School of Computing
Blekinge Institute of Technology

Dysfunctional aspects of Software Development

An analysis of how lip-service, deception and organisational politics may side-track the result of well-intended methodologies

Mikael Gislén

Thesis submitted for completion of Master of Science (60 credits)
Main field of study: Computer Science
Specialization: Informatics

June 2012

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science (60 credits) in Computer Science with specialization in Informatics. The thesis is equivalent to 10 weeks of full time studies.

Contact Information:
Author:
Mikael Gislén
E-mail: mikael_gislen@yahoo.com

University advisor:
Hans Tap, Tekn. Dr.
School of Computing, Blekinge Institute of Technology

School of Computing
Blekinge Institute of Technology
SE-371 41 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : + 46 455 38 50 57

ABSTRACT

This paper tries to identify and understand the human social obstacles for developing quality software. These include lip-service, cutting corners, deception and effects of internal politics. These obstacles can undermine the good intentions behind the software methodologies. The paper draws from the literature in different disciplines and uses an ethnographic research methodology to create a rich picture of the concerning aspects in the framework of one software development company in India. What stands out among the findings is that internal audits have mainly focused on finding errors in documentation procedures but study of the actual practices has often been shallow. In addition the understanding of business risks by the internal auditors has often been weak.

Context. The human based obstacles affecting Software Development Methodology analysed in the context of an ISO 9000 quality system in an Indian Software Development company working mainly with Swedish companies.

Objectives. Identifying and increasing the understanding of intrinsic negative social aspects such as lip service, cheating and politics which are affecting the results of Software Development Methodologies and if possible suggesting some means to mitigate them. In particular to create a deeper understanding of why people cheat and pay lip service to methodologies and to try to understand the political aspects of methodology and quality systems. There are other positive social aspects, but they are not considered here since the objective is about understanding the negative aspects and possibly mitigate them.

Methods. Ethnographic research using analysis of ISO 9000 and design artefacts, semi-structured interviews, participation in internal audits,

Results. Most focus in audits was on documentation and much less focus was on underlying methodologies, some indications of lip-service to process and processes were also mainly managed on a higher level in the organization while the understanding and practices were less well established on lower level. It was hard to get a grip on the internal political aspects since the perception of the subject in the informants' view was that it is mainly malicious and therefore embarrassing to speak about. Some conflicts between internal quality goals and customers' needs were also identified.

Conclusions. An ethnographic research methodology gives a rich picture. The analysis gives deeper understanding of the problem areas, but not necessary solutions. The author suggests that at the heart of the problem is a difference in world view. Software professionals generally tend to resolve [technical] problems using a reductionist approach, while these intricate challenges cannot easily be resolved by this approach. A more holistic systemic approach is required and while the software methodology is useful to structure the development it does not resolve these dysfunctions. They have to be resolved on another level. It was also found that further studies are required in particular to better understand Internal politics, the effect of Positive and Negative Incentives, the effect of software metrics on quality performance and subjectivity in customers' perception and expectation.

Keywords: Agile Software Development, Computer Supported Cooperative Work, Cultural Studies, Ethnography, Global Software Development, Human Computer Interaction, Information Systems, Internal Politics, ISO 9000, Micro Economics, Participatory Design, Psychology, Quality Systems, Software Engineering, Software Process, Software Quality, TickIT, The Human Role In Software Development

CONTENTS

1 INTRODUCTION 5

1.1 BACKGROUND 5
1.2 SOFTWARE DEVELOPMENT METHODOLOGIES 5
1.3 PURPOSE OF A SOFTWARE DESIGN METHODOLOGY 6
1.4 THE DARKER SIDE OF QUALITY SYSTEMS AND DEVELOPMENT PROCESSES 6
1.4.1 Cutting corners and deceptions 6
1.4.2 Organisational Politics 7
1.4.3 Agile Development 7
1.5 AIMS AND OBJECTIVES 7

2 LITERATURE STUDY 8

2.1 SELECTION OF LITERATURE 8
2.2 FORMAL QUALITY SYSTEMS 8
2.3 THE FLIP SIDE OF FORMAL QUALITY SYSTEMS 9
2.4 MOTIVES AND INCENTIVES 9
2.5 WHAT IS THE EFFECT OF USING METRICS 9
2.5.1 What about the agile community? 10
2.6 IS MANAGEMENT INFLUENCING SAMPLING FOR AUDIT? 10
2.7 POLITICS AND CULTURE 10

3 RESEARCH METHODOLOGY 13

3.1 ETHNOGRAPHIC RESEARCH 13
3.2 VALIDITY 14

4 THE STUDY 15

4.1 PROCESS DOCUMENTS 15
4.2 INTERVIEWS WITH STAFF 16
4.2.1 Definition of Software Quality 16
4.2.2 Purpose of methodologies 17
4.2.3 Familiarity with different methodologies 17
4.2.4 Advantages and disadvantages 17
4.2.5 Avoidance or missing to do things 17
4.2.6 Regarding why we have not found some things in audits 17
4.2.7 Organisational Politics 17
4.2.8 Most common dysfunctions 18
4.2.9 Review of NCs 18
4.3 ACTIVE PARTICIPATION IN AN INTERNAL AUDIT 19
4.4 CONDUCTING AN AUDIT 19
4.5 REVIEW OF CLASS DESIGN AND DESIGN DOCUMENT 20
4.6 INTERVIEW WITH THE EXTERNAL AUDITOR 20
4.6.1 Social aspects 21
4.6.2 Lack of knowledge of process by production staff 21
4.6.3 Lack of competence by internal auditors 21
4.6.4 Lip service to process 21
4.6.5 Organisational Politics 21
4.6.6 Other challenges 21

5 DISCUSSION 23

6 FURTHER STUDIES & SUGGESTIONS 25

6.1 NEED FOR FURTHER STUDIES 25
6.2 SUGGESTIONS TO PRACTITIONERS; 25

BIBLIOGRAPHY 26

APPENDIX A – TEMPLATES FOR RESEARCH AND INTERVIEWS 29

RESEARCH IN PROCESS DOCUMENTS ... 29
Non Conformity Reports ... *29*
Development Process Agreements ... *29*
TEMPLATES FOR INTERVIEWS ... 30
Template for Semi-Structured Interview of Project manager ... *30*
Template for Semi-Structured Interview QA manager ... *31*
Template for Semi-Structured Interview of External Auditor ... *32*
APPENDIX B – EXTRACTS FROM INTERNAL AUDIT REPORTS ... 35

Acknowledgement

I would like to thank my Indian staff and my clients for all the experience I have got over the last 19 years in India. In particular I would like to thank those who I have interviewed and observed, without their input this paper would not have been possible to write. Any errors or misinterpretations are entirely my own. I would also like to thank my tutor Hans Tap and also Kari Rönkkö for valuable input and direction on literature, methodology, perspectives and finally I would like to thank my wife and children for letting me spend time on my studies during the last year.

Mariestad, June 2012

Mikael Gislén

1 INTRODUCTION

This section will paint a background for the paper, discuss software development methodology including background to why we need processes in the first place, and focus on the problems we will cover in this paper and finally reach the aims and objectives of the paper.

1.1 Background

Hundreds, if not thousands, of Development methodologies have been developed for various purposes and for various types of projects and organisations (Wikipedia, 2005). However methodologies are only one factor in developing good software solutions. Daniel Tenner (2011) says that software project management success is based on people, process and tools. While that is certainly true, at least in the case of software development, the author would suggest that these can be broken down to factors such as;

- Clarity of vision,
- Objectives and requirements,
- The knowledge, expertise and talent of the leader as well as the other participants of the development team, and
- The attitudes and beliefs of the participants
- The choice or limitations of the development tools used.

In today's global environment with software outsourcing being an emerging strategy of development and where it is rarely possible to develop any larger piece of software in close proximate to the client. This leads to additional challenges such as tacit knowledge, cultural challenges and geographical distance that have to be taken into consideration[1].

1.2 Software Development Methodologies

The early approaches to structured Software development methodologies where developed from a positivistic ontological viewpoint (Avison & Fitzgerald, 2006), which implied a belief that it is possible to define rigorous objective non-ambiguous requirements which customers and developers alike could understand in the same way. In addition there was also a belief that the produced software would be operated in a stable environment where such requirements collected before development would be equally valid once the project was delivered. Even though this perspective was common, there were early concerns about such a model. In fact already the paper which by many is considered the defining paper of the waterfall model raise serious concerns about this perspective. Royce says that the water fall[2] model was basically flawed and he listed serious concerns and instead suggested enhancements such as prototypes and iterations (Royce, 1970).

In the decade that followed, various iterative and cyclic models where defined and today many methods in the agile camp have more or less completely left behind the positivistic claims of early methodologies which can be seen in the so called Agile Manifesto (Beck, K. et al., 2001) and even if the authors of the same do not clearly articulate it they seem to rather build their models on social constructionism (Cockburn, 2000) and to manage a constantly changing reality they propose frequent short iteratons since they do not believe it is possible to define requirements which are objective or lasting. By constant iterations it is at least partially possible to constantly refine a product, and also to catch changing properties of the environment the product is supposed to exist in.

[1] The same author has in his earlier work covered cultural challenges in outsourcing.

[2] Royce never himself called it the water fall model

1.3 Purpose of a software design methodology

Apart from the obvious need for management control of the software development as any other activities of any professional work the following purposes of using software methodologies can be found in the literature[3]:

1. To protect the company and its customers from business risks (Boehm, 1991).
2. To act as an engine for learning, an engine which may over time as your expertise develop may become unnecessary or even in the way (Mathiassen, 1998).
3. To provide and mediate collaborative learning and knowledge integration between different actors such as domain experts, developers, Management etc. (i.e. developing a common language helping to inform, plan and position where we are in relationship to the project during the progress of the project) (Kensing & Munk-Madsen, 1993)
4. To push political views, i.e. to drive politics in a specific direction. Since a method often makes it possible or provide for Management of certain aspects while it puts aside or even suppresses others and gives and takes power to actors (Grudin, 2003).

However, in the view of the author of this paper, many software development methodologies take a generic context and at the best downplay human aspects when describing the normative development process. The assumption is that if you only follow these steps then you will deliver quality software which is up to the expectation of the customer. This is not only dissatisfactory but actually concerning since it does not reflect the real dynamics of software development.

The literature includes surprisingly few references to dysfunctional aspects of Software Development, while some other disciplines offer some input.

The objective of this paper is to try to deepen the understanding of how some of these (negative) factors affect the software development methodology. This would hopefully be useful for professionals who can use this to understand underlying context and social constraints when using methodologies but also contribute to the knowledge base.

Drawing from Psychology, Information Systems, Cultural Studies, Global Software Development Studies, CSCW, HCI and PD as well as Micro Economics, the author will try to build a better understanding of the complex reality of human aspects affecting software development. In a small way he hopes this will contribute to better software being developed in the future.

1.4 The darker side of quality systems and development processes

To use rigorous methodologies in order to achieve quality software seems to be a good idea, but there is little evidence that rigorous processes systems actually delivers quality or improves business performance (Dick, 2000). One possible explanation for this is that formal approaches neglect behavioural aspects such as perceived software quality (Hofman, 2011) other factors may include;

1.4.1 Cutting corners and deceptions

In the purposes above we saw that as software developers get more experience process activities which were necessary to learn the trade may over time get in the way for an effective development methodology. Combined with overzealous Quality Managers the experienced developer may develop cynical views on process and may throw out the baby with the bath water and let go of sound professional principles. Other developers may never have learned sound rigorous methods but developed their technical skills and can manage to an extent and as long as nothing happens it may work well. Finally pressure to deliver may make developers cut corners in order to please their

[3] The three last aspects are inspired by a paper by Rönkkö, et al. (2008)

development managers or customers. Other reasons may include simple laziness in testing the software, carelessness, etc.

