

Association for Information Systems AIS Electronic Library (AISeL)

International Research Workshop on IT Project
Management 2016

International Research Workshop on IT Project
Management (IRWITPM)

12-10-2016

How Agile Methods Inspire Project Management - The Half Double Initiative

L. T. Heeager
lith@mgmt.au.dk

P. Svejvig
psve@mgmt.au.dk

B. R. Schlichter
brs@mgmt.au.dk

Follow this and additional works at: <http://aisel.aisnet.org/irwitpm2016>

Recommended Citation

Heeager, L. T.; Svejvig, P.; and Schlichter, B. R., "How Agile Methods Inspire Project Management - The Half Double Initiative" (2016). *International Research Workshop on IT Project Management 2016*. 13.
<http://aisel.aisnet.org/irwitpm2016/13>

This material is brought to you by the International Research Workshop on IT Project Management (IRWITPM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in International Research Workshop on IT Project Management 2016 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

How Agile Methods Inspire Project Management - The Half Double Initiative

Heeager L.T.

Aarhus University, Denmark
lith@mgmt.au.dk

Svejvig P.

Aarhus University, Denmark
psve@mgmt.au.dk

Schlichter B.R.

Aarhus University, Denmark
brs@mgmt.au.dk

ABSTRACT

Increased complexity in projects has forced new project management initiatives. In software development several agile methods have emerged and are today highly implemented in practice. Observations of general project management practice show how it has been inspired by agile software development, but very little research addresses the issue of agile project management. In order to understand and to provide suggestions for future practice on how agility can be incorporated in general project management, this paper provides an analysis which compares ten characteristics of agile software development (identified in theory) and the Half Double Methodology developed by the Danish Project Half Double initiative; a Methodology developed with practitioners and tested in seven Danish case companies. The analysis shows how the general project management to a great extent has been inspired by agile methods, but also that general project management may be able to find more inspiration from agile methods.

Keywords

Agile Software Development, Project Management, the Danish Project Half Double (PHD).

INTRODUCTION

Recent ideas on management of projects introduce an abundance of approaches, i.e. by understanding the project as a temporary organization, where learning, diversity, temporality, complexity, uncertainty and sociability are in play (Winter and Szczepanek, 2009). The increased complexity of projects has led to the development of agile methods and to the acceptance of the need for a contemporary organization of projects that not only mechanically executes a process toward a narrow, product-oriented goal, but accepts projects as business-like and value-creating (Davenport, 2013). Agile methods are highly used in the development of software, based on a belief that a project in broad understanding continuously needs to be adjusted based on the learning acquired during the process. The use of agile methods in software development settings seems to have a greater impact on success factors than just pure efficiency (Serrador and Pinto, 2015).

Traditionally a project has been seen as a tool applied to a single assignment, focusing on meeting time and quality under the available resources. This single-track approach applies a logical and chronological way through a set of (more or less) well-defined tasks. This approach has been challenged in the general understanding on how to modernize project management, especially in the stream of research related to Rethinking Project Management (Svejvig and Andersen, 2015).

But how can projects outside the software domain apply agile methods in practice in order to take advantage of the proven strengths of these methods? Doing so is not trivial as some practices are easier to transfer than others, and adopting agile methods in general project management settings requires a change from command and control management to leadership and collaboration (Misra, Kumar and Kumar, 2010; Nerur, Mahapatra and Mangalaraj, 2005). This again requires a reorientation not only for the project team, but also from the management (Moe, Dingsøyr and Dybå, 2010).

It is claimed that reorientation and construction of these above mentioned management models requires a deep understanding of the “agility” construct in management practices (Conforto, Salum, Amaral, da Silva and de

Almeida, 2014), which has been the inspiration for the present study and leading to the following research question: *How has Agile Software Development inspired the Half Double Methodology and what can general project management learn from this?* According to the Merriam-Webster dictionary, "to inspire" can be defined as "making (someone) want to do something" or "giving (someone) an idea about what to do or create". So to inspire is to influence someone else without enforcing your opinions or ways of doing on the recipient. Thus, in order to identify agile project management practices used in industry today, we are investigating which ideas from agile software methods the Half Double Methodology is building on.

We pursue to answer the research question by drawing on the Half Double Methodology developed by the Danish Project Half Double (PHD) initiative – an industry-driven initiative by a consultancy firm involving several private and public organizations including three universities (Svejvig and Grex, 2016). The Half Double Methodology consists of three core elements: impact, flow and leadership; the methodology is inspired by similar work about a new mind-set for management (Hamel, 2009), but also highly inspired by agile thinking (Schwaber, 2004) and is today tested in seven Danish case companies. This methodology is therefore a great representation of how current project management is done in industry. PHD has a profound desire to change the practices in projects and project management much in line with the agile manifesto (Beck et al., 2001). Project Half Double is overall conducted as collaborative research (Mathiassen, 2002; Van de Ven, 2007) where practitioners and researchers share ideas and are involved in activities to co-produce knowledge about the PHD initiative. In this paper we use comparative research where literature about agile methods is used as a theoretical lens to understand and explain aspects of the Half Double Methodology.

The remainder of this paper is organized as follows: section 2 presents a brief summary of the characteristic of Agile Software Development. The research setting and approach are then described as well as data collection and data analysis. The paper continues with a section presenting the empirical data, followed by the comparative analysis presenting similarities and differences between Agile Software Development and Project Management Practices of the Half Double Methodology, providing knowledge on how Agile Software Development has inspired general Project Management Practices. The paper ends with concluding remarks and suggestions for further research.

