criticize other people’s opinions or suggestions at this workshop.

### 4.2.2 Workshop 2 and 3

In advance of the second workshop, the post it-notes from the first was collected and put together in two lists, an action list for those notes which already was considered as practical improvement suggestions, and a simple list of the rest, both sorted by topic. The program was also divided in four blocks, one for each topic, and all items on the lists for the respective topic were worked through, re-formulated, detailed and made into

concrete actions, with the aim to be completely practicable after the end of the project. Since the discussions were so good, the workshop took longer time than predicted and another occasion for working with the last topics was scheduled, becoming workshop number 3. This last workshop had the entirely same structure as number two, continuing were the last one left off, after a short review of it.

#### **4.2.3 Anonymity**

All names on persons have been removed from this report in order to sustain the interviewees' anonymity. This is done in order to gain answers on interview questions that might be considered as sensitive and to keep the integrity of Nordea somewhat more intact. The results from the interviews are presented in summarized form as representing the staff as a whole, quotes being presented without a named source.

## 5 Empirical review

The empirical study consists of three parts: the interviews, the workshops and observations. The results from the interviews are presented first ordered by topic together with an encapsulating text, followed by presentations of the most significant replies. The topics are covering the PAS staff's view on their work and the department as a whole, what effectiveness is, problem areas, suggested improvements and opinions on what actually works well. Further follows summaries of the workshop discussions, derived from every post it-note that was produced and the action list developed of these. The complete action list is presented in Appendix VI.

### 5.1 Why adapt internal development?

The essence of this thesis is to investigate the efficiency of internal IT-development at PAS. Concerning the aim of the strategy chosen, to keep the development in-house that is, there are more or less outspoken motivations for it. It's natural that people in different roles see different benefits of internal IT-development, but how individuals understand the goals in the same way as the management may also have to do with the degree of involvement and the level of information that reaches the individual staff member.

According to Change Management, the aim of Nordeas internal IT development is to

“Deliver solutions to internal and external customers which are of high quality and are cost effective despite the area of application, and also competitive.”<sup>51</sup>

An IT manager expresses that the expectation of Nordea IT is to

“Deliver efficiency, good solutions to a low cost, and that they are proactive and see technological possibilities.”<sup>52</sup>

On the question of why Nordea pursues internal IT development in difference to external, managers suggest several reasons<sup>53</sup>:

- Nordea wants to keep the competence and knowledge in the company
- It's easier to move resources between different departments if you know how the bank works
- Internal staff knows the culture
- It's less expensive than hiring (Swedish)consultants
- It enhances the loyalty within the employees
- Bank employees have a specific sense for handling sensitive information.
- The business department should be able to order what they want without extra, unnecessary functionality, not to buy too large systems

---

<sup>51</sup> Interview with Change Management representative

<sup>52</sup> Interview with upper IT manager

<sup>53</sup> Interviews with local manager at PAS, upper IT manager, Change Management representative

- It's cost effective since one can scale the amount and proportion spent on development and maintenance.

The suggested motivations given by the interviewees are varying. The most common answers suggest that the reason is to keep the competence (5 respondents), to be closer to the users (5) or that it is because bank specific knowledge is required (3). Other significant answers are the need for integration between systems, platforms and tools (3), lower costs(3) and the possibility to design solutions for the exact needs without making amends(3).

## 5.2 Why adapt offshore development?

The primary reason, according to management<sup>54</sup>, is to reduce costs: it's cheaper to work in India than hiring expensive consultants in Scandinavia, which would've been the alternative. The aim is to achieve the same delivery for a lower price, and we already do, at the same quality and time. Quantitative measurements are being made but yet no qualitative ones. Other benefits are to liberate the local resources, for example by placing maintenance activities offshore.

Change Management emphasizes on the profitability for Nordea as a whole, even if it doesn't pay off for the IT division<sup>55</sup>.

Most of the staff recognizes the ambition to reduce costs by offshoring (14 respondents) but many of the PAS employees fail to see any other benefits. In total 5 respondents see the offshore resources as less competent developers, and 6 of the respondents express doubt that it is cost reducing at all. One significant reason for this could be, as some of the interviewees suggest, that no one on manager level has ever motivated the offshoring strategy nor showed any number proofing the success. As good as everyone in the staff has suffered with problems connected with the offshore staff, communication or shared development processes and requests for better directions on how it's supposed to work out. The common view is one of discomfort, but the developers are glad to have lost the responsibility of managing the developed systems (6). Also, the scalability of people (6).

## 5.3 Quality and efficiency

Managers at PAS establishes that quality is achieved when<sup>56</sup>:

- Customer and supplier are satisfied with the product, e g when the deliverance meet the functional demands set by Business side (as customer) and the technical demands set by IT.
- Appropriate resources has been available during the whole project

---

<sup>54</sup> Interview with upper IT manager

<sup>55</sup> Interview with Change Management representative

<sup>56</sup> Interview with local manager

- Budget and time estimates has been followed, e g the estimates has been realistic and with most of the most important demands included from the start
- The product is tested and documented
- The users are informed and educated

Upper management defines quality in a system when it's nice to work with for the end users, a well thought out product developed with the users in mind. The system should also have the expected behavior and have good architectural qualities, be easy to understand for users and new developers. It's further claimed that: "we should stop the process of adding more functionality and produce smaller, better code instead. We should increase quality, to prioritize high quality instead of more functionality. At the moment, we add even the last of the requirements even if we don't have the discipline for it; the quality is not good enough".<sup>57</sup>

The customers may or may not have the same view upon quality as the developers have. Upper management suggests that the customers don't see how the development has been pursued, but they do see if the deliverance is praiseworthy.

Most of the respondents, both managers, project leaders and developers, mention three aspects as primal characteristics of quality: deliverance within the set time limit, delivering the right functionality and to the right price.

Whatever is the right functionality is a matter of further discussion. Some people assume it's equal to the demands specified by the customer (9), while others claim that the right functionality is what the customer truly wants, specified or not. This leads to quality meaning the ability to determine what the customer actually wants or needs. The deliverance should also be bug free and run without disturbance (11).

Another opinion, primary one of the developers, is that of the product being maintainable and facilitate for further development in the future.

In total 4 of the respondents believe that the customers at the business side of Nordea have the same apprehension of quality as IT has.

### **5.3.1 How to ensure quality in the finished product?**

The outlook the staff has on this topic is very varying. The two major trends are that quality is ensured by either test (acceptance- and unit tests) or having a sense of quality kept in mind during the whole development process. Another significant idea is to plan for it, achieving high quality by a good estimate of time, price and resources. Some people mention the development methods (Scrum, pair programming etc.), for example agile methodology stresses short decisions and developing a good product by often make new choices of strategy and reviewing requirements.

---

<sup>57</sup> Interview with upper IT manager

Few numerical measurements are being made on these aspects, but some are available such as the number of new defects, the number of story points achieved, predictability and productivity by the system called Sonar which is used for measuring the quality of code by “rules violation” and test coverage.

## 5.4 Offshoring and outsourcing

Offshoring is defined as the activity of an organization relocating some its internal resources or functions to a foreign country but keeping others at the original location. Nordea is pursuing this by relocating some roles within the IT-departments to other countries, there among to Bangalore, India. The roles concerned are mainly developers and testing resources. Scandinavian staff members are not moved to India, but local staff is hired at the offsite, in this case as consultants from a certain agency.

Outsourcing is concerned with having another organization taking over the full responsibility of an activity, function or project, not using Nordea’s own employees or other resources.

### 5.4.1 How do we work together offsite and onsite?

The current configuration according to PAS management of the collective work is <sup>58</sup>:

- Disturbance in production (Incidents/problems with severity 1 and 2) should be handled by offsite via second level support, application managers and offshore team leaders to make sure offsite personnel works with the right tasks.
- Incidents/problems with severity 3 to 5 is prioritized by the Improvement Decision Group (IDG), the offshore resources are primarily to solve the issues, via applications managers and offshore team leaders.
- Change requests are to be become redirected by applications managers and prioritized by IDG. Smaller tasks are driven by application managers and performed by offshore personnel via the offshore team leader if there are no Swedish resources available. Greater tasks become separate projects with a responsible driver or project leader. In the future, the aim is to mix on- and offsite people to a greater extent.

An upper management representative claims that it is the Scandinavian Nordea employees that have transferred their way of working to the offsite personnel. He also points out the problem that the staff within Nordea is not heterogeneous, different departments has their own ways to work and that makes it harder to arrange for offsite to work in the same way, when there are no single Nordea way. It’s about give-and-take, converging from different places into a common way of working, not only from India converging into Scandinavia, but from the other side as well<sup>59</sup>.

---

<sup>58</sup> Interview with local manager

<sup>59</sup> Interview with upper IT manager

Most of the PAS staff recognizes the collective work as onsite does all new development and offsite handle incident and maintenance (13). The differences in responses probably depend on how much contact with the Indian staff the respective respondent has. Different people choose different means to communicate but email, phone and using the chat service seems to be the common methods. No one directly mentions visits as a mean to discuss issues and communicate, but more or less regular travelling does occur.

