SOME ASPECTS OF AGILE APPROACH IN PRAXIS

Jaroslava Kniežová[1]

Abstract: In modern times, competitiveness in the market depends on having a good information system. The companies developing and supplying information systems are in competition too, and having an effective system of delivery is critical for obtaining lucrative offers. Therefore, the software development companies continuously try to improve their development process to supply the product in a short time and with high quality. The agile approach potentially shortens this time and is very often used. This approach has almost replaced the traditional process. More and more companies implement agile approach in these times to be competitive in the software development market and hasten product delivering.

The traditional and agile approaches differ in certain perspectives. Hence, the question arises as to whether the agile approach is the best for the software development company in every case. This article contains a comparison of these two approaches, as well as a case study relating to the agile approach in a real software development company, which had previously used the traditional approach. The article also describes situation where replacing the traditional approach with agile would improve results.

JEL Classification Numbers: M15, **DOI:** http://dx.doi.org/10.12955/cbup.v3.623

Keywords: agile approach, traditional approach, scrum, team, software development, planning, work management

Introduction

Individuals and businesses, alike, rely on information systems (IS) daily. An effective information system is essential for businesses to remain viable. Hence, businesses want to invest in an IS that is reliable, relevant and regularly updated. This results in a competitive software development market, where information technology (IT) companies vie to supply the needed functionality in the shortest time with the greatest effect. Developing software means managing IT, teams of IT experts, and financial and time resources, to reach maximum profitability. Winning companies are those that deliver the required software within the shortest timeframe. Hence, all companies continue to investigate how to expedite the process that starts with gathering requirements and completes with product delivery. Newer approaches to the software development continually arise to reach the shortest timeframe with the lowest cost. Recently, the agile approach, which differs to the traditional process, has been defined and widely applied in the software development market. The phrase “be agile – be successful” is very popular, and this approach is often associated with company success. Even so, a reasonable question to ask is whether the agile approach guarantees success in the software development business.

Traditional approach

This approach firstly focuses on gathering requirements at the start of the software development process. For this, the IT companies hold discussions with the customer to identify the project needs. From this, finance and time allocation is defined. The whole process is planned and controlled, and where needed modified. The completion of this stage usually requires the customer’s approval.

Next, the approach involves modeling the proposed IS. This is a software engineering principle, as defined by Russev (2002), where the software analysts use the captured details in the IS modeling. UML (Unified Modelling Language) is used for solution visualization in models (Arlow, Neustadt, 2005). These models are the basis for later programming. Using MDA (Model Driven Architecture) in traditional approach gives the basic for programming in models created before (Tavač & Tavač, 2013). Kruchten (2004) reports that according to the Rational Unified Process RUP methodology, the traditional approach often uses the Analysis and Design phase (Agile Modeling and the Rational

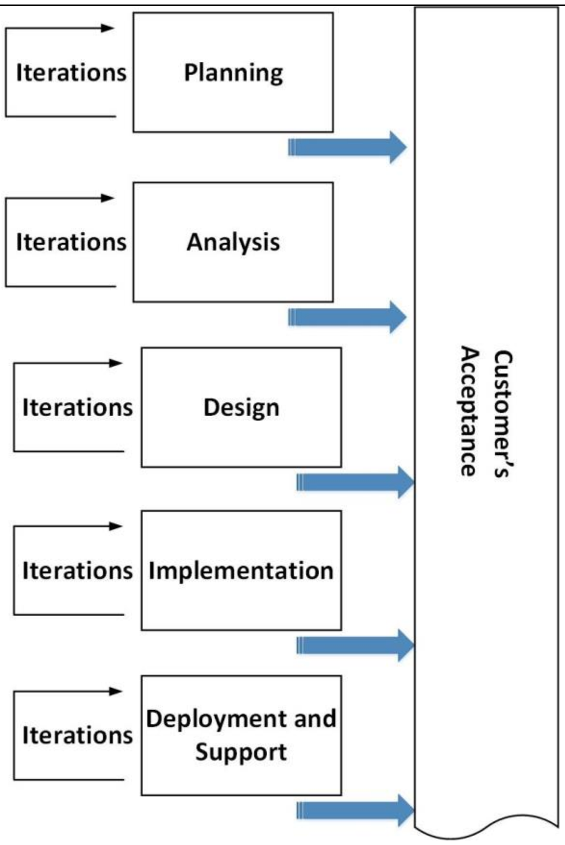
[1] Jaroslava Kniežová, Comenius University, Faculty of Management, Bratislava, Slovakia, jaroslava.kniezova@fm.uniba.sk

