www.ietdl.org

Published in IET Software
Received on 27th October 2011
Revised on 6th February 2012
doi: 10.1049/iet-sen.2011.0193

Special Issue: European Systems and Software Process Improvement and Innovation (EuroSPI)

ISSN 1751-8806

Improving agility and discipline of software development with the Scrum and CMMI

K. Łukasiewicz J. Miler

Gdansk University of Technology, Narutowicza 11/12, Gdansk 80-233, Poland
E-mail: katarzyna.lukasiewicz@eti.pg.gda.pl

Abstract: This study presents a method of combining the Scrum methodology with the CMMI maturity model to improve both agility and discipline of software development. First, the authors propose the CMMI–Scrum reference model, which maps Scrum practices onto 123 practices of CMMI staged levels 2 and 3. For 60% of CMMI practices, which are insufficiently covered by Scrum they add new practices that improve discipline while maintaining agility. The practices to improve an actual software development process are selected from the reference model with the P–Sel algorithm based on answers to a questionnaire with 25 single-choice questions. They have applied our approach to processes of two IT companies, where on average 72% of the suggested practices were confirmed, 24.5% were mismatched and 3.5% were rejected.

1 Introduction

Thus far, many large companies engaged in long-term and complex IT projects have improved their processes towards the disciplined development methods, which allowed them to control the process and ultimately enabled the introduction of maturity control models such as capability maturity model integration (CMMI) [1]. In recent years, the agile development methodologies stemming from the Agile Manifesto [2], such as Scrum [3], have drawn increasingly more attention by being successfully used to improve the processes of IT companies, though mainly the smaller ones.

The disciplined processes of mature organisations provide well-predictable results in stable environments and business conditions, whereas the agile approach delivers key business value faster and cheaper in response to the rapidly changing requirements. The problem is to successfully combine the mature approach to organisation and an agile approach to process to achieve benefits from both of them and minimise the costs and drawbacks of the culture change. If successful, the agile methodologies in mature environments can reduce cost as well as time to market. In contrast, bringing more maturity into agile processes can improve product quality, increase process manageability and increase the suitability of agile practices for larger, more complex IT projects.

The solution is to provide a model that will smoothly combine disciplined and agile practices into a coherent set practically applicable in various organisations. To acquire the widest adoption, the satisfactory solution should be based on current and commonly used disciplined and agile approaches, be applicable by both mature and agile organisations, be self-explanatory and require little external consultancy. Moreover, the solution should recognise and preserve well-working practices and focus on possible improvements.

Our solution combines the agility of the Scrum and the maturity of the CMMI model. It comprises

- the CMMI–Scrum reference model, which maps the Scrum practices onto the CMMI practices and extends them to balance agility and discipline;
- the questionnaire, which identifies the needs for process improvement;
- the P–Sel algorithm, which suggests the practices from the reference model based on the answers to the questionnaire;
- the process of solution application in a software development company;
- the MatureScrum software tool, which implements the questionnaire and the algorithm.

2 Related work

Some models combining Scrum and CMMI [4–7] as well as successful applications of both CMMI and Scrum have already been reported [8–12, 6]. However, most of the models insufficiently address compliance of these approaches, whereas the applications do not follow any defined improvement process, but extensively use costly top consultants in agile methodologies and usually mention very little about how exactly the whole process of finding the perfect balance has been prepared. Most companies are reluctant to share the details of their models. It applies to Babuscio's report [8] in which he describes the HTPi journey to embrace the agile principles while maintaining CMMI level 3 certification. He briefly sketches their ideas focusing on the results and experiences rather than the practices used during the process. In contrast, Pikkarainen and Mantyniemi's [6] approach concentrates on assessing

agile practices using CMMI rather than introducing an improvement process. As the implementations took place as a part of a research case study they describe their CMMI–Agile mapping briefly but not as a comprehensive model. Moreover, they narrow down their mapping to a set of a few selected CMMI areas only.

