Mentality Patterns: Recurring Turns of Mind as First-Class Concerns in Software Engineering

Thesis submitted for the degree of Doctor of Philosophy at the University of Leicester

Georgios Koutsoukos

July 2011

Abstract

Mentality Patterns: Recurring Turns of Mind as First-Class Concerns in Software Engineering

Georgios Koutsoukos

A wide variety of sources indicate the existence of certain recurring turns of mind, usually referred to as mentality, that have significant impact in software engineering practice. Some of those turns of mind are established to the point that certain designations, for instance “not invented here” or “us and them”, have already been attributed to them. However, whereas agreement on existence is clear, there is significant ambiguity or even inconsistency in the way those are discussed and considered. In other words, there is a noticeable absence of a standardised and systematic means to define, characterize and communicate such recurring mentality elements. Consequently, existing knowledge and practices on the matter are kept on people’s minds or used in narrow contexts. Moreover, very little has been published on methods that can assist colleagues to approach the subject in their work practices in a more organized way.

This thesis reports on research performed over several years, from both a researcher and practitioner perspective in the “real-life” field, and makes the following contributions:

- It presents the notion of *Mentality Pattern* as an abstraction and representation primitive through which we can capture, make explicit, systematise and communicate such human-mentality elements.
- It uses the primitive to define a *Mentality Innovation Sub-process* as an organized way to infuse such mentality issues as first-class concerns into software engineering practice.
- It provides a support system through which a repository of mentality patterns and associated knowledge and experiences can be built and shared.

Results in practice are very encouraging in what concerns the capacity of the Mentality Pattern primitive to organize different perceptions, facilitate the identification of recurring “mentalities” and act as a common communication mechanism.

Moreover, there is evidence that for some mentality patterns the sub-process can drive a constructive change in the way people operate in teams. On the other hand, there exist recurring mentalities that are more persistent.

Finally, based on relevant findings, this thesis calls for an intensification of research on the mentality phenomenon in software engineering and makes concrete recommendations in that respect.

Acknowledgments

I would like to thank Prof. José Luiz Fiadeiro for being both mentor and friend all those years. This thesis would not have been possible without his encouragement, guidance and continuous support, both moral and practical.

I am also grateful to Luis Andrade for providing me the opportunity to develop myself in the innovative environment of ATX Software and for constantly trusting my capacity as software professional. His useful insights in the initial phase of my work are also appreciated.

I would also like to thank:

Prof. M. Holcombe, Dr. A. Lopes, Dr. J. Pissarra, Dr. V. Stenning, D. Vallianos, Dr. R. O'Connor, and T. Kotridis for their encouraging comments at the initial stages of this work.

All colleagues at Oblog Software and ATX Software for receiving me so well during my "adventure" in Portugal and for our excellent collaboration all those years. Special thanks to P. Dimenza and N. Ferreira for engaging in useful discussions and providing valuable comments during the course of this work.

The members of the Department of Computer Science at the University of Leicester for their constant interest on my research progress, the helpful discussions on the matter and, mostly, for making me feel at home during my numerous visits.

All colleagues at Eurobank EFG that I collaborated with during the last four years.

Special thanks for several, though different reasons, are due to D. Melas, P. Mathiopoulou, A. Prodromidis and N. Fertakis. Also, many thanks to S. Gamvroulas, A. Koidis, L. Palaiologou, K. Pateli, A. Dailianas, V. Atherinos, C. Karatzas, E. Kiriakaki, A. Georgiadis and H. Margaritis for sharing their views on the matter.

I also thank Veronica and Eleanna for respecting my aspirations, putting up with my absences and providing moral support.

Finally, my parents, brothers and sisters — for numerous reasons.

List of Publications

- Koutsoukos G. Mentality Patterns: Recurring Turns of Mind as first-class Concerns in Software Projects. In *International Journal of Technology, Policy and Management (IJTPM), Special Issue on Human Aspects of Information Technology Development*, Volume 8, Issue 1, pp. 5-21, Inderscience Publishers, 2008.

- Koutsoukos G. Mentality Patterns: Capturing and Dealing Explicitly with Recurring Turns of Mind in Software Development. In I.Richardson, P.Runeson, R.Messnarz (eds.) *Proceedings of the European Systems & Software Process Improvement and Innovation Conference 2006 (EuroSPI 06),* Springer Verlag, LNCS 4257, pp. 147-158, 2006.

Contents

Chapter 1

INTRODUCTION

1.1 Personal Motivation

The research and direction one chooses to follow is directly related to various aspects such as background, personal philosophy and interests, influence received from mentors or other colleagues, work experiences over the years, and the resources available for research. In our case, the set of events that triggered the research reported in this thesis go back to the year 1999.

At that time, I was a young graduate engineer working in a setting quite different from Software Engineering (SE): a manufacturing plant. After an incident in one of the plant's machines that stopped production for several hours, I witnessed the following dialog between some senior engineers and the plant director:

— Engineers: *"...we assumed that the machine would resist such conditions. It made sense that it would..."*

— Plant Director: *"…and why haven't you consulted the machine's manual...?"*

— Engineers: *"actually we did, but could not find any relevant information. We thought of contacting the manufacturer but this would take time so we proceeded with our initial plan — we all agreed that our assumption made sense and we expected that everything would work fine..."*

Almost one year later, I was already at the beginning of my professional career as a software engineer in a small software company. To my surprise, an event similar to the above took place. In this case, it concerned a newly installed system that had been custom-built on behalf of a customer. After some days of operation it was found out that, under certain circumstances, the calculations performed in one of the software modules were slightly wrong, resulting, however, in a cascade of wrong calculations in other modules, which in turn led to significantly wrong outputs. The system had to be corrected and re-tested, a task that took approximately two weeks.

This led to disruption of customer's operations which, obviously, caused major customer dissatisfaction. As in the case of the manufacturing plant, in a meeting of the project team I witnessed the following dialog between the system senior architect, the system senior analyst and the project manager:

— Project Manager: *"I can not believe that this happened…how is this possible? We had developed thorough specs and had the customer sign-off and both we and the customer had tested the system…"*

— Senior Analyst: *"yes we had done all that…but when we discussed about the particular circumstances that caused this problem, we thought that it was not necessary to be very explicit about it…neither to test…it made sense that it would work as normal…"*

— Project Manager: *"so you made an assumption…OK…did you validate it with the customer? I was not even informed about this…"*

— Senior Architect: *"we made several assumptions for all the system functionality… we could not have validated each and every one of them with the customer…we did it for some that made sense to do so…we were in a hurry as you know…going back and forth with the customer would cost us time, and we did not have any time to spare…after all, the assumptions we made were valid…apart from this particular one…but it made a lot of sense…it was a surprise…"*

After the two incidents above I kept asking myself the following questions:

- *What actually went wrong? Was it a matter of incompetence of the people involved? Was it a problem of the procedures followed? Or was it something else?*
- *Is the common denominator of the two incidents — what I called "making assumptions and not validating them" — the only thing that can lead to problems or are there any other similar aspects related to human ways of thinking that can cause problems and constitute bad practice in projects or work tasks in general?*

- *What does the scientific literature or other software practitioners report on similar "turns of mind"? How can they be detected and taken into account in day-to-day practices?*

I decided that while the cost of ignoring these questions is potentially high, the benefits that may be realized by addressing them could be extremely compelling. From that point onwards, I started to observe such elements extensively in my work context for approximately 4 years. The results were not very dissimilar to the "making assumptions and not validating them" incidents outlined above. Neither were they different to what other practitioners had been reporting, some of which are pointed out in Section 2 of this chapter. In particular, I noticed that there were numerous cases concerning either the results of work tasks or teamwork issues that could be attributed to *characteristic, recurring attitudes, beliefs or ways of thinking of persons or even groups* — what I ended up calling a *mentality pattern*.

One such pattern that struck my attention quite early on and, therefore, is worth mentioning, is what I call "Fear to Admit Ignorance": many colleagues do not explicitly admit having limited or no knowledge of something, for instance theories, technologies or particular solutions. The consequences of this attitude range from minor ones, such as communication gaps or misunderstandings that often lead to conflicts in teams, to more severe ones such as wrong planning or delays in performing tasks.

From the early stages of my research, I was convinced that this was not merely a problem of technical competence or skills of the people involved: the vast majority of my colleagues were highly regarded by their peers and considered to be very experienced and knowledgeable professionals. Neither was it a process-related problem: in most cases, the processes followed were considered to be successful and sound by the managers of the respective organizations. It was at this point in time, and after having researched the related scientific literature, that I realized that this matter was not having the attention it deserved: there were no systematic means to describe and communicate such human-related elements in a consistent way; neither were organized mechanisms available to take those elements into account in SE practice. For the last 6 years, since 2005, I extensively explored the matter from both a researcher and practitioner perspective, applying sound research methodologies in real-life contexts. This thesis is the result of this research.

1.2 Context and Research Goals

The importance of the human element in software practice has been historically emphasised by many authors and has been considered from a variety of perspectives: psychological, cultural, managerial and organisational, just to name a few. Classic works in the field include, among others, the “Psychology of Computer Programming” [76], “The Mythical Man-Month” [13], “Peopleware” [22], the “Personal Software Process” [39], the “Team Software Process” [40] and the “People Capability Maturity Model” [20]. With the advent and increasing adoption of Agile Software Development Methodologies [1] this message has been further amplified. As stated in [17]:

“People's characteristics are a first-order success driver, not a second-order one. [...] Most of my experiences can be accounted for from just a few characteristics of people. Applying these on recent projects, I have had much greater success at predicting results and making successful recommendations. I believe the time has come to, formally and officially, put a research emphasis on what are the characteristics of people that affect software development, and what are their implications on methodology design”.

Although the publications above look at the human dimension from different perspectives, a common element can be found that is also shared by many other texts in the field: the use of the term “mentality” as a factor that significantly impacts Software Engineering (SE) projects, teams and individual tasks.

- DeMarco writes in [23]:*“The **can-do mentality** has the effect of stopping bad news from moving up a hierarchy. [...] The can-do organization then stays the course, ignores the truth that is known at all the lower levels, and thus escalates what might have been a minor setback into a true disaster.”*

- Humphrey [41] has found that *"one of the great disappointments of this process improvement experience has been the failure of the community to break out of its **NIH (not invented here) mentality**".*

- McMahon [57] observes that *"Another pitfall that has been observed when scaling up agile methods on large projects is a **stovepipe-mentality** among the individual agile teams".*

- In attempting to organize a global product team Microsoft [58] reports: *"This arrangement created an **"us vs. them" mentality**, because each team invariably had different priorities and felt the other was making its job harder*". Also, in [2] a software development practitioner writes: *"[...] the result is the classic **us vs. them mentality** that hampers my firm's ability to generate usable software in a timely fashion and creates an unhappy business partner*". Similarly, in a workshop report [27] participants identify as one of the ineffective aspects of OO teams the "***customers vs us mentality***".

- In [72] the authors write: *"The traditional '**siege mentality**' creates an adversarial climate that quickly infects the entire project team*" and in [42] an anonymous practitioner wonders: "*Why Do Some Testers/Test Managers Have a **Siege Mentality**?"*

- In [37], in the context of security, the authors find that: *"This often leads to the we have/require a layered defence – a firewall, user authentication, and file access controls – **what more do we need mentality**".*

- In [62] the author states: *"I know many programmers who employ an acquiescent "**it works**" mentality with respect to their work. Why refactor when it works?*" In [77], Weinberg refers to the term "***debugging mentality***".

Many other authors refer to similar factors using statements in which the term mentality is not explicitly used but is nevertheless clearly implied. For instance, in [13] Brooks discusses the "optimism" of programmers and McConnell [54] states that many problems in software development boil down to what he calls "wishful-thinking", i.e., *"hoping something works when you have no reasonable basis for thinking it will"*. Weinberg in [76] identifies the fact that many programmers attach their ego to their work products and coins the term "egoless programming" to refer to the opposite as a much better practice. Similar comments can also be found in discussions within the software community: copying from a weblog discussion [43], *"Programmers, in general, are extremely secretive. I remember one group had to kick out one of their members because he had like 12 pages of code and wouldn't let anyone else in the group see the printout. Otherwise, this guy was talkative and open."*

This discussion clearly indicates that, over the years, many researchers and software practitioners have concluded that there exist certain characteristic human turns of mind in software engineering teams, typically referred to as mentality or attitude. Those turns of mind have been repeatedly observed across projects and organisations to the point that certain designations, such as "not invented here mentality" or "us and them mentality", have already been attributed to some of them. Finally, there seems to be a consensus around the fact that such turns of mind can have serious consequences for software projects.

Unfortunately, another problem emerges from this discussion that needs attention: those observations concerning mentality issues are currently scattered in books, scientific papers, web pages, project notes and, mostly, people's minds. Moreover, they are often hidden in texts about project best practices, guidelines or recommendations, or even considered and discussed in redundant and inconsistent ways. For instance, whereas DeMarco [23] talks about the negative effects of the can-do mentality, there are several other texts, e.g., on IT job advertisements, in which the can-do attitude is considered an important competency.

As a result, it is not surprising that, although the serious consequences of certain mentalities[1] for projects and organizations have been repeatedly recognised, very little has been published on methods that can assist IT organisations, project managers, development teams and SE professionals in general to approach this subject in a more systematic and organised way. Indeed, in the absence of a clear, standardised and consistent means to name and communicate such mentality elements, it is difficult to even discuss them, let alone consider them in our work practices with the importance that they deserve. Consequently, the specific goals of this thesis are the following:

- **Goal 1:** to provide a systematic and consistent way of capturing, making explicit and communicating such human-mentality elements.
- **Goal 2:** to develop an organized way of supplementing software engineering practices so that mentality elements can be taken into account as a first-class concern.
- **Goal 3:** to provide a support system through which a repository of mentality elements and associated practices and experiences can be built and shared so that SE professionals can make effective use of this knowledge in their practice.

1.3 Direction of Solution

With respect to the three aforementioned goals, this thesis makes concrete contributions in three directions:

i. It presents the notion of *Mentality Pattern* and an associated representation language as an abstraction and representation primitive through which we can capture, systematise, communicate and reason about characteristic, recurring

[1] Throughout this thesis, we also use interchangeably terms such as "turn of mind", "mentality", "mentality elements" and "mentalities".

human attitudes, beliefs and ways of thinking that can have a decisive impact on software projects and work tasks.

ii. It uses the Mentality Pattern primitive to define a *Mentality Innovation Sub-process* through which mentality issues can be infused as first-class concerns into software engineering practice so that IT organisations, project teams and SE professionals can reflect on it, share relevant experiences and make decisions that are better informed.

iii. It specifies the set of envisioned requirements for a support system and repository of mentality patterns and associated practices. Moreover, it employs and extends a leading Open Source solution, namely the Alfresco Enterprise Content Management System (ECM) [5], to provide an implementation of such a system.

The research and solution directions reported herein are based on the intersection of two complementing roles that the author assumed over several years: as an IT practitioner and as a researcher. The former encompasses more than 10 years of professional experience in several organizations and in a variety of roles: as programmer, project manager, architect and IT "strategist", just to name the main ones. The latter covers a period of more than seven years of actively researching the matter, informally at the beginning and later in the context of a (part-time) PhD project. Consequently, the work evolved in a repeating cycle of activities such as literature review, collecting observations from different work settings, theory-building, testing in real-life contexts, reflecting on the results and adapting the theory. In this process, multiple standard research approaches[2] have been applied — field experiments, case studies, surveys and action research — which fits a collaborative practice research approach [51], [52].

[2] The term "research approach" is used throughout this thesis in the sense of a research method (or methodology), paradigm or technique. This is because such terms have been used in different ways in the scientific literature and therefore are open to many interpretations. It is not in the scope of this work to provide an analysis of the different semantics that these terms are used in the literature.

It should be noted that the research reported herein is directed to software engineering (SE) in the sense that it aims to improve judgment and decision-making and, in general, contribute to the advancement of knowledge in the SE field. Although it borrows results of the socio-psychological sciences, it does not aim to develop sociological, psychological or philosophical theories, not even if those are considered in a software engineering setting. Having this context in mind, we list below an indicative, but not necessarily exhaustive, set of research matters that this work does not seek to address:

- It does not aim to prove the existence of certain mentality elements, for instance by investigating large populations.
- It does not seek to provide a comprehensive list of such mentality elements, nor does it aim to scientifically explain how those originate.
- It does not propose methods to perform psychotherapy or to "cure" people by eradicating "bad" mentalities.

We believe that research questions such as the above could be the subject matter of studies performed by socio-psychological disciplines. In this respect, we make concrete recommendations in the Further Work chapter of this thesis.

1.4 Summary of the Remaining Chapters

Having the above in mind, this thesis proceeds as follows:

- In Chapter 2 we provide an overview of the various categories of works found in the broad field of human aspects in software engineering and we subsequently position our research in the appropriate context. Moreover, in Section 2.4 we outline work that has influenced our research and provide relevant comparisons.
- Chapter 3 presents the research methodology and is divided into two parts. The first, Section 3.2, describes the research environment and the way our work has been performed in practice. The second, Section 3.3, relates the aforementioned

research practice to established research approaches found in the scientific literature.

- Chapter 4 is devoted to the first contribution of our work: the Mentality Pattern primitive. In particular, we discuss the definition of the primitive, provide concrete examples of such patterns and present the associated language for representing them in a uniform and consistent way. Moreover, in Section 4.6 we provide a conceptual framework in the form of guiding principles which colleagues can apply in order to evaluate — out of the plethora of recurring turns of mind — the ones that may be characterized as mentality patterns.

- In Chapter 5 we present the Mentality Innovation Sub-process: the envisioned scenarios that the sub-process can be applied, its guiding principles and its detailed operating instructions — “modus operandi”. Moreover, in Section 5.5, we discuss the way widely accepted psychological theories provide a solid basis on the principles and operational aspects of the sub-process.

- The results and lessons-learned by applying in practice the Mentality Pattern primitive and the Mentality Innovation Sub-Process are presented in Chapter 6. Moreover, the experiences of a project management practitioner with the primitive and the sub-process are reported in Section 6.3. Finally, in Section 6.4 we present the results of a survey conducted with a number of software practitioners concerning the mentality matter.

- Chapter 7 is devoted to the support system. Firstly, we discuss its envisioned scope and detailed requirements. Secondly, we provide an analysis of the implementation alternatives, considerations and respective choices made. Finally, the chapter concludes by presenting the system’s implementation details.

- The overall discussion of our thesis takes place in Chapter 8. At a first stage, we position our work with respect to the diverse types of software engineering research in what concerns the nature of research questions and respective results. Subsequently, having this context in mind, we discuss the concrete contributions and limitations of our work. Finally, in Section 8.2 we examine the quality of our research based on the criteria of credibility, dependability, confirmability and transferability.

- Finally, in Chapter 9 we make concrete recommendations for further work along the respective contributions of this thesis: the Mentality Pattern primitive, the Mentality Innovation Sub-process and the support system.

Chapter 2

BACKGROUND WORK

2.1 Overview

This chapter has two objectives:

- To provide an overview of the various categories of works found in the broad field of human aspects in software engineering and, subsequently, position this research in the appropriate context.
- To outline work that has influenced this research and to provide relevant comparisons.

2.2 Human Aspects of Software Engineering

There is a multitude of works in the scientific literature that have as their central piece of study the human element in relation to different parts of the software engineering activity. Such works fit in a broad group that can be called "Human Aspects of Software Engineering". In what follows we aim to provide a suitable categorization of this group in order to put our research in context.

In reviewing the relevant literature, one can recognize two distinct, broad categories of works belonging to the aforementioned group:

- The first is mostly concerned with the human element of the *users* of computer systems and its implications on the design, development and use of such systems.
- The second category examines the human element of those involved in the software engineering activity, i.e., predominantly the *software engineers*, and its significance in practice.

The former category is usually known as "Software Engineering Ergonomics" or, often misleadingly, "Human Factors". Works in this category have different focuses and perspectives, the most widely known fields of study being Human-Computer

Interaction (HCI), Usability and User-Interface design, Ubiquitous and Pervasive Computing and Computer-Supported Cooperative Work (CSCW). Clearly, as explained in the introductory chapter, the emphasis of our work is entirely different to the one on software engineering ergonomics. Hence, a further analysis of the aforementioned fields of study is irrelevant in our context. Instead, we proceed to explore the latter category, which we call "Human Aspects of Software Engineers" in order to differentiate it from the aforementioned "Human Factors".

This second category encompasses works that belong to different *themes of interest* with each theme consisting of several *topics* of particular focus. In what follows we provide a classification of the main themes found in the literature together with their key topics of focus. Our aim is not to be exhaustive but to provide sufficient detail that can place this work in the right context. This classification consists of three broad themes of interest, which are discussed below:

1. *Individual-oriented*: this is the oldest theme, whose origins date back to the 50's. It consists of works that predominantly deal with the different characteristics of software engineers when examined *as human individuals*. Its main topics are related to the following:

    - Personality traits, for example which personality types are more suitable for performing certain SE tasks — requirements, development, design and testing.

    - Individual differences and their impact: how individual software engineers differ on certain characteristics such as personality, motivation, intelligence, interests and values and the impact of such differences in practice, for instance on performance and productivity.

    - Cognitive elements of software engineers: studies that are concerned with the way the human mind operates and processes information, and its relevance to the different tasks that software engineers perform, e.g., learning, program comprehension and problem-solving.

A first identification of the various topics in the theme was provided by Curtis [21] and, to a certain extent, is still applicable today.

2. *Team-oriented*: includes works that are concerned with the *social perspective* of software engineers, i.e., their existence in a team and the ways they collaboratively perform their tasks. Dominant topics are the following:

   - Team dynamics and behaviour: understanding and reasoning about teams' aspects such as patterns of communication and coordination, informal roles assumed, relationships, influences, patterns of dominance and emotions, their resulting behavioural effects and the corresponding implications in terms of the team itself (e.g., on team cohesion) and the tasks at hand (e.g., software development).
   - Learning and knowledge sharing: the ways teams collectively learn, establish common mental models and share knowledge.
   - Team organization, composition and collaboration models: the different models to organize teams in terms of roles, tasks and specializations and the paradigms and ways to approach in a collaborative way specific SE tasks, e.g., requirements and programming.
   - Multi-location and community-based SE: refers to approaches for organizing and performing software engineering tasks with participants in different locations, in virtual teams, as well as community-based SE, e.g., open-source software development. Cultural, diversity and ethical issues are also included in this topic.
   - Tools and infrastructures that support collaboration in SE teams.

   An overview of the theme's scope and topics, though to some extent different to the one above, is provided in [3], [83].

3. *Governance-oriented*: this theme refers to works that consider the human element in software engineering from *a process, managerial and team-leading perspective*. Its two main topics[3] are the following:

    - Software Processes, particularly the field known as Agile methodologies[4] [1], which consists of a set of process methods that advocate a people-centric approach in software development.

    - Software Project Management and Team-leading practices and paradigms that focus on the human dimension, e.g. [13], [22], [55].

Evidently, works in any of the above themes and topics may associate the particular human element they examine with any other constituent of the software engineering activity: particular tasks (e.g., requirements engineering, systems' design, development and testing), paradigms, artefacts (e.g., tools), team properties (e.g., cohesion), inter-alia. For example, [34] examines the relationship between personality composition of teams and the team performance. Moreover, the borders between the topics identified are not absolute in the sense that a given work may simultaneously relate to multiple topics.

Additionally, apart from the above themes there exist works, usually known as Empirical Studies, which employ quantitative or qualitative research approaches in order to report on the way software engineers actually operate in practice. In that sense they are complementary to the above themes and topics and are often applied in order to validate theories or make recommendations for improvements[5].

[3] The two topics identified are interrelated since each often borrows, adapts and feeds back certain practices to the other. However, the distinction is justified by the fact that the Agile set of methods is an established field on its own in the literature.

[4] "Agile Methodologies" subsume a number of different methodologies that share similar principles, the most well-known of which being Extreme Programming, SCRUM, Dynamic Systems Development Method (DSDM) and CrystalClear. It is not in the scope of our work to discuss these methodologies explicitly.

[5] Empirical studies are not limited to the human aspects of software engineering. They are employed in any part of the SE activity, for instance on paradigms and technologies used in practice.

Simultaneously, since the human element is also the central piece of study of social and psychological disciplines, there is a degree of intersection between such disciplines and human aspects of software engineering. The typical scenario found in the literature is that of software engineering works that employ theories, methods or results of socio-psychological disciplines in order to formulate SE theories, explain proposed frameworks and approaches, perform studies, evaluate results, inter-alia. Evidently, different works in SE tend to utilize the socio-psychological ones that best match their objectives and perspectives. For instance, works in Human Computer Interaction, Usability and the Individual-oriented theme usually utilize works from Cognitive and Personality Psychology, whereas works under the Team-oriented theme mostly employ Social Psychology. A pictorial representation of the above discussion is provided in Figure 1 below.

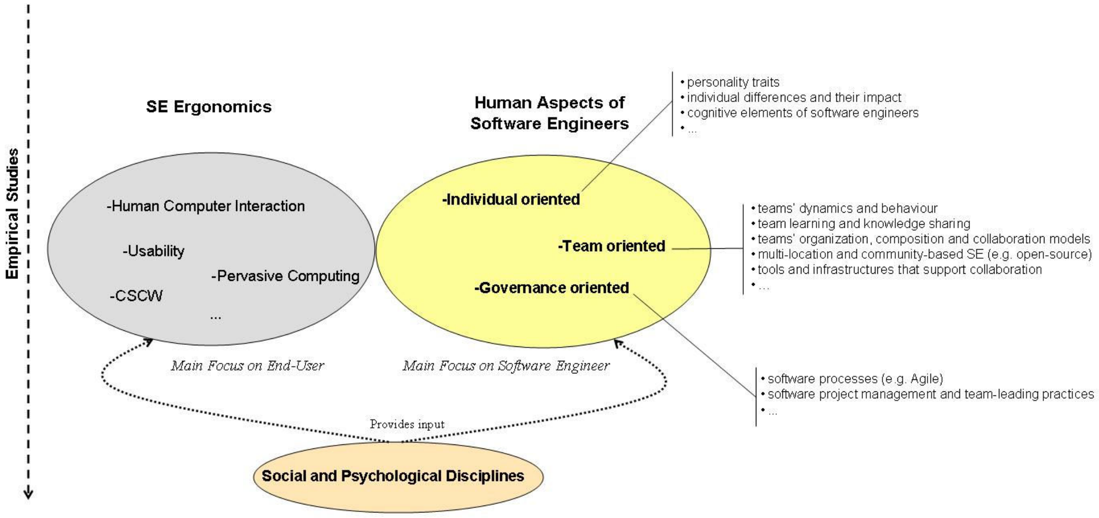

Empirical Studies
SE Ergonomics
-Human Computer Interaction
-Usability
-Pervasive Computing
-CSCW
...
Human Aspects of Software Engineers
-Individual oriented
-Team oriented
-Governance oriented
• personality traits
• individual differences and their impact
• cognitive elements of software engineers
• ...
• teams' dynamics and behaviour
• team learning and knowledge sharing
• teams' organization, composition and collaboration models
• multi-location and community-based SE (e.g. open-source)
• tools and infrastructures that support collaboration
• ...
• software processes (e.g. Agile)
• software project management and team-leading practices
• ...
Main Focus on End-User
Main Focus on Software Engineer
Provides input
Social and Psychological Disciplines

Figure 1. Overview of Human Aspects of Software Engineering

2.3 Where this work stands

As explained in the previous chapter, the goals and contributions of our research are not concerned with the explanation and analysis of the SE mentality phenomenon in general or the examination of the substance of its manifestations, i.e. of particular mentalities. For example, it does not seek to identify the causes or examine the factors that influence the emergence of the phenomenon and its instances; neither does it discuss how given mentalities are related to individuals or teams or provide a comprehensive list of them — it does not even claim originality in their identification. In other words, our research does not study SE mentalities per se. Consequently, in relation to the themes identified above, it does not examine the relationship of human personality, other individual characteristics and cognitive factors to mentality patterns. On the same grounds, it does not associate such patterns with teams' collaborative models or the various elements of teams' dynamics.

Instead, based on significant evidence on the existence and impact of certain mentalities in SE, our overall objective is to provide a systematic means for capturing and communicating specific manifestations of the phenomenon (the mentality-pattern primitive) and, by capitalizing on such means, to establish appropriate methodological support (the mentality innovation sub-process) in order to explicitly take such mentalities into account in people management and team-leading practices. Therefore, this work fits in the Governance-oriented theme and in the corresponding Project Management and Team-leading topic. In contrast to the affiliated Software Processes topic, the above statement is justified by the fact that the aforementioned means are independent of any particular software process as further explained in Chapter 5.

In addition, as discussed in detail in Chapter 3, a significant part of our work is empirical in the sense that established qualitative research approaches (essentially collaborative practice research [52]) have been applied in order to produce, adapt and assess its theoretical constituents.

Finally, similarly to other works concerned with the human aspects of software engineering, theories and results from socio-psychological disciplines have been borrowed. In particular, Social Psychology and its specific field related to the study of human attitudes [7] has been used in the definition of the mentality pattern primitive (see Chapter 4). Moreover, works related to the theories of Cognitive Dissonance [26] and Self-Perception [10] have been employed in order to provide the appropriate foundations to the mentality innovation sub-process. Those are discussed in detail in Section 5.5.

2.4 Influential and Related Works

The previous section discussed where our work fits in the scientific literature. This section relates it to specific works.

Human Aspects of Software Engineering

The significance of the human element in software engineering has been identified in a number of works, which have been very influential in motivating and guiding our research.

– Weinberg has been one of the pioneers in identifying and explaining the importance of human nature and human aspects to software organizations and has written several texts on the matter [76,78,79,80,81]. In fact, his observation and invention of the term "egoless programming" [76] was meant to stress the importance of having programmers that do not attach any ego to their code. This can be regarded as one of the first paradigmatic mentality patterns in the history of SE.

In [78,79,80,81] Weinberg also introduced the notion of "software subcultural (thinking) patterns" and, subsequently, of project management patterns that can be identified and characterized on the basis of such "thinking" patterns. Both such patterns implicitly establish a "maturity" framework in terms of the organizational "culture" towards the improvement of software processes and management practices. Compared to Weinberg's notion of patterns, mentality patterns are at a much lower level of abstraction and focus on individuals and teams rather than the culture and maturity of a software organization as a whole.

– DeMarco's and Lister's work on "Peopleware" [22] has also been very influential in the field. As they argue right from the beginning (pp. 4-6):

> *"[...]the major problems of our work are not so much technological as sociological in nature. [...]The main reason we tend to focus on the technical rather than the human side of the work is not because it's more crucial, but because it's easier to do. [...]If you find yourself concentrating on the technology rather than the sociology, you're like the vaudeville character who loses his keys on a dark street*

and looks for them on the adjacent street because, as he explains, "The light is better there."".

In the subsequent chapters of "Peopleware", the authors proceed by identifying a set of organizational and management practices that ignore this fact, illustrate their consequences and propose alternatives. Although this work is not directly related to mentality patterns, it is has been very influential in intensifying the call for considering the human element as a first-class concern in software practice.

A work that is even more interesting in our context is [24]. Therein, DeMarco, Lister et al capture, in a narrative way, specific, recurring project situations and associated "personas", in what they call "Patterns of Project Behaviour". Their main goal is to exemplify practices, cultures (e.g. organizational, project, team) and people's influence on the successful or less successful outcomes of such situations.

The particular interest of this work in our context is that, in our view, the descriptions of some of such situations also reveal certain recurring mentalities. For example, in the "Film Critics" scenario it is stated that: *"Some management cultures emphasize doing things right, while others emphasize not doing anything wrong. When managers are most concerned about not making mistakes, or at least not being seen as having made mistakes, they send obvious signals, both explicit and tacit, that catching people making mistakes is just as valuable to the organization as doing things right".* The authors consider the emphasis on the "not doing anything wrong" as a "management culture". We believe that this is consistent with our definition of mentality patterns. For instance, it could be characterized as "Fear of Not Doing Anything Wrong" pattern and be directly applicable to any individual or team without necessarily being the effect of a corresponding management culture. In fact, it could even be represented using the language we put forward: at least the consequences element is partially available in the quote above! Since the purpose of our work is not to provide an exhaustive list of mentality patterns, embarking in this direction would be meaningless. The discussion above further illustrates the need for a systematic definition and representation of the mentality element and therefore amplifies the relevance of the first goal of this thesis.

– Agile methodologies [1], which have become a mainstream area of study during the last decade, also advocate a people-centric approach in software development projects. As stated in [18]: *"If the people on the project are good enough, they can use almost any process and accomplish their assignment. If they are not good enough, no process will repair their inadequacy"*. Apart from sharing the same view, some of the practices proposed by Agile methodologies have also proven valuable in guiding the design of the Mentality Innovation Sub-process. The input Agile methodologies provided in that respect is discussed in detail in Section 5.6.

Design Patterns and Anti-Patterns

Although different in its nature and goals, the notion of mentality pattern capitalizes on the proven expressive power and communication benefits of the pattern concept. Such power and benefits have been observed and realized for many years and in various areas, the most notable of which probably being OO design [30].

Our work is also parallel to the idea of anti-patterns as used for software constructs [46], project management [12] and broader management, leadership, cultural and organizational issues [47]. However, where the main goal of those works is in capturing, documenting and dealing with bad software engineering practices, poor management approaches and styles, and inefficient organizational settings, our focus is the human mentality aspect.

Nevertheless, one could argue that mentality patterns can provide justifications for some of such anti-patterns in the sense that a given bad practice or poor approach may be the result of a particular mentality. For instance, in [6] the "Fear to Admit Ignorance" mentality pattern is identified as a possible cause for the "TowerOfVoodoo" and "VoodooChickenCoding" anti-patterns. An illustrative concrete scenario is the case of a programmer that fears to admit his ignorance on the internals of a particular piece of source code, does not ask for help from colleagues, and consequently ends up creating his own additional software layer in the system at hand, i.e. a "TowerOfVoodoo". However, we have to admit that since this has not been the subject matter of our research, we cannot claim that a cause-effect relationship between anti-patterns and mentality patterns can be established with certainty. Still, the “consequences” part of the mentality patterns’ representation language allows software engineers to use the primitive for capturing such potential relationships.

Finally, similarly to the Patterns of Project Behaviour case above, we believe that the notion of certain mentalities is implicitly manifested in some anti-patterns. To illustrate the point, in [47] the authors identify the “Cage Match Negotiator” anti-pattern as a manager that *“either thinks he is always right, or does not care when he knows he is wrong”*. In the authors’ perspective the above statement is just an illustration of a particular way of thinking and is used to further explain the respective anti-pattern. However, in our view, this particular way of thinking can also be considered in isolation: for instance, it may equally apply to a programmer, without necessarily implying the existence of the aforementioned management-specific anti-pattern. In other words, it can be characterized as a mentality pattern.