Unified Process, n.d.) (this phase contains the works of the analysts on the models), which is important, because it largely influences the time and financial expenditure later in the process. Similar to the starting phase, the company discusses these models with the customer to obtain approval, which is often written, before proceeding.

The models are an important part of this approach. The design patterns are used in modeling (Malcher, 2013) and needed algorithms defined (Meško, Kršák, & Hrkút, 2013) so as to keep the quality of the solution. The developers draw the IS solution from the models, and this ensures the code is developed to requirements. Kruchten (2004) notes that, according to the RUP methodology, this programming phase should be short and not financial arduous if the previous phase was done precisely and the models contain the needed information in a suitable form. The building and testing of the IS follows this phase. The completion of these phases also requires the customer's approval. Once the customer accepts the tests and results, the system can be implemented and deployed.

The "life cycle" of the traditional method guides the processes to develop the new IS. This life cycle involves iterative phases that start with identifying the requirements for the software development team. The points requiring customer approval represent the approval cycle. These two iterative phases are interdependent. Completing each phase (including iteration of the IS modeling) usually requires the customer's approval. Figure 1 indicates this dependency. The approval phase allows for quality control, with the particular customer's approvals increasing the probability of their satisfaction with the final product (when the IS is deployed).

Figure 1: Traditional methodologies and software development process



Source: Author

Additionally, the approval cycle is advantageous for managers of the software development team because it reduces the likelihood of the customer continually increasing their requirements, and endangering the cost and time allocations. Therefore, the approval cycle is considered beneficial.

Accordingly, the RUP methodology refers to the approval points as milestones (Kruchten, 2004). It incorporates these milestones with team self-evaluation (Kruchten, 2004). This is beneficial because it consolidates quality control with both the customer and the development team evaluating the output to find pitfalls and potential risks.

It is noted, that in the traditional approach, most communication between development team and the customer occurs during the modeling phase. The aim is to fully capture the needs and ideas of the customer. The subsequent programing phase is generally shorter with communications between the IT company and the customer relating to minor corrections and details.

Agile approach

This approach supposedly began in 2001 when a 17-member group met to find an alternative to the traditional approach that was based on documentation and IS modeling (Beedle et al., 2001). Fowler and Highsmith (2001) note that the document “Agile Manifesto”, was created with 12 principles of agile software development. The agile approach, as opposed to the traditional procedure, eliminates the modeling phase in the process. Instead, it focuses on installing the software for the customer, as soon as possible, irrespective of quality checks. It advocates further reworking of the product, rather than postponing delivery for the sake of capturing the customer’s entire requirements. Hence, the modeling phase in agile methodologies is much shorter, with the coding phase much longer than in the traditional approach. The communication with the customer is very intensive because of the need to hasten deployment. Essentially, the customer has become a team member in agile methodologies. The typical and very often used methodology in agile approach is “scrum”. According to this methodology, the start of the developing process defines the particular tasks (referred to as user stories), and for each task, the team plans the time and the needed resources. Scrum defines the product owner (one of the team members) and scrum master (ideally another team member). The product owner decides the time schedule, task allocation, and priorities for each task. The scrum master is responsible for solving problems that eventually arise, and overseeing the development progress. All phases are divided into so-called “sprints”, which are usually two-week time periods. Any team member can suggest a new task, but only the product owner decides if and when (in which sprint) the task will be allocated. Tasks already assigned to a sprint are in “sprint backlog” and all the unassigned tasks are in the “product backlog”. A review occurs after each sprint. This involves evaluating prior work and the team, usually through open discussions.

Approaches comparison

A comparative assessment of the two widely used approaches was undertaken to identify the possible advantages and disadvantages of each. The comparison involved the traditional and the agile approaches. A case study involving the agile approach in a real software company was also examined.

