Agile and FBM: A Match Made in Heaven

Inge Lemmens[1]([⊠]) and Rob Arntz[2]

[1] PNA, Geerstraat 105, 6411 NP Heerlen, The Netherlands
Inge.lemmens@pna-group.com
[2] i-Refact, Bruistensingel 138, 5232 AC 's Hertogenbosch, The Netherlands
Rob.artnz@i-refact.nl

Abstract. The agile way of working is often abbreviated to the principle: "working software over comprehensive documentation", which is interpreted as "not need for documentation at all". Looking carefully at the Agile manifesto, one also discovers the principle "continuous attention to technical excellence and good design enhances agility". In this paper, we will cover the Agile principles and values mentioned in this manifesto and demonstrate how well fact-based modelling fits these principles and values.

Keywords: Fact-based model · Agile · Agile manifesto

1 Introduction

Agile represents a group of software engineering methodologies which aim to deliver increased productivity, quality and project success rate in software development project when compared to more traditional methodologies. Methodologies that belong to the group are, amongst others, SCRUM [1], XP [2] and Crystal [3].

In [4] the following is stated: "What is new about agile methods is not the practices they use, but their recognition of people as the primary drivers of project success, coupled with an intense focus on effectiveness and manoeuvrability". That is, agile methods are highly people-oriented with focus to adaptivity.

Adaptability is also a key difference between the more traditional methodologies like waterfall and agile methodologies: in an agile methodology, if any major change is required, the work is not "frozen". Instead, the team determines how to handle the changes that occur throughout the project. This in contrast to more traditional methodologies where product requirements are frozen and a predictive approach is taken in terms of planning and deliverables. Traditional methodologies are therefore often referred to as "heavyweight methodologies" as they do not facilitate for change.

1.1 Agile Methods Are People-Oriented

All agile methodologies place emphasis on the social aspects of software development. The people (customers, developers, stakeholders, end users) are considered to be the most important factor for success. In [5] it is stated that the most important implication for managers working in the agile manner is that emphasize is to be given on people factors in a project: amicability, talent, skill and communication.

Close collaboration, with customer feedback on a regular and frequent basis is considered essential. As stated in [6]: *"Agile teams cannot exist with occasional communication. They need continuous access to business expertise"*.

1.2   Agile Methods Are Adaptive

Agile teams are not afraid of change: changes are allowed all stages of the project and are considered to be a good thing since changes in requirements means that the team has learned more about what it is required.

The agile methods are thereby more adaptive than predictive, whereby the challenge is not to stop the change but to handle the change in a cost-effectively manner [4]. This has also an effect on the planning strategy used, namely: only detailed plans for the next few weeks, very rough plans for the next few months and extremely crude plans beyond that [7].

According to Fowler [6], being adaptive and people-oriented is the essence of any agile methodology. Adaptive entails not that no plans are made: agile plans a baseline that is used to control change. People-oriented does not mean that there are no processes in place, but means that processes and tools are only needed to enhance the team's effectiveness.

1.3   Fact-Based Modeling

Fact-based modeling is a methodology for modelling the semantics of a subject area. It is a semantically-rich conceptual approach based on logic and controlled natural language. The result of applying the methodology is a fact-based conceptual data model that captures the semantics of the domain by means of fact types, which are attribute-free structures.

Fact-based modelling relies heavily on interaction with the domain experts as the guiding principles of fact-based modelling are the use of concrete examples, as provided by the domain expert, to determine the fact types. Moreover, fact-based modelling relies heavily on communication as the results are expressed in controlled natural language.

A fact-based conceptual model specifies the semantics of the information relevant to the domain: it defines the data of significance to the end-users, including its characteristics, the relationships between the data, the meaning of the data and the rules that apply. It is described implementation-independent and can be the basis for deriving logical and physical data models [8].

Fact-based modeling is a conceptual modelling methodology, meaning that the resulting model is free of any implementation-bias. As described in [9], for correctness, clarity and adaptability, information systems are best specified first at the conceptual level using concepts and language that people can readily understood.

In the remainder of this paper, we will focus on the values and principles of the agile movement and relate these to the fact-based modeling methodology as to demonstrate that fact-based modeling and the Agile movement are a match made in heaven.

2   The Manifesto for Agile Software Development: Its Values

The Agile Manifesto was written in 2001 by 17 independent developers with different backgrounds. Although these developers disagreed on much, they were able to come to an agreement on four values and twelve principles which form the foundation of the Agile movement.

The manifesto for Agile Software Development states the following: *"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more" [10].

