Influence of Software Product Management Maturity on Usage of Artefacts in Agile Software Development

Gerard Wagenaar[1(✉)], Sietse Overbeek[2], Garm Lucassen[2], Sjaak Brinkkemper[2], and Kurt Schneider[3]

[1] Avans University of Applied Sciences, Breda, The Netherlands
g.wagenaar@avans.nl
[2] Utrecht University, Utrecht, The Netherlands
{s.j.overbeek, g.lucassen, s.brinkkemper}@uu.nl
[3] Leibniz Universität Hannover, Hanover, Germany
kurt.schneider@inf.uni-hannover.de

Abstract. ***Context***: Agile software development (ASD) uses 'agile' artefacts such as user stories and product backlogs as well as 'non-agile' artefacts, for instance designs and test plans. Rationales for incorporating especially non-agile artefacts by an agile team mainly remain unknown territory. ***Goal***: We start off to explore influences on artefacts usage, and state our research question as: To what extent does maturity relate to the usage of artefacts in ASD in software product organizations? ***Method***: In our multiple case study 14 software product organizations were visited where software product management maturity was rated and their artefacts usage listed. ***Results***: We found maturity to be negatively correlated with the non-agile/all artefacts ratio. In other words, the more mature software product management is, the fewer non-agile artefacts are used in ASD. ***Conclusions***: This suggests that an organizational factor influences an agile team in its artefacts usage, contradictory to the concept of self-organizing agile teams.

Keywords: Agile · Artefacts · Maturity · Software product management

1 Introduction

Agile software development (ASD) has been introduced in the domain of product software development [1, 2], with product software defined as: "*A packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded in a specific market*" [3, *p. 534*], where auxiliary materials consist of software documentation, user material and the like. Product software differs from tailor-made software in, among other aspects, the importance of architecture [3]. The necessity of auxiliary materials and the requirement of a future-proof architecture are indicative for the use of documentation artefacts in the product software development lifecycle. Research in ASD has devoted attention to the usage of artefacts, where a distinction can be made between 'agile' and 'non-agile' artefacts. Agile artefacts are artefacts which are inherent to an ASD (for instance user

M. Felderer et al. (Eds.): PROFES 2017, LNCS 10611, pp. 19–27, 2017.
https://doi.org/10.1007/978-3-319-69926-4_2

stories or a backlog); all other artefacts are considered to be non-agile (for instance architectures or designs). In agile product software development, software product organizations (SPOs) as manufacturers of such software could be expected to use non-agile artefacts precisely because of their needs with regard to architecture and auxiliary materials. In this research we explore one influencing factor on the decision to use, especially non-agile, artefacts. To this extent we assume that artefacts usage is a quality consideration and relates to the quality of software product management (SPM) in an SPO, where software product management is the discipline and role, which governs a product from its inception to the market/customer delivery in order to generate biggest possible value to the business [4]. In the Capability Maturity Model Integration for Development (CMMI-DEV), documentation artefacts, for instance architecture documentation and design data, are explicitly mentioned and they contribute to achieving higher maturity levels [5].

To explore influencing factors we formulate our research question as: ***To what extent does SPM maturity relate to the usage of artefacts in ASD?***

Fourteen organizations were visited as part of a multiple case study. Our findings show a negative correlation between SPM maturity and the usage of non-agile artefacts. Altogether our findings contribute to a better understanding of factors that influence an agile team in its artefacts usage, an area in which research is scarce. From a practitioner's perspective one of the principles behind the agile manifesto, "The best architectures, requirements, and designs emerge from self-organizing teams" [6], is put to the test if an organizational factor can be shown to relate to artefacts usage.

The remainder of this paper is organized as follows. In Sect. 2 we outline the theoretical background with an overview of research on the usage of artefacts in ASD (Sect. 2.1), and SPM in general and a method to establish its maturity in particular (Sect. 2.2). In Sect. 3 we present our research method, a multiple case study, including data collection and coding, leading to our findings. Section 4 discusses our conclusions, which may be summarized as a new insight in the relation between SPM maturity in SPOs and the usage of non-agile artefacts in ASD in SPOs.

