The Journal of Systems and Software 149 (2019) 63–82

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Adopting integrated application lifecycle management within a large-scale software company: An action research approach

Eray Tüzün[a,*], Bedir Tekinerdogan[b], Yagup Macit[c], Kürşat İnce[c]

[a] Department of Computer Engineering, Bilkent University, Ankara, Turkey
[b] Information Technology, Wageningen University, Wageningen, the Netherlands
[c] HAVELSAN A.Ş. Ankara, Turkey

ARTICLE INFO

Article history:
Received 4 January 2018
Revised 19 September 2018
Accepted 16 November 2018
Available online 17 November 2018

Keywords:
Application lifecycle management
Application lifecycle management transformation
Action research
Change management
DevOps

ABSTRACT

Context: Application Lifecycle Management (ALM) is a paradigm for integrating and managing the various activities related to the governance, development and maintenance of software products. In the last decade, several ALM tools have been proposed to support this process, and an increasing number of companies have started to adopt ALM.
Objective: We aim to investigate the impact of adopting ALM in a real industrial context to understand and justify both the benefits and obstacles of applying integrated ALM.
Method: As a research methodology, we apply action research that we have carried out within HAVELSAN, a large-scale IT company. The research was carried out over a period of seven years starting in 2010 when the ALM initiative has been started in the company to increase productivity and decrease maintenance costs.
Results: The paper presents the results of the action research that includes the application of ALM practices. The transitions among the different steps are discussed in detail, together with the identified obstacles, benefits and lessons learned.
Conclusions: Our seven-year study shows that the adoption of ALM processes is not trivial and its success is related to many factors. An important conclusion is that a piecemeal solution as provided by ALM 1.0 is not feasible for the complex process and tool integration problems of large enterprises. Hence the transition to ALM 2.0 was found necessary to cope with the organizational and business needs. Although ALM 2.0 appeared to be a more mature ALM approach, there are still obstacles that need attention from both researchers and practitioners.

1. Introduction

Current software development projects usually have to cope with the development and maintenance of large scale and complex software systems. To this end, several lifecycle models have been introduced that define different lifecycle activities focusing on different goals such as requirements engineering, design, development and testing. The separation of the activities in the life cycles helps to focus on a single concern in each phase and likewise supports the management. In parallel with the separation of lifecycle activities, separate tools have been introduced for developing and managing the corresponding lifecycle artefacts. Hereby, each tool usually focuses on specific artefact types (e.g. requirements) and optionally provides some mechanisms (e.g. import and export functions) to support the integration with the other lifecycle artefacts and tools. Although this separation of activities is important to manage the overall process it is equally important to integrate and combine the various artefacts in the software development process. Integration requires that the various artefact types can be traced to each other. For example, code elements need to be traced to design elements, which on their turn need to be traced to the requirements. Further, to manage large scale projects the various activities need to be synchronized and aligned where needed. This requires that the individual teams that work on the same project cannot work independently and act in silos. Moreover, to monitor this process, the progress of the project must be visible. For small scale projects, the integration of the artefacts and tools could be carried out to some extent, but for large scale projects this is not scalable. As a result of this, the process and tool integration become an important obstacle for the development and management of software systems. Obviously, for the overall effectiveness

* Corresponding author.
E-mail addresses: eraytuzun@cs.bilkent.edu.tr (E. Tüzün), bedir.tekinerdogan@wur.nl (B. Tekinerdogan), ymacit@havelsan.com.tr (Y. Macit), kince@havelsan.com.tr (K. İnce).

https://doi.org/10.1016/j.jss.2018.11.021
0164-1212/

of the process it is important to provide an environment in which the activities, the artefacts and the tools are properly integrated (Grant, 2012).

Application Lifecycle Management (ALM) is a recent paradigm for integrating and managing the various activities related to the governance, development and maintenance of software products. Recently, an increasing number of companies have started to adopt the ALM process and several ALM tools have been proposed to support this process. Several studies have been published on ALM, primarily including white papers (Chappell, 2008; Pampino, 2011), books (Rossberg, 2014) and conference papers (K. and Välimäki, 2009; Peksens, 2013) . Yet, the topic does still not seem to have gained full attention from the software engineering research community, at least the publications in this domain have been limited so far. There could be different reasons for this, one of which is the refrainment of companies to share their experiences, and/or of the lack of experience with this relatively new concept. The ALM approach seems to be promising but so far, no systematic and empirical evaluation has been provided on the industrial adoption of ALM practices. Several studies can be found that discuss the main concepts (Chappell, 2010), (Rossberg, 2014), (Chappell, 2008) while other studies (Azoff, 2016; Murphy et al., 2013) present comparisons on ALM tools. However, the overall promised impact on ALM practices has not been empirically evaluated yet.

This paper presents the results of an action research study for analyzing the impact of ALM practices within a large company. Action research is an empirical research methodology whereby researchers attempt to solve a real-world problem while simultaneously studying the experience of solving the problem (Davison et al., 2004). Action Research has been widely used as a research approach in social science, and it has been gradually adopted for information systems and software engineering research during the last two decades. The action research of this study has been carried out within the context of HAVELSAN (Havelsan Corporate Web site), a large Information Technology (IT) company which delivers products in the domain of simulation systems, command and control systems, and e-government applications. The action research has been started based on the concrete and urgent need for a holistic management of the products and the optimization of the value of the delivered products. The research was carried out over a period of seven years starting in 2010 when the ALM initiative has been initiated in the company to increase productivity, and decrease maintenance costs. The paper presents the results of the action research that includes both the application of two different approaches of ALM, that is, ALM 1.0 and ALM 2.0. The transitions among the different steps are discussed in detail, together with the identified obstacles, benefits and lessons learned.

The remainder of the paper is organized as follows. In Section 2, we describe the background on ALM and the adopted research methodology, action research. Section 3 presents the context of the action research that has been applied within HAVELSAN. Section 4 describes the result of the first action research cycle including the adoption of ALM 1.0. Section 5 describes the second action research cycle including the application of ALM 2.0. Section 6 discusses the overall results. In Section 7 presents the related work, and finally Section 8 concludes the paper.

2. Background

2.1. Application lifecycle management

Application Lifecycle Management (ALM) includes the entire time from the idea of developing the application to the end of application's life (Chappell, 2008). ALM can be divided into three distinct areas including governance, development and operations (Chappell, 2008). In Fig. 1 we show the relations among these areas together with the elements of each area. The governance area encompasses all of the decision making and project management, and includes *portfolio management, risk management, project management, change management,* and *knowledge management*. Governance activities orchestrate the development and operations areas (DevOps). The development area corresponds to the traditional Software Development Lifecycle (SDLC) and encompasses *requirements engineering, architecture & design, development, build management* and *test management*. The operations area includes the *deployment of the application, customer feedback* and *monitoring*.

In general, three basic motivations are provided for ALM including traceability, visibility and process automation (Schwaber, 2006). *Traceability* defines the ability to trace various artifacts in the project and link them together. Poor traceability between related work artifacts that are produced in different stages of software development will lead to inconsistent artifacts. Hence it is critical to address both the traceability of changing customer needs and requirements during the lifecycle of a project. For most organizations traceability management is a manual process. For small size projects, traceability is to some extent manageable, but for large scale projects traceability becomes soon less tractable. *Visibility* defines the progress of the project and includes visibility of development artefacts. Visibility is important for large scale projects to support the coordination and monitoring of the teams. *Process automation* defines the automation of the adopted process throughout the projects. ALM stresses the importance of automating project tasks for a more effective and less time-consuming process. Having an automated process also decreases the error rate compared to handling the process manually.

In practice, there are two main solution alternatives for an integrated ALM approach (Schwaber, 2006), ALM 1.0 and ALM 2.0. The first category, ALM 1.0, aims to combine the best of breed product for each lifecycle activity. The second approach, ALM 2.0, aims to cover most if not all the lifecycle activities from a single tool vendor. The two different approaches are shown in Fig. 2. As we can see in Fig. 2, in ALM 1.0 there is a separate tool for each lifecycle activity, each of which is using a separate database. The advantage of this approach is the fact that it requires less orchestration and give more freedom to the individual lifecycle tool selection. However, this approach usually leads to silos of disparate information in the organization that does not scale well. Integration of the artefacts produced in each of these tools is primarily based on manual integration.

To address the shortcomings of ALM 1.0 approach, ALM 2.0 approach has been proposed. In contrast to ALM 1.0, the ALM 2.0 approach adopts an integrated data repository for all the lifecycle activities. In this way, the integration problems that were encountered in ALM 1.0 are largely resolved. However, this approach requires more conscious effort in the establishment, adjustment and reinforcement activities associated with enforcing a global set of company processes and practices associated with the selected platform and tool set.

2.2. Action research

In the previous section, we have discussed the concept and evolution of ALM. Despite its importance, no systematic and empirical evaluation has been provided so far on the industrial adoption of ALM. Several papers can be found discussing the main concepts while other papers present comparisons on ALM tools (Portillo-Rodríguez et al., 2012). However, the overall promised impact on ALM has not been empirically evaluated within an industrial context yet.

Several empirical evaluation approaches can be identified including experiments, surveys, case studies, ethnographies and action research (Easterbrook et al., 2008). Among these, action re-

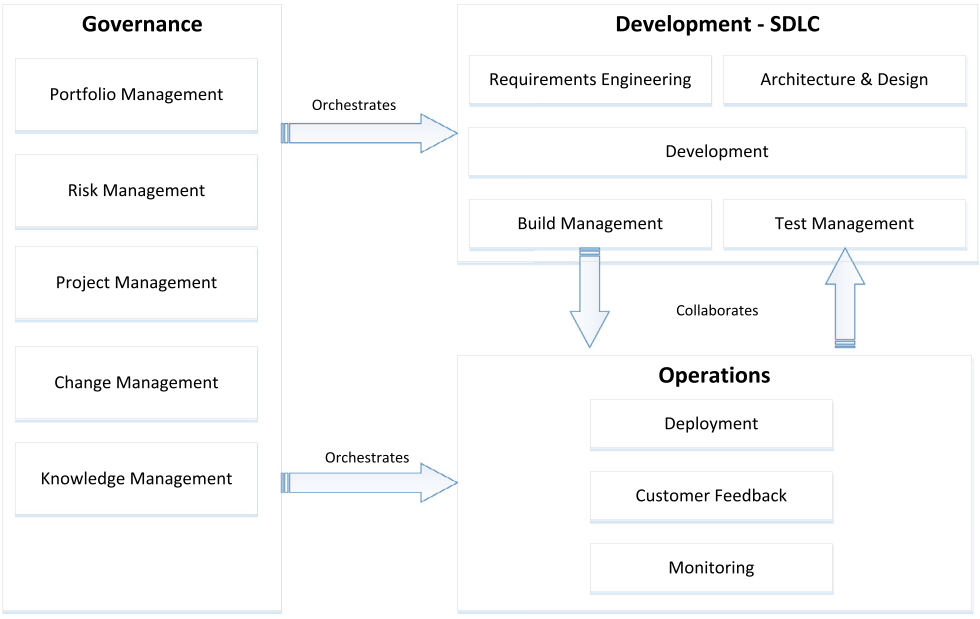

Fig. 1. Conceptual Model of ALM activities (adapted from (Chappell, 2008)).

