



Agile Software Engineering Practices and ERP Implementation with Focus on SAP Activate Methodology

Adnan Kraljić^(✉) and Tarik Kraljić

Ghent University, Ghent, Belgium
{adnan.kraljic,tarik.kraljic}@ugent.be

Abstract. The Enterprise Resource Planning (ERP) implementation is a complex and active process, one that involves a mixture of technological and organizational interactions. Often it is the largest IT project that an organization has ever launched and requires a mutual fit of system and organization. Concept of an ERP implementation supporting business processes across different departments in organization is not a generic, rigid and uniform process - it is a vivid one and depends on number of different factors. As a result, the issues addressing the ERP implementation process have been one of the major concerns in industry. Therefore, ERP implementation process receives profound attention from practitioners and scholars in academic or industry papers. However, research on ERP systems so far has been mainly focused on diffusion, use and impact issues. Less attention has been given to the methods/methodologies used during the configuration and the implementation of ERP systems; even though they are commonly used in practice, they still remain largely undocumented in Information Systems research domain. This paper aims to provide insight from practice (SAP ERP implementation team up with 20 SAP consultants including authors of this paper) regarding the agile engineering practices in ERP implementation process. One of stubbornly persists belief was that ERP systems cannot be part of agile development due to their complexity and nature. However, it is becoming obvious that agile engineering practices will not be anymore exclusively linked to software development as SAP (biggest world ERP vendor) recently introduced its first agile ERP implementation methodology named SAP Activate Methodology.

Keywords: ERP · ERP implementation · Agile implementation methodology
Sprints · Phases

1 Introduction

Implementing an ERP system is a major project demanding a significant level of resources, commitment and adjustments throughout the organization.

Problem Description. Often the ERP implementation project is the single biggest project that an organization has ever launched. As a result, the issues surrounding the implementation process have been one of the major concerns in industry. And it further worsens because of numerous failed cases including a few “fatal” disasters which lead

to the demise of some companies. In previous studies can be found that almost 70% of ERP implementations fail to achieve their estimated benefits [1]. Although ERP can provide many benefits for organization, goals are often changed to getting the system operational instead of realizing the goals. Reflecting such a level of importance, the largest number of articles in literature belongs to this theme. It comprises more than 40% of the entire articles. Less attention has been given to the methods/methodologies used during the configuration and the implementation of ERP systems; even though they are commonly used in practice, they still remain largely unexplored and undocumented in Information Systems research domain. However, practically relevant research that addresses industry sectors that have to apply more than just agile (e.g., due to the development of safety critical systems or legal regulations) is rare. Back in 2003, Boehm and Turner [1] described a first approach how to combine agile and traditional software development for defining a balanced software development strategy, and Diebold and Zehler described how agile can be integrated into rich processes. Yet, as also argued in, systematic construction procedures are missing as most available research documents ad-hoc and user specific approaches to construct organization- and project-specific development approaches.

Objective. The overall goal of the research presented in this paper is to provide an insight from industry regarding the ERP implementation methodologies trend (focusing on world's biggest ERP provider SAP). The research presented aims to show the key components of SAP ERP implementation methodologies focusing on differences between agile and waterfall elements in each of them (and their evolution).

Contribution. In this paper, we present the ongoing research on agile software engineering practices applied on ERP implementation methodologies in practice.

2 ERP Implementation Methodologies in Literature

Several models of ERP implementation methodologies are provided in literature and they vary according to e.g. the number of phases. The phases in ERP implementation frameworks are often counted as between three and six, according to Somers and Nelson. However, the model of includes 11 phases and it gives practical checklist-type guidance for an ERP implementation. On the other hand, the models of Markus and Tanis or Parr and Shanks are very general, and are merely used for analyzing ERP implementation projects [2]. The models are useful in studying, analyzing and planning ERP implementation. The selection of ERP implementation method mentioned in paper is based on the degree of "institutionalization" in the scientific community. Livari and Hirschheim described six criteria to determine institutionalization: including (1) the existence of scientific journals, (2) scientific conferences, (3) textbooks, (4) professional associations, (5) informational and formal communication networks, and (6) citations. There are number of different ERP implementation methodologies mentioned and described in literature. However, there is an issue with methodology scope, context and its ambiguity. For example, some methodologies treat the phases before the acquisition of an ERP system (and are focused on it), while some methodologies put stress on phases after the ERP system has started to be used (production phase) [3].