Although Marçal *et al.* [5] present a detailed description of the CMMI areas coverage provided by Scrum practices, they do not report a validation for their claims. What is more, they do not mention how the areas of insufficient coverage be supplemented. Diaz *et al.* [4] provide a more comprehensive model complemented with a well-documented case study but they concentrate on CMMI level 2 only and their model presents very little additional practices which may help to address all of the CMMI goals level 2 equally well. The same concern applies to a model presented by Fritzsche and Keil [7], which, while being more extensive by exploring all of the CMMI maturity levels and using both Scrum and XP practices, states only which goals and practices are in conflict and does not answer a question how to supplement agile approach. It is worth mentioning that all of the models support CMMI v.1.2 or older.

In our work, we address both the problem of supplementing Scrum practices in an agile way in order to make them fully compliant with CMMI v.1.2 [13] levels 2 and 3 as well as present the process which supports an implementation of appropriate practices.

3 CMMI–Scrum reference model

We propose a reference model mapping specific practices of the second and the third level of maturity in the CMMI Staged representation of the CMMI model onto the activities described by the Scrum methodology. Our reference model (hereafter called the C–S model) is based on the CMMI v.1.2 [13], which was the most current version of CMMI at the time of this research. The newest version at the time of writing this article is CMMI v.1.3 [14]. A detailed list of differences between the two versions can be found in [15].

Exclusion of the fourth and the fifth CMMI level stems from a wish to maintain a compromise between Scrum agility and CMMI discipline. Many researchers have considered the introduction of CMMI 5 while maintaining the agility hardly possible and have believed that CMMI levels higher than the third require some far-reaching compromises that significantly affect the benefits of agile methodologies [5, 10]. Adding more resources and organisational discipline might in fact burden the agility of Scrum when introduced in an oversimplified way. Moreover, effective process improvement towards levels 4 and 5 of CMMI implemented in an agile environment should involve intensive consultancy and cross-project organisational areas.

The C–S model covers 123 practices of CMMI levels 2 and 3 (a list of areas and practices of CMMI v.1.2 can be found in [13]). The practices from the Organisational Process Definition, Organisational Process Focus and Organisational Training areas of CMMI level 3 were excluded from the model. This decision was dictated by the organisational nature of these areas. The C–S model focuses on the attributes of a project and the product without referring directly to the company structure. This follows from the nature of Scrum, which focuses on the process of software development, without referring to the way the company is managed. However, it should be noted that these areas are of great significance for the functioning of the development process and they should also be analysed in a separate study.

In order to create the C–S model, we built upon an in-depth analysis of the practices, the literature and our experience with various development processes. For each of the practices from the selected CMMI areas, we determined to what extent and how well this practice was embraced in the Scrum methodology. We divided the CMMI practices into fully covered, partially covered and not covered at all by the Scrum practices. A CMMI practice is considered to be fully covered when its requirements can be met by using existing Scrum practices. Partial coverage means that Scrum practices should be supplemented with other more disciplined practices in order to meet the requirements of a CMMI practice. CMMI specific practices that are not covered at all by Scrum need a set of entirely new practices.

When it comes to fully covered CMMI practices, the C–S model specifies the appropriate mapping of these practices onto the Scrum practices. For the partially covered CMMI practices, the C–S model specifies the necessary extensions to the Scrum. They include the modifications to the Scrum practices and define some new practices. For the CMMI practices not covered by Scrum, the C–S model defines

Table 1 C–S model statistics

Level	Area	No. of practices	Full	Partial	None
CMMI 2	Configuration management	7	2	4	1
	Measurement and analysis	8	0	0	8
	Project monitoring and control	10	5	4	1
	Project planning	14	9	3	2
	Process and product quality assurance	4	0	0	4
	Requirements management	5	4	1	0
	Supplier agreement management	8	0	0	8
CMMI 3	Decision analysis and resolution	6	0	0	6
	Integrated project management + Integrated product and process development (Ippd)	14	6	2	6
	Product integration	9	8	1	0
	Requirements development	10	7	3	0
	Risk management	7	0	2	5
	Technical solution	8	2	3	3
	Validation	5	4	1	0
	Verification	8	2	6	0
CMMI 2 and 3	Overall	123	49	30	44

new practices. During the design of these extensions our goal was to preserve the agility of Scrum while making way for more maturity and discipline. Therefore in many cases the fact that the CMMI does not specifically define how to maintain the documentation for the artefacts of the practices [5] was beneficial. For instance, it enabled a convenient, agile-friendly approach to the documentation using images, videos, notes etc.