Five of the respondents knows that the contact is supposed to go via the application managers, others suggests or guess that it is via PAS managers (1) or directly to the desired person in India. Notable is that some of the respondents prefer to contact their “favorite Indian” directly, without the detour via application managers or team leaders, while some is ascertained that that’s exactly what not to do, since it makes the division of labor uneven at the offsite resources.

## 5.5 Information, attitudes and involvement

It’s not a sure thing that the staff knows exactly why a specific system is being developed. As many as 9 of the respondents maintains that they aim to find out, to better know what to do and why, for making a better product better fit to the needs of the users and Nordea. Also, 3 respondents tell that they know what’s necessary for their work, but there might be more to know outside of the developers view. Only a few don’t care, and some people think they get to know to less about the aims for the systems.

Most of the developers and project leaders feel affiliated with the systems that one has taken part in development of, but not other systems developed at PAS. In spite of that, the group of onsite personnel is proud of each other and each other’s workings.

Work is about as easy or hard all the time. Some elements of frustration occur, there among certain bottlenecks, problems with environments for developing and missing routines for handling the offshore development.

A lot of the onsite staff is unsure of the tasks of the expert group. The single most common answer on the questions is “I don’t know what they do/are supposed to do” (7). Among the common speculations and guesses are that they have responsibility for system architecture(7), the entirety of the systems(4), and keeping track of dependencies between the systems handled by PAS(3). 3 people mention that the expert group have mandate to dismiss bad solutions. Most respondents are positive and look upon the group as a valuable resource, recognizing the need for someone to think of the greater trans-system issues.

## 5.6 PAS compared to others

All IT departments at Nordea are unique in that sense that they have their own systems to develop and maintain and their own style of development, within some set restrictions of Nordea. The idea of the many different IT departments is to have one per

business side; the IT departments are organized according to the different business departments with one IT department for each.<sup>60</sup>

Subsequently, PAS are unique, just like everyone else. Managers at sister departments finds PAS as a well-functioning unit, with good feedback from the employee satisfaction index (ESI), positive, seeing possibilities and not just problems, a typical Nordea IT department with a configuration of people with the same competences working together, seniors and juniors, thinking they're more unique than they actually are, being just like other IT departments, compared to them who are working in an entirely new and special way<sup>61</sup>.

The employees at PAS however feel like they have something that other departments don't have; a specific style. This is due to having a fairly young average age(5), working with newer technology than most of the others within Group IT does(5) and having an especially high level of competence(4). In addition to that, PAS employees are a tight group, having fun working together both consultants and employees (3) and being young in mind. One respondent describes PAS as sort of a "Nintendo department", making quick changes whenever they want, living with fun and games and less rules and regulations, playing hockey in the corridors.

The young mindset is said to take form as willingness to try new methods and tools and developers aiming to think at many aspects of how to make good products. It's clear that the developers have high trust in each other and each other's competences, with a sense that they work with a collection of the best developers at Nordea.

Some opinions differ, possibly depending on the experience and references of the respondents, having been to other internal IT departments or not. The direct question of "Is PAS a typical IT department at Nordea or within the IT branch?" gave a range of answers from "It's not a typical IT department" (5) to "It's definitely a typical IT department like anyone else" (2).

## 5.7 Feedback and evaluation

The view of the workings, aims and processes of PAS that the employees have is not a collectively established one. Many respondents can't think of any evaluative activities been held at PAS, e.g. talking about common goals and values, how PAS works, why and how they should work.

In total 6 respondents answers "Not so much" on the question on how much evaluation and feedback occur. 6 people say there's need for more of evaluation and feedback. The most common suggestion for what kind of feedback activities that actually occurs are the sprint retrospectives (7), even though they're more focusing on the process, not the product, and only is applicable for the specific project, not the overall workings of PAS. Other suggestions is the employee satisfaction index, PDD's every year and other ad

---

<sup>60</sup> Interview with representative for Change Management

<sup>61</sup> Interview with IT manager at sister department

hoc solutions as the team agreeing to give each other regular feedback. Some of the respondents expresses that the lack of feedback is a sign of everything's fine, if you hear something it's because something's wrong (3).

## 5.8 Available resources and support

The respondents had the opportunity to interpret support and resources, making them to think of what's most important to them personally. For some people it might be bottomless coffee cups and access to gym, for others feedback meetings or software and IT support. Some of the respondents think they're not given enough of support, especially in the figurative sense, others think it's enough since it's up to the employee to ask for what one needs, and they have the possibilities to do so.

No one complains at the access to practical things such as the coffee machine, gym, and other benefits such as parental leave and staff discounts at banking services, which are the same for all Nordea employees. The staff is also satisfied with the technological assets, software and hardware (8) and opportunities for education and taking courses when specific competences are needed (5). 4 people mention the possibility to increase the number of resources working on a project as a good support.

Areas of support that needs some improvement are concerning the external supporting functions, such as Nordic Processor, POSS and ITSS – the IT support.

## 5.9 Documentation, test and user involvement

Written documentation doesn't seem to be prioritized; instead the developers make sure to communicate a lot with each other in order to spread system knowledge. Some have heard of the different standards and guidelines that Nordea have, but no one seems to use them to any greater extent.

Two typical replies are "Not so much" (4) and "It's different in every project" (5). When documentation actually happens, it's via QC (3), Sharepoint (5), written in the program code (5) or the wiki (4).

Many people thinks that the users are not enough involved in the development or at least that they should be more involved (11) since it's the users who know better than anyone else what they need. At least 3 people claims that they've never met a user. The actual end users are not very visible in the development process since their interests and wishes are covered for by Change Management, acting like some sort of proxy. The staff disagree upon whether this is a good solution or not. Some thinks that it is beneficial to have an instance that collects opinions from the users and act as a spokespersons, other find them more as a detour, which might not interpret the users' needs correctly which leads to trust issues.

Since the end users works in the same company as the developers, the developers feel a certain responsibility for delivering a good product. The PAS staff is anxious to make

something good that makes the users happy and gets frustrated when they find obstacles for doing whatever needed to fulfill this.

## 5.10 Problems

A breakdown of the suggested problem areas of the processes at PAS shows that they are focused on four different areas: management/organization, customers/requirements, development processes and techniques and outsourcing/offshoring. Detailed suggestions are presented below under each area.

### *Management/organization*

- Frequent reorganizations make it hard to focus on development when the developer often has to stop and take in something new.
- No benchmarks are available, for example regarding
  - o What's the cost of finding errors
  - o Are offshoring cost effective
  - o Are Scrum development cost effective
  - o Do we reach higher quality by adapting agile methods(the user's don't care as long as it satisfies their needs)
  - o Do we have the right number of Indian consultants
- Managing requirements is a missing role at Nordea, developers write the requirements documents themselves.
  - o Business staff haven't time, knowledge etc. to work with the requirements.
  - o Developments haven't got it either
  - o Business staff (customers) don't know how testing should be managed. The testing processes are not defined either.
- Nordea is not an agile organization since it is restricted by financial demands. Effective Scrum demands balance between time, price and quality and the possibility to tweak these parameters not regarding firm financial limits.
- Lack of prioritization, the loudest instance will get the attention and everything else is pushed aside.
- Lack of organized support from management, feedback and evaluation.

### *Customers/requirements*

- There is no product owner
  - o Instead there are ten people having different parts of that responsibility, all these ten have of course their own agenda
  - o These people are located far from us, in Denmark, Norway and other places in Sweden.
- 
- The lack of a product owner leads to the product not getting enough of tests, no one is having the overlook on where the development is or should be heading.

- The customer is either not interested or don't have time, they don't handle their responsibility as they should to as a customer and manager of requirements.
- These issues make it hard to deliver the right product.

### *Development processes and techniques*

- Agile methodologies have focus on here and now, with a risk of losing aim of the goal.
- Since the requirements specifications are not good enough we have to rework them so one can understand what it is they want before we can send them to India.
- One cannot plan maintenance issues in sprints since errors are not predictable.
- There's need for a better equipped and competent release manager.
- There's need for some people that can monitor the hardware and running software.
- There's need for testing personnel who are familiar with the system.
- Lack of feedback in projects since the business side not engages in the development even though it is required for Scrum development. At the moment, every project has to work it out for itself how to handle it.
- Most of the focus is set on Java development, at the expense of database development. These areas of development have different interests and should perhaps interfere more.
- The Scrum oriented development suffers because of the developers and customers do not work together in the same environment.
- The "definition of done" still allows errors.
- Nordea has recently let all the testing personnel go, it is not clear who is to test the systems now.
- The actual users are very far away from the development. Developers are not allowed to contact them or the users are too busy or interested. This makes the development process hard to steer in the right direction since it is not obvious whether one is moving in the right direction and where the goal (the true needs of the real users) actually is.
- The documentation is arbitrary and spread out on different places. It's hard to find information on old projects and this is a growing problem for the future.
- When the wrong things are being developed, it's discovered too late or becomes expensive to change.
- A lot of the developers' time is placed on requirements analysis even though they don't have time, desire or knowledge to do it.
- Frequent problems with the testing environments eat a lot of time and create irritation.