Next table summarize list of proposed implementation methodologies followed by the degree of institutionalization in scientific community [4] (Table 1).

Table 1. ERP implementation methodologies in literature

Author(s)	ERP implementation model
Bancroft et al. (1998)	(1) Focus, (2) Creating As – Is picture, (3) Creating of the To-Be design, (4) Construction and testing and (5) Actual Implementation
Kurupparachchi et al. (2000)	(1) Initiation, (2) Requirement definition, (3) Acquisition/development, (4) Implementation, and (5) Termination
Markus and Tanis (2000)	(1) Project chartering, (2) The project, (3) Shakedown, and (4) Onward and upward
Makipaa (2003)	(1) Initiative, (2) Evaluation, (3) Selection, (4) Modification, Business process Reengineering, and Conversion of Data, (5) Training, (6) Go – Live, (7) Termination, and (8) Exploitation and Development
Parr and Shanks (2000a)	(1) Planning, (2) Project: a. setup, b. reengineer, c. design, d. configuration and testing, e. installation (3) Enhancement
Ross (1999)	(1) Design, (2) Implementation, (3) Stabilization, (4) Continues improvement and (5) Transformation
Shields (2001)	Rapid implementation model of three phases and 12 major activates
Umble et al. (2003)	(1) Review the pre-implementation process to date, (2) Install and test any new hardware, (3) Install the software and perform the computer room pilot, (4) Attend system training, (5) Train on the conference room pilot, (6) Established security and necessary permissions, (7) Ensure that all data bridges are sufficiently robust and the data are sufficiently accurate, (8) Document policies and procedures, (9) Bring the entire organization on – line, either in a total cutover or in a phased approach, (10) Celebrate, and (11) Improve continually
Verviel and Halington	(1) Planning, (2) Information search, (3) Selection, (4) Evaluations, and (5) Negotiation

It is evident that there is no ground based ERP implementation methodology - widely accepted and tested. Even though they are commonly used in practice (ERP implementation methodologies) they still remain largely unexplored and undocumented in Information Systems research domain. Additionally, academic literature does not provide any relevant material regarding the influence of agile software engineering practices on ERP implementation methodology. In next paragraph we will describe newly introduced SAP ERP implementation methodology (SAP Activate Methodology).

3 SAP Activate Methodology (Agile Based Methodology)

In business informatics, software project methodologies define implementation and development guidelines for “out-of-box” software implementation as SAP. Historically, SAP had its own project methodology called ASAP, which stands for

Accelerated SAP (and was completely based on waterfall project management principles). Few years ago, SAP introduced SAP Launch project methodology for its Cloud product portfolio which was also based on waterfall principles. Launch methodology has been transformed into SAP Activate Implementation methodology (currently in use for SAP ERP products) and presents the first agile based ERP implementation methodology introduced by SAP. The SAP Activate methodology comprises of six phases as highlighted in Fig. 1, which is a disciplined approach to managing complex projects, organizational change management, solution management, & industry specific implementations [5]. In next paragraph, we will briefly describe SAP Activate ERP implementation phases.