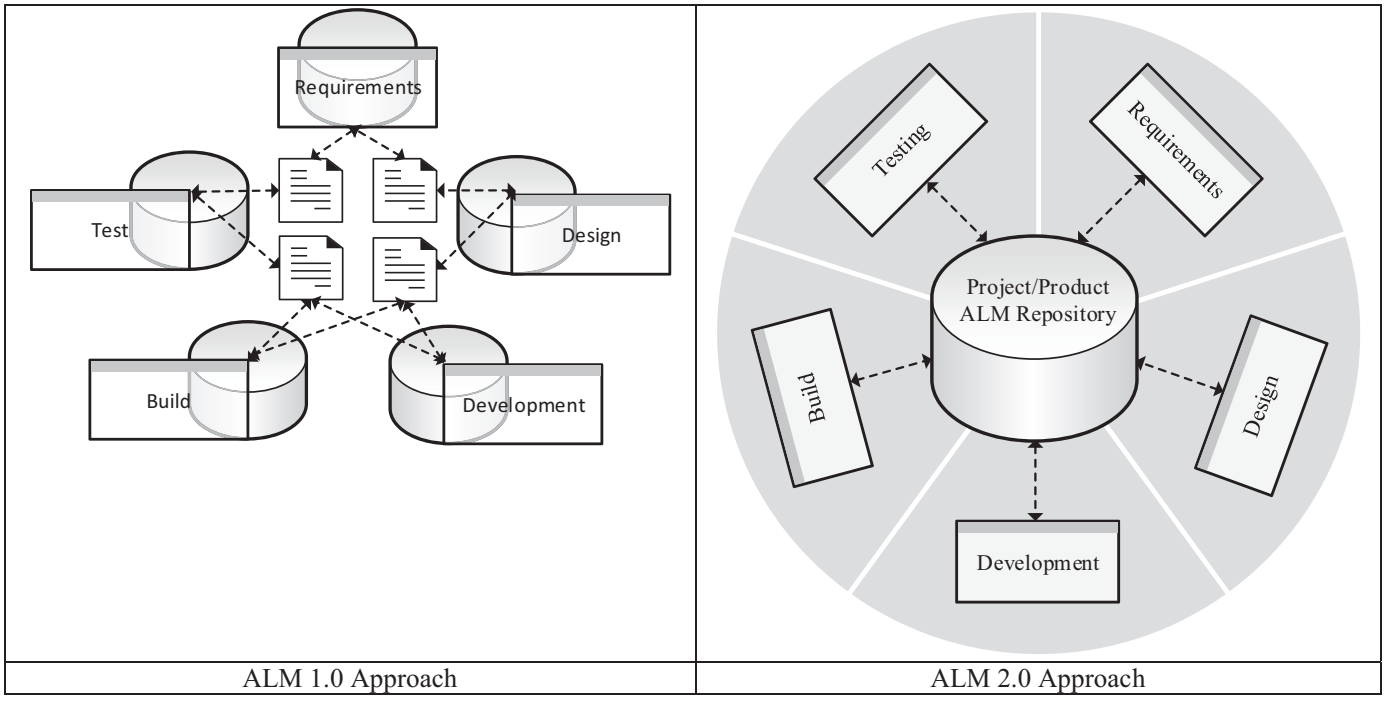

Fig. 2. ALM 1.0 vs ALM 2.0 approach (Schwaber, 2006).

search appears to be an important and valid instrument for evaluating the impact of ALM within an industrial context. Action research is an empirical research methodology whereby researchers attempt to solve a real-world problem while simultaneously studying the experience of solving the problem. Action Research has been widely used as a research approach in social science, and it has been adopted for information systems and software engineering research during the last two decades. According to Baskerville (Baskerville, 1999), action research studies share the four common characteristics: (1) An action and change orientation (2) A problem focus (3) An "organic" process involving systematic and sometimes iterative stages (4) Collaboration among participants. In general, action research consists of five basic steps as shown in Fig. 3:

Diagnosis: This phase corresponds to the identification of primary problems triggering the desire for a change in an organization. In this stage, theoretical assumptions about the nature of the organization and its problem domain are formulated.

Action planning: This is the phase where you plan the actions to address the problems that are identified in the diagnosing phase. In this phase, the desired future state is formulated and the actions to achieve this desired state are listed.

Action taking: This is the phase where the actions that are planned in the action planning phase are executed.

Evaluating: In this phase, the evaluation of the action taking is conducted. Here the researchers evaluate whether the theoretical effects of the actions are realized or not.

Table 1
Software development environment summary of the action research organization.

Number of Projects	50+
Engineers	800
Project sizes	5–200 persons
Programming Languages	C, C++, C# Java, ABAP
OS	Windows, Linux, OS X
Development Process	Waterfall (Large size contract projects) Agile (Non-contract projects)

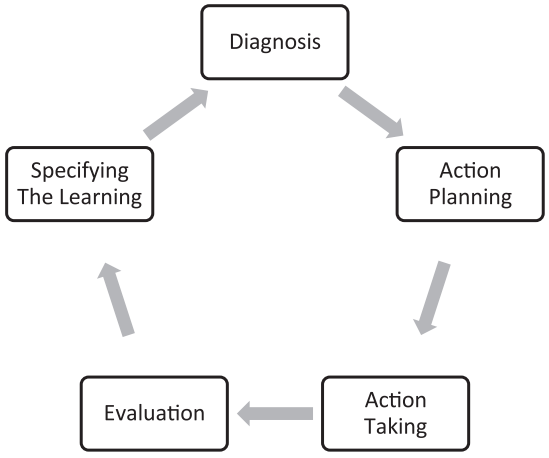

Fig. 3. Typical Action Research Cycle (Baskerville, 1999).

Specify the learning: Identification of what has been learnt from the cycle, regardless of implementation having been successful or not. These learning outcomes will be used to decide how to proceed for possible further cycles.

Very often action research is carried out as one complete cycle of the above activities. However, due to the unpredictable nature of the action research subject, the action research cycle can be iterated several times. Typically, action research is conducted by a team including the participants of the target system, members of the project group that is initiating the change, and the external researchers.

3. Context and motivation

The action research on the impact of ALM adoption has been carried out in a large-scale IT company, HAVELSAN, a Turkish software and systems company having business presence in defense and IT sectors. In the last decade, the company has rapidly grown in size to more than thirteen hundred personnel and increased its revenue tenfold to 250 million USD per annum (11). The company operates in three main business areas including command and control (C2), simulation and training systems, and e-government systems addressed by separate business divisions serving various customer segments.

The company has a diverse software development project portfolio. As shown in Table 1, at any given time, around 50 different software development projects may take place at a given time. These projects usually vary in size, the adopted operating systems, the adopted programming languages, and the development process.

To manage the overall activities in the organization, the company has put much attention and effort in adopting well-defined product development processes. The company was rather successful in this context and has been certified as Capability Maturity Model Integration (CMMI) 3 at the beginning of the action research. Because of this, the adopted development processes were well-documented and applied consistently. The processes were largely carried out manually without explicit process automation. These tools were not managed centrally, nor did any enterprise level installations exist. The required tasks were sometimes even carried out with office tools, such as word processors, and spreadsheets.

Due to the complexity and the size of the projects, it became more and more difficult to manage the overall process activities. The execution of manual processes was time consuming, error prone and difficult to maintain in the long term. Moreover, in the lifecycle process multiple different tools were adopted for the various activities. Integrating the artefacts developed during the lifecycle was mostly carried out manually. This integration has been largely handled by individuals, using import/export capabilities of the respective tools, or simply using office tools. For small scale projects, this could be managed to some extent, however, soon it was realized that the overall approach did not scale for larger projects. Tool integration problems substantially impeded the development activities and caused lots of effort and difficulties within and across the development teams. Even for the same activities sometimes different tools needed to be used that did not align properly and hence the integration had to be done manually. Achieving the tool integration could usually not be done in one step but often required multiple iterations.

Overall it was observed that the current approach and the attempted solution approaches were not feasible and maintainable for the long term. To support consistency among the separate process activities, process integration was needed that supported traceability and visibility of the developed artefacts. Hence, the company decided to find a systematic and sustainable solution for the identified problems.

It appeared that integrated ALM tackles the similar problems as we have identified within the context of the company. However, the transitioning to ALM was an important and radical decision which would have a large impact on the current process, tools and overall management. To evaluate the impact of ALM and introduce it carefully to the company context, it was decided to apply an empirical evaluation approach. To this end, action research has been selected as an evaluation approach since it explicitly targets the change within an organization. In the following sections, we present the results of the action research that was conducted within the company.

4. First action research cycle–ALM 1.0

In order to conduct action research, we followed the participatory action research protocol as described by Baskerville et al. (Baskerville, 1999). The main characteristic of the participatory action research is the active involvement of the practitioners as both subjects and co-researchers. Similarly, our action research team consisted of three internal researchers and one external researcher. The internal researchers were part of the developer productivity tools team in the Information Technology department of the company. They were actively involved in the project and worked from the inception phase to the maintenance phase of the project. The external researcher was familiar with the organization and had earlier worked in various projects within the company. The col-

laboration with the other members in the company (managers and developers) was carried out in selected pilot projects. During the whole process, we provided regular progress reports to the upper management.

To explicitly describe the objectives of the action research and guide its steps we have defined the following research questions:

RQ1: What are the current problems of the organization before the ALM adoption?
RQ2: What is the most feasible ALM platform for the organization?
RQ3: What are the feasible approaches to adopt and integrate ALM in the organization?
RQ4: What are the benefits of adopting ALM?
RQ5: What are the lessons learned in adopting ALM?

Note that these research questions largely align with the steps of the action research. Research question RQ1 will be primarily addressed in the diagnosing step. Research questions RQ2 and RQ3 will be primarily addressed in the action planning and action taking step. Research question RQ4 will be addressed in the evaluation, and finally research question RQ5 is discussed in the specification of the learning. In the following subsections, we describe this first action research cycle in detail and discuss the answers to these research questions.

4.1. Diagnosing

Before starting the adoption of ALM, the company was already aware of integration problems with the current approach. However, these were not explicitly discussed and described before. With the action research, we decided to explore the problems in a systematic and detailed manner. Hereby, an important aspect of the diagnosis was the decision on the scope of the adoption of ALM. At the time of the diagnosis, the concept of ALM in the organization was largely based on the integration of software development lifecycle activities, and actually did not consider governance and operation aspects. The reason for this was that at that time most of the ALM literature that we studied also seemed to have this view. As such, based on the identified problems it was decided to include all the following software development lifecycle activities in the ALM adoption: requirements analysis and document generation, design and document generation, development and unit tests, continuous integration and build, static code analysis, runtime analysis, automated tests, configuration management, test management and knowledge sharing. It should be noted that this focused view on integration of SDLC of ALM was later enhanced with governance and operations during the action research study.

The diagnosis was carried out by considering the following perspectives and the related questions:

Organizational perspective – To which extent does the organization support or impede the adoption of ALM? In case of the decision for ALM what are the necessary required adaptations?
Process perspective – What are the existing applied software development processes in the organization? What are the current needs and problems with respect to the adopted process?
Tool perspective – What are the currently applied tools? What are the existing problems with respect to the adoption of the tool for the corresponding software lifecycle activity (e.g. requirements, testing, etc.)? What are the problems with respect to the integration with other tools?

