

AGILE PROJECT MANAGEMENT – A FUTURE APPROACH TO THE MANAGEMENT OF PROJECTS?

Aljaž Stare

University of Ljubljana, Faculty of Economics
Kardeljeva ploščad 17, Ljubljana, Slovenia
aljaz.stare@ef.uni-lj.si

Abstract

Like other professions, project management is constantly developing, new techniques are emerging, while newer, more effective approaches are being established. As an innovative modern approach, agile project management is most frequently mentioned and many experts argue that it is becoming the project management of the 21st century. The approach was developed in the field of software development and has delivered a number of novelties and benefits for the project team and the project client. On the other hand, it highlights some 'new' approaches that are not foreign to 'traditional' managers and raises a series of dilemmas about the actual benefits it brings. Even greater dilemmas arise when we attempt to use the approach for other types of projects. This speculative article summarises the most significant innovations introduced by the agile approach and critically assesses the possibility of extending the agile approach to other types of projects.

Keywords: project, management, agility

1. INTRODUCTION

Agile approaches to the management of software development projects (hereinafter IT projects) have developed significantly since the Agile Manifesto was published in 2001. According to many, especially the manifesto's authors, due to the recognised response to changes agility will become even more important, while Bennekum and Van Hunt (in Bowles Jackson, 2012) even argue that agile thinking is crucial for success in the 21st century!

In an interview with Bowles Jackson, the manifesto's authors also stated that the agile approach is useful for all types of projects and beyond. Asked directly whether agility can be applied outside of IT projects, Hunt further argues that the agile approach has little to do with software; instead, it is all about recognising and applying feedback. Van Bennekum

adds that agile is holistic and applicable everywhere in business or life – he uses it as a concept wherever he is and for whatever he does. The third co-author, Highsmith, believes it is necessary to use agile practices and principles in every project that faces uncertainty.

However, the practice has not yet confirmed their foresight and efforts. We checked research studies that have examined the agile approach in the last decade and found that to 2009 almost all of them focused solely on IT projects; in that year, one article explored the agile approach in product development projects (Berger & Beynon-Davies). A year later, Doherty (2010) researched agility projects in the field of e-learning, while Gonzalez (2011) examined the agile approach with regard to the establishment and management of intellectual property. In sum, of thirty-three articles only three did not discuss IT projects.

In this article, we would like to critically assess the possibility of extending the agile approach to other types of projects, and to point out that we are not discussing a completely different type of project management, but rather agile methods that build upon and enhance traditional project management. We will show the beneficial innovations brought by the agile approach, while pointing out the advantages and disadvantages of the approach in the context of different types of projects. Our purpose is to acquaint managers with the approach, and to stimulate professional discussion and research that will confirm (or disprove) the presented considerations.

In addition, we would also like to highlight the inappropriate use of the term **'agile project management'**. Science defines management as a process of planning, organising, leading and controlling; however, when studying the characteristics of the agile approach, it would be difficult to find the type of differences that would justify management being called 'agile'. We also checked the terms used in the literature. In 35 articles published in the last ten years, the highest number of authors (11) write about **agile methods** (of programming, of project management), while only five use the term agile project management. Otherwise, authors often discuss agile software development, an agile approach, agile projects and teams, and rarely agile processes, factors, standards and companies. We also found five books that discuss agile methods, although only Wysocki (2009) writes about agile project management, while others do not mention this term:

- Rothman (2007) mostly uses *agile project lifecycle*, and also mentions the agile project, approach, development, and agile practices and techniques;
- Forsberg et al. (2005) mostly talk about agile methods, processes and practices and agile development;
- Brandon (2006) refers to agile methods, processes, methods, and agile programming; while
- Thomsett (2002), whose book deals with radical/extreme project management, mentions agile development and process.

2. THE AGILE WAY OF IMPLEMENTING PROJECTS

2.1 Agility and the agile approach

The Oxford Dictionary (www.oxforddictionaries.com) defines *agile* as *the ability to think and understand quickly (or to move quickly and easily)*. In the introduction, we mentioned the argument of one of the authors of the Agile Manifesto that an agile approach is based on recognising and applying feedback. After considering a variety of sources, we found that:

- the term and the concept derive from agile methods for developing information systems (the first emerged in the 1980s) and are mainly used for IT projects;
- methods emphasise the concurrent execution of the traditionally successive tasks of a project (a 'fountain' instead of a 'waterfall'), and the constant coordination of participants; in this sense, agile methods are comparable to concurrent engineering, which also emerged in the 1980s;
- the essence of the approach involves constant updating of the execution, and detailed planning of smaller cycles (iterations) to implement a project according to the current results, lessons learned, new ideas etc.; and
- the focus on the user is very important and the project team, therefore, usually involves a representative of end users who regularly checks the partial results of the project (to ensure greater suitability of the final product with regard to the wishes and demands of the users).