Table 1 shows the number of specific practices in selected areas of CMMI levels 2 and 3 fully covered (column 'Full'), partially covered (column 'Partial') and not covered at all (column 'None') by the Scrum practices.

The most widely covered CMMI areas are Requirements Management, Product Integration and Validation. It is not surprising as agile methodologies were designed to focus on the product and strongly address these particular issues. The least covered CMMI areas are Process and Product Quality Assurance, Supplier Agreement Management, Decision Analysis and Resolution and Risk Management. This also confirms that agile approaches do not explicitly address product quality and most of the managerial tasks.

Tables 2–4 are examples of how the CMMI practices are addressed in the C–S model. The CMMI practice in Table 2 is an example of a group of 49 practices that are fully covered by Scrum (40%), the practice in Table 3 is an example of a group of 30 practices that are partially covered (24%) and Table 4 is an example of a group of 44 practices that are not addressed by Scrum (36%). The "CMMI practice" column gives a brief description of the CMMI practice, the "Scrum practices" column presents corresponding Scrum practices and the "Additional practices" column defines our extensions to Scrum to achieve conformity with the CMMI model. Owing to volume restrictions, in this paper we narrowed down the overview of the C–S model's to these examples. A complete version of the C–S model can be found in [16].

It should be noted that the purpose of the C–S model is not to comprehensively present all practices that might contribute to improving a Scrum project, but to propose a potentially beneficial method of transferring the idea of CMMI practices into Scrum practices and to extend them with the necessary practices to achieve the CMMI compliance while maintaining the idea of agile development.

Table 2 CMMI SP 1.2 Monitor Commitments and its reference to Scrum

CMMI practice	Scrum practices
the purpose of this practice is to monitor the commitment level in comparison to the scheduled one [13]	tasks are distributed to the team members during the Sprint Planning Meetings and are monitored using Burndown Charts and during Daily Scrums as well as Sprint Review Meetings. Stakeholders cannot interfere with the current Sprint tasks [5]

Table 4 CMMI SP 1.4 Monitor Data Management and its reference to Scrum

CMMI practice	Additional practices
the purpose of this practice is to monitor and record the activities of project management [13]	it is advised to store (e.g. in the form of images) the backlogs and charts and to create a space to store all of the documents used by the team (e.g. in a specially prepared application). It is also important to determine at the project initialisation stage the document storage facilities

4 C–S model application

To effectively use our C–S model to improve an actual software development process, we defined a diagnostic questionnaire, a practice selection algorithm, an application process and built a software tool.

The questionnaire consists of 25 single-choice questions with suggested areas, practices and additional activities defined for each answer. Formally, the construction of the questionnaire is as follows. Let Ar denote the set of areas of the C–S model and P denote the set of practices of the C–S model. Let Q denote the set of questions q in our questionnaire, A_i denote the set of answers to ith question q_i from Q. With each answer a_{ij} in A_i, we associate three sets: (i) Ar_{ij} – a set of areas ar suggested from Ar; (ii) P_{ij} – a set of practices p directly suggested from P; and (iii) SA_{ij} – a set of supplementary activities sa, which support and facilitate the implementation of practices from Ar_{ij} and P_{ij} (e.g. training).

