

Chapter 4

The Agile and the Disciplined Software Approaches: Combinable or Just Compatible?

Lise Tordrup Heeager

1 Introduction

In information systems development research, agile software development is a popular topic. This chapter uses the terminologies of agile and disciplined software development approaches adopted from Boehm and Turner (2003a, 2004). The agile and the traditional disciplined software methodologies are two approaches to describe processes for developing software (Dahlberg et al. 2006). The agile software development methodologies (e.g. XP and Scrum) promise a way to deliver software without excessive cost. On the contrary, disciplined software development processes (Boehm and Turner 2004) (e.g. CMMI and the ISO standards) are well defined and proven, but require a lot of effort (Nawrocki et al. 2002a).

Offhand, the two seem contradicting (Turner 2002; Turner and Jain 2002), but several researchers agree that a software project needs both agility and discipline (Boehm and Turner 2003a, b; Nawrocki et al. 2006). Neither the agile nor the disciplined approaches provide the ultimate approach, both have shortcomings and pitfalls (Boehm and Turner 2003a). The approach of the disciplined methodologies (plan everything and follow the plan) works well for stable and less complex projects, but they lack the ability of handling change and complexity. The agile methodologies embrace change (Beck and Andres 2004), but offhand they lack the ability of quality assurance.

Software development organizations are increasingly interested in the possibility of adopting agile approaches (Pikkarainen and Mäntyniemi 2006); even organizations that have been employing process standards are now increasingly interested in the possibility of adopting agile approaches (Marçal et al. 2008). Rapid change and increasing software criticality drive organizations to balance the agility and discipline (Boehm and Turner 2004).

L.T. Heeager (✉)
Aalborg University, Aalborg Ø, Denmark
e-mail: liseh@cs.aau.dk

There are ongoing debates whether the quality of the products of the agile approaches are satisfactory (Hashmi and Baik 2007), and some projects, for example, safety-critical projects, require standards to be followed when developing software (Fritzsche and Keil 2007; Heeager and Nielsen 2009; Theunissen et al. 2003). Since the agile and the disciplined approaches are grounded in opposing concepts, balancing the two is not straightforward (Opelt and Beeson 2008; Reifer 2003; Vinekar et al. 2006). Some organizations attempt to use an agile software development approach and at the same time comply with a quality assurance standard (Vinekar et al. 2006; Nerur et al. 2005); however, it is not well understood how to do this in practice (Kähkönen and Abrahamsson 2004; Nawrocki et al. 2002b; Pikkarainen 2009). This has let research to focus on the compatibility and combinability of the agile and the disciplined approaches (Rönkkö et al. 2008; Zanatta and Vilain 2006).

With the research question ‘are the agile and disciplined approaches combinable or just compatible?’ in mind, the purpose of this chapter is to give an overview of the combinability and/or the compatibility of the agile and disciplined approaches and give a more detailed picture of the challenges facing an organization trying to use an agile software development approach in a disciplined setting. This is done through a review of 79 papers dealing with the subject of the combinability or compatibility of the agile and disciplined approaches. This review is relevant to researchers who wish to stay up to date with the state of research on the combinability and/or compatibility of agile and disciplined approaches and to practitioners who wish to implement an agile software development process in practice and at the same time comply with a quality assurance standard.

This chapter is structured as follows: The research approach is outlined in Sect. 2. Then the analysis on compatibility and combinability is presented in Sect. 3. In Sect. 4, the results of the analysis are discussed and also include a discussion of the guidelines and the challenges identified in the literature when trying to obtain compatibility between the agile and disciplined approaches. The analysis results in four propositions are also presented in Sect. 4. Finally, the results are concluded on in Sect. 5.

2 Research Approach

Previous research on the combinability and/or compatibility of the agile and the disciplined approaches was reviewed. The papers were found through a comprehensive search conducted in three steps (summarized in Fig. 4.1) (Webster and Watson 2002). First, the web was searched using relevant keywords. The keywords contained different combinations of names of process standards and agile methodologies and more general keywords as ‘compatibility and agile’ or ‘combine and agile’. In parallel, ranked journals and relevant conference proceedings were searched. In the second step, a backward search (following references of identified papers) and a forward search (used Google Scholar to find relevant papers citing the