1.4.2 Organisational Politics

Anecdotes about organisational politics are often told at the water cooler and when professionals meet at different occasions. One story which is often heard is quality department staff with an agenda based on internal quality system objectives, while developers instead focuses on pleasing the customer to the extent of cutting some corners which potentially may be risky for the company. The clash between these two categories may lead to corporate politics and it may be hard to avoid this conflict all together. In an ideal situation any staff would know how to balance the internal objectives with the customer needs and demands, but in reality it may not be that easy. When combining these relevant but sometimes conflicting agendas with personal weaknesses such as insecurity, need for recognition, perhaps need to satisfy metric based slightly misaligned financial compensation systems and other human factors it is easy to understand why organisational politics can negatively affect the organisations ability to effectively deliver quality software for its customers.

1.4.3 Agile Development

Even when agile software development is used, the author have personally observed similar dynamics as described above, while practitioners claim to adhere to agile methodology, but when discussing it was often revealed that they in reality paid lip service to the agile methodology and used more or less ad-hoc development and had no or limited understanding of what agile development really is. Surprisingly often people imply that agile methodology means no-process while in reality it means a very rigid no-nonsense process.

1.5 Aims and objectives

The objective of this paper is to identify and increase the understanding of intrinsic negative social aspects such as lip service, cheating and politics which are affecting the results of Software Development Methodologies and if possible suggest some means to mitigate them. In particular;

- To create a deeper understanding of why people cheat and pay lip service to methodologies
- To try to understand the political aspects of methodology and quality systems.

1.6 Limitations

This study focuses mainly no negative human and social aspects. Obviously there are plenty of positive human and social aspects such as professional pride, team spirit, (positive) group pressure, wanting to be accepted by the team, by management and by the customer, identifying with the profession etc. While to a large extent these aspects are left out in this study, they are by no means unimportant to create a full picture. However to somewhat limit the scope this study does not focus much on these aspects. Rightly used these can likely be used to mitigate any negative social aspects.

1.7 Outline of the paper

The rest of the paper is divided into a literature study a description of the methodology applied for this study, the study itself and a discussion followed by a conclusion. Appendixes are added listing material and semi-structured interview outlines

2 LITERATURE STUDY

This chapter will cover what others have written about the area covered by this paper and to give a starting point for the study of the dysfunctional aspects of software development.

2.1 Selection of literature

Most literature references used are based on systematic search on the Internet using ACM or Google scholar on central key words. Some based on advises from Kari Rönkkö and others based on general readings of the author. In general literature of normative nature or which lacks a critical stand has been avoided. Some literature references relate to different context but in the author's view this may have application in the context of this paper. Since the ethnographic methodology used for the method to some extent should stand on its own, the literature study is not necessary for the study but the intention is to create a starting point. Hence the purpose of this chapter is to create an introduction to the problem.

2.2 Formal Quality Systems

While the main interest of this paper is on Software development methodology, a corporate quality system is often used to support and measure the success and control of how a company uses certain software development methodologies[4]. While a Quality System is not the same as a certain Software Development methodology there is a connection, and the underlying reasons while companies achieve Quality System Accreditation may affect how well their methodologies works, and how well it is possible to measure and understand the underlying quality of the methodologies.

Companies achieve Quality System certification for different reasons. According to Nair & Prajogo (2009) suggests institutionalist (i.e. wanting to be like others since you are forced to, trying to match others, or believing in the impact since others claimed it) and functionalist reasons (the belief that the quality processes improve the efficiency, and quality of internal processes and that certification ensures that this is achieved). The paper find empirical evidence for that high performing companies were mainly looking for ISO certification based on functionalist reasons. Further they claim that the institutionalist approach leads to superficial implementation of the ISO process with discrepancies between documented processes and actual practices (Nair & Prajogo, 2009). The latest ISO 9000 standard (ISO 9001:2008) has a stronger focus on effectiveness of processes, and unlike the earlier version compliance is not enough, for compliance a company must define what it means to be effective and how to prove that its methods and processes are effective (Palmes, 2011).

ISO 9000 was not developed for Software Development, and unlike the Capability Maturity Model (CMM), ISO 9000 therefore does not prescribe any particular software development methodology or paradigm, even though it is easy (but not necessary) to read a more traditional waterfall paradigm into the standard's general engineering / manufacturing terminology. To more directly support Software Development, British Standard Institute developed the TickIT variety of the standard which covers some key practices and can be seen as an instantiation of ISO 9000 for Software Development. External Auditors for TickIT certified companies would for example have a solid IT background. TickIT is currently being developed into TickIT+ which includes maturity levels similar to CMM (IT Governance Ltd, 2012).

There is a built-in problem with assessing quality of Software Development since Software quality can be viewed from organisational, management and engineering-viewpoints. Since these viewpoints are based on different ontological and epistemological bases and Software can be seen as both an engineering product and an information system. While quantitative measures may be suitable from an

[4] There is no one-to-one relationship between the Quality System and a specific software development methodology; in fact the same company can use various software development methodologies for different projects within the framework of one quality system.

engineering viewpoint, customer satisfaction requires qualitative and interpretive perspectives (Hellens, 1997). Since the ISO perspective is mainly built on a quantitative engineering perspective and may therefore be insufficient for ensuring certain aspects of quality.

2.3 The flip side of formal quality systems

From the literature it is evident that use of formal Quality Systems can lead to a lot of lip-service and even outright cheating when preparing documents and filling in checklists afterwards to please Management, customers or the external auditor (Boiral, 2003).

Pfleeger & Hatton (1997) found no strong evidence that formal design techniques in themselves delivered higher quality source code but when they were combined with other techniques such as unit testing they delivered better quality.

While companies in their marketing say that they abide by the processes and methodologies this is not always the case (Boiral, 2003). Instead, there is evidence that formal methodologies are not as commonly used as proponent would wish (Fitzgerald, 1998). In a study of Scottish Software companies it was found that there is commonly a gap between rhetoric and practise implying that while managers where claiming adherence to the Quality System, developers were working in a much more arbitrary and ineffective manner, the study does not blame the practitioners for this but questions the underlying models which does not take economic and social forces into consideration (Beirne, et al., 1997). Practitioners may act in deceptive ways and pay lip-service to processes for different reasons, in case the methodologies are truly too rigid and an experienced developer fills in forms afterwards this may be based on pure pragmatism or need to survive, since the rigorous process is in the way or since the methodology is badly fitted for the needs of the professionals (Fitzgerald, 1997). Senior professional may, based on their experience, consider certain process activities unnecessary. This may be true, but perhaps experienced professionals do let go of process activities too easily as is evidence from other disciplines than software engineering[5] that use of checklists and methods deliver better results than without them even for experienced professionals.

2.4 Motives and Incentives

To get people to follow processes, Management may use various incentives which are intended to work for or against certain expected behaviour. These incentives are of three flavours; economic, social (such as insecurity, need for recognition, peer pressure etc.) and moral (Levitt & Dubner, 2006). However when Management introduces one incentive another may be weaken as a result. In a case study of a Crèche in Israel a fine was introduced for parents who came late to pick up their children. But the fine instead lead to more parents coming late, since the fine mean that the moral incentive against coming late was removed when you could pay for it. When the fine was later withdrawn the earlier social incentive against coming late was removed since there were many more people coming late so the number of people coming late was not reduced once the fine was withdrawn (Gneezy & Rustichini, 2000). Developers may as a result pay lip-service to the quality system and fill up forms and checklists since there are incentives against not doing this.

2.5 What is the potential negative effect of using metrics

Most Quality standards today imply using software metrics (Fenton & Neil, 1999). There is evidence that metrics in themselves often are selected in a way which supports the current way of developing software rather than in a way to support change (Damborg Frederiksen, & Rose, 2003). It is not hard to understand that in particular when metrics are linked to financial compensation employees may find ways to cut corners while satisfying the simple aspect which is measured. Bolinger & McGovan (2009) gives an example on how a developer can subvert the metric collection process. Since the

[5] In the medical context there are a number of cases showing that simple checklists have helped reduce complications and even mortality. One very convincing example is discussed in a couple of papers on how physicians in an emergency department diagnose patients with chest-pain (Reilly, et al., 2002), (Reilly, et al., 2002)

quality system demands low variation in schedule time and product quality but less about development efficiency, the developer can give a high estimate and then just filling the slots and delivering exactly what committed and on time and on budget.

A given incentive may lead to what economists call externalities, in which an initiative with an expected outcome also lead to an unexpected side effect. Focusing on internal incentives (such as always strictly following the quality system, satisfying the ISO auditor or trying to improve on internal metrics which would potentially give the developer a higher variable pay) may lead to externalities such as not delivering actual customer quality (Spitzer, 2005).

However metrics and incentives may also work in combination with other factors if a critical mass of developers rather than following a development process to satisfy quality departments expectations actually *believe* that the processes used delivers real customer quality, it is likely that social and moral incentives would lead to better quality. Looking at Quality metrics outside in and developing systems which responds quickly to challenges, measures which truly relates to what the customer and company values, and measures being in the hand of the people doing the work can avoid the occurence of the problems mentioned.

2.5.1 What about the agile community?

Since the agile community in many ways shy away from many of the aspects of process mentioned earlier in this paper, it is easy to believe that the concerns raised does not apply when agile methods are used. However in my the literature review it was found that very little empirical research has been carried out on process adherence in connection with agile processes and whatever found was more focused on adherence to *schedule* rather than adherence to *process* (Cohn, 2006). One paper (Zazworka, et al., 2010) suggests a framework for measuring process adherence for agile methods and uses it for one agile method (XP) and even though its study is only on beginners it does finds deviations in agile team’s adherence to process. This article uses a framework which is not intrusive since the measures are based on data from the source code version control system.

2.6 Is Management influencing sampling for audit?

Since Management have an influence in the selection of projects for external auditing, some companies with less commitment to internalising quality processes (institutionalists) would, if possible, influence the sampling of projcects for audit and in particular avoiding to put forward projects with known serious flaws. Putting forward such projects could lead to a lot of extra clean-up work or even in extreme cases to loosing the accreditation. An alternative concerning strategy could be to cook the books by ensuring that certain projects are in good shape[6]. Since both developers and companies can have various incentives (in particular economic and social incentives) in not following good practices this may lead to an unnecessary gap between process expectation and real practises.
Assuming that this would be the case the author searched the literature for any reference to this problem, but could not find a single reference. Normally the literature section of a paper of this sort would not include what you did not find in the literature. But since it was the authors belief that Management’s influence on sampling of data for audit would be a major potential concern this is still included. In case Management for reasons of adherence tries to influence the sampling to avoid non-conformities it would make it impossible for an external auditor of the quality systems to get a fair picture of the actual state of affairs of the company’s underlying processes.

2.7 Politics and Culture

In the methodology and quality literature the process of producing quality software is often assumed to be a straight forward approach. If you just do what you need to do then it will work. However somewhere in the process human aspects comes in the way. Organisational Politics in different ways

[6] If such clean-up activities would happen across the board from time to time to find discrepancies it may be an acceptable management practice to counter situations when staff have not had enough time to get documentation in order. The concern here is not this type of practice but selective correction of discrepancies in order to pass an audit.

are based on human control, positions, and power which affect who is allowed to ask and what questions are asked this affects what people hear. In addition, organisational Politics works against people taking responsibility for their own work. The difference between technology and politics is that in case of technology you can argue based on facts only, while in politics you cannot (Stribrny & Mackin, 2006).