AGILE SOFTWARE DEVELOPMENT

Although the roots of agile methodologies go way back (Highsmith, 2002), the agile manifesto published in 2001 by a group of practitioners set the wheels rolling on agile methods. The manifest constitutes four values and a set of practices for agile software development. The four values are: 1) individuals and interactions over processes and tools 2) working software over comprehensive documentation 3) customer collaboration over contract negotiation 4) responding to change over following a plan (Beck et al., 2001). Adhering to these values, agile methods suggest a way to deliver software without excessive cost (Boehm and Turner, 2004). Several methods fulfilling these agile characteristics have emerged. Table 1 summarises well-known agile methods.

Agile Method	Features	Reference
Dynamic Systems Development Method	DSDM focuses on people and on the needs of a business delivering software solutions as quickly as possible.	(Stapleton, 1999)
Adaptive Software Development	With the use of adaptive cycles ASD is highly iterative and tolerant toward change.	(Highsmith, 2013)
Feature Driven Development	FDD uses cycles which are customer and feature-centered. The first step is to discover the list of features to implement and then implement it - feature-by-feature.	(Palmer and Felsing, 2001)
eXtreme Programming	XP offers a range of practices and principles to apply in order to focus on close customer contact and flexibility in responding to change.	(Beck and Andres, 2004)
Crystal	The Crystal family of methods provides methods corresponding to the size and criticality of the project.	(Cockburn, 2006)
Kanban	Kanban is an approach to introduce change to an existing methodology.	(Kniberg, 2009)
Scrum	Scrum is an iterative and incremental method which emphasizes project management values and practices.	(Schwaber and Beedle, 2001)

Table 1. Seven Agile Methods and their features

Agility is a multidimensional construct (Lee and Xia, 2010). To further understand the basic characteristics of agile software development, we draw on the definition of agility developed by (Conboy, 2009):

“the continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality and simplicity), through its collective components and relationships with its environment.” (Conboy 2009, p. 340).

The definition shows how accepting change as an inevitable part of projects is vital. This is done while focusing on the customers and their values (referring to features of the product). It furthermore involves the collective elements and relationships (individuals and interactions from the manifest). This shows four important cornerstones of agility:

- 1) Change acceptance
- 2) Team focus
- 3) Customer focus
- 4) Product quality

Practices of Agile Software Development

In order to compare agility with the Half Double Methodology, the four cornerstones of agility are unfolded to identify the practices used in agile methods. In this section, 10 practices identified through a literature study of agile software development are presented. In Figure 1 these practices are mapped to the four cornerstones of agility.

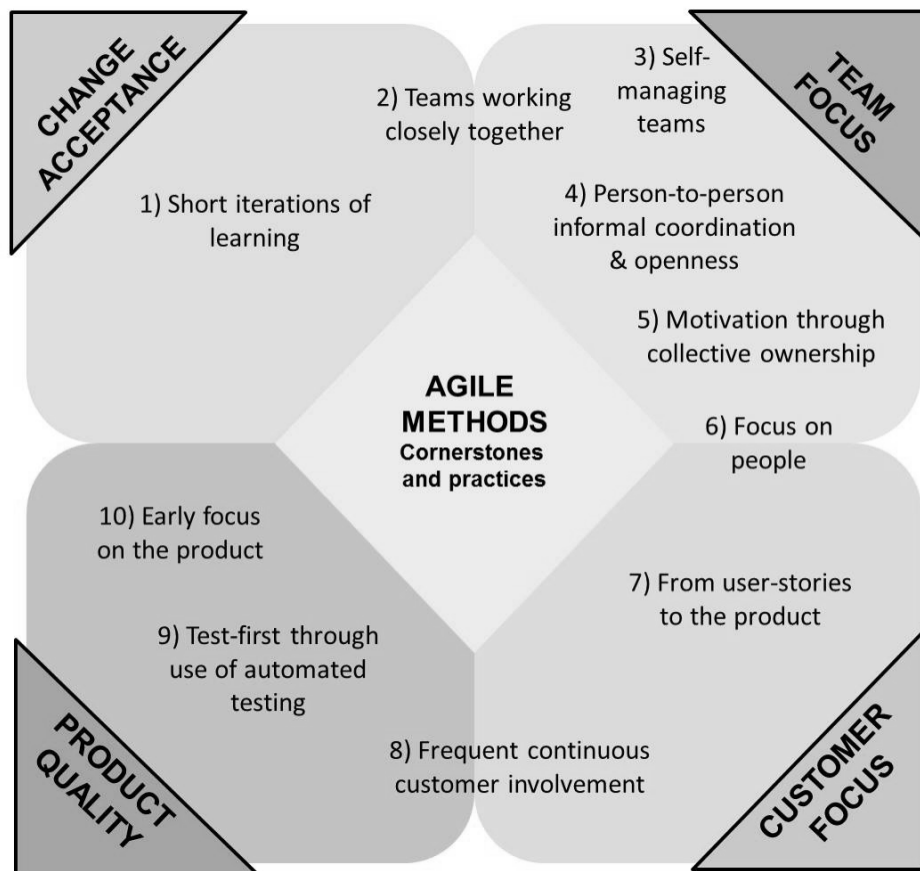


Figure 1. The cornerstones and practices of agile methods

1) Short iterations of learning: Agile methods propose iterations that include all phases of the development process (Cohn and Ford, 2003). The goals of the iterations are to create flexibility and possibilities of adapting to changes.

The focus on accepting change is essential in several papers investigating aspects of agility; the existence of uncertainty and complexity in environments is evident (Nerur and Balijepally, 2007). Changes can be both technical and new business opportunities as well as the fact that the people involved are learning throughout the project (Lyytinen and Rose, 2006). Dealing with changing requirements is imperative in software development (Lee and Xia, 2010). Iterations or cycles are heavily used in agile methods: XP, for example, introduces the concepts of weekly cycles (to plan work a week at a time) and quarterly cycles (to reflect on the work every quarter) (Beck and Andres, 2004), whereas Scrum implements the sprint in which the developers work without interruptions (Schwaber and Beedle, 2001).