The traditional approach represents a modern methodology for software developing that involves software modeling as a tool to ensure the quality of the final product. It is noted that this approach (traditional) is not the “waterfall” methodology. It is a modern methodology, which contains iterations that enable the optimization and supplementing of details in the modeling process for later coding.

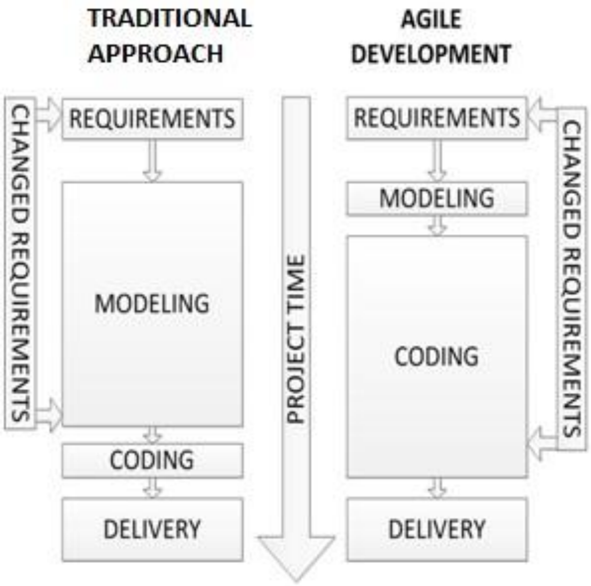
The first step in comparing these two approaches was to describe each, followed by a comparison and then a synthesis of the results. Next, the case study relating to the agile approach in a real software development company was considered. This examined the influence of newly implemented agile principles on the team, the customer involvement, and the project itself.

Results comparison

Life cycle – Types of processes

The life cycles of the two approaches primarily differed in the amount of modeling and programing undertaken (Figure 2). While the traditional life cycle allowed more resources for modeling, the agile approach diminished the modelling phase. The emphasis on work, which involved iterations, in the analysis and design phase of the traditional approach contrasted to greater emphasis on the coding phase in the agile approach (Figure 2).

Figure 2: Comparison of approaches based on the amount of work in particular phases



Source: Author

Other phases (requirements, delivery, and change management) were similar between the approaches, especially the phases of requirements and change management. However, the traditional approach dealt with changes during modeling, while agile dealt with changes in the coding phase. This difference has relevance for project financing (and human resource management).

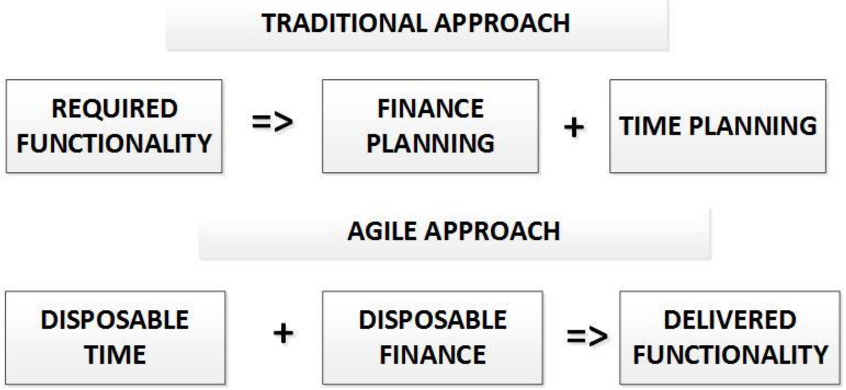
Quality control

Other criteria assure the desired quality of the final product. The main criterion is communication with the customer, for example to gain their approval. The traditional approach places the approval cycle within the main life cycle, compared to agile, which promotes intensive and flexible communication throughout. Nonetheless, in agile this is needed, as there is much less documentation in agile compared to the traditional approach, including that concerning the customer. In this respect, the agile approach is much more flexible regarding communication, but lacks the stability of the more structured traditional approach.