Power and politics have been researched in the IS literature by a number of researchers, among them; Markus (1983), Orlikowski (1991), Walsham & Waema (1994), Nicholson & Sahay (2001) and (Grudin, 2003). However outside of the IS Context, Merton (1996) defines that there are Manifest functions as well as Latent functions where the Manifest functions are the objective intended and recognised consequences of individual and collective initiatives and actions while Latent functions where those not intended nor recognised. Especially in the latent side there are often dysfunctional aspects of actions. I.e. an individual can gain from something which is functional for him or her, while it is dysfunctional for others or for the organisation as a whole. He introduced this to distinguish between the motivation and the consequences of certain behaviour. He says himself that the terms are adopted from Freud and Bacon. Merton states that the underlying reason for an individual's actions are rational from his or her perspective but others may perceive the same action as irrational since they don't know the underlying reason. In this paper this framework can be used to explain and understand the individual's actions.

The ISO 9001 process and the organisation defined to enforce and control it, have certain similarities to Government bureaucracies. Merton discusses bureaucracy at length. He gives in his chapter on "Bureaucratic structure and personality" (Merton, 1957) a long description of how a formal, rationally organized bureaucratic social structures works including extensive description of roles, rules and hierarchies, life time tenure, security, promotion based on years of service etc. The main manifest function[7] of the bureaucracy is the technical efficiency. However while the bureaucracy itself is generally only seeing the positive side of things, it is generally obvious for the society as a whole that there are also number of latent (dysfunctional) aspects such as that the system is not open for change - procedures and training are not applicable to new situations, the officers develop blind spots and their own original behaviour based on their work habits, such as literal adherence to the letter of a rule. Or with other words - The means becomes the end. Merton shows how this develops;

> *"1. An effective bureaucracy demands reliability of response and strict devotion to regulations.*
>
> *2. Such devotion to the rules leads to their transformation into absolutes; they are no longer conceived as relative to a set of purposes.*
>
> *3. This interferes with ready adaptation under special conditions not clearly envisaged by those who drew up the general rules.*
>
> *4. Thus, the very elements which conduce toward efficiency in general produce inefficiency in specific instances. Full realization of the inadequacy is seldom attained by members of the group who have not divorced themselves from the meanings which the rules have for them. These rules in time become symbolic in cast, rather than strictly utilitarian."* (Merton, 1957).

While software development methodologies to a large extent try to encourage individual responsibility and accountability, organisational politics is antagonistic against the same (Block, 1987). While Quality Managers cannot prove that quality processes will lead to improved bottom line, Production staff cannot prove that cutting corners will improve the bottom line either. There is more than one true perspective here and there are many "truths" (Stribrny & Mackin, 2006). While organisational politics will always be there (Stribrny & Mackin, 2006), there is positive as well as a negative organisational politics (Block, 1987) or as Stribrny & Mackin says it "Amid all the pressures that slowly erode the

[7] Merton does not use the terminology in this context so the application is the author's and not Merton's

goal of creating quality software, choose a positive, principled way of being political" (Stribrny & Mackin, 2006). Stribrny & Mackin also suggests the following principles:

1. Accept that organisational politics is always going to be there but also that there are many "truths"
2. Build relationships and partnerships, work together with others to build the company's as well as your own goals
3. Identify what kind of relationship patterns successful staff use to get result, who is their "patron"
4. Don't try to solve problems immediately, instead try to look at the situation from the other person's perspective, how can you act proactively to prevent situations to even occur

Closely related to politics are cultural aspects of how to get expected result. There may even be links between cultural dimensions (Hofstede, 1980) such as Power-distance, Individualism and Uncertainty avoidance and different mechanisms for controlling formal outcome and formal behaviour one side and informal control modes such as individual or clan on the other (Narayanaswamy & Henry, 2005). These differences can create challenges when outsourcing a project from one country to another[8]. In addition when comparing this with underlying values stated in the agile manifesto it can be argued that the agile manifesto (Beck, K. et al., 2001) assumes an individualistic, low power distance culture with low uncertainty avoidance. It is a relevant question to ask how well agile methodologies work in a collectivistic culture with high power distance and high uncertainty avoidance[9].

[8] Based on my own professional experience, during more than 25 years and in different countries, in my view if an individual would try to follow the process to avoid being punished for failure in collectivistic cultures and in larger organisations, while they would ask themselves if a deviation is the right thing more often in individualistic cultures and in smaller companies.

[9] Many western countries (In particular North West Europe, Australia, USA and Canada) would, according to Hofstede (1980), have a tendency towards the former while many developing countries would tend to the latter.

3 RESEARCH METHODOLOGY

This chapter describes the methodology for the Empirical Research and discuss the internal and external validity of any findings.

3.1 Ethnographic Research

An ethnographic research was selected and applied in the author's own company. The company was founded in 1994 and the company has gone through a maturity process starting from a poorly defined water fall development methodology in the mid 1990's to an ISO 9000 based quality system combining traditional methodologies with agile methodologies based on the needs of our customers.

Ethnography has its root in Anthropology and was originally used initially to study small ethnic remove groups of people and their interaction and their social institutions. The early pioneers where Malinowski and Radcliff-Brown who lived under long periods (years) together with the group of people, often learned their language and went very far in their identification with the people they studied. The methodology has since developed beyond its "tribal" origin and has also been used in mainline sociology for example within business studies. An anthropologist observes interviews and studies various artefacts. When the anthologists interviews people they are often referred to as informants (Hylland Eriksen, 1995/2001). The word ethnography normally refers to both the research method and to the writing. This paper may not constitute a proper ethnography but it will try to adopt the method[10].

Ethnographic research has been used in the software engineering arena before in particular within three traditions; Computer Supported Cooperative Work (CSCW), Human Computer Interaction (HCI) and Participative Design (PD) (Rönkkö, 2010). Rönkkö (2010) further stresses on the difference in perspective between the software engineer who to a large extent studies reality from a reductionist natural science perspective (breaks down customers' expectations into formal requirements and solves discrete problems in generic ways which ideally can be reused) and the Ethnographic perspective which is holistic, rich, qualitative and avoids simplistic generalisations.

Unlike many other types of qualitative research, Ethnographic research is a rich and fairly flexible method which allows for taking many methods and perspectives into consideration; this may include various artefacts, personal observations, People's own descriptions of how they experience their reality and it also gives opportunities for triangulation which means using the different perspectives or observation methods to ensure stronger validity when the same situation is supported by the different perspectives.

Artefacts will be used such as process documents together with observations of situations and personal experienced or second hand described anecdotes told by staff (including production and QA staff) from interviews[11]. In these interviews employees will also be allowed to explain some of the incidents found in the process documents. To try to get an external perspective an interview was also made with the external ISO auditor in order to get his perspective on the company's development process since he could compare with a lot of other companies he has audited. In addition the literature was used together with my empirical research to find explanations to some of the behavioural aspects.

Certain aspects of the study focused on sensitive aspects which made it difficult for informants to answer; personal observations by the author being from another culture than the informants has been used which in some cases may make these observations weaker.

[10] My tutor has recommended avoiding me using "I" and "me" (even though this is commonly done in ethnographic studies).

[11] Among the methods I intend to use includes my own observations, review ISO 9000 artefacts such as Internal and External Audit reports, process and project artefacts, semi-structured interviews, letting informants review my findings and give feedback, etc.

Apart from discussing the dysfunctions themselves, the interviews also tried to capture the informants view on how to mitigate the challenges and how they would suggest resolving the hurdles.

The literature was used as a starting point and a framework for understanding the observations, and also to identify areas where there were gaps in the literature. The empirical observations were used together with the literature in an iterative manner. But the empirical research was also used to deepen the understanding of the findings from the literature. This approach may be similar to grounded theory but rather than just create theory from data, the intention was to deepen or expand the theory based on data (Alvesson & Sköldberg, 2008).

3.2 Validity

Using a basis of documented non-conformities ensures a fairly high internal validity for conclusions drawn from these artefacts. However any interpretation of the data is subjective and there is also a risk of the researcher being home-blind when studying one's own company and interviewing staff that are also potentially home-blind. In social anthropology the term "going native" is sometimes used, which can imply that the researcher identify so closely with the people he or she studies that there is a risk that he or she gets too biased (Bryman & Bell, 2007). Another concern is that many of the aspects studied can be sensitive. It is hard to admit to the existence of lip service or to be part of playing organisational politics. In fact, since the political processes discussed may neither be known nor malicious but due to factors such as lack of confidence the involved people may be even unaware of the existence of them. Merton (1957) states that people are acting from what they perceive as rational positions, while the surrounding may not perceive it in the same way.

Studying only one company may not ensure any external validity. However there are reasons to believe that a profession such as Software Engineering in a specific cultural environment have some similarities and hence it is my hope that it would at least ad value for other practitioners to read the description. However the uniqueness of Gislen Software is that it is (mainly) Swedish owned and owner operated with mostly Indian staff. The cross cultural dynamics would possibly be fairly unique and may differ from other Indian or Swedish companies. For example the dynamics in differences such as Swedish very low hierarchy and Indian higher may give reasons to be careful with drawing to far reaching generalised conclusions. In addition to this ethnographic researchers in general are careful with general conclusions (Rönkkö, 2010).

Given the choice of methodology, the access to empirical material and the limitations based on personal observations in a limited context the external validity may be weak. The interview with the external auditor may give some perspective since his confirmation of what he agrees are typical would carry some weight.

4 THE STUDY

This chapter will describe the findings from process documents, interviews with different stake holders and other observations which jointly build a rich picture of the software development methodologies of the company.

4.1 Process Documents

As a base material all Production Department Non-Conformity reports (NCR) which are produced during internal audit based on the company’s ISO 9001:2008 certification according to BSI TickIT were used (Appendix B – Extracts from Internal Audit Reports) includes a table listing all such non-conformities (NCs) which have been reviewed).

During the walkthrough of all non-conformities, collected from the entire period the company have been ISO 9000 certified, it was found that while a lot of Non Conformities address mistakes or shortcoming in the documentation processes there was very little evidence of actual deviations in the underlying professional practices. This may in my opinion be due to any of the following factors:

- Production staff covering up underlying issues
- Inexperience by auditors or lack of understanding of the underlying processes
- Overly rigorous focus on formal documentation practices by the internal auditors/Management Representative[12].

The author selected a few of the NCs to use during the interviews with production and QA staff to find out to what extent either of these reasons could be relevant explanations to the lack of support for failure to conduct the professional underlying practice. At the outset, even while it was the belief of the author that the former probably happens, the author would assume that the reasons are more to be found in the latter two factors rather than the former.

Other Non-Conformities gives at hand lack of knowledge or understanding from the auditor’s side of what constitutes real business risks or how the business model of the company works[13]. In some cases risks are noted as if the lack of formal evidence means that professional practices have not been followed. Lack of evidence does not actually mean that a process has not been followed. Mistakes in documentation can certainly have a huge impact. In many cases there were evidence but not in the format expected. The lack of documented evidence is more likely to make people think that something have not been done which can lead to rework. I.e. it may be a false negative. Note that this does in no way mean that the NCs are not valid, only that it would have helped to dig in a bit deeper to verify whether the practice had been followed or not (by for example checking source code) and then still give a NC but the business risks of the NC would then be much different from the cases referred to.

It may be a valid suggestion that the auditors should spend more time trying finding false positives. I.e. evaluate cases where there is documentation but where looking at the underlying artefacts (source code, database etc.) are not done in the way stated. In fact not a single NC was found which indicated auditors focus on the underlying process, while plenty of NCs focused on the documentation process. This raises concern about what has not been found. If production staffs indeed at times do pay lip service to some practices and producing documents indicating that the practices have been followed there is a need of another type of auditing.