2) Teams working closely together: This practice supports the cornerstone of change acceptance, since reducing the cost and time of exchanging information between people helps the team respond more effectively to change. This is done by placing people physically closer together (Cockburn and Highsmith, 2001). It also supports the cornerstone of team focus. Various agile methods offer guidance on how to support close team work. The Scrum team is the developers and testers and other people working on the software. In order to ensure that the team works closely together, the teams are small (Schwaber and Beedle, 2001). XP advocates that the team sits together on a daily basis and use team activities such as pair programming (Beck and Andres, 2004). In Scrum the team is supported with a product backlog (controlled by the product owner) from which the team agrees and commits to the tasks included in each sprint backlog. These artefacts hereby support the work of the team (Schwaber and Beedle, 2001).

3) Self-managing teams: The overall management styles in agile methods are leadership and collaboration (Misra et al., 2010). Even though this practice is placed in the cornerstone of team focus, it has an indirect influence on both change acceptance and product quality. Agile teams are self-organization teams; hence teams that are able to organize in various configurations and meet challenges when they arise (Cockburn and Highsmith, 2001). Such teams are important in order to create flexibility and responsiveness to change (Nerur and Balijepally, 2007) and to create a high quality product (Chow and Cao, 2008). The manager takes a facilitation role. The scrum master is, for example, mainly in charge of ensuring that the scrum practice is followed and that the team gets to work focused during the sprint (Schwaber and Beedle, 2001).

4) Person-to-person informal coordination and openness: The agile methods rely on person-to-person communication and knowledge sharing and advocates openness; in the terms used by (Hansen, Nohria and Tierney, 1999), the agile methods primarily rely on a personalization strategy. The frequent meetings (in Scrum: stand-up meetings, planning meetings and retrospectives) proposed by the agile methods serve as ways to informally communicate and share knowledge (Chau and Maurer, 2004). Information on the project and its status must be provided in the open space, this could for example include various artefacts created by the teams, such as task boards and burn down charts (Cockburn, 2006).

5) Motivation through collective ownership: In agile software development, all team members are responsible for reaching the goal of the iteration. Ideally, every team member is able to work on any task assigned to the iteration. Working in self-organizing teams with common responsibility for tasks (Schwaber and Beedle, 2001) and code (Beck and Andres, 2004) enhances team motivation and collaboration. Committing to the goal of an iteration as a team is also a way of boosting motivation and team spirit (Heeager and Rose, 2015).

6) Focus on People: As the first value of the manifest suggests, agile methods focus on individuals over processes (Beck et al., 2001). The agile methods focus on the social aspects of software development and people issues are at the heart of the agile movement (Galal-Edeen, Riad, and Seyam, 2007). Both the team and the customer are essential and this practice is closely connected to the practice of self-managing teams and frequent customer involvement. This focus on people and less focus on processes poses new requirements for the qualifications of the project members, such as: talent, skill and communication (Cockburn and Highsmith, 2001). Attention therefore needs to be given to people-related factors such as staffing and communication (Boehm and Turner, 2003).

7) From visual user stories to the product: Agile methods break with the traditional term requirements, and instead they rely primarily on user stories written by the customer in a plain business-like language (Cohn, 2004). This practice primarily supports the cornerstone of product quality as user stories are implemented in order to make sure that the product fits the user. The practice also indirectly supports change acceptance and customer focus. User stories involve the user to a high degree as the system will be built based on their descriptions. User stories furthermore invite changes to a greater extent than traditional requirements that symbolize something mandatory

(Beck and Andres, 2004). In Scrum, stories are collected in the product backlog and prioritized by the product owner (a customer representative) (Schwaber and Beedle, 2001). Stories are to be displayed on large boards placed in the work area, called information radiators (Cockburn, 2006).

8) Frequent continuous customer involvement: This practice supports the cornerstones of product quality and customer focus. The customer plays an essential role in agile development by introducing heavy customer involvement as a means for striving for high quality of the product and at the same time, frequent customer involvement also feeds back to the customer on the implications of their choices (Cockburn and Highsmith, 2001). A strong customer involvement has been identified as an essential success factor for agile projects (Chow and Cao, 2008). XP suggests real customer involvement, as the customers are the ones you seek to please (Beck and Andres, 2004), while Scrum uses the role of a product owner who officially is responsible for the project, and ideally this role is filled by the customer (Schwaber and Beedle, 2001).

9) Test first through use of automated testing: Test-driven development has become increasingly popular within the agile community (Nerur et al., 2005). In XP, for example, testing is a core practice; it is not only suggested that test cases are written before the software, it is also more importantly suggested that all parts of an increment are tested and that completion is determined by the increment passing all tests (Beck and Andres, 2004). Due to the heavy reliance on testing, agile methods focus on automating the tests. XP aims at reducing the cost of testing by implying that all tests are automated (Beck and Andres, 2004) and in practice it is also acknowledged that automated tests are necessary in order to complete short iterations (Jakobsen and Johnson, 2008).

10) Early focus on the product: One of the agile principles is: “working software over comprehensive documentation”. Although agile methods do not focus on documentation, they do not cast all documentation aside (Baker, 2005). Agile methods cut away unnecessary documentation for a greater focus on the product and seek to have at least a part of the system/product ready early in the process (Theunissen, Kourie and Watson, 2003). In Scrum the product is released as often as possible (Schwaber and Beedle, 2001).