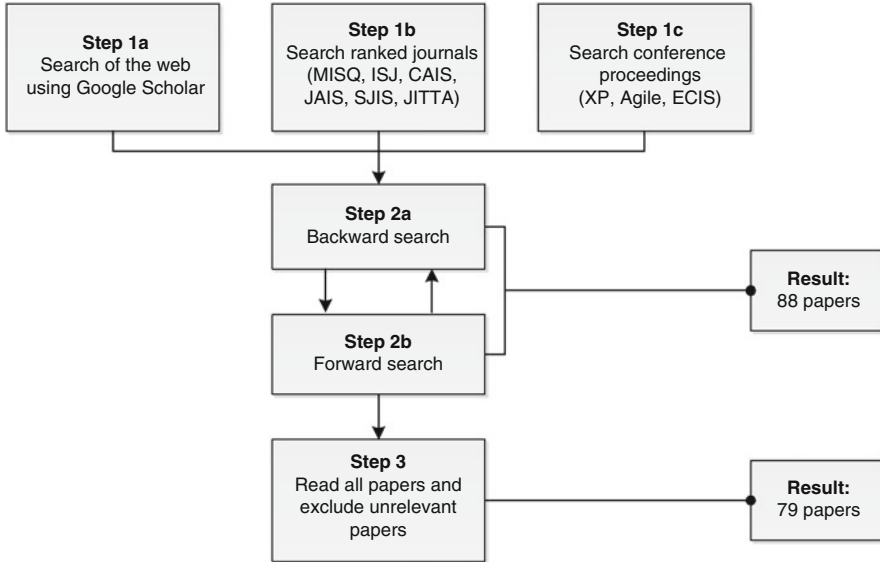


Fig. 4.1 The search strategy

identified papers) were done iteratively. At this point, the relevance of the papers was determined by reading the titles and abstract, so far the search had resulted in 88 papers. Each of these was read, and nine of them were discarded as they lacked relevance, either because the combinability and/or compatibility of the agile and the disciplined approaches was a very small and insignificant part of the paper or the fact that the paper only was dealing with either agile or disciplined approaches and not both, leaving 79 papers for further analysis (see reference list in Appendix A).

Conventional content analysis (Hsieh and Shannon 2005) was used as the primary analysis strategy (the analysis strategy is summarized in Fig. 4.2). This strategy was chosen as existing theory, and research literature on the combinability and/or compatibility of the agile and the disciplined approaches is limited. To structure the analysis, each of the papers was coded using Atlas.ti V6 (Muh 1991). The process consisted of three linear steps – each with a separate result. First, the papers were coded using open coding (Strauss and Corbin 1990). The focus when reading and coding the papers was on the combinability/compatibility and non-combinability/incompatibility of the agile and the disciplined approaches, on guidelines to overcome the challenges when using an agile software development approach in a disciplined setting and on the strengths and weaknesses of such a practice. This was done in order to get an overview of the content of the previous research. The first step resulted in 59 different codes applied to 1,038 quotations. The second step was to categorize the codes, thus creating coding schemes. The resulting coding schemes can be seen in Appendix B. One table for each type of codes include the following: general codes, codes describing combinability/compatibility, codes describing non-combinability/incompatibility, codes

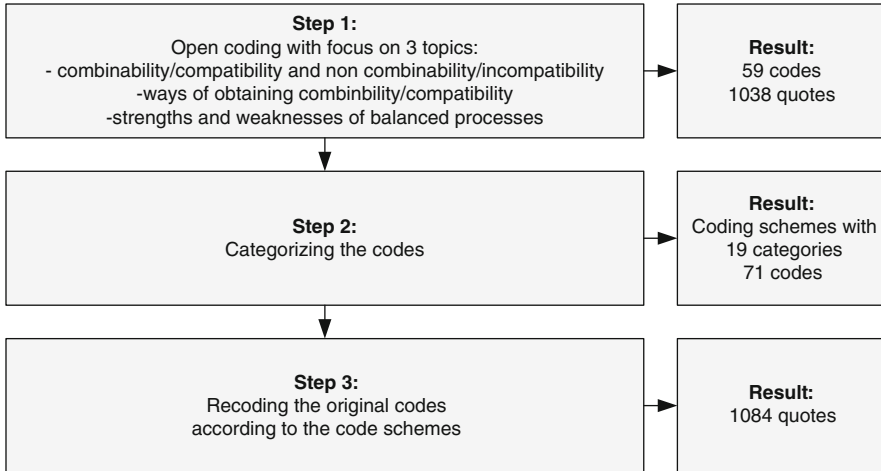


Fig. 4.2 The analysis strategy

describing guidelines of how to use an agile software development approach in a disciplined setting and codes describing strengths and weaknesses of the agile, the disciplined approaches attempting to balance agility and discipline. In the third step, the codes and quotes from step 1 were revisited and recoded according to the coding schemes containing 19 categories with 71 codes. This resulted in 1,084 quotations.