The method's essence therefore lies in the fact that a project's objectives (project scope, configuration, and the deadline) are defined in less detail at the start of the project, and a project execution schedule is also prepared roughly – the project is divided into equal iterations with assigned parts of the project scope to be created. In the beginning, a team undertakes the most important functions, while leaving the least important ones for the end. Less important demands can later be omitted on the basis of the results of already finished iterations, the client's changed wishes/requests, the performers' proposals, or the changes in the environment. A detailed specification of the iterations' products and

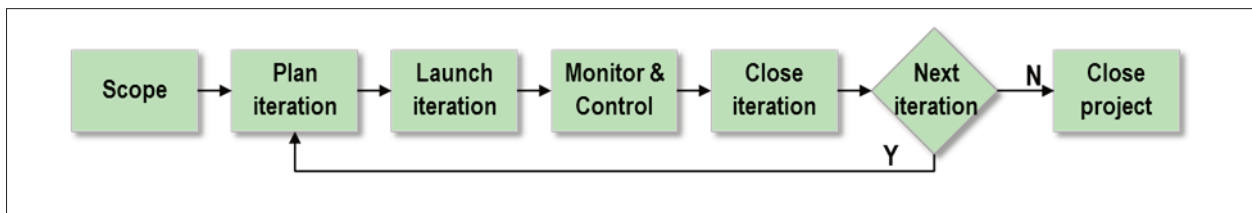
precise scheduling of the iterations (the way of implementing, tasks, hours of work, performers etc.) is created at the beginning of each iteration, taking into account the current results, new insights, the client's new wishes or the ideas of developers as well as changes to original assumptions and requirements. The execution plan for the iteration is made by the project team (not the project manager).

Gibbs (2006) mentions that the pioneers of the agile approach and iterative software development followed the example of Toyota's lean production, and their purpose was to eliminate much of the overhead and 'ceremony' common to waterfall-based lifecycles. Development iterations lasted two to four weeks then, and nowadays their duration has not changed – Rothman (2007) states that they last from one to four weeks. The testing of intermediate results is so prompt (and not at the end, as with traditional software development) and, after each iteration, the team also obtains feedback on client satisfaction. Wysocki (2009) adds that the iteration can be repeated if the client is not satisfied with the results (Fig. 1).

ing to Thomsett (2002), the latter is more flexible and is based on the dynamic requirements, shorter development cycles, virtual teams, changing technologies and joint participation of all stakeholders of the project. He emphasises that the relation between a client (user) and the contractor (project team) is based on a partnership! Wysocki (2009) indicates that the differences in the approaches result from the level of familiarity with the solution at the beginning of the project. The main differences are the detail of planning, the greater role of risk management, and more collaboration with the client. The author defines the use of individual approaches as follows:

- traditional – solution and requirements are clearly defined, large changes in scope are not expected, the projects are routine and repeatable, using a proven template;
- agile – the solution and requirements are partially known, there is a possibility of additional features that the team does not know of yet, a wide range of changes in the project's scope are expected (usually development or organisational projects); and

Figure 1: Iterative PMLC model, Wysocki (2009)



(PMLC = Project Management Life Cycle)

It is important to recognise that the agile approach first of all focuses on the execution phase of a project and does not define the whole project life cycle, which in principle remains the same (initiation, planning, execution and closing), except that the last part of the initiation (definition of specifications) and part of the planning are moved to the execution phase. The approach primarily affects the accuracy of planning – it is certainly necessary to define a rough schedule for entire project at the beginning of the project, while individual iterations are planned in more detail in the execution phase (e.g. tactics, tasks, hours of work, performers etc.).

Extreme project management is an upgrade of agile one (offering a higher level of agility). Accord-

- extreme – the objectives and requirements are unclear (usually research and development projects).

