Proceedings of the 35th International Conference on
Information Technologies, InfoTech 2021, 2021. p. 80-86
https://stepanovd.com/science/article/115-2021-6-agile

USING AGILE METHODOLOGY IN ERP-SYSTEM IMPLEMENTATION PROJECTS

Dmitry Yu. Stepanov
RTU MIREA, department of Corporate Information Systems, PhD, professor assistant MIREA

Abstract: *Agile is one of the most popular methodology used for implementation software systems. Agile based methods like Scrum, Kanban or eXtreme Programming are applicable for small and big developments. Corporate information system is a set of different applications integrated together and used all over the company. Example of such system is ERP-system automating logistic, finance and human resources of enterprise. ERP-system implementation project takes a few years. The paper considers if Agile methodology can be used in ERP-projects.*

Keywords:*. automate business, Agile methodology, arrange team, shape product, sprint backlog, transform project team, multi steps testing, preparation design, handover documents, rollout projects, reduce project costs, cascade strategy, incremental strategy, spiral strategy, SQL-based programming language, face-to-face demonstration, multi-pass model.*

I. INTRODUCTION

Today technology plays an important part in our life. Most of private and public companies spend a lot of money to automate business as usual activities. Using information technologies and systems is not advantage but normal practice to be competitive. Full digitalization of enterprise or automatization of already computerized transactions is next step in technology evolution. Information systems are implemented in medical, educational, bank, agricultural and other industries. Wide deployment of information systems requires fast, flexible and reliable development methods. Nowadays Agile methodology is very popular and well-known development approach, it allows implementing information systems from the scratch. There are a lot of books and papers saying Agile based methods can be used for any development project. Is this methodology applicable for implementing corporate information systems? This is the question we will try to find answer to within this paper.

II. STATE OF THE ART

There are many papers and books devoted to Agile principals, values and methods [1-5]. Let’s group it into three categories: describing methodology, developing corporate information systems (hereinafter – ERP-systems) based on Agile approaches and adopting methodology for ERP-system implementation. Books from 1st category [1-3] explain how to arrange team, shape product and sprint backlog, estimate solution complexity, schedule and perform sprints. It also has explanation, reasons, goals, key features using Agile methods for developing software systems. This good source of knowledge describes a lot, however after reading it there is no understanding how to adopt it for implementing complex, integrated and expensive ERP-systems.

Reading books from 2nd category [4-5] one can learn how to transform project team and shape activities to develop ERP-systems via Agile. Some tasks are not mentioned in Agile methodology but must be executed to follow specific of ERP-projects, for example, multi steps testing, preparation design and handover documents. Such books are mostly relevant for developing ERP-systems, but not customizing.

Papers of 3rd category [6-7] try to adopt a part of Agile principals in real life ERP-system implementation projects including both development and customizing activities. Such articles illustrate project types where methodology can be used: implementation from the scratch, rollout projects and modification of existing ERP-systems. However, results of analysis are more heuristic then probative.

III. PROBLEM STATEMENT

The purpose of this paper is analyzing Agile based methods can be used in ERP-system implementations to reduce project costs, man-days and duration. To cover this topic following tasks to be solved: considering differences between corporate information systems and other applications, exploring incremental and spiral implementation approaches, mapping Agile principals with ERP-system implementations.

IV. OVERVIEW

There are variety of information systems but not only ERP. Depending on business area for automation it can be MRP, CRP, ERP2, SRM and other systems [8]. Similarity of such systems is in a scope of covering enterprise business operations. Thus, corporate information system penetrates all company's transactions and has hundreds of internal and external interconnected programs grouped in single ecosystem. There are three main strategy to deploy complex information systems: cascade, incremental and spiral. Originally ERP-systems were implemented based on cascade or one-pass model. Agile methodology is a mixture of incremental and spiral strategies (multi-pass model).

A. ERP-system implementation projects

ERP-system is a complex software package solution. It has already existing programs which can be modified or configured for business needs. Moreover, there are SQL-based programming language and environment embedded in ERP-system to develop any new tool. ERP is OLTP-system, thus it stores information in database tables per each business process [9]. According to book [10], company's requirements are covered by ERP-system functionality following way: 30 % by configuration, 60 % by modification of already available programs or developing it from the scratch, the rest 10% can be configuration or development depending on customer specific.