[12] ISO 9000 defines one person being the main responsible for the operation of the ISO 9000 system. This person is called “Management Responsible” (often abbreviated as MR)

[13] For example the internal auditors have assumed that the company could make a loss, when in reality the reason for why the project had an overrun (measured in hours) since the project leader decided to use non-booked staff on a project to keep them occupied and give some training leading to a technical overrun. While in reality this did in no way affect the cost of the company.

One NC stated that no actions should be initiated before process documents where approved. The issue here is not whether the NC is a valid or not but that the mentioned Clause 7.3.4 in the Quality Manual, only covers *how* documents should be reviewed. The internal auditor had either not *understood* the clause, or may exercise political influence. This case will be discussed further based on the interviews with the different stake holders.

Table 1 - NCR Document GS-INR-NCR-085-2.0, Non Conformity: 13/PRD-P/03

Type	Data
Document Id:	GS-INR-NCR-085-2.0
Non Conformity:	13/PRD-P/03
Type:	Observation
Reference:	Clause 7.3.4 of Quality Manual (GS-QMS-MAN-001-7.0)
Expected Finding:	All the project and process related documents should be approved before initiating any actions based on such documents.
Actual Finding:	1. In [Project name] project, the estimate is not approved but the project has been released to the client. [Document names] 2. For [Project Name] project, the test plan is not approved but testing has been conducted based on this test plan. [Document names]
Business Risk:	It may not be possible to trace documents based on which the actions were initiated which could lead to internal and external dissatisfaction and could lead to quality issues

The NCs were categorised into the following categories;

Table 2 - List of possible underlying potential dysfunctions

Category	Description	NCs where this is a potential dysfunction
Auditor Competence	The wording indicates that auditor may lack competence	10
Politics	Potentially a case of internal politics from the auditors side	4
Hiding	Potentially production staff may have been hiding something or have made an after construction to satisfy the finding. In all cases this is based on the reply. In case the reply is correct my concern is that this has not been settled since in such case there is no NC or it is less serious than noted. One other observation to mention is that there were no NCs actually finding any such occurrences where it is clearly stated that the auditor found any incident of actual hiding. The categorisation of “hiding” is therefore based on very weak arguments.	6

4.2 Interviews with staff

The production manager as well as the Quality manager was interviewed using semi-structured interviews (based on the templates attached in the appendixes).

4.2.1 Definition of Software Quality

Both of the informants stated that Software Quality was defined in terms of how well the software met the customers need,

4.2.2 Purpose of methodologies

The production manager stressed the purpose of software development methodology as to make the whole operation more efficient and reliability and predictability on a company level, if a development process is used it can be expected that the fundamentals of execution will remain the same independent of the individuals working on the project. There is a base system which will cover a lot of things which are required.

The quality manager stated that the purpose of Software Development Methodology as being a framework to support converting requirements into code. This framework should not control the details of how things are done, but only support the developers to know what need to be done. To do this there is a need of documents and checklists etc. Sometimes this is also need due to business reasons. He stressed a situational perspective where methods need to be selected and defined for every project like how to proceed, and how to verify.

4.2.3 Familiarity with different methodologies

Both of them were familiar with both traditional and agile methodologies.

4.2.4 Advantages and disadvantages

The production manager considers the advantages of software development methodologies mainly to not having to worry about the basic stuff, easy to induct new people into projects when there are written down rules on how things should be done. Especially for larger teams and projects it is probably also a more efficient approach. The Quality Manager stated that methodologies gives structure to how to manage and track, and how to verify what has been done. It also makes sure the team is on path, are that we move in the right direction.

The production manager stressed that the disadvantages are mainly when using too heavy methodologies for smaller projects. This adds unnecessary overheads while keeping things simpler often ensures faster and less expensive delivery. In addition customers often demand keeping process overheads to a minimum both to meet budget and due to short deadlines. Even though we tailor processes at the beginning, in reality external factors [such as customer requests] often forces redesign of process on the spot, which may not fit into the expected methodology. The Quality Manager instead stressed difference in interpretations of people dealing with it. His view was that verification has to be done in an orderly manner but others may work less orderly but catch up from time to time, he said that there are also challenges since the process itself does not ensure delivery of the product, it has to work together with the skillset of the people and the tools.

4.2.5 Avoidance or missing to do things

Both the Production Manager and the Quality manager said that avoidance was not the main issue but missing or not having time to do certain things happened at times, it also happened that certain things like code reviews and some internal tests were done after delivery (in particular when there are later deliveries). Sometimes release notes were sent after a delivery since it demands a lot of time to write and the actual delivery is part of the contract. Another reason the production manager mentioned was that test plans were sometimes too extensive and the developers did therefore not have time enough to conduct repetitive internal tests.

4.2.6 Regarding why we have not found some things in audits

The Quality manager stressed the need to create more trust with different people, and to make sure that whatever is discussed is not misused. This would encourage people to disclose information. Focus should be on the system and not on the person. It may matter how we ask question. We may need to ask more open questions.

4.2.7 Organisational Politics

On the question on internal politics the production manager discussed the perception of what constitutes a business risk. She felt that these are sometimes invented since the auditor might not have the maturity to assess if there is a genuine business risk. Auditors often find minor human errors which should be done but where it is clear to the production manager that there is no real risk. But the auditors often miss the bigger picture in how to frame NCs. She did not think in general that auditors had any agenda, but the perspective is sometimes not right. For example she would rather get a few NCs than letting down a customer, while auditors sometimes think the other way. The business risk attached to not delivering an incomplete product with minor deviations may be much higher than the smaller business risks related to minor deviations. If the bigger picture is not taken into consideration the message given by the auditing process is wrong since it can create overly cautious attitude from production staff. In addition if we cover all possible risks the estimate to the customer would be so high that we would never get any projects. The attitude of the developers is that we have too many documents and too much work to satisfy the Quality systems therefore they add too much hours to the estimates, and she frequently have to cut down estimates in order to give the customers estimates which they will accept. If the QMS demands us to do so much that no customers will accept the projects there is no advantage with the system. Some risks must be taken and some corners must be cut in order to satisfy the customers need. In case we cover all internal requirements and all potential risks, there will not be any profit and she says that the auditors and Quality department people lacks this real-life perspective.

The Quality Manager on the other hand stressed that production often stress the delivery of a quality product as sufficient and that production manager sometimes protects her team. He also said that it happens that developers downplay deviations and that there have been cases where 50 actual defects have been quoted as a couple in order to downplay serious deviations.

4.2.8 Most common dysfunctions

Production Manager said that the development methodology is normally only tracked by the project leader or Scrum Masters, while the others often lack knowledge of the process. She also said that there are individual developers who pay lip service to the processes. Internal politics happens; in particular the QA department stresses an internal perspective while she had to mainly take the customer perspective in consideration.

The QA manager said that production staff lacked knowledge of the methodology but that auditors had improved their competence. He stressed that lip service in particular related to corrective actions. He did not consider internal politics being an issue, but his understanding of internal politics assumed malicious intent which may not be the case.

4.2.9 Review of NCs

We jointly reviewed a number of NCs and the following aspects stood out or were relevant for a number of NCs. The following is a list of statements phrased by either of the interviewees focused on their view of the different NCs we reviewed:

- Business risks often poorly framed. (13/PRD-P/04, 15/PRD-P/03, 01/PRD-P&T /05, 01/PRD-P&T /06, 01/PRD-P&T /10)
- Auditor competence need to be improved (13/PRD-P/03, 01/PRD-P&T /09)
- Auditor should dig deeper and focus more on underlying day to day practice and not stay on the surface (3/PRD-P/04, 15/PRD-P/02, 01/PRD-P&T /10)
- Root causes sometimes defined poorly and the corrective and preventive actions have also not been framed well in some cases.
- Too much focus is on finding faults and almost no focus on identifying what actually works and what is to be encouraged.
- Since NCs often create emotional response production staff often reacts in an emotional way rather than trying to improve the processes.
- Auditor should focus more on the system rather than on finding individual mistakes. If there is no evidence of any system fault, the issue or business risk may not be valid.
- When NCs are given to production and production responds with root causes which indicate that the NC is actually not valid, it can be considered a non-issue and a weakness in the

auditing system. There should be some feedback to auditors making mistake to ensure learning also. As of now the control of those who control is weak.

- When the process has obviously been followed and some minor documentation issue has occurred there should be no NC.

The production manager raised a question on how smaller companies manages ISO 9000 for getting higher quality while remaining flexible and agile and using the strength of being a small organisation. She further asked if our way of doing ISO is not effective or efficient enough and if there are ways of doing it with less documentation/bureaucracy. In her view the procedures, the internal and the external audits indicate that most things are not optional and the auditors seems to want things to be done in too time consuming ways without taking the specific customer or project need into consideration (i.e. one process fits all). She also suggested a more holistic perspective during audits, since now audit of development and of testing was done at different occasion even though the same project may be audited in both cases) merging audits to have only one audit for production and testing (in the past we have had one audit for projects, one for teams and one for testing). She also raised concerns that it is very difficult to really close a lot of NCs since the mistakes are of human nature and will often happen again, and the mistakes are often not very critical. In her view the systems to remove such irregularities are often very costly compared to the risks they give. I.e. the cost of quality is too high. Another aspect was that [internal] auditors rarely if ever noted noteworthy efforts of the teams, the main focus is on fault findings. In many cases such faults are already known and the impact of spending time digging into them is low. If ISO is mainly a fault finding exercise the value for the company is very low.

She confirmed that senior professionals often knows when to deviate from process and avoiding unnecessary process steps, while she did not recommend everyone [meaning juniors or not so mature people] from doing the same. She stated that production staff would rather do what is required at the spot to deliver to the customer and even risk NCs later in case required. This confirmed some of the references in the literature

She was perhaps too careful not to let down other people, but also stated that in hindsight she did not necessary agree with her own comments given to the NCs, since she said that they were somewhat reactive

She was concerned about the emotional aspects of the ISO audits. She also perceived that audits were mainly focused on finding faults and that it would help if the audits were more neutral and if auditors were also looking to find positive aspects. The too negative focus was sometimes demotivating.

4.3 Active Participation in an Internal Audit

The author participated in one internal Audit where a major weakness was that the project leader for the project being audited was on parental leave and the auditee was the production manager. Due to the absence of the project leader who had managed the project the focus was more or less purely on administrative aspects, the only aspects of interest was related to resource allocation, and certain weaknesses in the competence or experience of the team members

4.4 Conducting an Audit

The author conducted an audit himself of a project where both the project leader and one of the developer was available. This being a maintenance project meant that the process was a bit different. The author went through the DPA (Development Process Agreement) and the focus was mainly on the development practices. While the process itself was managed quite well, there were certain aspects which could be improved.

When deployments where made, the developer made entries in a log, but if deployment failed he just rolled back and did not note down the failed attempts with a reason.

The Software which is maintained is not originally developed by the company and as the developer fixed code which was not developed by the company and not well written or not commented, he generally only commented the new inserted or changed code and only ensured that his own code was of good quality. While refactoring and improvements should be done with care to avoid injecting new defects it was agreed that he should when he assessed the issue, also consider when refactoring was required and he should always add comments to explain what the section did even if it was not related to his own code.

During the audit it was found that a few of the existing modules (not developed by the company) have poor existing unit tests / code coverage of the unit tests. The need to add more codes to these unit tests were discussed to ensure that they covered the underlying code better.

In particular it was found that no consideration was done when a defect was reported why the existing unit test did not cover this error. The author suggested that his first approach should be to ask how he could strengthen the unit test(s) to ensure that the bug was detected there and only then fix the bug. This practice has its origin in Extreme Programming (Wells, 1999-2009) but is considered good practice for all test driven development.