2.2 The Manifesto and The Principles of Agile Software Development

Published in 2001, the Agile Software Development Manifesto (www.agilemanifesto.org) was written by 17 authors representing different agile development methods (e.g. Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming). On the aforementioned website, the following message from the authors is found: 'We are uncovering better ways of developing software

by doing it and helping others do it.' In doing so, the authors stress that individuals and interactions are more important to "agile software developers" than processes and tools; that the working software has the advantage of comprehensive documentation, that customer collaboration is more beneficial than contract negotiation, and that responding to change is better than following a plan. The authors call these Agile Software Development Values (Figure 2), while adding: 'That is, while there is value in the items on the right, we value the items on the left more'.

Figure 2: Agile Software Development Values

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

(www.agilemanifesto.org)

Although the ideas of agile manifesto are acceptable in principle, we believe they should not be the guiding principle (basic guidance) of project work. Namely, without agreed processes work can be chaotic. The results of the work and agreements must be at least briefly documented, and end users should participate (however, they should not invent new requirements daily). A project needs a baseline schedule, but it should be regularly adapted depending on the circumstances, ideas, additional requirements etc. In this regard, we believe that controlled mastery over changes to the scope of the project is very important.

On the other hand, we can say that those values are neither necessarily new nor the complete opposite of the values of the traditional approach. However, since the indicated values say little about how the authors imagine the agile approach, and for a clearer understanding of the manifesto, they added the Twelve Principles of Agile Software Development (Figure 3).

Let us look at the principles in greater detail and discuss them (the cited principles are in italics):

1. *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.* In any case, we agree that it is very important to satisfy the client, but not on

Figure 3: Principles behind the Agile Manifesto

1. highest priority: **satisfy the customer**
2. even **late change of requirements** is welcomed
3. Frequent **delivery of working software**
4. daily **work together**
5. motivated individuals is given environment and support they need, and trust them to get the job done
6. conveying information: **face-to-face conversation**
7. primary measure of progress: working software
8. agile processes promote **sustainable development**, stakeholders should be able to maintain a constant pace **indefinitely**
9. continuous attention to technical excellence and good design enhances agility.
10. simplicity - the art of maximising the amount of work not done - is essential
11. **self-organising teams** → best architectures, requirements, and designs
12. team regularly reflects on how to become more effective

(adapted from www.agilemanifesto.org)

account of own income so we need to point out the need for a proper contractual relationship between the client and the team. However, it is the second part of the principle that certainly convinces us, and it can be regarded as the biggest advantage of the agile approach: it is already possible to use some part of the developed software at the end of the first iteration, and each successive iteration brings new software functionality and thereby new value. This leads us to the conclusion that more iterations bring more benefits to the client, which encourages them to prolong the project. Unfortunately, this is only possible in certain types of IT projects, while it is even less possible for other types of projects.

2. *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.* Changing software is generally much cheaper than changing the product or a building, although in certain types of IT projects a small change can cause a large-scale correction of an already developed code, as well as additional errors in the functionality that are difficult to detect and remove. Otherwise, for all types of projects it is the case that each change can be approved to the extent that it brings more benefits than causes costs and, of course, when both costs and benefits are appropriately distributed between the client and contractor.
3. *Deliver working software frequently, from a couple of weeks to a couple of months, with a*

preference to the shorter timescale. Just like with the first principle, we also cannot deny this principle, which shows the advantage of the agile approach. The only question is whether this approach is useful in other types of projects.

4. *Business people and developers must work together daily throughout the project.* Business people? Unfortunately, we do not know whether the authors meant users or managers of the contractor. In theory, as far we understand the principle correctly, because of a desire to complete customer satisfaction and to achieve a high quality deliverable, the developers constantly check if they are on the right track. This seems reasonable (to some extent). On the other hand, when we delegate a task to someone we expect that they will not bother us every few hours wondering if they have understood the requirements correctly and if their idea or solution is sufficient. This way of working can be quite burdensome for the client, and the question is whether frequent coordination is really necessary, especially because an agile project can last for some time.
5. *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.* Anyone who is at least a little familiar with motivational theories and recommendations regarding the leadership of people and (project) teams will know that this principle was always important even in the 'traditional' approach. In any case, this principle is not disputed, yet it is not new!
6. *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.* Similar to the previous principle we need to point out the scientific recommendations regarding communications in the project, which emphasise the high value of interpersonal verbal communication. Theories of leadership, motivation and communication among co-workers started to develop in the 1960s, although it is possible that technically educated experts are not familiar with them so far. Otherwise, it is true that a verbal conversation (at least by phone or Skype, in the context of virtual teams), is the best way

to communicate, especially with regard to searching for ideas, coordinating work, resolving interpersonal conflicts, creating a good working atmosphere etc. However, we must point out that people may be bothered by too many meetings (as they often interrupt creative work) so, from that point of view, it is more efficient to use emails or the project's information system for sharing less important information.