Psychological Studies

During the course of our research colleagues have made occasional remarks of the following kind: *“Mentality patterns are similar to X”,* or *“Mentality patterns are related to X”,* where X is usually a paradigm from psychological studies. Characteristic examples of such paradigms are Kelly’s Personal Construct Theory [44] and Berne’s book “Games People Play: The psychology of human relationships” [11]. While we do not dispute the good intention of our colleagues in providing insights, our study over the years suggests that there are fundamental differences between such paradigms and our approach: firstly, the goal of our research is not to identify specific patterns; secondly, we have not aimed for a psychological examination of the various characteristics of mentality patterns; and thirdly, our purpose is not to eradicate “bad” mentalities. Having said this, such works are certainly useful in the context of future research developments, a matter that is discussed in Chapter 9.

Chapter 3

RESEARCH METHODOLOGY

3.1 Overview

The purpose of this chapter is twofold:

- In Section 3.2, we present the research environment in which our work has been carried out. Moreover, we describe the way research has been "operationalized": we explain and justify our choice of research methodology and provide details on the activities performed in practice during the course of this work.
- In Section 3.3, we relate the aforementioned research operationalization to established research approaches found in the scientific literature and explain the way they have been applied in our work.

3.2 The Research Practice

3.2.1 The Research Environment

As explained in the introductory section, our research was performed in parallel with a professional career in the IT industry. Therefore, our research environment has consisted of companies (in the context of employment) as well as several other organizations in the context of other professional engagements. In what follows, we present an overview of the main characteristics of those environments, which have informed and shaped our research in a significant way.

Since the beginning of this research, I have been directly employed by three different IT companies located in two countries, namely Portugal and Greece. In the years from 2000 to 2003, I was employed by Oblog Software — a company with approximately 30 staff, which belonged to one of the largest Portuguese financial groups. Its main mission was to develop and manage systems for its mother group while at the same time performing research and developing tools on topics such as model-driven development, software modelling and software architecture. In 2003, I moved to another small company, ATX Software, employing approximately 60 people, also based in Portugal. The focus of ATX was twofold: one the one hand, to develop custom-built software, particularly for the financial sector; and, on the other hand, to develop and sell in the international market products and services mainly related to application reengineering and migration, source-code quality assurance and software architecture. In 2007, I was hired by my current employer, a European banking organization with more than 23,000 people and an established presence in 10 countries. Compared to my previous professional experience, this has been quite different in several ways such as location, organizational size and nature of work: located in Greece, being part of a large, multinational organization and in a corporate IT department which is more an "IT consumer" than a "software house" or "IT producer".

The roles I assumed in such organizations have been quite diverse: programmer, researcher, software architect, pre-sales engineer, project manager, vendor "manager" and IT "strategist". Additionally, while at ATX Software, I also acted as a consultant on software architecture matters for two large multi-national banking organizations. In both cases I was based in the customer's headquarters in Portugal for a period of approximately one and a half years.

In an overall assessment of the research environment and of the roles assumed in relation to the goals of the work reported in this thesis, it is worth emphasizing the following two characteristics:

- The diversity of the environment in terms of size and organizational culture. However, I should also point out that the majority of assignments were related to the financial sector.
- The diversity of the teams and people involved in terms of roles, size, culture, experience, seniority level and position in organizational hierarchy. For instance, as a programmer, architect and project manager, I collaborated mostly with colleagues that had a technical role and background. As a pre-sales engineer I was engaged in several assignments with senior and middle managers of

customers and partners from many different countries (e.g., UK, US, Brazil, Spain, Germany). Recently, as an IT strategist, my interactions included colleagues from all levels of the organizational hierarchy (e.g., IT senior and middle-managers, business managers, users, project managers and, sometimes, even programmers). In many cases, some of those colleagues were based in different locations and countries.

The multi-dimensional diversity of this environment has had a great importance for my research as I was able to gather data, assess and adapt my findings in a considerable number of different SE contexts.

3.2.2 The Research Methodology

The research methodology and practice followed in this work is congruent to the one employed by Cockburn in his PhD thesis [15] on the relation between people and methodologies in software development projects. Cockburn's work has been very influential in the field of Agile methodologies. The way both Cockburn's and my work employ multiple standard research approaches fits into the research methodology of *collaborative research practice* [51], [52]. There are four main reasons that justify the choice of a research practice similar to Cockburn's:

- Although different in the concrete research questions addressed, in both cases the people element is the major component of study.
- In both cases, research was performed over a period of several years and, to a great extent, in parallel to a primary employment in SE-related jobs. In other words, in both cases the research was performed within real-life industrial contexts, i.e., not with mere situational access to such settings.
- In both works, research questions arose and results evolved, were adapted and validated in industrial, real-life environments that share similar characteristics (e.g., the types of roles involved, the types of tasks) and constraints (e.g., time, budget).

- We both have similar goals in what concerns the characteristics of the adopted research approaches. In [16], Cockburn identifies the following five categories of characteristics of the generally-acceptable research approaches:

  - "Real-life", i.e., approaches whose purpose is to apply to real-world situations.
  - "Hi-resolution", i.e., approaches that provide lots of detail and whose purpose is to discover unknown effects.
  - "No Observer Effect", i.e., approaches in which the presence of the researcher does not influence the outcome of the research.
  - "Statistically Sound", i.e., approaches that rely on significant amounts of quantitative data which can subsequently used for generalization of the outcomes.
  - "Controlled Variables", which have as a goal to eliminate irrelevant effects.

  A research approach may have characteristics of more than one of the above categories. However, some of those characteristics prevent certain others. In particular, as Cockburn puts it: *focusing on "real life" and "no observer effect" forces you to abandon "statistically sound" and "controlled variables", and vice versa. Since I'm a consultant and want to intervene in real-life situations, I care more about "real-life" than the others. Since I want to write down techniques that work even when I, as a consultant, am not present, I also care about "no observer effect."*

  As a SE practitioner I was also interested in applying research approaches that have the "Real-life" and "No Observer Effect" characteristic, hence refraining from using the ones that are more "Statistically Sound", allow for "Controlled Variables" and are of "Hi-Resolution". The concrete choices made in this respect are presented in Section 3.3.

Similar to Cockburn's work, my research practice consisted of two interconnected groups of activities[6] called "Thinking" and "Acting", as illustrated in Figure 2 below.

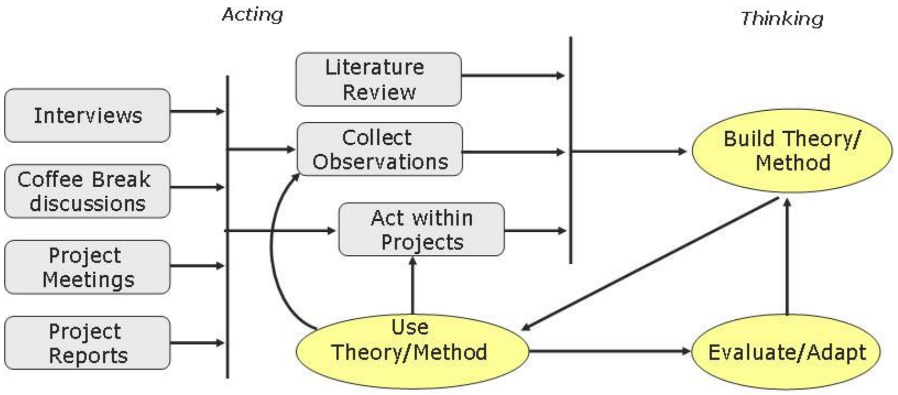


Figure 2. The Research Practice

The "Thinking" group consists of the mental activities performed in relation to the following goals of this work:

- The definition of Mentality Pattern and the establishment of a means to capture and document its instances and of a framework that guides their characterization. This is designated using the term "Theory" in Figure 2.
- The establishment of the Mentality Innovation Sub-Process, its refinements performed over the years and its respective evaluation and results. In Figure 2 this is designated using the term "Method".

The "Acting" group was concerned with the necessary "physical" activities for performing the research: reviewing the relevant literature, acting within projects and collecting and processing observational data. Such data was gathered in any occasion that provided an opportunity for investigation. Figure 2 depicts a sample of such

[6] The research practice described in this chapter and depicted in Figure 2 mainly pertains to the Mentality Patterns primitive (Chapter 4), the Mentality Innovation Sub-Process (Chapter 5) and the respective results (Chapter 6). The repository and tool support part (Chapter 7) is mostly related to developing a concrete piece of software and therefore in this case a different approach was adopted. This is explained in the respective chapter.

occasions, either intentional such as interviews with colleagues or opportunistic such as informal discussions, project meetings and reports. Simultaneously, I also had to carry out the assignments required by the roles assumed as part of my professional life. In what follows further details are given on the main activities of the "Acting" group: literature review, collect observations and act within projects.

Literature Review

The literature review activity can be split into three time periods:

- The period between 2001 and 2003, in which the main focus had been on the mentality issue in SE practice and particularly on what other authors had been reporting. It was at that point that I realized that there were numerous texts that were referring to the matter, though in "ad-hoc" or even redundant and inconsistent ways. Over this period, I also found that the attention given to the mentality element in software practice was limited compared to what, in my opinion, it really deserved.
- The 2003-2006 period, in which I read about software process and project management methodologies as well as about psychological theories and their application in the SE field. The review of such literature is justified by the fact that the main emphasis of my work at the time had been on the definition of the Mentality Pattern primitive and on the design and foundation of the Mentality Innovation Sub-process.
- The years between 2006 to 2011 in which I had been constantly revisiting the literature over the different parts of the research in order to get new insights and compare and adapt the results.

Collecting Observations

The collection and processing of observations related to the mentality matter coincides with the beginning of this research and has been very actively continuing ever since. The goals of the activity have been mostly related to the first two goals of our work as presented in Chapter 1. In particular, observational data have been used to:

- Identify concrete examples of Mentality Patterns, either based on the ones that had been already recognized or implied in the literature or new ones.
- Provide support in the development of a definition of such patterns and of a representation language that is powerful enough for their effective communication and experience-sharing. In other words, such observations have directed me in defining a representation language that is straightforward for colleagues to understand and use. Moreover, I have used such observations in order to identify concrete data in what concerns aspects of the patterns' representation language such as symptoms, anecdotal stories and consequences. Additionally, such observational information has assisted in the development of a conceptual framework in the form of guiding principles that colleagues can apply in order to evaluate which turns of mind may be characterized as a mentality pattern.
- Guide the development of the Mentality Innovation Sub-Process. In particular, such observations have been extremely useful in deciding its main principles and performing preliminary evaluations on its applicability and effectiveness in practice.

The means to collect and process such data have been adapted over the years based on the progress of the work and the experience gained on the matter. During the first years of research I adopted a strategy of being a more reserved observer of what was going on in teams. I was observing specific behaviours, often taking notes of them and was attempting to relate them to the mentality issue. Additionally, I was attempting to give them specific names and identify their possible root causes. Finally, I was comparing them to other behaviours in order to detect possible dependencies and was relating them to the literature theories and findings which I was also studying in parallel.

Often, I was also employing different means in order to drive change in specific ways of thinking and observe the outcomes. For instance, in team meetings or informal discussions with colleagues, intentionally, I was explicitly referring to mentalities such as the Fear-To-Admit-Ignorance, Making-Assumptions and Us-Vs-Them, and was associating them with specific problematic results. A typical example is a statement of the following form in an informal, coffee-break discussion with a colleague:

"Why do you sometimes fear to admit that you do not know something? It's not that bad if you don't – just admit it so that we can plan accordingly. None of us knows everything anyway. You are a much respected colleague in the team – you do not have to worry about anything"

Following such interventions, I monitored subsequent behaviour and validated my findings with other team members. In the case above, I asked colleagues whether they had noticed any change in the way of thinking and the behaviour of the aforementioned colleague in relation to Fear-To-Admit-Ignorance. I have not always been successful in driving the intended changes, but these practices have helped me significantly in setting the main principles and develop the initial design of the Mentality Innovation Sub-Process.

Apart from the above practices, I was also engaging or even intentionally triggering informal discussions in order to get more insights on other people's views on the mentality matter. This helped me a lot to identify concrete patterns, their causes, stories related to them and respective consequences. Often colleagues referred alone to such elements without any intervention on my part.

This practice allowed me to propose a definition and build a first indicative list of potential patterns. It also provided me with the necessary input to develop the first version of a representation language.

In this process, I was often evaluating and abandoning different alternatives and hypotheses. For instance, initially, I thought of using formal specification techniques in order to represent mentality patterns. I still think this is an interesting idea, but given that I was mostly interested in the average practitioner understanding the concepts, I ultimately rejected it. For some time I had also formed the hypothesis that the behavioural element should be part of the mentality pattern definition. I was later forced to abandon this hypothesis for reasons that are discussed in Chapter 4.

In subsequent years I adopted a more active strategy of collecting observations. This strategy was applied in two dimensions: on the one hand, in applying the Mentality Innovation Sub-Process in practice, a matter that is discussed in the following section — acting within projects. On the other hand, on the Mentality Pattern primitive. In this respect, I started to discuss the matter explicitly with colleagues, organizing more focused discussions or even interviews. The objective of such engagements has been to present my outlook and findings, receive feedback and exchange ideas on the matter in general. Consequently, I was able to adapt and enhance my results, validate many of my findings and direct the next steps. In most cases, I discussed with people in team-leading roles since their people experience could provide me with more useful insights on the matter. Naturally, whenever I had the opportunity to discuss with other colleagues, I did not let it go. This practice I continue even now, since colleagues still provide very valuable input.

Acting within Projects

The term "project" as part of the research practice depicted in Figure 2 is used in a broad sense: it refers to any non-trivial task that involves a team for its completion. In other words, my acting in projects was not limited to software development projects, but also included other assignments such as taking part in teams performing a study in order to produce a business case and present it to management, or evaluating products and solutions as part of a bid process. The collection of observational data discussed above refers and applies to all cases that I had been a member of such teams.

During my research, I had the opportunity to apply the Mentality Innovation Sub-Process to a number of software development projects. The roles I had in such projects varied: from programmer to software architect, migration consultant or project manager. A summary of the most relevant background information on such projects is given in Table 1 below.

Table 1. Summary of Projects' Characteristics

Project Characteristic	**Summary**
Project Size	Small to medium, ranging between 1 and 12 man-years effort
Team Size	Teams ranging from 3 to15 members
Team Age	Team average age: approximately 30 years old.
Team Experience	On average, 6 to 10 years of professional experience
Project Types	Projects were concerned with custom-built application development, CASE tools construction and application migration and reengineering.
Technologies Used	The kind of technologies applied has been quite diverse, with mainframe (COBOL, CICS), OO (C++, Java, C#), client-server 4GL development in Oracle Forms and Web-related being the most widely used.
Process Applied	Few projects were operating in a Waterfall-based approach while the majority had adopted, and frequently adapted, agile-based approaches.

When being part of any project, my first priority was to contribute to its success in whatever role I had. In all cases I did not reveal I was simultaneously conducting research. There are two reasons for this: firstly, because project leaders could assume that doing research would interfere with project progress; secondly, because I wanted to minimize the team awareness of being observed and the consequent changes and distortions in behaviour that this could bring about.

In applying the sub-process, I followed an iterative approach from project to project: in some cases, primarily in the first projects, I was selecting parts of it, observing the outcomes and reflecting on the difficulties so as to adapt my practices accordingly and the sub-process itself. In some other cases, I had to restrict myself to certain parts of the sub-process due to certain constraints: for instance, because my role did not fully empower me to drive things the way I wanted; or due to delays in certain tasks that I had to perform; or simply because the overall project atmosphere did not permit it and I was not willing to take the risk. At later stages, as I gained more experience or my role in projects permitted it, I was able to apply the sub-process in a more thorough way. The concrete findings and results of this practice are presented in detail in Chapter 6.

3.3 The Research Theory

In this section, I relate the research practice discussed above to established approaches found in the scientific literature. Galliers [29] identifies 14 approaches that can be used in information systems research: experiments (laboratory and field), surveys, case studies, theorem proving, forecasting, simulation, subjective/argumentative, reviews, action research, descriptive/interpretive, futures research, role/game playing. More recent works in the topic, e.g. [60], [61], include some additional ones of which the more notable are Grounded Theory and Ethnography. In what follows we summarize the key characteristics of the approaches used in this research, also identifying their main strengths and weaknesses. We also explain how such approaches operate in our work.

Experiments

Experiments have two variations: *Laboratory Experiments* and *Field Experiments*. The former have as goal to identify relationships between variables, typically a small number, that are studied in a designed laboratory setting. Quantitative analytical techniques are used with a goal to make generalizable statements that can be applicable to real-life situations. The main strength of the approach is that the control over a small number of variables permits that they are studied intensively. The key weakness is the "*limited extent to which identified relationships exist in the real world due to oversimplification of the experimental situation and the isolation of such situations from most of the variables that are found in the real world*" (Galliers, [29])

Field Experiments extend laboratory experiments into real-life situations, thus achieving more realism at the expense of control. However, in practice, it is difficult to find organizations that are prepared to be experimented on and even more difficult to achieve sufficient control in order to make replication possible with only altering the study variables.

Given that I have been interested in approaches with the “real-life” characteristic, I have not used laboratory experiments. However, I consider some of the practices I have applied to fall into the category of field experiments. The main purpose of such practices was to examine whether the methods I devised for considering explicitly the mentality matter result in any variations in people’s reactions over time.

To illustrate the case, one instance of this practice has been related to the Fear-To-Admit-Ignorance pattern:

“*In a project meeting I deliberately made a suggestion of using a fictional development framework as part of a system to be developed. To my surprise, nobody commented or wondered anything concerning this framework. As expected, I did not insist on my proposal. At a later time, and after having applied the mentality innovation sub-process with some of the participants of the aforementioned meeting, I repeated an analogous reference to a fictional technology. The result was that this time, some of the participants, though not all, explicitly admitted that they had never heard about this specific technology and asked me whether I could provide more information, a request I was able to “get away” with. However, I considered this to be a “successful” result of the sub-process.*”

I have used similar simple experimentation means in some occasions and concerning different patterns. However, I should also point out that in general terms I did not perform extensive or very controlled experimentation and respective measurements. The main reason has been that the matter is quite delicate, particularly since the research environment had been my employment environment and taking into account my concern of not jeopardizing the success of tasks and projects. I consider this fact to be in agreement with the main difficulty of the field experiment approach identified earlier.

Surveys

Surveys "*are essentially snapshots of practices, situations, or views at a particular point in time, undertaken using questionnaires or (structured) interviews, from which inferences may be made*."(Galliers, [29]). The advantage of the approach is that it is possible to study more variables at one time compared to laboratory or field experiments. Moreover, real-world situations are more accurately described and generalizations are more appropriate. On the other hand, it is more difficult to get insights relating to the causes or the processes of the phenomena studied. In addition, there exist possible biases, for instance, the choice of respondents, the point in time that the survey was performed or in the researcher herself in the design of the survey. Finally, the results obtained should be considered as provisional until further assessment is performed in order to validate them.

In the first stages of my research, I employed informal or semi-structured interviews as part of the data collection activity described in Section 3.2. At the final stages, I also performed a survey using structured interviews in order to explore the opinion that colleagues with practical software engineering experience might have on the matter. In particular, the goal of the survey was to investigate:

- The practitioners' view on the existence and impact of mentality patterns
- Possible practice already followed in relation to the mentality matter
- Opinions on the applicability and potential value of the innovation sub-process.

The questionnaire and the detailed answers of the survey can be found in Appendix A. The relevant discussion of the results of the survey is performed in Chapter 6.

Case Studies

As discussed in [33], the exact definition of what constitutes a case study is often ambiguous. Therein, the author defines a case study as "*an intensive study of a single unit for the purpose of understanding a larger class of (similar) units*". *A unit connotes a spatially bounded phenomenon - e.g. a nation-state, revolution, political party, election or person, - observed at a single point in time or over some delimited* *period of time*". In [15], case studies are described as "*descriptive reports of project or episodes from a project. From a research perspective, each case study provides a data point. The strength of the case study is that it captures the local situation in greater detail and with respect to more variables than is possible with surveys. Its weaknesses are the difficulty in generalizing from a single case study and the unintentional biases and omissions in the description. Case studies are helpful in developing and refining generalizable concepts and frames of reference*". In general, case studies can have several purposes, for instance exploratory, descriptive, explanatory or confirmatory. They can also be either qualitative or quantitative in nature [84].

Apart from my own acting in concrete projects, I consider the assessment performed by a Project Management practitioner, whose report is presented in Section 6.3, to be a confirmatory, qualitative case study. This assertion is based on a generic definition of case study such as the ones above. Therefore, I do not claim that it was performed with the methodological rigor that some authors prescribe. Still, it presents interesting results and provides useful insights for further work. Moreover, since it was carried out by a colleague, the weakness of the case study approach in relation to potential bias of the researcher has been, at least to a certain degree, circumvented.

Action Research

Action Research is an established approach whose origins date back to the 40's, particularly in social sciences. Nowadays, this form of research is also popular in the information systems and software engineering domains. In action research the researcher participates directly in real life contexts aiming to improve the contexts themselves, assist in practical problem solving and at the same time expand the scientific knowledge in the problem domain. The main difference to cases studies is that action research may include cases, whereas in a case study the researcher may assume only an observational role (not that of an active participant). According to Hult&Lennung [38]:*"Action research simultaneously assists in practical problem solving and expands scientific knowledge, as well as enhances the competencies of* *the respective actors, being performed collaboratively in an immediate situation using data feedback in a cyclical process aiming at an increased understanding of a given social situation, primarily applicable for the understanding of change processes in social systems and undertaken within a mutually acceptable ethical framework."*

Galliers indicates that the main strength of action research is the fact that the researcher's biases are made explicit in undertaking the research, while the main weaknesses are the "locality" of the results i.e. restriction to a single project and organization, thus leading to a difficulty in generalizing the results. Additional weaknesses are the lack of control over variables and the openness of interpretation.

Action Research is based on a cyclical research process. A frequently cited and used process is the one proposed by Susman and Evered [74]. It consists of five phases to be performed within each research cycle: diagnosing, action planning, action taking, evaluating and specifying for learning. In summary, diagnosing refers to identifying and defining the problem, a phase that in turn leads to action planning i.e. the identification of alternative solutions. Then, a course of action is selected and implemented (action taking). In the evaluation phase the outcomes of the action are analyzed, thus identifying results and findings (learning). Finally, the problem is re-assed and another cycle begins.

Our work is strongly identified with the action research approach. Firstly, I have been an active participant in real-life contexts. Moreover, the fundamental nature of the specific goals of this research, for instance the mentality innovation sub-process, is directly oriented towards practice improvement. At the same time, I also aimed to expand the knowledge in the SE domain. Additionally, as explained in Section 3.2, the research practice followed the cyclical action-research process identified above.

On the other hand, there has been partial divergence to the action's research principle that advocates a very close collaboration and agreement between researcher and participants. This divergence concerns the phases of problem identification, goals setting and action planning and it has been a conscious decision since my goal was to circumvent action's research characteristic that "*the research agenda is strongly* *dependent on how practice evolves*" [52]. Goal setting and action planning performed in close collaboration with colleagues could lead the research into other directions, e.g., on identifying how mentality patterns originate. I do not argue that such an approach would not have resulted in interesting outcomes. Nevertheless, due to the risks involved in such a scenario, particularly given the influences, constraints and different agendas found in a corporate environment, I opted to keep the research focused on the goals identified in the introductory chapter of the thesis.

Subjective/Argumentative and **Descriptive/Interpretive**

Reproducing Galliers, Cockburn [15] refers to the subjective/argumentative approach as "*creative research based primarily on opinion and speculation, useful in building a theory that can subsequently be tested. Its strength is in creation of new ideas and insights; its weakness is the unstructured, subjective nature of the research process. [...] Despite making the prejudice of the researcher known, there is still the likelihood of biased interpretation.*".

The descriptive/interpretive approach is phenomenological. The argument goes something like this: "*All we can ever know are phenomena, since there is no such notion as a "thing in itself". However, once we have understood phenomena correctly, we know all that there is to be known. [...] The strengths in this form of research lie in its ability to represent reality following an in-depth self-validating process in which pre-suppositions are continually questioned and our understanding of the phenomena under study is refined. The weaknesses relate to the skills of the phenomenologist and their ability to identify their biases and unheralded assumptions".*

As outlined in Section 3.2, I performed, over several years, various refinements of the Mentality Patterns definition and of the practices I employed in projects concerning the Mentality Innovation Sub-Process. I consider this to be a form of Subjective/Argumentative research. On the other hand, my reflection on observations and on the results of acting in projects falls in the descriptive/interpretive approach.

Collaborative Practice Research

The combination of methods employed in this research fits in the Collaborative Practice Research approach introduced by Mathiassen in [52][7]. This approach combines three different research approaches, namely action research, experiments and practice studies, and it is suggested as "*one practical way to strike a useful balance between relevance and rigor in practice research*". In the words of Mathiassen, the issue that Collaborative Practice Research attempts to address is that "*when designing and organising research projects based on collaboration with practitioners the challenge is not so much which methods to choose. Rather it is to find practical ways to combine qualitatively different research approaches to support the diverse, and partly contradictory goals involved in such an effort…ideally we want the research process to be tightly connected to practice to get first-hand information and in-depth insight. At the same time, we must structure and manage the research process in ways that produce rigorous and publishable results. Unfortunately, these two fundamental criteria do not always point in the same direction"*. According to Cockburn [15], "*the resolution is to move between the three different research approaches depending on the specific needs and intentions and to obtain benefits from each. Collaborative practice research requires three things, as its name implies. It must be collaborative — the investigator must collaborate with practitioners in both the practices and the research. It must involve practice — the investigator must "do" some of the work being studied. It must involve research — the investigator must build knowledge using the varied research approaches"*. As discussed in Section 3.2, the research practice employed in this work shares the same principles and operates in precisely this way.

The mixture of the aforementioned research approaches in the form of Collaborative Practice Research is depicted in Figure 3 below.

[7] The concrete cases that Collaborative Practice Research is applied by Mathiassen, as well as by Cockburn, focus on the software development activity of SE. However, Mathiassen refers to the approach as being relevant for Information Systems research in general.

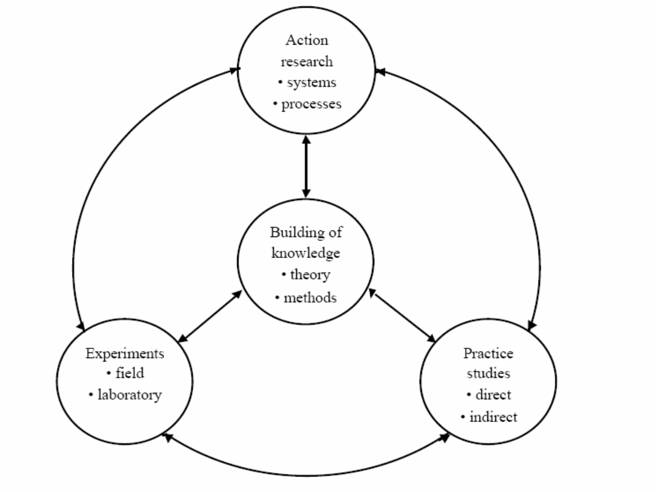


Figure 3. Collaborative Practice Research (from Mathiassen)

In what concerns the Action Research part, our research focused on the process element rather than on the systems being developed. In Experiments, field experiments have been used, but not laboratory ones. In Practice Studies, I have employed both direct methods (e.g., case studies) and indirect ones (e.g., surveys or interviews). This is consistent with Mathiassen's explanation of the way individual research approaches are combined in Collaborative Practice Research. Moreover, the resulting knowledge concerns both theory and methods: the Mentality Pattern primitive pertains to theory, whereas the Mentality Innovation Sub-Process to methodology.

Chapter 4

MENTALITY PATTERNS

4.1 Overview

This chapter discusses the research done towards the first goal and its contribution: to provide a systematic and consistent way of capturing, making explicit and communicating mentality elements that can have a decisive impact on software engineering practice. This was achieved via the *Mentality Pattern* primitive. The chapter is split into four sections: the first three discuss the definition of the *Mentality Pattern* primitive, provide concrete examples of patterns and present a representation language for documenting patterns in a uniform and consistent way. The last section presents a conceptual framework in the form of guiding principles that software practitioners can apply in order to evaluate which turns of mind may be characterized as a mentality pattern.

4.2 The Definition of the Mentality Pattern Primitive

As mentioned earlier, the goal in proposing the *Mentality Pattern* primitive is to be explicit, systematic and consistent about characteristic mentality elements of individuals or teams. In order to achieve this goal, it is necessary to provide, in the first instance, a definition of the primitive itself. In other words, it is necessary to be able to answer the question that most probably comes to mind when one hears the term for the first time: *what is a mentality pattern*?

It should be pointed out that, as discussed later, the existence of a definition does not necessarily guarantee that it will be used systematically and consistently. However, the absence of a definition significantly increases the probability of ambiguity and inconsistency in discussing mentality elements, a fact that has been illustrated in the introductory chapter of this thesis.

Having this in mind, we proceed by examining the definition of the *Mentality Pattern* primitive. Clearly, in this search there are two constituents of the primitive that need to be discussed: mentality and pattern.

The latter is relatively straightforward since the notion of pattern is consistently established in several disciplines, for instance in architecture [4], arts (e.g., in the form of motifs) and in mathematics (e.g., as repeating sequences of numbers). In general, patterns identify or imply sets of elements that are *characteristic* and have a *recurring* nature. Hence, the usage of the word pattern in our context should also reflect those two properties.

The former constituent, mentality, is more complex to define because it is directly related to the scope that the primitive has. In other words, the definition should clearly denote which are the elements that are characteristic and recurring. In exploring the mentality constituent, the first and most appropriate fields in which to look for answers are the socio-psychological disciplines [56]. Moreover, we can also consider how socio-psychological work is applied in other, non-software, disciplines in which the notion of mentality is also defined and applied. One such case from which we borrow results is, perhaps surprisingly, environmental studies [36].

In these disciplines, the conceptual framework proposed for performing relevant studies, distinguishes two elements:

- The mental or cognitive element.
- The behavioural element.

The mental element refers to two components: the cognitive (but non-evaluative) component and the affective (evaluative) component. The cognitive component includes generalized and specific knowledge (including misconceptions), beliefs and belief systems. The notion of belief is used *"to denote acceptance of some statement, proposition or ideology; or as an expressible idea or point of view"* [36]. Belief systems are systems of shared beliefs that regulate human activity, for instance social representations or cultural models.

The affective component refers mainly to attitudes and ways of thinking that, typically, reflect position-taking towards an object (someone or something): the usual empirical definition of attitude is "*a response locating an object of thought along some dimension of judgement*" [56]. Attitudes[8] may lead to what are called behavioural intentions: a predisposition to take some action. The actual actions, what we actually do in practice, comprise the behavioural element above. The relations between the mental and the behavioural elements are a controversial issue in socio-psychological disciplines. Some theories argue for a cause-effect chain between general beliefs, specific attitudes and beliefs, behaviour intention and actual behaviour. Other models consider the mental and behavioural system as relatively independent and not necessarily consistent with each other [36], [50].

The mental elements, i.e., both the cognitive and affective components, are considered to form a mental layer called *mentality*. In other words, the notion of mentality primarily consists of elements such as beliefs, attitudes, ways of thinking and predispositions to take some action. A pictorial summary of the above is provided in Figure 4 below.

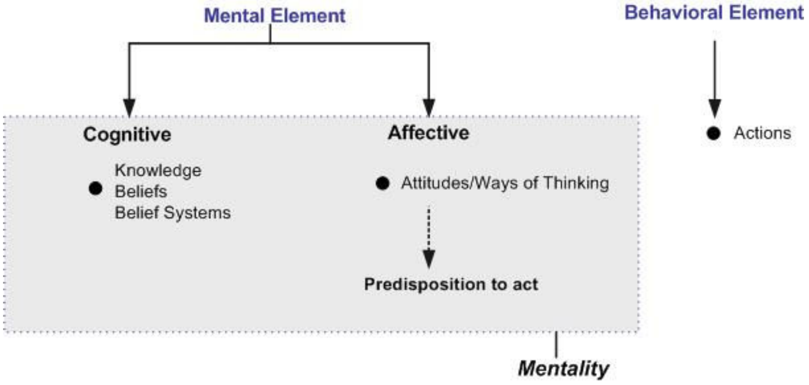


Figure 4. "Scope" and definition of Mentality

[8] The study of attitudes including aspects such as how those are formed and their relation to behaviour is one of the main subject matters of social psychology. Such an analysis is not in the scope of this work, since the goal is neither to study how mentality patterns originate nor to propose methods for performing psychotherapy.

Returning to the SE context, and based on the previous analysis, we define the Mentality Pattern primitive as: *a* ***characteristic****,* ***recurring belief****,* ***attitude [state of mind*** *or* ***predisposition to act]*** *or* ***way of thinking*** *of some (but not all) persons or groups, observed [perceived] in [within] multiple instances of any SE setting [context], without referring to [pronouncing on] any particular aspects (e.g., technical or project) of the context itself.*

The Fear-to-Admit-Ignorance pattern which was presented in Section 1.1 is a good example to illustrate the case: many colleagues do not like to make known explicitly that they have only limited knowledge of the concepts, technologies, solutions or other aspects that are essential for the work in which they are engaged either individually or within a team. This way of thinking, which may also be considered as an attitude towards ignorance, is characteristic and repetitive in several contexts, i.e., it can take place in various SE settings and circumstances. Moreover, it does not refer to the particular aspects of the context, for instance it is not an attitude towards a specific technology, task, or project feature.

The definition above is the result of an analysis performed by borrowing results from socio-psychological related works. A simple "sanity check" of the definition of "mentality" compared to the one provided by a widely-used English dictionary, the Oxford Advanced Learners, indicates a good degree of convergence: the aforementioned dictionary defines the word mentality as a *"characteristic attitude of mind; way of thinking"*. This simple check is important because, as illustrated later, one of the objectives of this work is to ensure that mentality patterns and associated processes can be used by the average software practitioner. Clearly, as practice has shown, difficulty in explaining the concepts in everyday language and reliance on prerequisites on the background knowledge of practitioners would jeopardize the applicability of our work.

4.3 Examples of Mentality Patterns

Based on the definition of mentality pattern presented above, and taking stock on the "on-the-field" research discussed in Chapter 3 and the references to the matter made by a number of colleagues, we list in Table 2 below concrete examples of mentality patterns. The goal is not to be exhaustive but just to provide enough examples that illustrate the point. Moreover, we are not claiming any originality in identifying those patterns: some are probably as old as mankind! In fact, one could argue that some of the patterns captured in Table 2 are not necessarily specific to software engineering but may also be found in other human activities as studied in the context of social and psychological sciences. However, at the present time, we are not aware of any work within social sciences that uses the notions of mentality or pattern in a similar way.