2 Theoretical Background

2.1 Artefacts in Agile Software Development

An artefact (in ASD) is defined as a tangible deliverable produced during software development [7]. In ASD artefacts, such as architectures, requirements, and designs, are used as a decision of the self-organizing ASD team [6], dependent on the value it attaches to them. ASD practitioners perceive their internal documentation as especially important but feel that too little of it is available [8]. A decision on usage of artefacts is in fact a decision on 'non-agile' artefacts, because agile artefacts already are part of an ASD method itself. Previous research shows the dilemma of the optimal level of agile and non-agile artefacts in ASD. Gröber [9] constructed an (agile) artefact class diagram with artefacts and relationships between them as result of a systematic literature study on the usage of artefacts in agile methods. Based on this research and adding findings from three case studies Wagenaar et al. [7] developed a Scrum artefact model,

distinguishing product from process artefacts and Scrum from non-Scrum artefacts. In a study on large-scale offshore software development programmes Bass [10] identified 25 artefacts on five levels of abstraction: Programme governance, Product, Release, Sprint, and Feature.

In summary, various models show a mixture of agile and non-agile artefacts, although based on different viewpoints varying from agile or Scrum development to offshore software development. The models classify both agile and non-agile artefacts, but, with one exception, do not explicitly address the distinction between the two. This precludes, as one consequence, insight in reasons for using them.

2.2 Software Product Management Maturity

Assessing maturity of software development processes and thus contributing to their improvement has led to several maturity models. General ones, like CMM [11], or ISO/IEC 15504 [12] and more specialized agile models are all composed of hierarchical maturity levels, but are otherwise quite different in their domains, backgrounds, structures, and contents [13]. However, because of our focus on SPM a more dedicated model, but similar in its constitution, is available, which describes the SPM process as consisting of four business functions: Portfolio management, Product planning, Release planning, and Requirements management [14, 15]. Each business function is in turn divided into focus areas. In case of the business function Requirements management, these are: Requirements gathering, Requirements identification, and Requirements organizing. The model has an associated method, the Situational Assessment Method (SAM), which can be used to measure a maturity level specifically for SPM [15]. To this extent the SAM provides a matrix with an overview of capabilities at different levels that need to be implemented to reach a full-grown maturity. The matrix is used in a bottom up way. Maturity is ranked per focus area, and then aggregated to SPM maturity on a scale from 0–10.

3 Research Method

To investigate our research problem, we used a multiple case study, which is an accustomed way to investigate phenomena in a context where events cannot be controlled and where the focus is on contemporary events [16]. Data collection took place through single-site case studies following Yin's widely accepted guidelines for case studies [16]. We first collected basic data on artefacts and maturity on basis of a protocol including: (1) SPM theory and research, (2) interview instructions, and (3) reporting guidelines. We found 14 organizations willing to participate, all using ASD[1]. The organizations develop product software (1) for a broad range of domains, from (semi-)government to software development, (2) with five to over hundred employees (organization as a whole), and (3) for ten to several thousands of customers.

[1] A description of the organizations is available at https://osf.io/dez9k/?view_only=3171388053194c549f09b22fe4fbcfc0.

In case a SPO produced more than one product, one of them was selected. In Sect. 3.4 we discuss threats to external validity regarding our participating organizations.

3.1 Data Collection

Data collection was the same for all organizations. Interviews were held, ranging from one interview through one interview with two interviewees to two or more interviews with several interviewees. Interviewees were in general product manager or owner, although some Scrum masters were also included. Interviews lasted on average one hour. They were semi-structured to allow interviewees to speak freely and to be able to ask follow-up questions. The interview instructions concerned two tasks: (a) Determine SPM maturity, and (b) Describe artefacts during ASD[2].

For the establishment of maturity, a description of capabilities required to achieve a certain maturity level is already provided in the SAM [14]. For each capability the organization being assessed has to answer the question "*Have you implemented this capability within your organization?*" with either Yes or No, for example: "*Can stakeholders submit requirements directly to the central database?*".

For the listing of artefacts the interview guidelines were based on the life cycle of a user story or a requirement, starting with the SPM's pre-development stage (portfolio management, product planning). Then they continued with questions about the activities in ASD, often starting with user stories in a product backlog and ending with the production of source code. Finally, post-development activities were identified, such as bugs, again leading to requirements. For the description of this life cycle a common vocabulary was established by using the FLOW modelling technique [17, 18]. FLOW's emphasis on information and its distinction between solid and fluid information makes it suitable for the representation of artefacts. Documented information is called solid information if it is long term accessible, repeatedly readable, and comprehensive for third parties. In contrast, undocumented or fluid information is information that violates any one of the above criteria.

