5-1-2017

Hybrid Project Management: Agile with Discipline

Olayele Adelakun
DePaul University, oadelakun@cdm.depaul.edu

Robert Garcia
DePaul University, rgarci11@cdm.depaul.edu

Ted Tabaka
DePaul University, ttabaka@cdm.depaul.edu

Redar Ismail
DePaul University, rismail2@cdm.depaul.edu

Follow this and additional works at: http://aisel.aisnet.org/confirm2017

HYBRID PROJECT MANAGEMENT: AGILE WITH DISCIPLINE

Olayele Adelakun
DePaul University
oadelakun@cdm.depaul.edu

Robert Garcia
DePaul University
rgarci11@cdm.depaul.edu

Ted Tabaka
DePaul University
ttabaka@cdm.depaul.edu

Redar Ismail
DePaul University
rismail2@cdm.depaul.edu

Abstract

Effective management of software projects will always be important regardless of the software development method (agile, iterative or waterfall) used. A recent movement in the software development industry towards adopting Agile practices have left many questioning the role of traditional project management. However, in practice companies often struggle with changing established practices. Many companies have adopted hybrid methods to adjust to changing requirements. For many of these companies these hybrid approaches are seen as the best of both worlds as they can leverage the advantages of Agile with the strengths of traditional practices. While researchers have begun proposing ways that these approaches can integrate there remains limited actual academic evidence that describes how these models are being integrated in practice. This research contributes to the knowledge by discussing findings based on a unique approach adopted by the IBM Center of Excellence called Agile with Discipline.

Keywords:

Project Management, Agile, Waterfall, Hybrid

1. Introduction

Over the past several years there has been a growing interest in technology development that steers away from traditional practices towards methods that embrace Agile principles. Many adopters believe that Agile practices provide advantages over traditional methods. Some of these advantages include adaptability, increased product quality, developer happiness, and earlier defect detection (Laanti, Salo, & Abrahamsson, 2011). These changes have followed what some view as a growth in the importance of knowledge based work and changes in management from hierarchical approaches towards more collaborative efforts with frequently changing requirements (Fernandez & Fernandez, 2008).

These practices are both changing and challenging the traditional approaches used in project management (B. Boehm & Turner, 2005). Studies show that management may view the benefits of Agile efforts as aiding in changing requirements, accelerating time to market and contributing to software (Papatheocharous & Andreou, 2013). Despite the perceived advantages of Agile there are a number of challenges facing management in its adoption (B. Boehm & Turner, 2005; Nerur, Mahapatra, & Mangalaraj, 2005). Further, Agile practices may not always be viewed as suitable for adoption in different environments, particularly those with more stringent requirements (McHugh, McCaffery, & Casey, 2012).

To manage many of the challenges with modern development in businesses many have resorted to developing hybrid project management practices that integrate Agile with traditional approaches (Binder, Aillaud, & Schilli, 2014; Rahmanian, 2014). However, because of the divergence between Agile and traditional practices, some have stressed the

importance in further examining these approaches, the decision making behind them and the characteristics adopted (Špundak, 2014).

This paper seeks to add to the knowledge on hybrid Agile approaches by discussing a hybrid management style observed during a case study at the IBM Center of Excellence in Chicago, Illinois, USA. Over the years companies have increased their adoption of Agile practices (Diebold & Dahlem, 2014). However, many of the Agile practices and methodologies used by companies in rapidly changing information technology environments continue to evolve. In order to understand the best approaches towards developing and improving the current state of Agile development it is important to examine this evolution and the contextual factors that influence them.

The approach observed at IBM, which we are calling Agile with discipline, is unique in that it integrates elements of the waterfall model used within an Agile framework. First a discussion will be presented comparing traditional project management (TPM) versus Agile project management (APM) to help discuss the different elements of these approaches. Next a discussion of the IBM environment will be presented. Following a discussion on the Agile with discipline approach will be discussed based on a case study and interviews with managers at IBM. Based on the discussions in the paper a proposed model based on Agile with discipline will then be presented.