3 Analysis: Combinability and Compatibility

To be combinable or compatible are two different things. In order to determine the combinability and compatibility of the agile and the disciplined approaches, these two terms have to be defined. This is done with a starting point in the dictionary interpretation of these key terms.

To combine is ‘to join two or more things or groups together to form a single one’ (Wehmeier 2010), meaning that you are able to do two things at the same time without compromising either. To be compatible is to be ‘able to exist together’ (Wehmeier 2010), meaning that something is able to work at the same time as something else. Thus, combinability is a Boolean concept while compatibility can be graded.

The interpretations of the two key terms used in this chapter in the context of determining the compatibility and combinability of the agile and the disciplined approaches are as follows: (1) The agile and the disciplined software approaches are combinable if it is possible to use an agile software development approach and at the same time comply with a quality assurance standard without compromising the agility of the agile approach. (2) The agile and the disciplined software

development approaches are compatible if it is possible to use some of the agile practices and principles and at the same time be able to comply with a quality assurance standard

The literature review shows that previous research does not distinguish between combinability and compatibility of the agile and the disciplined approaches. Several papers state that the agile and disciplined approaches are combinable; however, when reading into the analysis and results presented in the papers adopting the definitions of this chapter, the conclusion should instead be that the agile and the disciplined approaches are compatible, but not combinable (e.g. S10, S12, S26, S75).

3.1 *The Compatibility of the Agile and the Disciplined Approaches*

Offhand, the agile approaches seem in conflict with the disciplined approaches, but several researchers agree that it is possible to use (some of) the practices and principles of agile software development process and at the same time comply with a quality standard. The majority of the papers reviewed (e.g. S3, S10, S23, S24, S25, S36, S46, S61, S65, S69, S70, S79) state that the agile and the disciplined approaches (to some degree) are compatible. Most papers engage themselves in determining the overall compatibility or incompatibility of the agile and the disciplined approaches, and some of these engage in comparing specific methodologies, mainly using XP or Scrum from the agile methodologies and CMMI and ISO from the disciplined methodologies.

Thirty-nine of the papers reviewed outline which agile and disciplined approaches they compare. These are shown in Table 4.1. The relationship between XP and CMMI is the one most analysed. Scrum comes in second among the agile approaches, whereas ISO is the process model second most used. Five papers not only analyse the relationship of an agile and a disciplined approach but also mixed practices from two or more agile approaches. Vriens (S74) not only used a mix of XP and SCRUM but also a mix of CMMI and ISO.

Table 4.1 Comparison of the agile and the disciplined approaches in the reviewed papers

	XP	Scrum	RUP	XP/Scrum	Sum
ISO	(S42, S43, S48, S67, S77)	–	–	(S41)	6
CMMI	(S2, S3, S18, S28, S30, S40, S45, S47, S49, S51, S54, S55, S60, S62, S75)	(S16, S26, S27, S34, S35, S38, S39, S66, S78)	(S33, S37)	(S1, S12, S61)	29
CMMI/ISO	(S14)	–	–	(S74)	2
FDA	(S76)	(S24)	–	–	2
Sum	22	10	2	5	

In a point-counterpoint paper, Beck and Boehm agree that agility and discipline are not opposites (S4). In another paper, Boehm also states that even though advocates of the agile and the disciplined approaches consider them opposites, including parts of both in a project can be advantageous (S7), and Glass (S21) points out that ‘one-size-fits-all’ approaches do not work. Mahnic and Zabkar (S34, S35) conclude that it is possible to build software process that balances agility and discipline. Glazer (S22) concludes that the agile approaches and CMMI complete each other and can give fast, affordable, visible and long-term benefits. Through a comparison of the waterfall model and agile approaches, Huo et al. (S25) find that agile approaches contain QA practices and that these occur more frequently than in the waterfall model. Baker (S3) states that the agile approaches are focused and comprehensive, and according to Beck (S4), XP is disciplined, as it provides a clear picture of what activities to apply. DeMarco and Boehm (S15) support this statement by adding that XP involves more planning than CMM organizations.