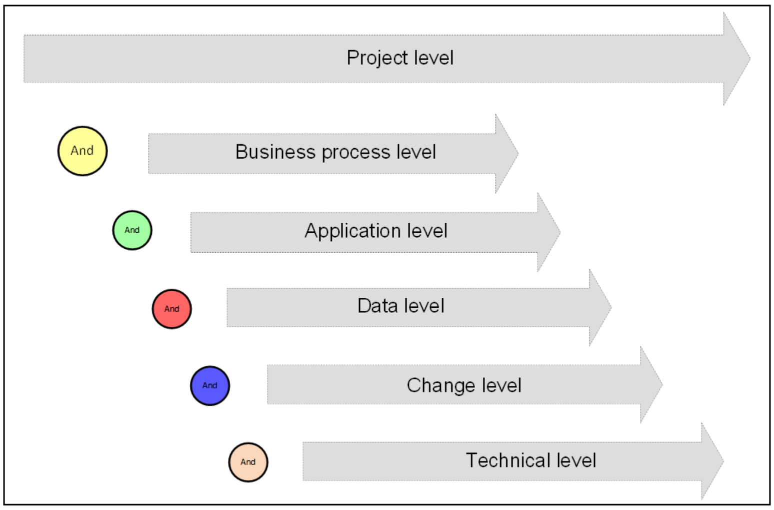


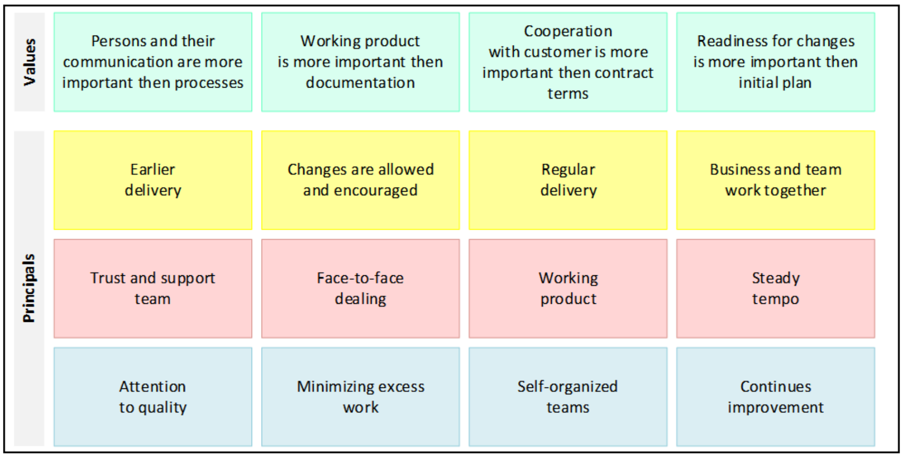
Fig. 1. *Levels of implementation ERP-system*

There is sharp segregation of duties in implementation projects: project manager controls project parameters like budget, durations, resources, scope, risks etc. [11], functional consultants design and test programs, programmers build applications. Taking into account classic approach for designing information systems [12], each developed program requires understanding business process itself, creating correct and normalized database tables for storing information under process, preparing technical infrastructure to make program work. There are summarized aspects of implementation projects presented on fig. 1: primary levels allow building software system (application, data, business process and technic), slave – support deployment (project and change management). Therefore, developing programs for ERP-system is difficult technical as well as organizational exercise. Other important ERP-project features are listed below:

- Legal, business continuity and other requirements to ERP-system come from company's employees.
- Predefined and limited number of business objects to realize and configure requirements.
- A great number of programs to be modified and developed from the scratch to cover requirements.
- ERP-system is considered as a core system in case of rollout and from the scratch projects.
- Each developed program must be supported by functional and technical specification documents.

B. Agile methodology

One-pass implementation model has serious weak point – there is no feedback loop between developed program and end user until final acceptance test (hereinafter – UAT). UAT can be considered as a fine tuning for software system, excluding significant program changes. Multi-pass models improved it by splitting implementation project into small iterations of building a part of program. Results of each iteration are shown to end users to take feedback as soon as possible and to plan improvements in next iterations. The main differences between incremental and spiral models are in risk management and stop criteria. Spiral implementation model pays much more attention to risk identification, analysis and mitigation, moreover it has more structured and transparent criteria for project ending.