2. Traditional and Agile Project Management

Project management has been defined as the planning, organizing, directing and controlling of resources for short term objectives that aid companies in meeting specific goals (Kerzner, 2013). Project management has traditionally been a very linear process that has relied on hierarchical methods. These methods rely heavily on planning, documentation and requirements analysis in the early stages of a project (Sixsmith, Freeburn, & Mooney, 2014). This is demonstrated in the views of process groupings in TPM.
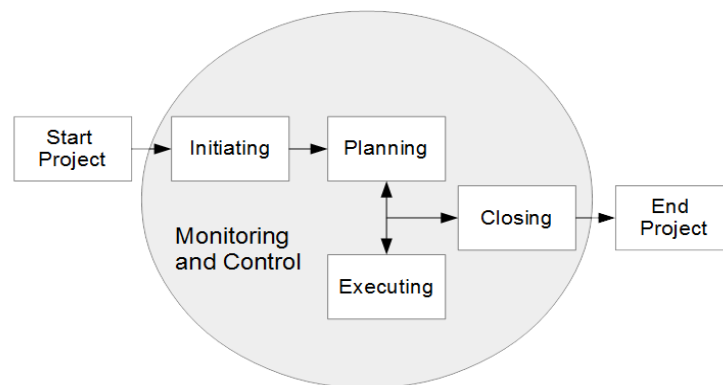

Figure 1. Traditional Project Management (TPM) Process Groups according to (Rose, 2013)

In TPM many processes are viewed as linear and sequentially based on the products of previous phases (Fernandez & Fernandez, 2008). For example, through visioning in the imitation phase teams are able to move on towards planning the entire product based on the vision. Once the plan is created it can then be moved to execution and eventually to closure. The control phase is used to handle the real world aspects of execution, not always following plans and processes in this phase are used to monitor and redirect efforts as needed (Rose, 2013). In this approach changes to the project are carefully controlled with proper documentation until the final item on the work breakdown structure is delivered.

Software development practices have typically followed TPM approaches. These practices are generally plan-driven and rely on command and control approaches (Rehman, Rauf, & Shahid, 2010). For example, the most widely used model for software development that follows this framework is the waterfall model.
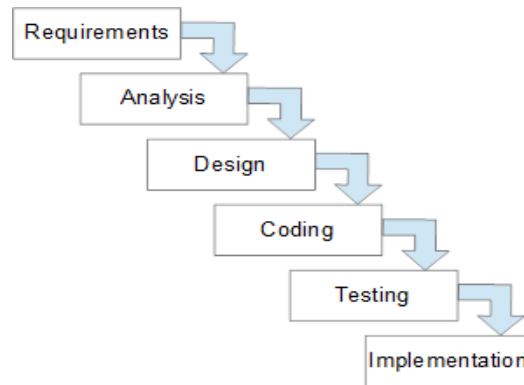

Figure 2: Waterfall development model (Royce, 1970)

Royce (1970) first discussed the waterfall model as the approach many organizations took towards software development. The waterfall model follows a sequential approach that separates development into unique phases. Each phase is performed sequentially and a new phase does not start until the previous one ends. During the end of a phase documentation is also developed (Balaji & Murugaiyan, 2012). In this manner the flow of development has a defined beginning and end. The project begins with the initial planning and flows through the cycle until the project is complete and implementation begins. However, this also becomes one of its main weaknesses. The waterfall method does not adequately address unexpected issues, that are common in software development occurring during any of its phases (Ken Schwaber, 1997). Issues such as this have led to a variety of other software development models such as the spiral model, v-model, iteration model, and extreme model (Munassar & Govardhan, 2010). These have resulted in new practices.

The development of new software development practices has also led to a need for new management approaches. While TPM approaches and practices have been suitable in previous decades, some have challenged the theory behind those approaches as obsolete (Koskela & Howell, 2002). These critiques are based on views of a changing business environment that requires increasingly complex development in uncertain project environments. The rapidly changing environments in which modern projects exist can create situations in which TPM can not only fail to address issues but can increase the problems (Williams, 2005). One of the main arguments against the traditional approach is its inability to adjust to the dynamic and volatile nature of business organizations, technology, market place, customer and socio-technological environments (Baskerville, Ramesh, Levine, Pries-Heje, & Slaughter, 2003). Among the alternatives towards project management in these environments are Agile approaches.

Agile is a term used to describe an approach towards software development that integrates a set of principles that encourage iterative and incremental development through the collaborative efforts of self-organizing and cross-functional teams. Agile is iterative, incremental, self-organizing, and emergent (Lindvall et al., 2002). Agile is iterative in the respect that development is completed over several cycles. It is iterative in that the product is not delivered at once, but in small completed parts. Teams in Agile are self-organizing and

determine on their own the best way to handle work. Agile is considered emergent as processes, principles and work structures are not pre-determined but rather determined during the project development. Agility in Agile development is about embracing change throughout the development process as opposed to traditional methods that lock requirements. Agile methods are willing to capture last minute changes as they believe such changes could produce unanticipated benefits to all stakeholders especially the end customers. Table 1 outlines some of the key components of Agile as described in the Agile manifesto (Beck et al., 2001).