3.2 Coding

Data on maturity needed no further coding, because answers to questions from the SAM directly translate to a maturity level for each focus area (see Sect. 2.2).

Data analysis for artefacts started by extracting solid information as artefacts from the FLOW models, identifying 201 artefacts. Because of differences in SPO's terminology this initial list was subject to: (1) lexical analysis, and (2) semantic analysis [20]. In lexical analysis we removed distinctions in singular and plural forms, for instance 'Bug report' (listed 5 times) and 'Bug reports' (1 appearance). We removed adjectives, for instance mapped both 'Product roadmap' and 'Company annual roadmap' on 'Roadmap', and we unified words having the same lexical roots, for instance 'Acceptance criteria' and 'Acceptation criteria'. This reduced the number of 201 to 123

[2] Interview instructions are available at https://osf.io/dez9k/?view_only=3171388053194c549f09b22fe4fbcfc0.

artefacts. In further semantic analysis we used the description of solid information in the FLOW model to identify similarities and differences in artefacts. Based on this description and also guided by the artefact model [7] and the artefact list [10] we further pruned our list. For instance, 'Application' with description "*Code implemented by developers based on the release and sprint plan*" and 'Code', "*A (set of) implemented and unit-tested product feature(s)*", were mapped.

Finally we excluded a number of artefacts, since not all artefacts in our findings are artefacts directly related to ASD. Since our interviews used the pre-development stage as starting point we identified some 'Business Artefacts': Business case, Business plan, Market intelligence, Market requirement, Strategy, and Roadmap. An SPO's strategy certainly influences decisions with an impact on ASD, but it is not an ASD artefact. Business artefacts are important SPM artefacts, but are neither produced nor used directly by an agile team.

3.3 Findings

We aggregated maturity in a maturity level per business function where this maturity is calculated as the average maturity of underlying focus areas (Table 1)[3]. For example, the focus areas 'Gathering', 'Identification', and 'Organizing' within the business function 'Requirements management', scored 7, 9, and 10 respectively for organization A. This results in (7 + 9 + 10)/3 = 8.7 for 'Requirements management' for organization A. The last row shows the overall SPM maturity as the average of the four business functions.

Table 1. Maturity of SPM

	SPO	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Maturity	Requirements management	*8.7*	*4.3*	*5.0*	*7.3*	*4.3*	*5.7*	*7.0*	*5.7*	*4.7*	*5.3*	*4.7*	*4.7*	*4.7*	*10.0*
	Release planning	*6.3*	*3.8*	*9.0*	*5.8*	*5.3*	*6.5*	*8.5*	*7.2*	*5.3*	*7.5*	*3.7*	*7.3*	*6.0*	*7.2*
	Product planning	*5.0*	*3.7*	*5.3*	*6.7*	*6.7*	*8.3*	*10.0*	*6.0*	*7.0*	*6.7*	*5.3*	*4.0*	*8.0*	*5.7*
	Portfolio management	*5.0*	*5.0*	*7.0*	*8.0*	*8.7*	*8.3*	*8.7*	*7.0*	*7.0*	*4.3*	*5.3*	*4.3*	*5.7*	*7.3*
	Overall SPM maturity	6.25	4.21	6.58	6.96	6.25	7.21	8.54	6.46	6.00	5.96	4.75	5.08	6.09	7.54

We found a total of eighteen artefacts, which were mentioned by at least two organizations (Table 2). The one but last row in Table 2 lists the number of artefacts (per organization) which were mentioned by that organization only. Since they tend to be rather organization-specific we have aggregated them in this way.

[3] Scores per focus area are available at: https://osf.io/dez9k/?view_only=3171388053194c549f09b22fe4fbcfc0.