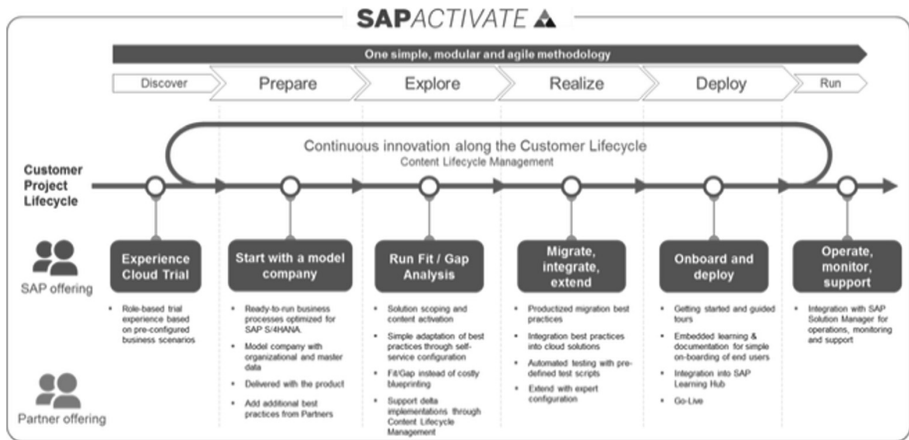


Fig. 1. SAP activate methodology

Prepare - This phase encompasses the entire project preparation and planning activities with infrastructure, hardware/network sizing requirements completed. It involves setting up the infrastructure, team, project goals, charter, and agree upon schedule, budget, risk baseline, proof-of-concept planning if applicable with implementation sequence. The project manager on the ground will discuss with the customer project manager to identify risks early on with a mitigation plan. The PM will be responsible for drafting a high-level project plan with all milestones with a detailed task level plan chalked out with critical dependencies. Each phase deliverable should be agreed between both parties. Finally, a project organization, steering committee is organized with assigned resources [6].

Explore - This is the most crucial phase of the project for a project manager as he just about to steer the ship, like a captain. The objective of this phase is to be on a common platform on how the company plans to run SAP for their business operations. Thus, a PM is responsible for analyzing the project goals and objectives and revise the overall project schedule if required. In simple terms, it is the critical requirements gathering phase, A PM might use appropriate tools to collect requirements with required traceability. The

result is the Business Blueprint, which is a detailed flow of business process AS-IS, how they run the business operations with a TO-BE mapped in SAP, on how these business operations will run in SAP. Depending on the implementation complexities, number of business process, Blueprint workshops might span for a few days or weeks or even months, in a complex environment. The output of this phase is the baseline configuration in SAP with detailed custom code requirements analysis done [7].

Realize - In simple terms, realization is the actual development phase of the project, where you'd configure, develop custom code and conduct required testing. It involves coding-unit testing-integration testing-User acceptance testing (UAT). As per the business blueprint and mapping the SAP system as agreed with business, all the business process requirements will be implemented. In reality, there are two major work packages: (a) Baseline (major scope); and (b) Final configuration (remaining scope). The success of any implementation project relies on how closely you're able to develop custom code, test and release it to the UAT phase, in order to support adequate testing by the users. Also, the challenge is to adopt changes as indicated during the UAT. This phase is resource intensive and the team is at peak team size to ensure all deliverables are met and sign-off. Often times Integration fail due to lack of test data, and testing in a "PRD" like environment to be able to test all critical business scenarios. A good practice is to copy a "PRD"-like environment and start testing if the system already exists. If it is GreenField environment, ensure adequate test data is available to test it rigorously [8].

Deploy - Final preparations before cutover to production ensure that that the system, users, and data are ready for transition to productive use. The transition to operations includes setting up and launching support, then handing off operations to the organization managing the environment.

In next few paragraphs we will provide industry report from major medical institution in Bosnia and Herzegovina.

4 Report from the Industry

This industry report provides insight into an implementation of SAP ERP solution in a major medical institution in Bosnia and Herzegovina, with several thousands of employees. The project was initiated by management of the hospital with purpose to eliminate the ineffectiveness of the current information system. Analysis of the current financial system and the list of new system requirements have been prepared by external consulting company. This was prerequisite for announcing a public tender for selection of ERP software solution integrator. After several months of tender procedure and assessment of the best vendor (price was eliminatory criteria, in accordance with the law), software integrator was selected. At the end, SAP All-in One solution was preferred one. In next few paragraphs are briefly provided some quick facts regarding the project. As recommended by external consultants the main tasks of the project were:

- To centralize the information system;
- To increase data integrity and consistency;

- To focus on accounting and financial department processes;
- To improve drug warehouse management and billing system;
- To provide comprehensive and accurate reports for top management.

The project incorporated five SAP modules: FI (Finance), CO (Controlling), MM (Material management), SD (Sales and Distribution) and HR (Human resources). SAP integrator offered a team of 20 SAP solution consultants, including one SAP system administrator. Additionally, two consultants (ABAP programmers) were teamed up for specific ABAP developments. The implemented system was standard SAP ERP All-in-One. Since it was not specialized health care solution, additional industry-specific functionalities needed to be developed to fulfill basic needs. They were mostly related to the processes of Materials Management and Sales and Distribution.