Agile		Traditional
Individuals and Interactions		Processes and Tools
Working Software	Over	Comprehensive Documentation
Customer Collaboration		Contract Negotiation
Responding to Change		Following a Plan

Table 1: Agile manifesto summarized by (Moniruzzaman & Hossain, 2013)

In Agile the traditional project manager (PM) is replaced with that of a team lead or scrum master. The PM is not required to do heavy documentation and the end user is much more involved in the process (Uikey & Suman, 2012). There are various agile methods (eXtreme Programming, Scrum, Dynamic Systems Development Method, Adaptive Software Development, and Crystal), Scrum is the most adopted agile method (Moniruzzaman & Hossain, 2013; K Schwaber & Beedle, 2002). Agile works well where the requirements are more uncertain and subject to change, that is, where components are more interdependent and subject to frequent changes (Augustine, Payne, Sencindiver, & Woodcock, 2005).

Agile Project Management (APM) implies that agile managers will lead small teams, clarifying roles and responsibilities; communicate a vision to the team; follow simple rules that allow for quick and flexible team work; allow for free and open access to information for the team to accomplish tasks and goals; lead with a light touch management style; and employ adaptive leadership (Augustine et al., 2005). This framework differs from the traditional PM (TPM) which is document, process, and plan heavy.

Software development practices in Agile are typically more dynamic. Among the most popular Agile framework used in software development is Scrum. Scrum is an iterative and incremental framework (Ken Schwaber, 1997). Scrum begins with an overall visioning session. In practice the product visioning moves towards product backlog creation and then the planning for sprint cycles. Sprint backlogs are then created. This then moves to an iterative sprint cycle in which planning, development and releases of small product deliverables. This cycle is completed until the final product deliverable is completed. In Scrum the project flow differs from traditional approaches such as waterfall.
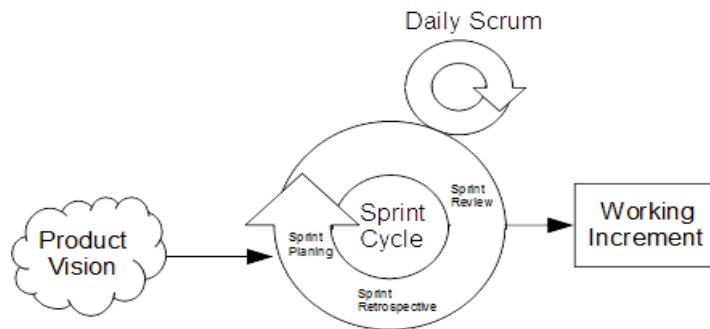
Figure 3: Modern Scrum Approach

While waterfall defines the entire life-cycle at the beginning of the project with known deliverables, Scrum is more of an evolving process. The incremental deliverables in Scrum allows a project to change midstream without reducing the ability to complete the end project. With traditional approaches, changing a project midstream requires a revision of the entire planning process and a return to earlier project stages. Further Agile approaches such as Scrum change the role of traditional PMs. In Scrum, traditional PMs are replaced by Scrum masters who move from playing the role of team leaders to coaches that motivate team members to complete project goals (Cervone, 2011). Included in Scrum teams are clients that take the role of product owners.

In some businesses Agile approaches are seen as an alternative that can rapidly be used to replace traditional practices (Laanti et al., 2011). Others caution that a slow and deterministic change is necessary (Nerur et al., 2005). Although in theory, there are wide differences between the two approaches, researchers have described how the reality of approaches in practice differ (Sixsmith et al., 2014). Researchers have described approaches towards project management that mix both TPM and APM (Binder et al., 2014; Hass, 2007). However, many of these have been experimental or theoretical frameworks. In the following sections a hybrid model that has been observed in practice at a corporation in the United States will be discussed.

3. Research Method

In order to examine Agile development further in 2014 a team of researchers from DePaul University began exploring the use of Agile in practice. The team was interested in exploring the research question of how Agile was being used by companies in practice and how this differed from theoretical approaches. During visits to IBM, one of the major software development companies being investigated, the team identified a unique hybrid management approach. The team explored the method further by conducting interviews with project development leaders at the organization   The goal of this research is to understand they hybrid TPM-APM practices at IBM.