Table 3 CMMI SP 1.3 Monitor Project Risks and its reference to Scrum

CMMI practice	Scrum practices	Additional practices
the purpose of this practice is the continuous supervision and inspection of previously identified sources of risk [13]	the risks are written on the whiteboard, flip charts or in the form of a list of difficulties and obstacles and are monitored by the Scrum Master. Therefore they are monitored on an informal basis [5]	during the planning stage of the project the project risks (the list of hazards) should be identified using techniques such as brainstorming and identified hazards should be prominently displayed. During the planning phase and subsequent Sprint Reviews this list should be reviewed and possibly modified or supplemented. Additionally, during the daily meetings a member of the team who reported the impediment should report whether further difficulty arises and has it already been resolved. After the meeting he should write down in one sentence the solution on a relevant table

doi: 10.1049/iet-sen.2011.0193

Table 5 shows a sample question from our questionnaire together with its answers, associated suggested areas and practices of the C–S model as well as supplementary activities.

The results of the questionnaire form two sets: (i) P_{sug} – the suggested practices of the C–S model; (ii) SA_{sug} – the suggested supplementary activities. As the number of practices in P_{sug} can be large, for practical reasons we propose to divide the practices into three subsets: (i) P_{imp} – important practices that should be implemented first; (ii) P_{rec} – recommended practices that should be considered for subsequent implementation; (iii) P_{add} – additional practices that might bring more value to the other practices.

To build the sets P_{imp}, P_{rec}, P_{add} and SA_{sug}, we introduce the P–Sel algorithm (Practice–Selection algorithm). Let Ar_{sug} denote an internal working set of all areas suggested from Ar. Referring to the sets and elements defined earlier, the P–Sel algorithm is defined as follows:

```
0, Ar_sug is empty
1, for each q in Q
2,      if the answer chosen in A_i is a_ij
3,          for each ar in Ar_ij
4,              if ar is in Ar_sug
5,                  add all p associated with ar to P_rec
6,              else
7,                  add all p associated with ar to P_add
8,                  add ar to Ar_sug
9,          add all p in P_ij to P_imp
10,         add all sa in SA_ij to SA_sug.
```

Referring to a sample question presented in Table 5, if the answer is 'Probably not', the given areas Ar_{ij} and practices P_{ij} from the C–S model (Verification SG 2, Product Integration SG 2 SP 2.2 and Technical Solution SG 2 SP 2.1, 2.2, 2.3) are added to Ar_{sug} and P_{imp}, respectively. The given supplementary activities SA_{ij} ('A professional training on the technology should be included in the Product Backlog.') are added to SA_{sug}.

We propose using our C–S model, questionnaire and P–Sel algorithm in the following seven-step process (hereafter called the C–S process):

1. Select a pilot project, in which a new approach combining the Scrum practices and the CMMI model will be implemented.
2. Analyse the selected project to provide a preliminary description, which will include information such as the size of the team, the product characteristics etc.
3. Identify a list of the major problems that occurred earlier in the projects in the company to highlight the weak points that require special attention.
4. Use the questionnaire and the P–Sel algorithm to select appropriate practices from the C-S model according to the analysis in steps 2 and 3. In case of an absence of adequate information, complete the analysis and return to step 4.
5. Analyse selected practices with the list of the problems identified in step 3 and decide whether to implement all recommended practices and additional areas.
6. Implement the set of practices (with possible extensions arising from the specific character of the company) in the pilot project selected in step 1.
7. If successful, implement the practices in other projects. Otherwise go to step 3 and continue with the analysis and selection of practices or go to step 1 and select a different pilot project.

Fig. 1 presents the workflow of the C–S process. The C–S process can be applied in three different situations:

1. A company achieved second or third CMMI maturity level and intends to introduce more agility to the process while maintaining a given level of discipline;
2. A company runs successful Scrum processes and intends to achieve more discipline while maintaining agility;
3. A company has developed and follows its own partly disciplined, partly agile development methodology and intends to improve it with elements of CMMI and Scrum.

In the first case, to preserve the company level of maturity according to the CMMI model, in step 4 the set of suggested practices P_{sug} is extended with all the practices of the areas of the specified maturity level. In the second and third cases, the set of suggested practices P_{sug} is built directly according to the P–Sel algorithm.