### *Offshoring/outsourcing*

- The company which Nordea has outsourced all hardware managing to is unpopular described as a black hole where things go and disappear, no one knowing who they are, what they do, and why their deliverances take so much time and are incorrect.
- The distances, both to business side and to the staff located in India make it more difficult to develop a good product since it adds on language and cultural difficulties.
- One often forgets to set common start points where one assures that everyone agrees and can work on the same prerequisites.
- Parts of the onsite staff thinks it's up to the offsite staff to conform to Scandinavian Nordea's processes and methods but they have their own ways to work.
- The different mindsets make it hard to meet each other's expectations. An example of this is that the Indian staff usually doesn't pose questions when they fail to understand but work around instead.
- It's much harder to be sure that the other part fully understands when there are difficulties with communicating methods. One can't for example show things to the other part or make instant drawings.
- The working processes of Nordea are not appropriate for offshoring, one need to ensure good requirements specifications and can't assume everyone is familiar with the systems.
- Education is also an important matter now, one cannot assume everyone agrees or has the same competence and knowledge. There's need for guidelines making sure that these information reaches anyone who needs it. Communication needs to be over explicit.
- The geographical spread has led to less control and overview when certain parts of the development is never seen by the single developer for example knowing which error has occurred in a system, who works on what or even if anyone does.
- The spoken languages are two versions of bad English, which makes it hard to understand each other leading to misunderstandings.
- Offsite has a high staff turnover. Indian consultants have the habit of leaving their company as soon as someone else offers a few rupees more. The turnover leads to problems with knowledge sharing since one has to educate new people often when others are leaving.
- It's hard to get the right expectations on the offsite staff, many of the consultants don't seem to have the knowledge that we expected.
- There are no guidelines from management on how to handle the offshoring; it is up to PAS staff to figure it out themselves.
- More moments of disturbance and irritation leads to changes in the atmosphere, a bad mood.

- It's hard to keep knowledge when the people working with our systems are consultants that disappear after a while, especially the Indian consultants.
- Everybody has their own interpretation on how to handle the situation and some of those are contrarious.
- One doesn't know the Indian consultants, not by name or face, or what their competences are.
- Nordea has ordered the Indian staff to work Swedish hours, which has increased the costs and is eventually not good for the members of the Indian staff. One doesn't use the opportunity to have the Indian staff solve problems before the Swedish staff comes to work.

### 5.11 Things that work well

Different members of the PAS staff have different favorite methodologies. Scrum is appreciated among many of the developers but is also seen as incomplete and not compatible with the methodology of Nordea. Few people claims to see benefits with the Change management organization.

The concepts that are suggested as the most valuable are:

- Pair programming, to avoid having key persons for specific knowledge.
- The horizon is short and cursory. There's no need to panic over decisions for a long time.
- Benefits of Scrum:
  - o Makes it possible to split things in smaller pieces, one can see the end of each project and the deliverance is always near.
  - o Makes the customer more engaged, one can catch misunderstandings early in the process instead of finding them in the deliverance. It's well known that the requirements are always changing along the process, it's natural to not know everything in the beginning.
  - o Activates the dynamics within the team, helping each other and taking on responsibilities together.
  - o Provides better insight in what everyone is working with.
  - o Requirements are not set in stone from the beginning. It's nice to be able to satisfy the customers' wishes.
  - o Before work starts, one goes through it all and gains an overview.
  - o Seeing the goal pushes the development further towards it.
  - o Gives orderliness with a schedule, meeting every morning, regular demonstrations, deliverances and retrospectives.
- It's a good thing when there's a common methodology both within IT and business, having a common language.
- It's a privilege to have specific resources to handle business aspects (Change Management). We don't need to find this assistance ourselves.
- There's a flexibility of scaling resources with the offshore consultants.

- Maintenance is taken care of by the offsite. Not needing to mix maintenance and development makes it easier to make good time estimates.
- The possibility to change technique or methodology keeps the work interesting.

The interviewees were asked to name one thing that the employee team at PAS is especially good at. The single most common answers were “teamwork” and “high technological competence”. Other answers connect to this and elaborate this further:

- They help each other to grow
- High professionalism
- They are self-acting and experienced
- Engaged in what they do, taking initiatives and challenge the solutions
- Extremely flexible
- Makes things happen
- Loyal against each other and their tasks
- Service minded
- Humble
- Likes their job, good mood, good colleagues, which gives good results
- Have a will to develop, try new things and keep the high competence
- The best developers in the bank
- Helpful, eager to aid.
- Thinking on the users’ best
- Thinks further on quality aspects

## 5.12 Improvements suggested

The interviewees also got the opportunity to suggest improvements on how working processes should be in comparison to how they are today. The interviewees were asked to present the one most important change they’d like to make.

Many different suggestions came up, many propose ways to enhance a tighter connection with CM to gain better understanding of each other and get what’s needed to facilitate the development. The suggestions are presented without regard to their feasibility.

- Have the business side, the customers, closer and more engaged since it spurs us and giving them the opportunity to influence the deliverance they get immediately instead of months later.
- Move CM to our department and have them convinced it’s a good thing.
- Have an absolute priority which cannot be changed.
- Get a TV and web camera connected with offsite so we can go there and show things when we need to.
- Get resources for testing.
- Arrange guidelines for how to handle the cooperation with offside, with methodology and transferring.

- Everybody uses the same tools and standards.
- Let offsite start work before we do, at their local time.
- Get requirements analysts to work up the requirements.
- Not to get more developing resources, the demands would scale as well.
- Change the division of work between onsite and offsite; they haven't enough improvements and errors to work with at the moment which is ineffective.
- Use Kanban for error fixing, it's good because you can address them when you have time for it.
- Have cross country teams to enhance integration within the staff.
- Work on our mindset, we should adapt as well.
- Become "we" instead of "them and us".
- Get a more even work load over time, it's better for making good time estimates and for the staff as well.
- We would be more productive if we didn't do so much of testing. Test coverage doesn't mean that much, 50% more tests don't give a 50% better product.
- Invest in more analyzing and design before a project starts, to think it through to get the most out of the work. Currently, our scoops are too large.
- Stop with the outsourcing, it disunites our group and increases frustration. We need to focus on development and no other things that are not our job to do.
- Get more focus on testing: good working requirements documents, testing environments, competent testing resources. It's our best chance to affect the quality; validation and verification.

### 5.13 Workshop results

The three workshops resulted in a list of concrete and feasible actions to improve the problem areas. The complete action list is presented in Appendix VI. The four topics defined as the problem areas were treated in order and a number of improvement oriented actions were listed. Actions that were suggested as improvements for more than one area are presented once.

#### *Improve the internal development and offshoring*

- Visit other departments with experience of offshoring to share their knowledge
- Instruct SDM to act immediately when there's disappointment from us
- Always inform the staff about the journey after visits to India, as a routine, including what was done, what the aim was with the journey and gained experiences.
- Ask management to present numbers on outsourcing/offshoring to us so everyone can understand how they see how it should be done and what are expected of us.

- Talk more about cultural differences, eventually together with an external facilitator and lecturer. Both onsite and offsite staff should participate in this.
- Arrange TV-communication and introduce each other.
- Administer competence tests by web before and during interviews with new staff members. These tests can be written or bought and the interviews could be performed by videoconferences, the aim is to know we get the right people with the right competences.
- Invite new offsite staff to Sweden to be introduced to the onsite staff.
- Clearly define all routines, roles, responsibilities and authorizations so everyone knows. These should be as good as finished before changes are made or as soon as possible. For example the responsibility for recruitment, mandates, compilation of teams.
- Solve the problems with technical work environments in India by informing management and create a “task force” which solves issues.
- Enhance the team spirit, perhaps to evade the high level of people coming and leaving.
- Manage for knowledge sharing. Decide what needs to be documented, how, where and who’s responsibility it is. The goal is to make it less person-dependent and to keep it at a good level, not too much nor too less.
- Gain knowledge of best practice, for example by documenting and spreading project evaluations.
- Hold relevant lectures for each other, even Indian staff. Decide which lectures are needed and when they can be held. All interested should have the possibility to attend. The topics could be developing techniques or bank- and business knowledge.
- Find out or decide what views management (Business an IT) has on quality within IT. Book a meeting and agree.
- Take courses in bank- and business knowledge, find out if the old ones are still given and perhaps make them mandatory for the PAS staff.
- Make management motivate the salary trend.
- Show this list to management.