7. *Working software is the primary measure of progress.* This principle is an upgrade of the first and the third ones, and is also connected with the main advantage of an agile approach (the possibility of using partial results), so the principle is not to be disputed. We also agree that control of the project's progress is easier, although the question is whether it is reasonable since such projects generally do not have an explicitly defined deadline; only the execution of a single iteration can be seriously controlled.
8. *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.* This principle is the most problematic in terms of a project-based approach. One of the basic characteristics of a project is the time limitation – if the endeavour has no pre-defined deadline then, by definition, it is not a project! We could say that every iteration is a small project (it has a clearly defined deadline, detailed specifications, schedule, performers, etc.), wherein the client and the project team have signed an agreement (intention) on long-term cooperation at the beginning of the project, and they agree that the costs and payments will be calculated for each iteration separately and that the project may be brought to a conclusion at the end of any iteration. And another dilemma: how can a company arrange a new project if the management does not know when they will finish the current projects and when the employees will be free to take on a new project?
9. *Continuous attention to technical excellence and good design enhances agility.* This sentence could be understood in both directions: that the technical excellence improves the agility (= flexibility, which is somehow difficult to under-

stand), or that agile thinking contributes to technical excellence. In any case, technical excellence also has underpinned the quality of work and deliverables in the 'traditional' approach. However, we must point out 'too high' quality! Searching for technical perfection can prolong a project and increase the costs. Some developers have a constant feeling that something can be done better, and they may get lost in a spiral of continuous improvement of their results! An effective team only provides the client with what it requires: anything supplemental could be a waste of time and money unless the client is willing to pay for higher quality.

10. *Simplicity – the art of maximizing the amount of work not done – is essential.* This principle may partly contradict the previous one but with reference to the study of other sources we assume that the authors meant simplicity in planning the iteration (how to achieve the objectives of the iteration as efficiently as possible). However, this principle also partly relates to simplifying the customer's requirements; the project team can propose a simpler solution to the client's need. Both aspects, of course, are not unknown to 'traditional' teams, but we definitely agree that it makes sense to point out the last two principles as often as possible to provide an adequate (efficient) way of thinking and working of all project stakeholders.
11. *The best architectures, requirements, and designs emerge from self-organising teams.* Can we say that the 'traditional' project team is not self-organised, no matter how complex the project is, and what level of team we are discussing? We suspect that here the authors are suggesting that teams do not need a manager as they are able to organise their own work; we assume that we know the background of this proposal: a precondition for excellence in management is also (or especially) the excellence of leadership, and a problem arises where the manager is relatively alienated from the team (makes his/her own plans regardless of the opinion of the team members, instead of coordinating he/she focuses on bureaucracy

etc.). However, this is a problem of individuals rather than the 'traditional' approach. On the other hand, the idea might not be so irrelevant if we refer to studies that have found that a team where the leader was not officially selected at the beginning eventually spontaneously asserts an 'informal' leader who takes over most of the manager's duties (leads planning meetings, coordinates the work, warns colleagues about delays etc.).

12. *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.* This principle, of course, cannot be disputed. It should provide valuable guidance to project teams, and we believe that teams often need to find the time to talk about their work and to discuss ideas for better and more effective approaches, also in the context of longer 'traditional' projects.

While reading the principles, we might consider that in the background of the principles of the Manifesto there is some sort of negotiation or bargaining along the lines of: if you give us (infinitely) more time (and you pay us), we can realise all your demands and desires. Could this represent some sort of 'connivance' for both the client and the contractor? In terms of: 'Come on, let's do something useful as quickly as possible, then we'll see what we'll do next' (this is, of course, written with considerable exaggeration). For the 'traditional' project manager, this could be a difficult conceptual leap. However, do we need a manager in such cases?