Finally, we developed an online software tool called MatureScrum [16]. The tool allows to browse the entire C–S model with a hierarchical table of contents. It also includes the questionnaire and uses the P–Sel algorithm. Unlike other Scrum tools like Agilebuddy, Acunote or Agilo, MatureScrum does not support actual process

Table 5 Sample question with its answers and associated elements

Question		
Does the team have sufficient knowledge of the technologies in which the product will be built?		
answers (a_{ij} in A_i)	CMMI–Scrum areas and practices (Ar_{ij} and P_{ij})	supplementary activities (SA_{ij})
yes, the team will be composed mostly of employees with a very good knowledge of the technology used in the project		
rather yes, the majority of the members of the team has a sufficient knowledge of the technology used, some training would be adequate	• verification	
probably not, most of the staff involved has only a basic knowledge of the technology used	• verification SG 2 • product Integration SG 2 SP 2.2 • technical Solution SG 2 SP 2.1, 2.2, 2.3	a professional training on the technology should be included in the Product Backlog

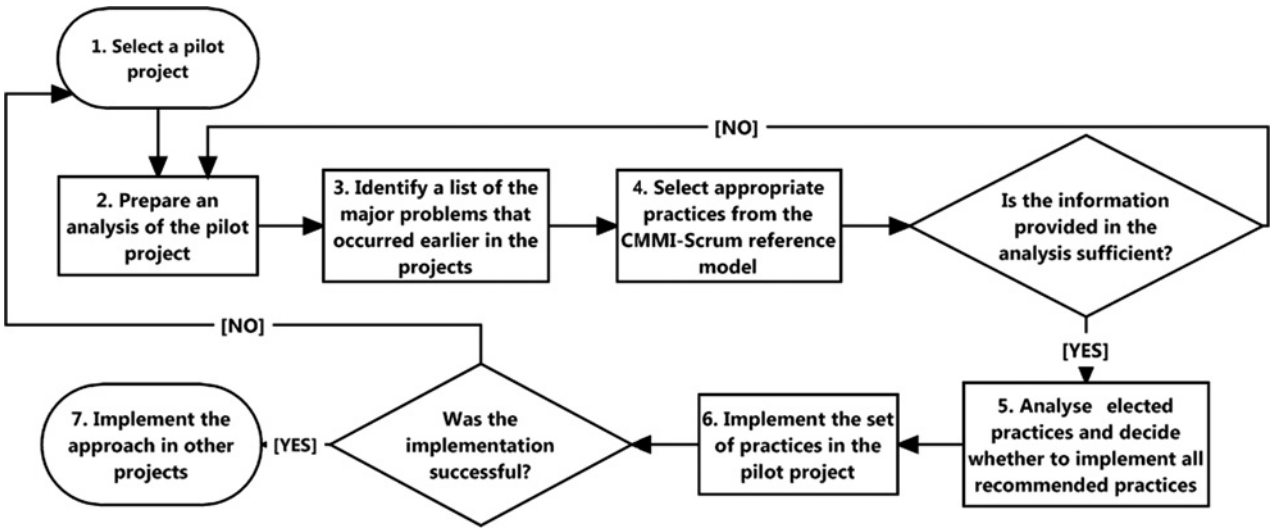

Fig. 1 *Workflow of the C–S model application process*

management. However, the extensions to Scrum practices suggested by MatureScrum can be used to define the extensions to such tools (e.g. Agilo allows to define new types of items and backlogs).

5 Case studies

The goal of the case studies was to assess the suitability of the practices suggested to the improvement of the process. Owing to limited resources and timeframe of the study, it was limited to the assessment of suggestions; the practices were not actually implemented to observe the real improvements.

The case studies have been carried out with two Polish SME IT companies Young Digital Planet (YDP) and MOL, which are potential users of the proposed approach. Software development processes of these companies were analysed and some improvement practices were suggested based on the C–S model and the results of the questionnaire. Then, the key managers of both companies assessed each suggestion of a practice in a four-level linguistic scale of

- 'Interested in' – the practice has not yet been implemented and is considered potentially beneficial, the company wants to implement the practice;
- 'Already implemented' – the company has already implemented the practice and it works well, the benefits of the practice were demonstrated;
- 'It did not work' – the company has already tried to implement the practice, but the results were unsatisfactory and it was abandoned;
- 'Not applicable' – the practice does not apply to the specific structure of the company or it cannot be implemented for reasons specific to the company.