Values	Persons and their communication are more important then processes	Working product is more important then documentation	Cooperation with customer is more important then contract terms	Readiness for changes is more important then initial plan
Principals	Earlier delivery	Changes are allowed and encouraged	Regular delivery	Business and team work together
	Trust and support team	Face-to-face dealing	Working product	Steady tempo
	Attention to quality	Minimizing excess work	Self-organized teams	Continues improvement

Fig. 2. *Agile values and principals*

Agile is a practical methodology clarifying, refining and making multi-pass models be alive. There are Scrum, Kanban and Feature Driven Development methods extending classical incremental model, eXtreme Programming and Rapid Application Development technics empowering spiral ones. The manifesto [13] summarizes key values and principals of Agile methodology (fig. 2). Grouping principals together will give us three major activities:

- Development a part of program to demonstrate it to user for fast feedback and possibility to resolve defects as soon as possible. Moreover, each built sub-program is working in production environment and gives value to company.
- Close communication with end users to prioritize requirements for further realization, make them be familiar with program and to advance solution to others.
- Continues education of team members, improvement ways of working, people interchangeability and self-organization.

Organizational structure of project team in one-pass and multi-pass models are completely different. Thus, if there is decision taken to use Agile based methods for developing programs instead of waterfall ones, then delivery strategy, project team and their knowledge must be transformed heavily. The last one takes much time and money, therefore decision must be made taking into consideration all pros and cons.

V. IMPLEMENTATION

ERP-system is a great number of integrated information systems or applied program software. Agile methodology is a set of methods allowing development of simple and complex programs. In theory Agile methods might be used in ERP-systems implementation. Is that true? Let's consider key Agile principals and their reflection in ERP-systems to answer to the question.

A. Earlier, regular delivery and working program

Earlier and regular delivery, working product are key principals from Agile manifesto. Application to be developed and used in a company can be divided into smaller parts: tools and functions. Each function, tool and application itself brings value to enterprise. If each part of the application can bring value, then it's reasonable to develop and implement it separately as well. Thus, company will take benefit of application not at the end of project, when it is fully developed, but much earlier. Moreover, end user can suggest, which part of the program or function has higher priority (that is more profitable for company) and be delivered at first. Therefore, company can return some investments before project end. This is in line with Agile methodology: project is divided into iterations, each part of application will be built under single/different iterations, application will be fully developed once all iterations passed.

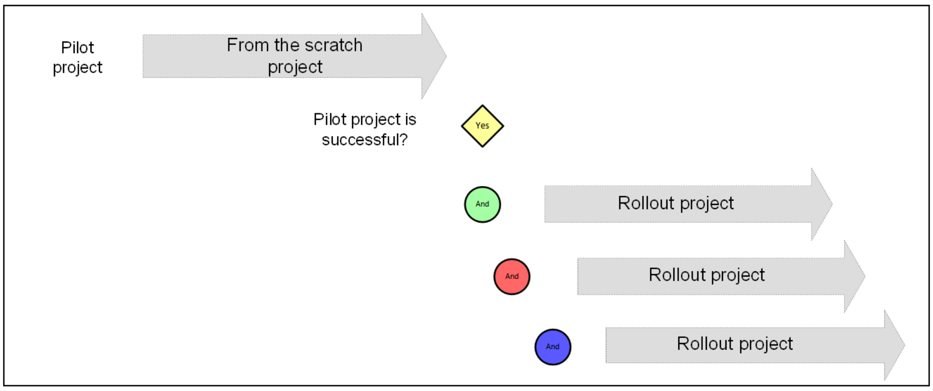


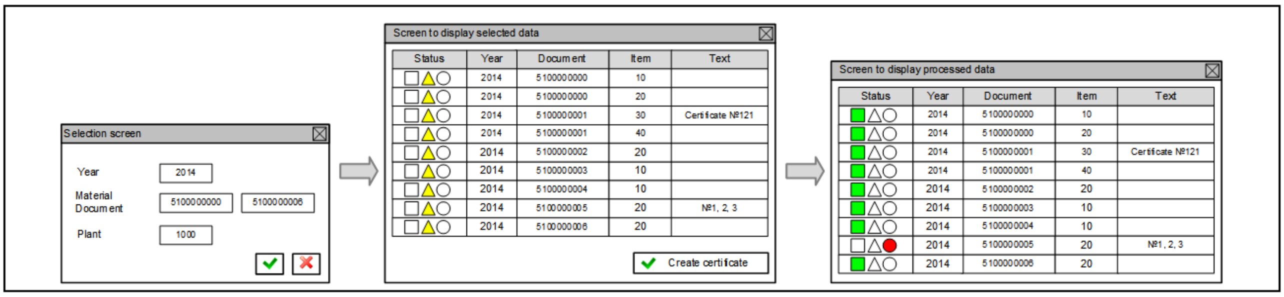
Fig. 3. *Pilot, from the scratch and rollout ERP-projects*

Functionality of corporate information system can automate most business processes of enterprise or only a few ones. Typically, ERP-system is implemented to cover full set of logistic, finance and human capital processes. If ERP is implemented in a group of companies, then to mitigate project failure risks following approach is applied: pilot ERP-system is deployed in single company to estimate its effectiveness, if implementation was successful and system justified end user expectations, then program solution will be rolled out for the rest of companies (fig. 3). The specific of ERP-project is that solution will be implemented fully or nothing. If one tries to deploy ERP-system only to automate single process, it will cost much money and give no advantages.