2.1   The Value: Individual and Interaction over Processes and Tools

The first value entails the recognition of people as the primary drivers for project success. Emphasis is to be placed on teamwork and *communication*. That is, communication is a prerequisite for successful application of an agile method. Hereby, communication is not limited to communication between the developers, but also the communication between the end-users and developers.

Communication is also the central principle for fact-based modeling. In particular, the fact-based modeling methodology adheres to the Helsinki principle [12]: *"Any meaningful exchange of utterances depends upon the prior existence of an agreed set of semantic and syntactic rules. The recipients of the utterances must use only these rules to interpret the received utterances, it is to mean the same as that which was meant by the utterer"*.

In other words, the fact-based modelling methodology relies heavily on communication to achieve its goal.

2.2   The Value: Working Software over Comprehensive Documentation

The second value is based on the idea that agile teams always take the simplest path that is consistent with their goals. The reason for simplicity is so that it is easy to change the design and application on a later date. That is, the developers are urged to keep the code simple and straightforward such that there is no need for extensive documentation. As described in [11]: "the larger the amount of documentation becomes, the more effort is needed to find the required information, and the more effort is needed to keep the information up to date".

Fact-based modeling is a model-driven approach to system development. Model-Based Systems Engineering is defined as: *"the formalized application of modelling to support system requirements, design, analysis, verification and validation activities,*

beginning in the conceptual design phase and continuing throughout development and later lifecycle phases." [13]. In [8], it is explained how a conceptual model developed using fact-based model can form the basis for logical and physical models, and illustrated with a concrete example in [15]. In [14], a concrete example of the model-driven approach is given. As demonstrated in that paper, the conceptual model, its underlying structure and the rules associated are the basis for both development of the application repository as well as the definition of the user interface.

A fact-based model is not only the application, it is also at the same time the documentation of the application, since a fact-based model complies to the ISO-100% principle which reads [12]: *"All relevant general static and dynamic aspects, i.e. all rules, laws, etc., of the universe of discourse should be described in the conceptual schema. The information system cannot be held responsible for not meeting those described elsewhere, including in particular those in application programs".* As such, fact-based modeling aids in taking away the extensive documentation burden.

2.3   The Third Value: Customer Collaboration over Contract Negotiation

Customer collaboration is valued over extensive contract negotiations, meaning that more value is attached to the continuous collaboration of the customer throughout the complete development of the software.

One of the most significant barriers to implementing an agile methodology, however, lies in the inability to establish and maintain close and effective customer collaboration [16]. Even when the relationship begins with the best intentions and the highest expectations on both sides, maintaining the relationship throughout time becomes a challenge. And this challenge becomes even greater if little to no attention is placed on documentation (the second value). Little to no documentation implies relying on memory, which becomes hard as time goes by and things change.

Fact-based modelling aids in establishing and maintaining the customer relationship over time as customer involvement is a prerequisite for the success of the modelling methodology. As depicted in Fig. 1, the fact-based modelling protocol is a stepwise procedure that requires involvement of the customer in most of its steps.

In particular, the customer is involved in step 2 in determining and/or constructing together with the modeler the examples that clarify the knowledge identified in the previous step. In step 3, the examples are validated and complemented by the customer to come to a complete and formally approved set of concrete examples that can be used as a formal starting point for the further development of the model. In step 4, the concrete examples are generalized to the formal fact-based model. This step consists of several smaller tasks, most of which require involvement of the customer.

In conclusion, since the fact-based modeling protocol relies heavily on the involvement of the customer, and the customer's input is required in each step and documented in the model, close collaboration with the customer is achieved easily and does not rely on in-memory knowledge only.
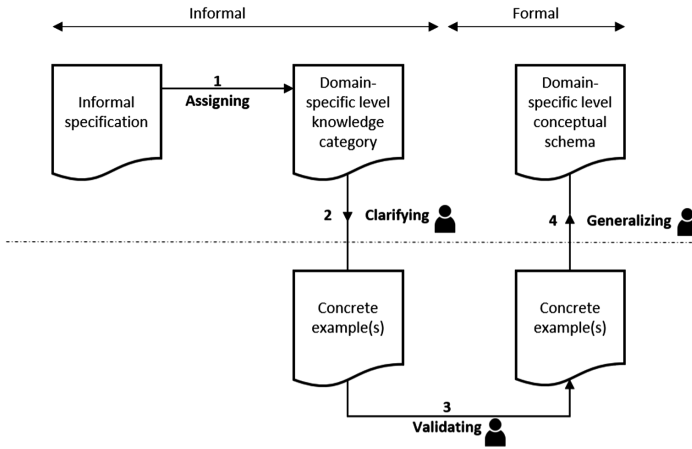
Fig. 1. The fact-based modelling protocol.