### *Improve quality and efficiency*

- Find out about the available standards and use them. Find out which ones are needed by whom and where new standards are needed. Require business cases. Provide eventual education needed, plan for it and provide time for them.
- Help business with formulation good requirements documents including effect goals for the requirements.
- Start with business cases and educate those who need it, even from the business side.
- Set a budget to improve architecture in products even if it doesn’t concern new functionality, perhaps a certain percentage for each project?

- Aim for small deliverances which actually reach production, no “big bangs”. This should be up to each project to decide on. Business and IT should together prioritize the functions so that the most important comes to production first.
- Set up surveillance and alarms in our systems. Nordic Processor and offsite should help with this, for example to make sure testing environments are running.
- Make sure all roles are filled in each project before it starts. Describe the responsibilities in technical design documents and people in resource documents.
- Make sure routines for testing are set, central on both high and low levels.
- Log the time spent on different activities to see what it is that consumes our time and finding the bottle necks.
- Pick up common technical solutions which doesn't connect to only one single project and develop these together.
- Work with functions and not systems.
- Include maintenance and other roles in the projects, for example test. Decide which other roles are necessary and when.
- Establish a channel for information from administrators to developers on how the systems are working.
- Find out which are the tools that causes problems and eventual alternative tools. Do we need for example support with these or education? Example: Weblogic.

### *Improve the teamwork*

- Create a common storing space and common tools. Find out which are our needs and requirements on such.
- Facilitate for regular meetings with onsite and offsite, for example once a month.
- Inform everyone about the expert groups work and aims.
- Present AMC/OFFSITE to everyone including role descriptions.
- Make it clear who owns the product; the same person cannot be the owner of all products as it is now.
- Arrange social events with CM.
- Push the business side to take some of the PM4U- and agile courses.
- Facilitate visits at the PB counselors to learn how they work with our systems.
- Arrange a lecture by a councilor to describe their work.
- Define what we want in the requirements specifications.
- Hire requirements analysts to unburden this from the developers.
- Create a common model for concepts, for internal use and towards business.
- Challenge offsite to cooperate internally to achieve a better base for Nordea with common technological initiatives for example.

- Make offsite more “Nordea”.
- Give offsite more interesting tasks; development tasks.

#### *Improve the cooperation with customers*

- Educate CM in basic IT and modeling to help them understand us better, for example by understanding the common conceptual model.
- Make sure all projects use a common documentation, templates, file structure etc in our communication with CM.
- Make CM come to us and inform about what their goals and responsibilities are.
- Achieve the right priority on incoming tasks.
- When a project is to start, allocate a percentage of time from CM/product owner and make sure there's commitment from the start until the end.
- CM should facilitate user contribution instead of substitute it with their own opinions.
- Political, internal country specific agendas should be taken care of at CM's level, provide PAS with the set priority.
- Have the project leaders at PAS agree on how to make technical design documents, when and why.

#### 5.14 Views on efficiency

The definition of efficiency is not obvious and everyone has their own view on what's efficient or not. Below are the individual opinions presented, formulated as answers on the question “What do you think is essential for effective internal IT-development?”.

- To have clearly defined assignments.
- To have a good view of the requirements in order to make a good analysis to allocate the right resources.
- To have the right knowledge within the resources.
- To be able to focus on developing.
- To have the customers participating in the development, to keep them updated and also to create a better sense of what to expect in the delivery, to what extent and to have the chance to make changes in time.
- To have testers participating as much as possible all the time.
- To know the expectations, mandates and limitations, it leads to better possibilities to make a good job.
- To work together in our group.
- To be able to give value for the money, making the whole department more effective.
- To make maintainable systems.
- When the customer knows what she wants and to be able to formulate good requirements.
- Efficient testing.

- To have a good development process on how to work towards production. It's currently too long and messy, with long regression tests and we find a lot of errors. We need to have shorter time to production.
- We have a large technical debt to pay. It's like Pandora's Box, when we open it we find lots and lots of defects and bad solutions that we have to take care of. Customers often prefer quick solutions which they see as more effective but it gives this kind of problems in the long run.
- To have a dialogue with business side is important, to sit together so IT can be a part of the ordinary work. In the modern world, IT is present everywhere.
- We could let anyone handle the drive but when we lose control and let them have demands on us we are lost. We need to keep the competence and to know how to take it all back if we want. It's not yet a problem; we still have control over the system.
- To have common ways of working and a common platform, set of systems, so everyone knows how to start when everything looks the same and there are set routines.
- To keep the high quality so one doesn't have to handle bugs a long time afterwards, taking time to get back.
- To have the same people around for a long time.
- To have a positive mind, at the moment the mindset is about us and them between business side and IT, both blaming the other side for problems. We shouldn't do that; we are striving towards the same goal after all.
- Predictability, to make the process easy to plan. Storypoints give predictability.
- It's necessary to have glue between customers and developers, some people who explains the business for technicians or the other way around.
- It's not very effective at the moment since new processes have been introduced and are not yet running as they should. Decisions should be quick so we can get to start working. The current process is slow started, a long time to wait until you can start to deliver something and the focus on costs is firm. We will be more effective when the new processes are working, and when we don't have to fix bugs.
- I would like to work according to Scrum since it would be efficient to be able to set goals, work towards them and change them. Everyone must contribute and that might be hard to reach. We should get less side tracks and lapses.
- With Java we use test driven development which is effective. What's not effective is the version handling in the database. In production, it's about getting the product delivered quickly. In the development, it could take its time, it should have high quality. In total it's about letting out as few errors as possible.
- To make the right things at the right time, it's harder than one thinks. It's easily happened that one gets stuck with irrelevant things; it's a challenge to

do everything in the right order. The right person should do the right things as well.

- When the customer is present, when we develop both requirements and solutions together.
- It's ineffective to send a lot of emails back and forth all the time but still not solving every issue.
- It's efficient when having all customers present at the same place, otherwise it's harder to communicate and it's harder to get to meet the customer when they aren't at the same floor in the same building.
- It's effective to not be bound to a specific methodology since sometimes another one might work better.
- It's effective to do the right things from the beginning which implies good planning. It saves time and the results become better. It's not necessary a bad thing not to produce a lot of code at once, if the code have higher quality.
- It would improve efficiency to measure the results of the offsite, what they do, if they do the right things and how much defects they produce compared to the onsite developers.
- We need better and more effective means to communicate. There are no routines, major reorganizations have been made and it feels like it's not even finished yet, some parts are still missing. Often you do not know who is supposed to do what.
- To have access to the right information needed to do your work, that is currently the problem with our customers. Agile development is good because if information is missing one can start to work on another task instead while waiting. It facilitates for good planning, one doesn't assume that everything will work smoothly, becoming prepared that information is missing or obstacles occur.
- Offshoring is good. We were afraid of losing jobs because of the Indian staff being better developers than us, but we have realized that that's not the case because they are not good, they're catastrophically bad. They sent us the best people first but when they left we got other ones that are only here to charge money. We should've moved the work at a slower pace and introduced and educated one developer at the time.
- To let go of the maintenance tasks as when we got the offsite staff.
- It's bad for the efficiency that the offsite staff is not good enough.
- It's bad for the efficiency that the means of communicating with the offsite won't work, with no possibilities and rooms to quickly and easily being able to share screens with each other.
- It would be more efficient to remove the bad developers at the offsite. We don't even know if they do anything at all, we can't notice.
- It's efficient when we reach a lot of business advantages for the money, as much as possible within reasonable time.
- I'm skeptical towards pair programming and test-driven development at this level. It's more efficient to work on your own; the problem is that the quality

might be lower. We need to find a compromise. If we shouldn't do it all the time we might get things done faster.

- It's efficient to finish everything planned, just to do a lot of things in a certain time is only rapidness and might be stressful.
- To have a good team, clear requirements and set standards to you know you are doing it right. Geographical distribution is tricky since it makes it hard to see how to work effective. Five offsite consultants seem to be slower than three of the onsite ones. The results get less good when you're looking at the clock all the time.
- You won't do a good job if you're not having fun. Having fun makes you produce ten times more!
- It's inefficient to not have testers here, it'll lead to low quality of the software. The trend in IT is to hire more testers, but Nordea is letting them of after 20 years, it's unintelligible.
- The testing management is ineffective. We used to have both testers and test managers in Sweden, but the Danish managers decided that that wasn't necessary since the Danish IT-departments only have test managers. But the difference is that in Denmark, that role includes performing the tests as well, it doesn't in Sweden so now we have no one testing. We tried with Indian testers, after six weeks three people hadn't found any defects, but when the customers ran their tests it took one hour to find four defects.
- Efficiency is to deliver what was agreed upon within the set time and to the set price.
- The lean process is ineffective since the developers don't use it.
- Testing is essential, when we find defects in one level of development by testing; we are assuring that they won't be left till the next level.
- We write way too much tests, many of them are overlapping; it's hard to see the benefits with that. Other projects without a great number of tests have the same quality anyway.
- There are risks with having Change management acting as a proxy between us and the users. That makes us even further away from the users and we have to trust CM to be representative for the real users. Other risks are that the people performing the tests know too little about the business or that they know too much and become advanced users instead of representing the real users' knowledge. Then they might ask for functionality that is not relevant for the common users.
- We are not agile anymore since we have no product owner. Our process states that we should set the requirements first, then designing and making estimates before we can start agile work. But agility doesn't matter if the requirements are already set. My speculation is that combining agile methods with a firm budget is ineffective.