RESEARCH METHOD

This paper is focused on comparing the Half Double Methodology with the agile research stream. The specific research design for this paper is qualitative comparative research (Bryman, 2008, pp.: 58-61). The primary data collection methods included participation in workshops and meetings from February 2014 to October 2015 with pilot organizations, the Danish Industry Foundation and a consultancy company. The workshops and meetings are documented by written material, videos and field notes taken by the authors – especially a report covering the preliminary results for phase 1 of PHD has been important for this comparative analysis (Svejvig et al., 2016). Informal talks with members of the PHD network have also broadened the understanding of PHD thinking and how practitioners relate this to their lifeworld (Berger and Luckmann, 1966; Schutz, 1967). This paper builds and extends on earlier studies about the Project Half Double (Heeager, Svejvig and Schlichter, 2016; Svejvig and Grex, 2016; Svejvig and Hedegaard, 2016).

THE HALF DOUBLE INITIATIVE

The Half Double initiative was started in 2013 as an informal network of committed people at different levels of Danish industry who discussed how to develop project management in the light of the apparent high failure rate of projects e.g. CHAOS Reports (Hastie and Wojewoda, 2015; Standish Group, 2015) and other studies claiming high failure rates, and with the ambition to manage projects in a radically different way. The initiative was centered on the “Implement Consulting Group”, a Scandinavian-based management consultancy company with more than 450 consultants on board, and with a global reach to help organizations change. The initiative matured and began gradually to formalize during the spring 2014, and the work manifested into 10 leading stars which have been developed and discussed at four workshops from February 2014 to January 2015 with a broad representation from areas such as manufacturing, finance, insurance, IT, public administration, management consultancies, universities, and the Confederation of Danish Industry. The 10 leading stars have later developed into a more carefully prepared Half Double Methodology focusing on impact, flow, and leadership during 2016. The formal project kick-off took place in June 2015 where seven pilot projects from seven industry organizations have tested the methodology until the end of June 2016 (Svejvig et al., 2016).

THE HALF DOUBLE METHODOLOGY

Project Half Double was initiated with a clear mission to succeed in finding a project methodology that could increase the success rate of projects while increasing the speed at which new products and services were developed. PHD was convinced that by doing so it could strengthen Danish competitiveness and play an important role in the battle for jobs and future welfare. The core elements, principles, methods, and tools are presented in their current form in Figure 2 (Svejvig et al., 2016, p.: 10):

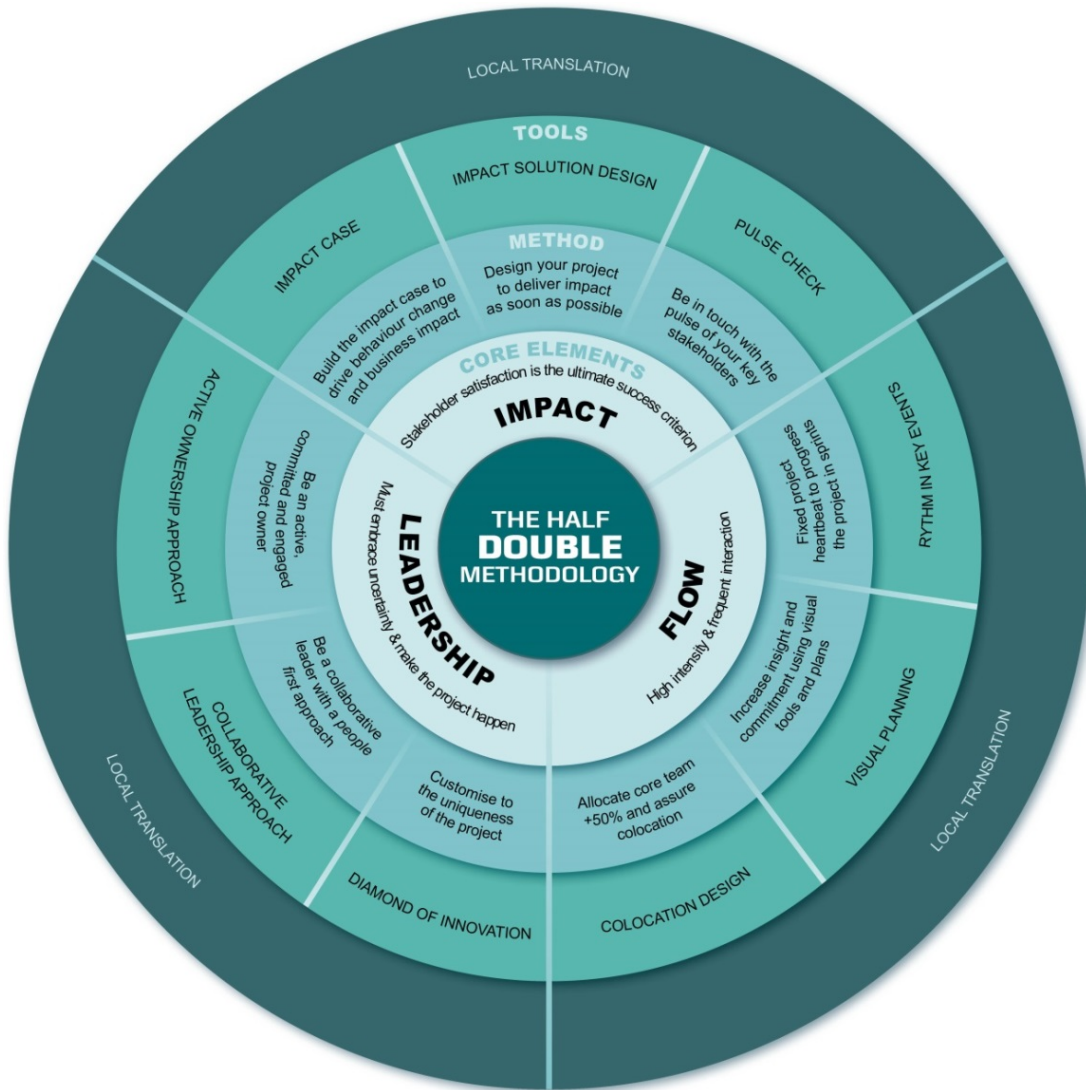


Figure 2. The Half Double Methodology

The intention in this paper is not to unfold the Half Double Methodology in detail, but instead to emphasize important characteristics of the methodology – details can be found elsewhere (Svejvig et al., 2016).