Table 2. Artefacts per SPO

Artefact	A	B	C	D	E	F	G	H	I	J	K	L	M	N
User story	v	v	v			v	v		v	v	v	v	v	v
Code	v	v		v	v	v	v		v	v	v			
Sprint backlog		v	v	v		v		v	v				v	v
Epic	v		v			v	v						v	v
Product backlog		v	v	v		v		v						v
Definition of done	v	v				v		v					v	
Estimated user story		v		v							v			
Agile artefacts	*4*	*6*	*4*	*4*	*1*	*6*	*3*	*3*	*3*	*2*	*3*	*1*	*4*	*4*
Product requirement	v		v		v				v	v	v		v	v
Bug report	v				v				v	v	v		v	
Release note		v	v		v		v		v		v			
Test deliverables		v			v		v		v			v		
Request for change		v			v	v						v	v	
Acceptance criteria	v					v					v			
Release	v				v					v				
Functional design		v										v		
Release plan					v				v					
Technical design		v									v			
User documentation			v				v							
Non-agile artefacts	*4*	*5*	*3*	*0*	*7*	*2*	*3*	*0*	*5*	*3*	*5*	*3*	*3*	*1*
Organization-specific	*5*	*2*	*3*	*2*	*1*	*2*	*1*	*0*	*1*	*0*	*0*	*6*	*2*	*0*
Non-agile ratio	***0.69***	***0.54***	***0.60***	***0.33***	***0.89***	***0.40***	***0.57***	***0.00***	***0.67***	***0.60***	***0.63***	***0.90***	***0.56***	***0.20***

Artefacts in Table 2 are also classified in one of two categories: (1) Agile artefacts, and (2) Non-agile artefacts, since we are especially interested in the usage of additional, non-agile artefacts. We identified 'Agile artefacts' as: Product backlog, Sprint backlog, Code, User story, Epic, Definition of done, and Estimated user stories, because those are explicitly part of agile practices [20, 21]. Various artefacts all are non-agile artefacts. To be able to compare between organizations we calculated the ratio of non-agile artefacts compared to the total number of artefacts.

Our research question was: To what extent does SPM maturity relate to the usage of artefacts in ASD? We identified both SPM maturity (Table 1) and the usage of ASD artefacts (Table 2). A measure of correlation dependency between two variables is the Pearson correlation coefficient [22]. We calculated it between SPM maturity and non-agile artefacts ratio as $\rho(14) = -0.3576$. This outcome is considered to be of a weak to moderate strength. The answer to our research question thus is: SPM maturity is negatively correlated with the non-agile/all artefacts ratio. In other words, the more mature SPM is, the fewer non-agile artefacts are used in ASD.

3.4 Validity

Validity of our research depends on four criteria: Construct validity, internal and external validity, and reliability [16]. To enhance construct validity we: (1) had interviewees comment on results of interviews, (2) complemented viewpoints in the interviews with more than one interviewee, and (3) followed a strict procedure in

interpreting an interview, by means of the FLOW modelling technique as well as in applying the SAM. Nevertheless, organizations were not visited by one and the same interviewer, so interpretation may have influenced especially the listing of artefacts. Additionally, the maturity level is based on self-assessment, which may introduce bias. Internal validity is mainly a concern for explanatory case studies, but we did apply pattern matching in translating solid information in the models through lexical and semantic analysis to our artefacts listing. External validity benefits from using a multiple case study on the basis of a common procedure. It has to be noted however, that our results only show a weak to moderate correlation. Furthermore, we visited SPOs, which was also reflected in our choice for measuring maturity. Generalizability to non-SPOs is therefore limited. From our findings organizations E and L show remarkable ratios, using (far) more non-agile artefacts than agile ones. This may be reason to question their application of indeed an ASD method. Finally, reliability increases because of our use of a procedure with interview instructions and the use of a case study database.

4 Conclusions and Future Research

We rated SPM maturity for 14 organizations and listed their artefact usage. We found evidence for SPM maturity to be negatively correlated with the non-agile/all artefacts ratio. A possible explanation could be that a 'mature' SPO' has organized its software product management already in such a way that additional documentation during ASD is hardly required, but further research should be carried out to prove this. Although a causal relationship has not been proven, our evidence suggests that an organizational factor – maturity in SPM – influences an agile team in its usage of artefacts. This would be quite contradictory to self-organizing teams, from which the best architectures, requirements, and designs emerge. Our research goes beyond the sole modelling of artefacts and provides initial knowledge about factors that influence agile teams in their artefacts usage.

Our research also strengthens empirical evidence with regard to the usage of artefacts in ASD. Our current findings confirm both artefacts that appeared in the artefact list [9], but not in the Scrum artefact model [7], as well as vice versa. The relatively small yield of 'new' artefacts, proves an already high degree of coverage in the research on the existence of artefacts in ASD.