Both 'Interested in' and 'Already implemented' were considered as positive assessments of a practice suggestion. The former means that the practice can be implemented in the company and it is considered potentially beneficial, whereas the latter indicates an already implemented and beneficial practice. 'It did not work' is regarded as a negative assessment of the practice suggestion indicating a false suggestion and demonstrated inadequacy of the practice. The 'Not applicable' assessment is treated as neutral because it means that a suggestion is potentially correct but is inadequate for a particular case as it is not possible to implement the practice within the company and to determine its usefulness. This assessment may also indicate that the questionnaire is not sufficiently detailed or that the answers were incorrect because of insufficient data.

5.1 Young Digital Planet case study

YDP provides advanced information technology solutions mainly in the field of education in Polish and foreign markets. The case study was limited to the Department of Software Development which employs about 40 people. The company is not certified for CMMI and develops software in its proprietary disciplined methodology, which roughly matches the CMMI level 2. The management of the department intended to introduce new disciplined practices as well as some agility.

As the company had not identified a pilot project, we omitted step 1 of the C–S process and examined a typical project of the company (step 2). In step 3, working together with the head of the department and a few employees, we determined the list of the most common problems in projects conducted in the company. The problems mostly concerned incomplete or lacking specification of the product, poor quality of project team meetings, team's low sense of responsibility for the product and clashes in terms of technical analysis.

Using our questionnaire, we analysed the needs of the department. As a result, the practices of the following C–S model areas were recommended (P_{rec}, in brackets the indexes of practices from P_{imp} are given): Configuration Management, Technical Solution (SP 2.1–2.4), Project Monitoring and Control (SP 1.3, 2.3), Project Planning (SP 1.2, 2.2), Measurement and Analysis, Verification (SP 2.1–2.3), Process and Product Quality Assurance, Requirements Management (SP 1.4) and Requirements Development (SP 3.1, 3.5). Practices of three C–S model areas were suggested as additional (P_{add}): Integrated Project Management (SG 1), Validation (SP 1.3) and Product Integration (SP 2.2). Some supplementary activities (SA_{sug}) were also identified. On the basis of these results in step 4 of the C–S process, we selected a list of 20 practices (P_{imp} + selected P_{rec}, P_{add} and SA_{sug}) that were most suited to the specific character of the

doi: 10.1049/iet-sen.2011.0193

company. The list of practices was sent for assessment to the department's managers (step 5 of the C–S process).

In total, 65% of the 20 practices have been evaluated positively. They mostly covered problems found within the company, answering most of the team integrity issues and organising product specification in a convenient agile way. In contrast, 35% of the suggested practices were considered 'Not applicable' mainly because of suggestions going beyond the scope of the department. For example, project managers and product managers are usually employed by the company business department, so the practices concerning a project manager have been dismissed. For similar reasons, the practices concerning the organisation of meetings (Daily Scrums, Sprint Reviews etc.) and methods of identifying risks and impediments in the project were also assessed as 'Not applicable'. In turn, the practices regarding failure analysis and code testing, in particular their measurements and metrics, were recognised as 'Interested in'. There were no practices assessed as 'It did not work', which would have showed false suggestions of the model. Fig. 2 presents a summary of the assessment made by YDP. So far, steps 6 and 7 of the C–S process have not been implemented.

5.2 MOL case study

The MOL company is currently the largest provider of software for libraries in Poland. It is a typical SME with seven programmers employed in the software development department. The development processes are more agile than in YDP; however, they do not match directly the Scrum methodology. Currently, MOL is planning to implement the second level of the CMMI.

Owing to a lack of appropriate pilot project at the time of the case study (steps 1 and 2 of the C–S process), the MOL owner was asked to prepare the list of major problems that had occurred in the company projects (step 3). Then, the MOL owner answered our questionnaire using the MatureScrum tool, which resulted in a set of suggested practices P_{sug} and supplementary activities SA_{sug} from the C–S model (step 4).

