Reviewing Commonalities between Agile Software Development Methodology and Grounded Theory Methodology. 2.3 15102018

Preprint · October 2018

DOI: 10.13140/RG.2.2.33374.28484

2 authors:

Sandip Mukhopadhyay
Institute of Management Technology
17 PUBLICATIONS 46 CITATIONS
SEE PROFILE

Rajen K Gupta
Management Development Institute Gurgaon
127 PUBLICATIONS 1,016 CITATIONS
SEE PROFILE

Reviewing Commonalities between Agile Software Development Methodology and Grounded Theory Methodology

Sandip Mukhopadhyay;
Institute of Management Technology (IMT), Ghaziabad, India.
sandipmukho@gmail.com.

Dr. Rajen K. Gupta.
rajenkgupta@gmail.com

Abstract:

Agile development method has attracted widespread corporate and academic attention as the limitations of traditional software development method become apparent. Similarly focusing on inductive knowledge creation, grounded theory method has achieved seminal status in social science. But there is no attempt so far to identify synergies between these two popular methodologies of diverse fields, so that social science researchers can benefit by using principles and practices of agile methodology. This paper identifies the similarities between these two methods in terms of the underlying philosophy as well as important practices. Focusing on these similarities, multiple recommendations regarding enriching grounded theory method with components of agile methodologies were made. The paper serves as an initial work on an important emerging methodology debate, which need to be further explored, enhanced and detailed.

Key Words:-

Agile methodology, Grounded theory method, Inductive, Research Methodology, Scrum, Iterative

Introduction

In the last 15 years, since the declaration of agile manifesto in 2001 (**Beck et al., 2001**; D**ingsøyr et al., 2012**), agile has become the most influential software development methodology. This is all the more exciting as during the same time-span, information technology and specifically the software industry, has seen massive growth both in terms of revenues and the number of people employed. Overall, the importance of software has grown significantly in business, industry and society. The popularity of agile in the field of software development can be gauged from multiple indicators; the large number of large IT organizations actively promoting it, the number of seminars being conducted, and the number of publications focusing on agile **(Dingsøyr, et al ., 2012**).

Similarly, since the introduction of grounded theory in the seminal book, *The Discovery of Grounded Theory* **(Glaser & Strauss, 1967),** it has created huge interests among social science researchers. There are countless applications of grounded theory in different domains of management research including information technology **(Urquhart, 2012)**. The popularity of grounded theory has given birth to multiple varieties of the core theory. Besides, grounded theory has immense direct and indirect impact on other research traditions of social science. Due to its popularity, there are multiple instances where researchers from other traditions claim or link their research methodology as derived or influenced from grounded theory **(Suddaby, 2006).**

Other than the high impact created by both agile and grounded theory methodologies in their respective field of software development and social science research, there are two other similarities worth highlighting. Both of these methodologies are born as a result of frustration with the existing way of doing things. Both of them are particularly focused on achieving their primary objectives. All principles of agile methodology are focused towards the development of working software, while the grounded theory methodology, and primarily wants to enable the researchers in generating new theories.

But till date, there has been little effort in academic research community to identify the commanalities between these two popular methodologies and to incorporate the relevant principles of agile into existing research methodologies or development of an independent agile research methodology **(Hazzan & Tozik, 2014).** This paper partially addresses this knowledge gap by explaining the following research objectives:

- To Identify the philosophical and methodological similarities between the two methodologies (grounded theory and agile)
- To explain conceptually how few of the agile practices and principles can be incorporated in grounded theory method.

The remaining part of the paper is structured as below. First we summarise the core principles and practices of agile methodology and grounded theory method in sections 2 and 3. Section 4 outlines the commnalities of grounded theory method (GTM) with agile.. Section 5 is the concluding section that discusses the possibilities of incorporation of agile principles in grounded theory method.

Agile Methodology

The agile software development methodology was developed in response to the perceived and actual limitations observed in existing plan and process driven software engineering methods (**Jalali & Wohlin, 2010**). The existing software engineering methods are not able to cope up with the present day realities. Organizations using software applications for competitive usage wanted development team to provide new or enhanced application on time and early **(Turk et al., 2014).** Besides, in today's continuously changing environments, market conditions change, new technologies emerge and the end user needs also evolve rapidly, so it is often becoming difficult to document how a computer application would evolve in future **(Pressman, 2005).** Conventional software methodologies expect detailed definition and documentation of all requirements before the development part of the project starts, which is not realistic in today's time. This led many to challenge the underlying philosophy of existing software engineering processes, which consider software development as a repeatable, predictable process (Sutherland, 2001).