To perform a proper diagnosis, we first identified the key personnel (from senior engineers to managers) from five different divisions. Subsequently, starting by the beginning of 2010, we planned regular structured meetings with the representatives of these five divisions. In these meetings selected key persons from different projects were invited to hold presentations on their experiences regarding the above three perspectives. Based on the input of these workshops, at the end of the series of meetings, the overall diagnosis report was written including the current practices and obstacles in the organization regarding the need and readiness for ALM. The results of the meetings and the diagnosis from the above three perspectives were the following:

4.1.1. Organizational perspective

One of the key issues was also the lack of alignment with the required organizational structure. Formerly, the organizational structure of the company was based on project-based organization structure in which each employee worked for a single project and reported to a single manager. Each project adopted in principle their own software development process and the corresponding tools. As such the number and diversity of the adopted tools was quite large. Later, the organization decided to adopt a matrix structure consisting of functional and project dimensions to better align the activities.

Our diagnosis activity resulted in several important observations. First of all, we could conclude that the matrix organization structure aligned with the ALM philosophy of better integrating the project activities. The matrix organization structure inherently supported the reporting and as such the collaboration and integration of the activities. On the other hand, due to the recent transition to the matrix organization the employees had still the culture of the former project-based organization structure. Further, since the matrix structure itself does not directly impose the adoption of a single tool and process, the transition to ALM was still an important challenge. Many employees still tended to proceed with using their own process and tools, which severely hampered the goal for a companywide ALM integration.

4.1.2. Process perspective

The company has completed many large-scale projects in the last decade which adopted a wide range of software development processes. Most of the projects used a waterfall lifecycle approach, while some projects also used agile development processes. In this context, it should be noted that the company had a CMMI 3 level certificate since 2003 and has a strict process discipline. The CMMI practices are enforced irrespective of the software development process.

The diagnosis activity for the process perspective led to the following observations. First of all, the application of different kind of processes within the organization led to several problems. The integration of projects that adopted plan-driven processes (Boehm and Turner, 2004) with projects that adopted agile processes was a serious problem both from the process activity level and the timing and planning of the activities. The identified problems were in fact also different for plan-driven and agile processes. Plan-driven software development process relies on the execution of a strict set of process rules and heavy documentation. This seriously impeded the integration of the various processes within the company. Moreover, due to the adoption of different processes the traceability between the process artefacts was a serious problem. For projects that adopted agile software development the process integration efforts were a bit easier because of the agile philosophy that relies less on strict rules and documentation. The traceability problem however was perceived irrespective of the process adopted for development.

4.1.3. Tool perspective

Each project adopted its own process and its own tools which were too often different from the adopted tools of other projects.

Table 2
Development tools.

Lifecycle Activity	# of different tools in use	Tool names
Requirements Management	6	IBM DOORS, MS Word, MS Excel, MS TFS, IBM Requisite Pro, Atlassian Jira
Design Management	2	Enterprise Architect, IBM Rational Rose
Integrated Development Environments	4	Eclipse, NetBeans, MS Visual Studio
Test Management	2	MS Excel, Inhouse applications
Configuration Management	4	SVN, CVS, MS TFS, Fileshare
Change Management	6	Bugzilla, Jira, MS TFS, MS Word, MS Excel, IBM ClearQuest
Build Management	5	Cruise Control, Jenkins, MS TFS, Team City, Manual build scripts
Knowledge Management	3	MS Sharepoint, MS Word, MS Excel
Project Management	4	MS Project, MS Outlook, MS Excel, MS TFS

Table 2 shows the number of adopted tools for each lifecycle activity. The data for this has been retrieved from the projects by active questioning of the corresponding project managers. As we can observe from the table even within the same lifecycle activity different tools have been used. This large diversity of the adopted toolset substantially impeded the central administration and maintenance of the tools which were left to the responsibility of the corresponding projects. This led to a serious waste of effort and time of the personnel.

In fact, tool integration was very important in the organization, and was required both within and across lifecycle activities. For example, it was required that the tools of the requirements management were integrated to provide an integrated view on the overall requirements. On the other hand, tool integration between, for example, requirements management and test management were also demanded to view, trace, import, and export the artefacts produced within different tools. Unfortunately, the wide diversity of the adopted tools severely impeded these necessary goals. Adding a new tool to the company very often required its alignment and integration with other tools. Due to the high diversity, the required number of integrations grew almost exponentially.

Besides of technical aspects for tool integration we could also identify important managerial aspects. The costs for acquisition of tools, the required training for using the tools, and the maintenance of the tools became an increasing unmanageable problem within the company.

4.2. Action planning

The diagnosis activity provided important insights for the company of the current situation and the decisions that needed to be taken for avoiding further problems in the future. To cope with the raised issues in the diagnosis activity the executive board initiated the ALM transformation project in May 2010. In a series of meetings, the division managers have been informed of the upcoming transformation project. Based on their directives, and after the adjustments in the scope, the official start of the project has been held in July 2010.

Within the action planning, it was decided to consider the problems of the organizational, process and tool perspectives as identified in the diagnosis process. From the organizational perspective, we had observed that adopting the matrix organization supported the alignment of the processes and the tools. The main problem was the legacy culture of the individual projects that impeded the adoption of ALM. To cope with these problems, it was decided to start a series of meetings and training on adopting ALM. Within these meetings, the company would also define the incentives to convince the project members for the smooth transitioning to the new situation. Further, meetings were planned to increase the knowledge sharing among the project members, and within the firm.

The action planning for the tooling and the process perspective were defined together. To cope with process diversity among projects it was decided to align the lifecycle activities of the adopted different development processes. The major focus of the alignment was put on the selection and adoption of the tools since these had a direct impact on both the organizational and process perspectives. First, it was decided to acquire the best of breed development tools for supporting the integration of lifecycle activities. An important requirement for the selection of tools was also that it had to support the CMMI activities which was strictly followed within the company. Further, if possible tools would be preferred that were already in use within the company. For selecting the tools, a designated number of employees would be made responsible for the installation and maintenance of the tools. The developers who were formerly responsible for the management of the tools, would then be relieved from these activities. Likewise, they could reserve more time for other important activities and herewith the overall productivity would be increased. Further, since the focus would be on selecting and aligning the similar set of tools for the various lifecycle activities the diversity of the tools would be reduced. Hence, the total cost of ownership for the development platform, including hardware, software, the training of the personnel, and maintenance costs would also be substantially reduced.

To analyze the tools and select the best tool combination, an *evaluation committee* has been formed in September 2010. The goal of the evaluation committee is to gather and compile the user needs, lead the evaluation and acquiring of the tools according to the user needs. The evaluation committee consists of the *transition committee* and *user representatives* from various divisions in the organization. The *transition committee* consisted of the Action Research team members, whereas the *user representatives* were selected from each division per their expertise in disciplines.

The evaluation committee would undertake the following steps as shown in Fig. 4:

Analysis: The evaluation committee would analyze the customer needs for each lifecycle activity, and publish customer needs documents. The customer needs would include required set of features for the tools and the use cases for each lifecycle activity.

Assessment: Based on the customer needs, the evaluation committee would assess and evaluate development tools, and identify the best of breed tools per each lifecycle activity. A tender would be placed to acquire selected products, training and integration services. The customer needs document would be used to develop bid document.

Operation: The transition committee would work with the bid winner to customize the tools for the company, and disseminate the tool set in the enterprise.

4.3. Action taking

After defining the action plan, we started executing the identified activities in Fig. 4.

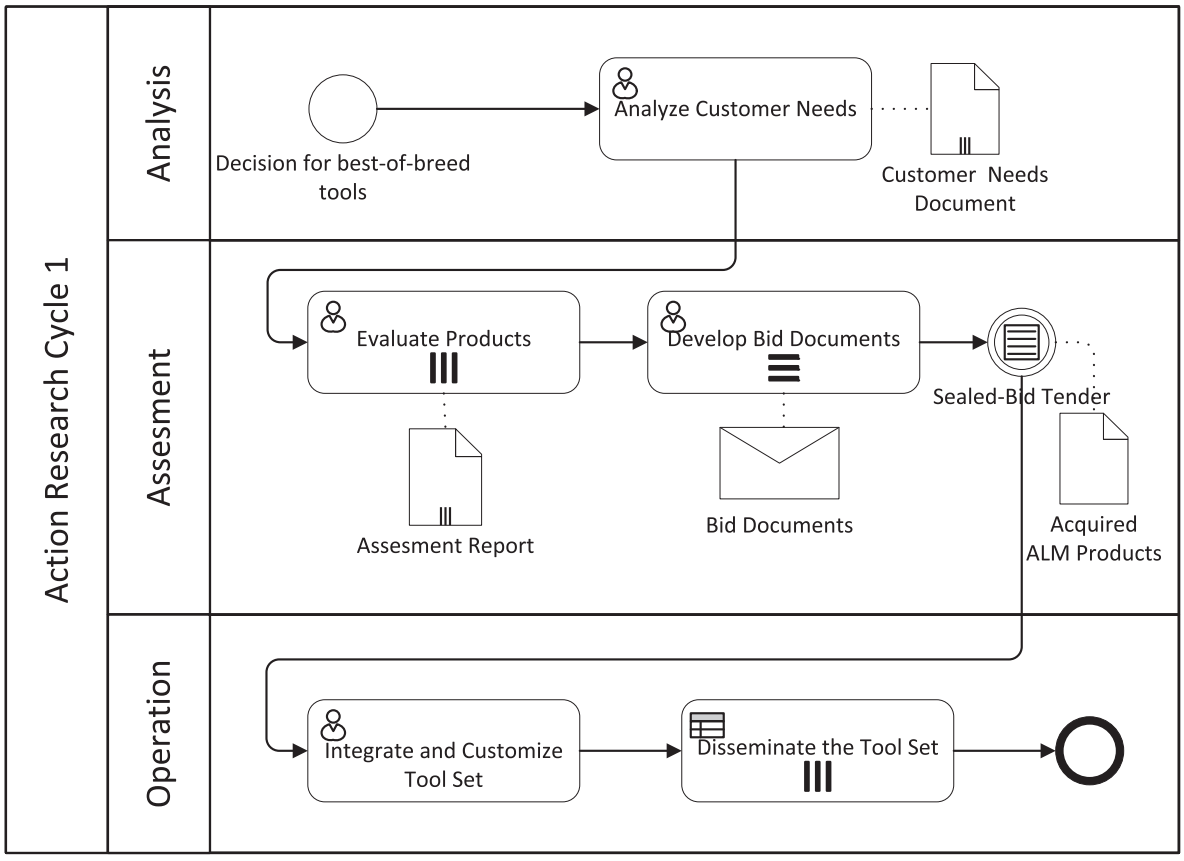

Fig. 4. Activity diagram for the first Action Research Cycle representing the action planning.

Table 3
Example of integration requirements between lifecycle activities (checked cells define required integration).

	Req.	Design	Source Control	Build	Test	Change Manag.
Req.		✓	✓	✓	✓	✓
Design	✓		✓		✓	✓
Source Control	✓	✓		✓	✓	✓
Build	✓		✓		✓	✓
Test	✓	✓	✓	✓		✓
Change Management	✓	✓	✓	✓	✓	

4.3.1. Analyze customer needs

The evaluation committee has analyzed the customer needs for the different lifecycle activities resulting in a customer needs document. The document explicitly described the dependencies with other lifecycle activities and the desired features for integration. Based on this information, the evaluation committee has further developed user scenarios for each lifecycle activity. These scenarios were represented as UML activity diagrams, and the corresponding step by step instructions were described to demonstrate the features.