In order to examine these practices further direct observations and Semi-structured interviews were conducted at IBM Center of Excellence, Chicago, USA in the winter 2014. The empirical data was collected using a semi-structured interview data collection technique. All the data were collected from employees at the IBM Center of Excellence in Chicago. All interviews were recorded with permission, except for one interviewee who preferred not to be recorded. In total four hours of interview data was collected. All of the interviewees are either PMs, project leads; Architects or Senior PMs. Five people were interviewed in total. Although, the sample size may be considered relatively small for quantitative research for qualitative studies samples are generally collected until saturation occurs (Guest, Bunce, & Johnson, 2006). This can generally occur within the first 12 interviews. After about the third

interview we noticed that we began getting similar responses from other interviewees. Further the interview data collected from high level practitioners at IBM supported what researches observed in practice and therefore the number of interviews collected is considered sufficient for describing the practices.

The structured interviews covered questions such as what PM methods are used in the center and what methods within IBM? Do all PMs use the same methods across the enterprise? What influences the decision of which software development method is used and thereby the PM approaches? Do PMs develop same type or same quantity of documentation regardless to the development method? These questions were based on descriptions

All interviewees noted that both the traditional and agile software development methods are used within IBM. The project development method, tools, that project team members use different approach depending on a number of factors. None of our interviewees gave us one method that supersedes all other methods. Instead all of them discuss a hybrid-agile approach often referred to as "Agile-with-Discipline."

Our research findings are discussed in the next sections.

4. Selecting a Software Development Method

The software development method applied to a project in IBM depends on a number of factors, and our data shows that no one method is always used. Table 3 summarizes some of the factors that our interviewees identified as determinant of the project development method.

1. The nature of the project (size, complexity, number, global site, customer preference, etc.
2. Project complexity: simple development or integrating many applications
3. Modification of an existing system or brand new development.
4. What type of technology will be used: a tool that will be customized or build from ground zero
5. The environment of the development project
6. Based on previous work, best methods in previous situations.
7. What is defined in the project contract/scope of work.
8. Has the project been determined in the contract as time and material (T&M) or fixed price? This can leverage the project methodology. If it is a fixed price, requirements will be locked down and the project will follow a waterfall methodology. T&M provides opportunities for the customer to make changes to the requirements and provide input, which is the purpose of agile development.
9. Does the customer prefer a particular method?

To summarize, many of our interviewees state that projects that require more customer/client collaboration, such as web-based projects are more conducive to agile development methods. All of the PMs noted the limitation with the traditional method, but none of them follows the agile method as discussed in the literature. Instead they have adopted a hybrid-agile method. This is partly because IBM has its own set of custom methodologies that it has developed through years of experience in working with clients, managing successful projects, and providing IT/Business solutions to its clients.

Our interviewees often refer to their hybrid-agile method at IBM as "Agile-with-Discipline." The Agile-with-Discipline is IBM's methodology that incorporates components of agile development into a more structured approach to project management.

> *"In this approach [hybrid-agile] sufficient documentation and timelines with flexibility can accommodate requirement changes, development sprints, and continuous customer/client feedback."* – A senior PM

5. The Role of the Project Manager with Agile Software Development

While current literature on the topic down plays the role of project managers within the agile methodology we found that project managers in our research are just as involved irrespective of the development method. PMs still play an integral part in managing people, leveraging resources, and overseeing the project's success. As one PM explains:

"The PM's role does not change with managing an agile project in a delivery organization. How the PM delivers his/her solution and involves customers is what changes. The PM still owns the project, still drives the project through the completion of sprints, still goes through all of the checks and balances needed, and still gets all of the required sign-offs. The things that change are that the PM has a lot more interaction with the customers, leads daily sprint calls with the team, and scopes the projects a bit differently (i.e. delivery in smaller chunks and frequent reviews with the customer."

– A Senior PM

Sometimes other people like architects, senior developers or other project stakeholders may play the role of the PM/ scrum master but only temporarily. It appears that every project has a project manager, especially client projects.