Statement 1. Earlier, regular and workable delivery principals of Agile methodology are not relevant for rollout and from the scratch ERP-projects. However, it can be applied in evolution ERP-projects, where core system has already been deployed and fine-tune modification needed. In this case Agile can drive fast and flexible solution delivery.

B. Continues face-to-face demonstration

Splitting implementation project by iterations, building a part of program solution per iteration, demonstration results of each iteration to end users are content of Agile methodology. Being an extension of incremental and spiral implementation models, Agile methods require feedback from users to keep on developing. By demonstrating semi-finished program solution to users, it's easier to clarify requirements and align expectations. This is a way to reduce business uncertainty by means of Agile. Moreover, regular demonstration of program to users allows to plan and do slight changes in application much before UAT.



Selection screen

Year: 2014
Material Document: 5100000000 5100000006
Plant: 1000

Screen to display selected data

Status	Year	Document	Item	Text
□▲○	2014	5100000000	10	
□▲○	2014	5100000000	20	
□▲○	2014	5100000001	30	Certificate №121
□▲○	2014	5100000001	40	
□▲○	2014	5100000002	20	
□▲○	2014	5100000003	10	
□▲○	2014	5100000004	10	
□▲○	2014	5100000005	20	№1, 2, 3
□▲○	2014	5100000006	20	

Screen to display processed data

Status	Year	Document	Item	Text
🟩△○	2014	5100000000	10	
🟩△○	2014	5100000000	20	
🟩△○	2014	5100000001	30	Certificate №121
🟩△○	2014	5100000001	40	
🟩△○	2014	5100000002	20	
🟩△○	2014	5100000003	10	
🟩△○	2014	5100000004	10	
□△●	2014	5100000005	20	№1, 2, 3
🟩△○	2014	5100000006	20	

Fig. 4. *Structure of application in ERP-system*

Graphical user interface for most programs in ERP-system is very similar, it contains selection screen, where user can put initial data restrictions; screen displaying selected data based on initial restrictions and screen presenting results of data processing (fig. 4). RICEF acronym is used to distinguish program types: report, interface, conversion, enhancement and form [14]. The number of screens depends on RICEF, for instance, report has two screens, conversion contains all three. As we said, some business requirements are covered in ERP-system by configuration. It's important to say, usually customizing can realize requirement only one way. If programs are very similar from development point of view and configuration is very limited, then there is no need in continues demonstration application to end users. Users are involved in ERP-project differently than in Agile: signing-off functional designs, preparing test scripts, performing training session for other users etc. Moreover, decreasing business uncertainty is to be done by creating assumptions during design and before build.

Statement 2. Due to restricted number of business objects used for ERP-system developing and limited variations how requirements can be customized, continues demonstration of product will not bring much value for any type of ERP-project.

C. Changes are allowed and encouraged

Implementation project is divided into small parts as per Agile methodology, thus each iteration of development brings working semi-program. Remarks and defects identified under unit testing and demonstration to users will be prioritized and allocated for next iterations. Identification, prioritization and allocation of new requirements are recurrent activities, it's clearly stated in methodology what to do with it and how to behave. While planning next iteration, requirements can be changed completely.

ERP is master system and single source of true for all other integrated applications. It's impossible to cover all requirements during rollout or implementation from the scratch. Thus, only important requirements are included in project scope: legal and business continuity. Requirements can be refined and amended, however majority will remain the same. This is the only way to run complex program solution interrelated with many other 3rd party systems and automating whole enterprise. All the other requirements can be realized as a part of ERP evolution project right after successful go-live.

Statement 3. Changes in requirements are not encouraged in ERP-projects from the scratch and rollouts. Due to ERP is a core system, to make it run only obligatory requirements will be realized.

VI. EVALUATION

Having considered Agile principals and its usage in ERP-projects, let’s summarize statements 1-3 (tab. 1). ERP is a core system, projects of implementing it has low technical uncertainty and high business ambiguity. Thus, it’s mostly ‘know how’ projects. On the contrary, multi-pass models are oriented for high technical and business uncertainties. Agile-based methods can be used in system evolution projects, as said in paper [5] and described above. However, it will not bring big advantage in comparison with waterfall methodology due to there are low technical risks.