Table 3 shows an example of the required integration between the tools that support the various lifecycle activities. The white cells define the relations that do not require integration. The checked cells have been indicated as integration requirements. As we can observe from the table we can identify many different integration requirements which will put important challenges for the tools to be adopted.

4.3.2. Evaluate development tools and identify best of breed products

Based on the identified scenarios in the previous step, the evaluation committee evaluated well-known tools in the software development industry, and those that were already in use within the company. The evaluation would look at features that would support the required capabilities for the lifecycle activities, but also

explicitly consider the integration capability with tools of the other lifecycle activities. The evaluation has been carried out from December 2010 to February 2011 and included in total 14 evaluation sessions with 34 unique participants. The evaluators were asked to rate each customer need on a numerical scale ranging from 1 to 10. Hereby, the score 0 implied that the given feature was not implemented by the tool, whereas 10 meant the complete implementation of the feature. All the scores were accompanied with the corresponding rationale of the evaluation. After the rating process, consolidations of the data and statistical analysis were carried out. Hereby, we have applied the Chebyshev's inequality to distributions that are 2 standard deviations far from the mean. The committee asked individuals for a review of the ratings that were too far from the mean. After the reviews were completed, the score for a product has been calculated from the average of the ratings and the weighted average of the feature/need. At the end of the evaluation, three products for each lifecycle activity were selected that seemed to be most feasible. All these selected products would form a short list for the bid stage. At this stage, it appeared that all the products in the short list largely implemented the required features for the lifecycle activities. However, the required integration capabilities as defined in Table 3, were still considered limited. The identified open integration problems would be addressed later when issuing the tenders in which the additional needs for the tool and the required integration capability would be discussed. Since the company had selected the best of breed tools this was the most feasible action that could be taken then.

4.3.3. Develop bid document

After the evaluation and creation of the short list for the products, we had to prepare the bid document that would be send to the vendors. For this it was necessary to identify the required number of licenses for each tool, together with the number of the personnel that required training. This information was extracted by the transformation team which interacted with the project teams within the company. The transition committee designed and conducted a survey to reveal the required approximate number of licenses and training sessions. All the information was collected, and a final bid document was completed. The bid document explicitly included also the required features and the remaining open integration problems.

4.3.4. Call for tender

A call for tender was issued to acquire the products that met the requirements of the bid document, and which was economically feasible. The tender has been placed in November 2011. The distributors/resellers of the selected products were invited to the tender. The evaluations for the offers took place during December 2011. Initially, many vendors indicated that they could not realize the integration requirements satisfactorily and therefore they withdrew from the tender.

The comparison of the different offers was not straightforward due to the different license requirements, and offered maintenance services. The transition committee asked for explanations during the evaluations, and compared the products for fitness to the bid document. Unfortunately, none of the vendors seemed to have a completely satisfactory solution, and also did not give a guarantee to provide solutions for the indicated integration problems. It became clear that the integration problems would in the end not be solved and only a brittle solution would be provided. The company would actually face the similar earlier integration problems but with very high service costs that had to be continuously paid to the tool vendors for patching the tool integration problems. Altogether at the end of the bidding process, the company decided to cancel the tender altogether. A different action had to be taken.

4.4. Evaluating

In the action research methodology, the evaluating step discusses the evaluation of the adopted approach. However, the cancellation of the tender led to the termination of the adoption of the ALM-based integration.

4.5. Specify the learning

Despite the premature ending of the first action research cycle, the whole process provided important insights and lessons learned. The action research helped to gather the ALM requirements and had a better understanding of the software development practices in the company. The team had acquired a considerable amount of knowledge about ALM concepts together with the best of breed tools for each process area.

The diagnosis showed that the company had to cope with a serious problem regarding the integration and alignment of the adopted processes and tools. The matrix structure organization helped to support the alignment to a limited extent. The diversity of the tools and the processes hampered the understandability, the communication, the analysis and as such the overall productivity. The diagnosis activity was very useful to highlight all the important problems and convince the managers and developers that concrete action had to be taken to solve the current problems and avoid future problems. During the action planning a systematic approach was adopted to plan the activities for tackling the integration problems from the organizational, process and tool perspectives. The plan looked promising and the company was confident that solutions to the problems would be provided. However, during the action taking step important concerns appeared that were not foreseen beforehand. First of all, the specification of the customer needs showed the complex requirements for the integration between the tools that supported the lifecycle activities. It became clear that the problems were indeed very challenging. Potential tools were identified that could meet the required features of the lifecycle activities, and at the same time support the integration. The bid document summarized the important needs as well as the required solutions for the integration problems. It was interesting that during the tender several vendors directly withdrew since they did not foresee a solution to our identified problems. The vendors that remained offered only brittle solutions for prices that was economically neither feasible nor sustainable. In summary, adopting a solution that used the best of breed tool from each category will fail due to the expense of integrating the solutions, thus integration of the tools was a must-have for the final solution. This was really an important insight which at that time was also not broadly discussed in the literature.

5. Second action research cycle–ALM 2.0

From the first action research cycle, we concluded that the ALM 1.0 approach would not work out in the end to provide a sustainable solution for the identified problems. Still, for the company it was business critical to provide satisfactory solutions to these problems. Although the first action research study did not provide tangible solutions, the company decided to continue the research activities and the efforts to find a feasible solution. As stated before, the first action research cycle provided a thorough insight into the current state of the organization and the adopted processes and tools. Further during the first action research cycle novel knowledge and insight was gained about the state-of-the-art developments in ALM research. An important aspect was the encounter with the concept of ALM 2.0 that provided a substantially different approach to the process and tool integration problems. Hence, it was decided to investigate the adoption of ALM

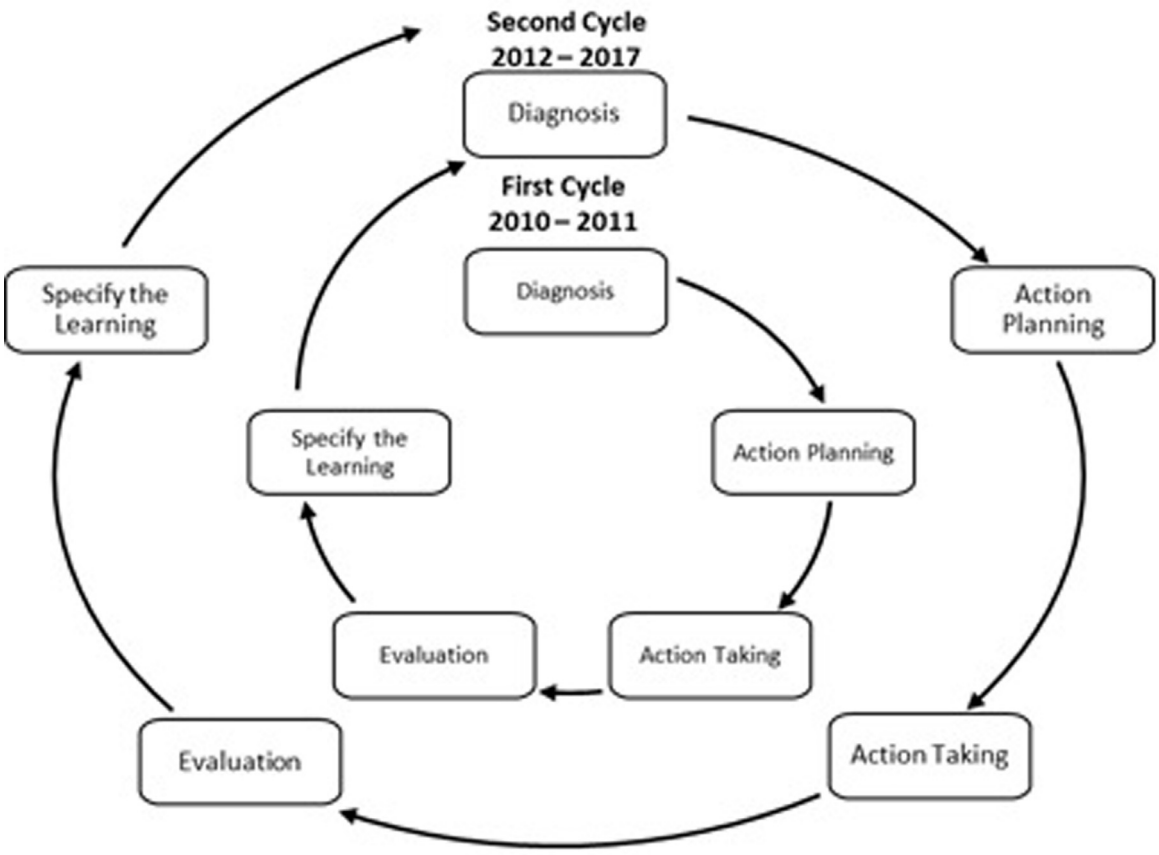

Fig. 5. Sequential Action Research Cycles.

2.0, which from the reports in the literature, as discussed in the background section, seemed to provide a more feasible solution. This was the reason for starting the second action research cycle which would research the application of ALM 2.0. The first action research cycle covered the period from 2010 to 2011, which was directly followed by the second cycle that covered the period from 2012 to early 2017. The phase *Specify the Learning* of the first action research cycle appeared to gradually transition and overlap with the first phase *Diagnosis* of the second action research cycle. The overall action research in the end consisted of the continuous integration of two action research cycles as shown in Fig. 5. In the following sub-sections, we will discuss each phase of the second action research cycle.

5.1. Diagnosing

The diagnosis step of the second action research cycle focused on analyzing the current state of the company with respect to readiness for ALM 2.0. The diagnosis built on the lessons learned of the first action research cycle. Similar to the first action research cycle, we analyzed this again from the organization, process and tool perspective. From the organizational perspective, there did not seem to be a large difference between the adoption of ALM 1.0 and ALM 2.0. The higher-level goals of both the paradigms are in fact similar; a smooth integration and alignment of the processes and tools. The adopted matrix organization structure seemed to also benefit the ALM 2.0. When starting the second action research cycle however, the organization was in a much different experience and knowledge state regarding ALM. Both the management and developers were now already convinced about the identified problems and also became aware of the limitations of ALM 1.0. This was a very positive trigger and support for initiating the efforts for adopting ALM 2.0.

With respect to the process and tool perspective the condition was not changed; there were still a diversity of the adoption of the processes and tools. This could not be changed due to the different needs for coping with different client expectations and requirements. Since the tender for ALM 1.0 tools was terminated the situation as such was similar to the initial situation of the first action research cycle.

5.2. Action planning

The action planning phase of the second action research cycle was defined using the lessons learned in the first action research cycle. The focus on the selection of best of breed tools was not considered anymore since it did not provide a sustainable solution. To provide a feasible solution to the identified problems it was now decided to adopt a unified platform tool as proposed by ALM 2.0. The software process activities would then not be fragmented over different tools but be integrated on the same platform. Likewise, this would lead to a natural integration of the process activities and the corresponding tools that support these activities. Similar to the first action research cycle the *evaluation committee* followed the subsequent steps to define the action plan steps. The activity diagram that includes the steps for adopting ALM 2.0 is shown in Fig. 6.