4.1 Remarks from Participations

Agile SAP implementation methodology named SAP Activate was presented to team members at the kick off meeting. Some of the remarks presented in next paragraphs are also mentioned in industry whitepaper named Whitepaper “Scrum and ERP – do they go together” from Mr. Boris Dloger. Some of the most repeated remarks from participants were:

- “What is the intention for using agile practices for standard ERP implementations? Typically ERP implementation does not involve any kind of development, just customizing of software to the customer’s needs.”
- “Agile terminology is ambiguous for us – it is terminology of software development process, not ERP implementation process!”
- “Team structure is ambiguous and not possible in ERP implementation process – lack of the rigid hierarchy will create chaos”
- “In order to customize ERP and adjust the diverse and complex ERP components we need dedicated experts. Work in cross functional teams is complex!”

We will describe those remarks in detail in next chapters providing the experience of team members.

4.2 Agile Terminology in SAP ERP Implementation

Getting the terminology right is often a complex task. In order to have a better insight in this topic author used industry experience and knowledge of Mr. Anton Karnaukhov, SAP expert, available on his blog. Agile project management brings a number of new terms that are critical to understand in order to be able to comprehend how agile can help you implement SAP solutions in a better way. Projects, especially large and complex, typically should have both waterfall and agile characteristics and elements to them. A combination of elements from both approaches usually works best for any given project, regardless of context, complexity, size or any other factor. The key roles that should be integrated in an organization that mixes traditional waterfall and agile practices in are simplified in Table 2 [9].

Table 2. Traditional waterfall vs. Agile practices

Waterfall role	Meet half way	Agile role
Sponsor	Same	Sponsor
Product manager	Product manager would typically be accountable for the whole product or product line, and oversee potentially many product owners, or act as a product owner herself	Product owner
Project manager	Solution architect would be elevated to oversee programs and dependencies of projects, leaving teams to manage more autonomously	Solution architect
Project methodology “champion”	Same idea here, if either practices are applied, care should be taken to apply with true understanding of why they are applied to not dilute the intent and effect	Scrum master (Scrum being by far the most widely adopted agile framework, hence this role used here)
End users	Active involvement from the beginning to the end. Traditional stage gates with wider presentation, key roles from core stakeholder groups involved continually	End users
Other stakeholders	Address and respect the product owner	Other stakeholders

Furthermore, those are “items” of terminology used regarding agile implementation process, shown in Table 3 [10].

4.3 Team Structure of Project

In the past, different variations of project team composition was suggested as standardized, but the fact is that there is no canned project team setup that will work on every project [9].

At the very beginning of any SAP implementation project there is a process of defining a baseline of features that make up the general scope of a particular product. A release is typically associated with a go-live - for example going live with Material Management (MM) and Finance (FI) modules of SAP ERP, or a particular business unit going live with an SAP Extended Warehouse Management (EWM) system. One important thing to note is that the initial list of features in a release is not “frozen for changes”. Most of the time, if not all of the time, new features will be added to the list as the project progresses and some of the old features will be either moved to a later release or even removed entirely from the project. This is perfectly normal and even encouraged. In fact, part of embracing change is constantly re-evaluating the prioritization of project features and user stories and focusing on the ones that bring the most business value.

Table 3. Agile terminology

Agile terminology	Description (analogy to SAP ERP implementation terminology)
Release	Each release is associated with some type of go-live where a number of features are moved to production. For example, a “big bang” SAP implementation project could have just a single large release. A more phased approach could lead to many releases within a single SAP implementation project - SAP ERP Finance and Human Resources in one release and Logistics modules taking place in a separate release shortly after
Feature	These are sometimes called Epic Stories or Epics. Features represent large sets of functionality, for example - sales order processing, warehouse management picking, month-end close, etc.
User story	Each user story describes a particular business requirement and is assigned to a single feature. In many ways a user story is the next level down in terms of detail after a feature
Sprint	A sprint is typically a pre-determined period of time (2 weeks, 4 weeks, 6 weeks, etc.) within which a set of identified user stories needs to complete
Kanban board	A visual board where columns represent a state that a user story can be in, for example - planned, blueprinting, realization, testing, done. Stories are arranged on the Kanban board and moved from one column to another as progress is being made. Many teams build physical Kanban boards by using tape and post-it notes
Retrospective	A retrospective is a focused session where your team looks back at how the current agile approach is working and which areas can be improved. Many agile teams conduct retrospectives at set intervals of time (every 8 weeks or at the end of every sprint)

A good starting point for building a list of features for an SAP implementation project is SAP’s Best Practices library. Once a baseline or industry-specific package is chosen customer is able to see its requirements presented with a list of SAP best practice scenarios, each designated by a unique number. For example, under the baseline package you will be able to find scenario 112 called “Sales Quotation”. The end result is a list of features that outlines the first pass at defining the overall scope of what we think we are going to accomplish in a particular project [11].