Table 1. Implementation models per ERP project

ERP project type	Implementation model
From the scratch	One-pass
Rollout	One-pass
Evolution	One-pass or Multi-pass (Agile)

Implementation models were identified for eleven ERP-projects different types. Author participated these projects last ten years. It was SAP ERP implementation in Russian and international companies (tab. 2). All rollout and from the scratch projects used one-pass implementation model, like ASAP (Accelerated SAP) or self-developed waterfall-based methodology. Evolution projects were driven mostly by waterfall model and sometimes – incremental (Agile Scrum).

Table 2. Percent of using one and multi-pass models

ERP project type	Number of projects	One-pass	Multi-pass
From the scratch	3	3 (100%)	
Rollout	5	5 (100%)	
Evolution	3	2 (66%)	1 (34%)

Statement 4. Agile methodology can be widely used in ERP-projects only if it’s adopted for rollout and ERP-projects from the scratch. Currently it’s too expensive to drive rollout and ERP-projects from nothing based on waterfall implementation model, then transform team for evolution ERP-projects, where Agile is used nowadays.

If average tab. 2 results, then one of ten ERP-projects is implemented by Agile methodology. Multi-pass models help to overcome business uncertainty by continues demonstration semi-product to users, however it can be done also in one-pass models by means of prototyping. Agile allow end users to select critical requirements to be developed in ERP-system at first, if adopt change request procedures properly, then it can be realized in waterfall methodology as well.

Considering implementation methodology for ERP-system deployment, a lot of facts must be analyzed: experience of team members, internal procedures in company, uncertainty, number of involved parties etc. Currently using Agile methodology in ERP-projects is more fashionable rather than reasonable.

VII. CONCLUSIONS

In this paper we have considered Agile methodology and its usage in ERP-system implementation projects. Key differences between ERP-system and other applications were discussed. A brief review of multi-pass methods was done to underline its strong points. Key Agile principals were mapped with ERP projects to understand if multi-pass model was applicable. It was shown the original idea of Agile was far away from complex ERP-system implementation. As a result, about 10% of ERP implementation projects are driven by Agile-based methods, mostly for modification of already existing system.

REFERENCES

1. M. Cohn, Agile estimating and planning: Pearson publisher, 2005, 368 p.
2. J. Sutherland, Scrum: the art of doing twice the work in half the time: Cornerstone digital publisher, 2014, 258 p.
3. A. Stellman, J. Greene, Learning Agile: understanding Scrum, XP, Lean and Kanban: O'Reilly media, 2013, 420 p.
4. S. Robson, Agile SAP: It governance publishing, 2013, 218 p.
5. A. Adi, SAP Activate: issues and challenges in large-, mid- and small-scale Projects: Createspace independent publishing platform, 2018, 218 p.
6. D. Stepanov, A. Velsovskiy, “Using Agile Scrum in SAP projects”, Corporate information systems, 2018, № 1, p. 9-17. (in Russian).
7. G. Blokdyk, ERP and Agile methodologies: a complete guide: 5STARCooks, 2020, 248 p.
8. M. Ali, L. Miller, “ERP system implementation in large enterprises – a systematic literature review”, Journal of enterprise information management, 2017, vol. 30, issue 1, p. 666-692.
9. A. Habadi, “An Introduction to ERP-systems: architecture, implementation and impacts”, International journal of computer applications, 2017, vol. 167, issue 9, p. 1-4.
10. G. Kalyanov, Consulting: from business strategy to corporate information system, Goryachya liniya telecom, 2011, 2010 p. (in Russian).
11. A Guide to the project management body of knowledge (PMBOK Guide): project management institute, 2017, 756 p.
12. D. Stepanov, “Strategy based ERP-system implementation approach”, Fundamental problems of radio-electronic instrumentation, 2017, №3, vol.17, p. 697-700.
13. Principles behind the Agile Manifesto. URL: http://agilemanifesto.org/iso/en/principles.html.
14. D. Stepanov, “Shaping universal requirements to ABAP-programs while preparing functional specification documents”, Actual problems of modern science, 2014, vol. 78, № 4, p. 258-268. (in Russian).

PAPER DETAILS

Dmitry Yu. Stepanov. Using Agile methodology in ERP-system implementation projects. // Proceedings of the 35th International Conference on Information Technologies, InfoTech 2021. – 2021. – p.80-86. – URL: https://stepanovd.com/science/article/115-2021-6-agile.