Core element impact: The principle is “*stakeholder satisfaction is the ultimate success criterion*”. No project exists for the project's own sake. All projects are initiated to create impact. Identifying and focusing on impact right from the start is the key. Impact changes the dialogue from being centered on technical deliverables to how to ensure stakeholder satisfaction throughout the project’s lifecycle. The following methods are used to realize impact in practice:

(1) Use an impact case to drive behavioral change and business impact. Projects should be driven by impact rather than deliverables. An impact case should be formulated that lists, prioritizes, and visualizes the business and behavioral impact which the project is set out to create together with key stakeholders and subject matter experts.

(2) Design the project to deliver impact as soon as possible with end-users close to the solution, and move away from the premise that projects only generate value at the very end of their lifespan. Seek to create early insights through fast prototyping and generate impact during the process – faster.

(3) Be in touch with the pulse of key stakeholders on a monthly basis. Acknowledging and working actively with the dynamic nature of projects is a key to succeed. Interests and focus change rapidly, so there is a need to gain insights and facilitate a dialog among the right people on an ongoing basis to ensure engagement and continuous focus on the right impact.

Core element flow: The principle is “*high intensity and frequent interaction to ensure continuous project progression*”. Pursue to create flow in the project. The whole project group should be busy at the same time – not just selected individuals in the project team. However, important project working hours are often lost in coordination, retrospective project reporting and shifts between multiple projects running simultaneously. Focus on the flow of the project and use simple methods to intensify project work, ensure the project progress every week and deliver results - faster. The following methods are used to enhance flow in practice:

(1) Allocate a team +50% and assure colocation. Reduce complexity in time and space to free up time to solve complex problems. At a portfolio level there is a best practice approach aimed at ensuring “short and fast” projects – meaning fewer projects with a more intense resource allocation.

(2) Define a fixed project heartbeat for stakeholder interaction to progress the project in sprints. A fixed project heartbeat creates higher energy, higher efficiency, better quality and ultimately faster development speed.

(3) Increase insight and commitment using visual tools and plans to support progression. When operating in a project mode with high intensity and many touchpoints with both internal and external stakeholders, it is important to find an efficient way of communicating progress and solutions as well as progress and traction.

Core element leadership: The principle is “*leadership embraces uncertainty and makes the project happen*”. Revolutionize the way projects should be lead with less bureaucracy, less formal steering committee meetings, and less contractual focus. More commitment is needed - and less compliance. Leaders have to cope with turbulence, conflicts and people. Leaders should focus on the human aspects, work closely together with the project team on a regular basis, handle issues and complexity in joint force and know the project in its core. The following methods are used to enhance flow in practice:

(1) Be an active, committed and engaged project owner to support the project and ensure stakeholder satisfaction. Research suggests one common denominator across all successful projects; an active, committed project owner who engages directly with the project on an ongoing basis.

(2) Be a collaborative project leader (not manager) with a “people first” approach to drive the project forward. It is no longer enough to be a trained technician who can follow detailed procedures and techniques prescribed by project management methods and tools. Collaborative project leadership is about leading a complex system of human beings, embracing the inevitable uncertainty and making the project happen.

(3) Customize to the uniqueness of the project. Projects are unique and hence one size does not fit all. Each project needs to be customized to the specific governance and local best practice models to succeed. The customization is the first step in the local translation of the Half Double Methodology to fit the context.

Mobilize the Half Double mind-set to assist the local translation: The Half Double Methodology is an approach to leading projects that requires rethinking current practice. It requires a change of mind-set and a change of behaviour. Implementing Half Double is implementing change.

It is a two-way street. On the one hand, there is a need for aligning and tailoring the Half Double Methodology to the situation at hand – that is organizational structures, cultures and to the local nature of the projects. There is no

“one-size-fits-all” and the project, the methods and tools must be designed to fit the local conditions. On the other hand, the organization also needs to be adapted to be in alignment with the Half Double mind-set. There must be executive level commitment and willingness to think differently (the presentation of the Half Double Methodology is an extract from Svevig et al. (2016, pp.: 9-16).

COMPARING AGILE METHODS WITH THE HALF DOUBLE METHODOLOGY

This section presents the comparison between the practices of the agile software methods and general project management as practiced by the Half Double Methodology. It is important to emphasize that the two methods were developed for two specific and distinct domains; thus some translation has been necessary to perform the comparison, shown in Table 2 and described shortly below.

#	Agile Practices	Influence on the Half Double Methodology			Fulfilled
		Impact	Flow	Leadership	
1	Short iterations of learning	Deliver impact as soon possible.	Fixed project heartbeat.		Yes
2	Teams working closely together	Pulse check of the project team (partly).	Colocation design.		Yes
3	Self-managing teams			Collaborative project leader (facilitating a people process).	Partly
4	Person-to-person informal coordination and openness		Colocation design. Fixed heartbeat (formal meeting facilitates informal dialogue). High intensity and frequent interaction.		Partly
5	Motivation through collective ownership		Allocate a core team +50% and assure colocation => Fertilizes the ground for motivation and collective ownership.	Be a collaborative manager with a “people first” approach.	Partly
6	Focus on people	Pulse check. Be in touch with your key stakeholders.		Be a collaborative manager with a “people first” approach.	Yes
7	From visual user stories to product		Visual planning through mock-ups and fast prototyping (partly).		Partly
8	Frequent continuous customer involvement	Be in touch with your key stakeholders (pulse check).	Heartbeat for stakeholder interaction.	Customer = Project Owner, at least three hours of biweekly interaction.	Yes
9	Test first through automated testing				No
10	Early focus on the product	Impact case focuses on early impact (value), creates a focus on the development of the product.			Yes

Table 2. Comparing the practices of agile software development with the PHD Method

The analysis showed how several of the agile practices (five of the ten) are fulfilled by the Half Double Methodology; furthermore four of the agile practices were partly fulfilled, whereas only one agile practice was not

fulfilled. The analysis also showed how all three core elements: impact, flow and leadership have found inspiration from the agile methods. Thus, the Half Double Methodology has up to a point been influenced by all four of the agile cornerstones and nine of the ten practices.