Once the initial feature list is put together we can start building our project team structure. Here are some general guidelines that we try to follow.

4.4 Leadership

In the context of an agile SAP implementations these roles are much less about management and are much more about leadership, which is often a source of tension in many organizations that are trying to do agile for the first time [9]. What is meant by this is that many traditional managers are used to working in an environment where they create layers of superfluous entities (status reports, WBS structures, functional and technical specification documents, detailed tracking of estimated vs actual time) all typically under the banner of “proper management” and for purposes of historical

reference, and then depend on “data” from those entities to reveal red flags in order to take some “corrective action”. With agile the use of such activities is greatly discouraged and many managers who are not successful leaders often struggle to find their niche within an agile project.

4.5 Self-directed Teams

In general, each team is responsible for managing all of the work (features, user stories, tasks, etc.) from start to finish. They are responsible for gathering user stories and tasks (blueprinting), prioritize the stories; performing the configuration and development work (realization), testing, providing documentation, and even of training end-users. All teams are expected to create their own user stories (and tasks) and keep their Kanban boards up-to-date. Teams are expected to engage in continuous dialog with business users and prioritize their own workload [9]. One of the most crucial techniques for promoting self-directed teams is the use of daily stand-ups. These are short and focused meetings, typically at the beginning of each day, where each team member answers the following 3 questions:

- What have I accomplished yesterday?
- What am I planning to accomplish today?
- Is there anything in my way that is preventing me from making progress?

Product Owner

Four product owners were assigned to project team structure. They are important part in providing help with prioritization of user stories, but given the volume of stories in an average SAP implementation (from 1000 to 5000), their involvement in prioritization at the individual story level is limited [12]. We can summarize product owner’s primary objectives as:

- Identification and prioritization of features (in tandem with project manager aka Scrum Master)
- Review and acceptance of delivered solutions through a live demo (usually a batch of 15–30 user stories)

Solution Architect

One solution architects was assigned to project team structure whose role was to ensure that the overall SAP implementation across sub-teams is synchronized. Solution architect is professional with extensive technical experience across many SAP systems and modules. And even though cross-functional sub-teams often rely on solution architects to get direction in complex integration scenarios, the teams themselves still retain the responsibility of aiming all of their user stories and tasks forward [13]. Solution architects are typically responsible for:

- Cohesiveness and robustness of the overall delivered solution for the entire project
- Integration design and testing across teams, systems and modules

Cross – Functional Teams

This is often a source of confusion in SAP project as the term cross-functional has historically referred to cross-module (for example SD, FI, MM, etc.) in the SAP world.

However, in the context of an agile team structure for an SAP project cross-functional refers to the various functions performed by the project team - requirement gathering, configuration, development, documentation, training, etc. [12]. Typically a number of such cross-functional sub-teams is build, each focused on a small number of SAP systems/modules. For example:

- Sales and Distribution team - 4 BPOs, 3 analysts, 1 developer, 1 trainer/instructional designer
- Materials Management team - 5 BPOs, 2 analyst, 1 developer, 1 trainer/instructional designer

4.6 Impact on Project Managers

In Agile engineering practices, a traditional project manager role may not be required to manage an implementation, as the agile teams are self-sufficient (common Project Management roles in an agile project are product owners, scrum masters and scrum team).

- Scrum master coaches the development team to use scrum principles.
- A product owner will generally come from the customer side and will own the requirements and will be part of the scrum team.
- Product owner will be responsible for documentation of requirements which are normally written as user stories. He will also be in charge of the prioritization of the requirements [10].

4.7 Impact on Project Stakeholders and Sponsors

SAP Activate Methodology ensures consistent involvement of project stakeholders and sponsors. Project stakeholders are involved in project planning and retrospective meetings every 2–4 weeks. In a traditional/Waterfall project management scenarios, the client gets involved at a much later stage which results in a mismatch of expectations and project delivery [12].