In addition, it is possible to feel the struggle between the developers (programmers) and the managers (a good example is the eleventh principle). We note here that up until 18 April 2013 when we checked the number of signatories at www.agilemanifesto.org, the agile manifesto had been supported by 13,705 individuals and organisations. We assume that the signatories are mainly professionals from the IT domain, yet it would also be interesting to know the structure of the supporters, i.e. how many were managers and how many programmers. In an interview with Bowles Jackson (2012), Hunt said: 'Agile probably has not impacted project management as much as it should have. Primarily, the agile movement has impacted development and developers more than management and managers.'

It is therefore possible that most of the supporters are developers who (under constant time pressure) are probably searching for a 'different truth' and desire to enforce their own rules.

3. ROTHMAN'S WEAKNESSES OF THE TRADITIONAL APPROACH

In order to show that the supporters of agile methods sometimes have quite a narrow viewpoint, and are probably less acquainted with the history and development of project management and related disciplines, we show and critically comment on Rothman's (2007) 'weaknesses of the traditional approach'. The author presents the listed weaknesses (the text in italics), while explaining the advantages of the SCRUM method:

- *The manager only allocates the tasks, and controls* – this presumably implies that a project manager independently plans the project (without the involvement of team members), but the assumption is not appropriate. In planning a project, at least members of the core team participate, and sometimes performers of other tasks if necessary, thereby ensuring a higher quality plan that is also more achievable, but it also affects their motivation (Stare, 2011). And control? Anyone who responsibly directs the execution of tasks must regularly check whether the work is being carried out according to the plan, and respond to any deviations as soon as possible in order to ensure the achievement of the objectives. Self-organised teams also control work within their daily morning standing meetings (*SCRUM*)!
- *Members of the project team work alone, without collaboration* – which is also untrue since project work is actually teamwork. In addition, a profession draws attention to the interdependence of the disciplines involved, and to the inadequacy/problems of the partial solutions. If nothing else, team members meet at the weekly progress meetings where, in addition to controlling the performance, a reconciliation of the ideas, solutions and tasks takes place.
- *Work is carried out strictly sequentially (the waterfall method), when someone finishes their*

task they hand the results over to the successor – once again this claim is 'outdated' because since the occurrence of concurrent engineering in the 1980s the profession has emphasised the need for the constant collaboration of all team members (which does not necessarily mean that someone works on the task, but only advises the task executors and/or participates in the development of solutions).

- *Pressure on the performers is pronounced only in the last few months, making them completely exhausted at the end* – this is usually a consequence of poor plans and an unorganised manager (team) and is not a weakness of the traditional approach. A quality plan that ensures a balanced workload, a matrix organisation in which line managers delegate additional work to employees when they do not perform project tasks, effective plan enforcement, an expedient mastering of changes, and regular control all refute this claim.
- *A waste of time (and increased costs) when someone waits for predecessors to complete their task* – this only applies in the case of pure project organisation where team members work full-time on one project, while in the case of a matrix organisation that dominates nowadays the line and project manager together ensure a steady workload of the employees according to relevant plans and weekly work/tasks coordination.

4. ADVANTAGES OF THE APPROACH FROM THE CLIENT'S PERSPECTIVE

Before we became more familiar with the agile approach, our biggest dilemma was whether it is really more beneficial than the traditional one. Our experience makes it very difficult to believe that agile projects are shorter and cheaper, as their supporters claim. Could this be achieved by reducing the scope of work (Gibbs, 2006) – so that the project team deletes less important/unimportant user requirements early on in the project (cycle) or leaves them for later cycles (when the client will ultimately find them unnecessary)? Can higher efficiency be hidden in the fact that self-organised teams are able to find more efficient ways to

achieve the objectives (of the project or of the iterations)? Or is the planning more exact for a shorter time frame?

Although we believe that the final product of an agile project can be higher in quality and better meet the client’s demands and hidden wishes, several factors (characteristics and principles of the agile approach) indicate a less efficient execution – the extension of deadlines and increased costs. And as we have already mentioned, the claim that the product is constantly evolving and upgraded brings us to the conclusion that this endeavour, by definition, is not a project at all! Above all, it would be interesting to know how frequently the already developed and operating/used parts are corrected and amended in the later iterations.