It should be noted that the mentality patterns listed below have a "negative" flavour in the sense that they capture attitudes, beliefs and ways of thinking that can prove to be problematic in practice. However, there also exist "positive" mentality patterns: for example, the patterns that capture the opposite turns of mind listed below. In fact, to describe a mentality pattern with a negative or positive flavour seems to be a matter of choice in the sense that, in practice, we should not work at the same time with one pattern and its opposite.

Table 2. Examples of Mentality Patterns

Mentality Pattern	Description
Fear to Admit Ignorance	Not explicitly admitting of having limited or no knowledge of something (theory, technology, solutions etc).
Better Is The Enemy Of Good	Fear or resistance of further improving or modifying something that already works.
Experience Driven Optimism	Thinking that a problem is easy to solve because it seems similar to something you have done in the past.
Subject Guru	The belief of having incontestable expertise on a topic.
Legacy Person Mentality	The belief that something cannot be done without you or the desire that something cannot be done without you.
It Works! (but I do not know why)	Accepting a result without having a solid explanation of how it was reached.
Not Invented Here!	Tendency to “reinvent the wheel” instead of using existing solutions of others.
Have The Right To Make Assumptions	Making assumptions and not appropriately validating them or explicitly stating them in deliverables.
The Best Is The One I (We) Am Comfortable With	Trying to impose a solution that a person or team is more comfortable with, for instance due to technical or other background.
Opportunistic Listening	Not paying attention to others words or work and in particular defining the level of attention according to the others’ position, age or experience.
It Is Not My Fault!	The attitude of not admitting error or blaming others for failure.
Negativism	Looking for any negative points on others’ approach, opinions or solutions instead of possible positive aspects.
Secretivism	Reluctance in sharing information or knowledge.
Technology Fundamentalism	The blind, rigid commitment to a specific method, technique, tool, solution or vendor accompanied by intolerance to opposing views.
Us and Them	The attitude of viewing other entities either internal (e.g. testing teams) or external (e.g. the customer team) as “enemies”.
No Coding = Useless	The belief that when people are doing non-coding work (e.g. design, documentation, administration) they are contributing less to the project or in general that non-coding tasks are a waste of time.

4.4 The Mentality Pattern Representation Language

In addition to providing a definition of mentality pattern, our aim is to provide a representation language that can be used to provide a common vocabulary for communicating and sharing experiences accumulated within and across projects and organizations. The elements and intuitive semantics of the representation language we have developed for this purpose are given in Table 3 below.

Table 3. The Mentality Pattern Representation

Element	Intuitive Semantics
Pattern Name	A catchy, easy to remember, name for the pattern.
Description	A concise description of the pattern.
Other Names	Other names that the pattern is known.
Related Patterns	Other patterns that the pattern could be related to.
Symptoms	Symptoms that are indications for the existence of the pattern, i.e., hints that help recognize the manifestation of the pattern.
Representative Quotes	Example phrases i.e. verbal statements that indicate manifestation of the pattern. Such quotes demonstrate the "spirit" of the pattern.
Consequences	Consequences and results due to the existence of the pattern
Anecdotal Stories and Examples	Concrete practice cases, drawn from real situations, demonstrating the existence of the pattern as well as its consequences and results. They are usually expressed in an informal way.
Possible Causes	A selective, but not necessarily exhaustive, list of possible causes and reasons from the point of view of human character, human background or working environment (e.g., organizational culture, organizational processes, or project factors) contributing to the emergence of the pattern. Such causes can guide some of the potential actions that could be taken in relation to it. It is an optional section mostly applicable to mentality patterns with a negative flavour.

As examples, we provide representations for the "Subject Guru" and "Fear to Admit Ignorance" mentality patterns in Tables 4 and 5.

Table 4. The "Subject Guru" Mentality Pattern

Pattern Name	Subject Guru
Description	The belief of having incontestable expertise on a topic.
Other Names	None Known
Related Patterns	1. Subject Guru ⇨ Fear to Admit Ignorance 2. Subject Guru⇨ Not Invented Here 3. Subject Guru⇨Opportunistic Listening 4. Subject Guru⇨ Negativism 5. Subject Guru⇨The Best Is The One I (We) Am Comfortable With 6. Subject GuruϟSubject Guru (two Subject Gurus tend to clash)
Symptoms	▪ Expressing opinion when not asked for it. ▪ Providing arguments that are based on experience and status and not on the concrete problem at hand. ▪ Difficulty in accepting others opinion or criticism
Representative Quotes	▪ *I know that stuff very well* ▪ *I have done that before…that's the way it should be done*
Consequences	▪ Sometimes critical tasks are left upon the "expertise" of Subject Gurus with potentially bad results. ▪ Other team members eventually get fed up with Subject Gurus ▪ Subject Gurus may lose their credibility even on topics they know well
Anecdotal Stories and Examples	*In a project we had a colleague that believed he was an expert in everything. "I know this", "I am an expert on that" was his tune every meeting, on every project. He would not let anybody else to speak, made negative comments on virtually everybody's opinion, would not accept suggestions on how he could do things better and he had difficulty in being told what to do. Most of us were really tired of him, we stopped inviting him in meetings and nobody would ask his opinion even if we knew that perhaps he had a contribution to make. Eventually he became so isolated that he had to quit the company. It was a pity, because, although not an expert in everything as he believed, there were some things he knew quite well and could be a valuable asset for us.*
Possible Causes	▪ Need for self-confirmation ▪ Feeling of current status in team being threatened

As stated in Chapter 2, the proposed representation is inspired on design patterns and anti-patterns. Moreover, references to design (anti-)patterns, e.g. [46], may be included in the consequences section, as in the "Fear to Admit Ignorance" example on Table 5 below, to illustrate the direct impact that a mentality pattern can have on software development practice.

Table 5. The “Fear To Admit Ignorance” Mentality Pattern

Pattern Name	Fear To Admit Ignorance
Description	Not explicitly admitting of having limited or no knowledge of something (theory, technology, solutions etc).
Other Names	None Known
Related Patterns	1. Subject Guru⇨ Fear to Admit Ignorance 2. Fear to Admit Ignorance ⇨It works! (but I do not know why)
Symptoms	▪ Silence when one should give an opinion. ▪ Talking about something else or moving the discussion to another point when one should comment on something. ▪ Using the power derived from a superior position in a team hierarchy instead of convincing arguments in order to impose to somebody a solution to a problem.
Representative Quotes	▪ *Is it X... that you mean by Y...?*
Consequences	▪ Programming practices according to the “TowerOfVoodoo” and “VoodooChickenCoding” anti-patterns
Anecdotal Stories and Examples	*In a project we were supposed to use an external API. Most of the people knew very well its functionality. However, there was a colleague that did not. He did not say anything even when the manager asked if everybody was comfortable with the work or there were anything to be taken care before we start. When the team started working, our colleague was, apparently, trying to understand what each piece of the API was doing judging by the method names! Result: It was taking him ages to write a piece of code that was working properly, not to mention the code that seemed to be working but in fact was not.*
Possible Causes	▪ Feeling of superiority e.g., due to position or experience ▪ Feeling of inferiority ▪ A shy person reluctant to ask questions

As illustrated above, the representation language contains references to related mentality patterns. Such relationships can be of two kinds: cause-effect, represented with “⇨” or “!”; or conflict, represented with “ϟ”.

The “⇨” symbol is used to denote cause-effect relations in which a given mentality pattern can be the underlying cause for other patterns. For example, an individual who believes to have incontestable expertise on a topic (Subject Guru) often also exhibits the Fear-to-Admit-Ignorance pattern. This relationship can be represented as:

Subject Guru ⇨ Fear to Admit Ignorance

The symbol “!” is used for implying that the existence of a pattern can trigger the emergence of other pattern(s) in other team members. For example, the existence of the Technology-Fundamentalism pattern in one team can intensify an Us-Vs-Them attitude in another team that has an opposing view; similarly, a person exhibiting the Negativism pattern tends to amplify secretiveness in others.

Technology-Fundamentalism ! Us-vs-Them

Negativism ! Secretivism

Conflict relations, on the other hand, capture cases in which persons with a given mentality pattern tend to clash with peers that exhibit the same or another mentality pattern. For example, Subject Gurus tend to clash with their Subject Gurus peers:

Subject Guru ϟ Subject Guru

Note on the Representation of Mentality Patterns

In these circumstances it is imperative to discuss the reasons why the proposed representation does not explicitly include an element designated to present possible “solutions”, for instance in the form of actions for dealing with the pattern. Pairs of the form problem-solution are very common in the concepts and associated representation languages proposed by most works related to design patterns or anti-patterns. Originally, we also shared this view. However, during our own practice, it turned out that the mentality matter is complex and delicate, especially given the idiosyncrasies of individual people and the particularities of different settings. In other words, establishing a unique pair of the form problem-solution is not as straightforward as in the case of design patterns and anti-patterns. Therefore, providing practitioners with the choice of sharing their subjective views on possible resolution actions on specific patterns would be risky: such recommendations might not be equally applicable in other settings or with different people. As a result, other colleagues adopting such recommendations may not only be unsuccessful, but could also have unintentional effects given the complexity and delicacy of the matter. In our view, the way to mitigate this risk is by performing comprehensive studies in order to reach proven results on how to deal with each distinct mentality pattern. With such studies in place, corresponding recommendations would have a more solid basis instead of being merely subjective and situational. In the absence of such studies, an element of the mentality patterns representation that provides recommendations on actions is meaningless. Hence, we argue for its omission, at least until the materialization of such studies. In contrast, the possible-causes element of the representation language works in a different way. Still being subjective, it provides an opportunity for sharing experiences on the different factors that could lead to a given mentality — how such factors can subsequently be turned into concrete actions entirely depends on the way that each individual practitioner evaluates and uses such information in his particular operational context.

As far as this work is concerned, we provide a complementary means of providing support for dealing with mentality patterns, namely the Mentality Innovation Sub-Process. As explained in the following chapter, such means can be applied independently of any particular pattern and operational context.

4.5 Mentality Patterns Vs. The Behavioural Element

In the definition of the Mentality Pattern primitive, the behavioural element of human socio-psychological nature has not been included. In other words, we have distinguished between mentality patterns and what can be called "patterns of behaviour". The latter refer to specific, recurring actions or reactions of individuals or teams, which can be either general — for example, an individual constantly being late at meetings — or specific to SE, for example the well-known copy-and-paste programming practice. In what follows, we further discuss the relationship between mentality patterns and the behavioural element.

The cause-effect relations between mental and behavioural elements are a controversial issue in socio-psychological disciplines in the sense that predicting behaviour from pre-existing attitudes or beliefs is not a linear problem [56]. Some models argue for a causal chain between beliefs, attitudes, behaviour intentions and actual behaviour (actions), whereas other models consider the mental and behavioural systems to be relatively independent [36].

When considering the relationship between mentality patterns and concrete behaviours, there are two possible options. The first one is to assume that a given behaviour is the direct result of a corresponding, recurring predisposition to act (behaviour intention) and consider the behaviour element to be a mentality pattern in its own right. In other words, by assuming such a linear relationship, the copy-and-paste programming practice can be turned into a copy-and-paste mentality pattern. The second option is to consider a many-to-many relationship between mentality and behaviour elements: a given pattern can be manifested via multiple behaviours and a given behaviour may be the result of many mentality patterns, among other possible cause factors. For example, in the former case, different people fearing to admit ignorance can behave in different ways: one person may stay silent in order not to manifest ignorance while another may attempt to lead the discussion to another topic. In the latter case, the copy-and-paste practice might not necessarily be the direct result of a corresponding, recurring, predisposition to act in such a way, but could be the effect of another mentality, for example the Fear-To-Admit-Ignorance pattern: if one does not know how to appropriately re-factor a given piece of code, but is afraid to admit it and ask for help, one may end up performing the copy-and-paste alternative.

Therefore, given that the relationship between mentality and behaviour cannot be established with certainty, the borders are subtle. Moreover, the mentality elements — beliefs, attitudes and ways of thinking — are always manifested or inferred through the effects that they have: some sort of action (behaviour), for instance when expressing an opinion. Furthermore, such forms of inference are always sensitive to the perception of the observer: what they infer as attitude, belief or way of thinking when evaluating a particular action. For this reason, the choice of the words "*observed [perceived]*" in the definition of mentality pattern is intentional: it is meant to point up that any such pattern captures real phenomena as perceived by the author of the pattern. The value of the mentality-pattern primitive is that it provides the means through which the perceptions of various practitioners can be captured, systematized, made consistent and effectively shared across projects and organizations.

Why not Patterns of Behaviour as the "focal point"?

Evidently, one could argue that, given the more "invisible" and "intangible" nature of mentality patterns, the, directly observable, patterns of behaviour should, instead, be chosen as the point to "put emphasis" on. There are two reasons that justify our choice in that respect.

Firstly, the references that can be found on the matter, some of which were presented in Section 1.2, focus mainly on the mentality element, not on the behavioural one. In other words, in such references, the indications of particular actions or practices (behaviour) are made in an abstract way or are mixed with consequences, typically as a means to justify respective perceptions or give evidence on the existence of a particular mentality.

Secondly, given the difficulty to establish the relationship between mentality and behaviour and the multitude of behavioural variations that can be found in practice, it would be impractical to have behaviour as the focal point. It is mostly a scale issue: such a choice would lead to a significantly increased amount of "behavioural" patterns that would make it unfeasible to systematize, communicate and share experiences about such patterns. Moreover, such a complexity would become even more unmanageable if we take into account the fact that, according to socio-psychological studies, our behaviours are very sensitive and dependent on the particular situational context in which we all operate.

Consequently, particular behaviours are considered as "symptoms" in the mentality pattern representation language, providing evidence or justifying the perception of the existence of a particular pattern.

On the other hand, we have to admit that, in practice, nobody can restrain colleagues from considering mentality patterns that are controversial, i.e., they could also be considered as "behaviour patterns" by other people. However, this does not diminish the need for and value of the mentality pattern primitive. To illustrate the point, it is interesting to draw an analogy with design patterns as used in software development [30]: irrespectively of the fact that there exist well-established definitions of design patterns, associated representation languages and a wide variety of instances, software practitioners may define and use design patterns that are controversial. This does not diminish the proven value of design patterns in what concerns the systematization, consistency and effective communication of software design problems and decisions. We believe that the same applies to the case of mentality patterns. In any case, the sub-process (discussed in the next chapter) through which the mentality element can be explicitly taken into account in SE practice has been designed to be independent of the choice of mentality patterns.

Finally, the statement "some, but not all persons or groups" in the definition of mentality pattern given earlier is meant to distinguish mentality patterns from the psychological tendencies that all humans follow. Nevertheless, such tendencies may provide useful insights and justifications for the emergence of mentality patterns. For example, the It's-Not-My-Fault pattern could be the effect of such a psychological tendency as explained by Attribution Theory: when we make an error, we are likely to attribute the cause to external or situational factors rather than blame ourselves. This potential relationship between psychological tendencies and mentality patterns can point to an interesting research direction as discussed in Chapter 9.

4.6 Towards a framework for capturing Mentality Patterns

The last element of the Mentality Pattern definition that needs to be discussed is the statement that such patterns are *"observed [perceived] in [within] multiple instances of any SE setting [context], without referring to [pronouncing on] particular aspects (e.g. technical, project) of the context itself"*. This part of the definition summarizes a conceptual framework in the form of guiding principles that practitioners can apply in order to evaluate which turns of mind may be characterized as a mentality pattern, i.e., to provide a means for deciding, out of a list of candidates, which ones should be considered as mentality patterns.

The examples of mentality patterns presented earlier in this thesis have also been considered based on this framework. In what follows, the aforementioned framework is further explained.

As stated earlier, mentality patterns are perceived by colleagues in their work practices. Such practices occur in various SE settings with various characteristics, for instance in different organizations, teams, types of projects or domains. We refer to such SE settings as *contexts*[9] and to their characteristics as *aspects* of the context. Work practice and context have also a *time* relation: as software professionals we operate in plenty of different contexts over time: in one or more organizations, in various teams, in a multitude of projects, or while performing diverse tasks, possibly in diverse domains and using different technologies. We denote each of those discrete contexts that one operates over time as a *context instance*.

In the framework we propose the relation between Mentality Patterns and the contextual element is considered in three dimensions:

- The *Persistency* dimension: this refers to the number of different context instances where a pattern is perceived to occur. In other words, whether a turn of mind takes place only within one context instance, even if it is recurring within this context instance, or if it "persists" across multiple instances over time. For

[9] In general the concept of context has been discussed extensively in various areas such as Computer Supported Cooperative Work (CSCW), Context Aware Computing, User Experience and Mobility just to name a few. It is not in the scope of this work to provide an analysis of the different ways context is defined and used in those fields. However, a common element shared between the areas above is that the notion of context mainly refers to computer systems or users of systems. In our case, we are interested in the context that software colleagues operate in practice. Therefore, we adopt a more generic definition of the term as found in the disciplines, namely Organizational Psychology, that have their focus on workplace environments and individuals and teams in such environments. Rousseau in [67] defines context as "*the set of circumstances or facts surrounding an event ... context can refer to characteristics of the organizational setting, of the individual, of his or her role in the organization, and of any other environmental factor that may shape responses*". Moreover, the notion of *context aspect* we put forward in our framework is close to what in Organizational Psychology is known as *contextual variables*.

example, the Fear-to-Admit-Ignorance pattern is a recurring way of thinking that may be repeated over time across different organizations, projects and teams.

- The *Context Aspects* dimension: this refers to the type of the context, as characterized by its different aspects. In [8], the authors propose a framework in which context in software development is characterized by the following aspects: Individual/Personal Context, Roles, Team, Project, Organization, Client, Product, Software Engineering Domain (e.g., technologies used), and Business Domain (e.g., financial services or telecommunications).
- The *Association* dimension, which can be one of the following two types:
  - Patterns *towards* the Context and its Aspects
  - Patterns *within* the Context

  The former, refers to Mentality Patterns that have one or more of the context aspects as their object of thought. In other words, patterns which represent a belief, attitude and way of thinking towards one or more of the context aspects. For instance, if we think of the software engineering domain and tasks as context aspects, one may consider beliefs and attitudes towards elements of such aspects as a mentality pattern of this form. Examples of such expressions could be "*Java is a poor programming language*", "*Testing is not an interesting activity*" or "*Documentation is a waste of time*".

  The latter form, Patterns within the Context, refer to what has been already discussed so far: mentality patterns are always identified by colleagues in their practices i.e. when colleagues operate within different context instances. In other words, patterns originate in context(s) or even influenced by them.

Taking into account the above discussion, we propose the following two principles in order to set specific criteria for identifying mentality patterns. The definition of the primitive is formed in such a way so that it reflects those principles.

Principle 1: *A mentality pattern should have a persistent nature, i.e., observed [perceived] in [within] multiple instances of any SE setting [context].*

Principle 2: *A mentality pattern is a recurring turn of mind that does not refer to [pronounce on] particular aspects (e.g. technical, project) of the context itself.* In other words, we should exclude patterns that have one or more of the specific context aspects as their object of thought. As an example, turns of mind such as *"C++ is a cumbersome programming language"*, *"Testing is not an interesting activity"* or *"Documentation is a waste of tim*e" can all be candidates for being characterized as mentality patterns. However, given that they pronounce on specific aspects of the context — on specific technologies or tasks — we argue that they should not be considered as such.

A pictorial representation of this discussion is given in the image below. The image consists of a set of boxes, each referring to a context aspect along the dimensions of persistency and the association of the pattern to the context. In each set, red boxes marked with (X) indicate types of patterns excluded by the two principles above, whereas blue boxes marked with (✓) indicate types of patterns that are in agreement to the principles.

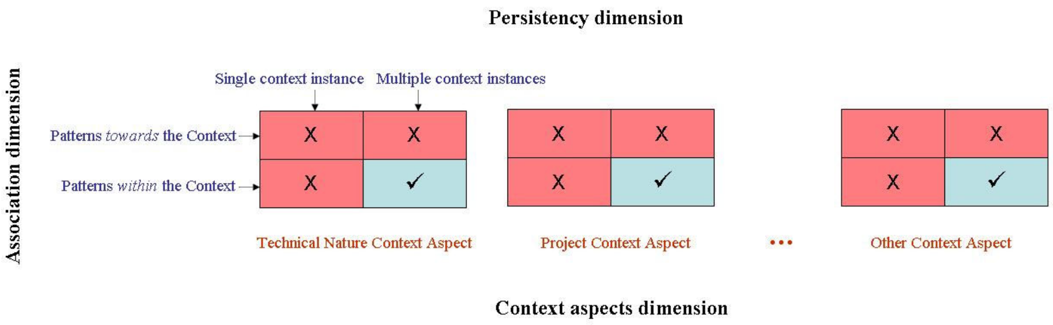

Persistency dimension
Single context instance
Multiple context instances
Association dimension
Patterns *towards* the Context
Patterns *within* the Context
X
X
X
✓
Technical Nature Context Aspect
Project Context Aspect
•••
Other Context Aspect
Context aspects dimension

Figure 5. Mentality pattern and the contextual element

In what follows we discuss the rationale for the two principles we suggest, which is based on a quantitative and a qualitative dimension.

On the quantitative dimension, one should note that both principles act as a filtering mechanism that aims to reduce the number of mentality patterns that can be identified and reported. Given the numerous individuals, contexts, context instances and context aspects that can be found in practice, identifying mentality patterns that are not persistent across context instances (or ones that refer to beliefs and attitudes towards context aspects) would lead to an explosion of different patterns. Such a volume complexity would make it unmanageable to systematize, communicate and share experiences about mentality patterns in an effective way. As a result, the current status of redundancy and inconsistency in identifying and reporting such patterns and associated experiences in the SE community will most likely continue, thus jeopardizing the first goal of this work.

On the qualitative dimension of the rationale, and considering the specific goals of this research, our criterion comprises two factors. The first is the level of interest that the wider community of software practitioners may have in knowing about the pattern. In other words, how interesting is the pattern for other colleagues to know. The second factor is the value that can be achieved if a pattern is taken into account in the SE practice. Having this in mind, we proceed by discussing how the principles in question are related to this criterion.

The principle of persistency across context instances puts further emphasis on the recurring and periodical feature of the Mentality Pattern primitive: a turn of mind should be recurring not only in a single context instance but should also be repeated in several instances. Evidently, this amplifies the interest the wider SE community to know about such a pattern and the value of taking it into account in their work practices: if a given turn of mind is consistently repeated and influences our work practices then we need to act on it. Moreover, by considering non-persisting patterns, one takes the "risk of coincidence": the fact that a pattern "takes place" only within a given context instance may be absolutely coincidental, even for reasons related to the context itself. Patterns that are only "coincidental" can only have limited interest for other colleagues to know and the value of considering them in their work practices is debatable.

The rationale of the principle for not considering turns of mind *towards* the context has also a qualitative dimension. Expressing beliefs and attitudes towards our work context is what actually each of us does every day[10]. We all have likes and dislikes, positive or negative dispositions about different aspects of our work. For instance, not all SE practitioners like to perform tasks such as systems maintenance, testing, documentation. Moreover, we all express different opinions and views about projects and technologies. In a wider sense, all those can be considered as patterns of turns of mind since they are all characteristic and recurring, often across context instances. However, because those particular beliefs, attitudes and ways of thinking are part of the day-to-day *modus operandi* of all software teams worldwide, the interest in and value of any corresponding patterns for people outside the immediate reach of such teams is debatable.

[10] The goal of this work is not to study attitudes towards organizational settings or how context characteristics influence attitudes. This is extensively studied in other disciplines, of which the most prominent is Organizational Psychology.

Chapter 5

THE MENTALITY INNOVATION SUB-PROCESS

5.1 Introduction

In order to make effective use of the notion of mentality pattern, we put forward what we call the "Mentality Innovation Sub-Process" — an organized way to supplement software engineering practices with a means for considering such recurring turns of mind as first-class concerns. The sub-process is independent of any particular method or framework in the sense that it does not depend on any specific team configuration in terms of organization, composition or collaboration model: it only requires collective interaction, for example through team meetings. Because it is complementary to those aspects of any software development process, it is named as a sub-process, not a process. Moreover, its goal is not to "cure" people in a psychological sense. Instead, the goal is to assist software practitioners in driving change in their own environments, share relevant experiences and make better-informed decisions, thus enabling innovation in mentality-related matters.

In what follows, we discuss the envisioned scenarios to which the sub-process can be applied (scenarios of use), its guiding principles and its detailed operating instructions (modus operandi). In Section 5.5, we discuss its psychological foundations.

The principles and detailed instructions of the sub-process are not an idealized, a-priori-developed way to deal with the mentality matter. Instead, they are the distilled product of our practical experience as explained in Chapter 3. Moreover, the sub-process is not a one-size-fits-all prescription to be followed rigidly. Its modus operandi allows for a degree of flexibility in order to account for the particularities of real-life environments. In other words, there exist aspects of the sub-process in relation to which concrete choices should be made "on the field", i.e., guided by contextual information, such as particular people, project and organizational settings. Where applicable, we outline relevant cases and options available.

Another important aspect is that the sub-process operates at the team level. On the other hand, we can not ignore the possibility that some managers, team leaders or people involved in any other role may prefer to approach it on an individual basis, e.g., via one-to-one engagements with members of their teams. Based on our own experience in such face-to-face discussions, we believe that the principles and foundations of the sub-process are still useful at the individual level. However, the goal of our work is not to provide instructions on how such individual engagements should be performed.

Finally, the practices described as part of the sub-process are based on the assumption that teams are, at least to a certain degree, receptive in discussing the mentality matter. In other words, individuals willing to apply the sub-process in their own practices should establish in advance a level of confidence on its receptiveness in their specific environments. In our own experience, two main approaches to examine receptiveness are useful: what we call "occasions of opportunity", i.e., occasions that arise of which there is an opportunity to discuss the matter (a strategy outlined further on); and one-to-one discussions with colleagues on the mentality matter before engaging in concrete actions. However, given that such confidence depends on the judgement of individuals when evaluating their specific environments, this aspect is not the subject matter of the sub-process itself.

5.2 Scenarios of Use

The sub-process requires someone to drive it, i.e., someone responsible for facilitating and leading the way it is applied in practice[11]. In what concerns the envisioned scenarios of use and the corresponding drivers of the sub-process we consider two cases. On the one hand, teams whose membership is essentially invariant, e.g., specific departments of an organizational hierarchy. In this case, the driver can be the departmental head to whom the team members report. On the other hand, variable teams that are assembled for a specific period of time in order to

[11] The author has been the driver of the sub-process in this research.

collaborate in fulfilling a task. We refer to those as "project teams". In this case, the respective driver of the sub-process can be the project manager[12]. The sub-process is applicable in both scenarios[13], but there are certain adjustments in its operationalization details that need to be made in order to account for the differences between the aforementioned configurations. These adjustments are outlined in Section 5.4.

In what follows, we discuss the principles on which the sub-process is based and the subsequent "operating instructions" for applying it in practice. Those are formulated in a language style that primarily considers the point of view of the sub-process driver. In particular, the principles provide the overall philosophy and act as guidance, whereas the specific instructions give details on the way the sub-process execution should be led.

5.3 The Principles of the Mentality Innovation Sub-process

The sub-process is designed and operates in practice based on four principles:

- *Principle 1: Discuss openly on mentality patterns*. Talk about specific mentality patterns openly and explicitly – do not attempt to conceal the matter. Engage colleagues in providing and collectively discussing cases of such patterns that they have identified. Give emphasis on referring to the practical consequences of specific mentality patterns.
- *Principle 2: Commitment to mentality-related operating principles.* Encourage colleagues to commit to a set of principles that we call "mentality-related operating principles". Such principles are directly derived from mentality patterns and guide the way teams and individuals will operate in practice. For

[12] In both scenarios, the driver of the sub-process can be any other individual designated to assume this role. In principle, the task of driving the process can be even performed collaboratively by more than one person. Moreover, in large departments or teams, if deemed necessary, the sub-process can be applied in smaller teams by the respective team leader or other driver.

[13] In both cases, we have applied the sub-process in teams in which all its members are physically present, i.e., not to virtual or distant teams.

instance, for the Have-the-Right-to-Make-Assumptions mentality pattern, a principle of this form can be:

> *Whenever, for any reason — including making our work progress faster, we are making assumptions on system requirements or technical functionality, we will make sure that: if possible, we validate our assumptions with whoever necessary in order to make sure that they hold; if such validation is not possible we will explicitly state our assumptions in our work deliverables.*

- *Principle 3: Feedback, but not finger pointing or hidden agendas.* Provide well-organized, constructive feedback and encourage relevant discussions when individuals and teams seem to be departing from the aforementioned principles. On the other hand, mentality patterns should not be used as a way of blaming specific colleagues explicitly. The emphasis is on discussing specific mentalities and their consequences, abstracting from the identity of individual people. Even if particular mentality patterns are often related to specific individuals of a team, the way feedback is provided should aim to trigger reflection on the matter, not to push individuals to adopt a defensive position by presuming feedback as an accusation. Moreover, mentality patterns should not be applied with a "hidden agenda" in mind, e.g., as a means to diminish other colleagues' views in order to support one's particular position on a given matter. For example, it is improper (in terms of timing, setting and intention) that in a team discussing over the adoption of a particular technology, someone pinpoints a colleague using a statement of the form *"you are a technology fundamentalist"* as an argument for implicitly supporting the opposite view. Such aspects, as well as the idiosyncrasies of individual personalities, should be carefully considered by the driver of the sub-process in their task of systematizing mentality-related feedback.

- *Principle 4: Reflection and Learning.* The team should reflect on the lessons-learned and results obtained from taking into account mentality patterns in particular projects or tasks at hand. For example, if a team or individual operates according to a principle derived by a mentality pattern and achieves positive outcomes, then reflection should take place on the respective lessons-learned and on the advantages or disadvantages in comparison to previous practices.

The above discussion is summarized in Figure 6 below. In the following section we discuss the operationalization details of the sub-process.

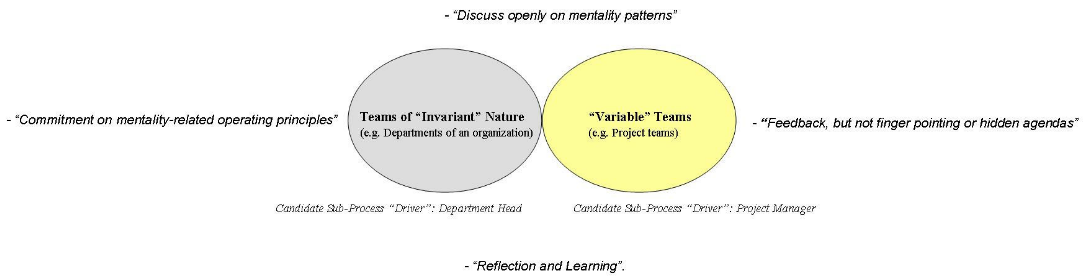

- "Discuss openly on mentality patterns"
- "Commitment on mentality-related operating principles"
Teams of "Invariant" Nature
(e.g. Departments of an organization)
"Variable" Teams
(e.g. Project teams)
- "Feedback, but not finger pointing or hidden agendas"
Candidate Sub-Process "Driver": Department Head
Candidate Sub-Process "Driver": Project Manager
- "Reflection and Learning".

Figure 6. Scenarios of Use and Principles of the Mentality Innovation Sub-Process

5.4 The Sub-Process Modus Operandi

In its overall design, depicted in Figure 7 below, the Mentality Innovation Sub-Process consists of three building blocks: two phases, namely the Mentality Principles Setup and the Mentality Learning, and one continuous activity called Mentality Feedback. In what follows we discuss each of these blocks, providing sufficient level of detail[14] for explaining the way the sub-process can be applied.

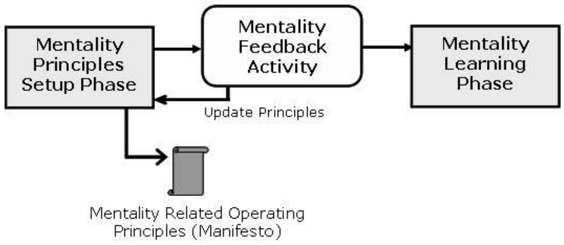

Mentality Principles Setup Phase
Mentality Feedback Activity
Mentality Learning Phase
Update Principles
Mentality Related Operating Principles (Manifesto)

Figure 7. The building blocks of the Mentality Innovation Sub-Process

[14] Unless otherwise stated, the sub-process driver is responsible for leading and facilitating the tasks described.

Mentality Principles Setup Phase

Purpose: the goal of this phase is directly related to the first two principles of the sub-process: firstly, to engage teams or individuals in open discussions about specific mentality patterns; secondly, as a result of the setup phase teams commit to a set of principles according to which they will operate in practice. As stated earlier, such principles are directly derived from respective mentality patterns.

Before initiating the setup phase, the sub-process driver has to make a number of decisions on the overall "strategy" and concrete steps to be followed depending on the particular situation in which the sub-process will be applied. These decisions concern the following:

- *Scope (the "which patterns" question)*. Identify the specific mentality patterns to be discussed; the ones for which the commitment of the team should be attained. There are different motives that guide this decision, usually related to the specific goals of the driver and their level of acquaintance with the team. Examples of motives are:
  - To drive change in specific mentality patterns that, in the driver's view, the team[15] already exhibits –either in general terms or at a particular occasion, for example during the course of a project.
  - To avoid the emergence of patterns that exist in other teams within the organization.
  - To be proactive in preventing the manifestation of patterns that the driver judges as important to circumvent, for example in cases where the driver is not entirely familiar with a particular team (e.g., as a manager of a new project).

  A combination of the above motives is also possible[16]. In any case, the driver

[15] The use of the term "team" does not have a holistic sense but also implies particular individuals.

[16] A combination of motives has been the case in our experiences with the sub-process.

produces a concrete list of mentality patterns[17] to be put forward for discussion and commitment. In general, the number of patterns to be discussed is not important; it can even be only one. On the other hand, it should be expected that an extensive number of patterns may lead to practical difficulties[18]. Finally, the driver, after deciding the scope, prepares a documented version of the mentality patterns agreed upon using elements of the patterns' representation discussed in Chapter 4.

- *Timing (the "when" question)*. The next decision to be made concerns the point in time when the setup phase will take place. For example, in the case of a departmental team, the setup phase can take place at any occasion deemed appropriate by the driver. In the case of a project, it can take place at team-building sessions, e.g., at the start of the project, or at a meeting on project progress. Similarly to scope, the timing decision depends on the particular scenario and situation at hand. Moreover, it also depends on the level of confidence that the driver has on the receptiveness of the team to discuss the mentality matter.