In the following we describe these steps in more detail.

Analysis: The evaluation committee would start the analysis of the ALM adoption approach of peer companies which are part of the same holding and sector in Turkey. The reason for this was to enhance the lessons learned further, by also learning from the

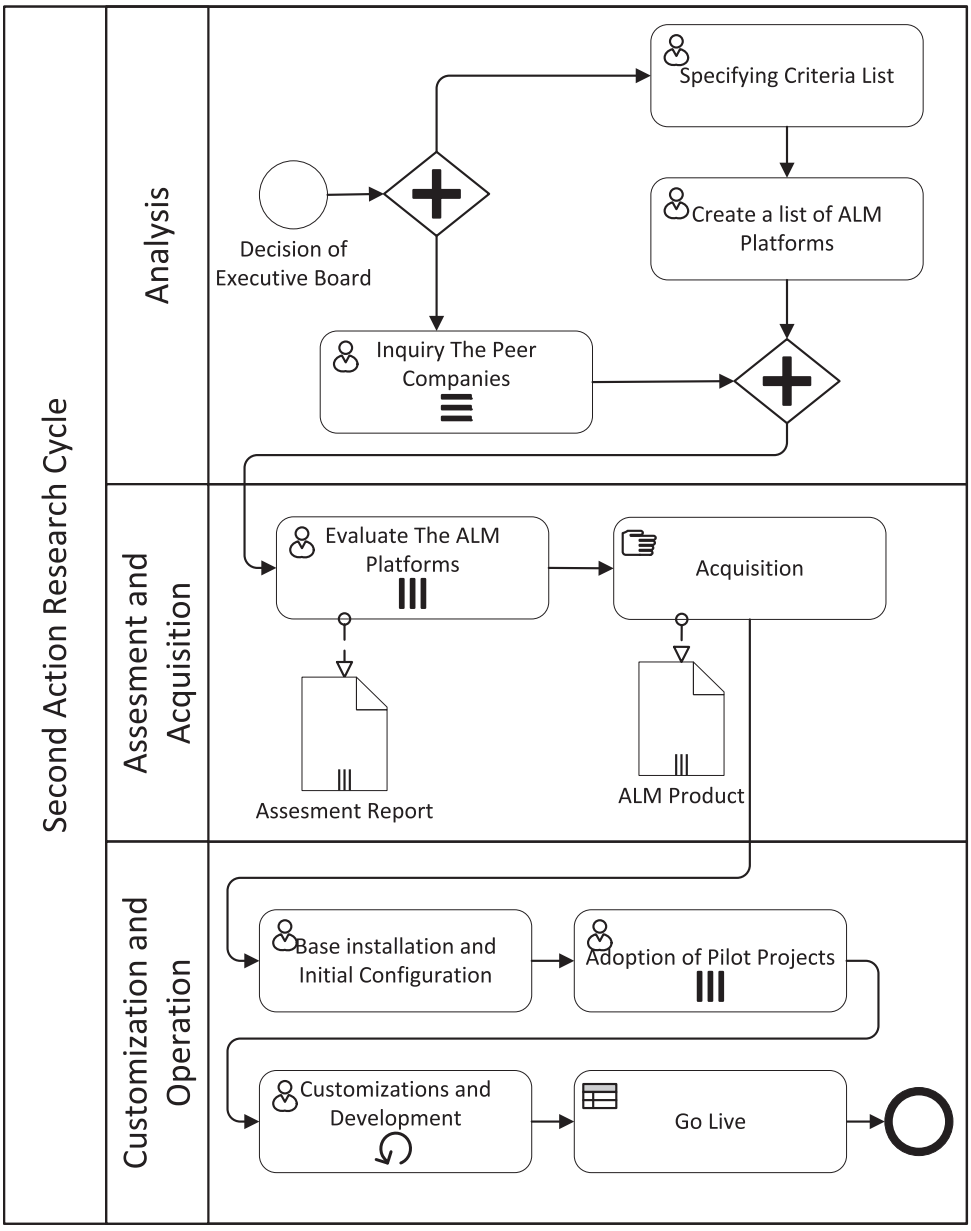

Fig. 6. Activity diagram for second action research cycle.

experiences of other companies. In parallel to this, the state-of-the-literature on current ALM 2.0 platforms would be analyzed to get more insight on the experiences with ALM 2.0. These platforms would then later be evaluated for which the corresponding criteria needed to be selected. For this the earlier defined criteria in the first action research cycle would be taken as a basis and further enhanced for evaluating ALM 2.0 platforms. To sum up, the output of the analysis phase would be the list of ALM 2.0 platforms together with the criteria for evaluating these platforms.

Assessment and Acquisition: In this phase, the actual evaluation of the identified platforms would take place based on the predefined criteria. Thus, one platform would be selected and acquired. Note that in this cycle the company did not decide to issue a tender again. This was because the cost of adopting the platforms was explicitly considered in the criteria and as such an early and independent evaluation could be made from this perspective. Furthermore, the lessons learned in the first research cycle had already provided a broad insight in the existing ALM platforms and the important features. This substantially helped to provide a faster and more precise evaluation of the ALM platforms. As a result of this phase one ALM 2.0 platform would be selected and acquired.

Customization and Operation: Different from the first action research cycle, this time, the transition committee would be responsible for the installation, customization, operation and dissemination of the selected ALM 2.0 platform. The installation would be carried out gradually and start first with the installation of the base version. In parallel, the relevant stakeholders would be identified and their participation actively supported and guided. To support the acceptance and dissemination, the ALM platform services would be installed on a private cloud. In general, the base version of the platform would not be sufficient and additional customization would be necessary to meet the company's specific requirements. For this reason, selected set of representative pilot projects would be started to identify the precise demands and the customization requirements. To guide the pilot projects explicit help and documentation would be constantly provided to miti-

gate unnecessary resistance for adopting ALM 2.0. Each of the pilot projects would indicate the required customizations to the platform. This whole process would be carried out in an incremental and iterative manner until no additional customizations were needed. As a result of the process it would be decided to go live throughout the company. This would mean that all the oncoming projects would then be developed and operated on the ALM 2.0 platform. Further, plans would be made for transitioning also the existing projects.

5.3. Action taking

We have conducted a series of steps to achieve our action plan. The main steps are outlined below.

5.3.1. Analysis

In the analysis step, we carried out the search for ALM platforms. For this we looked in parallel to both the literature (state-of-the-art) and the tools adopted in the state-of-the-practice. The company is in close contact with peer companies that work in the same domain and which had similar kind of problems and approaches. To share the common knowledge and experiences, we organized a set of meetings with a selected set of the peer companies in Turkey. The focus of these visits was mainly to share ideas about the process and tool integration problems and the adoption of ALM platforms. We visited in January 2012, in total three companies which can be characterized as large-scale companies consisting of thousands of employees. It appeared indeed that these companies had to cope with similar problems as that of the company. These problems were more or less the problems that we discussed in the first action research cycle and the diagnosis part of the second action research cycle. Furthermore, all of these companies also had adopted or made plans to adopt the best of the breed tools, that is, ALM 1.0 implementations. For all these companies, the tool integration was not realized at the system level and the companies had still to cope with the identified problems of ALM 1.0 as discussed in the literature and as experienced in our first action research cycle. It should be noted that at the time of our visits ALM 2.0 was also recently introduced in the literature. Hence, the identified problems of HAVELSAN and the visited companies were also not solved yet in the state-of-the-art. In fact, our visits did not leave us any novel insight into our answers but the common knowledge sharing events helped us to confirm our conclusions and the continuation of our need to search for a sustainable solution. Further, thanks to our planned action research we observed that HAVELSAN had gained important insights that other companies had not acquired yet. Since HAVELSAN had started the ALM activities much earlier than the other companies, there were few novel lessons learned from the other companies. It should be noted however that visiting other companies and the novel lessons learned that could be shared will be dependent on the level of the other companies.

In our literature study, we extended our earlier study that we carried out during the first action research cycle. However, we now focused on ALM 2.0 platforms. To evaluate the presented ALM 2.0 platforms in the literature, we also explicitly searched and identified the criteria for evaluating these platforms. Our criteria search focused on both a thorough domain analysis to the related literature as well as interviews with the managers, developers and tool experts in the company. As a result, we identified a total of 29 evaluation criteria that we classified in five categories as shown in Table 4. Each criterion is related to the questions that is defined in the description column.

The category *ALM Process capability* defines the criteria for evaluating the capability of the platform for addressing the corresponding lifecycle activity. *Architectural Capabilities* criteria are used to check the architectural styles and approaches. *Extension Capabilities* define to which extend the platform can be extended for additional customizations that are needed for the company. *Licensing policy* criteria consider the licensing aspects. Finally, *Customizations as a Marketable Product* criteria are used to evaluate the sustainability of the platform.

After the identification of the evaluation criteria we also investigated the existing ALM 2.0 platforms. For this we did not only consider commercial platforms (such as Atlassian Suite, HP ALM, IBM ALM, codeBeamer, MKS Integrity, Polarion and Microsoft TFS) but also looked at open source software ALM platforms (such as Endeavour, Topcased and Tuleap) and ALM integration frameworks (such as Tasktop). As a result, we compiled a list of 28 ALM 2.0 platforms as shown in Table 5.

5.3.2. Assessment and acquisition

An assessment group consisting of three researchers in the action research cycle has been formed to evaluate the compiled list of ALM platforms based on the identified criteria. The assessment group worked for 2 months for completing the assessment of the 28 platforms of Table 5. Hereby, each of the assessment group member has evaluated 8 to 10 platforms.

For the assessment, a questionnaire was prepared including questions for rating the selected platforms. In parallel, the weighting factor for each criterion was defined with respect to the concerns of the various stakeholders including both the development teams and the management. Each ALM platform was assessed in detail giving sufficient time for the assessment group. The assessment was able to provide detailed feedback for each criterion and the rated platform. Hereby, the assessment was realized by direct installation on local servers or using online cloud versions. In addition to these steps the assessment was further supported by following the assessments and insights of others as presented in video tutorials. Since the ALM platforms were different with respect to the required customization effort, the provided assessments were normalized with respect to the total cost for customization and ownership.

Based on the results of the assessment three alternatives were selected for detailed analysis before committing to a platform. Each of these three alternatives meet a large percentage of the demanded features and are largely cost-effective. Despite of the high ranking none of the three alternatives completely satisfied all the requirements of the company. Based on the criteria we provided in Table 4, we have selected Microsoft (MS) Team Foundation Server (TFS) 2012 RTM as our integrated ALM platform choice because of the highest score among alternatives. Since design management and knowledge management activities were not explicitly supported by the MS TFS, the Sparx System Enterprise Architect, and MS Sharepoint Portal 2010 have been included to support these activities as well. The latter tools were already in use within the company and as such could be easily adopted.

The acquisition of the MS TFS went rather smoothly because the company had already an ongoing volume licensing agreement with Microsoft. MS TFS 2012 RTM, and MS Sharepoint 2010 licenses were added to corporate level agreement and purchased as regular MS products. The licenses for Sparx System Enterprise Architect have been purchased from a local representative.

5.3.3. Customization and operation

The MS TFS appeared to meet most of the ALM requirements as defined by the company. As stated before none of the ALM platforms completely satisfied all the ALM features as required by the company. For customization and operation of the ALM platform we applied the following four steps.