"I am an architect but sometimes I take the role of the scrum master on an agile project when needed." – An Architect

6. Project Planning Within the Agile Method

All PM interviewees note that they are responsible for creating a project plan in the Agile-With-Discipline method. While time, scope and budget are loose, project success ultimately still depends on good project organization. The PMs are also responsible for providing tools to team members needed to accomplish project tasks. Some spoke of using internal company collaboration tools or software such as SharePoint. The selection of the collaboration tool that works best often depends on the client's preferences and the type of project., The PM is always responsible for maintaining the tool and ensuring team collaboration regardless of the tool used.

"The plans developed for agile-with-discipline projects may not be as detailed and as far thought out as a traditional project plan may be." – A PM

Part of the project plan can also include setting project milestones and implementing a loose project timeline. Often, a payment schedule or some type of success-measurement marker is attached to project milestones. Scheduling team meetings/scrum meetings and communicating key project information to the team is also an important part of the PM's overall project plan.

"We have noticed that the Traditional approach to project planning doesn't work very well. You can't plan everything before you execute. It is impractical... So we do the best to plan some [hybrid-agile] but be ready to adapt as needed." – Senior PM

7. Project Constraints

In traditional project management, the project manager scopes out a project within the very tight constrains of scope, time, and budget. In the agile development process, these constraints are defined as the project progresses. What we find in the case of hybrid-processes is that these constraints are loosely defined at the start of the project for the PM. The PM is to monitor these constraints, mitigate the expansion of any one area to the point that it would lead to customer dissatisfaction or violate a contract. Also the PM

communicates the effect that changes in one area can have on other areas to the customer and the team.

8. Project Reports and Documentation

To control project constraints and to ensure the customer is getting a valuable and useful product, documentation becomes an integral part of hybrid-agile/Agile-with-Discipline approach. PMs need to keep documentation on project requirements, changes, resources used, and the project timeline. While requirements can evolve over the project lifecycle, documentation is important in holding project stakeholders accountable for the project's success, keeping track of the project's requirement changes and iterations, and establishing an overall project plan.

> *"When we start any project we at least try to get a blueprint or an initial set of requirements and what we do is internally manage a change request process…if there are changes to the original requirement, we go through the change request process to make sure the customer is aware of this particular change before we make it happen so, it's more documented; any changes to the timeline or the budget is reflected based on the new change request."* – Senior PM

Documentation is also important for project communication. IBM rarely has teams that are co-located. A team usually consists of members across the globe. In these instances, documentation and the use of project collaboration tools are integral for project hand-offs and keeping the project moving forward.

Document version control, reporting to the team, the client, and to other project stakeholders, and creation/collection of test scripts and design decision documents are all part of the PM's documentation management role. The formality and frequency of documentation again depends on the context of the project, including the client being served, the nature of the project and the business culture.

9. Project Team

The literature emphasizes that one of the distinguishing characteristics of the agile development process is its allowance for self-organizing teams with decision making power. In theory, this may be true. In reality, a team is often part of a larger company organization and ecosystem where organizational impediments can obstruct team success. In this case, a PM can serve as a liaison for the team to the larger organization to gather resources and remove obstacles. One PM gave the example of when he needed to procure an additional team member for the team that suddenly lost one of its members. In such a case, it is rarely appropriate for any team member to select a new person, and no one on the team has access to that type of resource. Depending on the organizational culture and the larger context within which the team is working, it is easy to imagine situations and scenarios where a PM has access to more resources and support that he/she can funnel into the team, as opposed to the team itself.

The tasks outlined above illustrate some of the larger roles and responsibilities the PM interviewees mentioned in the conversation. They are not inclusive of all responsibilities a PM might undertake in managing an agile or agile-like development project. These roles and responsibilities can vary by project, team, and company.

10. Industry Opinion – Agile vs. Traditional

As the agile development method becomes more popular, more companies are requesting it and expecting it to be a part of the development processes. Companies want to be shown progress and be involved in the development of their products.

"More PMs are managing agile or agile-like projects. Interest in Agile Project Management seems to be increasing in the project management field." – Team Lead

Our interviewee noted that to meet customer demands for hybrid-agile, IBM has been providing resources, tools, and processes (such as agile development education) to its PMs internally. This support has often come through its project management office, which serves more as a professional development and resources office for its project managers than as a center for managing the company's portfolio of projects.

11. Insights into Agile with Discipline

Based on our findings we performed a comparative analysis between elements described as Agile and Traditional in the literature versus the hybrid Agile with Discipline approach identified at IBM. Table 2 below outlines some of the differences between approaches.