## 6 Analysis

All this empirical data has been collected with the intent to draw conclusions on how to support management to create more effective processes. Some of the findings point out areas in need for improvement but the strong sides of Nordea PAS are also visible from the view of the interviewees.

To somewhat elucidate the characteristics of PAS the analysis below is in form of a SWOT-analysis. SWOT is a tool for making strategic overlooks over organizations by evaluating the strengths, weaknesses, opportunities and threats of the organization. After this is done, the objective of the organization can be set. Another benefit is that competitive advantages become clearer. The strengths and weaknesses are internal attributes of the organization and can often be affected. The opportunities and threats are external and harder to predict, the challenge are to manage these situations.

Below follows an outline of the SWOT's of Nordea PAS together with a set of suggested actions to manage these derived from the workshops' results.

### Strengths – don't lose them!

The strengths are the attributes that the organization should aim to preserve and develop further. The two most distinct strengths of PAS are concerned on the mentality. These have become visible both by the interview responses, workshops and from observing the work and the staff. The major strengths of Nordea PAS are:

- High competence – the staff members are both very professional and competent and also helpful towards each other which further enhances this. This is important because it poses an advantage for both this department and for Nordea as a whole raising the competitiveness. It's a strength since the competence level is not just high, it's kept high and increasing.
- Team spirit – having fun and trusting each other enhance the team work altogether by the staff members elevating each other.

It's important for the future prosperity of PAS to make sure to keep these strengths that are the core of it. One can of course gain other strengths as well if there are opportunities.

Both the high level of competence and team spirit affects the efficiency since they affect the image of the department itself and thereby the will to contribute.

The suggested actions in order to retain these strengths are:

- Install TV-based communication with India which would give a possibility to get to know each other better, making onsite and offsite closer to each other, enhancing the team spirit instead of breaking it down by not knowing or recognizing each other.

- Talk about cultural differences. There are obvious differences that sometimes cause difficulties. By discussing these issues onsite and offsite can come closer and work from both sides for a better communication and cooperation, supporting each other.
- Invite new offsite staff to Sweden to get introduced to Nordea and the onsite staff. This is to make the new staff members a part of the one team that PAS are.
- Arrange social events with CM. The team spirit should not just be within the PAS department, other stakeholders and partners should be included as well.
- Have competence tests before hiring offsite staff to make sure the new employee can add the right competence.

### Weaknesses – be aware of them!

Weaknesses could or could not be possible to affect, the importance lies within awareness. Weaknesses could be adjusted or turned in to strengths.

Three distinguishable weaknesses are noted at Nordea PAS. These are:

- The considerable distance to customers eventually makes it hard to hit the goal. One cannot just speak with the customers or informally ask for their opinions which make development to sort of a guessing game.
- The current confusion of roles, methods and goals might lead double work or that some tasks are not being taken care of.
- Deficient communication technology might lead misunderstandings between staff members which affects the system development in a negative manner, especially if rework is needed. Ensure that there are channels for quick and clear communication as much as possible.

The suggested actions in order to neutralize these weaknesses are:

- Write measurable requirements to make it clear what needs to be done and when it is done, avoid the misunderstandings by everyone involved agreeing on the requirements document.
- Use the available standards and tools. Standards are useless when everyone uses their own. Using standards also sets what to expect of each other, avoiding misunderstandings.
- Have everyone informed about how to work with offshoring and the expert group. By involving everyone team spirit increases. There's also a need for agreeing on a common strategy on how to handle offshoring. Again, involve people makes them feel more responsibility for their work.

- Create a common conceptual model to avoid misunderstandings in communication with internal and external partners. Arbitrary constructions increase confusion.
- Have everyone informed at a higher level about CM's aims and assignments, at the moment the trust for CM is too low.
- Clearly define all routines, roles and responsibilities before changes are made.
- Discuss quality aspects with business side to learn about each other's expectations. This should lead to more satisfied customers since developers will know what they value.
- Visit the advisors; shrink the distance to the users.
- Decide how testing should be performed to make quality measurable.
- Make sure all roles are filled before a project starts, including a specific person acting as product owner.
- Find requirement analysts to get good and clear requirements and liberate the developers.

#### Opportunities – make the most of them!

Opportunities are not always predictable but when they appear they are to be caught. Currently two important opportunities are available for PAS to make the most out of. These are:

- The many Indian resources –since management hasn't set any limitations for their use, PAS can have them help with anything they want. This is also about mentality, the staff shouldn't think of them as a burden to babysit or entertain but as what they are – an opportunity for extra resources at a considerable price.
- Change Management –as one of the interviewees mention; it's beneficial to have an organization to serve the developers with finding and collecting the users' and organization's needs served on a silver plate. This is also about mentality, not how to avoid them but how do we support them to be most beneficial for us.

Suggested actions to make the most out of these opportunities are:

- Educate CM in basic IT and modeling to help them understand us and our needs better.
- Get presentations of offsite including role descriptions to better know the possibilities of what they can do for PAS, together with PAS.

- Have SDM act immediately when there are issues, giving offsite staff a chance to do something better to PAS benefit.
- Achieve the right priority on incoming assignments. Discuss with customers to have both sides view on what's most important and what's feasible.
- Have all projects using a common documentation to communicate with CM to facilitate for everyone.
- Push the business side to take some of the PM4U courses so we can talk about the same things.
- Regular meetings between onsite and offsite in order to see each other, meet and recognize each other. Also, to get to know what everyone is working with and who is available for assignments.

### Threats – avoid them!

The last set of attributes is the threats which of course are to be kept in check. Three distinct threats are distinguished:

- Lack of trust for management, offsite staff and change management excavates the team spirit and makes cooperation an unnecessary burden which might lead to impaired efficiency, job satisfaction and lowered sense of responsibility.
- Losing knowledge by not having a strategy for knowledge sharing, documentation and storing of information makes it hard for future employees and projects to work efficiently on maintaining systems. It might also cause situations of re-inventing the wheel and repeating the same mistakes over again.
- Team spirit gets lost without involvement, both in decision and with integration with each other, as described above.

Suggested actions to avoid these threats are:

- Hold presentations after each visit to India to inform those who were left in Sweden about what was done, what the purpose was and what experiences was gained.
- Make management present numbers on the profitability of offshoring, feedback on efficiency and customer satisfaction. This is to make everyone more involved and encourage the staff when showing how they affect good results, showing them the importance of their work. This is also to motivate decisions that the staff are not involved in, which can be hard to understand for those not participating in the decision-making.
- Have other departments with experience of offshoring to share their experiences to PAS for example by holding a presentation. When someone else has

succeeded and found good routines, it might be unnecessary for PAS to go through all the pitfalls.

- Get a common disk for sharing documents and common tools to use for enhancing knowledge-sharing.
- Create a standard for documentation and knowledge sharing. It's important that it's also available outside PAS and is searchable.

## 7 Discussion

The definition of what quality is or is not is elusive. Even if adopting a specific definition, people have their own view, and within an IT-organization it gets very visible especially when customers and developers don't agree. In the case of Nordea PAS, there are even internal disagreements whether quality is to deliver what was requested or instead what's actually needed. Quality might relate to price or to efficiency, and efficiency is further a matter of definition. It might be the same as fast development or not. In addition to this, it's not obvious how to measure quality and Nordea has no standards on how to evaluate either quality or efficiency. At least, no one that the staff knows about.

It's uncertain whether Nordea looks upon itself as still a young organization or not, but later (indicated frequent) re-organizations seems to have shaped an unfinished organization, at least within the IT functions. The lack of routines, standards and definition of roles and responsibilities seem to have created insecurity and confusion within the staff, leading to that some process being more ineffective than necessary.

Nordea's goals for quality improvements during 2012 are concerning the bank as a whole, presuming to include customers, advisors, IT-staff and so on. The goals affect the work of IT, but the IT-staff themselves might have different needs than other groups. Since Nordea is not competing with IT as its core competence but banking services, the view upon IT within the organization might be slightly different than what a developer in other organizations might assume. Nordea's IT departments work as means to create the good banking services that is the selling point and are therefore primary seen as a part of the banking business instead of IT business.

Since one or two decades, agile methods are considered as the new normal of the IT business. What's becoming visible is that companies today have adapted agile methods, become familiar with these and grown to develop their own adaptations or methodologies. It might soon be time for a new paradigm shifts because of companies realizing that the formal agile methodologies don't hold in real practice and need to be customized anyways since it's often impossible to customize the organization to the methodology. Nordea is officially pursuing RUP methodology and when the day comes when RUP is considered obsolete; Nordea probably won't be able to abandon it for a few years. Major changes are of course seldom possible to perform overnight, there are too high costs associated with changes except for the monetary ones: education of staff, dissatisfaction with new tasks, new standards, documents and templates needed and so on. Nordea is not an agile organization but haven't either stated itself as one.