4.5 Review of class design and design document

The author reviewed a class design and also a high level design document for one project. The designer was not very experienced, so the case was mostly a learning case and would not be typical for the company. However since the attitude of the developer may be typical for some more junior developers it raises some concerns.

The author have earlier done this kind of design review a number of times, but since this design happened during the time this paper was written, this is included since some of the learning’s add value to this study.

The class design was not good but when giving feedback to the person who had done the design he gave feedback that since this was an iterative project it was not essential to have a good design at this stage and it was more important to deliver something which worked. In my view a good class design has significant impact on future work and in this case since a proper design pattern would lead to much less reuse of code and easier and less expensive maintenance. Even more knowledge of sound design is essential for developers and hence unless highly motivated a good design should not be put off for the future. However it was also clear that the developer’s knowledge of design patterns and good object oriented design was very limited.

In the review of the High level design for the same project, it was found that some of the sources used to motivate certain design were not properly researched since the external source was irrelevant or there were comments in the external source which when applied in the particular case nullified the arguments stressed by the developer.

The class design and the high level design document both indicated an understanding by the developer where proper design was something which the company demands and not truly necessary.

4.6 Interview with the External Auditor

Our external auditor has 10 years of experience as an ISO auditor but totally about 33 years of experience in the IT industry. His experience is very wide. His experience of different development methodologies is wide, he has worked with waterfall, iterative, incremental and even agile methodologies, some of them hands-on, and others he has only audited. He defined software quality as consistent delivery of what the customer wants. He stated that different methodologies deliver best in different situations. If requirements are well known up front and there are no drastic changes expected, than a traditional life cycle model or more traditional iterative model may be the best, but such a model would not be successful when customers frequently change features implemented in earlier iterations. But even if agile methodologies are promising, they are not suitable for every situation. Their strength is that they keep away unnecessary complexity while following a rigorous daily routine where all deviations are settled during the daily stand up meeting. But agile requires all-rounders and is good for more mature developers. It does require a lot of training. He mentioned that in his opinion continuous integration which is often used in agile development adds a lot of value.

4.6.1 Social aspects

In particular when using CMM there is a tendency to be very rigorous on the projects selected for auditing since that is known in advance. If the company is more careless on other projects the CMM auditors would still authorize the company. But in general the human nature does play in, and there are always people who are scarred of getting Non Conformities. In reality the number of NCs does not really matter. It is reducing business risks which matters. If the auditee trusts the auditor the result is better for all parties. In general he said that he can easily make out if people are hiding information from him. But finding faults is not his intent. He wants to help the company to reduce business risks. There is a risk that the audit process will become too bookish. It is important to train the internal auditors on how to deal with fear and other human aspects. It is also important to evaluate how serious are the deviations found. Internal audit should be based on common sense exercise with focus on Business risks and organisational learning, instead of just looking for records.

4.6.2 Lack of knowledge of process by production staff

Unless the leaders at different levels believe in the system there will not be quality or benefit by the system then the teams will follow a path of least resistance. Failing once in a while is not a crime, but when you get up you should be wiser. We must strike the individuals heart. It is not possible to get quality using the stick!

4.6.3 Lack of competence by internal auditors

This can be resolved by using more than one internal auditor where one is more mature or experienced than the other. One may be technical and the other may be from another department, it does not hurt if an auditor is not technical since the customer perspective is sometimes more important to consider

4.6.4 Lip service to process

People sometimes follow a process they don’t believe in since they fear their position. It is important to allow and encourage criticism. If someone says that they don’t believe in something this can be an opportunity to explain. If you supress criticism what you get is lip-service.

4.6.5 Organisational Politics

Conflicts such as those between quality and production can be healthy as long as they are not overdone. However in his experience there are a few concerns. QC time are not always added by the production team and then perceived as optional. In addition in case of overruns quality control time is often swallowed by the project team and this can give conflicts.

4.6.6 Other challenges

He raised the problem in education. Since educational institutions often only teach theory, without really knowing how the industry is working. The educational institutions themselves often pay lip-service to what they say, and students learn that. Unless teachers show by example and work with reality students would not appreciate theory. There are also cultural aspects. Development methodologies and quality systems have to be implemented slightly different in different cultural contexts.

Deviations from process are allowed. The question is who can give concessions? It is important to define who is authorised to deviate and if there is deviation what is the process to get it regularised afterwards. As long as a deviation is authorised there is no deviation. It may be good to get customer involved in authorisation of deviations. But other aspects such as frequency of deviations should also be considered. It is not good if there are frequent deviations. Relevant authority should generally be outside the project. Better if the person is not directly responsible.

No clear answer answers were given on internal politics. Even though the concept was explained in detail for example that there may be different agendas between quality department and production and that both may be right to some extent no proper answer was given.

5 DISCUSSION

When using an ethnographic methodology, the material should normally mostly speak for itself. However this section will anyway try to draw some conclusions based on the study and in light of the literature in order to try to answer the research question.

The study gives a mixed picture of the area of study. The author being a Software Professional recognises that looking for generic meaning comes very natural for his profession but using an ethnographic research methodology creates a potential conflict between his professional perspective as a practitioner and as a researcher and a mixed picture is therefore to be expected. While various factors were discussed it was very hard to get a clear simple understanding from the different sources. At the same time given the methodology the discussion should perhaps not go too far in trying to get such generic perspectives out of the material but the material may need to speak for itself.

Still trying to draw some general conclusions from the material; Looking at the Non Conformity Reports there were indications that they mainly focus on documentation issues and there was very less focus on whether the underlying actual practices were followed. This is also supported by the interviews with the Production Manager and the Quality Manager. In addition the interviews show that business risks are also poorly analysed and auditors must focus mainly on system and process error and not individual short comings. This may be explained by auditors' competence level which was indicated in the early review of NCs and supported by the interviews.

The external auditor in particular supported that the functionalist view of quality was essential for value of a quality system (Nair & Prajogo, 2009) while in particular the production manager questioned raised concerns about some aspects of process which did not add value which may support the same view since unnecessary process may be linked to taking the process from outside which is basically a non-functionalistic view.

The interviews as well as the authors own design review indicated a low level lip-service to process and methodologies which was also supported by the literature. There were suggestions that lip-service and cheating can be a result of people feeling insecure and since they may believe that they are not allowed to criticize the processes (something which may be explained by the hierarchical perspective in the Indian culture). The external auditor strongly suggested that criticism must be allowed to ensure that people don't pay lip-service to the process. Further he stressed the need for people to believe in the system, if that was not the case lip-service automatically follows. But this is not a one-sided process and has to be built up by trust from all parties.

It was hard to get a grip on to what degree organisational politics plays a role since informants were either silent on the issue or gave limited answers. However the author has in his own observations noticed signs of internal political conflicts, but it is hard to pin-point the exact mechanisms. Organisational politics is hidden and may even be denied since the informants understanding of internal politics assumed it to always be malicious. This view is in contrast with some references in the literature that there may be good organisational politics as well, as long as there is an alignment from the participants to company's objectives (Block, 1987) and (Stribrny & Mackin, 2006). Nevertheless, whether organisational politics is malicious or not, it is a factor which has to be dealt with. Recognising it as based on human weakness and not necessary malicious may be a start and if it can be brought to the surface perhaps more effective means to deal with the hurdles caused can be mitigated.

In the author's view the heart of the problem may be a confusion based on different world views which are not always spelt out. The ISO 9000 is a framework which to a large extent assumes a reductionist view on software development, based on a positivistic world view. The three informants on the other hand, stated that quality, in their view, was defined as giving customers consistently what they wanted, a view which may actually be in conflict with the reductionist view since the latter includes a subjective definition of quality. I.e. unless developers and managers take customers' *perception* of quality into consideration it may not be possible to create what the customer would consider quality software. The philosophy behind ISO 9000 is limited and unless it is extended to take subjective and human behavioural aspects into consideration it may not be possible to get optimal results.

While all three informants are dedicated to quality, their views differ at least to some extent. The production manager being a pure practitioner is the one with the strongest concerns about unnecessary process while the QA manager and the external auditor while both having strong IT careers behind they have stronger belief in the process itself to deliver quality. The external auditor also makes many more normative statements than the others (which is to be expected given his role and experience). The external auditor and the QA manager communicate stronger belief in that the quality system in itself is capable of eliminating human weaknesses or at least reduce them while the production manager (who in no way questions the system itself) still has more concerns about the system's ability to resolve all such problems.

Perhaps a better way to observe the challenges to quality software development is by viewing it in two layers. On the higher level are technical and procedural challenges which are addressed by good requirements, formal methodologies, metrics, incentives and tools, and on the lower level there are underlying human factors, which have to be addressed by other means (leadership, team-building, allowing criticism, building professional pride, training, etc.)[14], the quality system can at best be a filter ensuring that poor software is not released, but cannot address the problems at source. When trying to address the human factors only with the procedures there may be more lip-service, organisational politics and other dysfunctions disturbing the quality and reducing the profitability and customer satisfaction.

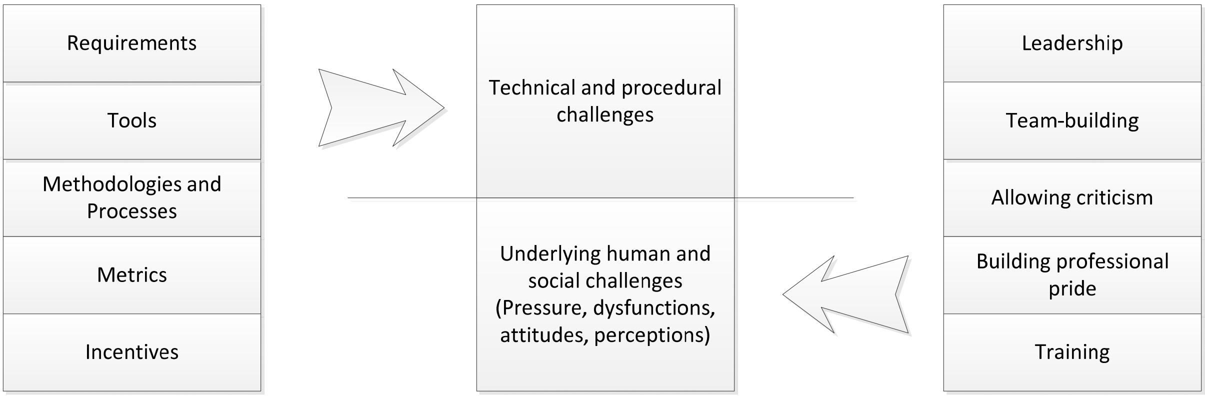
Requirements
Tools
Methodologies and Processes
Metrics
Incentives
Technical and procedural challenges
Underlying human and social challenges (Pressure, dysfunctions, attitudes, perceptions)
Leadership
Team-building
Allowing criticism
Building professional pride
Training

Figure 1 – A suggested layered view of challenges to quality within software development

As we saw in the introduction on page 6 under the heading "Purpose of a software design methodology" a software development methodology (as any system) is not a complete, once and for all, solution to all problems but should be seen more as a starting point and a roadmap. Unless the human aspects are addressed (through other means than the formal methodology and the quality system) it may not be possible to create true quality software.

[14] I am just suggesting different potential remedies which are mostly outside the scope of this paper and only included to give a differentiated view.

6 FURTHER STUDIES & SUGGESTIONS

This chapter suggests needs for further studies and also give some advice for practitioners.

6.1 Need for further studies

The following areas would need to be studied further;

- Internal politics
- The effect of Positive and Negative Incentives
- The effect of software metrics on quality performance
- Subjectivity in customers’ perception and expectation
- How positive social aspects (such as passion, professional pride etc.) can be used to mitigate the negative ones.