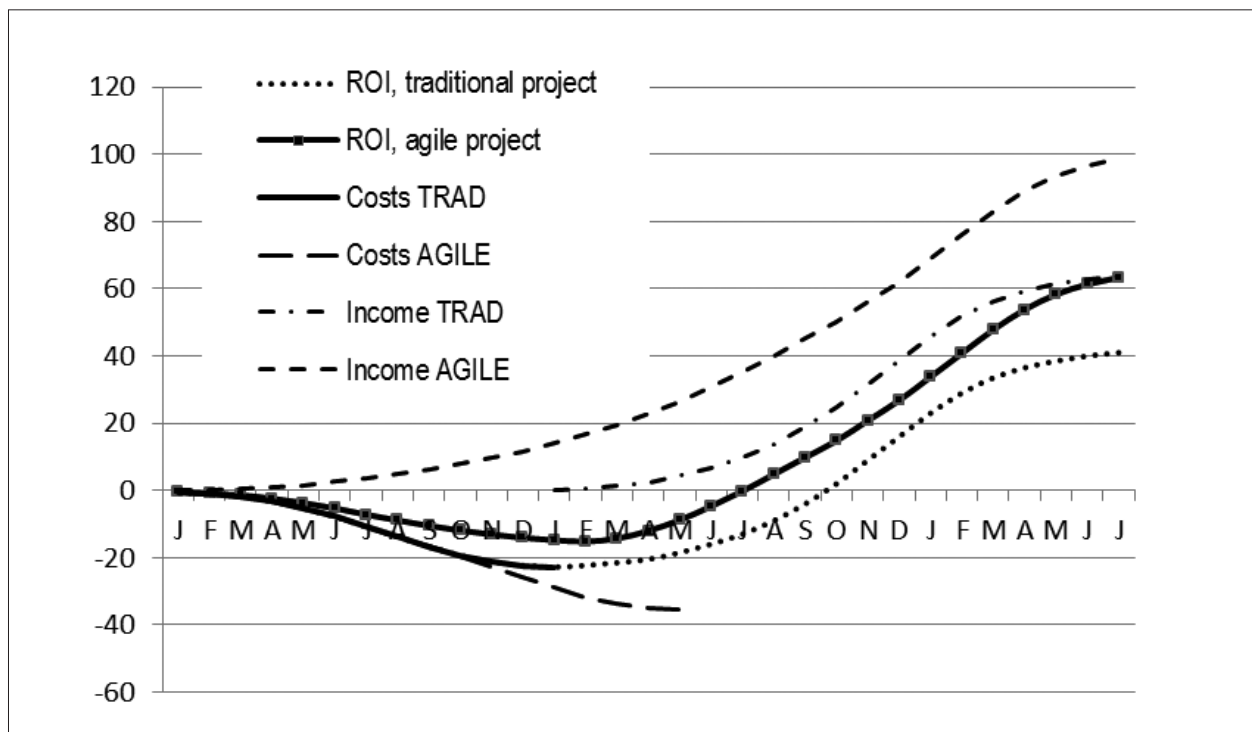
Why then should a client choose a project team which offers agile software development? We believe two prominent characteristics/principles favour of such a decision:

- the constant possibility of changing the products and adding new requirements; and in particular
- intermediate deliverables that can already be used.

The start (and time) of exploiting the deliverables strongly weighs in favour of the agile approach. Early exploitation ensures the early start of the return of the invested money (so we can invest the earned money in additional iterations in new projects). If the first project deliverables bring benefits already after one month (and every new iteration delivers additional value and increases monthly income/savings), the client will probably not object to an extension of the project and a cost increase (see the simulation in Figure 4). The advantage of this approach is therefore mainly on the revenue side and not in terms of effective execution!

However, the possibility of using the partial deliverables is very limited. In addition, it is only possible for IT projects and, moreover, it only applies to a few types of IT projects. Even if the partial products can be made use of in the development of websites or business information systems, this does not apply to development software programs (to be sold in large quantities) or to the development of software for new products or for the automation of production processes. The latter is mainly because only the finalised product/system can be sold, and the software can only operate in conjunction with

Figure 4: Comparison of the costs and benefits of traditional and agile projects



developed electro-mechanical parts – everything is related, interdependent and needs to be developed concurrently.

However, there is one other dilemma associated with the development of business information systems: partial solutions can only be used if the information support is being developed for a process that has not yet been computerised. Namely, if users already use an old IT solution – will they partly use the old one and partly the new one? This can be quite inconvenient ... perhaps it would be better to wait for an entire new solution to be finalised and to then completely switch over to it?