Table 4
Selected criteria for evaluating ALM 2.0 platforms.

Evaluation Category	Criteria	Description
ALM Process Capability	ALM completeness	For each criterion, what is the capability of the platform for addressing the corresponding lifecycle activity?
	Project management	
	Requirements management	
	Design management	
	Integrated development environment	
	Test management	
	Software configuration management	
	Change management	
	Continuous integration	
	Knowledge management	
Architectural Capabilities	Service orientation	Does the platform provide support for SOA?
	Cross-platform Client Support	Does the platform provide client side support for different operating systems and platforms? Which platforms?
	Database requirements	What are the requirements for the adopted databases?
	Web browser support	Which web browsers are supported?
	Add-on support	Are there any add-on repositories for the platform? How rich are the add-on repositories in terms of number of add-ons and add-on features?
	Open community	Are the any open communities for the platform?
Extension Capabilities	Server extensibility language	What is the main language for the platform on the server side? What is the language for server side extensions?
	Client extensibility language	What is the language for the client side development?
	Web extensibility language	What is the language for the web development?
	Ease of developing server extensions	What is the main extensibility framework? Does it support SOA or developer's API?
	Ease of developing client extensions	Does client development support SOA or developer's API?
	Ease of developing web extensions	Does web development support SOA or developer's API?
Licensing Policy	License type	Is the platform licensed as open source or commercial?
	Licensing models supported	What licensing models does the platform support? Named user, floating user, per server, per processor, etc.
	License ownership	Who owns the purchased license? Is it based on subscription or perpetual?
	Assurance policy	What is the assurance policy for the platform? Will there be hotfixes along with regular updates? How often will the platform be updated?
	Need for additional framework or database licenses	Does the platform require additional frameworks, or databases?
Customizations as a Marketable Product	Sustainability in the company	Are the platform customizations sustainable with the existing resources in the organization?
	Market value of the customizations	Do the platform customizations have any market value?

Table 5
Selected ALM 2.0 platforms.

ALM Platform	License Type	Reference/Web Address
Atlassian Suite	Commercial	http://www.atlassian.com/
Axosoft OnTime11	Commercial	https://www.axosoft.com/
CollabNet	Commercial	http://www.collab.net/
Dynamsoft	Commercial	http://www.dynamsoft.com/
HP ALM	Commercial	http://www8.hp.com/us/en/software-solutions/application-lifecycle-management.html
IBM Jazz	Commercial	https://jazz.net/
Inflectra Spira	Commercial	https://www.inflectra.com/
Intland codeBeamer	Commercial	http://codebeamer.com/
KovAir	Commercial	http://www.kovair.com/
MKS Integrity	Commercial	http://www.ptc.com/application-lifecycle-management
Polarion	Commercial	https://www.polarion.com/
Rally	Commercial	https://www.rallydev.com/
Rommana	Commercial	http://www.rommanasoftware.com/
SeaPine	Commercial	http://www.seapine.com/
Serena	Commercial	http://www.serena.com/
SmartBear	Commercial	https://smartbear.com/
Tasktop	Commercial	https://www.tasktop.com/
TFS	Commercial	https://www.visualstudio.com/en-us/products/tfs-overview-vs.aspx
Vault	Commercial	http://www.sourcegear.com/vault/
VersionOne	Commercial	https://www.versionone.com/
Countersoft Gemini	Open source	https://www.countersoft.com/
EmForge	Open source	http://www.emforge.net/
Endeavour	Open source	http://endeavour-mgmt.sourceforge.net/
Jabox	Open source	http://www.jabox.org/
OSEE	Open source	https://eclipse.org/osee/
TIKAL	Open source	http://www.tikalk.com/
TopCased	Open source	https://www.polarsys.org/topcased
Tuleap	Open source	https://www.tuleap.org/

5.3.4. Base installation and initial configuration

The corporate level MS TFS base installation took place in June 2012. For this it was decided to install the platform on a private cloud to support the management of the installation and maintenance efforts. To individually support the various divisions in the company, each division was considered as a separate tenant of the MS TFS cloud services. The base installation included by default the MS CMMI Process Template, a process template developed by MS to support CMMI projects. Initially we adopted this template but soon it appeared that we had to enhance this to align it with the adopted process within HAVELSAN that includes also agile software development processes. TFS authentication would be handled through MS Active Directory service, and it was decided that, TFS authorizations would be based on MS Active Directory user groups also. Hereby, users would be member of at least one Active Directory Group based on their role in the projects.

5.3.5. Adoption of pilot projects

To adopt the ALM platform within the company, first, the new platform had to be shared with all the relevant stakeholders. For this a kick-off meeting was held in May 2012 to discuss the decision on the selected platform and the oncoming actions to be taken. In addition, the meeting aimed to gather additional concerns from the stakeholders that might be important for integrating the ALM platform. The meeting was attended by the transformation team, engineering division managers, and quality representatives.

All the stakeholder concerns were recorded including concerns related to process support, process alignment with the quality procedures, and service availability. To systematically plan the integration process, it was decided to adopt a pilot project approach. Technical divisions within the company were asked to use the ALM platform in at least one of their projects. As a result, five pilot projects have been identified; three divisions provided one pilot project, one division provided two pilot projects. The transition of pilot projects took place from June 2012 until September 2012.

The implementation of the pilot projects using the ALM platform included the creation of a team project, the migration of existing documents to team portal, migration of existing artifacts (requirements, source code, tasks, etc.) to the team project and the tool training (for each discipline). During the pilot project stage, the divisions were actively supported, and on-site help was provided directly when needed. In addition, we regularly asked for the feedback of the divisions and analyzed the progress of the implementation. As such, the assistance was taken seriously, foremost to support the integration and mitigate unnecessary risks that would lead to a failure of the ALM platform integration. After six months from the initial deployment, the divisions have started to request using the ALM platform for the projects beyond the selected pilot projects.

5.3.6. Customizations and development

The pilot projects provided important insight and feedback for the ALM integration. During the pilot projects, further customizations to the base installation of the ALM platform were continuously realized. Hereby, we used the mechanisms of the MS TFS for tailoring the platform to the company context. Most of the features that needed to be added could be indeed realized using the MS TFS customization mechanism. For the remaining part of the required features we had to provide add-on development beyond the MS TFS customization mechanisms. The major customizations and add-on development have been performed using the Scrum approach (Schwaber and Sutherland, 2011) started on 2012 June, using 14 iterations of four weeks each, thus in total 56 weeks. At the end of each iteration new features and customizations were deployed to the ALM platform. After these customizations, the ALM platform was almost stable and no important customizations or add-on development actions were required anymore. Since the required customizations were now less in frequency and the required overall effort, we decided to decrease the iteration length to two weeks instead of four weeks. After this decision 18 more iterations were realized in total requiring 36 weeks. In total the customization effort took thus 56 weeks plus 36 weeks, that is, 92 weeks. or 21 months. The customization team started initially with 3 persons, but this number eventually increased to 6 persons at the end of the first year. In total the task required around 90 man-months.

The major customizations have been made to support the CMMI processes. Further notable customizations included customizations to support task management, requirements engineering, change management, and test management. In total, we have customized 5 work items of the ALM platform, defined 6 new work items, defined more than 100 fields, and finally defined more than 500 workflow rules for the corresponding work items.

Major add-on developments were required to support CMMI processes and project management. Notable add-on developments included baseline development for requirements engineering, traceability, suspect analysis, user defined numbering, and working log.

Table 6
Usage statistics of the ALM platform.

General Usage	
# of Projects	44
# of Users	300
Total # of Work Items	87,456
Revisions of Work Items	471,560
Source Control Items	548,624
Revisions of Source Control Items	1,054,647
# of Builds	13.732

5.3.7. Going live

After the pilot projects and the realized iterations, the ALM platform was found ready to be used throughout the whole company. The success of the pilot projects was soon shared by the participants in the company. Surprisingly, this led to additional voluntary projects to be implemented on the ALM platform. As a result, after the initial 5 projects, an additional number of 39 projects were implemented on the ALM platform. Hereby more than 300 engineers of the company were involved.

To provide the final upper-level managerial decision for the company wide usage of the ALM platform, an executive board meeting has been held in February 2014. During the meeting, the experiences from the pilot projects together with the current state of the ALM platform was presented. During the meeting, Table 6 was presented to the executive board. Table 6 shows the overall statistics of the implementation of the projects.

The presented information and the overall experiences within the company were convincing for the executive board. Consequently, in the meeting it was decided that from then on, the newly starting projects had to be implemented using the ALM platform. This was a mandatory requirement and hence the projects had to use the ALM platform. For the ongoing earlier started projects, it was decided to provide a case-based assessment to provide the decision to whether to migrate these to the ALM platform or not.

5.4. Evaluating

Within the action research study, the next step includes the evaluation of the ALM application. This would provide an answer to research question RQ4 which considers the discussion of the benefits of the ALM application.

Table 7
Acquisition benefits.

Factor	Before ALM 2.0 application	After ALM 2.0 application
License and Hardware Costs	20 units	1 unit
Acquisition Activities	30 man months	1 man month
Installation Activities	45 man months	9 man months

Table 8
Operation benefits.

Factor	Before ALM 2.0 application	After ALM 2.0 application
Maintenance and Backup Operations	248 man months	60 man months
Helpdesk and Functional Support	495 man months	180 man months
Customization Effort	15 man months	4 man months

Table 9
Organization benefits.

Factor	Before ALM 2.0 application	After ALM 2.0 application
Additional # of Training Sessions	150	50
Workforce Loss because of Training	204 man months	15 man months

Table 10
Production benefits.

Factor	Before ALM 2.0 application	After ALM 2.0 application
Project Initiation Speed	3 months	3 hours
Traceability Management	57 man months	0 man months
Decision Support	75 man months	8 man months
Process Audit Time	18 man months	4 man months

The evaluation of benefits of the ALM 2.0 has been difficult because prior to the ALM transformation, software production and maintenance efforts could not be observed. In the literature, it is also very difficult to find this kind of information. In this case, the mathematical model has to be established in order to identify the values before and after this study. A mathematical model is defined based on the following assumptions: operating and maintenance costs/ benefits that have been recognized for the period of 5 years, the costs and benefits that vary with the number of projects was adopted for the existence of 50 projects at any given time which is typical for the company. We further assumed all the projects were of the similar size.

The overall benefits of applying ALM 2.0 were considered from the perspectives of Acquisition Costs, Operation, Organization and Production. The results are illustrated in Table 7, Table 8, Table 9, and Table 10. The tables provide the results before (second column) and after (third column) the application of the ALM 2.0 platform.

The data for the *Before ALM 2.0 Application* field were obtained from interviews with project managers and technical managers. According to these interviews, some projects have set up, adapted and operated their own customized ALM 1.0 tools. The license and educational values have been derived from accounting records and based on the statements of project managers and technical managers. The data for the *After ALM 2.0* were obtained from accounting records, help desk records, and by classifying training records issued by the ALM maintenance and operation team.