According to the reviewed papers, using an agile approach in a disciplined setting makes it possible to take advantage of the strengths and compensate for the weaknesses of the two (S10, S19, S44). Turner and Jain (S70) state that the agile and the disciplined approaches have much in common and their strengths and weaknesses are complementary, while Boehm and Turner (S9, S11) present a risk-based method for developing balanced strategies.

However, only few papers deal with the strengths and weaknesses of processes balancing agility and discipline, and those who do only point out the strengths. The disciplined approaches support the agile approaches by providing a disciplined framework (S17), while agile methods ensure that processes are implemented efficiently while embracing change (S14, S26, S66), this balances adaptability and predictability in order to better serve customer needs (S27). The disciplined approaches are also helpful for the agile approaches, as they are able to help identify the shortcomings of the agile approaches (S18), support the decision of which processes to address (S26, 66), provide structure to help ensure your agile processes are followed (S41) and reduce the risk in agile development (S17).

The researchers disagree on how compatible the agile and the disciplined approaches are. The majority of the papers reviewed conclude that it is possible for a company using an extended agile approach to comply with process standards such as ISO and CMMI level 2 or 3. In the following subsections, the compatibility and/or combinability of the agile approaches and, respectively, the ISO standards and the CMM/CMMI will be analysed.

3.2 Agile Software Methodologies and ISO

The ISO standards ISO 9000 and ISO 9001 are those mostly used for analysis by researchers dealing with agile software development methodologies and ISO (S41, S42, S43, S48, S64, S77). Three papers do not specify which agile methodology they use for analysis, while the remaining five papers either use Scrum or XP (see Table 4.2).

Table 4.2 Comparison of agile methods and ISO in the reviewed papers

	XP	Scrum	Agile	Sum
ISO 9000	–	(S43, S48)	–	2
ISO 9001	(S42)	(S77)	(S41, S64)	4
ISO/IEC15504 (SPICE)	–	–	(S29)	1
ISO/IEC12207/1995	(S67)	–	–	1
Sum	2	3	3	

Agile approaches can be adapted to ensure compatibility with ISO standards (S67). Based on empirical evidence, Wright (S77) concludes that companies using XP can meet the requirements of ISO9001. McMichael and Lombardi (S41) conclude that ISO9001:2000 is able to help ensure the agile processes are followed. Melis (S42) concluded that several requirements of ISO9001:2000 are implemented by the existing tools in XP project management, while Stålhane and Hanssen (S64) suggest few changes to both the ISO standard and the agile approaches. Lami and Falcini (S29) showed that ISO/IEC15504 in principle is applicable to agile contexts, but that in practice, problems may occur, for example, in creating an agile process reference model and in order to find an assessor that has experience from software development in agile contexts.

3.3 Agile Software Methodologies and CMM/CMMI

Several researchers of the papers (18 of the review papers) compare the components of the CMM (S30, S37, S40, S55) or CMMI (S1, S12, S16, S17, S18, S30, S38, S39, S40, S45, S51, S66, S70, S78) process model with agile methods, such as XP and Scrum. Tables 4.3 and 4.4 give an overview of which agile methods are compared with the generic practices and process areas of, respectively, CMMI and CMM.

An analysis of the results on the comparison of agile methods with the process areas of CMMI shows that the researchers agree that CMMI level 2 is largely compatible with the agile approaches, CMMI level 3 are partially compatible with agile approaches while CMMI level 4 and 5 are less compatible with agile approaches. There does not seem to be a big difference in the compatibility of CMMI and, respectively, XP and Scrum.

Fifteen papers state that the agile approaches are compatible with CMMI level 2 or 3. Marcal et al. (S38, S39) say that it is possible to reach CMMI level 2 without compromising the agility. Alegria and Bastarrica (S1) state that each process area needs to be defined explicitly by the organization and complemented with elements obtained from other sources. Baker (S3), Bos and Vriens (S12) and Kähkönen and Abrahamsson (S28) conducted case studies of companies complying with CMMI level 2 using agile approaches. Santana et al. (S62) conducted studies of two companies complying with CMMI level 2 and 3, respectively, using XP and

Table 4.3 Comparison of agile methods and CMMI in the reviewed papers

	XP	Scrum	RUP	Agile	Sum
CMMI generic practices	(S18)	(S16, S18, S66)	–	(S1, S70)	6
CMMI process areas	(S12, S18, S30; S40, S45, S51)	(S12, S18, S38, S39, S78)	–	(S1, S17, S44, S70)	15
Sum	7	8	0	6	