Steps: following the decisions above, the sub-process driver initiates the setup phase whose basic steps are described below:

- The driver arranges a dedicated session with the team in order to discuss the mentality-related operating principles of the department, the team or the project. At the beginning of the session the driver sets as goal the agreement and establishment of the mentality-related operating principles. With this goal in mind, he presents and justifies the mentality patterns case[19]. An alternative to the

[17] In principle, such patterns can have either a negative or positive flavour. We have used only negative-flavour patterns in our practice.

[18] One to six patterns has been the case in our own practice. Moreover, the driver does not necessarily decide alone on the concrete list of patterns. Depending on the case at hand, we have consulted other colleagues in reaching such a decision.

[19] The way that the sub-process driver can present his argumentation and manage such a session is very specific to the particular setting and his own style. Therefore it cannot be prescribed by the sub-process.

dedicated-session arrangement is that the driver initiates and justifies the need to establish the operating principles in an occasion that he considers as appropriate for that purpose, e.g., during a team meeting. We refer to those as “occasions of opportunity”. An example of such an occasion is whenever problematic practices are perceived to be the direct consequence of certain mentalities. Such an alternative can be applied by colleagues that wish to assume the role of the sub-process driver, but are not sufficiently empowered by their role in a team. Also, it can be applied in order to check the level of receptiveness on the matter as explained in the introductory section of this chapter or in cases the driver considers the overall project “atmosphere” as inappropriate for a dedicated session, for instance due to time pressures. We have frequently adopted this alternative in our own practice with the sub-process.

- Following his argumentation on the need to establish mentality-related operating principles, the driver initiates the discussion by putting forward the list of mentality patterns in scope. In presenting these patterns particular emphasis is given to their concrete consequences. For each mentality pattern in the list, team members acknowledge whether they observed the pattern in their previous experience or whether it is applicable to the team itself.[20] Anecdotal stories or examples from the teams own experiences can also be discussed. The driver acts as the facilitator of the session, resolving conflicts and making sure that it is conducted in a “smooth” manner, particularly in avoiding personal allegations and finger pointing. Upon agreement on the list of mentality patterns and their characteristics, typically[21] captured by the respective representation language, the team discusses and agrees on its mentality-related operating principles, which we call the team’s Mentality Principles Manifesto.

[20] Such discussions are not about individuals “confessing” their own mentality patterns. This would require expertise in human psychology and, in practice, it would be hard to achieve. Instead, the objective is to give light to relevant observations and lessons learned so that experiences can be recalled and shared.

[21] The representation language is used as guidance. The goal is not to thoroughly document the pattern, but to reach consensus on the mentality-related operating principles.

- The manifesto is a statement of principles, directly derived from the above list of mentality patterns, according to which the team will operate. For instance, a manifesto principle, derived from the Fear-to-Admit-Ignorance mentality pattern, could state something like:

  "Whenever we are not confident that we have enough knowledge of a subject of any nature (technical, project related or other) in order to perform our work with the highest quality standards, we will ask whoever we feel necessary until we are confident. All project participants commit to answering the questions of fellows in the most comprehensive way possible".

As a general guideline, the manifesto principles should be simple, straightforward, understandable and non-ambiguous. They may also include some implications on the team's actions[22]. The manifesto itself should not be lengthy: "people can keep only a small amount in their heads" [14]. Moreover, at the driver's discretion, it can have several forms: verbal agreement, meeting minutes or a document[23].

Mentality Feedback Activity

Purpose: during this activity team members engage in feedback discussions and share views with respect to the mentality patterns and mentality principles already established in the manifesto. There are no specific steps involved since this is an activity of a continuous nature.

Scope: feedback should always be provided with respect to the mentality principles manifesto that everybody has agreed on. Any other principles or individual impressions of mentality patterns that have not been agreed a-priori should not be considered. If team members believe that new mentality patterns have been observed

[22] In order to facilitate the task, the sub-process driver can have the manifesto principles prepared in advance. However, the team should be given the opportunity to discuss and collectively agree on them. Such principles should not be "enforced".

[23] The manifesto form depends on the number of operating principles it contains. For example, in our own practice we have used "verbal agreement" for up to four principles.

then, on the initiative of the sub-process driver, the whole team should update the manifesto in the same way it was created during the setup phase.

Timing (the "when" question): the feedback activity should occur regularly, particularly whenever there are symptoms of mentality that could lead to problematic situations. However, it is not obligatory to arrange dedicated team sessions for that purpose — feedback can also be provided in occasions of opportunity.

In case of teams in a project setting, mentality feedback can take place between project phases, especially if there exist aspects or deliverables of the project that are not satisfying.

Approach (the "how" question): feedback should be provided according to the guidelines of the third principle of the sub-process as discussed in Section 5.3. It is not a means to criticize, put a stamp on, or condemn teammates. The sub-process driver is responsible for guaranteeing that feedback is provided according to these terms. Effective, constructive feedback permits revealing and reflecting on mentality aspects in a timely manner, thus leading to better practices such as more effective decision-making and action-planning. For instance, mentality-oriented feedback can help the team realize that narrow, non-validated assumptions are being made and, as a result, "trigger" the necessary corrective actions.

Finally, the sub-process driver is also responsible for dealing with specific cases of team members being "sensitive" to specific forms of feedback, communicate appropriately to other members and decide the most effective way of providing mentality-related feedback to them. For instance, in some cases feedback can be provided in an informal, face-to-face basis and not at a team meeting. In fact, acting as an overall facilitator, the sub-process driver is responsible for ensuring that the means chosen for providing feedback enforce the sense of identity and cohesion of the team and do not deteriorate it.

Mentality Learning Phase

Purpose: as its name indicates, this phase is directly related to the fourth principle of the sub-process. The objective is to facilitate collective reflection and learning on the practices and respective results in considering explicitly the mentality patterns matter. It is also an opportunity to praise positive attitudes, recognize issues, make comparisons to previous practices, inter-alia.

Scope: in the learning phase, typical questions that the team should discuss are:

- Are the mentality-related operating principles been followed? If not, why?
- Have any (and which) mentality patterns been observed? Was feedback given on those patterns? Was it effective? If not, why? What could have been done to make it more effective? Which are the problematic aspects of our work that could have their origins on those or other mentality patterns?
- Have any new mentality patterns been discovered? Which? Do we need to incorporate them in the initial list and the manifesto?
- How can the whole sub-process be improved in the future?

Timing (the "when" question): as in the feedback case, dedicated team sessions for this purpose are not mandatory and can take place in an occasion of opportunity with the initiative and facilitation of the sub-process driver. Moreover, it does not have to be an extensive or formal exercise: its agenda can be even decided in an ad-hoc way. The sub-process driver has to evaluate how the team perceives its necessity and value and decide accordingly. Actually, in the case of teams with invariant membership the learning phase can be combined with the feedback activity in a continuous course. In the case of a project team, it should preferably occur whenever the team is relaxed: that is, when there are no further project details to be taken care of and people have already got some rest after the project effort. This is because it is on relaxed conditions that people are more willing to contribute to the learning task, discuss in an open manner, accept points of view or even criticism from others and contribute new ideas.

5.5 Foundations and Rationale

In the previous section we have discussed the *Who, When, What and How* of the Mentality Innovation Sub-Process. In this section we focus on its foundations — the *Why*. The rationale behind the sub-process is in the realization that, in what concerns attitudes, beliefs and ways of thinking of persons or groups, one can distinguish two different scenarios:

- There are individuals and groups that, in some occasions, are conscious of what attitude or way of thinking will have a positive impact on both teamwork and individual tasks. However, for several reasons, they may choose not to follow that particular constructive attitude or way of thinking. For instance, many people may be aware of the fact that they fear to admit ignorance but often choose to continue operating and acting according to that state of mind.
- The second scenario is that people are not conscious of the particular attitudes or way of thinking they follow, and just assume they are thinking in the "right way" or doing the "right thing". "Experience Driven Optimism" and "Have the Right to Make Assumptions" are possible examples of this case.

Our objective is not to identify which mentality patterns belong to one or the other case, but to explain why and how the sub-process deals effectively with both.

The Theory of Cognitive Dissonance

As far as the first scenario is concerned, we believe that this phenomenon should be considered and can be explained by the widely accepted socio-psychological theory of Cognitive Dissonance [26] whose basic arguments can be outlined as follows.

Humans dislike inconsistency between two cognitions (knowledge, belief, attitude, way of thinking) or between a cognition and a behaviour. Such an inconsistency causes the arousal of an unpleasant psychological state, called cognitive dissonance. According to Cooper and Fazio [19], the arousal of such a psychological state depends on the degree of aversive consequences to us or those we like and the personal responsibility we take for attitude-discrepant behaviours. Personal responsibility consists of two factors: freedom of choice and the belief that potential negative consequences of the actions were foreseeable. Humans need to reduce such dissonance and there are four different available paths in order to do so:

(i) When two cognitions are in conflict, change one or the other cognition.

(ii) When two cognitions are in conflict, make one cognition more important or reduce the importance of the other, possibly by adding new cognitions.

(iii) When cognition is in conflict with behaviour, change behaviour.

(iv) When cognition is in conflict with behaviour, change cognition.

The following example, borrowed and adapted from [26], may assist in better understanding the theory and the paths above: A smoker who learns that smoking is bad for his health will experience cognitive dissonance because this piece of knowledge is dissonant with continuing to smoke. He is motivated to reduce the dissonance by:

- Changing one or the other cognition: "Smoking is bad for one's health" could become "I know lots of smokers that never had health problems", and "I am a smoker" could become "I do not really smoke that much".
- Looking for positive effects of smoking, for instance "smoking reduces tension and keeps one from gaining weight" (make one cognition more important), or "the risk to health from smoking is negligible compared with the danger of traffic accidents" (reducing the importance of the other cognition).
- Changing behaviour: that is, stop smoking.
- Changing cognition: The smoker could reduce dissonance by changing his cognition about the effect of smoking on health and believe that smoking does not have a harmful effect on health.

The choice of moment and path to reduce dissonance has been a matter of psychological debate [35]. As far as the choice of path is concerned, i.e. the nature of our motivation that underlies dissonance causes and effects, psychological studies conclude that we tend to choose the path of *least resistance*. The degree of such resistance is determined, among others, by factors such as threats to self-concept — the perception we have for ourselves, or self-presentation — our concern for the perception of others for us, possible rewards, the degree to which cognitions are consonant with many other cognitions, the extent of pain or loss that must be endured for changing a behaviour, and the satisfaction obtained from a behaviour.

Coming back to mentality patterns and the first scenario identified at the beginning of this section, we believe that this discussion on Cognitive Dissonance explains why people do not always follow constructive attitudes and ways of thinking even if they are conscious of them:

- Either they do not experience dissonance arousal, that is, based on Cooper and Fazio remarks above, either the inconsistency between attitudes, beliefs and behaviour does not have aversive consequences for them or those who they like, or they do not take personal responsibility for such inconsistency because they may feel that they do not have freedom of choice or because they think that potential negative consequences of the actions were not foreseeable;
- Or the constructive attitudes and ways of thinking that they are aware of have a degree of low resistance, making it easier to reduce dissonance by following paths (i), (ii) or (iv) (change cognition) than (iii) (change behaviour). In other words, they tend to change or diminish the importance of the constructive cognition. For instance, "Fear to Admit Ignorance is a problematic attitude" could become "Sometimes it is good not to admit ignorance". This is particularly true when such a change of cognition, altering or diminishing the importance of constructive mentality, does not impose a threat to self-concept, self-presentation or possible rewards. We are convinced that in the context of teamwork the self-presentation factor in particular has a very significant contribution to the degree of resistance of constructive mentality. Humans are social animals and the perception of others about them is of very high importance. In fact, Rogers [66] argues that the unconditional positive regard from others is necessary for the psychological health of people. Therefore, often when a choice between changing

a cognition or a behaviour must be made, we tend to go for the one that has less negative impact on the perception of others for us. In other words, the less the impact of a certain cognition on self-presentation, the less the resistance of such a cognition. In the context of mentality patterns, this implies that when there is no impact on self-presentation, it is more likely that one will change one's knowledge of a constructive mentality, instead of changing one's behaviour. Coming back to the smoker example, it comes to no surprise that many smokers do not quit smoking although they are conscious that it is an extremely unhealthy habit. Many anti-smoking campaigns follow a strategy that is based on promoting an image of smoking as being out-of-fashion, in other words stimulating the smokers' evaluations on how people around view them.

Guided by the socio-psychological theories and observations above, the Mentality Innovation Sub-Process is therefore designed to achieve the following goals. The Mentality Principles Setup phase and the associated manifesto stimulate the experiencing of dissonance arousal: people agree to what constitutes "constructive mentality" and commit to following it. This implies that they will take personal responsibility for possible inconsistencies between what has been agreed and how they operate. Moreover, the power of the possible argument that they do not have freedom of choice is diminished; they have participated on the principles setup phase and the mentality principles manifesto was not imposed to them. Finally, they cannot argue that potential negative consequences of actions were not foreseeable because the use of anecdotal stories in mentality patterns implicitly forces individuals to apply the story metaphors to their own situations, hence considering the gains or pitfalls of following a certain mentality.

The Mentality Innovation Sub-Process also attempts to *raise the resistance* of constructive mentality, thus making it more likely, when dissonance arouses, that a choice be made for changing behaviour instead of changing or diminishing the importance of constructive cognitions. This effect is achieved in two ways:

- People agree with and commit to the fact that specific mentality elements are an important success factor; this implies that the specific constructive attitudes and ways of thinking identified during the setup phase become part of their system of values and principles (the Manifesto), a fact that significantly raises the resistance of such constructive attitudes and ways of thinking.

- People become aware that not following the agreed constructive mentality would be a threat for their self-presentation: the team will constantly observe whether constructive mentality is followed and continuous feedback on this matter will be provided.

Self-Perception Theory

So far we have related the rationale of the Mentality Innovation Sub-Process to the first scenario identified at the beginning of this section and we have based our discussion on the Cognitive Dissonance theory. It should be noted that the Cognitive Dissonance theory is directly related to attitude and cognition change and its applicability typically implies pre-existing established attitudes and cognitions.

As far as the second case is concerned, in which people are not conscious of particular attitudes and ways of thinking, we need to consider another psychological paradigm — Bem's Self-Perception theory [10].

According to Bem, "*Individuals come to know their own attitudes, emotions and internal states by inferring them from observations of their own behaviour and circumstances in which they occur. When internal cues are weak, ambiguous, or uninterpretable, the individual is in the same position as the outside observer...*".

In other words, according to the Self-Perception theory, we develop our attitudes by observing our own behaviour and concluding what attitudes must have caused them. For instance, "if I often eat Indian food, I like Indian food". People use inferential processes to determine the attitudinal significance of their actions. It should be noted that, for many years, this view seemed to challenge Cognitive Dissonance in the sense that according to Self-Perception we do not necessarily change our attitudes in response to our behaviour. However, more recent studies, for instance by Fazio et al [25], concluded that both theories are right — it all depends on the circumstances:

inferential processes postulated by self-perception theory are especially likely to influence attitudes that are not pre-existing and well-established, or when the discrepancy between attitude-behaviour is fairly small. At the same time, there is substantial evidence that larger attitude-discrepant actions do produce effects described in the Cognitive Dissonance theory. Therefore, the use of Cognitive Dissonance for the first scenario — people conscious of constructive mentality but do not follow it — is justified and supported; and so is the use of Self-Perception theory for the second scenario — people with no pre-existing notion of attitude or way of thinking.

We still have to discuss how the Mentality Innovation Sub-Process, guided by the Self-Perception Theory, deals with the second scenario. In our view, when people are not conscious that they follow a particular attitude or way of thinking, we need to devise ways of stimulating and guiding people explicitly when applying Bem's inferential processes to determine the attitudinal significance of their actions. In other words, with respect to mentality patterns, we need to explicitly guide the individuals' natural tendency for observation of behaviour and respective inference of internal state. The objective is to explicitly guide people in order to become conscious of a particular mentality, and subsequently change it if it leads to problematic practice or continue following it if proven to be constructive. For instance, if people operate according to the "Have the Right to Make Assumptions" mentality, we need to explicitly trigger the inference mechanism so that they can become aware of the mentality of making assumptions without validating them. In our view, there are mainly two ways in which this can be achieved.

The first consists in increasing the levels of what we call "self-enlightenment": stimulating people to look inside themselves, judge their own past and present actions, the underlying attitude and way of thinking-related reasons for those actions, and how those contribute to their personal development and work results. The Setup Phase aims to explicitly serve this goal.

The second way is stimulating interpersonal communication: only when people around communicate with and alert individuals (and teams) on specific attitudes and ways of thinking can they become aware of possible problematic or constructive attitudes and adjust accordingly. It should be noted that interpersonal communication is by itself a way of stimulating self-enlightenment. However, mere communication exhibits a problem: often people are not willing to hear the comments and opinions of others. The main cause is that people tend to judge comments from others as being their particular view and, hence, tend to just ignore them. This is particularly true for ad-hoc comments that come from fellows that people either do not know well or on whom they do not have a positive opinion. Only when there is an agreed-by-all basis on the established aspects upon which communication will be performed and feedback will be given can people be willing to accept more and think about the views of others. The purpose of the mentality principles manifesto is, precisely, to act as such a team-agreed, mentality-related communication and feedback basis.

Similarly, the Feedback Activity and Learning Phase are also about communication and self-enlightenment: stimulating individuals and teams to consider their attitudes and ways of thinking, become aware of the current negative and positive cases, improve their project-related practices, and use those experiences in the future.

5.6 Relevance to Software Engineering

The Mentality Innovation Sub-process is also based on and guided by proven concepts, methods and sound observations found either in the specific context of software process methodologies or in software engineering and other human activities in general. In what follows we outline some of the evidence that justifies the relevance of our proposal.

In what concerns software project methods, in [14] the author argues that a properly performed software team-building phase and relevant exercises are very advantageous for achieving team morale and effective communication. The PMBOK Guide [65] also refers to team-building activities as a crucial component of effective project management. Sharing this view, we have "injected" the Mentality Principles Setup as a team-building practice, but in a narrower context and with different objectives in mind: to use the notion of mentality pattern in order to trigger a collective culture, establishment of effective communication practices, and commitment in considering the mentality element as a first-class concern.

As far as the Mentality Principles and Manifesto are concerned, one should observe that the existence of principles is found implicitly or explicitly in all software process methods and the notion of a manifesto is also found in agile methodologies. Those principles form the basis upon which each process is explained and organized, provide the rationale for prescribing certain practices and ruling out others, and guide the way each process should be executed. Therefore, projects adopting specific process methods implicitly adopt, at least to a good extent, the principles defined by those methods. Being only complementary to and an independent component of software process methods, the Mentality Innovation Sub-Process adopts a team-created and mentality-oriented notion of principles that is more appropriate for the objectives it is designed to meet.

The Mentality Feedback activity and the Mentality Learning Phase are also not entirely novel: the importance of feedback and learning in applying software process methods has also been independently recognized in different works. In [48] the authors identify those two factors as important in the context of the Rational Unified Process. Feedback, learning and effective communication (called collaboration) are also important components of agile methodologies. However, in those methods, the importance of feedback is considered more in the sense of feedback given by users-clients after deliverable portions of software. Moreover, the feedback, learning and communication aspects of those methods do not target explicitly and in a systematic way the mentality-related innovation as presented in our work.

Chapter 6

RESULTS

6.1 Overview

This chapter is split into four parts:

- The first part, Section 6.2, presents the results and lessons-learned by applying in practice the Mentality Pattern primitive and the Mentality Innovation Sub-Process. As explained in Chapter 3, the results we report are not merely our own impressions and observations, but have also been established through relevant engagements with peers.
- Section 6.3 presents what a project management practitioner has reported in his experience with the primitive and related practices.
- Section 6.4 presents the results of a survey which we conducted on the matter with a number of software practitioners.
- The chapter concludes with a summary of the results[24].

6.2 Results in our own practice

6.2.1 On the Mentality Pattern Primitive

Before discussing the results of using the mentality pattern primitive in practice, we should first re-emphasize that we are not looking to describe and reason on particular mentalities in a comprehensive way, but to provide the means for individuals to capture and systematize the different perceptions that they may have on the character and attributes of mentalities. Therefore, in considering concrete results, discussing properties such as completeness and accuracy of the mentality pattern primitive is

[24] In order to protect their privacy, we do not disclose the names of the colleagues who have participated in any part of this research.

meaningless. Perceptions, by default, are not necessarily widely-accepted facts; they imply subjectivity and uncertainty. In other words, we are not examining whether the definition and representation of the primitive is comprehensive in describing the human "mentality" phenomenon.

Instead, we report on the capacity of the primitive to organize the different perceptions on the matter, facilitate agreement on the identification of concrete recurring mentalities and its effectiveness as a common communication mechanism. Moreover, our aim has not been to explicitly examine cognitive aspects such as learnability or user-satisfaction in relation to the representation language. This is an interesting topic for further research but has an individual-oriented focus. For instance, the ability for each individual to learn and remember the elements of the mentality pattern representation does not necessarily mean that the primitive allows teams to achieve agreement on the matter.

Having said this, our results have been very encouraging in what concerns the aforementioned dimensions. In particular:

1. The notion of mentality pattern provides an explicit abstraction that establishes a "focal point" and common basis for organizing relevant discussions among colleagues. Our practical engagements indicate that colleagues generally agree on the existence of certain human-thinking elements that affect SE practices. However, those human elements are generally indistinguishable. First of all, a number of colleagues do not even characterize such elements — they just perceive their existence. Others use explicitly the term "mentality", while others talk about attitudes and behaviours. Finally, there exist colleagues that relate such elements to individual personalities and organizational or team culture. As an example, we refer to the case of the "Better is the Enemy of Good" mentality pattern. Some colleagues regarded this as "conservative behaviour", others called it "culture" and some more as a sign of "a conservative person". Even worse, colleagues have not always been consistent in their own characterizations of such human related elements. While in some cases they refer to "behaviour", in other cases they use the term "culture". However, this lack of common terminology and definition is

not merely "theoretical" but also has practical consequences: the different ways people perceived and characterized such human elements had implications in their predisposition to discuss them, share relevant experiences and consider them in their practices. In many occasions, I kept hearing phrases such as: *"that's the way people are in here – what can we do?"*. We have found that the mentality pattern primitive is a good response to this lack of a common vocabulary for organizing relevant discussions and sharing insights on the matter. It may not respond to the question, and we did not aim to, of what the "absolute truth" actually is, i.e. whether a given human element is in fact a mentality, behaviour, culture or personality characteristic. However, it provides a common notion and respective focal point that colleagues understand and recognize. On the other hand, the existence of a common abstraction and representation does not always imply agreement on "content", i.e., on which human-thinking elements should be characterized as mentality patterns. We discuss relevant findings below.

2. There are three scenarios in what concerns the ability of the primitive to facilitate agreement on identifying concrete recurring mentalities and capture them as corresponding mentality patterns.

   - The first scenario concerns mentalities that have been straightforward to agree on. By straightforward we mean that colleagues immediately agreed that such recurring turns of mind deserved to be captured as mentality patterns. Characteristic examples of this case are patterns such as "Fear-To-Admit-Ignorance", "Experience-Driven-Optimism", "It-Works (but I do not know why)", "The Best is the One I (We) are Comfortable With", "Have the Right To Make Assumptions" and "Opportunistic Listening".

   - The second scenario, which was implied above, concerns recurring turns of mind that were perceived by colleagues to be related to behaviour, the character of particular individuals and organizational, or team culture. Typical examples of this case are patterns such as "Not Invented Here", "Legacy Person Mentality", "Subject Guru", "Better is the Enemy of Good", "It's not my fault", "Negativism" and "Secretivism". The particular difficulty encountered with behaviour (and the choices made in relation to behaviour) were discussed in detail in Section 4.5. In general, the primitive has been successful in facilitating agreement in such cases. As discussed above, this stems from the power of the primitive to act as a focal point and to establish a common vocabulary. In other words, given the difficulty to agree on the exact character of such manifestations, capturing them as mentality patterns provided a "point for consensus" between the views of different colleagues. Moreover, the "Possible Causes" element of the pattern representation allowed for the relationships to the character of particular individuals and to cultural elements to be explicitly captured.

- The third scenario has been the most difficult one. It concerns "bad practices" or recurring turns of mind "towards the context" as explained in Section 4.6. In particular, we have detected a tendency to attribute the causes of particular problems to a mentality-related issue and subsequently referring to them as distinct mentality patterns, for example, statements such as "poor communication", "metrics abuse", "analysis-paralysis" or "not sufficient automated source-code control". Consequently, in order to deal with the matter, and given that we aimed for patterns identified to be applicable and shareable independently of specific projects and teams, we had to extend the definition of the primitive with the guiding principles for characterizing mentality patterns discussed in Section 4.6. The effects of this extension were twofold. On the one hand, we were able to establish another common basis on which to distinguish between, on the one hand, recurring, wide-applicable mentality elements and bad practices and, on the other hand, ways of thinking that reflect personal opinions or preferences. This distinction has been effective in achieving consensus in some occasions such as the "Technology Fundamentalism" and "No Coding = Useless" examples. However, it has not been successful in all scenarios, e.g., on deciding whether "analysis - paralysis" is also a mentality pattern. The good news is that, even in occasions in which no consensus was reached during the application of the mentality innovation sub-process, we were still able to consider some of such controversial cases as mentality patterns and set the corresponding mentality-related operating principles. On the other hand, this extension had a penalty: it resulted in a more complex definition of the primitive that requires more effort to explain when engaging in relevant discussions.

3. The third aspect to report on concerns the effectiveness of the primitive as a communication mechanism between individuals. In this respect, we have repeatedly found that the name of a pattern alone was sufficient for colleagues to immediately recognize the particular turns of mind implied and actively engage in relevant discussions. This has been apparent for all the patterns that were identified in the first two of the scenarios above. However, in the third scenario, the name of the pattern would not always suffice. In other words, there is a direct relationship between the level of agreement in characterizing a mentality pattern and its subsequent capacity to act as a communication mechanism. Moreover, the particular choice of name, e.g., in terms of precision, and the degree of "exposure" of colleagues to particular patterns were factors that influenced that capacity. For example, the name "Not Invented Here", which is well known in the literature, is much more recognizable than "I will do it my own way", which implies the same pattern. Finally, colleagues often devised their own terms to refer to given patterns or minor variations of them. The purpose of the "Other Names" element of the mentality pattern representation is, precisely, to satisfy the creativity of individuals.

4. An additional result of our work is what we call "*mentality patterns interference*". That is, people and their associated mentality patterns influence one another, either negatively or positively. For instance, in the former case, the "Negativism" of one person tends to amplify the "Secretivism" of others whereas, in the latter case, people who are willing to admit ignorance affect in a positive way their peers who do not. We believe that this observation is of high importance for guiding decisions on the way people should be distributed in teams or engage in joint tasks, for example, in pair programming. Moreover, even when certain assignments of people to teams are inevitable due to technical, team size or other constraints, being aware

of such forms of interference is essential for more effective team management. Therefore, the “Related Patterns” section of the mentality pattern representation is explicitly designed to capture such forms of interference so that the experiences accumulated within and across projects and organizations can be effectively communicated and shared. Given that our goal has not been to examine individual mentality patterns, we do not report on a proven catalogue of interferences between the patterns that we identified. However, the examples that we have provided clearly illustrate the matter. We believe that this has been a direct consequence and result of our work that is worth exploring in the future along the dimensions discussed in Chapter 9.

6.2.2 On the Mentality Innovation Sub-process

In what concerns the Mentality Innovation Sub-Process we report relevant findings in two dimensions: its applicability in practice and its capacity to drive what we call “observable change” with respect to concrete mentality patterns.

Applicability

The set of factors that we have found as having the most decisive impact on the applicability of the sub-process are the following: the level of receptiveness of a given team to discuss the mentality matter and engage in respective practices; the choice of timing that the sub-process activities are to be performed in relation to existing constraints such as progress in tasks and the overall team atmosphere; the specific role that the sub-process driver has within the team; and the way the driver leads the sub-process execution particularly in what concerns its feedback aspect. For instance, in one occasion, we faced significant resistance in discussing the mentality matter when we underestimated the pressure that some team members had in meeting a given deadline. As a project manager, it has been considerably easier to set up dedicated sub-process sessions or discuss the matter in occasions of opportunity. Finally, in occasions where feedback turned very personal and I did not prevent it, the result has contradicted my intentions. Consequently, our first finding is that the role, skill-set and determination of the driver are a critical success factor for the applicability of the sub-process.

In what concerns the receptiveness of colleagues in engaging with the sub-process and discussing mentality patterns, our experience has been encouraging: irrespective of the particular forms of conceptualization and characterization discussed in the previous section, the vast majority of colleagues are very conscious of the existence of such recurring turns of mind and their consequences in SE settings. Also, many of them argue that those are not specific to SE but can be found in any other setting. The interesting conclusion in our practice with the sub-process is that software practitioners are not only aware of the phenomenon but most of them are also willing to discuss it openly, share views and, in many cases, even reflect on their own practices. In other words, there are strong indications that we are reaching a level of maturity that allows SE teams to consider the matter explicitly in their practices, irrespectively of the specific means that they may use to do so.

We have also found that the sub-process does not impose any significant overhead in teams' established practices in terms of additional tasks to be performed or time and effort to be allocated to it. In its simplest form, it only requires the identification of what we called "occasions of opportunity". Such occasions can be frequently encountered during a project or while being a member of a team.

Another aspect that is worth discussing is the impact of team size on the applicability of the sub-process. We have found that applying the sub-process in a three to five members team is considerably easier than applying it with a larger team of, say, ten members. The obvious reason is that smaller teams are easier to coordinate, the various points of view of team members are easier to align, and constructive feedback schemes are easier to establish. We also have to admit that the sub-process has not been applied in teams larger than 15. One could argue that in this case the sub-process could be applied to smaller sub-teams. However, how such split should be made and whether there is any kind of interference among sub-teams that may result as a consequence, is a matter that requires further research.

Driving "observable change"

In what follows, we report on the ability of the sub-process to drive "observable change" with respect to specific mentality patterns. By observable change we mean that, following the application of the sub-process, particular behaviours of (different) colleagues, which were perceived by (some of) their peers as manifestations of a corresponding "mentality pattern", appear to have changed to a point that the same peers were able to observe and confirm[25]. The concrete effects obtained through the application of the sub-process in its various forms and with different teams are the following:

- There exist mentality patterns for which observable change has been established in all occasions[26], i.e., in *all* occasions that the sub-process has been applied and such patterns have been identified, *all* peers in question have observed change in the manifestations of those patterns. The patterns concerned are "Fear to Admit Ignorance" and "Have the Right to Make Assumptions". In other words, after applying the process, particular colleagues appeared to be more willing to admit their ignorance on a matter and were more careful in making and stating assumptions.
- Observable change has *not* been established in relation to the following mentality patterns: "Subject Guru", "Not Invented Here" and "Secretivism". In other words, for these patterns and in *all* occasions that the sub-process has been applied and such patterns have been identified, *all* peers in question have not observed any notable change in the behaviours and actions of colleagues perceived to exhibit such patterns.
- For patterns such as "Better is the Enemy of Good", "The Best is the One I am Comfortable With" and "Technology Fundamentalism", the result was unclear in

[25] "Observable change" refers to the manifestations of a given pattern, but does not necessarily imply a corresponding change in the established perceptions of peers. Moreover, in all occasions we have asked at least two peers to express their views.

[26] A different occasion implies different teams in different contexts.

the sense that there is no convergence in what the peers in question have observed. In other words, in *all* occasions that the sub-process was applied and such patterns have been identified, there were peers who observed subsequent change and others who did not.

- There exist patterns for which, in *some* occasions, *all* peers agreed on "change" or "no change" while, in *other* occasions, there was no agreement on way or the other. In the affirmative scenario, the patterns concerned are "Us and Them", "It Is Not My Fault" and "Negativism". In the opposite ("no change") scenario, these were "Opportunistic Listening" and "No Coding = Useless".
- In what concerns the rest of the mentality patterns identified in this thesis, namely "Experience Driven Optimism", "Legacy Person Mentality" and "It Works (but I do not know why)", there were no relevant effects that can be reported. Although colleagues agreed on their existence in general, there has been no consensus on any related manifestations within the teams at hand.

We should point out that the above effects should be interpreted with caution and are not suitable for statistical analysis[27]. There are two reasons that justify this statement as well as the deliberate use of the terms "drive" ("observable change") and "effect", instead of "achieve" and "result". The first reason is non-homogeneity. Given our specific goals and the characteristics and constraints of the research environment discussed in Chapter 3, the findings above are not based on controlled-variables approaches, i.e., they do not result from controlled experiments. In other words, during the course of our work, the sub-process has been applied in several ways and forms, with variable teams, in different organizational environments and involving different peers. Furthermore, the way such peers reported to us their observations has also varied: in some cases we used informal discussions, whereas in other cases we applied more structured ways. Moreover, the degree of the perceived manifestation of a given pattern and the level of confidence of peers in confirming subsequent

[27] For this reason, on purpose, we do not provide numerical details e.g. on number of occasions and peers for the "effects" above.

"observable change" varied from case to case. Finally, we have to consider the possible influences of peers in expressing their views, for example, due to the type of personal relationship a peer may have with an individual perceived to manifest a given pattern, or to pre-established opinions of colleagues with respect to other colleagues, or because of the particular timing I have engaged in such discussions, or finally because of the personal relationship that such colleagues had with me. In all occasions, I have tried to circumvent such influences in my practice, e.g., by choosing peers I thought to be more objective in expressing their views and by engaging in such discussions at times I considered as most appropriate for that purpose. However, it is unlikely that such influences were entirely avoided.

Secondly, the size of the sample is not large enough to allow for statistical analysis. In that respect, it should be taken into account that the mentality patterns reported herein were identified at different times during the course of our research. For instance, we recognized some of the patterns quite early on: "Fear to Admit Ignorance", "Subject Guru", "Having the right to make Assumptions", "Us and Them" and "Not Invented Here". However, some others, e.g., "Technology Fundamentalism" and "Experience Driven Optimism", became apparent only at later stages.

Having said this, we should remind that, in accordance with our specific goals, the sub-process has emerged as a result of our research; it was not conceived "a-priori" as a method to be tested in a statistically sound way. In other words, *we have aimed to develop means that can offer themselves as useful in taking explicit account of mentality patterns in real-life settings*. The research approaches that we followed were based on that objective and, therefore, justify the absence of results based on controlled-variables or on statistically relevant methods. Moreover, the results presented above indicate that, in several cases, our goal of making mentality patterns a useful and applicable mechanism has been met. Naturally, having defined the innovation sub-process as a result of our work, a next step could be to perform statistically relevant studies along the aforementioned dimensions. We outline relevant recommendations in Chapter 9.