The Half Double Methodology uses a fixed project heartbeat, early delivery of impact and pulse check of the collocated team which corresponds to the iterations and the close teamwork proposed by the agile methods (Practice 1 and 2). This shows how the Half Double Methodology has been highly inspired to introduce flexibility and change acceptance.

The heavy focus on the team by the agile methods has, however, only been partly adopted by the Half Double Methodology, as it focuses on collocation, facilitation and collaboration (Practice 3-5). Still, teams are not labeled fully self-managing, no guidance on how to support communication strategy based on personalization and openness is provided, and the motivation and collective ownership of the team is only supported indirectly.

Like the agile methods, the Half Double Methodology has focus on the people aspects (Practice 6), both stakeholders and the project team. The frequent, continuous customer involvement (Practice 8) has been fully adopted in the Half Double Methodology and influences all three core elements of the methodology. User stories (Practice 7) are a specific artifact in software development and as such not a part of the Half Double Methodology, however creating visibility through mock-ups is part of the method. The Half Double Methodology has therefore to a great extent been inspired by the customer focus of agile methods.

The product focus is however more indirect in the Half Double Methodology. Since it is designed for general project management, tests and automation (Practice 9) have not been given focus. Yet, it does focus on early impact (Practice 10), and through this practice a secondary focus on the product is found.

The comparison between agile methods and the Half Double Methodology is at a rather high level, and an analysis at a more detailed level might give more fine-grained similarities and differences, but this high-level comparison does in itself emphasize that agile methods from software development certainly have diffused into general project management practices at least in this specific example of the Half Double Methodology. Other studies support the same tendency (Conforto et al., 2014; Serrador and Pinto, 2015).

DISCUSSION AND CONCLUSION

This paper aims to open up the discussion of introducing or maintaining agility in general project management. In this section we discuss the degree to which the agile methods have inspired project management as practiced by the Half Double Methodology and what implications this has for future project management practice.

With the agile methods in software development, the domain witnesses a shift from the traditional methods. This shift has come from several project failures and the acknowledgement that not all change can be avoided (Abrahamsson, Conboy and Wang, 2009). During the last 15 years, since the agile manifest was published, the use of agile methods has increased rapidly (Lindvall et al., 2002). Within the field of project management there has also been a break with the traditional way of doing projects. This has given rise to the literature stream of rethinking project management (Svejvig and Andersen, 2015; Winter, Smith, Morris and Cicmil, 2006). Agility has also been given some focus within project management, the literature on the topic is however very scarce.

This paper answers the first part of the research question: How has Agile Software Development inspired the Half Double Methodology? By identifying how the ten practices of agile software development have *influenced* the Half Double Methodology, what can general project management then learn from this? It appears that most of the practices are mirrored in the Half Double Methodology (see Table 2). The analysis presented showed how the Half Double Methodology to a large extent has been inspired by agile methods, but also how there is room for increasing agility in the Methodology. Tripp, Riemenschneider and Thatcher (2016) suggest that agile practices can be divided into two primary categories: 1) practices related to the software development approach (such as different types of testing) and 2) practices related to project management (such as iterative delivery). The majority of the ten practices, which we have identified in this paper, can be categorized as project management practices (practice 1-6, 8 and 10), whereas only a few are more specific software development practices (practice 7 and 9). It is not very surprising that the software development specific practices: "From visual user stories to product" and "Test first through automated testing" were scarcely addressed. However, the analysis also showed how some of the people-oriented practices:

"self-managing teams"; "person-to-person information and openness"; "motivation through collective ownership" are scarcely addressed even though they can be characterized as project management practices.

The paper also answers the second part of the research question: How has Agile Software Development inspired the Half Double Methodology and what can general project management learn from this? Based on the analysis, we derive five lessons learned (illustrated in Figure 3), which are the main contribution of this paper. No other paper has explicitly documented the agility of Project Management Practices as practiced in today's industry.