Table 4.4 Comparison of agile methods and CMM in the reviewed papers

	XP	Scrum	RUP	Agile	Sum
CMM generic practices	–	–	–	–	0
CMM process areas	(S30, S40, S55)	–	(S37)	–	4
Sum	3	0	1	0	

Scrum, while Manzoni et al. (S37) conclude that reaching CMMI level 3 using RUP is possible. Rönkkö et al. (S61) found that the process areas of CMMI level 2 and 3 are adopted in parallel and can be supported by the use of agile approaches. Jakobsen and Johnson (S26) recommend extending the agile approaches using the mandatory goals and expected practices of CMMI level 2 and 3. Laurila (S30) concluded that CMM and agile methods are practice compatible, but not idea compatible.

Several researchers state that agile methods do not address the CMMI level 4 or 5 in their original state (e.g. S8, S30, S40, S55), but only one paper states that it is not possible to reach the highest maturity levels when being agile (S18). Few results show that it is possible to comply with CMMI level 4 or 5. Through an analysis of XP and CMM, Martinsson (S40) found it possible and advantageous to use XP to reach the highest CMM level. At Microsoft, they created an agile life cycle complying with CMMI level 5 (S2), and at Systematic, they used Scrum to reach the highest level (S66, S77).

The disciplined approaches tell what to do while the agile approaches tell how to do it (S16). CMMI is a maturity model while agile is a development philosophy (S31).

3.4 Agile Approaches for Developing Safety-Critical Software

Another ongoing discussion in the reviewed papers is whether or not agile approaches are useful when developing safety-critical software (S32). Safety-critical software is software in which failure can result in direct injury to humans or cause severe economic damage (Turk et al. 2002). In security engineering, the home grounds of the agile and disciplined approaches are considered far from each other as a high level of discipline is needed throughout the development. The process standards for safety-critical software are often more strict than, for

example, ISO9001 (S77) as they require a robust development process that ensures the quality and safety of the product (S76).

Even though developers of safety-critical software are experimenting with agile approaches (S65, S73), only two case studies of the reviewed papers included a company developing a safety-critical product (S24, S76). They use elements of, respectively, Scrum and XP while complying with the US Food and Drug Administration's (FDA) standard for software development. These cases are proof that it is possible to implement agility in a safety-critical software development process. This proof is supported by other researchers. According to Beznosov and Kruchten (S6), pair programming naturally facilitates internal design and code review and motivates developers to follow coding standards. Wäyrynen et al. (S75) also conclude that XP is aligned with security engineering.

However, the review also shows consensus among the researchers on the fact that agile approaches in their pure form are not suited for developing safety-critical software. Beznosov (S5) introduces eXtreme Security Engineering (XSE), an application of XP practices to security engineering. Boström et al. (S13) also proposes extending XP practices, while Siponen et al. (S63) give an example of how to add security techniques to agile approaches in general.

The researchers have different advice on how to fit the agile methods. For example, Pohjola (S58) concludes that the test-driven development practice of XP is the key. Lindvall et al. (S32) gathered the experiences on agility at a workshop. They found that safety-critical projects can be conducted using agile approaches; the key is to make the performance requirements explicit and plan proper levels of testing early in the process. In order to introduce agility in the development processes of safety-critical software in general, a body of evidence that agile approaches provide secure software is needed (S73).

4 Discussion

Addressing the research question, this section discusses the combinability and compatibility of the agile and the disciplined approaches. It discusses the guidelines and challenges identified in the literature on obtaining compatibility between the two approaches. Furthermore, four propositions created based on the analysis and discussion is presented.

4.1 *Compatibility, Not Combinability*

The analysis showed that the agile and the disciplined approaches are highly compatible, that is, in practice it is possible to implement several agile practices and principles in a software development process and at the same time be able to comply with a quality assurance standard. The agile and the disciplined approaches

are however too different to be combined, meaning, that the agility of a project and/or the practices and principles of an agile software development process will be harmed in some degree by the regulations of a disciplined quality standard or a process model.

Proposition 1: The agile and the disciplined approaches are compatible, but not combinable.

Taking into consideration that the agile software approaches were developed as a counterpoint to the heavyweighted traditional, disciplined software approaches, this proposition seem natural but however important for both practitioners and researchers. It is important that practitioners understand the consequences of introducing a quality standard in their agile process or that it may be possible to heighten the agility of a software development process complying with a quality standard. For further research, it is important to understand the difference between combinability and compatibility when comparing an agile and a disciplined approach.