Accordingly, capitalizing on the effects reported above and taking into account the previous discussion, it is more appropriate to formulate the results achieved through the application of the sub-process in a more generic way:

1. There is evidence that, for some mentality patterns, the sub-process facilitates a constructive change in the way individuals operate in a team setting. This positive change is often observable by their peers.
2. There exist recurring mentalities that are more persistent. In such cases, the sub-process appears to have smaller or no effect in the way people operate.

A final remark that is interesting to make relates to the ability of the mentality pattern primitive to facilitate agreement on the identification of concrete recurring mentalities, which was discussed in section 6.2.1. It appears that some of the mentality patterns for which observable change has not been established through the mentality innovation sub-process, e.g., "Subject Guru" and "Not Invented Here", are also perceived by colleagues to be related to the character of particular individuals and organizational or team culture. In other words, one could argue that the more a pattern is perceived to be related to such factors, the more persistent it is. We believe that this matter also deserves further investigation.

6.3 Results of an assessment by a Project Management practitioner

In order to further assess the results discussed in the previous section, we have asked an experienced project manager to apply the mentality pattern primitive and the innovation sub-process. We did not directly participate or act as an observer on any of his engagements, apart from providing the necessary initial guidance and insights on practices to be followed and on outcomes to look for. The report on his views and findings is presented below:

"It was around October 2010 that Georgios presented to me the mentality pattern idea and associated practices. Right from the beginning I found this to be an extremely interesting and pioneering concept. As IT professional and particularly project manager for many years, I have frequently encountered such mentality-

related issues in many occasions, in my collaboration with various colleagues in different teams and tasks. However, this had been the first time I came across some sort of a "model" that could "give a name" and systematize this human mentality matter.

We had several discussions with Georgios on issues such as the existence of such "patterns", on which human thoughts and behaviours could be characterized as such, on the factors that influence them, on what is their impact and consequences in projects and teamwork in general. I have to admit that reaching conclusions on such questions is not easy and I have often found such conclusions to be subjective. After all, to my view, the human factor and its impact at work is not linear or predictable. On the other hand, there is no doubt that being able to explicitly characterize "mentalities", discuss about them and share experiences is a step forward to the alternative of "doing nothing" or consider them in ad-hoc ways. It was mainly for this reason that I decided to apply in my work practice some of the ideas that Georgios had also applied — what he calls the mentality innovation sub-process — and observe the results.

After discussing with Georgios on the matter, I decided to apply the main principles of this "sub-process": talk openly about patterns, engage colleagues in such discussions, feedback and reflect on lessons learned. I did not wish to follow a very "prescriptive" approach for two reasons: firstly, due to the fact that I am not very keen on such approaches and secondly because of my lack of experience in the specific matter. Nevertheless, after several months of "experimenting" in practice with "mentality patterns", in summary, I report the following:

- *The notion of mentality pattern has been a quite powerful communication tool when engaging in discussions with various colleagues on this matter. People can immediately recognize each particular mentality case, often only by the name of the pattern, and refer to examples, variations and related stories for each pattern. I believe that some colleagues were often actually identifying themselves in such stories, though there is a genuine difficulty for them to explicitly* *admitting so. In general, those "mentalities" are aspects that we all know about and having a means of "modeling" them is extremely useful.*

- *It is often difficult to distinguish between "mentalities" and just bad practices during work. However, I would say that this is important only if one wishes to be "purist" in precisely identifying mentality patterns. The opportunity to explicitly discuss and even disagree on such elements, no matter if we call them mentalities or bad practice, triggers very useful discussions and reflection on current work practices.*
- *Some patterns, e.g. "FearToAdmitIgnorance", "Us Vs Them", "Technology Fundamentalism" are extremely interesting on their own right. I can definitely argue that considering those explicitly helps colleagues to reflect on their own thoughts and behaviour and adjust accordingly in a positive way. However, I have doubts on whether this would be "enough" for the subsequent work scenarios that such colleagues will participate in future tasks. Maybe some of those patterns need special attention from organizations and/or team leaders. This needs to somehow be verified and proven.*
- *I also believe that the mentality patterns associated practices would often need to be adapted to take into account the specific work context, teams, personalities or even tasks at hand. They provide useful principles and guidelines but in practice it can be difficult to be applied in a "by the book" way. For instance, some colleagues are quite "difficult" in accepting feedback, particularly in this delicate matter. In other cases the "timing" or the team composition does not seem appropriate to engage in such discussions. I guess that adapting and improving the way such practices are applied needs further "experimentation".*

In general, I found Mentality Patterns to be a very important, interesting and useful concept and I would definitely continue to consider them in my work practices. There are some elements of how to deal with such patterns that need to be further worked out in practice and be more extensively applied so that the lessons learned can help improve our methods on the matter. It is not an easy task given the complexity and uniqueness of the human factor, but it is worth trying."

In a series of follow-up sessions with the colleague, we were able to establish a good degree of convergence on the findings presented in the previous section. Such convergence is also noticeable in the above report. In summary:

- The colleague used occasions of opportunity in order to engage in relevant sub-process practices.
- He identified the level of receptiveness and the skill of the sub-process driver as the important factors for the sub-process applicability.
- We share a similar view in what concerns the capacity of the mentality pattern primitive to act as a focal point in engaging in relevant discussions, its effectiveness as a communication mechanism and the difficulty to distinguish between mentalities and bad-practices.
- There is convergence in what concerns the ability of the sub-process to drive a positive change in certain mentality patterns — we have both identified the "Fear to Admit Ignorance" case — as well as the more persistent nature of other patterns.

6.4 Survey on Colleagues' Views on Mentality Patterns & Practices

In this section we report on a survey we have conducted with the objective to directly explore the opinion of practitioners on the mentality matter in SE. Our goal was not only to refer to specific elements of our research, but also to look into the views of colleagues in a broader sense.

The survey was conducted using an interview approach in May-June 2011. The choice of the interviewing approach over other survey methods, such as questionnaires, was justified by the fact that it allows the researcher to explore in greater depth the topic addressed and the views of participants. This attribute has been vital given the aforementioned objective of the survey. The way that the interviews have been conducted was based on the guidelines and practices prescribed in the relevant literature, as for instance [45], [63].

In what follows, we provide the necessary background information on the survey and present the results obtained. Those should be considered as provisional until a further larger-scale assessment is performed to support them. Still, they are very useful for three reasons: firstly, they present the insights of individuals who have been active in software development for several years and in a variety of organizations; secondly, because they relate and provide further evidence on some of the findings presented in the previous sections; finally, because they provide insights on subsequent work on the matter.

Preparing and Conducting the Survey

The "sample"

Given that we did not aim to provide statistically meaningful conclusions, a different kind of sampling approach was used for the survey. The participants were chosen from our work environment (convenience sampling) based on three eligibility criteria that we deemed appropriate for this study (purposive sampling):

- To have experience in both team-leading and technical roles
- To have more than five years of overall experience in the software field
- To have had no previous contact with the mentality pattern primitive and the sub-process

The participants were contacted and the objective and topic of the study was explained. Following their consent to participate, we provided all participants with the same background information. More specifically, such information concerned the details of the notion of mentality pattern, its definition and representation, a list of patterns (Chapter 4), and the description of the mentality innovation sub-process (Chapter5). Finally, the interview was scheduled at a time convenient for each participant. Out of the fifteen candidates initially contacted, we managed to conduct interviews with twelve of them. On average, the participants had fourteen years of professional experience in three or four different organizations.

The interview setting

In what concerns the setting in which the interviews were conducted, we followed the guidelines provided by the relevant literature [45]. In summary, interviews took place in a quiet setting in which privacy was assured. As interviewers, we have tried to build rapport and adopted an attitude of being a good-listener, open, honest, flexible and non-judgmental.

The interview approach

The survey was conducted using the standardized open-ended interview approach [63]: *"the exact wording and sequence of questions are determined in advance. All interviewees are asked the same basic questions in the same order. Questions are worded in a completely open-ended format"*. The specific set of questions asked as well as all the responses of the interviewees can be found in Appendix A.

Comparing with other known interviewing approaches, such as "informal conversational", "interview guide" and "closed, fixed-response interview", the main strengths of standardized open-ended interviews are that *"respondents answer the same questions, thus increasing comparability of responses; data are complete for each person on the topics addressed"* [63]. On the other hand, this approach has the potential weakness of *"less flexibility in relating the interview to particular individuals and circumstances; and standardized wording of questions may constrain and limit naturalness and relevance of questions and answers"* [63]. In order to circumvent this possibility, we systematically used a widely known interviewing practice known as "probing": during the interviews we were closely following the interviewees' verbal and non-verbal communication and encouraged them to give more information; we asked for further clarifications from the interviewees and we were also clarifying the questions in cases we detected that they could have been misunderstood; we were also repeating, in our own words, the opinions and ideas of the interviewees in the form of a reflective summary; finally, the answers were documented on written notes[28]. In all cases, my interventions were neutral in order to avoid biasing the responses.

The questions.

The set of questions belong to "opinion and values" category [63]. They ask about the respondents' opinion and judgment on the issue at hand. In particular, the questions were split into four themes. The first two themes comprise:

- Questions concerning the practitioners' opinion on the existence of mentality patterns, their impact on software practice and the identification of concrete examples (out of the list presented in Chapter 4) that had been encountered frequently or found to have a decisive impact in practice.
- Questions on whether they currently employ any specific means in order to take into account mentality patterns.

These questions allowed us to explore the matter in a wider sense. For example, we deliberately included the questions on the existence of mentality patterns or on concrete examples of them, although such aspects have not been direct research questions of this thesis. On the other hand, in further exploring colleagues' views during the interview, we were able to relate some of the answers to specific parts of our work, for example, in what concerns their opinion on the mentality pattern representation explained in Chapter 4.

The remaining two themes consist of questions that are directly related to the results of this thesis. In particular:

- We asked participants to share their opinions on the potential applicability and value of the mentality innovation sub-process.
- We concluded the interviews by asking whether the participants would consider applying the mentality pattern primitive or the sub-process in their work

[28] We deliberately have not used audio recording in order to avoid any effect due to its presence, given that the majority of the participants had no previous experience of being interviewed as part of a research project.

environments. We also provided them the opportunity to comment on any aspect we did not explicitly ask.

Results and Analysis

Following the interviews we analyzed the data gathered using data analysis methods such as coding and identification of themes and patterns [59]. The results of this analysis, per survey question, are presented below:

- *Can you comment on the existence of Mentality Patterns in practice? (e.g. do they exist only occasionally, frequently, a lot?)*

  Colleagues have unanimously stated that mentality patterns exist in SE practice, with the vast majority of them indicating that they have found them a lot in their experiences over the years — two of the colleagues have essentially characterized the existence of such patterns as "endemic" or "prevalent" in the IT industry.

- *Which of the patterns identified are the ones which you have encountered more frequently?*

  The mentality patterns that colleagues consider to be the most frequent are "Fear to Admit Ignorance" and "Not Invented Here". Also, the "Subject Guru", "The Best is the One I am Comfortable With" and "Us and Them" patterns were found to be frequently encountered.

  An interesting remark is that we have also identified quite early the "Fear to Admit Ignorance", "Us and Them", "Subject Guru" and "Not Invented Here" patterns. Moreover, in his report, the project management practitioner refers explicitly to the first two. In other words, it appears that there is convergence in identifying specific patterns that are frequently encountered in practice.

- *Which of the patterns identified are the ones you believe that have the most decisive impact in practice?*

  The responses to this question have been very similar to the above: "Fear to Admit Ignorance" and "Not Invented Here" are considered to be the patterns that have the most decisive impact in practice. Moreover, “The Best is the One I am Comfortable with” and “Us and Them” have also been indicated by several colleagues as having decisive impact.

In general, with a few exceptions (e.g. “Subject Guru”), in the majority of cases individual colleagues provided the same response to the second and third questions of the survey. That is, although in all cases I made clear during the interview that the two questions could have different answers, colleagues consider the most frequent patterns to also have the most decisive impact. One can assume that this is merely a matter of perception: if a pattern is encountered frequently, then its impact has been repeatedly recognized and, therefore, it is perceived to have the more decisive impact. This potential association appears to be an interesting topic to be taken into account by statistically-oriented research on the matter, a case that we put forward in Chapter 9.

- *Can you provide additional examples/cases of mentality patterns you have encountered?*

  Answers to this question have been twofold: on the one hand, there are colleagues who did not provide any additional examples of mentality patterns. A simplistic interpretation would be that the list of identified patterns is comprehensive. In fact, some colleagues made statements to that effect. My own view as interviewer is that these colleagues could not think of any particular pattern during the interview. On the other hand, other colleagues provided examples of such patterns. These fall into two categories. The first, explicitly admitted by some interviewees, refers to patterns that are variations of the ones presented to them. The second category concerns some patterns that appear to have the same character as the ones we have provided. For instance, “Fear to Delegate”, “Fear to Decide” and “Fear to Become Unpleasant” are examples of patterns that have a similar character to “Fear to Admit Ignorance” — with a focus on individuals in team leading or managerial positions. We do not claim that these are essentially mentality patterns. However, the fact that these turns of mind were mentioned, together with subsequent positive responses on the mentality pattern primitive, is very encouraging in what concerns the ability of the primitive to provide a common focal point in order to explicitly capture such turns of mind. These findings are also consistent with what we presented in Section 6.2.1: although colleagues perceive the existence of certain recurringmentality elements, they do not have a means to characterize them in an explicit way.

A second important finding from responses to this question, and from some related comments in subsequent questions, is that many colleagues agree with(and give strength to)the mentality-pattern interference phenomenon. For instance, in one case, a colleague considered that a pattern he proposed — "Fear to Delegate" — can be the effect of the "Subject Guru" one.

- *Do you think that the Mentality Patterns issue influences S/W practice and, if so, in which respect and to what degree?*

All colleagues have responded that mentality patterns impact software practice. In most cases such impact concerns the collaboration and atmosphere in teams, wrong planning, delays and quality of results. Some colleagues argue that the impact also depends on the specific context, e.g. the presence of a competent project manager or on organizational factors. We believe that this unanimous response on the significant impact of mentality patterns further amplifies the relevance of our work.

- *Do you take into account the mentality patterns issue in your current work practice, even if not explicitly? (e.g. on team management, on individual coaching or on tasks assignment)*
  - *a) If yes, can you elaborate on the means that you use?*
  - *b) If no, do you think that it would be helpful to do so?*

This question revealed an interesting sequence of actions that most colleagues currently follow in order to take mentality patterns into account in their practices. At first, usually unconsciously, they are aware of their existence or attempt to recognize their presence and associate them with specific peers.

Then, they devise their own means to deal with them. There are three scenarios in this respect. A number of colleagues adapt themselves accordingly. Some others have generic means, e.g. intensifying personal communication, of engaging other colleagues when things are blocked because of a pattern, or setting ground rules in the sense of establishing operating principles. Finally, some deal with them empirically on a case-by-case basis per pattern. In our view, the sequence of actions above gives further strength to the value of our work in two dimensions. On the one hand, on providing means that allow people to consider mentality patterns explicitly rather than unconsciously. On the other hand, it provides additional evidence that current practices and experiences in dealing with the matter are kept in people's minds or used in narrow contexts instead of being more widely shared. To further illustrate the point, I could argue that if it had not been for this survey (and exposure to our work), the concrete practices mentioned by colleagues would not have been identified and made known: we would not have been able to discuss the matter on common grounds in the first place.

- *How do you evaluate the degree to which the sub-process could be applied in practice? (e.g. easy to apply, relatively easy to apply, neither easy/nor difficult, relatively difficult to apply, difficult to apply)*

  The answers provided to this question have to be combined with two other factors. The first is the reluctance of the project manager, as illustrated in his report, to apply the sub-process in what he called "*a very prescriptive approach". The second is the non-verbal communication of colleagues during the interviews. My interpretation of both factors is that many colleagues, even some of the ones who have not explicitly admitted it, have doubts on the degree to which the sub-process can be applied in practice — some of them set certain conditions such as "non-formality" or overall organizational culture. We have also identified this issue in what we called "receptiveness" in Section 5.1. On the other hand, based on our experience in applying it, we do not share the same level of hesitation in what concerns its applicability. Therefore, given these different perspectives, it seems that more consistent results on the matter can only be obtained in subsequent applications of the sub-process.

- *Do you think that the process is beneficial in dealing explicitly with the mentality issue? If yes, in what respect?*

  The majority of colleagues responded that the sub-process is beneficial and provided a variety of reasons for justifying this statement. Taking into account that the respondents had not engaged with it in practice, the answers should not be considered as an assessment of the sub-process. Still, similarly to what this thesis advocates, the responses clearly identify the need for a more explicit and systematic way of dealing with the mentality matter in SE.

- *Would you consider applying the primitive and/or sub-process in your work environment?*

  The survey results are twofold and relate to findings already discussed earlier. On the one hand, colleagues shared a positive opinion on the mentality pattern primitive and its concrete representation in what concerns its capacity to capture, organize and communicate recurring turns of mind. On the other hand, although colleagues agreed on its potential value and benefits, they were hesitant on the way the mentality innovation sub-process could be applied.

6.5 Summary of Results

Following the discussion in the previous sections, we present below a summary of the results in what concerns the first two themes of this thesis.

On Mentality Patterns

- The Mentality Pattern primitive establishes a common basis and focal point that individuals understand and recognize, thus systematizing relevant discussions on the matter. Along the same lines, it provides an effective response to the issue that, although people perceive the existence of certain recurring mentality elements, they do not have a means to characterize them explicitly and methodically.

- The primitive facilitates agreement on identifying and capturing several recurring mentalities, including cases on which it can be difficult to achieve consensus such as the ones perceived by colleagues to be related to the character of particular individuals and organizational or team culture. On the other hand, it is more difficult to distinguish between, on the one hand, recurring, widely applicable mentality elements and, on the other hand, bad practices or ways of thinking that reflect personal opinions and preferences.
- The primitive provides an effective communication mechanism between software practitioners. In this sense, it also provides a way of sharing relevant experiences and practices in dealing with matter. Moreover, there is a direct relationship between the level of agreement in characterizing a mentality pattern and its subsequent capacity to act as a communication mechanism.

In addition to the above, and although they have not been direct goals of this thesis, we have provided indications on the following:

- The existence of a mentality-pattern interference phenomenon, a matter that we have explicitly taken into account in the mentality-pattern representation language.
- The convergence of different sources on the specific mentality patterns that are more frequently encountered in practice. Moreover, indications exist that people consider that the most frequent patterns are also the ones that have the most decisive impact.

On the Mentality Innovation Sub-process

- The level of receptiveness of teams and the capacity of the sub-process driver are the most critical factors for the applicability of the sub-process. Consequently, colleagues have expressed a degree of scepticism in what concerns its potential acceptance. On the other hand, there is consensus on recognizing possible benefits as well as on the overall need to devise methods that take into account the mentality matter in a more methodical way in team-leading practices.

- There is evidence that, for some mentality patterns, the sub-process triggers the reflection of individuals and teams in their current practices and drives a constructive change in the way they operate. On the other hand, there exist specific recurring mentalities that are more persistent. In such cases, the sub-process has a smaller or no effect. Therefore, additional means to deal with certain mentality patterns should be devised. Moreover, it appears that the more persistent mentalities are the ones perceived by colleagues to be related to the character of individuals or team (or organizational) cultures. Thus, there is a need to examine the substance of mentality patterns. We make concrete recommendations in this respect in Chapter 9.

Chapter 7

REPOSITORY SUPPORT

7.1 Overview

This chapter is concerned with the third goal and contribution of this work: to *provide a support system through which a repository of mentality elements and associated practices and experiences can be built and shared so that software engineers can make effective use of this knowledge in their practice*. In particular, we discuss the following aspects:

- The system's envisioned scope and requirements (Section 7.2).
- The implementation alternatives and considerations as well as the necessary justifications for the concrete implementation choices (Section 7.3).
- The details of the implementation over the Alfresco Enterprise Content System [5], an open-source platform (Section 7.4).

7.2 Envisioned Requirements

7.2.1 Scope of the System

Before examining the specific requirements and respective implementation, it is important to first reflect on the overall goal and discuss how it impacts the scope of the system in question. In other words, the goal should guide us in recognizing the potential users and functionality expectations, as well as to identify constraints that should be taken into account in requirements' definition and system implementation. In reflecting on the goal, one immediately recognizes two key elements:

- The system should provide the capability to *build* a *repository* of mentality elements (patterns) and associated practices and experiences.
- The repository information should be able to be *shared*.

These two key elements result in a number of direct implications on the scope of the system and its concrete requirements as outlined below:

- Any software practitioner who is interested in mentality patterns is a potential user of the system. Therefore, the system should be multi-user: it should support the definition of different user profiles and respective access rights, allow for user authentication and manage concurrent access.
- Since the system is intended to be shareable, it should allow for multiple means of deployment: in the PC of any individual who wishes to use it for personal purposes, in an intranet for users belonging to specific teams or specific organizations, and over the Web for disperse communities of users.
- Users should be able to access the system using a standard Web browser.
- In order to support the notion of repository, a suitable mechanism for data persistence, such as a database management system, should be employed. The need for a database management system is further justified by the fact that the number of users is not known a-priori and may potentially be significant.
- The system should allow for users to contribute mentality patterns.
- The system should allow users that have the necessary access rights to view and modify information on mentality patterns that other users produce.

The figure below illustrates the envisioned scope of the system. In particular it depicts its multi-user and multiple access mode characteristics, the presence of a database management system and of a user interface for mentality patterns' authoring and viewing.

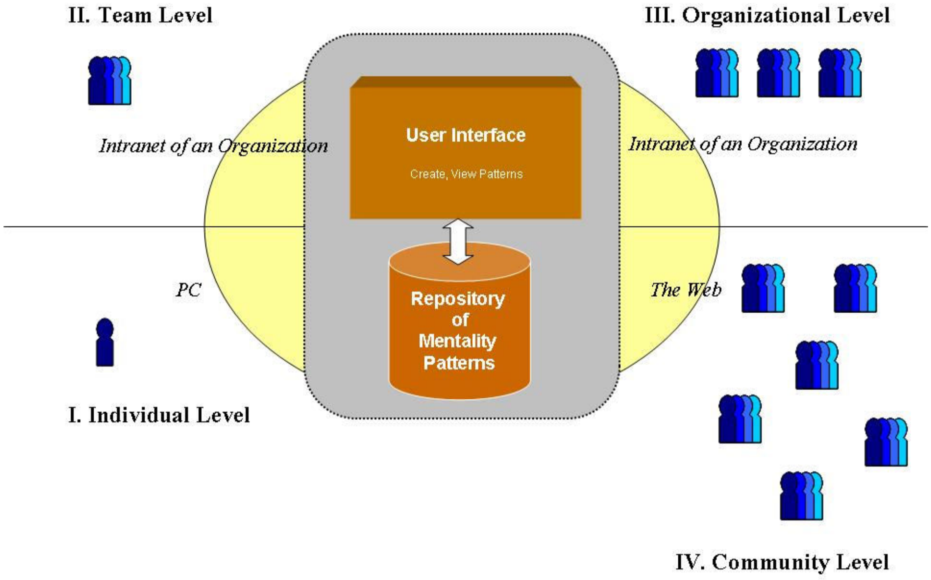

II. Team Level
III. Organizational Level
Intranet of an Organization
User Interface
Create, View Patterns
Intranet of an Organization
PC
Repository of Mentality Patterns
The Web
I. Individual Level
IV. Community Level

Figure 8. Scope of the support system

7.2.2 Functional Requirements

Having a more articulate view of the scope, we proceed by detailing the high-level functional requirements of the system, which we split into three categories:

- Pattern Creation, Registration and Update Requirements
- Search and View Requirements
- Security Requirements[29]

For each requirement we provide a name and a high-level description, including possible acceptable variations. It is important to note that we only present the requirements that we envision as the minimum set necessary to make such a system usable in practice. We are convinced that practitioners will have their own inspirations on how to further enrich the functionality that such a system can offer. In fact, the ability to incorporate new functionality and the existence of a rich set of features that can be employed or adapted by colleagues in order to implement their

[29] Security requirements are often considered as non-functional requirements in SE literature. It is not in the scope of this thesis to discuss or position on the matter. For the readers' convenience, i.e. present a single list of the envisioned requirements, we discuss security as a functional aspect.

own ideas has been a major criterion for the implementation we discuss in Section 7.4.

Table 6. Functional Requirements of the Support System

Name	*Description*
Pattern Creation, Registration and Update Requirements	
1. Pattern Authoring	The system should provide suitable means e.g. an editor or suitable fields in its user interface, in order to be able to create (author) new mentality patterns using the associated representation language. The users should, at any time, be able to "save" their work in an underlying database.
2. Pattern Import	Support the import to the underlying database of pattern representations created in external text editors. Microsoft Word or XML should be considered as the minimum supported formats.
3. Pattern Update	Users should be able to update the information about patterns stored in the system. Users should also be able to delete patterns. Since multiple users may simultaneously update the same pattern(s), the system should guarantee that such updates are performed in a consistent way.
4. Pattern History	The system should maintain the history of pattern updates as well as the different versions of patterns in time.
5. Pattern Metadata	Maintain and manage fixed or user-defined information about pattern metadata, for example author, registration date, project, organization.
6. Pattern Folders	Allow the creation of user-defined folders of patterns, for instance, per organization or per project.
7. Pattern Associations	Provide the capability for users to define associations between patterns.
8. Approvals	Provide the capability to enforce and manage the approval by different users (at least by an administration role) of the registration or modification of a pattern in the repository
Search and View Requirements	
9. Search by Pattern Contents	Provide suitable means for users to search for patterns stored in the repository based on user-specified criteria referring to the pattern contents.
10. Search by Pattern Metadata	Provide suitable means for users to search for patterns stored in the repository based on user-specified criteria referring to the pattern metadata information e.g. date, author.
11. View Pattern Details	Users should be able to view the detailed contents of a pattern stored in the repository. Users may also be able to view the stored information concerning the pattern metadata and the different versions of a pattern in time.
Security Requirements	
12. User Roles & Access Rights	The system should allow for defining multiple roles for users and define respective access rights per role.
13. User Authentication	The system should allow for user authentication by a means of username and password.

Notes on the Requirements

The need for the “Pattern Authoring”, “Pattern Update” and for all the search, view and security requirements is evident given the envisioned scope (discussed in Section 7.2.1) and the established industry practices in what concerns multi-user, repository-oriented systems. However, the remaining requirements should be further justified. In what follows we present the rationale and respective justification for each of those requirements:

- Pattern Import: it should be expected that many users will prefer to use standard, popular editors (e.g. Word, XML) for specifying mentality patterns instead of the means provided by the system itself. The ability to import to the repository mentality patterns authored in such editors will encourage the wider adoption of the system.
- Pattern History: since one of the goals of the system is that individuals can share their experiences on mentality patterns as they are accumulated over time, the system should be able to maintain the evolution of such information.
- Pattern Metadata: given that we envision the repository to be widely shareable, it should maintain and manage information on metadata for each pattern such as author, date, particular projects, teams and organizations. It should also permit users to define their own metadata in order to meet their specific needs on using the notion of mentality pattern in their environments.
- Pattern Folders: following the need for metadata management, the repository should allow users to organize patterns using folders, e.g., per organization, project, team, inter-alia.
- Pattern Associations: this requirement is directly related to the “mentality patterns interference” phenomenon we have discussed in Chapters 4 and 6. The repository should allow users to capture associations between patterns — either based on the types we have already identified herein or to even define new types of associations between patterns as they result from their experiences.

- Approvals: it refers to the need to have approval mechanisms for registering or modifying a pattern in the repository so that the consistency of the stored information can be maintained. This need is particularly relevant for a deployment of the repository in an organizational setting.

7.2.3 Non-Functional Requirements

Although the term “non-functional requirements” has been used for decades in SE, there is still no consensus on their exact definition, classification, scope and representation. Most of the works discussing the matter introduce other terms such as “constraints”, “quality” or “attribute” and define a set of properties that the system should satisfy, such as the well-known “-ilities”, for instance, availability, reliability, maintainability, portability, and usability. In general, non-functional requirements are understood as characteristics of a system that are less visible to its end-users[30]. In what follows, we outline some of those characteristics without claiming that we have provided a thorough analysis of the system non-functional requirements. Given the scope and purpose of the system, the elements we refer to are the ones that we consider to be the most important to take into account for subsequent choices made for the implementation of the system.

Performance and Scalability. The load of the system is not known in advance. In other words, characteristics that are related to performance and scalability such as the numbers of users, the number of transactions and the amount of data stored are not established. Therefore, the system, by design, should permit for both “horizontal” and “vertical” scalability if such a need arises.

Integration. The system should provide appropriate means for accessing data and exchanging information with other systems, e.g., other applications or repositories.

[30] It is not in the scope of this thesis to provide an analysis of the various definitions of non-functional requirements.

Adaptability. The system should allow for "customizations" to be performed depending on the needs of its users such as modifying the user-interface or the structure of the underlying repository.

7.3 Implementation Considerations

Before discussing the implementation of the envisioned system, we present the main technological alternatives and justify the decisions made in relation to them. Those are considered in the following three dimensions:

- *"Structured vs. Unstructured" data support*. This is mainly concerned with the following dilemma: should the mentality-patterns related information stored in and managed by the system be mapped to a database schema designed specifically for this purpose over an RDBMS, or should it be captured, stored and managed as "unstructured" content as typically performed by Enterprise Content Management (ECM) systems [31]?
- *"Built from scratch" vs. existing solution"*: should the system (or a significant part of it) be custom-built or should available solutions be used that provide, either "out of the box" or via customization, the required functionality?
- *Open Source vs. Commercial software*: should the system be based on open-source components and technologies (or even entire solutions) or should a comparable commercial solution offered by software vendors be preferred?

In reflecting on the above dimensions, one realizes that the first two are related. If one opts for the structured data-support approach, then the space for using an existing solution becomes limited: whereas existing RDBMS and available programming tools for building the system can be used, the database schema, the system logic and the user interface should be built from scratch given that the mentality patterns representation is very specific to our proposal. On the other hand, if the mentality patterns' information is captured as "unstructured" content, for instance as text files, then an existing ECM solution must be applied and be customized as necessary to account for the specificities of the envisioned system.

The word "must" above is used on purpose to immediately exclude the option of building from scratch an ECM-like system. This is a task that requires significant and specialized know-how that only few organizations have. Even if a subset of such a system could be built from scratch in a reasonable timeframe, the value obtained compared to reusing an existing solution would be highly debatable and the principle of software reuse instructed us to disregard such an option.

Therefore, in deciding the approach to be followed for the implementation of the system, two options existed:

- To build the system from scratch using a "structured" data approach.
- To adopt an "unstructured" data support approach, thus using and customizing an existing ECM solution.

Clearly, in deciding which approach to follow, specific decision criteria had to be set. In fact, the problem at hand resembles to a great extent a typical "make vs. buy" decision. Therefore, we compared the two options above using a set of four criteria[31], namely:

1. The coverage of the envisioned requirements, i.e., the extent to which each of the two options meets the requirements stated in sections 7.2.2 and 7.2.3. Evidently, if the system is built from scratch, then one can argue that the functional requirements coverage can always be 100%.

2. The effort required and the respective cost to implement the system.

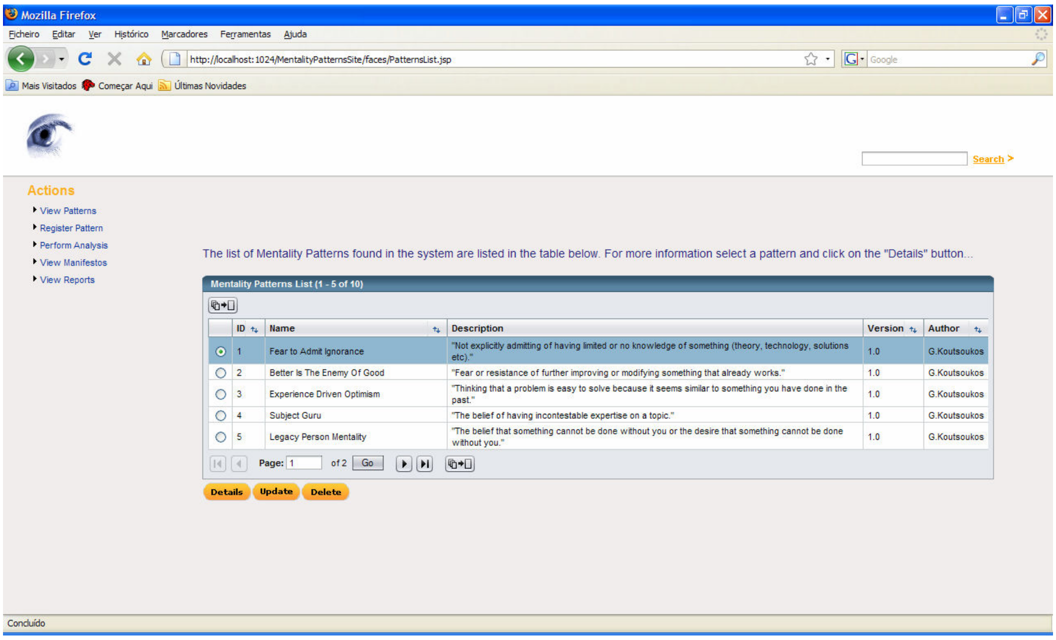
3. The risk that the system will exhibit high error rates or even failures.

4. The capability to extend/evolve the system in the future.

Having the above in mind, and in order to identify the effort required, we first built a prototype of a "structured" data support system using Java, JSF, and the Sun GlassFish V2 Server over the MySQL RDBMS. For illustrative purposes, a screen-

[31] The goal of this analysis is to justify the implementation choices made and not to perform a detailed comparison across a wide variety of factors or a thorough cost-benefit analysis as one may expect in a "full blown" evaluation.

shot of this prototype is given in Figure 9 below. It was soon realized that the implementation of the full system implied significant effort. Therefore, we had to


Mozilla Firefox
Ficheiro Editar Ver Histórico Marcadores Ferramentas Ajuda
http://localhost:1024/MentalityPatternsSite/faces/PatternsList.jsp
Google
Mais Visitados Começar Aqui Últimas Novidades
Search >
Actions
View Patterns
Register Pattern
Perform Analysis
View Manifestos
View Reports
Page: 1 of 2 Go
Details
Update
Delete
Concluído

The list of Mentality Patterns found in the system are listed in the table below. For more information select a pattern and click on the "Details" button...