Further research is necessary, not only to prove a causal relationship between (SPM) maturity and artefacts usage, but also to identify other factors influencing agile teams in their choice for (non-agile) artefacts. Candidates are team composition (size, experience), project characteristics or explicit team decisions as opposed to maturity of an organization as a whole. This would provide an answer to the question whether, especially non-agile, artefacts emerge from an agile team or are used for reasons which originate from outside the team. More general, how does an agile team reach a balance between agile and non-agile artefacts?

Acknowledgements. The authors express their gratitude to the students and organizations involved in this research; the first author also expresses his gratitude to Avans University of Applied Sciences for facilitating and supporting this research.

References

1. Dzamashvili Fogelström, N., Gorschek, T., Svahnberg, M., Olsson, P.: The impact of agile principles on market-driven software product development. J. Softw. Maint. Evol. Res. Pract. **22**(1), 53–80 (2010)
2. Vlaanderen, K., Jansen, S., Brinkkemper, S., Jaspers, E.: The agile requirements refinery: applying SCRUM principles to software product management. Inf. Softw. Tech. **53**(1), 58–70 (2011)
3. Xu, L., Brinkkemper, S.: Concepts of product software. Eur. J. Inf. Syst. **16**(5), 531–541 (2007)
4. Ebert, C.: The impacts of software product management. J. Syst. Softw. **80**(6), 850–861 (2007)
5. CMMI Product Team: CMMI for development, version 1.3. resources.sei.cmu.edu/library/asset-view.cfm?assetID=9661 (2010). Accessed 04 Sept 2016
6. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., et al.: Agile Manifesto. agilemanifesto.org/principles (2001). Accessed 24 Sept 2012
7. Wagenaar, G., Helms, R., Damian, D., Brinkkemper, S.: Artefacts in agile software development. In: Abrahamsson, P., Corral, L., Oivo, M., Russo, B. (eds.) PROFES 2015. LNCS, vol. 9459, pp. 133–148. Springer, Cham (2015). doi:10.1007/978-3-319-26844-6_10
8. Stettina, C.J., Heijstek, W.: Necessary and neglected? An empirical study of internal documentation in agile software development teams. In: Proceedings SIGDOC, pp. 159–166. ACM, New York (2011)
9. Gröber, M.: Investigation of the usage of artifacts in agile methods. www4.in.tum.de/~kuhrmann/studworks/mg-thesis.pdf (2013). Accessed 23 May 2016
10. Bass, J.M.: Artefacts and agile method tailoring in large-scale offshore software development programmes. Inf. Softw. Tech. **75**, 1–16 (2016)
11. Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V.: Capability maturity model SM for software, version 1.1. sei.cmu.edu/reports/93tr024.pdf (1993). Accessed 24 June 2013
12. ISO/IEC 33002:2015. iso.org/standard/54176.html (2015). Accessed 11 Aug 2016
13. Leppänen, M.: A Comparative Analysis of Agile Maturity Models. In: Pooley, R., Coady, J., Schneider, C., Linger, H., Barry, C., Lang, M. (eds.) Information Systems Development, pp. 219–230. Springer, New York (2013). doi:10.1007/978-1-4614-4951-5_27
14. Bekkers, W., Brinkkemper, S., van den Bemd, L., Mijnhardt, F., Wagner, C., Van de Weerd, I.: Evaluating the software product management maturity matrix. In: Proceedings RE, pp. 51–60. IEEE (2012)
15. Bekkers, W., Spruit, M., Van de Weerd, I., van Vliet, R., Mahieu, A.: A situational assessment method for software product management. In: Proceedings ECIS, paper 22 (2010)
16. Yin, R.K.: Case Study Research: Design and Methods, 4th edn. Sage Publications, Thousand Oaks (2009)
17. Stapel, K., Knauss, E., Schneider, K.: Using FLOW to improve communication of requirements in globally distributed software projects. In: Proceedings CIRCUS, pp. 5–14. IEEE (2009)

18. Stapel, K., Schneider, K.: Managing knowledge on communication and information flow in global software projects. Expert Syst. **31**(3), 234–252 (2014)
19. Jurafsky, D., Martin, J.H.: Speech and Language Processing, 2nd edn. Prentice Hall, Englewood Cliffs (2008)
20. Schwaber, K., Sutherland, J.: The Scrum guide™. www.scrumguides.org/scrum-guide.html (2016). Accessed 25 May 2017
21. Cohn, M.: User Stories Applied for Agile Software Development. Addison Wesley, Boston (2004)
22. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29044-2