The owner of MOL assessed both the questionnaire and the suggested practices (step 5). The questionnaire has been regarded as relevant to the company situation. Most of the suggested practices had a reference to projects in the company. In total, 107 practices from areas such as Requirements Management, Product Integration and Validation were suggested in P_{sug}. According to our P–Sel algorithm, this unmanageably large number of practices, was divided into important, recommended and additional. Among them, 37 practices were important or recommended (P_{imp} and P_{rec}) and 70 practices were additional (P_{add}). This much more practical

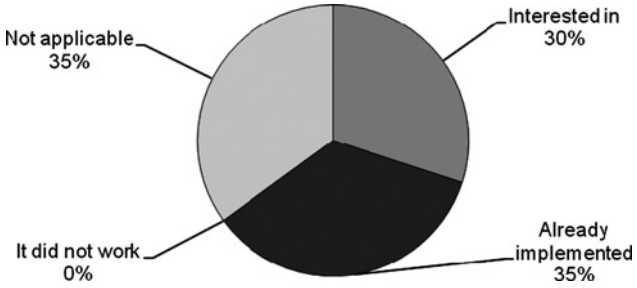

Fig. 2 *Results of the evaluation by Young Digital Planet*

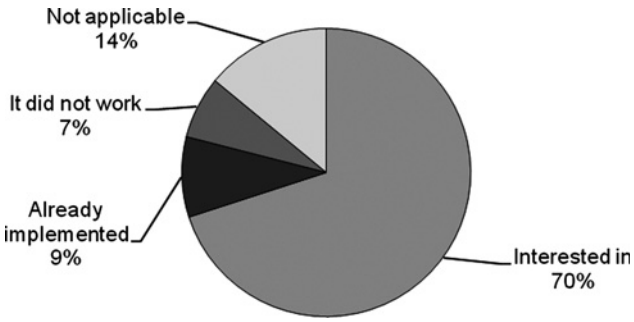

Fig. 3 *Results of the evaluation by the MOL*

list of 37 important and recommended practices should be analysed and implemented first.

The detailed assessment of all the suggested practices resulted in 79% of the practices evaluated positively ('Interested in' or 'Already implemented'), 14% of them evaluated as 'Not applicable' (majority of them from Measurement and Analysis, Project Monitoring and Control, Project Planning, Decision Analysis and Resolution areas), and 7% of the practices evaluated negatively ('It did not work'). Negatively evaluated practices had already not worked in the company because of insufficient cooperation with customers and the lack of economic benefits (e.g. pair programming or Process and Product Quality Assurance practices). The summary of the assessment of the suggested practices by MOL is presented in Fig. 3.

In addition, the MOL owner was asked to confront the list of problems prepared in step 3 with the set of practices P_{sug}. He evaluated that the suggested practices P_{sug} addressed 75% of the problems from the list. Nevertheless, in order to completely verify this number, the practices would have to be implemented and their positive impact measured after some time. In general, the MOL owner considered our C–S model 'very useful'. So far, no feedback concerning implementation of the suggested practices in the pilot project has been obtained.

6 Conclusions

The paper presented a new C–S reference model (the C–S model) together with the application process involving a proprietary questionnaire, the P–Sel practice selection algorithm and the MatureScrum software tool. Our C–S model combines widely adopted maturity model – CMMI and currently the most popular agile methodology – Scrum. Our questionnaire and P–Sel practice selection algorithm are designed to identify and preserve the existing level of agility and maturity while providing suggestions for improvements.

Our approach was used by two IT companies utilising different mixes of CMMI and Scrum practices. In both, the suggestions from our approach were mostly valuable as, on average, 72% of the suggested practices were regarded as already or potentially beneficial. Only 3.5% of the suggested practices were rejected by the companies. Some 24.5% of suggestions were evaluated as not applicable for organisational or economic reasons, which leaves room for further improvement of our questionnaire. This shows that our solution is capable of supporting organisations with different levels of maturity and agility. The companies used the approach with little assistance, in one case even without a software tool. The collected data show that the approach to a large extent provides valuable suggestions for

successfully combining agility of Scrum practices with CMMI discipline. Naturally, final confirmation of its effectiveness would require the actual implementation of the suggested practices and the measurement of the added value.