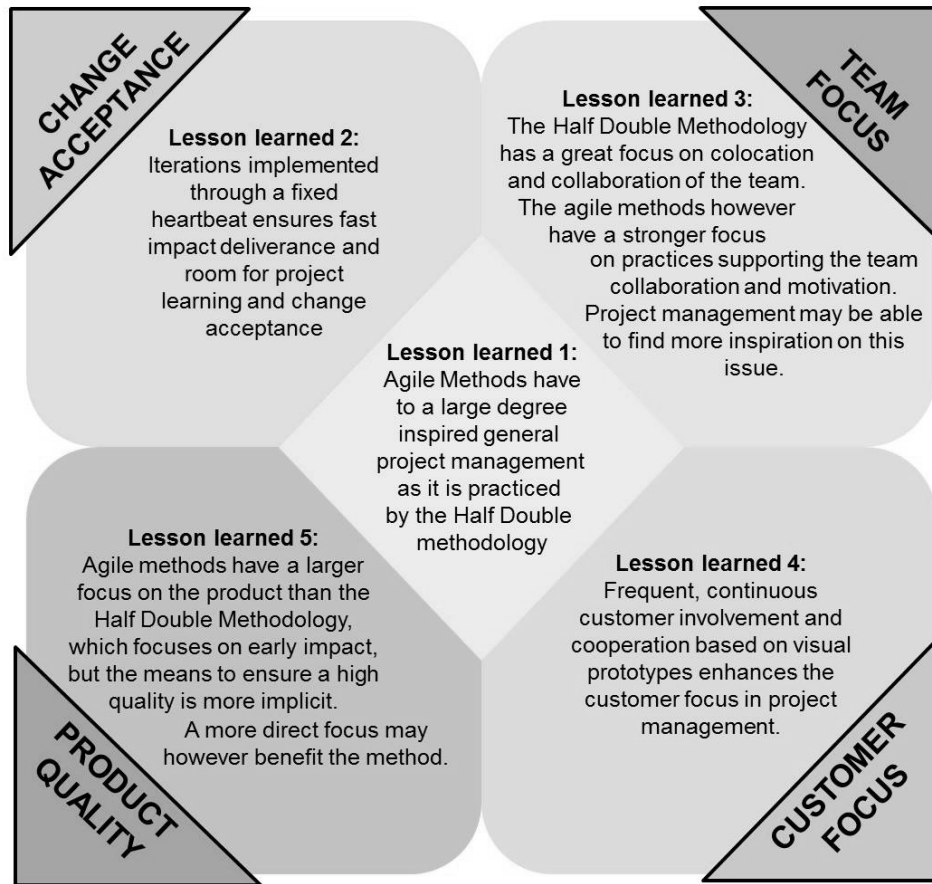


Figure 3. Lessons Learned

We dare to suggest that by focusing on Impact, Flow, and Leadership, agility can (partly) be conceptualized and applied in general project management in a useful way, conceptualized using the last four Lessons Learned. But it also seems that the lessons learned tell us that in some areas, more effort is needed to fully take advantage of the agile principles. More focus is especially needed on team collaboration and motivation as well as a more direct and contextualized focus on the product to be produced as a result of the general project management.

This conceptual mapping opens a set of interesting avenues for further research. Among these we may look deeper into specific aspects of the Half Double Methodology and how this can be mapped into an agile approach or we may look deeper into more specific types of agile development, such as XP or Scrum. The above mentioned issues are in line with an executive report where (Highsmith, 2003) presents nine principles, five stages and their corresponding tools and also in line with (Boehm and Turner, 2003) who discuss people issues for agile software management.

Limitations

We acknowledge the limitations of the present research and understand that the conceptualization must be challenged further by applying it on more cases and by analyzing both the small discrepancies and how the actors understand and apply these agile principles in their daily project practices.

ACKNOWLEDGEMENT

The authors would like to thank the Danish Industry Foundation for funding this work, and acknowledge contributions from Danish organizations involved in the Project Half Double and the Implement Consulting Group. The authors declare that they have no conflict of interest regarding the funding agency and other parties involved in the Project Half Double.

REFERENCES

- Abrahamsson, P., Conboy, K. and Wang, X. (2009) "Lots done, more to do": the current state of agile systems development research, *European Journal of Information Systems*, 18, 4, 281-284.
- Baker, W. (2005) Formalizing agility: An agile organization's journey toward CMMI accreditation, in Mary Manns and William Wake (Eds.) *Proceedings of the Agile Development Conference*, July 24-29, Denver, USA, IEEE, 185-192.
- Beck, K. and Andres, C. (2004) *Extreme programming explained: Embrace change*, Addison-Wesley, Boston.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Sutherland, D. and Thomas, D. (2001). Manifesto for Agile Software Development, Retrieved from: <http://agilemanifesto.org/>
- Berger, P. L. and Luckmann, T. (1966) *The Social Construction of Reality. A Treatise in the Sociology of Knowledge*, Doubleday, New York.
- Boehm, B. and Turner, R. (2003) People factors in software management: Lessons from comparing agile and plan-driven methods, *Crosstalk - The Journal of Defense Software Engineering*, December, 4-8.
- Boehm, B. and Turner, R. (2004) Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods, in Anthony Finkelstein, Jacky Estublier and David Rosenblum (Eds.) *Proceedings of the 26th International Conference on Software Engineering (ICSE 2004)*, Vancouver, Canada, July 1-3, IEEE, 718-719.
- Bryman, A. (2008) *Social Research Methods*, Oxford University Press, Oxford.
- Chau, T. and Maurer, F. (2004) Knowledge sharing in agile software teams, in Wolfgang Lenski (Eds.), *Logic versus approximation*, Springer, 173-183.
- Chow, T. and Cao, B. (2008) A survey study of critical success factors in agile software projects, *Journal of Systems and Software*, 81, 6, 961-971.
- Cockburn, A. (2006) *Agile software development: The Cooperative game*, Addison-Wesley Professional, Boston.
- Cockburn, A. and Highsmith, J. (2001) Agile software development, the people factor, *Computer*, 34, 11, 131-133.
- Cohn, M. (2004) *User stories applied: For agile software development*, Addison-Wesley Professional.
- Cohn, M. and Ford, D. (2003) Introducing an agile process to an organization, *Computer*, 36, 6, 74-78.
- Conboy, K. (2009) Agility from first principles: reconstructing the concept of agility in information systems development, *Information Systems Research*, 20, 3, 329-354.
- Conforto, C., Salum, F., Amaral, C., da Silva, L. and de Almeida, M. (2014) Can agile project management be adopted by industries other than software development?, *Project Management Journal*, 45, 3, 21-34.
- Davenport, H. (2013). *Process innovation: reengineering work through information technology*, Harvard Business Press.
- Galal-Edeen, G., Riad, A. and Seyam, M. (2007) Agility versus discipline: Is reconciliation possible? in Ahmed Zaki Badr and Ain Shams (Eds.) *Proceedings of the International Conference on Computer Engineering & Systems (ICCES)*, November 27-29, Cairo, Egypt, 331-337.
- Hamel, G. (2009) Moon Shots for Management, *Harvard Business Review*, 87, 2, 91-98.
- Hansen, M., Nohria, N. and Tierney, T. (1999) What's your strategy for managing knowledge?, *Harvard business review*, 77, 2, 106-116.
- Hastie, S. and Wojewoda, S. (2015) Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch, Retrieved from <http://www.infoq.com/articles/standish-chaos-2015>
- Heeager, L. and Rose, J. (2015) Optimising agile development practices for the maintenance operation, *Empirical Software Engineering*, 20, 6, 1762-1784.