2.4 The Fourth Value: Responding to Change over Following a Plan

As described in the introduction, adaptivity is one of the core elements of the agile methodology. As work progresses, the understanding of the problem domain and what is being built changes, and these changes need to be well catered for. As stated in the introduction: responding to change does not mean that no plans are made. Instead, agile plans are constructed as a baseline that is used to control change.

Being able to handle change easily is one of the key feature of fact-based modeling. That is, fact-based modeling is a linear modeling methodology, which means that whether you create the first element of the model of the 100th element, the way of doing is always the same. Moreover, fact-based models are semantically stable, with no design choices that need to be made upfront. With fact types as the building block, there is no need during the process to take into account upfront the possible effect of changes that can take place. This in contrast to other analysis/design methods like e.g. UML, and ERD, where one has to consider upfront whether something must be represented as an attribute or potentially a class (entity) with a relationship.

The other aspect of fact-based modeling that tailors for change, is that fact-based modeling gives true focus as facts that do not belong to the subject area do not hamper the model, as stated through the 100% principle.

3 Conclusions

In this paper, we have demonstrated how fact-based modeling fits the values of the agile methodology. We could have done the same for each one of the 12 principles of the Agile methodology. For example, we could have argued that the early validation principle of fact-based modeling, which consists of validating whether the model is representative and validating whether the model is correct, is part of the protocol step 3 (see Fig. 1.), is an easy and consistent manner to fulfil principle 1 of the agile manifesto

which reads *"our highest priority is to satisfy the customer through early and continuous delivery of valuable software"*. Or, for example, that the fact that fact-based modeling focusses on atomic fact types in combination with continuous focus is input for achieving the principle *"deliver working software frequently from a couple of weeks to a couple of months, with a preference to the shorter timescale"*.

Fact-based modeling and the agile methodology have in common that they are both people-oriented and can handle change. We believe that fact-based modeling shows that documenting does not need to be a burden in software development as the model is the application and the documentation. Therefore, we consider fact-based modeling as the fulfilment of principle 9 of the Agile Manifesto: "*Continuous attention to technical excellence and good design enhances agility*".

References

1. Schwaber, K., Beedle, M.: Agile Software Development with SCRUM, 1st edn. Prentice Hall PTR, Upper Saddle River (2001)
2. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change. Addison-Wesley Professional, Boston (2004)
3. Cockburn, A.: Agile Software Development. Addison-Wesley Professional, Boston (2001)
4. Highsmith, J., Cockburn, A.: Agile software development: the business of innovation. Computer **34**(9), 120–127 (2001)
5. Cockburn, A., Highsmith, J.: Agile Software development: the people factor. Computer **34**(11), 131–133 (2001)
6. Fowler, M.: The new methodology. https://www.martinfowler.com/articles/newMethodology.html. Accessed 25 July 2018
7. Highsmith, J.: Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House Publishing, New York (2000)
8. Lemmens, I., Sgaramella, F., Valera, S.: Development of tooling to support fact-oriented modeling at ESA. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009. LNCS, vol. 5872, pp. 714–722. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05290-3_87
9. Halpin, T.: Object-role modeling (ORM/NIAM). In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Architectures of Information System, pp. 81–101. Springer, Berlin (1998). https://doi.org/10.1007/978-3-662-03526-9_4
10. Homepage manifesto for agile software development. http://agilemanifesto.org/. Accessed 24 July 2018
11. Wendorff, P.: An essential distinction of agile software development processes based on systems thinking in software engineering management. In: Third International Conference on eXtreme Programming and Agile Processes in Software Engineering (2002)
12. Van Griethuysen, J.: Information processing systems – concepts and terminology for the conceptual schema and the information base. Technical report ISO TR9007 (1987)
13. INCOSE: Systems Engineering Vision 2020 (incose-tp-2004-004-02), INCOSE (2007)
14. Lemmens, I., Koster, J.P.: The rule configurator: a tool to execute a model and play with the rules. In: Debruyne, C., Ciuciu, J., Panetto, H., et al. (eds.) OTM 2016 Workshops. LNCS, pp. 155–163. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-55961-2_15
15. Lemmens, I., van de Laar, B., Saton, J., Bulles, J.: How to fulfil regulatory requirements consistently: a semantic-based approach. In: Debruyne, C., et al. (eds.) OTM 2017. LNCS, vol. 10697, pp. 202–211. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73805-5_21
16. Paulk, M.C.: Agile methodologies and process disciplines. Crosstalk: J. Defense Softw. Eng., **15**(10) (2002)