4.2 Obtaining Compatibility

Several researchers focus on how to extend either the agile approaches, the disciplined approaches or both of them in order to make them compatible. Most researchers focus on how to extend the agile approaches, whereas only two see a need in extending the disciplined approaches. Researchers have different ways of extending the approaches.

Four papers make suggestions on extending XP. Beznosov (S5) focuses on extending XP for security engineering. Nawrocki (S45) proposed three modifications to XP: the written documentation and requirements are to be managed by a tester, the planning game is to be modified so that it allows having multiple customer representatives and a requirements engineering phase in the beginning of a project to provide wider perspective of the product being developed. Boström et al. (S13) focus on the planning game as a way of extending XP practices. Nawrocki et al. (S47) have built a maturity model for XP (XPMM), a 4-level maturity model with a structure that resembles CMMI. Their aim was to build a maturity model that is simple and lightweight. Two papers focus on extending Scrum. Zanatta (S78) has developed an extension, called xScrum, which consists of guidelines that allow Scrum to be compatible with the requirements management and requirements development process areas of CMMI. Marcal et al. (S38) suggest that few adaptations on Scrum will make it much more compatible or even fully combinable with CMMI project management process areas. Manzoni and Price (S37) focus on extending RUP and state that an organization must customize its own practice and develop its own procedures to satisfy key practices of CMMI not supported by RUP. Zuser et al. (S79) propose a standard for quality support in software process models. Port and Bui (S59) introduce two strategies to obtain compatibility of the agile and disciplined

approaches. One adding cost-benefit to the agile approaches, the other modulates development iteration size to maximize the expected cost-benefit for each iteration. Visconti and Cook (S72) have developed an ideal process model for agile approaches.

Laurila (S30) suggests the creation of a CMMI-like framework that is more compatible with agile ideas. Instead of a plan-driven focus, the framework should embrace change. Stålhane and Hanssen (S64) suggest that the guidelines of ISO should include more specific guidelines, for example, a definition of acceptable reviews.

Risk management is another approach that can assist the process of obtaining compatibility between the agile and the disciplined approaches and help organizations answer the question of how much documentation is enough (S7). Boehm and Turner (S9, S10, S11) therefore suggest a risk-based approach when incorporating both agile and disciplined practices and principles. They have developed a five-step model. Galal-edein (S19) discusses two approaches for obtaining compatibility, the ambidextrous organization approach and the risk-based approach. They found the risk-based approach of Boehm and Turner most useful, as the home grounds concept is practical when dealing with development projects. Geras et al. (S20) also concludes that the home grounds of Boehm and Turner represent a way to obtain compatibility between the agile and the disciplined approaches.

Proposition 2: Obtaining compatibility of the agile and the disciplined software development approaches requires an extension of either one or both approaches.

4.3 The Challenges for Obtaining Compatibility

The analysis revealed several challenges when using an agile software development approach and at the same time being compliant with a quality assurance standard. The two main challenges identified in the literature are on the issues of handling the documentation and the requirements. This section deals with these two issues and the advice presented by the literature to overcome these.

4.3.1 The Documentation

A difference between the disciplined approaches controlled by quality standards and the agile approaches is the amount of documentation required. Agile approaches do not support the degree of documentation demanded by disciplined approaches (S8, S53, S67, S78). The different focus on documentation is also reflected in the agile manifest which says ‘working software over comprehensive documentation’ (Beck et al. 2001). In the case study presented by Kähkönen and Abrahamsson (S28), they had to do some additional documentation to comply with the standard. According to Nawrocki et al. (S45), the lack of documentation is the

main weakness of XP. Fortunately, both agile (S67) and disciplined (S3) approaches are highly flexible and adaptable. Some researchers state that documentation does not contradict the core principles of agility. Bos and Vriens (S12) argue that it is not necessary to write piles of documentation to comply with CMM level 2, and according to Diaz (S16), it is proven that the CMMI model can be applied in a light manner.

Research provides experiences and guidelines on how to deal with the amount of documentation. Stålhane and Hanssen (S64) suggest adding activities such as review meetings and writing design documents to the agile approaches, while still keeping the most agile ideas such as short iterations, building in increments and including the customer. The case study of Namioka and Bran (S43) showed how the short iterations and continuing updates to the documents enhanced the flexibility. Other guidelines are to treat the written documents as deliverables at the end of an iteration (S43) and to provide just enough documentation to help with enforcement of existing processes (S41). Process standards do not explicitly state how much documentation is required, and many companies end up doing more documentation than needed.