In the IT business, people are used to quick changes, but there are a lot of prerequisites that has to be met in order to perform a smooth change. As Eason suggests<sup>62</sup> the process of change is a delicate one, an organization has to be aware of what might affect the outcome.

---

<sup>62</sup> Eason, (1988), pages 44-49

Sommerville<sup>63</sup> states that the real cause of low quality within software is not poor management, inadequate processes or poor quality training but the struggling of organizations to be competitive which might lead to strict budgets or impossible schedules. Whether Nordea is agile or not doesn't matter in this sense, whichever methodology can be more or less wrong or right, but other variables such as external forces affect the whole organization and are very difficult to manage.

---

<sup>63</sup> Eason, (1988), pages 44-49

## 8 Conclusions

The aim with this study is to point out the most interesting areas of improvement at Nordea PBIT and Portfolio and Advisory Solutions in order to pursue more effective IT-development. These areas have been derived from empirical investigation and interviews with the staff and the results are presented in the analysis chapter above.

Nordea PB has stated a vision “to be the preferred Private Bank across all our markets - acknowledged for our people and creating superior value for our customers”. PB IT is of course contributing to this by “being a proactive and trustworthy IT Partner of Excellence, and is continuously investing in our ability to be more efficient” by the means of

- Fulfilling the business area's need for daily deliveries from IT, and enable them to execute their long-term business IT strategies
- Helping the business to reach their goals and objectives
- Enabling us to attract and retain motivated, competent and empowered employees who provide superior IT solutions

Eason<sup>64</sup> concludes that the benefits that one can derive from IT are cost reduction, productivity improvement, support improvement and organization enhancement. These benefits are practical implications that could contribute to such goals as mentioned above but which development methods that are chosen is up to the organization to decide. Nordea is not in the IT-business but have chosen to pursue IT-development anyways for various reasons. This implies that most of the organizations energy is invested in banking activities, IT-related activities are invested in only because they're supposed to enhance the core business.

As an organization, Nordea is not specifically agile, and it can in fact be claimed that Nordea by its specific processes is violating the essential agile principles by not being able to make quick changes, having the firm focus on costs and so on. On the other hand, it can be stated that it's the agile principles that are violating the essence of Nordea by not being fit to Nordea's interests. According to Sommerville, agile methods are successful only when the system development is performed within an organization where there is a clear commitment from the customer to become involved in the development process and where there are not a lot of external rules and regulations to affect the software<sup>65</sup>. Nordea has not these qualities but do not claim to either. It's obvious that the organization needs its customized development processes in order to fulfill its objectives.

In fact, the list of counter implementation strategies probably always is accomplished in obliviousness. To be able to benefit from effective development, the organization and the staff need to be aware so it doesn't halter itself. Eason states that the organization

---

<sup>64</sup> Eason, (1988), page 13

<sup>65</sup> Sommerville, (2011), page 59

needs to design change as much as technology<sup>66</sup> and Nordea's strategy is the organization of Change Management. In comparison to the Company Maturity Model, Nordea reaches level three, since there are a defined a set of processes and standards across the organization and they're applied as well. This is probably a great advantage when dealing with change issues but in practice it has developed certain risks for the organization. The problem is that even small changes for improvement might be hard to reach.

The virtual distances within Nordea are large, both between developers and users and between employees and management, except for the most local. There are signs that this fact affects the feeling of working together, in the same organization and towards the same goals. The IT-developers for example have the aim of supporting the users in best possible ways, but the users are people that the developers have never met nor have any sense of, only their representation in form of Change Management. On the other hand, most employees haven't met the uppermost management either, which makes it less likely to feel as a stakeholder. And when management also fails to inform the employees about the reasoning behind their intentions for Nordea, it's also decreases the likelihood of people appreciating it. Making the staff stakeholders will make them more willing to make new processes work, according to Sommerville<sup>67</sup>.

Another aspect of this is visible locally on PAS. The staff consists to a large share of consultants and offsite developers which makes the Nordea employees a minority. It should be in the interest of both Nordea and of PAS management to start digging where you stand and make sure the whole staff is involved in decisions and treated as a part of the success of the organization. Management having a positive and supportive mindset in certain issues will positively affect the staff, the willingness to adapt to changes and the efficiency. A negative mindset will probably support the opposite.

---

<sup>66</sup> Eason, (1988), page 33

<sup>67</sup> Sommerville, (2011), Chapter 26

## 9 Reference list

### 9.1 Literature

- [1] Dicken, Peter, (2007), Global Shift, 5<sup>th</sup> edition, SAGE Publications, London
- [2] Eason, Ken, (1988), Information Technology and Organisational Change, 1<sup>st</sup> edition, Taylor & Francis Ltd, London
- [3] Gross, John M., McInnis, Kenneth R., (2003), Kanban Made Simple: Demystifying and Applying Toyota's Legendary Manufacturing Process, 1<sup>st</sup> edition, AMACOM Books, New York, NY, USA
- [4] Smite, Darja, Moe, Nils Brede, Ågerfalk, Pär J(Eds), (2010), Agility Across Time and Space, 1<sup>st</sup> edition, Springer Heidelberg Dordrecht London New York
- [5] Sommerville, Ian, (2011), Software Engineering, Pearson Education Inc, Boston, USA

#### **Supportive literature not referred to in text:**

- Broberg, Hanna, (2006), Verksamhetsanpassade IT-stöd, Licentiate thesis, Institution for Computer Science, Linköping University.
- Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. Article published in IEEE Software, 18(2), 22–29.
- Gulliksen, Jan (2010), Användbar IT: användarcentrerad IT-utveckling i den statliga sektorn, Partsrådet, Stockholm
- Lárusdóttir, Marta Kristín, (2009), Listen to your users: the effect of usability evaluation on software development practice. Licentiate thesis, Uppsala University
- Nordström, Malin & Welanders, Tommy (2007), Mera affärsmässig förvaltningsstyrning: en bok om (system-)förvaltning, Dataföreningen, Stockholm
- Söderström, Jonas (2010). Jävla skitsystem!: hur en usel digital arbetsmiljö stressar oss på jobbet - och hur vi kan ta tillbaka kontrollen, Publit Sweden, Stockholm
- Yourdon, Edward, (2003), Death March, 2nd edition, Prentice Hall, Upper Saddle River, USA

### 9.2 Internet

- [6] Björkholm, Tomas (2009-06-09), What is best, Scrum or Kanban?, published at <http://www.agilejournal.com/articles/17-articles/1737-what-is-best-scrum-or-kanban>, (2012-07-25)

- [7] FC Garage, (2012), The Pomodoro Technique, [www.pomodortechnique.com](http://www.pomodortechnique.com), (2012-07-25)
- [8] Gusmán, Rebecka, (2012-03-20), Cobol är i ropet igen, published at <http://www.idg.se/2.1085/1.438900/cobol-ar-i-ropet-igen>, (2012-07-31)
- [9] Karlsson, Fredrik (2012-06-04), Här försvinner din arbetstid, published at [www.ingenjörskarriär.se/din\\_utveckling/article3487570.ece](http://www.ingenjörskarriär.se/din_utveckling/article3487570.ece), (2012-07-31)
- [10] Nordea AB, (2012), Facts and Figures, <http://www.nordea.com/About+Nordea/Nordea+overview/Facts+and+figures/1081354.html>, (2012-07-25)
- [11] Nordea AB, (2012), Merger history, <http://www.nordea.com/About+Nordea/Nordea+overview/History/Merger+history/1472963.html>, (2012-07-25)
- [12] Rosengren, Lina, (2012-05-17), published at <http://www.idg.se/2.1085/1.448483/indiska-outsourcing-leverantorer-bast-pa-kundnojdhet>, collected 2012-07-31
- [13] Wikipedia, (2012), Capability Maturity Model Integration, [http://sv.wikipedia.org/wiki/Capability\\_Maturity\\_Model\\_Integration](http://sv.wikipedia.org/wiki/Capability_Maturity_Model_Integration), (2012-07-25)

## 10 Appendix I: Questionnaire 1, PAS Staff

- Vad gör du/vad är din roll?
- Hur länge har du jobbat här?
- Har du jobbat på en intern IT-avdelning och/eller med offshoring förut?
- Vad tycker du bör känneteckna Effektiv intern IT-utveckling?
  - Varför är intern IT-utveckling bättre för Nordea än att köpa extern utveckling?
  - Varför är intern IT-utveckling med offshoring bäst för Nordea?
- Vad betyder ”att leverera kvalitet” för dig?
- Om du skulle gissa, vad betyder ”att leverera kvalitet” för era beställare?
- Hur ser en typisk arbetsvecka ut för dig?
- Vet du alltid varför ni gör ett system? Ex. för att spara pengar eller för att öka användandet av systemet.
- Vad är det bästa med sättet att arbeta?
- Vad anser du vara de största problemen i dagens sätt att arbeta för just dig?
- Hur borde arbetet se ut i jämförelse med hur det ser ut idag, förslag? Om du bara fick ändra en sak, så som det fungerar idag, vad skulle det vara?
- Vad är du mest stolt över att ha utvecklat för Nordea PB? Varför?
- När är det enkelt att gå till jobbet? Vad händer på jobbet då?
- Om du fick lyfta fram en sak som teamet är riktigt, riktigt bra på – vad skulle det vara?
- Vilket stöd/resurser ges för det arbete? – upplevs det som tillräckligt
- Finns det något som skiljer *Nordea PB Portfolio and advisory solutions* från liknande verksamheter?
- Har ni någon återkoppling/utvärdering? Hur ofta? Anses det värdefullt?
- Vilka är användarna? Hur är de delaktiga?
- Hur säkerställer ni kvalitet i er utveckling?
- Vad är expertgruppens uppgifter?
- Exempel på hur expertgruppen har påverkat produkt/process?