In principle, there is truth in the statement by Highsmith, one of the authors of the manifesto (Bowles in Jackson, 2012): ‘Agile is changing the way organizations measure success, moving from the traditional iron triangle of scope, schedule and cost to an agile triangle of value, quality and constraints.’ However, we must point out that traditional authors have developed several different triangles (<http://projektni-management.si/2010/12/12/projektni-management-namen-narava-cilji/>), that quality is part of the traditional triangles, and that the constraints include time and money! The main novelty the agile view brings could be its taking of the value of the project into account, which has traditionally been dealt by the client (the project team was only responsible for effectively carrying out the project). The agile approach is expected to lead to the greater commitment of the project team to develop the most useful solutions so that the team should constantly think about better solutions that would bring maximum benefits to the client (as opposed to rapidly developing something less useful). In doing so, Rothman (2007) points out that the modern project manager should upgrade their organisational skills with technical and value management skills.

5. EVALUATING THE UNIVERSALITY OF THE APPROACH

Table 1 presents a subjective evaluation of the potential use of agile methods (and an agile project life cycle) in the context of typical project types. We evaluated the applicability on the basis of the approach’s characteristics, the principles of the agile

manifesto, and the perceived benefits of agile methods. We emphasise that this is a subjective assessment (we do not claim that the arguments are beyond critique). Above all, the purpose of the table is to encourage further research and professional discussion!

As can be seen in the table, the agile approach can most easily find a place in research projects in which a team can carry out tests, experiments etc. by iterations; with each new iteration being defined on the basis (findings/results) of the previous one. Irrespective of the type of research (natural studies/sociology), partial results may be used for publication in scientific articles, while in technical fields the iteration deliverables can be used for other (current or later) projects, such as product development or civil engineering. And research projects are the only type listed in the table that in principle can take an ‘indefinite’ period of time (e.g. till the discovery and validation of a new health treatment method).

Based on the possibility of using partial results, which we previously identified as a significant advantage of agile methods, it could be reasonable to use the approach in the development of services and business process reengineering since both satisfy other specific requirements for an agile execution. Although it could be exceptionally beneficial to explore the partial results of building construction projects (see the notes below the table), the other factors clearly indicate that the agile approach would not bring significant benefits, especially due to the high cost of changes in construction. It would also be difficult to think about agility in the organisation of events, although this area probably calls for further consideration as there are significant differences among types of events (conferences, sport and cultural events; on the regional, national or international level).

Product development projects and business expansion projects meet the criteria for being implemented in an agile way, yet the partial results cannot be used: a product cannot be sold until it is finally developed, the new enterprise will start to return the investment only after the start of operations (being put in service). However, we believe there is a possibility to use some agile techniques in the execution of these projects.

Table 1: Evaluation of the possibility of the agile execution of typical types of projects

	(Basic) research studies	Applied development of products/services	Projecting (construction, engineering)	Building construction (without projecting)	Reengineering, reorganisation, processes optimisation	Establishing companies/takeovers; other business unification	Organisation of events
Relatively unknown objective/end product	√	√ (1, 2)	√ (2)	x	√	√	x
Rough demands at the beginning / exact specifications later	√	√	√	x	√	√	√
Expected changes							
• due to activities of the competition (e.g. a new product)	(√)	√	(√)	x	x	√	x
• client proposals	√	√	√	(√)	√	√	√
• the emergence of new technologies/materials	√	√	√	(√)	√ (3)	(√)	x
• new market demands (unknown client)	x	√	(√)	x	x	√	x
• new ideas/solutions of team members	√	√	√	x	√	√	√
Consequences of the changes in the later stages (compared to the total project costs)	low	high	low	high	middle	middle	middle
Possibility to eliminate irrelevant ideas/requirements of the client	(√)	√	√	x	√	√	√
The most important functions are created at the beginning (less important ones can be eliminated)	x	(√)	x	x	(√)	√	x
Rough schedule at the start / micro planning for each iteration	√	√	√	x	√	√	√
Planning a new iteration based on the results of the previous one	√	√ (4)	√ (4)	x	√ (5)	√	x
Iteration execution (equal or different iteration durations)	√ equal	(√) different	√ different	x	√ equal	√ different	(√) different
Necessary flexibility/adaptation to the situation	√	x	x	x	√	√	x
Partial results can be used/exploited	√	(√) (6)	(√) (7)	(√) (8)	√	x	x