6.2 Suggestions to practitioners;

There is more to software methodology and quality systems than what meets the eye. Everything is not what it looks like. Software developers are professionals looking for simple answers. The human side of software development is messy and complex and the undercurrents can hardly be resolved with simple methods. In particular using simple carrots and sticks such as monetary incentives to get specific behaviour may backfire. In my view the developer’s personal passion for good quality and serving the customer are the most essential motivators to increase the quality of software development. If these social aspects are encouraged it may be a strong force to mitigate the negative aspects. It is important that audit of software process focus on the effectiveness of the underlying professional practices and avoids shallow focus on documentation only. Quality is built at source. Professionals must believe in their methodology, only then would they avoid lip-service and value the process. This is achieved by allowing criticism and empowering the developers to constantly suggest improvements. In addition it is important to teach the developers that deviations are allowed as long as they are approved and documented and if they are not too frequent (too frequent deviations indicate poor process).

#

BIBLIOGRAPHY

Alvesson, M. & Sköldberg, K., 2008. *Tolkning och Reflektion.* 2nd edition ed. Lund: Studentlitteratur.

Avison, D. & Fitzgerald, G., 2006. *Information Systems Development - Methodologies, techniques & tools.* 4th edition ed. Maidenhead: Mc Graw Hill Education.

Beck, K. et al., 2001. *The Agile Manifesto.* [Online]
Available at: http://agilemanifesto.org/
[Accessed 26 Feb 2012].

Beirne, M., Panteli, A. & Ramsay, H., 1997. Going soft on quality?:Process management in the Scottish software industry. *Software Quality Journal,* Volume 6, pp. 195-209.

Block, P., 1987. *The Empowered Manager: Positive Political Skills at Work.* 1 ed. San Francisco: Jossey-Bass.

Boehm, B., 1991. Software risk management: principles and practices. *Software, IEEE ,* 8(1), pp. 32-41.

Boiral, O., 2003. ISO 9000: Outside the Iron Cage. *Organization Science,* 14(6), pp. 720-737.

Bollinger, T. & McGowan, C., 2009. A Critical Look at Software Capability Evaluation - An update. *IEEE Software,* Issue September/October 2009, pp. 80-83.

Bryman, A. & Bell, E., 2007. *Business Research Methods.* New York: Oxford University Press.

Cockburn, A., 2000. Selecting a projects methodology. *IEEE Software,* pp. 64-71.

Cohn, M., 2006. *Agile Estimation and Planning.* Upper Saddle River: Pearson Education Inc..

Damborg Frederiksen, , H. & Rose, J., 2003. The social construction of the software operation - Reinforcing effects in metrics programs. *Scandinavian Journal of Information Systems,* Volume 15, pp. 23-37.

Dick, G. P., 2000. ISO 9000 certification benefits, reality or myth?. *The TQM Journal,* 12(6), pp. 365-371.

Fenton, N. E. & Neil, M., 1999. Software metrics: successes, failures and new directions. *The Journal of Systems and Software,* Volume 47, pp. 149-157.

Fitzgerald, B., 1997. The use of systems development methodologies in practice: a field study. *Information Systems Journal,* Volume 7, pp. 201-212.

Fitzgerald, B., 1998. An empirical investigation into the adoption of systems development methodologies. *Information & Management,* 34(6), p. 317–328.

Gneezy, U. & Rustichini, A., 2000. A Fine Is a Price. *Journal of Legal Studies,* 29(1), pp. 1-7.

Grudin, J., 2003. The West Wing: Fiction can Serve Politics. *Scandinavian Journal of Information Systems,* Volume 15, pp. 73-77.

Hellens, L. A. v., 1997. Information system quality versus software quality - A discussion from a managerial, an organisational and an engineering viewpoint. *Information and Software Technology,* Volume 39, pp. 801-808.

Hofman, R., 2011. Behavioral economics in software quality engineering. *Empir Software Eng,* Volume 16, pp. 278-293.

Hofstede, G., 1980. *Culture's consequences: International differences in work-related values.* Beverly Hills, CA: Sage.

Hylland Eriksen, T., 1995/2001. *Small Issues - Large Issues.* 2nd Edition ed. London: Pluto.

IT Governance Ltd, 2012. *What is TickITplus?.* [Online]
Available at: http://www.tickitplus.org
[Accessed 29 May 2012].

Kensing, F. & Munk-Madsen, A., 1993. PD: structure in the toolbox. *Communication of the ACM,* 36(6), pp. 78-85.

Levitt , S. D. & Dubner, S. J., 2006. *Freakonomics.* Revised and Extended Edition ed. London: Penguin Books.

Mathiassen, L., 1998. *Reflective Systems Development. Dr. Techn. Dissertation, Aalborg University.* [Online]
Available at: www.larsmathiassen.org/1.pdf
[Accessed 26 Feb 2012].

Merton, R. K., 1957. *Social Theory and Social Structure.* Third Edition (1968) ed. New York: The Free Press.

Merton, R. K., 1996. *On Social Structure and Science.* Chicago and London: The University of Chicago Press.

Nair, A. & Prajogo, D., 2009. Internalisation of ISO 9000 standards: the antecedent role of functionalist and institutionalist drivers and performance implications. *International Journal of Production Research,* 47(16), pp. 4545-4568.

Narayanaswamy, R. & Henry, R. M., 2005. *Effects of culture on control mechanisms in offshore outsourced IT projects..* s.l., Proceedings of the 2005 ACM SIGMIS CPR conference on Computer personnel research. .

Palmes, P., 2011. *Achieving Effectiveness and.* [Online]
Available at: https://www.usd.edu/business/manufacturing-and-technology-solutions/upload/Achieving-Effectiveness-and-Compliance-Breakout.pdf
[Accessed 29 May 2012].

Pfleeger, S. L. & Hatton, L., 1997. Investigating the Influence of formal methods. *Computing Practices,* Issue February, pp. 33-43.

Reilly, B. M., Evans, A. T., Schaider, J. J. & Wang, Y., 2002. Impact of a Clinical Decision Rule on Hospital Triage of Patients with Suspected Acute Cardiac Ischemia in the Emergency Department. *Journal of American Medical Association,* Volume 288, pp. 342-350.

Reilly, B. M., Evans, A. T., Schaider, J. J. & Wang, Y., 2002. Triage of Patients with Chest Pain in the Emergency Department: A Comparative Study of Physicians Decisions. *American Journal of Medicine,* Volume 112, pp. 95-103.

Rönkkö, K., 2010. Ethnography. In: P. Laplante, ed. *Encyclopedia of Software Engineering.* New York: Taylor and Francis group, pp. 278-286.

Rönkkö, K., Hellman, M. & Dittrich, Y., 2008. *PD Method and Socio-Political Context of the Development Organization.* Indiana USA, Conference on Participatory Design: Experiences and Challenges.

Royce, W. W., 1970. *Managing The Development of Large Software Systems.* Los Angeles, IEEE, pp. 328-338.

Spitzer, R., 2005. *The Journal for Quality & Participation,* Issue Winter, pp. 28-31.

Stribrny, S. & Mackin, F. B., 2006. When Politics Overshadow Software Quality. *IEEE Software,* Volume 06, pp. 72-73.

Tenner, D., 2011. *People, processes and tools.* [Online]
Available at: http://swombat.com/2011/10/24/people-processes-tools
[Accessed 26 Feb 2012].

Wells, D., 1999-2009. *Extreme Programming:A gentle introduction.* [Online]
Available at: http://www.extremeprogramming.org/
[Accessed 29 May 2012].

Wikipedia, 2005. *List of software development philosophies.* [Online]
Available at: http://en.wikipedia.org/wiki/List_of_software_development_philosophies
[Accessed 26 Feb 2012].

Zazworka, N. et al., 2010. *Are Developers Complying with the Process: An XP Study.* Bolzano-Bozen, ESEM '10 Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement.

APPENDIX A – TEMPLATES FOR RESEARCH AND INTERVIEWS

Research in Process documents

Non Conformity Reports

These reports are prepared by the internal auditors together with the QA Manager

1. Are there any signs of production staff avoiding process steps or paing lip service to process or methodology? (Compare with DPA's (Development Process Agreements). Are there any patterns?
2. Compare the clause reference with the Quality Manual, is the ISO system supporting or complicating the software development methodology?
3. Are there any signs of internal politics?
4. Are the auditors knowledgeable enough to understand the area they audit?

Development Process Agreements

These documents describe the development methodology and the process to be used for each project

1. To find out aspects which seems overly zealous and areas which would be likely that production staff would prefer to avoid

Templates for Interviews

Template for Semi-Structured Interview of Project manager

To explain that what the objective is to understand deviations and problematic aspects of how development methodologies are applied, including lack of understanding, where it is unclear that rigour of the methodology adds value, or where for other reasons people have with or without good reasons avoided or not done what was agreed as per the DPA. Explain that I will change names of people, projects and customers to make it impossible for a reader to link it directly to anyone.

1. What is Quality software
2. What software development methodologies to get quality software are you familiar with
3. What are your positive experiences as well as negative experiences with software development methodologies
4. What are your main concerns with the software development methodologies you have used

5. Can you tell me of a few Projects where you or people in your teams have avoided, missed or in other ways not done what was agreed as per the methodology (DPA). I want both cases which have been found in internal or external audits and also cases which have not been found and in particular if anyone have avoided to tell others and the case is not known. Also can you list projects or examples of internal politics related to project management, quality assurance or process adherence in general

6. What are as per your view the most common dysfunctions in software development methodologies

    a. Lack of knowledge of process by production staff
    b. Lack of competence by auditors
    c. Lip service to process
    d. Internal politics
    e. Other reasons
7. In my studies of our internal NCs and through interviews of staff I have found various potential dysfunctions:
    a. Lack of knowledge of process by production staff
    b. Lack of competence by auditors
    c. That developers consider certain process activities being pure bureaucratic and not contributing to the quality of the software
    d. Lip service to process and last minute cover up to satisfy the product
    e. Internal politics, where the reasons for finding NCs or avoiding following process are related to secondary agendas between departments
8. For each project example:
    a. If possible open the DPA
    b. Please explain the deviation(s)
    c. What is as per your understanding the underlying reason for the deviation(s)
    d. Is the intent of the person malicious or not
    e. Can you come up with a good rational explanation for the deviation
    f. Is it possible that the processes are too rigorous and pressure to deliver or other pressure made it more or less impossible to deliver?
    g. What control measures could be introduced to ensure that deviations would not happen?

   h. How do you think the process should be modified, the culture of the company change, or some other adjustment to either avoid this type of deviation or the need of it.
   i. In case of political aspects, can you see any rational reason for the other side's actions?
9. Can you explain cases when you have not let the auditors or QA staff knows about deviations? What was the motive?
10. Have you had any influence of selection of projects for auditing by external auditor??

Template for Semi-Structured Interview QA manager

To explain that what I am looking for is to understand deviations and problematic aspects of how development methodologies are applied, including lack of understanding, where it is unclear that rigour of the methodology adds value, or where for other reasons people have with or without good reasons avoided or not done what was agreed as per the DPA. Explain that I will change names of people, projects and customers to make it impossible for a reader to link it directly to anyone.