Mentality Patterns List (1 - 5 of 10)

	ID	Name	Description	Version	Author
●	1	Fear to Admit Ignorance	"Not explicitly admitting of having limited or no knowledge of something (theory, technology, solutions etc)."	1.0	G.Koutsoukos
○	2	Better Is The Enemy Of Good	"Fear or resistance of further improving or modifying something that already works."	1.0	G.Koutsoukos
○	3	Experience Driven Optimism	"Thinking that a problem is easy to solve because it seems similar to something you have done in the past."	1.0	G.Koutsoukos
○	4	Subject Guru	"The belief of having incontestable expertise on a topic."	1.0	G.Koutsoukos
○	5	Legacy Person Mentality	"The belief that something cannot be done without you or the desire that something cannot be done without you."	1.0	G.Koutsoukos

Figure 9. Snapshot of “build from scratch” repository prototype

also explore the ECM option. In researching the ECM market we found out that such solutions provide, out of the box, a good coverage of the system functional requirements discussed in section 7.2.2 and therefore the intended result could be reached with significantly less effort. The main drawback identified was that we had to allow for any constraints imposed by the chosen ECM platform. For instance, an ECM platform, even customized, may not be able to fully support the envisioned functional requirements and therefore a more relaxed position will have to be adopted. On the other hand, it was obvious that in terms of support for non-functional requirements as well as the risk factor, an ECM platform was a better choice: such systems had been thoroughly tested and had been in use and proven scalable for years and in several organizations worldwide. The same applied for the capability to extend and evolve as wells as documentation, problem solving and support: the wide community of users and developers of such systems guaranteed that their capabilities will evolve with a pace that could not be competed by a custom-built system developed solely by us. As a result, the ECM option was adopted. This led to the final decision that had to be made: whether to opt for an open-source or a commercial ECM platform and in turn, the particular platform to use.

The choice between open-source and commercial software[32] has been straightforward considering the fact that open-source software provides the opportunity to use it for personal or experimental purposes at no cost[33]. Given the research nature of this work and the fact that we aimed for the system to be used by other colleagues we opted for an open-source solution. Following this choice, the next question concerned the particular solution to use. In that respect, the main criterion was the popularity of a solution in what concerns its installed base and its track record in industrial settings. In our view, these two criteria implied maturity of the solution in terms of its functionality offering, future evolution by its community of users and developers, capacity to scale in an environment with large volumes and good integration and adaptability features. In researching the open-source ECM market, the Alfresco solution had been the one that appeared to best satisfy the aforementioned criteria. For instance, at that time, Alfresco had been the only open-source solution that appeared in Gartner's Magic Quadrant [31], a well-known IT market research report that takes into account criteria such as the ones above. Moreover, we consulted colleagues that had used Alfresco in industry projects. All opinions had been positive.

[32] It is not in the scope of this work to refer to the different criteria that can be used for reaching such a decision in general, neither to analyze the advantages and disadvantages of each approach.

[33] The use of commercial software may also provide this option but usually for evaluation purposes only, e.g. for a limited period of time. However, it would be impractical and far from the focus of our work to perform an extensive search of relevant commercial software in order to determine whether we could use such a solution at no cost.

7.4 Implementation[34]

This section discusses the actual implementation of the support system based on the Alfresco ECM solution as well as the extensions we have developed.

7.4.1 Introduction to Alfresco

Alfresco is built on Java and incorporates a variety of 3rd party open-source components. It belongs to the category of software solutions known as Enterprise Content Management (ECM). Such suites have as their core a repository, known as content repository, that is used to store, search, access and control content. The types of different content supported can be documents, images, web pages, multimedia, e-mails and so forth. Over this content repository, ECM solutions offer a set of content related functionality, or else called “services”, which we outline below:

- Document Management: it involves features such as the ability to check in/check out documents in the repository, version-control, document import, taxonomies and security over documents. This set of functionality is also known with the name “Library Services”. Additionally, there exist specialized tools that manage the content of websites based on the core repository. These are known as Web Content Management (WCM) tools.
- Imaging capabilities such as capturing and managing images of paper documents.
- Workflow, i.e. the ability to assign tasks to specific users and route content to such users in order to support content intensive business processes.
- Records Management: such services provide the ability to define rules for archiving content and enforce retention of documents, for instance in order to meet regulatory requirements.

[34] In what follows we do not aim to present a thorough analysis of all the Alfresco capabilities and features. Instead we provide sufficient background information in order to subsequently explain the parts that are more relevant to our work.

- Rights Management: functionality related to protecting content once checked-out of the repository. For instance, preventing documents to be viewed by unauthorized users, being printed or being forwarded.

Apart from the above repository services, ECM solutions offer out-of-the-box "client applications" through which users interact with the repository and use the aforementioned services. In other words, such applications are the "presentation" layer of the solution. Alfresco in particular provides a set of such applications. The two main ones are the Alfresco Explorer and Alfresco Share. The former is oriented to document management, whereas the latter targets the document centric collaboration of teams. In our implementation we have used the Alfresco Explorer application. Figure 10 below, borrowed from [70], provides a pictorial representation of the Alfresco functional architecture.

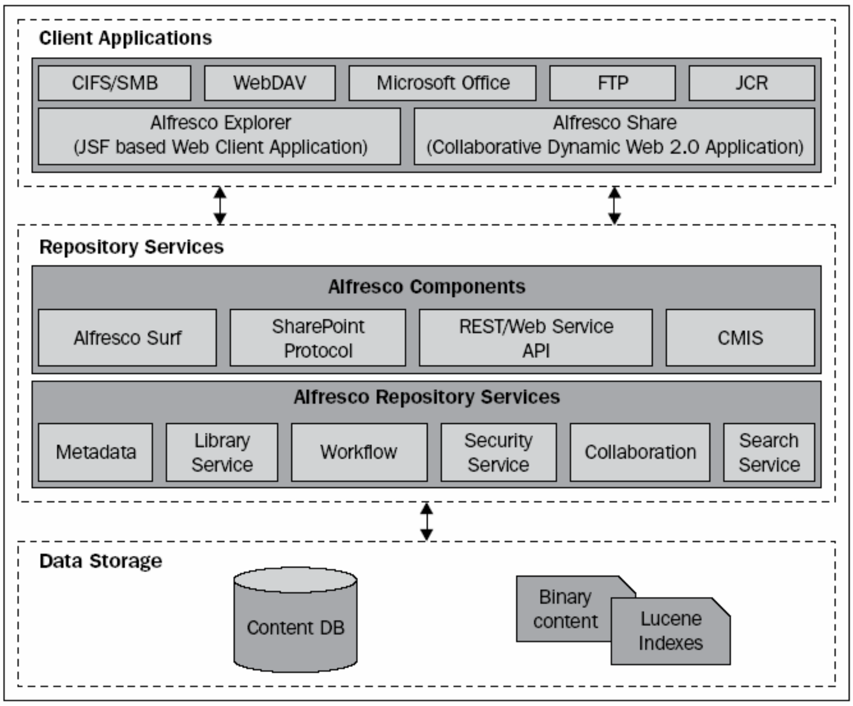


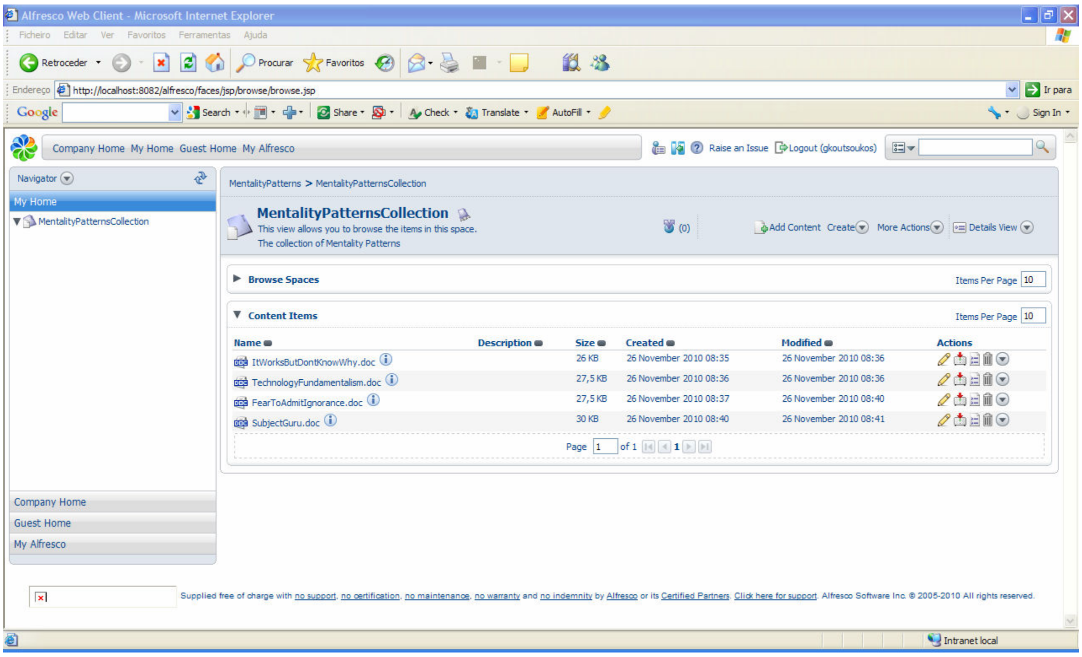
Figure 10. Alfresco V3 Functional Architecture (from [70])

In our implementation we used the Alfresco 3.3.0 version, in a Java 1.5.0-12 environment, configured over the Apache Tomcat server (version 6.0.18) and the MySQL database system (version 1.35-community).

7.4.2 Mapping of Requirements to Alfresco Capabilities

In this section we discuss the way Alfresco supports the envisioned requirements presented in Section 7.2.

In what concerns the scope of the system, Alfresco fully supports all envisioned means of deployment depicted in Figure 8, accessed through the browser-based Alfresco Explorer client application. The following figure depicts the "look and feel" of the implementation based on the Alfresco Explorer with the repository containing a collection of mentality patterns in an MS Word document format.



http://localhost:8082/alfresco/faces/jsp/browse/browse.jsp

MentalityPatterns > MentalityPatternsCollection

MentalityPatternsCollection

This view allows you to browse the items in this space.

The collection of Mentality Patterns

Name	Description	Size	Created	Modified	Actions
ItWorksButDontKnowWhy.doc		26 KB	26 November 2010 08:35	26 November 2010 08:36	
TechnologyFundamentalism.doc		27,5 KB	26 November 2010 08:36	26 November 2010 08:36	
FearToAdmitIgnorance.doc		27,5 KB	26 November 2010 08:37	26 November 2010 08:40	
SubjectGuru.doc		30 KB	26 November 2010 08:40	26 November 2010 08:41	

Page 1 of 1 1

Supplied free of charge with no support, no certification, no maintenance, no warranty and no indemnity by Alfresco or its Certified Partners. Click here for support. Alfresco Software Inc. © 2005-2010 All rights reserved.

Figure 11. A view of a mentality patterns' repository in Alfresco

In what concerns functional requirements, Table 7 below provides a summary of the requirements supported "out-of-the-box" by Alfresco. In some cases, we had to provide suitable extensions to Alfresco in order to implement those in a more appropriate way. These are discussed separately in the subsequent section. In general, as illustrated in Table 7, Alfresco provides a comprehensive coverage of the envisioned requirements for a repository of mentality patterns.[35]

Table 7. Mapping of Functional Requirements to Alfresco capabilities.

Requirement Name	***Supported "out-of-the-box" by Alfresco***	***Notes on Implementation/ Customizations***
1. Pattern Authoring	✓	Extension. Details in Section 7.4.3.
2. Pattern Import	✓	Extension. Details in Section 7.4.3
3. Pattern Update	✓	
4. Pattern History	✓	
5. Pattern Metadata	✓	Extension. Details in section 7.4.3
6. Pattern Folders	✓	
7. Pattern Associations	✓	Extension. Details in section 7.4.3
8. Registration Approval	✓	
9. Search by Pattern Contents	✓	
10. Search by Pattern Metadata	✓	Extension. Details in section 7.4.3
11. View Pattern Details	✓	
12. User Roles & Access Rights	✓	
13. User Authentication	✓	

7.4.3 Extensions performed in Alfresco

In this section we discuss the cases in which we had to perform extensions to Alfresco in order to fully support some of the requirements presented in Table 7. A good introduction on the details of performing such extensions to Alfresco in provided in [64].

[35] In Table 7, in the cases in which we indicate that Alfresco supports the respective functional requirements without extensions, it is meant that it fully supports the requirements' descriptions as provided in Table 6. We do not discuss these cases separately given that it will require us to provide a comprehensive analysis of the respective Alfresco capabilities, which is not in the scope of this thesis.

There are four cases in which we provided extensions to Alfresco: the “authoring” of patterns, the import to the repository, defining pattern metadata and associations and providing the ability to search using additional conditions on such metadata. The implementation details and respective source code for such extensions are provided in Appendix B. In summary, we have performed the following:

- Developed a custom content model whose core element, titled “Mentality Patterns Description Document” is a content type designated to explicitly capture descriptions of mentality patterns, either as text documents or in any other format. Moreover, we have defined two custom properties, i.e. pieces of metadata associated with this type: the publication date of the pattern and the name of the person that authorized the pattern.[36]

- Defined three types of optional associations for the aforementioned content type in order to be able to capture the relationships between patterns that we have discussed in Chapter 4. The “Causes” association is used for cause-effect relations in which a given mentality pattern can be the underlying cause for other patterns. The “Triggers” association denotes that a pattern can trigger the emergence of other pattern(s). Finally, the “In Conflict With” association is used to capture cases of potential conflicts between peers exhibiting the patterns in question.

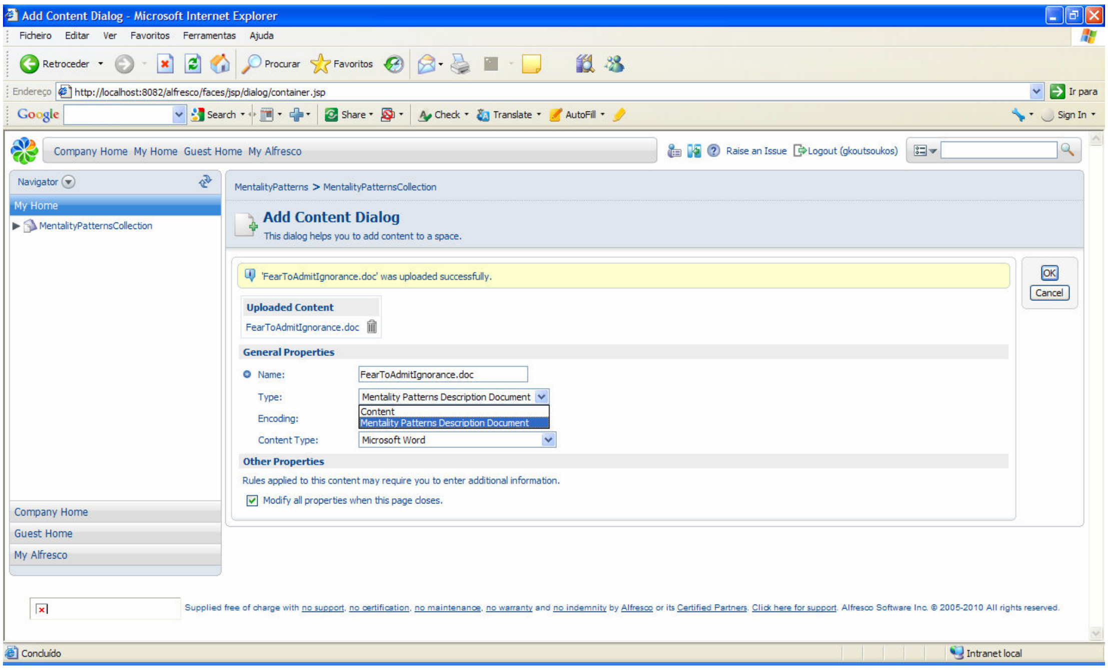
[36] Alfresco provides a rich set of metadata “out-of-the-box” such as author, description and so forth. Hence, it has not been necessary to define a broad set of custom metadata. Moreover, the properties we have defined should also be considered as an illustration on how such custom metadata for the mentality pattern representation can be defined in Alfresco. It should also be noted that the different elements of the mentality pattern representation are data rather than metadata and therefore should not be defined as such. Moreover, most of such elements, namely “Other Names”, “Symptoms”, “Representative Quotes”, “Consequences”, Anecdotal Stories and Examples” and “Possible Causes” make sense only within the context of a given pattern. Therefore, we have not defined separate content types for these. On the other hand, in order to be explicitly captured, the “Related Patterns” element is handled via the definition of appropriate associations in Alfresco.

- Configured the Alfresco Explorer in order for users to be able to work with the custom model above. The interaction of users with the repository in terms of illustrative screens is presented below.

Pattern Authoring and Pattern Import

In Alfresco, users have two main options in order to publish content in the repository. The first option is to author the content using the Alfresco Explorer inline content editor and subsequently publish it in the repository. This is performed using the respective “Create Content” menu. The second option is to import to the repository content that is “authored” via other means, e.g. the Microsoft Word editor. The Alfresco menu used for this purpose is called “Add Content”.

In our extension, when a user clicks the “Create Content” or “Add Content” menus, the “Mentality Patterns Description Document” custom type above, appears as choice in the list of possible content types available in the repository. For the “Add Content” scenario this is illustrated in Figure 12 below. The case is similar for creating content.



Endereço: http://localhost:8082/alfresco/faces/jsp/dialog/container.jsp

MentalityPatterns > MentalityPatternsCollection

Add Content Dialog

This dialog helps you to add content to a space.

'FearToAdmitIgnorance.doc' was uploaded successfully.

Uploaded Content

FearToAdmitIgnorance.doc

General Properties

Name: FearToAdmitIgnorance.doc

Type: Mentality Patterns Description Document

Content
Mentality Patterns Description Document

Encoding:

Content Type: Microsoft Word

Other Properties

Rules applied to this content may require you to enter additional information.

☑ Modify all properties when this page closes.

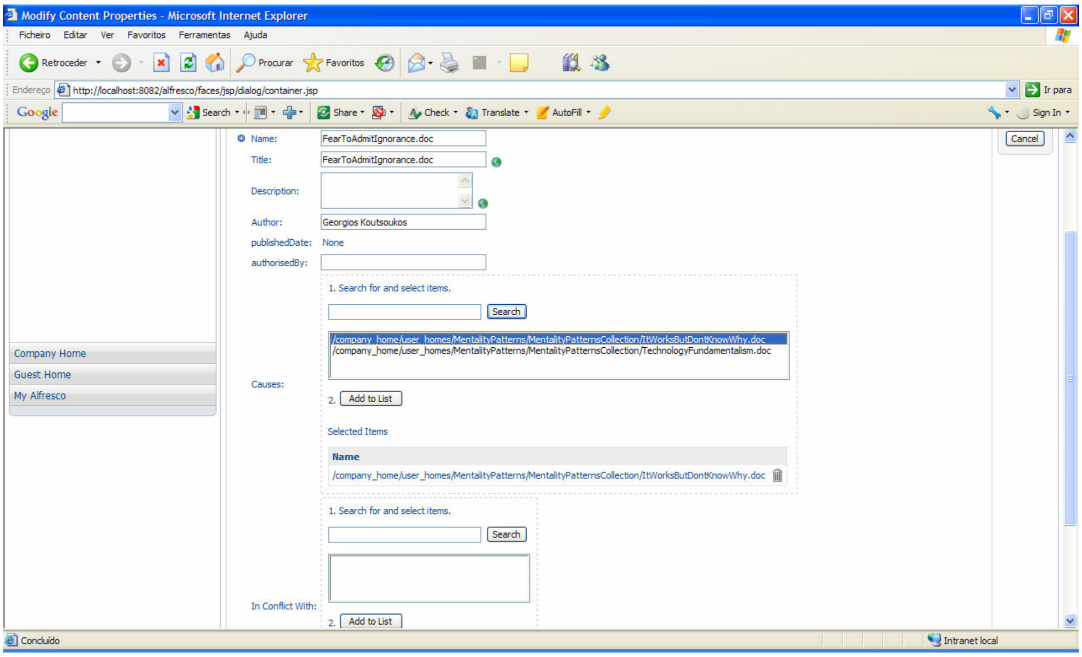
Supplied free of charge with no support, no certification, no maintenance, no warranty and no indemnity by Alfresco or its Certified Partners. Click here for support. Alfresco Software Inc. © 2005-2010 All rights reserved.

Figure 12. Import of a mentality patterns document in Alfresco

A limitation of our implementation is that in the “Pattern Authoring” case (“Create Content”) we have not been able to extend the Alfresco Explorer inline content editor in a way that, upon selection of the mentality patterns’ content type, the various elements of the patterns’ representation language become available “out-of-the-box” in the editor.

Pattern Metadata and Associations

The result of our extension in what concerns custom metadata and the definition of associations between mentality patterns is illustrated in Figure 13. In any pattern, users can define the publication date of the pattern, the person that authorized the pattern and the three types of associations discussed earlier. In the figure below, the “Fear-To-Admit-Ignorance” patterns is associated to “It Works (but I do not know why”) and the “Technology Fundamentalism” patterns using a “Causes” relationship. The other two types of association are defined in a similar way in the Alfresco Explorer.



http://localhost:8082/alfresco/faces/jsp/dialog/container.jsp

Name: FearToAdmitIgnorance.doc

Title: FearToAdmitIgnorance.doc

Description:

Author: Georgios Koutsoukos

publishedDate: None

authorisedBy:

Causes:

1. Search for and select items. [Search]

/company_home/user_homes/MentalityPatterns/MentalityPatternsCollection/ItWorksButDontKnowWhy.doc
/company_home/user_homes/MentalityPatterns/MentalityPatternsCollection/TechnologyFundamentalism.doc

2. [Add to List]

Selected Items

Name
/company_home/user_homes/MentalityPatterns/MentalityPatternsCollection/ItWorksButDontKnowWhy.doc

In Conflict With:

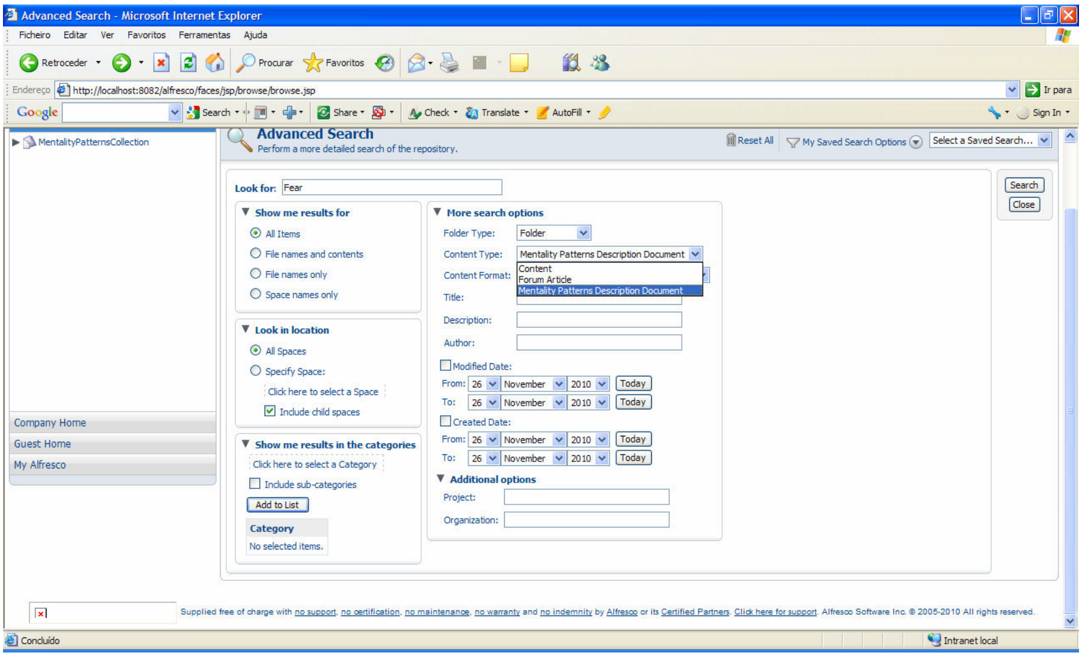
1. Search for and select items. [Search]

2. [Add to List]

Figure 13. Mentality Patterns Custom Metadata and Associations

Search

Figure 14 illustrates the ability to perform a repository search based on the mentality patterns custom model. Additionally, we have provided the ability to include additional options for search, based on information about mentality patterns in specific projects and organizations.



Endereço: http://localhost:8082/alfresco/faces/jsp/browse/browse.jsp

MentalityPatternsCollection

Company Home
Guest Home
My Alfresco

Advanced Search
Perform a more detailed search of the repository.

Look for: Fear

▼ Show me results for
● All Items
○ File names and contents
○ File names only
○ Space names only

▼ Look in location
● All Spaces
○ Specify Space:
Click here to select a Space
☑ Include child spaces

▼ Show me results in the categories
Click here to select a Category
□ Include sub-categories

Category
No selected items.

▼ More search options
Folder Type: Folder
Content Type: Mentality Patterns Description Document
Content
Forum Article
Mentality Patterns Description Document
Content Format:
Title:
Description:
Author:
□ Modified Date:
From: 26 November 2010
To: 26 November 2010
□ Created Date:
From: 26 November 2010
To: 26 November 2010

▼ Additional options
Project:
Organization:

Supplied free of charge with no support, no certification, no maintenance, no warranty and no indemnity by Alfresco or its Certified Partners. Click here for support. Alfresco Software Inc. © 2005-2010 All rights reserved.

Figure 14. Search Screen for Mentality Patterns

Chapter 8

DISCUSSION

8.1 Reflection and Contribution

This section has two interrelated objectives. Firstly, to appropriately position our work with respect to the diverse types of software engineering research in what concerns the nature of research questions and respective results. At the same time, to discuss its contributions having this context in mind.

In dealing with the first objective, we employ the classification of software engineering research provided by Shaw in [71]. Although this work mostly refers to research papers, we believe that it is equally applicable to a PhD project. Shaw's classification of SE research is based on three dimensions: the type of research question, the type of result produced and the criteria with which such results are evaluated. Table 8 below provides a summarized version. A more detailed view of this classification is presented in Appendix C.

Table 8. Software Engineering Research Classification (Shaw)

Type of Research Question	**Type of Result**	**Validation Criteria**
Method or means of development	*Procedure or technique*	Analysis
Method for analysis	*Qualitative/Descriptive Model*	*Evaluation*
Design, evaluation, or analysis of a particular instance	Empirical model	*Experience*
Generalization/Characterization	Analytic model	Example
Feasibility	*Notation or tool*	Persuasion
	Specific solution	Blatant Assertion
	Answer or judgment	
	Report	

The part of this thesis that concerns the Mentality Pattern primitive addresses a "Generalization/Characterization" type of question and produces a "Qualitative/Descriptive Model". The Mentality Innovation Sub-Process part is related to a "Method or Means" type of research and results in a "Procedure or Technique". In both cases, the validation of the results is based on "Evaluation" and "Experience". The repository part of this work is complementary to the above in the form of an implemented "Notation or Tool" that embodies and supports the aforementioned model and method. In what follows, we discuss the specific contributions of our work and exemplify their associations to the aforementioned classification[37].

Table 9. First research goal and result of the thesis

Goal 1: provide a systematic and consistent way of capturing, making explicit and communicating recurring human mentality elements in SE.

Summary of 1st Result (Chapter 4): the Mentality Pattern as an abstraction primitive along with its definition, representation, and guiding principles for deciding, out of the plethora of possible mentality elements, which ones should be characterized as mentality patterns.

Discussion and Contribution

The first result of this thesis addresses a research question of a "Generalization/Characterization" type. As demonstrated in the first chapter, there is substantial evidence from a variety of sources, for instance the literature and discussions in the wider-community, on the existence of certain recurring human turns of mind, usually referred to as "mentality", that have significant impact in SE practice. Many of these sources even use certain designations to refer to such mentality elements and identify their respective consequences. Our own interaction with many colleagues over several years of researching the matter in real-life

[37] Some of the examples provided in Shaw's classification model are in the form of "keywords". Therefore, their "exact" semantics are open to many interpretations. The way we understand and associate those to this work are explained in the "Discussion and Contribution" parts.

settings, including the results of the survey we reported in Chapter 6, has shown that the vast majority share the same view on the existence of such mentality elements. The problem is that, whereas there is agreement on the matter of existence, there is significant ambiguity on the matter of "semantics", i.e. *"What exactly do we mean by such recurring mentalities?"* and *"What are their important characteristics?"*. Consequently, references to certain mentalities are either inconsistent or are characterized and interpreted using different terms such as "organizational cultures", "management styles", "personality types", inter-alia. Even worse, in the absence a common meaning, a uniform and systematic way to capture and communicate such recurring mentalities is also lacking. As a result, existing knowledge and associated practices on the matter are kept on colleagues' minds or used in narrow contexts instead of being available to the wider community.

With respect to the problem above, the first set of contributions of this thesis is twofold. Firstly, it makes explicit and gives "semantics" to such recurring mentality elements via the mentality pattern primitive and a set of principles for the uniform and consistent characterization of recurring mentality elements as mentality patterns. Secondly, it provides a pattern-style representation as a methodical way to capture the characteristics and communicate such recurring elements.

The mentality pattern primitive and representation has been inspired by work in the patterns field, and its definition has been guided by works from psychological disciplines. It has been iteratively evaluated and adapted based on the input of colleagues and the experiences obtained by its application in various industry settings. In other words, using the terminology of the above classification, the first result of this thesis is a "Qualitative/Descriptive Model" for recurring mentalities in SE.

In accordance to the specific goals of our thesis, the properties of the model examined in practice have not been its completeness or accuracy when reasoning about the mentality phenomenon, but its capacity to organize the different perceptions on the matter, facilitate agreement on the identification of concrete recurring mentalities and its effectiveness as a common communication mechanism.

As discussed in detail in Chapter 6, although some difficulties have been found, the results across all such dimensions have been very encouraging.

Finally, it is worth mentioning that the mentality patterns representation captures a phenomenon that we call “mentality patterns interference”. We argue that this aspect of mentality patterns should be taken into account explicitly in team management and decision-making, and we have identified a number of cases that illustrate the point.

Limitations

The main limitation of our work resides in the extension of the definition of the primitive with respect to the guiding principles for characterizing mentality patterns that was discussed in Section 4.6. On the one hand, as explained in Chapter 6, we were “forced” to develop this extension in order to be able to distinguish mentality patterns from bad practices and the personal opinions and preferences of colleagues. On the other hand, this extension works by exclusion and is more complex than what we had initially intended to provide. In other words, we have not been able to formulate a definition of the primitive that is straightforward to the point that the average software practitioner can immediately characterize recurring turns of mind as mentality patterns without considering the accompanying guiding principles.

Having discussed the first goal and contribution of this work, we now proceed to perform a similar analysis for the remaining two.

Table 10. Second research goal and result of the thesis

Goal 2: develop an organized way to supplement software engineering practices so that mentality elements can be taken into account as first-class concerns.
Summary of 2nd Result (Chapter 5): the Mentality Innovation Sub-process along with its principles, its detailed modus operandi, its foundations based on widely accepted psychological theories, and the results and lessons-learned from its application in real-life scenarios.

Discussion and Contribution

The second goal of this thesis is logically derived from the first one: once a model is made available to capture recurring mentalities in a systematic way, we should devise the means through which people can take them explicitly into account during their practices. In other words, using the terminology of the SE classification above, the type of research question is “Method or Means”[38] and the result is a detailed “Procedure or “Technique”: the Mentality Innovation Sub-process. The sub-process is not merely a set of guidelines or advice. Instead, starting from a set of principles, we have provided detailed operating instructions in the form of phases to be followed and activities to be performed. Both the guiding principles and the operational details emerged, were adapted and evaluated by applying the sub-process in practice as described in Chapter 3. Hence, using the aforementioned terminology the validation criteria have been both “Evaluation” and “Experience”. Moreover, we have employed the widely accepted psychological theories of Cognitive Dissonance and Self-Perception to provide a solid basis on the principles and operational aspects of the sub-process.

In other words, our thesis contributes a range of instruments in order to deal with mentality patterns as first-class concerns in practice: from psychological foundations and guiding principles to operationalization instructions and respective results and lessons-learned derived from practical experience. In particular, the obtained results indicate that the sub-process effectively drives constructive change in the way SE teams operate with respect to certain recurring turns of mind. On the other hand, the sub-process alone may not suffice for dealing with all relevant cases in all contexts. In spite of this, and given the consequences that certain mentality patterns may have in software practice, we believe that the sub-process (as proposed) is a significant step forward compared with the alternative of either ignoring the matter or dealing with it in ad-hoc ways.

[38] In Shaw’s classification, this type of research question appears to primarily refer to software development. In our case, the term has a wider sense, mostly related to team management in SE.

Having said this, we consider as another important contribution of our work the fact that it acts as a valuable guidance for individuals who may be interested in devising their own methods of accounting for the mentality matter in their practices.

Limitations

In order to account for the constraints of real-life settings, different options exist in certain parts of the sub-process. Therefore, for a particular instantiation, corresponding choices need to be made. This aspect increases flexibility, an important factor for the applicability of the sub-process in a real-life environment. The side-effect is that it has not been possible to identify and reason about the consequences of different choices in the various instantiations of the sub-process, for example, to isolate and reason about the impact of performing the mentality principles setup phase in "occasions of opportunity" as opposed to dedicated sessions. I consider this restriction to be the direct result of employing "Real-Life" and "No Observer" research approaches instead of "Controlled Variables" and "High Resolution" ones, for instance, controlled experiments.

The second limitation concerns the dependency that we designed the operationalization of the sub-process to have on the driver. To a large extent, this dependency stems from the fact that the sub-process has emerged as a result of our research practice, in which we had to assume the driver role. Consequently, it is unclear whether an alternative sub-process design could have been devised in which the criticality of the driver role would have been mitigated.

Having discussed the mentality innovation sub-process part, we proceed to reflect on and discuss the third goal and contribution of our thesis — the support system.

Table 11. Third research goal and result of the thesis

Goal 3: provide a support system through which a repository of such mentality elements and associated practices and experiences can be built and shared so that practitioners can make effective use of this knowledge in their practice. **Summary of 3rd Result** (Chapter 7): the set of requirements for such a support system and a concrete implementation using the Alfresco ECM solution.

Discussion and Contribution

In what concerns the relationship between the support system and the mentality innovation sub-process, there are two cases that merit discussion. On the one hand, individuals interested in applying the sub-process in their practices can take advantage of the system in order to organize relevant engagements in a more methodical way, for example, in order to keep track of mentality patterns and of sub-process practices and results in different projects and teams. Unfortunately, due to the point in time in which the support system was concluded, we did not have the opportunity to use it in that context. However, the support system can be used independently of whether our sub-process is also applied or not.