## 11 Appendix II: Questionnaire 2, External managers

- Vad gör du/vad är din roll?
- Hur länge har du jobbat här?
- Har du jobbat på en intern IT-avdelning och/eller med offshoring förut?
- Vad tycker du bör känneteckna Effektiv intern IT-utveckling?
  - Varför är intern IT-utveckling bättre för Nordea än att köpa extern utveckling?
  - Varför är intern IT-utveckling med offshoring bäst för Nordea?
- Vad betyder ”att leverera kvalitet” för dig?
- Om du skulle gissa, vad betyder ”att leverera kvalitet” för era beställare?
- Hur ser en typisk arbetsvecka ut för dig?
- Vet du alltid varför ni gör ett system? Ex. för att spara pengar eller för att öka användandet av systemet.
- Vad är det bästa med sättet att arbeta?
- Vad anser du vara de största problemen i dagens sätt att arbeta för just dig?
- Hur borde arbetet se ut i jämförelse med hur det ser ut idag, förslag? Om du bara fick ändra en sak, så som det fungerar idag, vad skulle det vara?
- Vad är du mest stolt över att ha utvecklat för Nordea PB? Varför?
- När är det enkelt att gå till jobbet? Vad händer på jobbet då?
- Om du fick lyfta fram en sak som teamet är riktigt, riktigt bra på – vad skulle det vara?
- Vilket stöd/resurser ges för det arbete? – upplevs det som tillräckligt
- Finns det något som skiljer din avdelning från liknande verksamheter?
- Finns det något som skiljer *Nordea PB Portfolio and advisory solutions* från liknande verksamheter?
- Har ni någon återkoppling/utvärdering? Hur ofta? Anses det värdefullt?
- Vilka är användarna? Hur är de delaktiga?
- Hur säkerställer ni kvalitet i er utveckling?

## 12 Appendix III: Questionnaire 3, Upper manager

Vad är din roll på Nordea?

Vad är målsättningen med Nordeas IT-utveckling?

Varför är intern IT-utveckling bättre för Nordea än att köpa extern utveckling?

Vad konkurrerar Nordea med inom IT-branschen?

Vad förväntar sig Nordea att de olika IT-avdelningarna levererar?

Vad betyder kvalitet?

Vad betyder det för beställarna?

Hur säkerställer ni kvalitet i utvecklingen?

Vad är det för skillnad på IT:s och CM:s ansvar?

Vad tycker du definierar effektiv IT-utveckling?

Mäter ni effektivitet på något sätt?

Varför har ni valt att ägna er åt offshoring?

Är offshoringen lönsam?

## 13 Appendix IV: Questionnaire 4, Change Management

Vad tycker du bör känneteckna effektiv intern IT-utveckling?

Varför intern IT-utveckling?

Varför finns denna uppdelning av olika IT-avdelningar?

Skiljer sig PAS på något sätt i jämförelse med andra IT-avdelningar?

Varför ägnar sig Nordea åt offshoring?

Vad är det för skillnad på offshoring och outsourcing?

Vad innebär kvalitet?

Hur ska man säkerställa kvalitet i utvecklingen?

Hur är tanken att arbetsfördelning och kommunikation med offsite ska se ut?

Vilka problem upplever ni med offshoring-strategin?

Vilka är användarna och hur är de delaktiga?

Hur bra är CM på att representera användarna?

Finns det något Nordea borde ändra på jämfört med hur det ser ut idag?

## 14 Appendix V: Practical tips

Practical tips for agile projects in globally dispersed projects collected from Agility across time and space.

1. For highly temporally distributed project teams it would probably be beneficial to hold periodic team meetings to discuss the story boards involved in the next release of the software.
2. Planning game meetings will be more effective if conducted using synchronous communication such as telephone or videoconference. Chat software may also work but can be prone to miscommunication. With chat software it may also take longer for team members to reach a common understanding of issues.
3. If you are considering implementing the collective ownership practice, communicate the roles and responsibilities clearly during the early phases of the project. Consider having a team meeting involving all sites for this discussion.
4. Clearly outline the norms surrounding the behavior encouraged by collective ownership. For instance, if any member can modify any part of the code at any time, it would be important that other team members be notified when such a change is being effected. If not, then safe-guards need to be put in place to prevent defects from inadvertently being introduced.
5. Coding standards should be discussed and established early in the software project lifecycle. Managers should make sure that all members of the project team participate and understand the standards that will guide their work.
6. Consider the composition of your team when implementing the metaphor practice. Are all team members familiar with the metaphor? Does the metaphor hold the same meaning for all team members? This may be especially important if your team has members in another country that is culturally different from yours.
7. Managers should encourage project teams to implement the simple design practice early in the project life cycle. Adherence to, and reinforcement of, this practice will facilitate behaviors such as refactoring that will yield benefits for the duration of the project.
8. The simple design practice will be especially beneficial in project teams that are highly temporally and/or configurationally dispersed. These types of project teams face considerable coordination challenges. The simple design practice reduces complexity in the structure of the software code, making coordination more manageable.
9. Although control over sustainable pacing is largely determined by project parameters and customer deadlines, managers can enhance the ability to manage this practice through the design of the team. Consider composing the team of members who are located across different time zones. Alternatively managers can compose larger teams to spread the workload. However, the benefits of this approach need to be weighed against the added cost of additional employees devoted to the project.

10. When implementing continuous integration with multiple sites involved, it is important to be very clear about the roles and responsibilities of each site. It is also important to be explicit about where handoffs will occur. It is preferable for handoffs to occur between sites because each site then has a clear understanding of its role and responsibilities.
11. Take great care in implementing refactoring across geographically distant sites. Managers should ensure that the implementation of this practice is coupled with systematic checks through testing and continuous integration. This will reduce the need for developers to notify each other when making enhancements to the software code.
12. To the extent possible, managers should ensure that at least one project team site is co-located with the customer. This will facilitate better customer-to-project team knowledge sharing through face-to-face interaction. Synchronous communication media can be used to facilitate site-to-site transfer of information.
13. Managers should also try to arrange periodic team meetings that involve the client member. This will ensure that developers at other sites remain in tune with the customer's needs as the project progresses.
14. Small releases can be an effective tool for coordinating developer work across time and space. The key to the effective implementation of this practice is to prioritize and distribute the development of the core features and functionality across the geographic sites involved. This will ensure that development efforts are focused on a core set of features and functionality at a time. This will also make it easier for developers at different sites to coordinate their schedules for delivering required functionality. This is much more difficult to orchestrate when priorities across sites are not aligned.
15. It is important to create a common underlying culture through extensive training within the organization, albeit with local cultural variations. The investment is significant, but the rewards are proportionally greater because it enables individuals to experience the unique cultures of all the nationalities within the organization; enables leaders within the organization to be teachers; creates a common base of work practices that individuals can then expand from; and emphasizes the importance and relevance of the local operations
16. Given India's premier position in the software services market, the available talent base is large but the true skill set of a given individual often varies from the picture presented by their resume. Experience suggests that a rigorous recruitment process should pay attention to both technical competence and cultural fit.
17. The majority of experienced people in the Indian job market have been trained on plan-based development approaches within organizations with high process maturity (as assessed using the CMM or CMMI framework). Experience shows that when experienced people are able combine the discipline of generating the

requisite process artifacts at the appropriate level of detail, with the technical rigor associated with agile software development, they act as the change agents that drive successful agile adoption.