Table 7 shows the benefits from the Acquisition perspective and considers the factors *License and Hardware Costs, Acquisition Activities, and Installation Activities*. The License and Hardware costs have been defined as units and for confidentiality reasons the actual costs as money units have not been given. It appears that the application of ALM 2.0 the license and hardware costs have substantially reduced the license and hardware license costs. The same holds for acquisition and installation activities. These values have been carefully computed based on the company administration input and results. The main reason for a substantial reduction of these costs is because the adoption of ALM 2.0 was related to volume licensing which allowed to adopt cheaper license keys in case of large acquisition. Formerly, since each division had its own set of tools no profit could be acquired from the licensing for a larger scale of computers.

Table 8 shows the Operation benefits and includes the factors *Maintenance and Backup Operations*, and *Helpdesk and Functional Support*, and *Customization Effort*. Note that here the customization implies the individual customization and integration costs for an incoming project, and thus not the customization of the ALM platform.

Again, we can observe a substantial reduction of the costs. This is because of the central management that is characteristic to ALM 2.0. The maintenance and backup, the help desk and functional support, as well as the customization of the platform, are performed by one central IT department. Since the work tasks were not distributed and all handled by one entity, we could achieve economies of scale and likewise reduce the costs.

Table 9 shows the Organization benefits and includes the factors *Additional Training Costs*, and *Workforce Loss because of Training*. Before the adoption of ALM different type of tools were regularly acquired and the corresponding training needed to be organized. Training for the tools incurred an important cost including the costs of the training itself and the 'lost' time that could not be spend for the development activities. This situation fundamentally changed with the adoption of ALM 2.0. Hereby, training had only to be organized for only one type of tool which reduced the training costs. Subsequently, it also helped to reduce the workforce loss since training was only initially required, and not time needed to be reserved anymore for trainings of additional tools.

Table 10, shows the *Production Benefits* and includes the factors *Project Initiation Speed, Internal and External Data Transfer Costs, Decision Support,* and *Process Audit Time*. The integrated work environ-

ment of the ALM platform substantially reduced the cost (from 3 months to 3 hours) of the project initiation. Formerly, for initiating a project specific tools needed to be selected for each lifecycle activity, acquire these, install and customize and then try to integrate these. This initiation time could be done in a much faster and easier way with the ALM 2.0 platform since it provided already an integrated work environment. The product initiation only includes the basic configurations such as the name of the project, the project members and the initiation of the system parameters. This could be done indeed within 3 hours.

The traceability management costs include the costs for internal and external traceability of the lifecycle artefacts. Internal traceability refers to the traceability of the artefacts within a lifecycle activity, whereas external traceability considers the tracing across lifecycle artefacts. Since the ALM 2.0 platform naturally provides the tracing mechanism no additional efforts needed to be carried out to implement and manage the traceability links.

The decision support costs include the costs related to decision making based on the artefacts in the system. Formerly, since the artefacts were distributed and residing in different tools, and traceability was not fully defined deriving important information for the decision making was more cumbersome. With the ALM 2.0 platform all the artifacts are related and reside in one central repository which supports the decision-making process.

The process audit time defines the cost for the audit of the applied processes within the company. As stated before the company has a CMMI 3 certificate and adopts a strict process discipline. For this purpose, process audits are carried out regularly. Before the adoption of ALM 2.0 each process audit took substantial time since different divisions adopted different tools and processes. With the adoption of ALM 2.0 the adopted tools were similar and aligned. Thus, the process audit did not need to consider many different tools, and all the artefacts and the process state were visible. Altogether this reduced the process audit time and costs.

From the above tables, we can conclude that the adoption of ALM 2.0 had a clear benefit for the company. The evaluation also considered the issues which were less positive and might require further attention. For new projects, the transition to ALM 2.0 was not a serious problem. However, for existing projects a migration plan was necessary. This was not always easy due to a diversity of needs of each project. As such we could not define a common migration plan that could be automatically and smoothly applied to migrating all projects. Furthermore, since these projects were already running, the project team members often resisted to migrating their projects because they had to 'suddenly' change their existing tools and habits. Further, from the technical perspective we can note that the adopted MS TFS only supports build management for Windows platforms. Existing projects which were not using Windows platform as such had additional problems besides of the cognitive resistance. At the time, there was no explicit cross-platform support from MS TFS, and likewise this was considered as a serious impedance for the migration of the projects to the MS TFS platform. To mitigate this risk, the transition committee developed an add-on solution for non-Windows platforms that would enable to support cross-platform build management.

5.5. Specifying the learning

The adoption of the ALM 2.0 platform proved to be a successful decision in the end. In the diagnosis process the insight and input from the lessons learned from the efforts to adopt ALM 1.0 helped to apply a sustainable solution for the identified problems.

We decided to apply a gradual approach for adopting ALM 2.0 to mitigate the risks. For this we considered the application of pilot projects from various divisions. This helped not only to get the feedback for customizing the platform but also to directly involve the engineers in the overall transition process. During the overall transition process, the non-technical problems seemed to have a continuous impact. From the very beginning, we could observe a resistance to changing the tools and sometimes the adopted processes. Sometimes this cognitive resistance could be justified due to, for example, the need to migrate a project from a different platform than the Windows platform, to the MS TFS. However, very often the resistance was also due to the often indicated "not-invented-here syndrome" which tends to categorically reject new developments. For convincing the project teams for adopting ALM 2.0 platform, we observed that approaching the right key persons was crucial. Once these persons were convinced they became like ambassadors of the project which substantially helped to manage the non-technical problems and focus on technical problems instead.

The application of pilot projects together with the provided full and continuous assistance helped to reduce the effect of this behavior to a large extent. To support the gradual introduction, we used an agile approach (Scrum) to realize the customizations. The agile approach supported on-site development/customization, and helped the team to continuously deliver new features and user requests.

A thorough knowledge on the adopted MS TFS technology was necessary for the customizations and add-ons development to address the needs of the various divisions. For this a close contact with the tool vendor was crucial to provide direct support when necessary.

The upper-level management approached the transition process from a higher-level business perspective. The results of the pilot projects and the consecutive voluntary projects showed the important benefits for adopting ALM 2.0. The extracted metric values for these projects helped to convince the executive board to transition to ALM 2.0. After a management decision was taken to apply ALM 2.0 for all the projects the earlier critiques and resistance also decreased. Clearly, the upper-level management support and their decision had a strong impact on the adoption of the ALM 2.0 platform.

Besides of the many benefits we have also identified several obstacles that still need attention. Apart from the "not-invented-here syndrome", cross-platform compatibility seemed an important concern. To cope with new concerns, it seems that a continuation of the research activities is necessary.

6. Discussion

The main objective of the article is to report on the investigation for the impact of adopting ALM in a real industrial context and as such to understand and justify both the benefits and obstacles of applying integrated ALM. The objective of the company was to first analyze ALM 1.0 but later on this turned it out not to be feasible and we had to transition to ALM 2.0 instead. The adoption of action research as an instrument for the study helped to adapt the methodology and investigate ALM 2.0. Based on the adoption of ALM 2.0 we could indeed identify several lessons learned that could be of interest for both researchers and practitioners. In the following we first reflect on the experiences on ALM adoption, and then also discuss the interesting additional insight for an action research study. We conclude the section with the threats to validity.

6.1. Reflection on the outcome of the study

Based on our long experience in transitioning to ALM we can identify several clear critical success factors and gains of adopting ALM. Table 11 shows the identified critical success factors for transitioning to ALM. The factors *Carry out External Research, Acquire Management Support* and *Assess the Stakeholders' Needs,* and *Create*

Table 11
Identified critical success factors for transitioning to ALM.

Identified Factor	Explanation	See Section
Carry out external research	Before starting an ALM project it is important that sufficient knowledge is gained in the group. We have done this by studying the literature and investigating other related companies.	Section 4.3 and Section 5.3
Acquire management support	Adopting ALM is a costly activity that has a pervasive impact on the organization. Hence, the different management levels including upper level management is necessary to support the change in the organization.	Section 6.2 and Section 5.2
Assess the stakeholders' needs	It is crucial to identify the concrete concerns and obstacles of all the stakeholders to come up with a proper solution.	Section 4.2, 4.3, 5.2, and 5.3
Create an evaluation framework for tools assessment based on stakeholder needs and external lessons learned	Many different ALM tools can be identified from various different vendors. Adopting an ALM tool is a costly and long-term investment. Selecting the proper tool for the organization that meets the needs is essential. For this an evaluation framework appeared to be necessary to justify the selection of the ALM tools and to mitigate risks as much as possible.	Section 5.3
Training	ALM process and the tool adoption is not trivial and has a disruptive impact on the existing practices in the organization. To smoothen the transitioning process, it is important to provide the proper training to the corresponding stakeholders.	Section 5.3 and 5.5
Work closely with the tool vendor	Thorough knowledge on the adopted ALM platform is necessary for the customizations and add-ons development to address the needs of the various needs of the users. For this a close contact with the tool vendor was crucial to provide direct support when necessary.	Section 5.3 and 5.5
Attack Pain Points first	The adoption of ALM can provide solutions for a diverse range of problems. For the acceptance of ALM however it is important to attack the key problems of the organization first, which can be resolved using ALM.	Section 5.3 and 5.5
Persuade key developers and users	New technology introduction usually has to face with psychological resistance (not-invented here syndrome) and a general resistance to change of tool habits. For convincing the project teams for adopting ALM 2.0 platform, approaching the right key persons is crucial. Once these persons were convinced they became like ambassadors of the project which substantially helped to manage the non-technical problems and focus on technical problems instead.	Section 5.3 and 5.5
Start with pilot projects	Due to the large impact on the organization it is less feasible to apply ALM at once. Instead an incremental approach in which pilot projects are used appeared to be more feasible and acceptable. Selecting pilot projects for different divisions can also help to identify the needs of these separate divisions. Success stories from initial projects can further support and progress the adoption of ALM in the organization.	Section 5.3 and 5.5

Table 12
Identified gains from ALM 2.0.

Identified Gain	Explanation	See Section
Acquisition Benefits	These include reduced license and hardware costs, easier management of acquisition activities, and easier management of installation activities	Section 5.4 Table 7
Operation Benefits	These include reduced maintenance and backup operations, improved helpdesk and functional support, and reduced customization effort	Section 5.4 Table 8
Organization Benefits	These include reduced number of training sessions, and reduced workforce loss because of training	Section 5.4 Table 9
Production Benefits	These include reduced project initiation speed, enhanced traceability management, improved decision support, and reduced process audit time	Section 5.4 Table 10

an evaluation framework for tools assessment based on stakeholder needs and external lessons learned, are typically needed before the adoption decision of ALM. The factors *Training, Work closely with the tool vendor, Attack Pain Points first, Persuade key developers and users*, and *Start with Pilot Projects* relate to the later phases of the ALM adoption process.

Table 12 shows the identified gains from adopting ALM 2.0 which have been categorized as *acquisition benefits, operation benefits, organization benefits,* and *production benefits.* As discussed in Section 5.4, we could identify clear benefits for all of these categories.

Based on Table 12 we could indeed observe that ALM 2.0 has some clear benefits. However, it should also be noted that the decision for transitioning to ALM 2.0 implies some cost which should be taken into account. The overall customization of the ALM 2.0 platform is based on the *cost for initial assessment of platforms* and the *cost for customization of the selected ALM 2.0 platform.* As stated before, in our case, in Section 5.3 (Assessment and Acquisition), the initial assessment of platforms was performed by 3 persons who worked for 2 months, and as such this task required 6 man-months. As stated in Section 5.3 (Customization and Development) the cost for customization of the selection ALM 2.0 platform required 90 man-months. In total, the customization of the ALM 2.0 platform thus required 6 man-months + 90 man-months = 96 man-months. This required number of man-months could change for different situations, but clearly the assessment and the transitioning also require some non-trivial cost.