The system has been built with the goal of acting as a knowledge- and experience-sharing repository for mentality patterns. In that sense, contrary to the mentality pattern primitive, it does not stand as an entirely independent contribution but as a complementary tool. Clearly, without the underlying pattern primitive in place, such a repository can be of only limited value: in the absence of systematization in "semantics", the repository can merely become a collection of inconsistent information. In other words, it is because of the level of consistency and systematization provided by the mentality pattern primitive that the system can provide tool-support for organizing the contributions that individuals may wish to make and allow those contributions to be more widely available, either at a team level, at an organizational level or to the wider community. We believe that this is a significant step forward in mitigating the fact that existing knowledge and practices associated with dealing with the mentality matter in SE are often kept on the minds of those individuals or used only in narrow contexts[39]. To further illustrate the point, we have indicated the existence of a mentality-patterns interference phenomenon that we observed when applying the sub-process, provided concrete examples of interference, and identified the need for further examination of this matter. Clearly, other people may have their own examples and insights to contribute from relevant

[39] Therefore, the contribution of the support system does not refer to a "technical" innovation achieved in the way it has been implemented.

experiences in their particular contexts. The existence of the support system allows for such an examination to be performed in a more organized way and results from those experiences to become more widely available.

Limitations

In reflecting on our work on the support system, there are two main limitations to report. Firstly, the fact that the choice of Alfresco has not been based on a thorough comparison with other open-source ECM solutions. Given that this has not been the main emphasis of our work, and in view of the fact that performing such a task would be very time-consuming, we opted for Alfresco having as sole criterion market reports and colleagues' opinions. The consequence of having made this choice is that it is unclear whether better coverage and support for the system envisioned requirements could have been provided by other open-source solutions.

The second limitation of our work is that, due to the point in time it was implemented, the support system has not been used by other people. Consequently, apart from our own use of it, we have not been able to evaluate its capabilities in a wider context. Having said this, our next step is to make the tool widely available on the Web in order to start materializing the objective of sharing knowledge and experience on the mentality matter in SE.

8.2 On the Quality of the Research Process

The subject of quality in qualitative research has been controversial for several decades in the scientific community. In general, three opposing views can be found in the literature. The first is that qualitative research should be evaluated using the same criteria as quantitative research — usually known as internal validity, external validity, reliability and objectivity. The second is that a different set of criteria should be used for qualitative research. Over the years, many authors who advocate this view have proposed a number of different approaches for establishing such criteria. Still, no consensus has been achieved. The third view questions the appropriateness of establishing any criteria for evaluating qualitative research. As put in [69],*"Yet* *after all of this effort, we seem to be no closer to establishing a consensus on quality criteria, or even on whether it is appropriate to try to establish such a consensus."* Therein, the authors argue for a shift from evaluation to "appraisal" — "*the exercise of wise judgment and keen insight in recognizing the nature and merits of a work"*. As the debate persists in the scientific community, the criteria proposed by Lincoln and Guba in [49] are the ones generally employed for evaluating quality (referred to as "trustworthiness") of qualitative research. The trustworthiness of research can be judged by using four criteria: *credibility*, *dependability*, *confirmability* and *transferability*. Lincoln and Guba also offered a set of techniques to help ensure that such criteria are met. In what follows, we present those quality criteria and discuss the way they have been established in our work.

Credibility

Credibility corresponds to internal validity in quantitative approaches. In order to ensure credibility Lincoln and Guba suggest the following five techniques[40]:

a. A set of three *activities*, namely:

   i. *Prolonged engagement:* it refers to the "*investment of sufficient time to achieve certain purposes: learning the culture, testing for misinformation introduced by distortions either of the self or of the respondents, and building trust"*.

   ii. *Persistent observation:* the purpose of this activity is to *"identify those characteristics and elements in the situation that are most relevant to the problem or issue being pursued and focusing on them in detail. If prolonged engagement provides scope, persistent observation provides depth"*.

   iii. *Triangulation:* it refers to the application and combination of multiple and different *sources*, *methods*, *investigators* and *theories* in the study of a phenomenon. The first type of triangulation, *sources*, is also known as data

[40] Text in italics refers to the explanations of the techniques as provided by Lincoln and Guba.

triangulation, and principally refers to gathering data from different locations, at different times and from different people. The second type, *methods*, refers to using more than one method for data gathering, such as observations, interviews, questionnaires, inter-alia. *Investigator* triangulation is essentially using multiple investigators in performing a study. Finally, *theories* triangulation is the application of more than one theoretical view in interpreting the phenomenon.

b. *Peer debriefing:* it is *"a process of exposing oneself to a disinterested peer in a manner paralleling an analytic session and for the purpose of exploring aspects of the inquiry that might otherwise remain only implicit within the inquirer's mind. [...] there is no formula to prescribe how a debriefing session should be conducted [...] it is clear the debriefer must be someone who is in every sense the inquirer's peer, someone who knows a great deal about the substantive area of the inquiry and the methodological issues".*

c. *Negative case analysis*: it is*"a process of revising hypotheses with hindsight. The object of the game is continuously to refine a hypothesis until it accounts for all known cases".*

d. *Referential adequacy:* it refers to the need of keep research data in archived form in order to enable the "auditability" of the result.

e. *Member checks:* "*data, analytic categories, interpretation and conclusions are tested with members of those stakeholding groups from whom the data were originally collected, is the most crucial technique for establishing credibility [...] a member check is both informal and formal, and it occurs continuously".*

In what concerns our work, the aforementioned techniques to ensure credibility have been applied in several dimensions.

Prolonged engagement and *persistent observation* have been a central characteristic of our research. As explained in Chapter 3, we employed a collaborative research practice approach in which we have been continuously part of software teams and organizations in real-life industrial contexts for approximately 10 years. This research practice allowed us to be peers rather than "*a stranger in a strange land that draws undue attention to the inquirer*" as Lincoln and Guba put it. Consequently, possible distortions or overreactions due to the presence of an "outsider" have been circumvented. Moreover, we have been able to identify, gather, evaluate and compare data from a considerable number of different situations relevant to our enquiry. On the other hand, this prolonged participation had the risk of "going native", i.e., lose research perspective, focus and judgement. The way to mitigate this risk has been our awareness of its existence since the early stages of this research. This awareness is consistent with Lincoln and Guba's view on the way to prevent this risk.

Triangulation has also been a central element of our work. In particular, we have applied all forms of triangulation: firstly, we have gathered data from multiple organizations, residing in different countries, at different times, and from a variety of colleagues; secondly we have applied multiple methods of data gathering such as observations, informal discussions, unstructured and structured interviews and the literature; we have also engaged another practitioner to independently experiment with our research findings — a form of investigator triangulation; finally, we have used multiple theoretical "lenses" from various disciplines, both software engineering and socio-psychological ones, in our search for an appropriate definition of mentality patterns and in order to provide the theoretical foundations of the mentality innovation sub-process.

We have also used *peer debriefing* in our work. In all the organizations in which we have been employed during the course of this research, we have extensively and repeatedly discussed the mentality matter and our research findings with peers, usually one per organization. These peers fit Lincoln and Guba's criteria for the role: *"the debriefer should be neither junior – lest his or her inputs are disregarded – nor senior – lest his or her inputs be considered as mandates, or lest the inquirer "hold back" for fear of being judged incompetent [...] the debriefer should not be someone in an authority relationship to the inquirer"*. In all cases, the debriefers have been colleagues that were members of different departments in the organizational hierarchy. Moreover, we rarely had interactions with them as part of our job assignments, i.e., they were disinterested peers. On the other hand, we cannot claim that they had significant expertise in the area of inquiry. Nevertheless, their comments and insights have been extremely useful. Finally, we consider as a form of peer debriefing the fact that we have frequently discussed our research with several colleagues in academia and industry during conferences and workshops we have attended and as part of our professional life engagements in general.

Negative case analysis has been used in the form of making, evaluating and refining or even abandoning different hypotheses and alternatives. In Chapter 3 we provided some examples of our practice in that respect. This kind of retrospection was used considerably in our search for a systematic definition of the mentality pattern primitive as well as in the interpretation of the results presented in Chapter 6. For example, initially we had formed the hypothesis that a mentality pattern refers to cognitive but non-evaluative aspects only. Later, in the process of testing this hypothesis, we encountered recurring mentalities that had a more affective character, i.e. attitudes. Consequently, we had to adapt the mentality patterns definition accordingly.

In what concerns *referential adequacy* some of the data we have gathered is of rigorous and archivable form, for example, through the use of notes from projects or meetings, discussions with colleagues, and the interviews conducted as part of the survey presented in Chapter 6. We periodically reviewed and processed these in order to formulate, reflect on and validate our findings and to steer the next steps of our research. On the other hand, some of the data gathered has been of a transient form. To illustrate the point, consider the case of a project manager who told us: "*I am fed up with this 'us vs. the business' mentality of some people in here"*. We could have continued the conversation for some additional time to further understand the context, i.e., what project, which people, for what reason, and so forth. However, we were not always able to follow up on such conversations due to the constraints of real-life environments. Sometimes, we had the opportunity to transcribe parts of such conversations on paper so that we could follow them up later. In some other occasions, we did not have time to put those into written form, so we used them as hints for subsequent inquiry on the matter. Following the congruence of our research approach to the one employed by Cockburn, as discussed in Chapter 3, data gathering also shared common elements. In [15], Cockburn provides a good synopsis on the matter:

> *"The characteristics of the (research) opportunity carried with them implications about the quality and archivability of the information gathered. In the worst cases, I had only my memories to replay when a new theory was brought into view. Where possible, I at least created a shared memory with another person through explicit discussions about what was happening at the time and the theories that might reflect on it. This allowed me to check my memory later with someone who had been there at the time. In the best cases, I had written artifacts on file that coincided with the theory being tested. The news is, of course, good and bad. The bad news is that large parts of the ongoing store of research information are in my own memories and my shared memories with other people. The good news is that I was able to make use of more information than otherwise would have been possible. The factor that mitigates reliance on unreliable data is that, however the theories were generated, they were tested for effectiveness on live projects, as well as for predictive value while listening to accounts of project life."*

Finally, we have also applied the fifth technique of ensuring validity of research, namely *member checking*. We have regularly discussed our findings and interpretations with colleagues in our research (and, essentially, employment) environment. Those discussions were mostly of an informal nature. During such engagements, controversial points were raised and clarified. In general, our fellows provided useful input not only on the research itself but also on ways to improve our practices in projects and teams, e.g., on how to deal with particular people. The results presented in Chapter 6 on both mentality patterns and the observable change achieved with the mentality innovation sub-process are characteristic cases of outcomes derived from our member-checking practice.

Dependability

The dependability criterion refers to reliability in quantitative research, i.e., if the work were to be repeated it would produce the same results. Techniques such as triangulation of research methods, replication of steps and inquiry audits are often used to address this issue. We have discussed triangulation and the way it has been applied in our work in the validity section above. In what concerns replication, in [73] the author proposes that *"in order for the future researcher to repeat the work, if not necessarily to gain the same results [...] the research report should include sections devoted to the research design and its implementation, describing what was planned and executed on a strategic level; the operational detail of data gathering; reflective appraisal of the project"*. Finally "inquiry audits" refers to employing others to follow the decision-making process ("decision trail") of the research.

In what concerns replication, we believe that our thesis provides plenty of detail on research planning, operationalization, results and respective assessment. On the other hand, we have not employed any colleague in order to perform an inquiry audit. This would have been a quite complicated matter given the constraints, e.g. privacy and confidentiality aspects, of the corporate environments where our work was conducted.

Transferability

Transferability is the equivalent of external validity in quantitative research. It refers to the extent to which the results of a research can be transferred to another similar situation or participants. In other words it refers to the problem of the generalization of the research.

As discussed in [53], the problem of generalization is controversial in the scientific community. Some criticize the concept in general, whereas different viewpoints exist even between advocates of the idea, in both quantitative and qualitative research. Lincoln and Guba argue that all scientific findings are context specific and that *"the only generalization is that there is no generalization"*. They suggest that transferability depends on the degree of similarity between two contexts and they use the term “fittingness” to determine such similarity. Consequently, they also argue that the original context must be described accurately and in detail, an activity called “thick description”, so as to determine transferability. However, more recent standpoints adopt a more relaxed approach. Sandelowski [68] states that a *“study meets the criterion of fittingness when its findings can ‘fit’ into contexts outside the study situation and when its audience views its findings as meaningful and applicable in terms of their own experiences”*. Williams puts forward in [82] the notion of “moderatum” generalization as a form of generalization in which aspects of a general statement can be seen to be instances of a broader recognizable set of features. Adopting such notions of generalization, the results of our work on both the definition of mentality patterns and the mentality innovation sub-process are not universal, probabilistic or statistically proven laws. However, they can be considered general statements and rules. Rules have exceptions though. In that sense, our findings will be further tested by other colleagues in subsequent studies. One preliminary form of such a test, reported in Section 6.3, has been very encouraging in that respect.

Confirmability

Confirmability corresponds to objectivity in quantitative research. Confirmability is a way to ensure neutrality in the sense that the results are not the outcome of biases and subjectivity of the researcher. Similarly to the dependability criterion above, Lincoln and Guba refer to “audits” as a technique for ensuring confirmability. However, they also indicate that confirmability is established when credibility, transferability and dependability have been achieved. Given that we have already referred to the audit matter and discussed the ways credibility, transferability and dependability have been established in our work, in what follows we discuss and present our position only with respect to researcher bias.

Researcher Bias

In any research, an important issue is the potential personal bias of the researcher in what concerns aspects such as the way research approaches have been applied or data has been interpreted. This tension can be expressed as “what you want is what you see Vs what you see is what you want”. The biased approach is the former: the researcher’s predisposition to verify their predetermined belief or hypothesis. However, in our case, the particular research goals and the way research has been conducted have been vital for avoiding, to a large extent, such bias: we have not tried to confirm a predetermined belief, namely the existence of mentality patterns, or to illustrate the positive outcomes of a pre-existing methodology. Instead, the definition and representation of mentality patterns as well as the innovation sub-process have emerged during this research — those had been our concrete goals in the first place. Neither are we comparing those to other similar approaches in order to prove that they are better. In other words, we are not trying to enforce or confirm our own predetermined hypotheses, but rather to develop means that can be useful in practice. Evidently, during the course of the research, we had our own ideas and had to make several concrete choices. For example, initially we had the idea of using a graphical notation for representing mentality patterns. This idea was later abandoned. Given that our aim was not to conduct research that compares representations of mentality patterns, we do not consider this as a bias against graphical notations. The same statement applies to all other methods that we had thought of using at different times over the course of this work.

Having said this, we believe that one of the most important contributions of our work is that it provides significant prospects, across several dimensions, for further research on the mentality matter in software engineering. We discuss such opportunities in the following chapter.

Chapter 9

RECOMMENDATIONS FOR FURTHER WORK

9.1 On Mentality Patterns

The existence of a systematic and consistent definition and representation of recurring mentalities in SE using the notion of Mentality Pattern provides a wide horizon of research opportunities on examining the substance of the phenomenon itself[41] or at least for some of its recognized manifestations. Given that such an examination intersects with the focus of social and psychological disciplines, it seems that this is an interesting topic for interdisciplinary research. In what follows we discuss the main opportunities for further work across several axes.

On the existence of mentalities/mentality patterns

In this thesis, we have provided sufficient evidence on the use, explicit or implicit, of the term "mentality" as a factor that impacts software engineering practice. For this purpose, we have used several sources: the scientific literature, relevant comments in the wider community, and input from several of our own colleagues over the years, including the survey results reported in Chapter 6. Still, the phenomenon has not been "formally" confirmed, for instance by investigating large populations. Performing such larger-scale, possibly quantitative, studies would shed more light in various dimensions, thus allowing us to consider the matter in an even more efficient manner. Indicative questions that deserve further research are the following:

- Which recurring mentalities are specific to SE and to what parts of it? For example, examining whether *"testers have a siege mentality"* as stated in [42].
- Which mentalities also exist in other domains and what do such domains share

[41] This statement assumes a wider adoption of the results of this thesis. However, it would be inappropriate to explicitly recommend such adoption. Nevertheless, we believe that any other, widely-adopted, means that provide a systematic and consistent definition and representation of SE mentalities can potentially lead to the same research directions.

with SE in that respect? A starting point could be “affiliated” domains such as electrical engineering.

- Are there mentality patterns that are more frequently encountered? For instance, the ones that we have relevant indications in our results i.e. “Fear to Admit Ignorance”, “Us and Them”, “Subject Guru” and “Not Invented Here”. Moreover, are any of those more prevalent in certain cultures, types of organizations (e.g. small Vs large), application domains, technology domains and so on and so forth?
- What is the relationship between patterns that are more frequently encountered and their respective impact in practice? Is it the case, as indicated by the results of our survey in Chapter 6, that the patterns most frequently found are also the ones that are perceived to have the most decisive impact?

On the causes and influences of mentality patterns

The second area of further research concerns the determination of the factors that either cause or influence the emergence of given patterns. The results we presented in Chapter 6 indicate that several colleagues tend to associate specific patterns to the character of particular individuals and organizational or team cultures.

In that respect, the three themes of interest of human aspects of software engineers identified in Chapter 2 can be used in order to provide a categorization: factors associated with certain characteristics of human individuals, team-related factors and managerial/organizational ones. The examples one can think of are numerous – to illustrate the case we refer to some of these below.

On the human individual dimension, it is interesting to investigate the relationship between personality and mentality patterns. In other words, whether there exist certain personality characteristics that cause given patterns. For example, in [47], the authors identify “personas” that they call “negativists”. We have also identified a mentality pattern which we call “negativism” and our results also indicate that colleagues often associate certain patterns to the character of individuals. However, how the two are related is unclear. For instance, there has been no answer to the following questions: do particular types of personalities exist whose mentality is manifested via corresponding patterns? If so, what types of such “personas” can be found and which are their corresponding mentalities?

Along the same line, a second factor that deserves investigation is the relationship between psychological tendencies and mentality patterns. For example, according to the Confirmation Bias theory, people have a tendency to search for and interpret evidence in a biased way so as to reinforce their beliefs, hypotheses or decisions. This can lead to attitude polarization, a case in which the opposing views of different parties become more extreme as they consider evidence on the issue of disagreement. In our view, this polarization can provide a justification for the Technology-Fundamentalism mentality pattern. However, this is merely a subjective view. Therefore, further research is needed in order to establish such relationships.

A third factor that needs further investigation concerns the relationship of recurring mentalities to age and professional experience. In our experience, there exist signs that as age and experience increase, new or different mentality patterns emerge. For instance, the “Experience-Driven Optimism”, “Subject Guru” and “Opportunistic Listening” are examples of mentality patterns that usually appear after some years of experience.

Finally, several sets of factors that cause or influence certain mentalities can be associated with the team and managerial/organizational dimensions. For example, in the former case, we can consider teams’ dynamics and the different organization, composition and collaboration models. In the latter case, we can consider organizational cultures, management styles, software processes, inter-alia. It is evident that the list of such factors is quite large and therefore providing more concrete examples serves little point. We believe that the above discussion illustrates that the absence of definition and systematization in what concerns recurring mentalities in SE has resulted in a shortage of research that looks into the matter in a comprehensive way. Apart from giving strength to the relevance of this thesis, this observation guides us to propose the intensification of research across the different dimensions of the matter such as the ones above, having as ultimate goal the development of a pattern language, in C.Alexander's terms [4], for mentality patterns in SE.

Towards a pattern language for mentality elements in SE

In Chapter 4, we justified the reasons for why in the mentality patterns representation there exists no element dedicated to capturing actions, recommendations or solutions for dealing with given mentality patterns. The essence of our argument has been the lack of relevant studies that would provide a solid, instead of subjective, basis for making such recommendations. The discussion above exemplifies the lack of concrete results in the various factors that may cause or influence given mentality patterns. Needless to say, that given the absence of both, there is also no way to appropriately target proposed actions or solutions with respect to the concrete underlying causes of given patterns. In this thesis we have provided a means to capture and systematize the insights and knowledge of individuals on the manifestations of a number of mentalities in SE. However, we did not aim to provide content on such patterns: the concrete examples we offer are only illustrative and are based on experience that is small compared to the one possessed in the wider SE community. We believe that, as the field gradually matures and such content becomes available, a pattern language for such recurring mentalities in SE can be established. A case that further amplifies this need is the mentality-patterns interference phenomenon which we have identified through our work.

Evidently, the mentality pattern primitive and other results of this thesis may have to be adapted in order to account for such developments. We already have some preliminary ideas in that respect, for example, introducing a separation between type and instances of a given mentality pattern. The former could be used to capture the essence of the pattern, while the instances could be applied to deal with different pairs of the form "cause-solutions" within appropriately defined contexts. Along the same lines, additional research directions that could generate useful outcomes are:

- The establishment of an ontology and formal representation for mentality patterns, for instance using the Web Ontology Language [75].

- The evaluation of the mentality pattern representation in what concerns a number of attributes, e.g. “learnability”, “memorability” and user-satisfaction, that are common in the Ergonomics, Usability and Conceptual modelling disciplines [32]. Such an evaluation could be performed in both dimensions of creating new patterns and of understanding existing ones.

Overall, we are aware that all the above constitute an ambitious goal, but we are convinced that it is worth pursuing.

9.2 On the Mentality Innovation Sub-Process

On subsequent applications of the sub-process

In Chapter 6 we have presented the results obtained on the applicability and capacity of the sub-process to drive change in teams. However, our work did not aim to provide statistically sound results. A reasonable next step is to perform studies that would allow for statistical analysis of the sub-process practices and outcomes to be obtained. For example, by using more purposive and bigger samples, such studies can provide quantitative results on the applicability of the sub-process as well as the concrete mentality patterns for which it drives change and the ones that tend to persist.

Moreover, we have also identified another direction for future work, namely the application of the sub-process using “controlled-variables” research approaches, e.g. controlled experiments. Such studies can examine various aspects, some of which we outline below:

- Examine the way the different choices made during the operationalization of the sub-process impact its outcomes and make respective recommendations for relevant adaptations and improvements.
- Explore the relationship between various teams’ characteristics such as composition, dynamics and cohesion, and the applicability and outcomes of the sub-process.

- Investigate the effects of different personality traits and skills of the sub-process driver in order to make suggestions on what are the types of persons more suitable for assuming this role.
- Examine the ability of the sub-process or any other means to drive “change” in patterns that are perceived to be associated with the character of particular individuals and the organizational or team culture. We outlined the case in Chapter 6. Moreover, this research direction appears to be related to the investigation of the concrete causes and influences of mentality patterns, a matter that we referred to in Section 9.1.

Establishing the sub-process at an organizational level

In our work we have adopted a team-oriented approach in what concerns the modus operandi of the Mentality Innovation Sub-Process. It would be interesting to also experiment with the sub-process defined at an organizational level. To illustrate the scenario, an organization could identify a set of mentality patterns, formalize and communicate the respective operating principles as applicable to the whole organization, and prescribe organizational procedures for feedback and learning. Such a scenario is entirely different from our view of the mentality innovation sub-process, in which we advocate the active participation of individuals and teams in the development and commitment on the mentality-related operating principles. The implications and results due to the differences of the two designs are unknown, for example, whether people would ignore or view as overhead processes and principles defined at an organizational level. We believe that such a scenario is worth exploring in order to generate results that can be compared with our approach.

9.3 On the support system and repository

Given our goal for the support system as a knowledge and experience sharing means, we believe that making concrete recommendations for further work on it would be immature before its current version is more widely used in practice. We do not claim that the implementation of the support system is comprehensive — on the contrary: a broader usage would reveal functional limitations or other constraints of its implementation. Moreover it would provide the opportunity to assess aspects such as correctness, usability, performance, inter-alia. Having established these, we could subsequently make more useful recommendations on the evolution of the support system.

Appendix A

Survey on Practitioners’ Views

1. Interview Questions

GENERAL INFO

Name:
Years of experience in the IT industry:
Number of different organizations:
Main roles assumed:
Current role:

QUESTIONS

On the Existence and Impact of Mentality Patterns

1. Can you comment on the existence of Mentality Patterns in practice? (e.g. do they exist only occasionally, frequently, a lot?)
2. Which of the patterns identified are the ones which you have encountered more frequently?
3. Which of the patterns identified are the ones you believe that have the most decisive impact in practice?
4. Can you provide additional examples/cases of mentality patterns you have encountered?
5. Do you think that the Mentality Patterns issue influences S/W practice and, if so, in which respect and to what degree?

On Current Practice

6. Do you take into account the mentality patterns issue in your current work practice, even if not explicitly? (e.g. on team management, on individual coaching or on tasks assignment)
    a. If yes, can you elaborate on the means that you use?
    b. If no, do you think that it would be helpful to do so?

On the Mentality Innovation sub-process

7. How do you evaluate the degree to which the sub-process could be applied in practice? (e.g. easy to apply, relatively easy to apply, neither easy/nor difficult, relatively difficult to apply, difficult to apply)
8. Do you think that the process is beneficial in dealing explicitly with the mentality issue? If yes, in what respect?

Concluding Questions

9. Would you consider applying the primitive and/or sub-process in your work environment?

10. Is there anything we have not covered that you would like to refer to?

2. Participants' "Profiles"

Interviewee No	Years of Experience	No of Organizations	Main Roles	Current Role
1	*17*	*2*	*Developer, Team Leader, Project Manager*	*Business Architect*
2	*20*	*4*	*Developer, Researcher, Architect, Vendor Management*	*IT Strategy*
3	*25*	*4*	*System Analyst, Project Manager, Consultant, QA Manager*	*Quality Control and Methods Manager*
3	*8*	*3*	*Project Manager, IT Auditor*	*IT Risk Professional*
5	*17*	*3*	*Developer, Team Leader, Project Manager*	*Business-IT Relationship Manager*
6	*14*	*2*	*Developer, Project Manager*	*Analyst*
7	*15*	*6*	*Developer, Project Manager, Designer/Architect*	*Demand Management and Project Planning*
8	*9*	*4*	*Developer, Business Analyst, Project Manager, IT Strategy Analyst*	*Demand Management and Project Planning*
9	*7*	*3*	*Software Developer, Team Leader*	*Software Developer, Team Leader*
10	*10*	*1*	*Software Developer, Team Leader, Analyst*	*Analyst*
11	*12*	*5*	*Software Developer, Team Leader, Architect*	*Architect*
12	*14*	*4*	*Software Developer, Team Leader, Architect*	*Architect*

3. Survey Answers

Q1. Can you comment on the existence of Mentality Patterns in practice? (e.g. do they exist only occasionally, frequently, a lot?)

Int/wee No	Answers
1	*They exist a lot*
2	*They are prevalent through IT industry, though many of them are not specific to IT.*
3	*In general mentality patterns in industry are endemic. Potentially they can affect any team or organization. But, mature organizations have mechanisms, explicit or implicit, to mitigate the negative effects coming from the exaggerations of these patterns. For example, how people are promoted to management positions in the sense that managers should have the capabilities to mitigate such patterns. Also, by nets of checks and balances e.g. by process or role assignment to specific nodes in the process. Its not only within a given function but also across functions in an organization. Its also a trend in Management Science to find appropriate organizational models with self-healing capabilities. All human individuals and personalities are susceptible to such patterns. Whether they are going to surface it's a matter of external conditions. In summary, in mature organizations such patterns are considerable less visible in terms of their effects.*
4	*They exist frequently*
5	*They exist a lot*
6	*I believe they exist a lot. Either as developer or Project Manager I always had those in mind since they characterize certain people*
7	*They exist a lot. I found the representation quite clear and expressive, particularly the sections on "Representative Quotes" and "Consequences" because not only you identify given mentalities but you also go further to identify concrete consequences. It helps a lot on the maturity of a team in terms of reflecting on the consequences of particular mentalities. In summary, the representation is concise.*
8	*They exist a lot. In all projects that I have participated such patterns existed, although at that time I was not able to recognize and characterize the pattern. I just realized the behavior.*
9	*They exist a lot. They are encountered in many projects and occasions. Particularly in early phases of the project, e.g. requirements and technology options. When requirements are not clear enough, then some mentality patterns emerge since IT Managers are trying to satisfy the customers and their own status.*
10	*I would say frequently. You find them often, but not always and not in everybody*
11	*Yes they exist*
12	*They exist a lot*

Q2. Which of the patterns identified are the ones which you have encountered more frequently?

Int/wee No	Answers
1	*All of them exist. I would say more frequent are "Fear to Admit Ignorance", "Legacy Person", "Us and Them", "Its not my Fault". In the past I was also witnessing frequently the "No Coding = Useless" one. I believe its less frequent now due to the fact that we now mostly care about integration and solution evaluation rather than coding.*
2	*"Have the right to make assumptions", "The Best is the One I am Comfortable With", "Fear to Admit Ignorance", "Not Invented Here"*
3	*"Not Invented Here", "The Best is the One I am Comfortable With", "Negativism"*
4	*The ones that typically has as impact (negative) in communication and teamwork, e.g. "Not Invented Here", "The Best is the One I am Comfortable With", "Us and Them"*
5	*"Subject Guru", "Us and Them", "Negativism", "Not Invented Here"*
6	*"Fear to Admit Ignorance", "Better is the Enemy of Good", "The Best is the One I am Comfortable With", "Negativism", "Secretivism"*
7	*"Fear to Admit Ignorance", "Subject Guru", "Us and Them" (a lot), "Legacy Mentality", "Opportunistic Listening", "No Coding = Useless"*
8	*"Not Invented Here" is very common. Also "No Coding = Useless" particularly in teams with junior members, skilled people but not interested in anything else than writing code.*
9	*"Experience Driven Optimism", "Legacy Person", "Its not my Fault", "Secretivism", "Fear to Admit Ignorance",*
10	*"It's not my fault", "Subject Guru", "Us and Them", "No Coding = Useless"— the last a lot.*
11	*"Not Invented Here", "Subject Guru", "Experience Driven Optimism" (the last one is not only related to past experience but is also a tendency of developers)*
12	*"Not Invented Here", "Fear to Admit Ignorance",*

Q3. Which of the patterns identified are the ones you believe that have the most decisive impact in practice?

Int/wee No	Answers
1	*"The Best is the one I am Comfortable with" and "Legacy Person" because they can lead to wrong decisions. Also "Us and Them" has impact on projects due to the issues it brings in collaboration.*
2	*"Have the right to make assumptions", "The Best is the One I am Comfortable With", "Not Invented Here"*
3	*The ones that are encountered more frequently also have the most decisive impact. I could also add "Us and Them" that has very significant impact and "Fear To Admit Ignorance", though they are less encountered.*
3	*Some have serious consequences, e.g. "Negativism", "Fear To Admit Ignorance" (e.g. on wrong planning due to the false assumptions on particular team members knowledge)*
5	*"Us and Them" has the most decisive impact, "Negativism" also.*
6	*"Fear To Admit Ignorance", "Not Invented Here", "Have the right to make assumptions", "Us and Them"*
7	*"Fear To Admit Ignorance", "Subject Guru", "Us and Them"*
8	*"Fear To Admit Ignorance" because particularly in being a Project Manager it does not allow for early identification of risks. "Not Invented Here" has also a decisive impact because it can be counter-productive since you do not use already available products (in a wide sense)*
9	*"Fear To Admit Ignorance", "Better is the enemy of good", "The Best is the One I am Comfortable With". Particularly the last one is "dangerous" in decision making in teams because it can lead to problematic choices. There is also a pattern called "Technology Fundamentalism" which can be related to the "The Best is the One I am Comfortable With". The relationship is not always precise, it depends. Other relationships exist: "Fear to Admit Ignorance" leads to "Its not my fault", but as I said earlier its not always straightforward to establish such relationships.*
10	*All have impact potentially; it depends on the particular context.*
11	*"Not Invented Here", "Experience Driven Optimism"*
12	*"Fear to Admit Ignorance", "Subject Guru"*

Q4. Can you provide additional examples/cases of mentality patterns you have encountered?

Int/wee No	Answers
1	*The mentality of some people that new technologies should be early adopted before evaluating their maturity – "Technology Freak"*
2	*"Better than Average Assumption", "Own experience thought of others", i.e. believing that others have the same experience, thoughts, understanding with us.*
3	*I believe the list is comprehensive in the sense that any other similar psychological attributes can somehow traced back to the list.*
4	*"Its not my responsibility" i.e. not being helpful, not facilitating work using ownership of responsibility as an excuse.*
5	*There are variations and combinations that belong to more than one patterns, e.g. I will do it my own way because I am the Subject Guru. The are additional ones of people that are afraid to make decisions due to some sort of fear, "Fear to Decide", "Fear to Become Unpleasant". I believe this is due to insecurity. Moreover, there exists a "Process/Procedural" mentality. In other words, people that are inflexible and not pragmatic in taking action when there is no procedure or formality for this action, i.e. it is not prescribed in a process.*
6	*When somebody views criticism as a personal insult instead of a constructive comment for improvement.*
7	*Not really, I cant think of something*
8	*"Fear of Delegation", the tendency to do it myself. It can be a consequence of Subject Guru – somebody that considers himself as expert tends to concentrate workload and in turn becomes a bottleneck.*
9	*I liked the representation of patterns in general. Sometimes you exhibit such patterns deliberately to achieve other goals. In some other cases they are unconscious. I have doubts over the existence of the "No Coding = Useless" pattern, its rather a preference on the type of work you want to do.*
10	*I was thinking of some while reading the background info you sent. In fact, that's precisely what I was trying to do, but I can't remember something right now. In general terms the list is comprehensive.*
11	*I can't think of something right now*
12	*The list is quite comprehensive, I can't think of something right now.*

Q5. Do you think that the Mentality Patterns issue influences S/W practice and, if so, in which respect and to what degree?