- Heeager, L., Svejvig, P. and Schlichter, B. (2016) How Agile Methods Conquers General Project Management - the Project Half Double Initiative, *Proceedings of the 39th IRIS Seminar*, August 7-10, Ljungskile, Sweden.
- Highsmith, J. (2002) What is Agile Software Development? *Crosstalk: Journal of Defense Software Engineering*, 15, 10, 4-9.
- Highsmith, J. (2003) Agile project management: Principles and tools, *Cutter consortium*, 4, 2, 1-37.
- Highsmith, J. (2013) Adaptive software development: a collaborative approach to managing complex systems, Addison-Wesley.
- Jakobsen, C. and Johnson, K. (2008) Mature agile with a twist of CMMI, in Grigori Melnik, Philippe Kruchten, Mary Poppendieck (Eds.) *Proceedings of the Agile Conference*, August 4-8, Toronto, Canada, IEEE, 212-217.
- Kniberg, H. (2009) Kanban vs Scrum, *Crisp AB. Viitattu*, 1, 1-41.
- Lee, G. and Xia, W. (2010) Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility, *MIS Quarterly*, 34, 1, 87-114.
- Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams L. and Zelkowitz, M. (2002) Empirical findings in agile methods, in Don Wells and Laurie Williams (Eds.) *Proceedings of the 2nd XP Universe and First Agile Universe Conference*, August 4-7, Chicago, IL, USA, Springer, 197-207.
- Lyytinen, K., & Rose, G. M. (2006) Information system development agility as organizational learning, *European Journal of Information Systems*, 15, 2, 183-199.
- Mathiassen, L. (2002) Collaborative practice research, *Information Technology & People*, 15, 4, 321-345.
- Misra, S., Kumar, V., and Kumar, U. (2010) Identifying some critical changes required in adopting agile practices in traditional software development projects, *International Journal of Quality & Reliability Management*, 27, 4, 451-474.
- Moe, N. B., Dingsøy, T. and Dybå, T. (2010). A teamwork model for understanding an agile team, A case study of a Scrum project. *Information and Software Technology*, 52, 5, 480-491.
- Nerur, S. and Balijepally, G. (2007) Theoretical reflections on agile development methodologies, *Communications of the ACM*, 50, 3, 79-83.
- Nerur, S., Mahapatra, R. and Mangalaraj, G. (2005) Challenges of migrating to agile methodologies, *Communications of the ACM*, 48, 5, 73-78.
- Palmer, R. and Felsing, M. (2001) A practical guide to feature-driven development, Pearson Education.
- Schutz, A. (1967) *Phenomenology of the Social World*, Northwestern University Press, Illinois.
- Schwaber, K. (2004) *Agile project management with Scrum*, Microsoft Press Redmond, Washington.
- Schwaber, K. and Beedle, M. (2001) *Agile software development with Scrum*, Prentice Hall, New Jersey.
- Serrador, P. and Pinto, K. (2015) Does Agile work? - A quantitative analysis of agile project success, *International Journal of Project Management*, 33, 5, 1040-1051.
- Standish Group. (2015) CHAOS Report, Retrieved from <https://www.standishgroup.com/store/services/chaos-report-2015-blue-pm2go-membership.html>
- Stapleton, J. (1999) DSDM: Dynamic Systems Development Method, in Nancy France (Eds.) *Published by the IEEE Computer Society*.
- Svejvig, P. and Andersen, P. (2015) Rethinking project management: A structured literature review with a critical look at the brave new world, *International journal of project management*, 33, 2, 278-290.
- Svejvig, P., Ehlers, M., Adland, T., Grex, S., Frederiksen, H., Borch, M., and Pedersen, E. (2016) Project Half Double, Preliminary Results for Phase I, June 2016, Retrieved from Aarhus.
- Svejvig, P. and Grex, S. (2016) The Danish Agenda for Rethinking Project Management, *International Journal of Managing Projects in Business*, 9, 4, 822-844.
- Svejvig, P. and Hedegaard, F. (2016) The Challenges of Evaluating and Comparing Projects – an Empirical Study of Designing a Comparison Framework, in Jan Pries-Heje and Per Svejvig (Eds.), *in Project Management for Achieving Change*, Frederiksberg, Roskilde University Press, 107-129.
- Theunissen, H., Kourie, G. and Watson, W. (2003) Standards and agile software development, in Jarr Eloff, Andries Engelbrecht, Paula Kotzè and Miriam Eloff (Eds.) *Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, Johannesburg, 178-188.
- Tripp, F., Riemenschneider, C. and Thatcher, B. (2016) Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign, *Journal of the Association for Information Systems*, 17, 4, 267.
- Van de Ven, A. (2007) *Engaged scholarship: A guide for organizational and social research*, Oxford University Press, Oxford.

- Winter, M., Smith, C., Morris, P. and Ciemil, S. (2006) Directions for future research in project management: The main findings of a UK government-funded research network, *International journal of project management*, 24, 8, 638-649.
- Winter, M. and Szczepanek, T. (2009) *Images of projects*, Gower Publishing.