Project management

The process of planning and allocating resources is the third criterion for comparison. The traditional approach involves the first phase of requirements to identify the required functionality for planning resources. In contrast, agile defines the delivered functionality based on the disposable finance and time (Figure 3).

Figure 3: Comparison of approaches based on resources planning



Source: Author

Hence, the agile approach allows greater scope for competitiveness on the market, because a company can attempt to satisfy the customer’s demand, irrespective of timeframes and finance conditions.

Overall, the approaches largely differ in the planning process. In the traditional approach, managers identify time (and finance) costs for particular tasks and assign tasks to individual team members. For this, the managers need to know the skills and ability of team members to allow them to schedule and assign tasks to optimize completion. The more complex tasks can be assigned to the more experienced team members, for example. In agile, the tasks are assigned to the active sprint, and the individual team members choose their tasks from the sprint. For each task, the team member choosing the task estimates the time to complete the task. This can lead to a situation where (especially with less experienced) team members estimate the time incorrectly, resulting in project delays. Additionally, a natural human rational is to choose simple tasks first. If the more experienced team member or specialist decides to take the simple tasks, then the more complex ones are left for the beginners or less skilled members. Therefore, the agile approach is more flexible, but introduces a high risk to projects from problems arising from misguided estimates or choices that cause unnecessary delays.

Human resources management

The sprint log was described in a previous section as an agile tool for assigning tasks. Such assignment differs between approaches. A skilled manager fairly evaluates team members based on their work. In the traditional approach, a successful and motivating environment for team members depends on such a skilled manager. In the agile approach, the team evaluates itself (retrospective) with individuals choosing tasks from the sprint log. Therefore a successful work environment in this case depends, not only on the skills of individuals, but also on the team dynamics, which are often affected by the relationships within the team, differences in popularity, influence of outsiders, and other interactions. Hence, the team self-evaluation process may not always be fair, given the possible interplay in terms of human relations and individual popularities, which may deter effective team motivation and thus, a successful job outcome.

Comparison summary

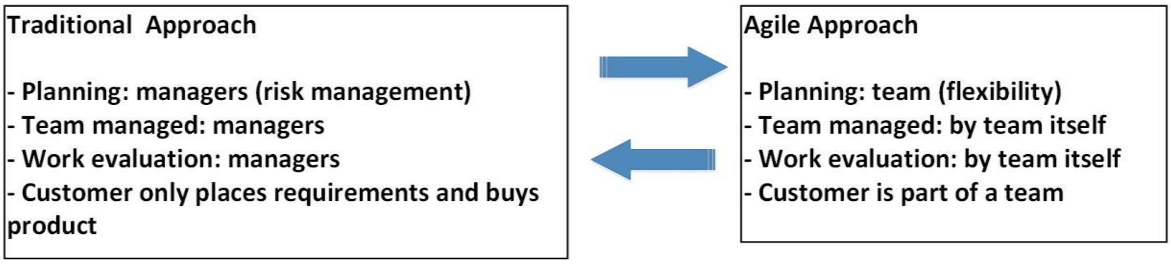
The main points summarizing the differences between approaches are as follows:

- Most important phase of the life cycle
  - Traditional approach – Analysis and design, changes mostly affect modeling
  - Agile approach – Coding, changes mostly affect coding

- Quality control
  - Traditional approach – documentation based on communication with the customer
  - Agile approach – communication by spoken word throughout
- Project and resources planning
  - Traditional approach – requirements driven
  - Agile approach – available resources driven
- Human resources and task management
  - Traditional approach – done by the managers
  - Agile approach – done by the team itself

The summary of differences between approaches is shown in Figure 4.