5 Conclusion

It is evident that agile practices will not be exclusively linked to software development anymore. Some of the remarks that arise from team members were stated in previous chapter. We will restate them again, but this time, providing an insight as a result of “hands on” experience in agile SAP ERP implementation project.

“Why should agile methods be used for standard ERP implementations? This does not normally involve any kind of development, just adapting software to the existing processes.”

ERP implementations intervene in the way many employees work, whether standard or in-house development: changes give rise to uncertainty. In agile engineering practices, great scope is given to communication with the user by means of regular interviews. The users test the product increments and give feedback as to what works well and what does not.

“We don’t understand agile terminology – it is terminology of software development process, not ERP implementation process!”

Getting the terminology right is very important part adopting agile engineering practices truly [14]. Agile project management brings with it a number of new terms that are critical to understand in order to be able to comprehend any of the detailed discussions on how agile can help you implement SAP solutions in a better way. Listed in this paper, in the form of table, we showed that it is possible to map agile and waterfall terminology “one – to – one”

“Team structure is ambiguous and not possible in ERP implementation process – lack of the rigid hierarchy will create chaos”

In general, each team is charged with owning and managing all of their logical units of work (features, user stories, tasks, etc.) from start to finish. That means they are responsible for gathering user stories and tasks (blueprinting), working with the business to prioritize those stories, performing the configuration and development work (realization), testing, documentation, demoing solutions and even training end-users [6].

“To be able to configure and adapt the diverse and complex ERP components we need specialized experts. That makes work in cross-functional teams more difficult.”

It has also proved effective in ERP projects to unite different skills in one team: ERP consultants work with ERP developers, CRM experts with MM experts, while business analysts or system architects also enhance such teams. In this way the requirements are viewed from different perspectives and the exchange of ideas within the team brings aspects to light that the individual alone would not have detected – this again ensures that the “right” product is delivered [12].

In upcoming years we will see more and more research papers and case studies about the influence of agile practices on ERP implementation process. It is expected that all major ERP providers present their unique agile driven ERP implementation methodologies [14]. In upcoming papers (and as a part of PhD project) we will focus on providing hybrid Agile Waterfall ERP implementation methodology designed on design science postulates.

References

1. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile Software Development (2002)
2. Aiken, L.S., West, S.G.: Multiple Regression: Testing and Interpreting Interactions. Sage, Newbury Park (1992)
3. Akgün, A.E., Byrne, J., Keskin, H.: Knowledge networks in new product development projects: a transactive memory perspective. *Inf. Manag.* **42**, 1105–1120 (2005)
4. Akgün, A.E., Keskin, H., Byrne, J., Imamoglu, S.Z.: Antecedents and consequences of team potency in software development projects. *Inf. Manag.* **44**, 646–656 (2007)
5. Al-Fatish, F., Roemer, M., Fassunge, M., Reinstorf, T., Staader, J.: ASE: Immer busser mit starken Teams! Agile Software Engineering bei der SAP. *Object Spectr.* **1** (2011)
6. Anderson, D.J.: Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results. The Coad Series. Prentice Hall, Upper Saddle River (2004)
7. Andreessen, M.: Why software is eating the world. *Wall Street J.* (2011)

8. Argote, L.: Input uncertainty and organizational coordination in hospital emergency units. *Adm. Sci. Q.* **27**, 420–434 (1982)
9. <http://www.xpertminds.net/blog/111>
10. Baccarini, D.: The concept of project complexity: a review. *Int. J. Project Manag.* **14**, 201–204 (1996)
11. Balijepally, V.G., Mahapatra, R.K., Nerur, S., Price, K.H.: Are two heads better than one for software development? The productivity paradox of pair programming. *Manag. Inf. Syst. Q.* **33**, 91–118 (2009)
12. <https://go.forrester.com/consulting/>
13. Jalali, S., Wohlin, C.: Global software engineering and agile practices: a systematic review. *J. Softw.: Evol. Process* **24**, 643–659 (2012)
14. Glogler, B.: Scrum and ERP: do they go together? Industry Whitepaper (2014). https://borisglogler.com/wpcontent/uploads/2014/07/Whitepaper_Scrum_and_ERP_en.pdf