1. What is Quality software
2. What software development methodologies to get quality software are you familiar with
3. What are your positive experiences as well as negative experiences with software development methodologies
4. What are your main concerns with the software development methodologies you have used
5. Can you tell me of a few Projects where people in the development teams have avoided, missed or in other ways not done as per the methodology (what was agreed in a DPA). Also can you list projects or examples of internal politics related to project management, quality assurance or process adherence in general
6. What are as per your view the most common dysfunctions in software development methodologies

   a. Lack of knowledge of process by production staff
   b. Lack of competence by auditors
   c. Lip service to process
   d. Internal politics
   e. Other reasons
7. In my studies of our internal NCs and through interviews of staff I have found various potential dysfunctions:
   a. Lack of knowledge of process by production staff
   b. Lack of competence by auditors
   c. That developers consider certain process activities being pure bureaucratic and not contributing to the quality of the software
   d. Lip service to process and last minute cover up to satisfy the product
   e. Internal politics, where the reasons for finding NCs or avoiding following process are related to secondary agendas between departments
8. For each project example:
   a. If possible open the DPA
   b. Please explain the deviation(s)
   c. What is as per your understanding the underlying reason for the deviation(s)
   d. Is the intent of the person malicious or not
   e. Can you come up with a good rational explanation for the deviation

f. Is it possible that the processes are too rigorous and pressure to deliver or other pressure made it more or less impossible to deliver?
g. What control measures could be introduced to ensure that deviations would not happen?
h. How do you think the process should be modified, the culture of the company change, or some other adjustment to either avoid this type of deviation or the need of it.
i. In case of political aspects, can you see any rational reason for the other side's actions?

9. Are you aware of cases when production staff has hidden information about project deviations?
10. Have you had any influence of selection of projects for external auditing?
11. In case you have, have you ever avoided putting forward a project with deviation for external auditing? Why?

Template for Semi-Structured Interview of External Auditor

To explain that what I am looking for is to understand deviations and problematic aspects of how development methodologies are applied, including

- Lack of understanding,
- Where it is unclear that rigour of the methodology adds value, or where
- For other reasons people have with or without good reasons avoided or not done what was agreed as per the Procedures.

If he would refer to any other company or experience he does not need to give any details which can identify the people, company or places he refers to, even if any references are made to Gislen Software I will in the paper change the names so that no reader can identify who it applies to or any other such details which can uniquely identify the conditions. I will change names of people, projects and customers to make it impossible for a reader of the paper to link it directly to anyone.
The author is in no way critical to ISO, but may be critical to when the implementation of ISO means lip service to quality or whenever the process becomes just a bureaucratic instrument. Some papers indicate that a QMS only delivery clear cut business value when the objective of the QMS is true quality (functionalistic view) while when QMS is mainly something the company assumes they have to use (for compliance to bid for government RFP's, marketing purpose) there is little indication of any business value derived from the implementation of a QMS.

Further, ISO 9001:2008 has unlike earlier versions of ISO a strong focus on effectiveness of processes and it is the author's strong belief that effectiveness of processes is not an option but a core part of what creates trust and belief in the QMS.

The study is a case study of the situation at Gislen Software. The methodology is Ethnography which means using various means to study the situation. It is a qualitative methodology, and I am using semi-structured interviews, passive methods such as participation in internal audits and even active methods such as conducting an audit and conducting reviews. With these different methods it is hopefully possible to review the challenges from different perspectives and get a richer picture.
In addition you will be given a chance to read anything written based on your input before the paper is finished.

1. How many years of experience do you have as an auditor of Quality Software
2. What was your earlier experience before becoming an auditor?
3. What is in your view the meaning of Quality software
4. What software development methodologies to get quality software are you familiar with

5. What are your positive experiences as well as negative experiences with software development methodologies
6. What are your main concerns with the software development methodologies in use

7. As an auditor how do you ensure sampling of projects for auditing ensures that the audit covers the main part of the company's software development?

8. What are as per your view the most common dysfunctions in software development methodologies in the industry as a whole and in particular in Gislen Software?

    a. Lack of knowledge of process by production staff
    b. Lack of competence by auditors
    c. Lip service to process
    d. Internal politics
    e. Other reasons

    Are we typical or different?
9. In my studies of our internal NCs and through interviews of staff I have found various potential dysfunctions:
    a. Lack of knowledge of process by production staff
    b. Lack of competence about business risks by auditors
    c. That developers consider certain process activities being unnecessary and pure bureaucratic and not contributing to the quality of the software
    d. That production staff before internal audit have as a last minute cover up fixed documents and process steps?
    e. Internal politics, where the reasons for finding NCs or avoiding following process are related to secondary agendas between departments

    What is your view on each of these? Are they common? What is the reason, how do you work with it and how common or big problem is this in our industry?
10. In the process of sampling projects we have had a significant influence of which projects you have audited. Do you consider influence of Management like this being a problem? Is there a risk that we may influence the sampling or could this in any way affect your ability to judge the quality of our processes?
11. Are you aware of any situations where anyone from our company has hidden information from you?
12. In your view how does internal politics affect the quality of software methodology?
13. In your view how does last minute getting process in order affect the quality of the software produced?
14. How in your view should a company like ours get quality at source rather than fixing things afterwards?
15. Is it common that there is organisational politics between production and QA staff? I am thinking of focus on customer demands and focus on QMS demands? How is such conflicts best managed?
16. In my walk through of internal audits I found that most NCs are only directed at surface practices and very less focused on the core development work. What would you think are the main reasons for this?

17. Given examples from my auditing where I found such NCs, is it fair to give such things as observations even though the lack of proper practices related to test driven development did not have strong support in our quality manual?
18. How to deal with agile methodology in the context of ISO? What should be controlled by ISO and what should be controlled by the team?

Appendix B – Extracts from Internal Audit Reports

The following is a selection of Non Conformities found during Internal Audits. Any Non Conformity not relating to software development methodology has been removed (e.g. things related to obvious human mistakes such as document numbering). Note that any text within square brackets is replacements of real customer, project or personal names.

Document	Id	Description	Corrective actions	Analysis
GS-INR-NCR-085-2.0	**13/PRD-P/03**	**Type** – Observation **Clause 7.3.4 of Quality Manual (GS-QMS-MAN-001-7.0)** **Expected Finding:** All the project and process related documents should be approved before initiating any actions based on such documents. **Actual Finding:** 3. In [Project name] project, the estimate is not approved but the project has been released to the client. [Document names] 4. For [Project Name] project, the test plan is not approved but testing has been conducted based on this test plan. [Document names] **Business Risk:** It may not be possible to trace documents based on which the actions were initiated which could lead to internal and external dissatisfaction and could lead to quality issues	1. The estimate was sent to customer but it was not approved. The delivered release didn't include the feature estimated in the referred document. No further actions suggested 2. In future the test plan will be base-lined before start of QA testing, In case QA manager is not available then testers will self-approve the test plan for the purpose of base-lining	I checked the concerned clause and it only discusses the way documents should be reviewed, hence the expected. The NC is not consistent with the underlying process manual Category: Politics, Auditor Competence
GS-INR-NCR-085-2.0	**13/PRD-P/04**	**Type** – Observation **Clause 7.3.6 of Quality Manual (GS-QMS-MAN-001-7.0)** **Expected Finding:** The status of the test cases should be updated when test are conducted **Actual Finding:** In [Project Name] Changes, the status is not updated for a test case - Test result document id : [Document references]	The feature was tested but tester had missed to update the status. In future we will ensure more detailed reviews are done to see all cases are given status	The wording in the business risk is concerning. The issue is that someone has forgotten to update something, not forgotten to test. This real risk is that we don't know if the test has been conducted or not, but the PL seems to know, but is this just stated or is there evidence for this? There is a chance that the PL is hiding real

		Business Risk: When all the test results are not filled, there is a chance of delivering a product with less quality.		aspects. I intend to ask the project leader and tester if there is evidence that the test was conducted. There may be an underlying lack of maturity from the internal auditor's side. The issue is real but the risk is not correctly written. I am concerned about the discrepancy between the finding and the risk. Categories: Politics, Auditor Competence, Hiding
GS-INA-NCR-098-3.0	**15/PRD-P/01**	**Type:** Observation **Clause : 7.5.1 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected finding:** A development process agreement should be agreed between Project In-charge, Production in-charge / Quality In-charge. **Actual finding:** The current process followed for [Project Name] spring release 2011 does not match with the one defined in the DPA [Document name]" – the DPA specifies the process model followed as "Other: Customer defined" However there is no description of customer defined process in the document **Business Risk:** There may be gaps in the process and quality due to un-clarity in the defined process for a given project. This may lead to delayed deliveries and low quality output as team member may not be aware of their roles and tasks.	The process followed is waterfall model and all other information in the DPA indicates this. The process specification has been left out as other by mistake. As this is an one off human error, I am not proposing any corrective action apart from sending out a mail to the project lead and the QA rep to be more careful during their reviews in future.	As in the last case the issue may be a matter of documentation rather than a matter of actual process deviation. Categories: Politics, Auditor Competence, Hiding
GS-INA-NCR-098-3.0	**15/PRD-P/02**	**Type: Observation** **Clause : 7.5.1 of Quality Manual (GS-QMS-MAN-001-8.0)**	Root cause: Error in [Project Planning System]	The internal auditor could have asked what other methods or evidence is available for control

		Expected finding: The Project In-Charge / Production In-Charge should monitor the project progress; deviations if any should be identified well in advance based on inspection / hold points. **Actual finding:** Milestones specified for the project [Project name] in the document [Estimate Document Name] doesn't tally with the milestone present in the document management system. **Business Risk:** Early identification of any delays in delivery dates may not be possible. This may lead to lack of control in schedule and cost overruns. Customer may not be updated in advance about the delays if any.	[Project Planning System] is not updating the milestones in the scorecard when it is updated in the plan. This has been reported and has been brought to the product responsible and MR's notice. Fixing the errors reported would be a good enough corrective action for this. Till then I will be tracking the milestones manually, client related milestones will reviewed during production review meetings	of milestones in case the software system is not working as expected. Categories: Auditor Competence, Hiding
GS-INA-NCR-098-3.0	**15/PRD-P/03**	**Type: Observation** **Clause : 7.3.5 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected finding:** The testing of developed software at all the stages should be done and the same should be documented to make sure there are conformances to the requirements. **Actual finding:** The Internal test result for the intermediate release done for [Project Name] is not available in document management system. **Business Risk:** Slippage of defects to subsequent project phases and to customer	Though the team has uploaded all documents for this project they missed out on this one single document. Further it is only an intermediate version and hence the risk is not high. We have mail communication between the team members to prove that the internal testing was done. Further one of the team member went on leave right after the release and hence missed to upload The document would be uploaded and shown to the internal auditors for verification	If a document is uploaded or not can hardly be a reason for slippage and subsequent project phases. The business risk is not relevant for the discrepancy Categories: Auditor Competence, Hiding
GS-INA-NCR-105-1.1	**01/PRD-P&T /02**	**Type** – Observation **Clause 7.3.3 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected Finding:** ***7.3.3 Design and development output***	Person reviewing/approving the design document (including self-approval as in this case) should look for traceability matrix and check for completion of the same.	Straight forward case