Agile methodology was formailised in 2001 with the formulation of agile alliance and agile manifesto **(Beck et al., 2001**) with twelve core principles.

Drawing inspiration from the manifesto, agile methodology focuses on iterative and evolutionary development, rapid and flexible response to changes, early development of code and open communication in place of extensive documentation **(Turk et al., 2014).** One of the more important characteristics of agile is that it emphasizes rapid delivery of operational software instead of multiple intermediate work products and deliverables **(Pressman, 2005)**. Two other important characteristics worth mentioning are: a) incorporation of customer or its representative as part of development team to get continuous feedback; and b) underlying assumption that planning has its inherent limitation due to environmental uncertainty and this calls for flexibility in response to changes.

There are varieties of implementation of agile methodology, here we have focused on scrum. Though Scrum is initially developed for complex product development in an agile environment, it has potential for being used for any complex, innovative work. .Scrum allows development team to deliver incremental software in multiple short-duration development cycle or sprint. At the end of each sprint , the team delivers potentially usable software for the business users.   The business team gets the opportunity to revise the requirement list and their priority at the beginning of each sprint; while at the same time can enjoy the benefits of incremental features delivered by development team. The scrum process with important meetings known as ceremonies is depicted in Exhibit 1.

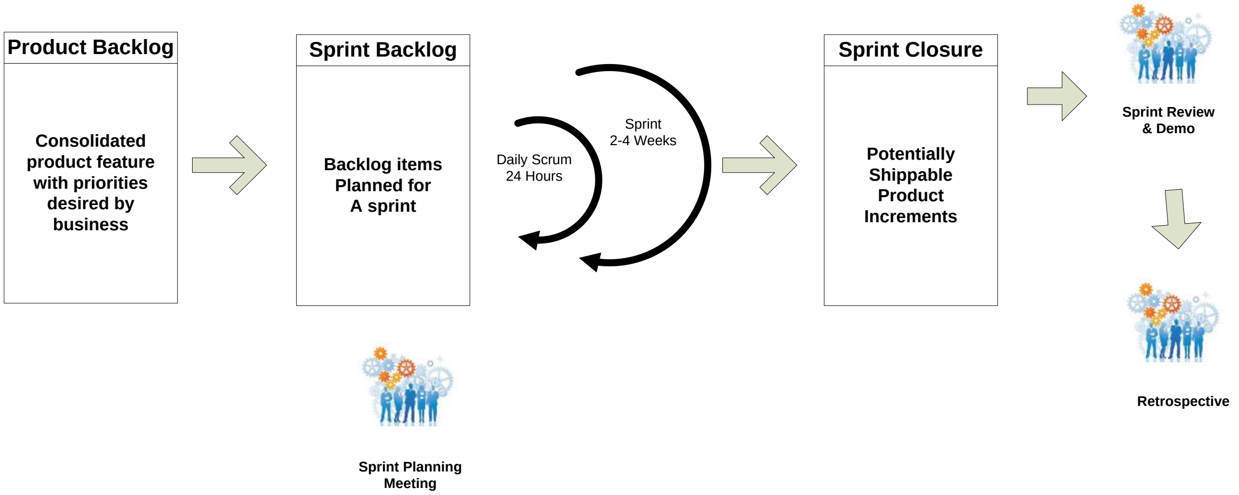
Product Backlog
Consolidated product feature with priorities desired by business
Sprint Backlog
Backlog items Planned for A sprint
Daily Scrum 24 Hours
Sprint 2-4 Weeks
Sprint Closure
Potentially Shippable Product Increments
Sprint Review & Demo
Retrospective
Sprint Planning Meeting

Exhibit1: Agile Scrum Framework and Ceremonies

Sprint planning is done at the beginning of a sprint for planning the scope and activities of a sprint.   Daily stand-up is the most important meeting; it is a 15-minute face-to-face mini-meeting for the entire team to sync. To keep the meeting effective, each team-member only focuses on three basic things: what you did since last meeting, what you are planning to do before next meeting and what are the obstacles you have. At the end of the sprint, the team demonstrates the newly developed features of the software in sprint demo. Sprint retrospective concludes a sprint. Here, the team collectively reviews of what did and didn't go well in that sprint.