Comprehensive definition of our C–S model, the questionnaire and the MatureScrum tool as well as the full results of the case studies are presented in [16].

7 References

1 Capability Maturity Model Integration. Available at http://www.sei.cmu.edu/cmmi, accessed May 2010

2 Agile manifesto. Available at http://www.agilemanifesto.org, accessed March 2010

3 Schwaber, K., Beedle, M.: 'Agile software development with Scrum' (Prentice Hall, 2001)

4 Diaz, J., Garbajosa, J., Calvo-Manzano, J.A.: 'Mapping CMMI level 2 to Scrum practices: an experience report', in O'Connor, R., Baddoo, N., Cuadrado Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (Eds.): 'EuroSPI 2009. Communications in computer and information science' (Springer, Heidelberg, 2009, vol. 42), pp. 93–104

5 Marçal, A.C., de Freitas, B.C., Furtado Soares, F.S., Furtado, M.S., Maciel, T.M., Belchior, A.D.: 'Blending Scrum practices and CMMI project management process areas', *Innov. Syst. Softw. Eng.*, 2008, **4**, (1), pp. 17–29

6 Pikkarainen, M., Mantyniemi, A.: 'An approach for using CMMI in agile software development assessments: experiences from three case studies'. Proc. of SPICE, Luxembourg, May 2006

7 Fritzsche, M., Keil, P.: 'Agile methods and CMMI: compatibility or conflict?', *e-Informatica Softw. Eng. J.*, 2007, **1**, (1), pp. 9–26

8 Babuscio, J.: 'How the FBI learned to catch bad guys one iteration at a time', in Dubinsky, J., Dyba, T., Adolph, S., Sidky, A. (Eds.): 'Agile Conference 2009' (IEEE Computer Society, Los Alamitos, 2009), pp. 96–100

9 Glazer, H., Dalton, J., Anderson, D., Konrad, M., Shrum, S.: 'CMMI or agile: why not embrace both!', Technical Note CMU/SEI-2008-TN-003, (Software Engineering Institute, 2008)

10 Potter, N., Sakry, i.M.: 'Implementing Scrum (agile) and CMMI together', *Process Group*, 2009, **16**, (2), Available at http://www.itmpi.org/assets/base/images/itmpi/Potter-ScrumCMMI.pdf

11 Sutherland, J., Jakobsen, C., Johnson, K.: 'Scrum and CMMI level 5: the magic potion for code warriors', in Eckstein, J., Maurer, F., Davies, R., Melnik, G., Pollice, G. (Eds.): 'Agile Conference 2007' (IEEE Computer Society, Los Alamitos, 2007), pp. 272–278

12 Vriens, S.: 'Certifying for CMM level 2 and ISO9001 with XP@Scrum'. Proc. of the Agile Development Conference June 2003, IEEE Computer Society Washington, USA, pp. 120–124

13 Capability Maturity Model Integration v 1.2. Available at http://www.sei.cmu.edu/library/abstracts/reports/06tr008.cfm, accessed May 2010

14 Capability Maturity Model Integration v 1.3. Available at http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm, accessed October 2011

15 The complete set of changes to the models moving from CMMI Version 1.2 to CMMI Version 1.3. Available at http://www.sei.cmu.edu/cmmi/tools/cmmiv1-3/upload/CMMI-DEV-v1-3-compare.pdf, accessed October 2011

16 Bulska, K.: 'Integracja zwinnych metodyk wytwarzania oprogramowania z metodami kontroli dojrzałości procesu – wspomaganie wyboru odpowiednich praktyk ('Integration of the agile software development methodologies with maturity models – good practices assistant')'. MSc thesis, Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Gdansk, 2010

doi: 10.1049/iet-sen.2011.0193