√ – YES, possible (√) – possible, but rarely x – not possible

Additional notes concerning the table:

- (1) depending on whether the team develops a retail product or semi-product for a specific client
- (2) depending on the desires/requirements of the client (the extent to which the client invented a product itself)
- (3) if the project also includes the development/modification of IT support
- (4) development plan (of prototype, final drawings) based on the conceptual design (product, building)
- (5) a plan of the introduction of the reengineered process (organisation) on the basis of the designed process (organisation)
- (6) applies to services
- (7) conditionally, for example: on the basis of the electricity line designs the investor can start looking for a contractor
- (8) conditionally, for example you can live in a house that does not have insulation and roughcast, and where only the ground floor has been completed

6. CONCLUSION

When reading the literature by supporters of the agile approach, the agile manifesto and the principles of software development, experienced project managers are somewhat surprised by the criticism of traditional project management – as if the authors had never worked in real teams, as if managers never planned projects together with team members (and vice versa – as if team members never collaborate with managers on project planning), as if managers were not team leaders but estranged administrators, and as if teams have never regularly collaborated with the client or end users. Strange arguments, however, the research findings that only one-quarter of IT projects have been successfully implemented are now more understandable!

On the other hand, it is difficult to believe that the agile approach provides for the efficient execution of a project. We agree that it provides higher quality results and that the final products meet the desires (and not only the requirements) of the client significantly better, but we believe that projects are more expensive and last longer than they would if executed in the traditional way. However, it is true, as regards IT projects in particular, that an agile project always brings added value to the client, as opposed to projects implemented in the traditional way, which have often been terminated before the end, or where the end product has been unusable. Both, of course, mean that the project was a waste of money. We therefore see a key advantage of agile methods on the revenue side, especially since the partial results can already be put to use.

Authors of the Agile Manifesto and the other proponents of agile methods like to predict that the agile way of implementing projects will in the future replace the traditional approach, but their predictions have yet to come true. A comprehensive agile approach has proven to work only for certain types of software development and, according to our evaluation, it could find a place in research projects, the development of services and business process reengineering projects, but certainly not in construction and other engineering projects (excluding the projecting phase).

Of course, we are unable to claim that certain best practices of the agile approach cannot be utilised for projects that still adhere to the traditional approach. When it is prudent to apply particular practices and in which ways are issues that will be revealed by future research studies.

REFERENCES

- Berger, H., & Beynon-Davies, P. (2009). The utility of rapid application development in large-scale, complex projects. *Information Systems Journal*. 19 (6), 549–570.
- Bowles Jackson, M. (2012). Agile: a decade in. *PM Network*. 26 (4), 58-62.
- Brandon, D. (2006). *Project management for modern information systems*. Hershey: IRM Press.
- Doherty, I. (2010). Agile project management for e-learning developments. *Journal of Distance Education*. 24 (1), 91-106.
- Doherty, M. J. (2011). *Using Organizational, Coordination, and Contingency Theories to Examine Project Manager Insights on Agile and Traditional Success Factors for Information Technology Projects*. PhD dissertation. Walden University
- Forsberg, K., Mooz, H., & Cotterman, H. (2005). *Visualizing Project Management*. Hoboken: John Wiley & Sons.
- Gibbs, R. D. (2006). *Outsourcing and the IBM rational unified process*. Upper Saddle River: IBM Press.
- Gonzalez, W. (2011). *Agile Project Management and the Creation of Intellectual Property*. PhD dissertation. University of Maryland University College.
- Manifesto for Agile Software Development (2001). www.agilemanifesto.org.
- Oxford Dictionary. www.oxforddictionaries.com
- Rothman, J. (2007). *Manage it*. Dallas: The Pragmatic Bookshelf.
- Stare, A. (2010). Projektni blog, www.projektni-management.si.
- Stare, A. (2011). *Projektni management: teorija in praksa*. Ljubljana: Agencija Poti.
- Thomsett, R. (2002). *Radical project management*. Upper Saddle River (NJ): Prentice Hall PTR.
- Wysocki, R. K. (2009). *Effective project management: traditional, agile, extreme*. (5th ed.) Indianapolis: Wiley Publishing.