Grounded Theory Method

Prior to the emergence of grounded theory, social science research was primarily dominated by deductive method borrowed from natural science **(Suddaby, 2006).** Glaser and Strauss (1967) developed grounded theory to challenge the domination of deductive method of knowledge creation in Social Science. As per their vision, the objective of social science research is to generate grounded theory from data. “Generating a theory from data means that most hypotheses and concepts not only come from data, but are systematically worked out in relation to the data during the course of the research” **(Glaser & Strauss, 1967, p. 6)**. They challenged the existing wisdom of hypothesis testing based on known theories as the primary role of researchers. They forcibly made the point of generating new theories rather than forcing data into a few existing theories **(Urquhart, 2012).** Grounded theory method is useful in social science scenarios, when theories concerned with the topics may not exist, or existing theories are too remote and abstract to be of any practical use (Martin & Turner, 1986). In other research methodologies; a great deal of effort is spent on reviewing existing literature as well planning different stages of research process **(Heath & Cowley, 2004).**

Grounded theory method encourages researchers to start gathering data for developing the research plan as well as to refine the research questions. This is a key feature of grounded theory method, as in the absence of prior theoretical ideas in researcher’s mind, substantive theory emerges from the data **(Urquhart, 2012).**

Grounded theory method rests upon two key concepts: “constant comparison” and “theoretical sampling”. Due to Constant comparison, data collection and analysis happen simultaneously **(Suddaby, 2006; Urquhart, 2012)**. This is a very powerful tool, as instances of data labelled in one category are continuously compared with other data instances labelled for that category here

(Urquhart, 2012). 'Theoretical sampling' determines the decision related to the type of data to be collected next based on the emerging concepts developed so far from the analysis.

Synergies between agile method and grounded theory methodology

The best way to analyse the synergy between these methods should start from their history. Agile was initiated to address the limitation of popular plan and process driven software development methodologies. The existing methodologies like waterfall were unable to take into account the rapid uncertainty in environments due to change in technologies, business models, customer preference and regulation. The pace of change in industry in recent years has increased, which asked for a flexible methodology. Though agile does not discount the importance of prior planning, it provides more importance to short-term planning compared to long term planning. It also allows the project sponsor to change the priority with the passage of time. Similarly, grounded theory methodology was a reaction to the predominately deductive mode of knowledge creation prevalent in the time of Strauss and Glaser. In the deductive mode, the knowledge creator goes from theory to fact **(Arbnor & Bjerke, 2008)**. The researcher develops forecasts about a specific event from existing theories and verifies in empirical world. During the time of Glaser and Strauss, the arena of social science was rapidly expanding, new phenomena were encountered on a regular basis. Glaser and Strauss **(1967)** argued that existing grand theories does not adequately cover and explain all areas of social life, that are getting researched **(Urquhart, 2012).** So, instead of focusing on verification of existing theories, new methodology enabling researchers to generate theories from data is the need of the hour.

Both the methodologies share common traits with respect to their treatment towards reality. Agile methodology concedes that reality is dynamic and evolving, so there is a limit of what can be achieved based on rigid prior planning. It advocates flexibility and agility in planning process; it welcomes new changes to make the outcome consistent with changing business environment instead of resisting it. Grounded theory rejects scientific definition of fixed reality and the notion that social and natural sciences deal with similar subjects (Suddaby, 2006). Grounded theory focuses on interpretation of the reality by individuals. Researchers in grounded theory provide attention on daily realities related to important or substantive matter and how the individuals interact & interpret with phenomenon being observed.

Both the methodology provides great importance in achieving their objectives and provides relatively less importance on intermittent things. For agile methodology practitioners, the most important objective is to deliver working software at regular intervals. Many critiques of agile methodology object to it, as it provides less importance to documentation and adherence to other existing software engineering best practices, while providing the foremost importance to software code delivery. Similarly, grounded theory is primarily about theory development (**Urquhart, 2012**), though its procedures are nowadays are increasingly used for data analysis by researchers even not using grounded theory. Agile advocates not to spend enormous amount of effort on documentation, but to make available adequate documents ongoing basis. Similarly in grounded theory, though researchers get involved in literature survey; but unlike in the traditional method, they need to make sure that literature survey does not influence the data analysis process **(Urquhart, 2012).** As a result, the focus is not on an exhaustive literature survey and creation of prior hypothesis and propositions.