Previous research also suggests extending the agile approaches to have a stronger emphasis on documentation. In parallel, disciplined approaches need to keep the documentation to a minimum (S54). The key to making the agile and the disciplined approaches compatible on the issue of documentation seems to be a happy medium between focusing on working software and on writing documents.

Proposition 3: To obtain compatibility between the agile and disciplined approaches, the focus needs to be on both working software and on documentation.

4.3.2 The Requirements

The agile method XP bases the requirements on user stories created by the customer (Beck et al. 2001); these differences in how the agile and the disciplined approaches handle requirements can cause problems in obtaining compatibility (S8). According to Nawrocki et al. (S45), a weakness of XP is that this approach lacks documentation of requirements. User stories written in a plain business-like language cannot be used directly as requirements by a company wishing to comply with a process standard (S5, S77). Instead, each user story needs to be translated into functional test cases (S5). Nawrocki et al. (S45) propose a modification of the XP life cycle introducing a requirements engineering phase at the beginning of a project and to make the tester responsible for managing the requirements.

Several tools supporting the XP process has been developed (S77). Melis (S42) suggests using such tools to manage the gathering of requirements and planning activities.

Proposition 4: The different strategies for handling requirements proposed by the agile and the disciplined approaches can cause problems when trying to obtain compatibility.

5 Conclusions

A large number of software companies have a desire to adopt an agile software development approach and at the same time comply with a quality standard. The two approaches seem contradicting, but case studies which successfully balance agility and discipline are emerging. Therefore, the purpose of this chapter is to determine whether the agile and the disciplined software development approaches are combinable or just compatible. This is done by reviewing previous research comparing the agile and discipline approaches for analysis or through a case study. The review includes 79 papers which were analysed using conventional content analysis and in this process coded using Atlas.ti V6.

The analysis showed that the agile and the disciplined approaches are highly compatible, but not combinable. In practice, it is possible to implement several agile practices and principles in a software development process and at the same time be able to comply with a quality assurance standard; however, the agility of a project and/or the practices and principles of an agile software development process will be harmed in some degree by the regulations of a disciplined quality standard or process model.

The agile approaches are mostly compared with CMMI level 2 or 3 or ISO. Few cases prove that it is possible to use parts of an agile approach with CMMI level 4 or 5. Agile practices and principles can also be introduced in the software development process when developing safety-critical software in order to make the development process more flexible. The research however also agrees that obtaining compatibility between the agile and the disciplined approaches is not straightforward. The research provides guidelines and different models on how to extend the agile approaches in order to make them fit the disciplined approaches. The amount of documentation and the way the requirements are handled seems to be the biggest difference between the agile and the disciplined approaches which proposes some challenges. Previous research does not distinguish between combinability and compatibility of the agile and the disciplined approaches; hence, the researchers are advised to reconsider their terminology when comparing the agile and the disciplined approaches and adopt the definitions of compatible and combinable presented in this chapter.

Only few empirical studies have been conducted on software development teams attempting to use an agile software development process in a disciplined setting. The research within this area would therefore gain from further empirical data on the compatibility of the two approaches and on the strengths, weaknesses and pitfalls when trying to obtain compatibility between the agile and the disciplined approaches.