18. A large number of people in the Indian job market are either used to a hierarchical governance structure with controlled customer access (if they come from plan-based organizations), or are inexperienced and have to be shielded from the customer. The switch to a more fluid development approach requires adaptability on the part of the individual as well as transparency on the part of the team. Mentoring of people either by their peers or by their more experienced managers is critical to effective agile adoption
19. Having senior leadership understand the dynamics of using agile methods allows them to articulate a realizable vision, and provides an anchor point for assessing progress towards achieving that vision.
20. A large number of companies starting their operation in India leverage the capabilities of consultants drawn from other organizations. By clearly defining roles and expectations of consultants and employees, BankCo has enabled a smooth culture transition.
21. Having a common project manager working across two different development approaches provides a means of creating a shared understanding between the teams, and driving towards a common baseline process.
22. An agile coach with experience in working in a global market is critical to successful adoption and sustainment of agile methods. In summary, the key success factors to successful implementation of agile development in the studied Indian companies were:
  - a. Designing a human capital strategy that supports growing an agile-infected culture
  - b. Creating a shared language to communicate within and across organizational boundaries
  - c. Finding a balance between experienced and inexperienced personnel to ensure effective project management
  - d. Establishing formal and informal organizational learning mechanisms
  - e. Mentoring to institutionalize work practices
  - f. Crafting incentives to increase adoption of agile practices
23. Install always-on web cams and big screens at informal locations, such as by the coffee machine at each site. This way, it is quick and easy for the teams to discuss issues and they get the benefit of visual body language (if picture quality is good) and tone of voice. Most laptops come with web cams these days—those should be used. Buy one if they don't—they are cheap and easy to use. Software that supports video calls is widely available and you only need Internet access to communicate. Simple collaboration tools, like Wikis and discussion boards have also proved to be useful when working across distances.
24. Spend money on travel; be at the other location for an extended period of time; 2–3 months in order to really know the people. When spending money on travel, consider the following:

- a. Avoid spending all the travel budget on managers—although unfortunately that seems to be the normal priority.
  - b. Prioritize travel to those who do the actual work in order to establish a good relationship with the other parties before working distributed.
  - c. Travel from offshore location to onshore—the psychological effect of commitment to the organization and the project is usually stronger that way as you get a stronger sense of being part of something larger—rather than being a province.
  - d. There is a risk of losing people after their “journey of a lifetime”—in that case consider a requirement to work for 2 years locally before being awarded travel.
25. Show that you care about people on the other side by celebrating birthdays, being respectful of local holidays, showing pictures of those on the other side—small things that matters significantly.

## 15 Appendix VI: The action list

Prio Kommentar

## Varför har Nordea intern IT-utveckling med offshoring?

**Besöka Fonda och andra avdelningar med erfarenhet kring offshoring delar med sig av sina erfarenheter.**

**SDM(Prashant) måste agera direkt vid missnöje från oss. Dvs. skapa en instruktion.**

**Få information efter Indienbesök (Vad gjordes, mål, erfarenheter). Gör det till en del av rutinen runt sådana resor.**

**Management borde presentera siffror runt outsourcing/offshoring och hur de ser att det ska bedrivas på ett bra sätt, vad förväntas av oss så att VI förstår. Begär möte om detta.**

## Prata om hur man arbetar med kulturskillnader. Både vi och indien deltar. Extern facilitator och föreläsare?

## TV-kommunikation. Introducera varandra.

**Tillämpa kunskapstester (via webben) innan, under intervjuer med mera. Köpa färdiga tester? Förbereda frågor? Köra intervjuer via onlinemöten (bild). Vi MÅSTE få rätt personer. Ny offshoringpersonal bör komma till Sverige och introduceras.**

**Rutiner, roller, ansvar (bla mellan offshore och onshore), befogenheter, uppföljning bör vara definierade till stor del innan förändring genomförs, annars så snabbt som möjligt. Ex. ansvar för rekrytering, mandat - team sammansättningar.**

**Lös problemen med Indiens tekniska arbetsmiljö genom att informera om och lyfta problematiken till KIM(?). Förslag, "Task force" som gör åtgärder.**

**Teamkänsla. Folk kommer och går. Hur gör vi?  
Knowledge sharing - vad behövs dokumenteras?  
Hur? Var ska det finnas? Ansvar? Metatagging?  
Mål: Mindre personberoende. OBS! Håll  
dokumentationen på lagom nivå! Best practice  
(efter utvärdering av projekt?)**

**Håll interna föreläsningar för varandra (även Indien) om, för oss, relevanta ämnen. Vilka föreläsningar behövs? När? Alla intresserade ska få gå... Ex. utveckelmöten, Patriks infomöte, bank- och affärskunskap**

Jens håller på med detta.

1

Vi blir piloter i ett  
initiativ för att lösa  
detta.

Vad anser Marti och Jesper om IT-kvalité? Boka infomöte...

Utvärderingar efter varje projekt - nya Best Practice?

Gå utbildning på intranätet i bank- och affärskunskap (Obligatorisk för alla?) Finns dom kvar?

Be ledning att motivera löneutveckling.

Visa denna actionlistan för Marti.

Kicki

Vad innebär kvalitet vs effektiv utveckling?

Använd de standarder som vi har och följ dem!

Vilka är det? Vem behöver känna till vad?

Behövs nya standarder skapas, vilka? Begär

Business case. Ex. PM4U. Behövs

utbildningsinsatser för teamet/vissa personer?

OBS! Planera och ge det tid.

Hjälp business att formulera bra krav för oss

utvecklare. Glöm inte effektmål för kravet.

Börja med Business caset. Utbilda alla som

behövs (även de runt omkring oss, läs

"Business")

Budget för att förbättra arkitekturen i

produkten även om det INTE innebär nya

funktioner. Budgetera en %-sats för alla

projekten?

Små leveranser som går i produktion. Inga "big

bang"-leveranser. Upp till respektive projekt att

bestämma detta. Prioritering av funktionerna,

så att det viktigaste går i produktion snabbt.

Business + IT.

Övervakning och larm i våra system. NP,

offshore ska hjälpa till med övervakning av

system. Ex. kolla om testmiljöer är uppe.

Alla roller besatta i projekt från start. Beskriv

ansvar i Tekniska Design Dokumentet och

personer i resursdokumentet.

Komma överens om testning. Verkar vara

centralt. På både hög och låg nivå.

Logga tid (tidrapport vid sidan om). Vad lägger

vi verkligen tid på? Vad är flaskhalsarna?

Gemensamma tekniska lösningar, som inte

tillhör bara ett projekt, ska fångas upp och

verkligen göras gemensamt.

Jobba med funktioner och inte system.

Inkludera förvaltning/förvaltningsbarhet i

projekten. Även andra roller, ex. test. Vilka mer

roller behövs involveras? När? Dvs. metodik.

Ida

Som Jens mail

Expertgruppen

**Info från förvaltning till oss utvecklare - hur går våra system? Hur ofta? På vilket sätt?  
Vilka verktyg har vi problem med? Finns alternativ? Utbildning? Support? Ex. Weblogic**

Hur arbetar vi ihop(inklusive offshorepersonalen)?

**Gemensam disk, lagringsyta (sharepoint idag) och verktyg. Ta fram PAS krav på detta!  
Regelbundna möten Sverige och Indien(1 gång i månaden?)**

**Informera alla om expertgruppen.**

**Presentation av AMC/EDC à Jens och Magnus inklusive rollbeskrivningar.**

**Tydliggöra vem som är produktägare. Frode kan INTE vara för alla.**

**Sociala evenemang med CM**

**Uppmana business att gå vissa av PM4U-kurserna (och Agile).**

**Studiebesök hos rådgivare. Hur jobbar de med våra system.**

**Få en presentation av en rådgivare. Göra en typisk rådgivarsession och en vanlig dag.**

**Vi behöver definiera vad vi vill ha i en kravspecifikation.**

**Vi behöver kravanalytiker för att avlasta utvecklare och ta fram bättre krav.**

**Skapa gemensam begreppsmodell (mot business och internt hos oss)**

**Uppmana EDC att samarbeta för att få till en bättre Nordea-bas, gemensamma teknik initiativ, etc. Uppmana Kim.**

**Få OFFSITE att bli Nordea!**

**Mer utvecklingsuppgifter till EDC-resurser -> roligare jobb.**

Hur arbetar vi med våra beställare/användare?

**Utbilda CM i grundläggande IT och modellering etc så att de förstår oss bättre. Ex. förstå vår gemensamma begreppsmodell.**

**Alla uppdrag ska använda gemensam dokumentation, mallar, filstruktur med mera i vår kommunikation med CM.**

**Jag vill veta CM:s ansvar och arbetsuppgifter. De kommer till oss och beskriver detta.**

**Ordningen på inkommande uppdrag. Vi måste få rätt prioritering på våra uppdrag.**

Holgers bag!

**Vid projektuppstart, tilldela % i tid från  
CM/Produktägare. Och committment från  
början till slut!!!**

**CM bör facilitera användarmedverkan snarare  
än att ersätta den med egna åsikter**

**Politiska, interna landspecifika agendor bör tas  
om hand på CM nivå - vi vill ha en slutgiltig  
prioritering.**

**PL-möte: Hur gör vi ett tekniskt  
designdokument? När gör vi det? Varför?**