Agile, unlike other traditional methods of software development, does not follow a sequential chain of activities. Most of the traditional methods are sequential in organising core activities:- requirement, design, coding, testing. Sequential closure of these activities would allow business requirements to finally convert to software applications **(Pressman, 2005).** The activities for the next phase only start once there is a closure or a sign-off in the earlier phase. This is one of the reasons, why this methodology takes longer time to deliver, as well as ineffective in handling changes. Instead of sequential phase-gated delivery, agile development is based on iterative, incremental software development to facilitate early delivery of working software **(Turk et al., 2014).** So, a project team may be developing and testing features for the first spritnt, but at the same time can be refining requirements for the second sprint . The sequential activity as followed in standard software development methodology (Requirement Finalization- Design- Development- Testing- Commercialization) is not followed in agile. GTM (Grounded Theory Method), also does not follow sequential method of research (Literature survey- data collection- data analysis- report development); as it allows the overlapping of data collection and analysis (**Urquhart, 2012**). The key concept of 'constant comparison' indicates data are collected and analysed simultaneously (**Suddaby, 2006**). Similarly, the concept of 'theoretical sampling' highlights that the emerging concept arising out of data analysis decides where to sample next. So, new data collection in grounded theory is not driven by a-priori hypothesis generated during literature survey, but based on the researchers' interpretation of data and emergence of concepts. So, in a typical grounded theory research, there are multiple rounds or phases of data collections and analysis (**Charmaz, 2006**) similar to iterative software developments in agile.

Software development methodologies including agile method can be evaluated based on where it stands on a continuum from 'adaptive' to 'predictive' **(Boehm & Turner, 2003**). The exponents of agile methodology put agile in the adaptive side; indicating that this methodology is not 'plan-driven'. On the other hand, it adapts quickly to changing reality and does not depend heavily on future planning. The agile exponents would not spend too much time planning and describing what would happen in future; the further away a date is the less explanation an agile method would be able to provide about what would happen in that time. Rather, they focus on what is happening now and the related bottlenecks. Grounded theory is also shows similar traits of adaptiveness in its approach. As mentioned before, data collection strategy in grounded theory is defined by the outcome of the analysis, emergence of the meaningful categories. Emergence is a key concept in grounded theory (**Glaser, 1992**); it ensures that the researchers remain true to the data and construct meaning. Strauss and Glaser highlighted this with the focus on "the discovery of theory from data" instead of verification of existing theories in different context **(Urquhart, 2012; Suddaby, 2006).** So, researchers in grounded theory does not strictly follow a research plan created in the beginning, rather updates it based on emerging categories and other discoveries from the data. Even the concept of theoretical sampling indicates the limitation of long-term planning and moving ahead with the outcome of the data.

Applying key concepts of agile development into grounded theory method

The above discussion shows though the two methodologies discussed are used in different fields, both share similarities in underlying philosophy, assumptions about reality and developed practices. Both of them have reached leadership status in the irrespective fields and attracted many researchers and managers, so there is significant scope of enriching each other by cross pollinizing. In this article we are focused on two possible applications of agile development principles on grounded theory mechanism (Hazzan & Tozik, 2014) and the resultant benefits.

Agile’s focus on close coordination with business stakeholders and customers: Agile incorporates close coordination with customer; and it advocates superiority of ‘customer collaboration over contract negotiation’ in its manifesto. The traditional software development methods treat contract as supreme, while believers in agility focuses more on customer collaboration **(Mellor, et al., 2002**).

Similarly, if we look into overall management research program in business schools, they contains very limited real life problem and their possible solutions **(Maital et al., 2008; Panda & Gupta, 2014)**. Like in traditional software development methodologies focusing on process instead of the logical objective of early delivery of useful software, business researchers have focused more on methodological rigour compared to providing insight into actual troubles faced by business (**Khurana & Spender, 2012**

As methodology, grounded theory has shown the maximum potential to bring industry relevant wisdom in academic research **(Maital et al., 2008).** Incorporating and prioritizing the agile principle of working closely with customer and business owners would have very positive impact in outcome of business researches. In this collaborative knowledge creation, practitioners can highlight critical research problems that arise out of their experience and the researchers can take the lead in application or expansion of existing body of knowledge for providing insight into the problem **(Panda and Gupta; 2014; Maital et al., 2008).**

Adapting Scrum ceremonies and practices in research: The concept of scrum as explained in exhibit 1 has been modified and widely used in multiple other industries and corporate initiatives. Hicks and Foster **(2010**) had utilized and adapted the key concepts of scrum for managing an academic research group in university and achieved significant benefits. They were managing a large group of PhD students and earlier method of ad-hoc meetings and discussions were proving to be inadequate. So, they adopted best practices from scrum for a more orderly management of researchers. They organized status meetings of 15 minute duration on alternate days of the week with all the research scholars. These meetings were run similar to daily stand-up meeting of scrum, where all PhD students are expected to describe the brief progress since the last meeting, the obstacles and what they intent to do before the next meeting. In tune with agile principles of collocation and preference of face-to-face interactions, they also organized collocation of researchers and professors leading to more interaction between students as well as between students and professors. Through a survey, they found significant improvements and enhanced student satisfaction after using the new method over the old ad-hoc method.