		Design and development output will be verified to ensure that *1. Input requirements are met.* *….* **Actual Finding:** For [Project Name], traceability matrix ([Document name]) is updated however no information could be found related to the requirements PSP 001 till PSP 006 if those were reviewed during the design review and which part of the design covers these requirements **Business Risk:** Could lead to a product that does [not] meet expected behaviour and requirements	We will update the traceability for [Project name] requirements in the traceability matrix	
GS-INA-NCR-105-1.1	**01/PRD-P&T /03**	**Type** – Observation **Clause 7.2.2 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected Finding:** ***7.2.2 Review of Product requirements*** *...* *7. The requirements review includes* □ *Software quality characteristics* □ *Verifiability of software quality characteristics* □ *Acceptance criteria to software quality characteristics* □*Regulatory and legal requirements and responsibility* □*Environmental Requirements* **Actual Finding:** For [Project Name], some of the requirements e.g. NPQ006, NPQ010, (NPQ008 and NPQ009 are conflicting) is at high level and does not have supporting behaviour of the function to verify the requirements against it to ensure if the requirements are met or not met. It would be good if reviews can include a check if listed requirements are verifiable **Business Risk:** There may be a difference in understanding, implementation and verification of such requirements as different people may interpret requirements differently	I asked [Name of person who handled requirements] to answer this since he was involved in eliciting these non-functional requirements:- [Name of responsible] The requirement specification has been reviewed and approved by the client. The requirements were defined by the customer and NPQ006 and NPQ010 are guidelines for design and hence intentionally not detailed. In other cases where we needed more details to verify during test phase we have taken this up with the client and documented the same, but these specific requirements are as precise as what you can expect for design guidelines. Regarding conflicting	Why was this not sorted out during the audit? When reading through the requirements it is quite obvious that there is no conflict. The only value in this observation is to suggest more clarity in definition of a requirement and a guideline. However in this case the design guidelines are customer defined requirements. I think the NC implies overly rigorous perspective from the auditor/MR. Categories: Auditor Competence,

			requirements mentioned – They are not conflicting, they describe two distinct cases, one when the user does not have access rights where the control should not be shown and one when some criteria is not fulfilled, once that is fulfilled, the control should be enabled. No further actions required	
GS-INA-NCR-105-1.1	**01/PRD-P&T /04**	**Type** – Observation **Clause 7.3.4 and 7.3.5 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected Finding:** ***7.3.4 Design and development Review*** *1. The design and development output will be reviewed as defined and planned for the project* *2. The design and development review will be conducted persons, for the design and development done by the team.* *3. Design and development review records are maintained.4. The design review includes one or more of the following:* □ *Architectural design review* □ *High Level / Detailed design review* □ *Source code review* □ *Software test plan review* □ *Software test results review* □ *User guide review, If applicable* **Actual Finding:** For [Project Name], code reviews are done for different team members but could not trace related actions and closure for all the reported feedback for: [Name] code review done by [Name] on on 28 Dec [Name] code review done by [Name] on 26 Dec Also the product was delivered on 26th Jan 2012 however code review were still being performed as on audit date	The risk has been lowered by informing the client about this and also we ensured that code review closure/verification was done for all the modules before the product went live. We had intentionally prioritised the major bug fixes over the code review closure. This has been a strategic decision and hence we have kept the client informed too. No further actions required.	Since the customer was informed, and since it was not a final release, I cannot see the relevance of the perspective of the delivery being done without finishing the code review. However the lack of review by the technical lead's own code is an issue. The former indicates an internal agenda rather than a customer focused agenda from the auditor. The customer in this case definitely preferred an incomplete delivery rather than a delayed one. It may be worth asking the PL for the evidence of the information to the customer? Categories: Auditor Competence, Politics

		31 Jan 2012 **Business Risk:** Delivery of product which may have potential errors in the code leading to security threat to customer's business		
GS-INA-NCR-105-1.1	**01/PRD-P&T /05**	**Type** – Observation **Clause 7.3.4 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected Finding:** ***7.3.4 Design and development Review*** *1. The design and development output will be reviewed as defined and planned for the project* *2. The design and development review will be conducted by authorized persons, for the design and development done by the development team.* *3. Design and development review records are maintained.* *4. The design review includes one or more of the following:* □ *Architectural design review* □ *High Level / Detailed design review* □ *Source code review* □ *Software test plan review* □ *Software test results review* □ *User guide review, If applicable* **Actual Finding:** For [Project name], class design review ([document names]) comment, actions or minutes are not traceable **Business Risk:** May lead to ambiguous issues on feedback given and actions initiated when any design issues arise in future, also it is a risk then a incomplete or improper design might slip into final delivery and could have an impact of client's business and their customers	Class design was not listed as part of review artefacts for [project name & version]. We had committed to update the design for [Project name & version]. It was a team effort where each person was responsible for coming up/identifying the changes required for their module. High level presentation of the design was given to [Name] and when further changes were done, there is mail communication between [Name] and [Name] (5/1/12). We have also authorised [Name] to self-approve the document. No further changes required	The comment from the PL sounds somewhat avoiding. Is there scope for improvements here? Categories: Hiding
GS-INA-NCR-105-1.1	**01/PRD-P&T /06**	**Type** – Observation **Clause 4.2.3 and 4.2.4 of Quality Manual (GS-QMS-MAN-001-8.0)**	I am not sure of the business risk involved in this. It is not mentioned in this report either. Sprint backlog is a document	Why was no business risk given? Categories: Auditor

	Expected Finding: *1.1 Purpose of GS-QMS-PRC-001-3.0* *Documents required by the quality management system must be controlled. The purpose of this procedure is to define the controls required;* *a) To approve documents for adequacy prior to issue* *b) To review and update as necessary and re-approve documents* *c) To ensure that changes and the current revision status of documents are identified* *d) To ensure that relevant versions of applicable documents are available at points of use* *e) To ensure that documents remain readable and readily identifiable* *f) To ensure that documents origin (determined by the organization to be necessary for he planning and operation of the QMS) is identified and their distribution controlled, and* *g) To prevent the unintended use of obsolete documents, and to apply suitable identification to them if they are retained for any purpose.* **Actual Finding:** Document change control log is not available for [Document Name] [Document Name] For the following analysis document, it could not be traced who created, reviewed or approved the project analysis and it also cannot be ascertained in what status the below listed documents are at the moment: [Project Name] – Supporting document for analysis – Article list.docx [Project Name] – Supporting document for analysis – Encode.docx The analysis document is not clear on the background and scope of the document as some of the stored procedures related to changes made to media database and export are listed and some	which is updated every day during the sprint and is updated by the scrum master. Change log maintenance may not add any value for it. These documents follow standard SCRUM practice and they are basically live. Regarding analysis document, there is no business risk as there are clear mail/Skype conversations on these. However for future to ease the follow up procedure, we could add a change log. Scope and background may not be required in each document as it is an internal document between the team members to ensure that everyone is on the right track. The person reviewing the document would understand all this. I would suggest that you give this as a suggestion for improvement unless there are business risks that I am missing.	Competence,

		are not, it cannot be determined the why certain stored procedures were not listed as part of the analysis document **Business Risk:** [Not given]		
GS-INA-NCR-105-1.1	**01/PRD-P&T /07**	**Type** – Observation **Clause 7.5.1 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected Finding:** ***7.5.1 Production / service control*** **Gislen Software** shall plan and carry out production and service provision under controlled condition through one or more of the following: … 2. Compliance with standards, Development Process Agreement, and acceptance criteria … 4. Monitoring of process parameters (e.g. project progress, variances) and software project characteristics … **Actual Finding:** [Project name] project is completed however the risk status ([Document Name]) for the project left without tracking as the project is closed but the risks are still left with open status. **Business Risk:** The company may not know the actual status of the risk which can lead to decisions which may not be appropriate to the situation or in some cases it may lead to delay in decisions	We have corrected the status and updated the document. The risks were acted on/mitigated. So there is no business risk as anyone looking at the risk document would also read the rest of the comments too.	The issue is once again about the paper work, even though there are risks attached to these, there is no indications of deviation in sense of the work of the developers. Categories: Hiding
GS-INA-NCR-105-1.1	**01/PRD-P&T /08**	**Type** – Observation **Clause 8.2.3 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected Finding:** ***8.2.3 Monitoring and Measurement of Processes Receiving Inspection and Testing*** … *6. The final inspection and testing is carried out by quality assurance personnel for adherence of developed software to its*	We had deployed the version on the 25th on their test environment and we had a few support cases going on in parallel then. We had requested for an extension of date for the release notes. This test scenario document was expected to be released with the release notes.	Is there any way in which PL can provide such evidence in audits or even upload the Skype discussion in document system to ensure that there is evidence? Categories: Auditor

		requirements as documented and as planned in the development process agreement *7. Quality Manager will ensure that the inspection and test including those specified either on receipt, in-process and final, have been carried out by respective people and that the result meets the specified requirements.* … **Actual Finding:** [Project Name] project is delivered on 25th Jan 2012 however key test scenario document which was listed as part of deliverables in [Document Name] was not delivered as part of the delivery till 31st Jan 2012 (audit date) **Business Risk:** There may be issues with the verification procedure which customer may not be able to comment till he / she tests the delivered product.	However the content was communicated between [Customer Name] and [Employee name] on 25/1/12 and there is Skype text available for the same where [Customer Name] has confirmed it. When the release notes were ready, the customer reported a bug and we had to solve it and redeploy. So we sent this document after that feedback cycle was over. Also we had to update this document after the feedback cycle. As the customer has confirmed the same via Skype on 25/1/12, I don't foresee any business risk there. [Skype text discussion listing all steps which were successful] and ending with text "Yes! It looks good!"	Competence
GS-INA-NCR-105-1.1	**01/PRD-P&T /09**	**Type** – Observation **Clause 7.3.1 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected Finding:** **7.3.1 Design and Development Planning** 1. A development process agreement will be agreed between Project In-charge, Production in-charge/ Quality In-charge. 2. The Project In-charge will be responsible design and development planning in line with the development process agreement for each project. 3. The development process agreement will also include appropriate software development life cycle activities or	Sprint burn down chart is not the plan. Plan is available in [Project Planning System] Sprint burn down chart is used for tracking sprint tasks only.	Auditors seem to have limited knowledge, why was this not sorted out during audit? Categories: Auditor Competence

		development model, output of various stages of SDLC, estimates, resource requirements 4. The design and development activities are assigned to competent personnel. If agreed in software development agreement, the design documentation will include the software architecture, and other design related details, risks mitigation plan, assumptions, dependencies, formulas and calculation if any. 5. Organisational and technical interfaces of personnel involved in design and development will be identified, and regularly reviewed. 6. In case there is a need to deviate from the standards set forth in the development process agreement, such deviation must be agreed between Project In-charge, Production in-charge and Quality In-charge and either the document should be updated accordingly or records of the approved deviation should be maintained. **Actual Finding:** [Project Name] project Scrum plan ([Document Name]) does not have plan for the complete project, the plan is only available till 17 Jan whereas the project was continued after that. Also the plan does not include plan for integration testing whereas in reality integration testing was performed. The burn down chart is not complete chart for first one week is not available. Code review plan for article list and encode changes were not included in the plan were as it was included for export part **Business Risk:** In complete planning at the time of initiation of project or during the project and unclear plan may lead to risk such as incomplete or poor quality delivery to client		
GS-INA-NCR-105-1.1	**01/PRD-P&T /10**	**Type** – Observation **Clause 7.5.1 of Quality Manual (GS-QMS-MAN-001-8.0)** **Expected Finding:** **7.5.1 Production / service control** **Gislen Software** shall plan and carry out production and service	There is no profit loss in either of the above cases. [Project Name] – We were able to bill all the hours. We checked and it was less than 10% deviation from the whole budget for the project. Have	Why was this not discussed during audit? Categories: Auditor Competence

	provision under controlled condition through one or more of the following: … 4. Monitoring of process parameters (e.g. project progress, variances) and software project characteristics 5. Preventive actions for continuing process improvement 6. Criteria for workmanship. … **Actual Finding:** [Project Name] Planned in [Project Planning System] is 1095 hours Actual in [Project Planning System] is 1362 hours (as on 31 Jan 2012) [Project Name]: Planned in [Project Planning System] is 342.50 hours Actual in [Project Planning System] is 541.17 hours Estimated : 319 hours **Business Risk:** Reduced business profit	kept [Name of manager] in the loop. [Project Name] is a fixed price project and we delivered on time. Extra hours spent does not bring down the profit.