Appendix A. Studies Included in the Review

See http://heeager.com/files/combinability_or_compatibility_studies.pdf

Appendix B. Coding Schemes

See http://heeager.com/files/combinability_or_compatibility_coding_schemes.pdf

References

- Beck K, Andres C (2004) *Extreme programming explained: embrace change*. Addison-Wesley Professional, Reading
- Beck K, Beedle M, van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, Kern J, Marick B, Martin RC, Mellor S, Schwaber K, Sutherland J, Thomas D (2001) *Manifesto for agile software development*. Available at: <http://agilemanifesto.org/>. 30 June 2011
- Boehm B, Turner R (2003a) Observations on balancing discipline and agility. In: *Proceedings of the agile development conference*. IEEE Computer Society, Salt Lake City, pp 32–39
- Boehm B, Turner R (2003b) Using risk to balance agile and plan-driven methods. *Computer* 36 (6):57–66
- Boehm B, Turner R (2004) Balancing agility and discipline: evaluating and integrating agile and plan-driven methods. In: *Proceedings of the 26th international conference on software engineering*. IEEE Computer Society, Washington, DC, pp 718
- Dahlberg H, Ruiz FS, Olsson CM (2006) The role of extreme programming in a plan-driven organization. In: *The transfer and diffusion of information technology for organizational resilience: IFIP TC8 WG 8.6 international working conference*. Springer, Galway, pp 291
- Fritzsche M, Keil P (2007) Agile methods and CMMI: compatibility or conflict? *e-Inforn Softw Eng J* 1(1):9–26
- Hashmi SI, Baik J (2007) Software quality assurance in XP and spiral-A comparative study. In: *International conference on computational science and its applications, 2007. ICCSA 2007*. IEEE, Fukuoka, pp 367
- Heeager LT, Nielsen PA (2009) Agile software development and its compatibility with a document-driven approach? A case study. In: *Australasian conference on information systems, Melbourne, Australia*, pp 205
- Hsieh HF, Shannon SE (2005) Three approaches to qualitative content analysis. *Qual Heal Res* 15 (9):1277–1288
- Kähkönen T, Abrahamsson P (2004) Achieving CMMI level 2 with enhanced extreme programming approach. *Lect Notes Comput Sci* 3009:378–392
- Marçal ASC, de Freitas BCC, Soares FSF, Furtado MES, Maciel TM, Belchior AD (2008) Blending scrum practices and CMMI project management process areas. *Innov Syst Softw Eng* 4(1):17–29
- Muhr T (1991) ATLAS/ti – a prototype for the support of text interpretation. *Qual Sociol* 14 (4):349–371
- Nawrocki JR, Walter B, Wojciechowski A (2002a) Comparison of CMM level 2 and eXtreme programming. *Lect Notes Comput Sci* 2349:288–297

- Nawrocki JR, Jasiński M, Walter B, Wojciechowski A (2002b) Combining extreme programming with ISO 9000. *Lect Notes Comput Sci*:786–794
- Nawrocki J, Olek L, Jasinski M, Paliswiat B, Walter B, Pietrzak B, Godek P (2006) Balancing agility and discipline with xprince. *Lect Notes Comput Sci* 3943:266–277
- Nerur S, Mahapatra RK, Mangalaraj G (2005) Challenges of migrating to agile methodologies. *Commun ACM* 48(5):78
- Opelt K, Beeson T (2008) Agile teams require agile QA: how to make it work. An experience report. In: *Agile, 2008. AGILE'08*. IEEE Computer Society, Toronto, pp 229
- Pikkarainen M (2009) Towards a better understanding of CMMI and agile integration-multiple case study of four companies. In: *Product-focused software process improvement: 10th international conference. PROFES 2009*, Springer, Oulu, pp 401
- Pikkarainen M, Mäntyniemi A (2006) An approach for using CMMI in agile software development assessments: experiences from three case studies. In: *SPICE 2006 conference*, Luxembourg, pp 4.
- Reifer DJ (2003) XP and the CMM. *IEEE Softw* 20(3):14–15
- Rönkkö M, Jarvi A, Makela MM (2008) Measuring and comparing the adoption of software process practices in the software product industry. *Lect Notes Comput Sci* 5007:407–419
- Strauss AL, Corbin J (1990) *Basics of qualitative research: grounded theory procedures and techniques*. Sage, Newbury Park
- Theunissen WH, Kourie DG, Watson BW (2003) Standards and agile software development. In: *Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*. South African Institute for Computer Scientists and Information Technologists, Republic of South Africa, Johannesburg, pp 178
- Turk D, France R, Rumpel B (2002) Limitations of agile software processes. In: *Third international conference on eXtreme programming and agile processes in software engineering*, Citeseer, pp 43
- Turner R (2002) Agile development: good process or bad attitude? *Lect Notes Comput Sci* 2559:134–144
- Turner R, Jain A (2002) Agile meets CMMI: culture clash or common cause? *Lect Notes Comput Sci* 2418:153–165
- Vinekar V, Slinkman CW, Nerur S (2006) Can agile and traditional systems development approaches coexist? An ambidextrous view. *Inf Syst Manage* 23(3):31–42
- Webster J, Watson RT (2002) Analyzing the past to prepare for the future: writing a literature review. *MIS Q* 26(2):13–23
- Wehmer S (2010) *Oxford advanced learner's dictionary*. Homepage of Oxford University Press. Available at: <http://www.oxfordadvancedlearnersdictionary.com/>. December 2010
- Zanatta AL, Vilain P (2006) Extending an agile method to support requirements management and development in conformance to CMMI. *HIFEN* 30:58