Int/wee No	Answers
1	*Yes it does: in collaboration, evaluating solutions, can lead to mistakes that could have impact in projects, usually to a large degree*
2	*It can influence quality of results, collaborations aspects and in some cases in planning and directions in the sense that if certain patterns are not identified, the focus can be diverted.*
3	*Already answered in Q1*
4	*Yes they influence significantly S/W projects due to the nature of S/W development. Since S/W nowadays is performed in large scale (in the large) the required interfacing at the systems' level depends a lot on the interfacing and communication ability of individuals and teams responsible for individual technical components. The paradox that I see is that while we have advanced tools and more automation in S/W development, i.e. the impact of the human element should have been less, it appears that it remains constant.*
5	*Yes definitely. Depending on the mix of personalities and patterns the impact can be 20% or even 80%. You can have a mix that can be catastrophic. Even worse, if you do not have a competent project manager that is able to "absorb" and resolve the results of mentality conflicts, then you have a major impact. I do not think the mentality matter is personality related in general. You may have a person that you get well with in personal terms, but still at works he exhibits some of such patterns.*
6	*Yes definitely: in teams' interaction and in results in given tasks. For example, the "Experience Driven Optimism" case leads to underestimating of effort for certain tasks and in other elements e.g. in paying attention or thinking on the solution instead of listening to the requirements of business.*
7	*In different organizations there are variations on the way such patterns influence practice and on the respective degree. It is clear that they influence as any human related aspect such as personality (though I do not think there exists a one to one relationship between personality and patterns – two different people in terms of personality can both be "Subject Gurus"). The other point is that they are not specific to SE but to any other domain and type of job. In some cases, the consequences are not negative only. It depends on who the particular person is.*
8	*Definitely yes. Main impact is on the quality of S/W delivered and respective delays e.g. when a risk is identified late as the result of the "Fear to Admit Ignorance" pattern, this leads to delays.*
9	*It is evident that it influences a lot in the development of the people themselves, on project plans and deliverables and on the well being in the work environment.*
10	*The ones you encounter more frequently may not directly have impact on the outcome of tasks, but they do have impact on the overall team environment. There are some others, e.g. "Fear to Admit Ignorance" and "It Works (but I do not know why)" that do have impact on work outcomes.*
11	*Yes they influence in general. Sometimes they can have impact on the time needed to complete a project rather than on the quality of the outcome. It's just an overhead to deal with them; eventually you manage to deliver the quality you intended in the first*

	place. At the same time, the pattern interference is also real in practice.
12	*Yes, they influence. Particularly when one does not have sound methodological practices in place. When work is based on "personal" effort, they influence more.*

Q6. Do you take into account the mentality patterns issue in your current work practice, even if not explicitly? (e.g. on team management, on individual coaching or on tasks assignment)

a. If yes, can you elaborate on the means that you use?
b. If no, do you think that it would be helpful to do so?

Int/wee No	Answers
1	*Yes. Based on my understanding of certain mentalities of given people I adapt accordingly on the way I discuss with them. In general, based on my impression on particular mentalities e.g. "Legacy Person", I am influenced on the level of trust I can have in particular persons. Therefore, in such cases, I am trying to validate with other colleagues if what a person promises can be achieved. In other cases, e.g. "Optimism", I will adapt plans and increase the time allocated by the person to perform a task so to account for the "Optimism" mentality.*
2	*Not in a clear way. Tendencies and behaviours are identified but not explicitly. I can adjust by knowing that they are there; try to understand their possible impact. Evidently it would be helpful.*
3	*In my view, the best ways to influence/mitigate such patterns are the following directions. Do not use your personal judgment that a pattern exists in order to give direction or justify a position – be objective, specific on the content. Do not use such patterns as arguments openly to support your view. If you use them openly in this way, you trigger psychological defense on your peers leading to detaches from focus on the collaboration subject. The matter is delicate and can not be easily addressed on a person-to-person level. "Open the circle" of participants when you see that something is "blocked" due to a certain actor that is susceptible to a given pattern, i.e. ask the opinion of somebody else, another actor – in some cases he would be more detached and unbiased. This creates, ad-hoc, as needed "check and balances" even when they are not in place at organizational level.*
4	*I do, by employing more personal communication and team-building. Also you try to create some "flags" on certain mentalities of persons, which can potentially lead to issues, and manage them accordingly. Those are at two levels: not doing properly tasks they are supposed to do; and influence negatively the rest of the team and the overall environment.*
5	*Always. First thing I do is to understand what kind of people I have to deal with in what concerns such patterns. This is a first risk factor and therefore you have to adapt your practices and behaviour accordingly to account for them.*
6	*As a risk for the result of the project I am trying to mitigate them per case. For example, if I know that somebody exhibits "Negativism", I think of arguments on how to convince him proactively. In the case of "Secretivism", I can even pretend that I do not know something in order to convince the respective person to reveal more information.*
7	*Definitely yes – even when hiring somebody, e.g. looking on particular characteristics. Then, on deciding on the mix of people in projects, even in meetings. You always want to have a balance. For instance you can use other people with opposite characteristics to achieve such balance. You do all this empirically. Usually you do not distinguish between personality and pattern in this practice. There is definitely interference between those; however it is not clear whether it is personality related, pattern related or mixed. Nevertheless, the mentality pattern is a useful abstraction mechanism in separating particular ways of thinking to the underlying personalities. Thus you can put emphasis*

	on specific thinking rather than the attributes of a specific personality.
8	*I try to recognize them in the individuals or teams that I have to collaborate and I tend to adapt my own way of performing tasks (in that context) according to such patterns. In general, I do not try to change them but I rather adapt myself, particularly in the cases that I am not the team leader. If I am the team leader, I set ground-rules, not collectively, in the sense of operating principles.*
9	*Not very much to be honest. I do not take them that much into account proactively. When encountered, obviously, I am trying to take advantage or resolve them. One way is to assign certain tasks to certain people.*
10	*Unconsciously I do. For instance when you have to work with somebody that you have in your mind a specific pattern he may exhibit. It is an empirical process. Generally, I do not do something in specific, at least proactively, but one is "prepared" in that sense. For example, if you have someone that operates in an "It Works (but I do not know why)" manner, then I may insist in having him show the details to me.*
11	*For some patterns that I am aware of I will try not to fall in the "trap" of acting according to that pattern, e.g. the "Not Invented Here" or "Subject Guru". For team leading, I would try to guide the team not to fall in such "traps" e.g. not to build something that already exists.*
12	*I try to recognize in people I have to work with the existence of such patterns in order to judge if they facilitate or are obstacles in goals attainment. In the latter case, I try to find ways to overcome or minimize the effects. I do that empirically per case. For example, if somebody operates as a "Subject Guru", it does not make sense to "polarize" or directly dispute his judgment or knowledge. Instead, you have to establish credibility because such people respect peers that they consider equal in terms of knowledge, and then lead them towards believing that they are the ones that actually provided the solution. I also believe there is interference between patterns. The simultaneous manifestation of multiple patterns might create a new, collective, pattern. Two "Subject Gurus" particularly if they disagree can be an "explosive" composition.*

Q7. How do you evaluate the degree to which the sub-process could be applied in practice? (e.g. easy to apply, relatively easy to apply, neither easy/nor difficult, relatively difficult to apply, difficult to apply)

Int/wee No	Answers
1	*In general, it is neither easy, nor difficult. One could say that, in principle, it would be easy (e.g. you do not need any particular tools), but in practice it would be difficult to convince people in participating in such a process because the matter is delicate although we all know about it. Probably, one needs an iterative approach to convince and commit people in engaging in such a practice. It is a long process.*
2	*To a certain degree it is relatively easy to apply. The "team building" part appears to be easier particularly in environments with "tradition" to have "mentoring" or "people oriented" training. The feedback part is more difficult because of the delicacy of the matter and since comments and remarks can pertain to specific people and actions.*
3	*Not answered.*
4	*In general terms it can be applied relatively easy. The issue is that people are not used to such approaches – it seems distant to practices we are used to. On the other hand, it does not take a lot of time. I have concerns over the cyclic nature of it in what concerns feedback. After some iteration, people stop providing valuable feedback and input. I consider this to be the greatest challenge and risk of the method.*
5	*I would say it is relatively difficult to apply it. I have doubts over the exact operationalization of the process.*
6	*Neither easy/nor difficult. It depends on the particularities of a given organization. If the drivers are quality and continuous innovation and improvement then it can be applied and is useful to be applied. If we are constantly under constraints such as time, then it is difficult to apply it.*
7	*I have doubts on how the sub-process can be applied in practice. Particularly whether the "team" nature of it can work effectively or if it is better to have one-to-one engagements by the team leader.*
8	*Neither easy/nor difficult if applied in an "informal" manner. If it is to be enforced or "formalized" it would be difficult to be applied as people may become defensive and introvert.*
9	*It depends on the key person that can orchestrate this process. A good manager could have this sub-process in his "arsenal" for team leading. I do not think that people will be reluctant to participate – in some sense it is a sort of collective "psychotherapy".*
10	*It is a good idea, but it requires an "innovation spirit" that is not easy to find. Sometimes, it is also difficult due to other constraints, for example workload. It may start, but in the process it may be "abandoned" due to various circumstances.*
11	*It appears to be too "formal". Therefore, something that could be more easily incorporated in the workflow of a team/project may be more easily applicable.*
12	*Yes, it can be applied. In some sense it is already applied by some managers implicitly. That is actually the reason that managers typically want to have "stable" team members so that they can manage accordingly respective mentality patterns.*

Q8. Do you think that the process is beneficial in dealing explicitly with the mentality issue? If yes, in what respect?

Int/wee No	Answers
1	*I believe it is beneficial because by recognizing such patterns (the first important aspect) you can reflect on your own behaviour and avoid thinking or acting in respective ways. If everybody has committed to considering those and operating according to some principles, it is easier to accept feedback on the matter. At the same time, it is also easier for others to provide feedback to particular individuals.*
2	*Yes. It identifies a very critical but overlooked factor for the success of projects: the human factor. The principles of the method apply to any such effort of incorporating mentality pattern in practice*
3	*Not answered.*
4	*Yes. It defines explicitly a "horizontal" layer between software project practices and work environment practices. It is a connecting means between the particularities of a work environment and the specific tasks of a given team and the team's human personalities. With the process (e.g. the manifesto) people adapt themselves in order to achieve amore coherent team behaviour independently of the project at hand. The manifesto should be "intelligent", clearly specified in order to trigger "positive" mentality.*
5	*It would be beneficial if one can find a way to "operationalize it"*
6	*Yes, because you proactively consider the risks associated to Mentality Patterns. You communicate them early and their management is smoother during the project. Moreover, you make the teams more aware of those issues so that each can adapt accordingly. For example, if I realize that I operate on "Experience Driven Optimism" then in the cases that such a way of thinking comes into play (e.g. in capturing requirements) I will reconsider my attitude.*
7	*Maybe. I recognize the potential value. I would be willing to experiment and decide on subsequent action based on the results.*
8	*Yes, because often people attribute certain behaviours to the individuals' personalities and given that they consider this as invariable, they just tend to adapt themselves accordingly. However, such patterns are not necessarily the "result" of personalities only. Given this fact the method can help in changing such mentalities.*
9	*Obviously, it can be beneficial. Sometimes it can have effects that are unknown. In long-lasting teams is good. For teams that are together for shorter periods one can apply other, more opportunistic, techniques.*
10	*It also depends on the "innovation spirit"; and on the particular people. It may bring conflicts sometimes, but this is not necessarily bad. However, it requires personal commitment and reflection.*
11	*It depends on the way you apply it. If somebody really understands the issue, then it becomes beneficial. If you give emphasis on the substance and not on the process then people will reflect on such patterns.*
12	*Yes, I believe it would be beneficial to consider this explicitly, even in mentoring or coaching sessions, i.e. on an individual basis.*

Q9. Would you consider applying the primitive and/or sub-process in your work environment?

Int/wee No	Answers
1	*Yes, although in some form we all consider such patterns in our day-to-day, it would be beneficial to use a more "formal" and systematic way to do so. However, it is not easy to introduce such concepts in an organization.*
2	*I hope so. It needs some training and preparation in order to do so. Its not easy to feel comfortable in applying it.*
3	*The primitive can be used in a given case that one has identified predominant patterns. Its representation can be useful to support relevant proposals in mitigating them e.g. recommendations to management. A formal documentation of such patterns in a work environment can have conflicts in what HR practices prescribe. On the other hand you should alert such cases – the only way to alert is some sort of formality. It can also be useful input for HR consulting companies that engage in evaluation criteria and the HR departments of IT organizations.*
4	*I would apply the primitive as a means to recognize and capture the patterns in a particular organization or organizational division. The way to represent patterns is efficient for analysis and communication, but it can be challenging in case of significant numbers in the sense that convincing people to document them this way. In large projects I would apply the sub-process if I had the chance. In medium, smaller projects I would think of a lighter version.*
5	*I would apply the primitive and sub-process in the sense of principles and psychological foundations. The expressiveness and communication aspects of the primitive are very good, but oriented for IT people. I could find a way to apply the sub-process depending on the participants.*
6	*Yes, and its not only related to SE but in any case that involves teams and projects. I found the representation useful and complete and it is important that it gives specific examples. It's a good way to communicate things. I would use it if I was to apply it in an organization.*
7	*Yes, provided that I had a clear view on how to operationalize it.*
8	*Yes the sub-process but in a disguised way. I will try to incorporate it in the "day-to-day" practices of a team. I think the elements of the patterns representation are very correct. It has the symptoms that help recognize the pattern, it has the consequences so its helps to understand the impact in a team and it has the causes so that you can deal with the "problem" at its origins. The anecdotal story part is probably an overhead in understanding the pattern.*
9	*Yes I would consider. Probably not in a formal way. Maybe I would try to categorize some decisions and respective people in some of those patterns in order to justify actions and proactively prevent possible new actions.*
10	*I am not sure I could personally do it due to lack of confidence on my ability to drive it. It is difficult to convince everybody. However, being part of it sounds interesting, i.e. to participate in it.*
11	*Unconsciously a team leader does it in a certain extent, not in a checklist approach though. Such patterns are known in general but they are not explicitly identified the way you do it. The codification in a more "formal" way was missing.*
12	*I would consider applying both. I believe the framework for the definition and representation of such patterns is a solid one.*

Q10. Is there anything we have not covered that you would like to refer to?

Interviewee No	Answers
1	*No*
2	*No*
3	*No*
4	*No*
5	*No*
6	*No*
7	*No*
8	I would be very interested to see further developments in the field
9	*No*
10	I cannot think of something in specific. It is an interesting topic, you find those things everyday. It is an interesting way to capture them, sometimes you think who (person) could have these patterns in your work environment. It depends on the personality and the team culture on which pattern each individual exhibits.
11	*No*
12	*No.*

Appendix B

Extensions performed in Alfresco ECM

In order to extend the Alfresco Explorer and content model the relevant files containing the extensions have to be included in the Alfresco directory designated for this purpose. For Tomcat installations, the directory is located at $TOMCAT_HOME/shared/classes/alfresco/extension.

In order to provide the necessary extensions, four files had to be deployed.

- A custom model context file. This file instructs Alfresco to include our custom content model to the list of content models. The content of the custom model context file is provided below:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE beans PUBLIC '-//SPRING//DTD BEAN//EN'
'http://www.springframework.org/dtd/spring-beans.dtd'>


<beans>
<!-- Registration of the custom model -->
<bean id="extension.dictionaryBootstrap"
parent="dictionaryModelBootstrap" depends-on="dictionaryBootstrap">
<property name="models">
<list>
<value>alfresco/extension/gkModel.xml</value>
</list>
</property>
</bean>
</beans>
```

- The next step is to create a model file that implements our custom content model and name it “`gkModel.xml`”. The code for this file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Definition of custom model for mentality patterns -->
<model name="gk:gkmodel"
xmlns="http://www.alfresco.org/model/dictionary/1.0">


      <!-- Meta-data about the model -->
      <description>Mentality Patterns Model</description>
      <author>GK</author>
```

```
    <version>1.0</version>

    <!-- Imports: references to definitions in other models -->
    <imports>
        <!-- Import Alfresco Dictionary Definitions -->

    <importuri="http://www.alfresco.org/model/dictionary/1.0"
prefix="d" />
        <!-- Import Alfresco Content Domain Model Definitions --
>
        <importuri="http://www.alfresco.org/model/content/1.0"
prefix="cm" />
    </imports>

    <!-- New namespaces defined -->
    <namespaces>
        <namespaceuri="http://www.cs.le.ac.uk/model/gk"
prefix="gk" />
    </namespaces>

    <types>
        <!-- Content type for mentality patterns, properties,
associations and aspects (optional) -->
        <type name="gk:doc">
            <title>Mentality Patterns Description
Document</title>
            <parent>cm:content</parent>
            <properties>
        <property name="gk:publishedDate">
            <type>d:datetime</type>
        </property>
        <property name="gk:authorisedBy">
            <type>d:text</type>
        </property>
        </properties>
            <associations>
                <association name="gk:causes">
                    <title>Causes</title>
                    <source>
                        <mandatory>false</mandatory>
                        <many>true</many>
                    </source>
                    <target>
                        <class>gk:doc</class>
                        <mandatory>false</mandatory>
                        <many>true</many>
                    </target>
                </association>
                <association name="gk:triggers">
                    <title>Triggers</title>
                    <source>
                        <mandatory>false</mandatory>
                        <many>true</many>
                    </source>
                    <target>
                        <class>gk:doc</class>
                        <mandatory>false</mandatory>
                        <many>true</many>
                    </target>
                </association>

                <association name="gk:inConflict">
                    <title>In Conflict With</title>
                    <source>
                        <mandatory>false</mandatory>
                        <many>true</many>
                    </source>
                    <target>
                        <class>gk:doc</class>
                        <mandatory>false</mandatory>
                        <many>true</many>
                    </target>
                </association>

            </associations>
            <mandatory-aspects>
```

```
                  <aspect>cm:generalclassifiable</aspect>
                  </mandatory-aspects>
            </type>

      </types>

      <aspects>
            <aspect name="gk:projectRelated">
                  <title>Project Related</title>
                  <properties>
                        <property name="gk:project">
                              <type>d:text</type>
                              <mandatory>true</mandatory>
                        </property>
                        <property name="gk:organization">
                              <type>d:text</type>
                              <mandatory>true</mandatory>
                        </property>
                  </properties>
            </aspect>
      </aspects>
</model>
```

- The third file is concerned with the extensions to the user interface in order to work with the custom model previously defined. For this purpose the file named “`web-client-config-custom.xml`” should be modified accordingly. The content of this file in our case is shown below:

```
      <alfresco-config>

      <!--  addition of aspect properties in property sheet -->
      <config evaluator="aspect-name" condition="gk:projectRelated">
            <property-sheet>
                  <show-property name="gk:project" display-label-
id="project" />
                  <show-property name="gk:organization" display-
label-id="organization" read-only="true" />
```

```
            </property-sheet>
      </config>

      <!-- show related associations on doc property sheet -->
      <config evaluator="node-type" condition="gk:doc">
            <property-sheet>
                  <show-property name="gk:publishedDate" />
                  <show-property name="gk:authorisedBy" />
                  <show-association name="gk:causes" />
                  <show-association name="gk:triggers" />
                  <show-association name="gk:inConflict" />
            </property-sheet>
      </config>



      <!--  add content types to add content list -->
      <config evaluator="string-compare" condition="Content
Wizards">
            <content-types>
                  <type name="gk:doc" />
            </content-types>
      </config>

      <config evaluator="string-compare" condition="Action Wizards">
      <!-- The list of aspects to show in the add/remove features
action -->
            <aspects>
                  <aspect name="gk:projectRelated"/>
            </aspects>

            <!-- The list of types shown in the is-subtype condition
-->
            <subtypes>
                  <type name="gk:doc" />
            </subtypes>

            <!-- The list of content and/or folder types shown in
the specialise-type action -->
```

```
            <specialise-types>
                  <type name="gk:doc" />
            </specialise-types>
      </config>

      <config evaluator="string-compare" condition="Advanced
Search">
            <advanced-search>
                  <content-types>
                        <type name="gk:doc" />
                  </content-types>
                  <custom-properties>
                        <meta-data aspect="sc:projectRelated"
property="gk:project" display-label-id="project" />
                        <meta-data aspect="sc:projectRelated"
property="gk:organization" display-label-id="organization" />

                  </custom-properties>
            </advanced-search>
      </config>
</alfresco-config>
```

- Finally, the "`webclient`" property file has to be included in order to contain the entries below:

```
#gk:projectRelated
project=Project
organization=Organization
```

Appendix C

Research Classification in SE

Software Engineering Research Classification (Shaw)

Type of Research Question	Type of Result	Validation Criteria
Method or means of Development How can we do/create (or automate doing) X? What is a better way to do/create X?	**Procedure or technique[1]**	**Analysis[5]**
Method for analysis How can I evaluate the quality/correctness of X? How do I choose between X and Y?	**Qualitative or Descriptive Model[2]**	**Evaluation[6]**
Design, evaluation, or analysis of a particular instance What is a (better) design or implementation for application X? What is property X of artifact/method Y? How does X compare to Y? What is the current state of X / practice of Y?	**Empirical model** Empirical predictive model based on observed data	**Experience[7]**
Generalization or Characterization Given X, what will Y (necessarily) be? What, exactly, do we mean by X? What are the important characteristics of X? What is a good formal/empirical model for X? What are the varieties of X, how are they related?	**Analytic model** Structural model precise enough to support formal analysis or automatic manipulation	**Example[8]**
Feasibility Does X even exist, and if so what is it like? Is it possible to accomplish X at all?	**Notation or tool[3]**	**Persuasion[9]**
	Specific solution[4]	**Blatant Assertion** No serious attempt to evaluate result.
	Answer or judgment Result of a specific analysis, evaluation, or comparison	
	Report Interesting observations, rules of thumb	

[1] New or better way to do some task, such as design, implementation, measurement, evaluation, selection from alternatives. Includes operational techniques for implementation, representation, management, and analysis, but not advice or guidelines.

[2] Structure or taxonomy for a problem area; architectural style, framework, or design pattern; non-formal domain analysis. Well-grounded checklists, well-argued informal generalizations, guidance for integrating other results

[3] Formal language to support technique or model (should have a calculus, semantics, or other basis for computing or inference). Implemented tool that embodies a technique.

[4] Solution to application problem that shows use of software engineering principles – may be design, rather than implementation. Careful analysis of a system or its development. Running system that embodies a result; it may be the carrier of the result, or its implementation may illustrate a principle that can be applied elsewhere

[5] e.g. …For a formal model … rigorous derivation and proof. For an empirical model … data on use in controlled situation. For a controlled experiment … carefully designed experiment with statistically significant results

[6] Given the stated criteria, my result... For a descriptive model … adequately describes phenomena of interest … For a qualitative model … accounts for the phenomena of interest… For an empirical model … is able to predict … because …, or… generates results that fit actual data …Includes feasibility studies, pilot projects.

[7] My result has been used on real examples by someone other than me, and the evidence of its
correctness/usefulness/effectiveness is …
For a qualitative model … narrative
For an empirical model or tool … data, usually statistical, on practice
For a notation or technique … comparison of systems in actual use

[8] Here's an example of how it works on.
For a technique or procedure …a "slice of life" example based on a real system …

For a technique or procedure …a system that I have been developing …
For a technique or procedure … a toy example, perhaps motivated by reality

[9]I thought hard about this, and I believe passionately that ...
For a technique … if you do it the following way, then …
For a system … a system constructed like this would …
For a model … this example shows how my idea

Bibliography

[1] Agile Alliance: http://www.agilealliance.org — last accessed June 2011.

[2] Agile Manifesto Signatories. http://www.agilemanifesto.org/sign/display.cgi?ms=000000072 — last accessed June 2011.

[3] Ahmadi, N., et al. A Survey of Social Software Engineering. In *Proceedings of the First International Workshop on Social Software Engineering and Applications*, 2008.

[4] Alexander, C., Ishikawa, S., Silververstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S. *A Pattern Language*. Oxford University Press, 1977.

[5] The Alfresco Open Source Content Management System.http://www.alfresco.com — last accessed June 2011.

[6] Anti-Pattern Catalog. http://c2.com/cgi/wiki?AntiPatternsCatalog/ — last accessed June 2011.

[7] Allport, D.A. Attitudes. In Murchison (ed), *Handbook of Social Psychology*, Worcester, MA, Clark University Press, 1935.

[8] Araujo, R. M., et al. Context Models for Managing Collaborative Software Development Knowledge. In Proceedings of the *First International Workshop on Modelling and Retrieval of Context (KI 2004),* pp. 61-72, 2004.

[9] Avison, D., Lau, F., Myers, M. and Nielsen, P.A. Action Research. In *Communications of the ACM*, Vol. 42, No.1, pp.94-97, 1999.

[10] Bem, D. J. Self-perception theory. In *Advances in experimental social psychology,* Vol. 6, Academic Press, New York, 1972.

[11] Berne, E. *Games People Play: Psychology of human relationships*. NY, Grove Press, 1964.

[12] Brown, W.J., Malveau, R.C., McCormick, H.W. and Mowbray, T.J. *AntiPatterns: Rafactoring Software, Architectures, and Projects in Crisis*. John Wiley & Sons,1998.

[13] Brooks, F. *The Mythical Man-Month*. Addison Wesley, 1975.

[14] Cockburn, A. *Agile Software Development*. Addison-Wesley, Boston, 2002.

[15] Cockburn, A. *People and methodologies in software development*. D.Ph. dissertation, University of Oslo, 2003. Online at http://alistair.cockburn.us/People+and+methodologies+in+software+development. — last accessed June 2011.

[16] Cockburn, A. Research Methods in IS Research. http://alistair.cockburn.us/get/2423 — last accessed June 2011.

[17] Cockburn, A. Characterizing People as Non-Linear, First-Order Components in Software Development. In Proceedings of the *4th International Conference on Systems, Cybernetics, and Informatics*, 2000.

[18] Cockburn, A., Hignsmith, J. Agile Software Development: The People Factor. *In IEEE Computer*, pp.31-33, November 2001.

[19] Cooper, J. and Fazio, R. H. A new look at dissonance theory. In L.Berkowitz (ed.) *Advances in experimental social psychology*, pp. 229–262, 1984.

[20] Curtis, B., Hefley, B. and Miller, S. *The People Capability Maturity Model*. Addison-Wesley, Boston, 2001.

[21] Curtis B. Fifteen Years of Psychology in Software Engineering: Individual Differences &Cognitive Science. In Proceedings of the *7thInternational Conference on Software Engineering (ICSE 1984),* pp. 97-106, 1984.

[22] DeMarco, T. and Lister, T. *Peopleware: Productive Projects and Teams*. Dorset House, New York, 1987.

[23] DeMarco, T. *Why Does Software Cost So Much*. Dorset House, New York, 1995.

[24] DeMarco, T., Hruschka, P., Lister,T., Robertson,S., Robertson,J., McMenamin, S. *Adrenaline Junkies and Template Zombies: Understanding Patterns of Project Behaviour*. Dorset House, 2008.

[25] Fazio R. H, Zanna, M. and Cooper, J. Dissonance and Self-perception: An Integrative View of Each Theory's Proper Domain of Application. *Journal of Experimental Social Psychology*, Vol. 13, pp. 464–479, 1977.

[26] Festinger, L. *A Theory of Cognitive Dissonance*, Stanford University Press, 1957.

[27] Fraser, S., Caster, B., Cherry, T., Segel, H., How Do Teams Shape Objects? - How Do Objects Shape Teams?.*Workshop Report, OOPSLA*, 1994.

[28] Futrell, R., Shafer, D., Shafer, L. *Quality Software Project Management*. Prentice Hall Professional, 2002.

[29] Galliers, R. Choosing Information System Research Approaches. In Galliers, R. (ed.), *Information System Research: Issues, Methods and Practical Guidelines*, Blackwell Scientific Publications, pp. 144-162, Oxford, 1992.

[30] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995.

[31] Gartner. Magic Quadrant for Enterprise Content Management 2009. Retrieved from http://www.oceanus.co.uk/Assets/Downloadablefile/2009-gartner-magic-quadrant-ecm.pdf — last accessed June 2011.

[32] Gemino, A. and Wand, Y. A Framework for Empirical Evaluation of Conceptual Modelling Techniques. In *Requirements Engineering*, 9 (4), pp. 248–260, 2004.

[33] Gerring, J. What Is a Case Study and What Is It Good for?. *The American Political Science Review,* May 2004.

[34] Gorla, N., Lam, Y.W. Who should work with whom. In *Communications of the ACM,* Vol.47, No. 6, June 2004.

[35] Harmon-Jones, E. and Mills, J. (eds.): *Cognitive Dissonance: Progress on a pivotal theory in social psychology*. American Psychological Association, 1999.

[36] Heidmets, M., Raudsepp, M. A conceptual framework for studying environmental mentality and behaviour. In *Trames, 5 (55/50), 3*, pp. 198-210, 2001.

[37] HP website. The Software Development Life Cycle: When to Secure Your Process. http://h71028.www7.hp.com/ERC/cache/568356-0-0-0-121.html — last accessed June 2011.

[38] Hult, M., Lennung, S.A. Towards a Definition of Action Research: a Note and Bibliography. In *Journal of Management Studies*, 17 (2), pp. 241-250, 1980.

[39] Humphrey, W. *Introduction to the Personal Software Process*. Addison-Wesley Professional, 1996.

[40] Humphrey, W. *Introduction to the Team Software Process*. Addison-Wesley Professional, 1999.

[41] Humphrey, W. Three process perspectives: organizations, teams, and people. *Annals of Software Engineering*, Kluwer Academic Publishers, Vol. 14, pp.39–72, 2002.

[42] Johanna Rothman Blog (2007). http://johannarothman.findtechblogs.com/default.asp?item=239706 — last accessed July 2007.

[43] The Joel on Software Forum. Programmers Attack. http://discuss.fogcreek.com/joelonsoftware/default.asp?cmd=show&ixPost=46216 — last accessed June 2011.

[44] Kelly, G. A. *The psychology of personal constructs (Volumes 1 and 2)*. New York, Norton, 1955.

[45] King,N., Horrocks,C. *Interviews in Qualitative Research.* Sage Publications, 2010.

[46] Koenig A. Patterns and Antipatterns. *Journal of Object Oriented Programming*, Vol. 8, No.1, 1995.

[47] Laplante P.A., Neill C.J. *Antipatterns: Identification, Refactoring and Management*. Auerbach Publications, New York, 2006.

[48] Larman C., Kruchten P., and Bittner K. How to Fail with the Rational Unified Process: Seven Steps to Pain and Suffering.*White Paper, Valtech and Rational*, 2002. Retrieved at http://www.agilealliance.org/system/article/file/941/file.pdf — last accessed February 2007.

[49] Lincoln Y.S. and Guba E.G. *Naturalistic Inquiry*. Sage, 1985.

[50] Malim T. and Birch A. *Introductory Psychology*. Palgrave, 1998.

[51] Mathiassen, L. Collaborative Practice Research. In *Information, Technology & People*, Vol. 15, No. 4, 2002.

[52] Mathiassen L. Collaborative Practice Research. In *Organizational and Social Perspectives on Information Technology*, Kluwer Academic Publishers, Boston, MA, pp.127-148, 2000.

[53] Mayring, P. On Generalization in Qualitatively Oriented Research [23 paragraphs]. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 8(3), Art. 26, 2007. http://nbn-resolving.de/urn:nbn:de:0114-fqs0703262.

[54] McConnell, S. *Rapid Development*. Microsoft Press, 1996.

[55] McConnell, S. *Software Project Survival Guide*. Microsoft Press, 1997.

[56] McGuire W. *Constructing Social Psychology*. Cambridge University Press, 1999.

[57] McMahon P.E. Extending Agile Methods: A Distributed Project and Organizational Improvement Perspective. *CrossTalk: The J. Defense Software Eng*., Vol. 18, No. 5, pp. 16–19, May 2005.

[58] Microsoft MSDN. Designing a World-Ready Program. http://msdn.microsoft.com/en-us/goglobal/bb978439.aspx — last accessed June 2011.

[59] Miles, M. and Huberman, M. *Qualitative Data Analysis: an expanded sourcebook*, 2nd edition. Sage Publications, 1994.

[60] Mingers, J. The paucity of multi-method research: a review of the information systems literature. *Information Systems Journal*, 13, pp.233-249, 2003.

[61] Myers, M.D. *Qualitative Research in Business & Management*. Sage Publications, London, 2009.

[62] Osgood, C., Blog. Lazy Programming and the Acquiescent "It Works" Mentality Post, March 2003, http://weblogs.asp.net/cosgood/archive/2003/03.aspx?PageIndex=3 — last accessed June 2011.

[63] Patton, M. *Qualitative Research & Evaluation Methods,* 3rdedition. Sage Publications, 2001.

[64] Potts, J. Alfresco Developer: Working with Custom Content Types.http://ecmarchitect.com, June 2007, — last accessed June 2010.

[65] Project Management Institute. *Guide to the Project Management Body of Knowledge, A (PMBOK Guide)*. Project Management Institute, 2004.

[66] Rogers, C.R. *Client-centered Therapy*. London, Constanble, 1951.

[67] Rousseau, D. Characteristics of departments, positions, and individuals: Contexts for attitudes and behaviour. *Administrative Science Quarterly*, 23, pp. 521-538, 1978.

[68] Sandelowski, M. The problem of rigor in qualitative research. In *Advances in Nursing Science*, 8, 27, 1986.

[69] Sandelowski, M. and Barroso, J. Reading qualitative studies. In *International Journal of Qualitative Methods*, 1 (1), Article 5, 2002. Retrieved at 03/2011 from http://www.ualberta.ca/~ijqm/

[70] Shariff, M., Choudharry, V., Bhandari, A., Majmudar, P. *Alfresco 3 Enterprise Content Management Implementation*. Packt Publishing 2009.

[71] Shaw, M. Writing Good Software Engineering Research Papers – Minitutorial. In *Proceedings of the 25th Intern. Conf. On Software Engineering (ICSE'03)*, IEEE Computer Society, pp. 726-736, 2003.

[72] Shelford T., Remillard G. *Real Web Project Management: Case Studies and Best Practices from the Trenches.* Addison-Wesley Professional, 2002.

[73] Shenton, A. K. Strategies for ensuring trustworthiness in qualitative research projects. *Education for Information,* (22), pp. 63-75, 2004.

[74] Susman, G.I. and Evered, R.D. An Assessment of the Scientific Merits of Action Research. In *Administrative Science Quarterly*, (23), pp. 582-603, 1978.

[75] Web Ontology Language. http://www.w3.org/standards/techs/owl#w3c_all—last accessed June 2011.

[76] Weinberg, G. *The Psychology of Computer Programming*. Van Nostrand Reinhold, 1971.

[77] Weinberg, G. *High-Level COBOL Programming*. Winthrop Publishers, 1977.

[78] Weinberg, G. *Systems Thinking; Quality Software Management*, Volume 1. Dorset House Publishing, 1992.

[79] Weinberg, G. *First-Order Measurement; Quality Software Management*, Volume 2. Dorset House Publishing, 1993.

[80] Weinberg, G. *Congruent Action; Quality Software Management*, Volume 3. Dorset House Publishing, 1994.

[81] Weinberg, G. *Anticipating Change; Quality Software Management*, Volume 4. Dorset House Publishing, 1997.

[82] Williams, M. Generalization in interpretive research. In Tim May (ed.), *Qualitative research in action,* pp.125-143. London, Sage, 2002.

[83] Whitehead, J., Mistrík. I., Grundy, J., Van der Hoek, A. Collaborative Software Engineering: Concepts and Techniques. In *Collaborative Software Engineering*, pp. 1-34, Springer, 2010.

[84] Yin, R. *Applications of case study research*. Newbury Park, CA, Sage Publishing, 1993.