Elements	TPM	APM	Agile w/ Discipline
Applicable Development Life Cycle	Favors Linear (waterfall) and Iterative (spiral) - (B. W. Boehm, 1988; Fernandez & Fernandez, 2008; Royce, 1970)	Iterative, Evolutionary (scrum or XP) (Augustine et al., 2005; Victor, 2003)	Varies based on project context
Style of Development process	Predictive (Ken Schwaber, 2004)	Adaptive (Paetsch, Eberlein, & Maurer, 2003)	Limited adaptiveness based on project context
PM (Requirement and scope management)	Clearly defined scope, knowable early, largely stable, well documents, WBS, scope creep - (Rehman et al., 2010)	Scope emerges, rapid change, unknown requirement discovered during the project development, no WBS and no scope creep - (Cockburn & Highsmith, 2001; Rehman et al., 2010)	Loosely defined constraints at project beginnings, constraints are monitored throughout process, constraints are modified when leading to customer dissatisfaction or contract violation
Project Management Approach	Plan and process centric, monitoring and control - (Cockburn & Highsmith, 2001; Hoda, Noble, & Marshall, 2008)	People centric, collaborative, adaptive - (Hoda et al., 2008; Paetsch et al., 2003)	Loose planning but extensive organization with PM serving as gatekeeper
Project Goal	Clear and Predictable (Ken Schwaber, 2004)	Exploration or Adaptation - (Paetsch et al., 2003)	Project Context dependent

Project Documentation	Heavy Documentation in General - (Sixsmith et al., 2014)	Generally Light or Insignificant Documentation - (Beck et al., 2001; Lindvall et al., 2002)	Documentation is integral but formality and frequency is project dependent
Requirement Changes	Controlled, Changed Averse - (Fernandez & Fernandez, 2008)	Embrace Change - (Beck et al., 2001; Cockburn & Highsmith, 2001; Fernandez & Fernandez, 2008)	Based on product owner satisfaction and contract limitations
Team Members	Dispersed Team, Specialists, Task-Skill Alignment (Cockburn & Highsmith, 2001)-	Agile, Knowledgeable, Favors Collocated and Collaborative - (Beck et al., 2001; Cockburn & Highsmith, 2001)	Dispersed or collocated teams, multiple roles, collaborative
Team Orientation	Structured, headed by PM - (Nerur et al., 2005)	Self-organizing teams, decision making empowered - (Beck et al., 2001; Cockburn & Highsmith, 2001)	Teams exist as part of larger corporate ecosystem with PM serving as liaison and roles can vary based on context
Client and Stakeholder Involvement	Low involvement mostly requirement and validation - (Hoda et al., 2008; Thompson, 1991)	Actively Involved, Client is part of the team - (Beck et al., 2001; Racheva, Daneva, Herrmann, & Wieringa, 2010)	Continuous client feedback non-direct involvement
Organization Culture	Hierarchical, command and control culture - (Nerur et al., 2005; Rehman et al., 2010)	Collaborative, flat organizational culture, team empowerment and decision making leadership - (Beck et al., 2001; Rehman et al., 2010)	Hierarchical outside of project team, internal team empowerment

Table 3: Comparison between TPM, APM and Agile with Discipline

Overall the practices at IBM seem to be structured as a means of addressing a transition from traditional practices towards Agile. In order, to obtain some of the benefits of Agile while retaining some of the structure of traditional approaches, the IBM team has managed to merge approaches to fit their development needs.

12.Conclusion

Based on our research we conclude that one development method does not fit all software development projects in the case of IBM center in Chicago. Both agile and traditional (modified waterfall) are in use. We also conclude that agile methods have not replaced traditional methods. One important deciding factor is the customers' preference. Our data shows that more and more project managers in IBM are equipped to manage agile projects. Project managers in IBM do not follow any agile method as described in literature. They follow a hybrid model that they call Agile-with-Discipline. This hybrid-agile allows flexibility to have continuous changes to requirements throughout the project development process but at the same time ensures that proper tools, techniques and supporting documentation are done. While the literature downplays the need for documentation in agile projects our data shows that proper documentation is still needed especially on external / customer projects. Lastly, our data does not support the opinion that agile teams are completely self-organized and self-managed.

REFERENCES

Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005). Agile project management: steering from the edges. *Communications of the ACM, 48*(12), 85-89.

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management, 2*(1), 26-30.