Other ceremonies of scrum, like retrospective and sprint planning can also be incorporated in grounded theory research structure. Retrospective meetings can be used as . can be used as a structured way to include reflection and feedback from co-researchers in the entire methodology. Similarly, sprint planning meetings can be planned once in a month or six weeks. But, we need to remember; though scrum can be a very useful tool in any research practice, this can be primarily practiced in a group setting.

References

Arbnor, I., & Bjerke, B. (2008). *Methodology for creating business knowledge*. Sage Publications Ltd.

Banerjee, A. (2013). Academic Research Productivity: What may be “reining” in the Indian B-School?.

Beck, Kent, et al. "Manifesto for agile software development. Feb. 2001." *Snowbird, UT* (2005).

Boehm, B., & Turner, R. (2003). *Balancing agility and discipline: A guide for the perplexed*. Addison-Wesley Professional.

Burgoyne, J., & Reynolds, M. (Eds.). (1997). *Management learning: integrating perspectives in theory and practice*. Sage.

Charmaz, K. (2006). Constructing grounded theory: A practical guide through qualitative analysis (Introducing Qualitative Methods Series).

Conforto, E. C., Salum, F., Amaral, D. C., da Silva, S. L., & de Almeida, L. F. M. (2014). Can agile project management be adopted by industries other than software development?. *Project Management Journal*, *45*(3), 21-34.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213-1221.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, *50*(9), 833-859.

Glaser, B. G. (1992). *Emergence vs forcing: Basics of grounded theory analysis*. Sociology Press.

Glaser, B. G., & Strauss, A. L. (1967). Discovery of grounded theory. Mill Valley..

Hazzan, O., & Tozik, S. (2014, May 14). InfoQ. Agile Research. Retrieved from https://www.infoq.com/articles/agile-academic-research.

Heath, H., & Cowley, S. (2004). Developing a grounded theory approach: a comparison of Glaser and Strauss. *International journal of nursing studies*, *41*(2), 141-150.

Hicks, M., & Foster, J. S. (2010). Adapting Scrum to Managing a Research Group.

Jalali, S., & Wohlin, C. (2010, August). Agile practices in global software engineering-A systematic map. In *2010 5th IEEE International Conference on Global Software Engineering* (pp. 45-54). IEEE.

Khatri, N., Ojha, A. K., Budhwar, P., Srinivasan, V., & Varma, A. (2012). Management research in India: Current state and future directions. *IIMB Management Review*, *24*(2), 104-115.

Khurana, R., & Spender, J. C. (2012). Herbert A. Simon on what ails business schools: More than 'a problem in organizational design'. *Journal of Management Studies, 49*(3), 619-639.

Maital, S., Prakhya, S., & Seshadri, D. V. R. (2008). Bridging the chasm between management education, research and practice: moving towards the 'grounded theory'approach. *Vikalpa*, *33*(1), 1-18.

Martin, P. Y., & Turner, B. A. (1986). Grounded theory and organizational research. *The journal of applied behavioral science*, *22*(2), 141-157

Mellor, S. J., Jacobson, I., & Henderson-Sellers, B. (2002). The Great Methodologies Debate: Part 2. *The Journal*, *15*(1).

Mukherjee, R., & Mukherji, P. N. (2000). Methodology in Social Research: Dilemmas and Perspectives: Essays in Honor of Ramkrishna Mukherjee.

Mukhopadhyay, S., & Gupta, R. K. (2014). Survey of qualitative research methodology in strategy research and implication for Indian researchers. *Vision: The Journal of Business Perspective*, *18*(2), 109-123.

Panda, A., & Gupta, R. K. (2014). Making academic research more relevant: A few suggestions. *IIMB Management Review*, *26*(3), 156-169.

Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.

Sinha, J. B. P. (1984). Towards partnership for relevant research in the Third World. *International Journal of Psychology*, *19*(1-4), 169-177.

Suddaby, R. (2006). From the editors: What grounded theory is not. *Academy of management journal*, *49*(4), 633-642.

Sutherland, J. (2001). Agile can scale: Inventing and reinventing scrum in five companies. *Cutter IT journal*, *14*(12), 5-11

Sutherland, J., & Schwaber, K. (2013). The scrum guide. The definitive guide to scrum: The rules of the game. *Scrum. orgOctober*.

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard business review*, *64*(1), 137-146.

Turk, D., France, R., & Rumpe, B. (2014). Assumptions underlying agile software development processes. *arXiv preprint arXiv:1409.6610*.

Urquhart, C. (2012). *Grounded theory for qualitative research: A practical guide*. Sage.