Figure 4: Comparison of approaches – brief summary



Traditional Approach

- Planning: managers (risk management)
- Team managed: managers
- Work evaluation: managers
- Customer only places requirements and buys product

Agile Approach

- Planning: team (flexibility)
- Team managed: by team itself
- Work evaluation: by team itself
- Customer is part of a team

Source: Author

Implementing the Agile Approach – Case Study

The case study involved a real software development company, and a team of 18 individuals. The scrum methodology was used in the processes, followed by the agile principles. Tasks were defined and assigned for the product, and later entered into the sprint backlog. Individuals chose particular tasks and estimated costs and timeframes. The following are the findings in terms of the observed behavior of the team members:

1. Motivation. Almost all team members had become highly motivated as they could choose ‘their own’ task from the sprint log. This outcome was not ongoing, however. After choosing tasks for a long time, the member’s motivation tended to decrease.
2. Calculation in task choosing. The majority of the team members chose simple tasks rather than more complex ones, even if they were able to solve the more complex ones. This aspect was usually ongoing. After having to choose their task over a long period, the member’s tendency generally remained or even increased.
3. Cooperation. Some of the team members (usually friendlier team members) cooperated more intensively than before.
4. Human relations in evaluation. Evaluation was almost always driven by the relations. Members that were more popular were considered to do a better job than those less popular. One interesting fact was that the less friendly relationships worsened, while the more friendly ones strengthened.

The following are the findings relating to project planning (time aspects):

1. Time estimations. Some team members repeatedly gave poor estimations (interestingly, the same team members), and this affected the time planning, which had to be repeatedly re-evaluated.
2. Change management. The changes were divided into two groups – serious and not so serious. The serious changes were covered by a new agreement, or at least by an change agreement, which was valuable. However, determining which change was serious or not (by the team members) was problematic and led to regular finance re-evaluations.

Conclusion

The above reveals that the agile approach is more flexible than the traditional method. A team following the agile methodology can quickly react on any needed changes. The communication inside this team is intense, similar to the communication with the customer. There is no modeling of the analysis information, and there are time savings where there is no need for documentation. Hence, the agile approach has the ability to rapidly shorten project time. However, the success of the agile approach is greatly influenced by the dynamics of the agile team, affected by individual preferences, relationships within the team, and the overall team experience. Decisions on whether to document analysis information may also affect outcomes. These elements can cause the project to fail in certain situations. Based on the identified differences and observed situations, there are some situations and project aspects where using the agile approach is suitable, while in others the traditional ways would suit better.

Using the agile approach may be the ideal in these cases:

- The scope of the project is small, the information system is known to the team. The team members can readily recall the project proposal and relevant information, as these are not documented. Suited to small or maintenance projects.
- The team is well coordinated, friendly, experienced, with no or few conflicts. Sound decisions and reviews are carried out by specific team members, within a well-functioning team.
- The customer is time-disposable, communicative. Communications with the customer are typically intensive.

Conversely, the traditional approach is more suitable in cases of large projects, without easily recognizable conditions, which pertain to new information systems. In addition, this approach would be better in cases of loosely coupled teams, or for example, where members are novices or have limited skills. Finally, the traditional approach is more suitable for the customer that is not freely communicative or available geographically, especially if in a different time zone.

References

Agile Modeling and the Rational Unified Process (RUP) (n.d.). Retrieved from: http://www.agilemodeling.com/essays/agileModelingRUP.htm.

Arlow, J., & Neustadt, I.. (2005). *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*. (2nd ed.). Upper Saddle River: Addison-Wesley.

Beedle, M., Bennekum, A., Cockburn, A., Cunningham, A., Fowler, M., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, C. R., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. Retrieved from: http://agilemanifesto.org.

Fowler, M., & Highsmith, J. (2001). *The Agile Manifesto*. Retrieved from: http://www.pmp-projects.org/Agile-Manifesto.pdf.

Kruchten, P. (2004). *The Rational Unified Process: An Introduction*. Boston, USA: Addison-Wesley.

Malcher, V. (2013). Decorator pattern in web application. *International journal of advanced information technology, 3*(4), 13-18.

Meško, M., Kršák, E., & Hrkút, P. (2013). The recursive segment 3D reconstruction algorithm. CICSyN2013, Fifth International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, Spain, 261 – 264.

Russev, S., Závodný, P., Grell, M., Polášk, I.: (2002). *Metódy vývoja informačných systémov* [*Methods of Information Systems Developing*]. Bratislava, Slovakia: Ekonóm,

Tavač, M., & Tavač, V. (2013, June 5 – 7). The General Algorithm for the Design of the MDA Transformation Models, CICSyN2013, Fifth International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, Spain, 171-179.