Baskerville, R., Ramesh, B., Levine, L., Pries-Heje, J., & Slaughter, S. (2003). Is internet-speed software development different? *IEEE software, 20*(6), 70.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001). Manifesto for agile software development.

Binder, J., Aillaud, L. I., & Schilli, L. (2014). The project management cocktail model: An approach for balancing agile and ISO 21500. *Procedia-Social and Behavioral Sciences, 119*, 182-191.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE software, 22*(5), 30-39.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer, 21*(5), 61-72.

Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International digital library perspectives, 27*(1), 18-22.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer, 34*(11), 131-133.

Diebold, P., & Dahlem, M. (2014). *Agile practices in practice: a mapping study.* Paper presented at the Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering.

Fernandez, D. J., & Fernandez, J. D. (2008). Agile project management—agilism versus traditional approaches. *Journal of Computer Information Systems, 49*(2), 10-17.

Guest, G., Bunce, A., & Johnson, L. (2006). How many interviews are enough? An experiment with data saturation and variability. *Field methods, 18*(1), 59-82.

Hass, K. B. (2007). The blending of traditional and agile project management. *PM world today, 9*(5), 1-8.

Hoda, R., Noble, J., & Marshall, S. (2008). *Agile project management.* Paper presented at the New Zealand Computer Science Research Student Conference.

Kerzner, H. R. (2013). *Project management: a systems approach to planning, scheduling, and controlling*: John Wiley & Sons.

Koskela, L., & Howell, G. (2002). *The underlying theory of project management is obsolete.* Paper presented at the Proceedings of the PMI Research Conference.

Laanti, M., Salo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology, 53*(3), 276-290.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., . . . Zelkowitz, M. (2002). *Empirical findings in agile methods.* Paper presented at the Conference on Extreme Programming and Agile Methods.

McHugh, M., McCaffery, F., & Casey, V. (2012). *Barriers to adopting agile practices when developing medical device software.* Paper presented at the International Conference on Software Process Improvement and Capability Determination.

Moniruzzaman, A., & Hossain, D. S. A. (2013). Comparative Study on Agile software development methodologies. *arXiv preprint arXiv:1307.3356.*

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *IJCSI, 5*, 95-101.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM, 48*(5), 72-78.

Paetsch, F., Eberlein, A., & Maurer, F. (2003). *Requirements Engineering and Agile Software Development.* Paper presented at the WETICE.

Papatheocharous, E., & Andreou, A. S. (2013). *Evidence of agile adoption in software organizations: An empirical survey.* Paper presented at the European Conference on Software Process Improvement.

Racheva, Z., Daneva, M., Herrmann, A., & Wieringa, R. J. (2010). A conceptual model and process for client-driven agile requirements prioritization.

Rahmanian, M. (2014). A Comparative Study on Hybrid IT Project Managment. *International Journal of Computer and Information Technology, 3*(05).

Rehman, I. U., Rauf, A., & Shahid, A. A. (2010). *Scope management in agile versus traditional software development methods.* Paper presented at the Proceedings of the 2010 National Software Engineering Conference.

Rose, K. H. (2013). A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition. *Project Management Journal, 44*(3), e1-e1.

Royce, W. W. (1970). *Managing the development of large software systems.* Paper presented at the proceedings of IEEE WESCON.

Schwaber, K. (1997). Scrum development process *Business Object Design and Implementation* (pp. 117-134): Springer.

Schwaber, K. (2004). *Agile project management with Scrum*: Microsoft press.

Schwaber, K., & Beedle, M. (2002). Agile Software Development with Scrum.

Sixsmith, A., Freeburn, C., & Mooney, G. (2014). *Project Management in Practice: Views from the Trenches.* Paper presented at the The 24th International Business Information Management Association Conference.

Špundak, M. (2014). Mixed agile/traditional project management methodology–reality or illusion? *Procedia-Social and Behavioral Sciences, 119*, 939-948.

Thompson, P. (1991). The client role in project management. *International Journal of Project Management, 9*(2), 90-92.

Uikey, N., & Suman, U. (2012). *An empirical study to design an effective agile project management framework.* Paper presented at the Proceedings of the CUBE International Information Technology Conference.

Victor, R. (2003). Iterative and incremental development: A brief history. *IEEE Computer Society*, 47-56.

Williams, T. (2005). Assessing and moving on from the dominant project management discourse in the light of project overruns. *IEEE Transactions on engineering management, 52*(4), 497-508.