6.2. Reflection on action research

In this paper, we have applied action research for investigating the applicability of ALM. The timeline of the overall process is shown in Fig. 7. To the best of our knowledge, action research has not been applied yet for the investigating the applicability of ALM. We have meticulously followed the steps for conducting an action research. In this context, usually two key criteria are provided for

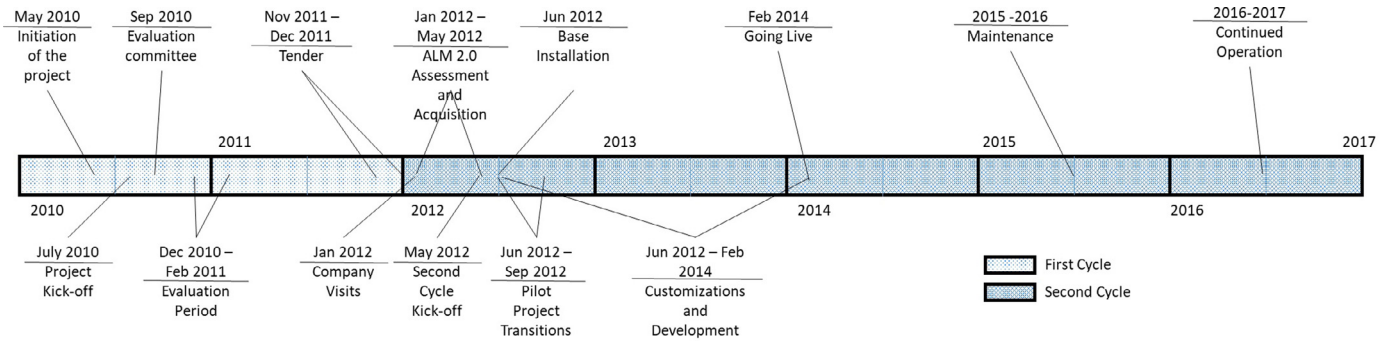

Fig. 7. Timeline of the adopted transitioning process to ALM.

judging the quality of action research (Easterbrook et al., 2008). First, one needs to check whether the original problem is authentic, that is, the identified problem should be real and worth solving. A second criterion defines whether there are authentic knowledge outcomes for the participants. Denscombe (Denscombe, 2010) states also that an action research strategy's purpose is to solve a particular problem and to produce guidelines for best practice. We can fairly state that the addressed problem is indeed very important, not only for HAVELSAN itself, but also for other companies that have to cope with process and tool integration problems. Further, the outcome of the research was indeed of high value for various stakeholders in the company including both developers and managers. Our study produced valuable guidelines for the company which could be also used by companies that have similar problems. Considering these observations, we can state that the execution of the action research was justified, and resulted in important output for the participants.

For justifying the action research and the followed steps we have also considered the framework from Davison (Davison et al., 2004) in which five principles have been defined for action research including the Principle of the Researcher–Client Agreement (RCA), the Principle of the Cyclical Process Model (CPM), the Principle of Theory, the Principle of Change through Action, and the Principle of Learning through Reflection. We had a solid agreement with the client on carrying out an action research and full commitment was given for the project. We followed the rigid guidelines of the cyclical process for action research. All our steps were based on existing theory and where needed we always searched for additional theoretical basis. Further during the action research both the researchers and clients were highly motivated to improve the situation, and necessary actions were taken for this. Finally, we also met the last principle since we had a thorough and planned reflection process during and after the action research. All together we can state that we have very carefully performed the action research to derive the most benefit out of it.

During the action research, it appeared that ALM 1.0 did not meet all the requirements for solving the identified technical problems for the context of the company. Simply adopting a piecemeal solution that used the best of breed tools from each category as it is the case in ALM 1.0 did not appear to be feasible. Hence, we had to start another action research cycle that focused on the applicability of ALM 2.0. In principle, we could consider each cycle as a separate action research, resulting in some concrete lessons learned. However, since the second action research cycle was initiated as a result of the lessons learned from the first action research cycle it was worthwhile to report the overall research for the adopted broader period. In this way, we could provide the rationale for the second action research cycle. The outcome of the second action research cycle was the selection of a feasible ALM 2.0 platform together with the corresponding guidelines.

From our study, ALM 2.0 appeared to fit the purposes of the company better than ALM 1.0 did. However, from our experience we can also state that ALM 1.0 demands less central orchestration than ALM 2.0, whereas ALM 2.0 demands a more conscious effort in the establishment, adjustment and reinforcement activities associated with enforcing a global set of company processes and practices associated with the decreed platform and tool set. As such, depending on the context of the company, and if the challenges and risks do not pose an obstacle, one could consider ALM 1.0 a feasible solution. This could be, for example, the case for smaller development organizations do not want and cannot afford the overhead of a full ALM 2.0 suite and process adoption. These companies typically use heterogeneous tool sets and supporting practices, built upon a combination of in-house, open source and vendor provided tools. Hence, ALM 1.0 could be a sustaining option for such a context.

ALM primarily considers software solutions and does not focus on a system engineering level. The latter is defined by product lifecycle management (PLM). The combined use of ALM and PLM was explored in earlier studies in the automotive sector (Ebert, 2013). In our future work, the focus on providing solutions for PLM and the integration with ALM 2.0 will be investigated.

6.3. Threats to validity

Although we have carefully carried out the steps of the action research we can state several issues that could be considered a threat to the validity of our study. We discuss these together with the actions that we have taken to mitigate these threats.

Construct validity defines to what extent the operational measures that are studied, really represent what the researchers have in mind and what is investigated according to the research questions (Runeson and Höst, 2008). Table 13 shows various identified threats to construct validity together with the counter measures.

Internal validity focuses on the study design, and particularly whether the results really do follow from the data. In our study, we have carried out two action research cycles each of which had its own data collection and result. In the first action research cycle, we focused on ALM 1.0 and derived the customer needs as well as the possible ALM 1.0 tools. The customer needs reflected the needs of the different divisions in the company and indicated the need for different tools. Based on our qualitative analysis on the integration of the tools we concluded that the adoption of ALM 1.0 using best of breed tools does not provide a sustainable solution. This was a valid reasoning that was not only confirmed by the different divisions but later on also shared by results of other studies.

The second action research cycle focused on ALM 2.0 and here we had to identify the potential ALM 2.0 tools and the criteria for evaluating and selecting the tools. In the end, we selected the most feasible ALM 2.0 tool and implemented and customized this within

Table 13
Threats to construct validity and applied counter measures in action research.

Threats to Construct Validity	Countermeasure
Researcher Bias	Since three of the researchers were part of the company this could lead to a researcher bias which defines the tendency of the researchers to observe subjects and interpret findings in the light of their own values, the tendency to selectively observe and record certain data at the expense of other data. We have tried to mitigate this risk by separating parallel evaluations of the three researchers which were later compared and discussed. Further, the outcome of the evaluations was constantly checked with those of the independent external researcher. In case of conflicting interpretations of the researchers with those of the external researcher, then these were resolved using additional analysis and meetings.
Inappropriate analysis of customer requirements	We identified representatives of each division to identify and collect the customer requirements regarding tool and process integration. The customer needs were explicitly described, iterated and eventually confirmed by the corresponding divisions. The involvement of multiple divisions helped to clarify and complete the important requirements.
Incomplete short list of ALM tools	We did a thorough analysis to existing tools based on different perspectives. We looked at the literature including scientific papers and white papers. Further we considered existing survey papers and applied snowball search. Finally, the identified tools were also discussed with peer companies.
Wrong evaluation and selection of ALM tools	We applied three important steps here. First, we looked at existing evaluations that were carried out by reliable sources. Secondly, we evaluated each tool ourselves. For this we downloaded each tool and carefully evaluated the tools based on the needs. For the second action research cycle, we arranged meetings with peer companies. The tools were evaluated and discussed by several companies together. In this way, we applied triangulation (data, observer, and methodological).
Wrong selection of pilot projects	We ensured that we had pilot projects from all the related divisions. Further we aimed at selecting pilot projects that were representative. After the initial pilot projects, many other voluntary projects used the ALM platform which led to a very broad if not full coverage of possible projects.
Infeasible base installation and customization	The base installation was important to serve the different divisions. To mitigate the risks, the team that was responsible for the installation and customization were selected from those who had mature knowledge on the selected ALM tools. In case of lack of knowledge necessary courses were followed. Further close interaction with tool vendors was realized during the whole project.

the company. For ensuring the internal validity we applied *triangulation* which means taking different angles towards the studied object. We have applied *data triangulation* (multiple pilot projects), *observer triangulation* (multiple researchers/observers) and *methodological triangulation* (indirect data extraction from literature and direct data extraction from stakeholders in the corresponding division) to increase the precision of the empirical research.

External Validity concerns the ability to generalize the results of the study. The action research has been applied within HAVELSAN that consists of multiple divisions. We have applied multiple pilot projects, analyzed and compared our results with that in the literature, and justified our conclusion based on meetings with peer companies. All together we believe that our study has a high external validity. One minor threat could be the evaluation of the final selection of a single platform. We do not claim that we had the best ALM 2.0 tool, but we selected the one that seemed to be the best regarding the defined criteria. A replication of the study for a different company could lead to the selection of a different ALM 2.0 platform simply because the context of that company would be different. However, the approach itself is systematic and generic; a company with similar characteristics and needs as in our case would end up with a selection of a similar tool.

7. Related work

To address the ALM needs, several tool vendors are working on developing ALM tools. Currently, we can identify dozens of ALM tools that have an increased capability for supporting the ALM activities. In this context, Gartner periodically publishes the Magic Quadrant reports (Murphy and Duggan, 2012), (Murphy et al., 2013) to assist business and IT leaders who are developing ALM strategies to assess whether they have the right products and enterprise platforms to support them. According to the Gartner report, the ALM market and the tools continues to evolve and play a key role in producing quality software. Several studies have provided an assessment of ALM tool suites (Grant, 2012; Murphy et al., 2013; Azoff, 2016).

The adoption of ALM has been considered by several authors in the literature. Jwo et al. (Jwo et al., 2013) discuss the ALM adoption problem from three aspects; defining the team's current software development activities, configuring the ALM platform to support these activities, and finally enforcing the ALM discipline. To facilitate this process, the authors propose a new approach called Rapid Application Lifecycle Management (RALM) which features a reference model that is described by a number templates for ALM processes. In this study, the authors designed an experimental setup with students to measure the success of proposed approach. The results of the experiment showed that their proposed approach achieved a time-saving in ALM adoption. This study is applied on an artificial setting with students, whereas our study is applied in a real-life scenario in a large-scale software company.

Kääriäinen et al. (2008) report on a case study on introducing ALM in the automation industry. To measure the impact of ALM, the authors designed a survey and asked the respondents how they felt that previous and new solutions supported the management of different project artefacts. Although the study provides survey results to show the impact of an ALM solution, the case study does not explicitly consider the ALM platform details, ALM adoption and strategy lessons learned from the ALM adoption story. The same authors describe their experience of the ALM improvement study in more detail in follow-up studies in (K. and Välimäki, 2009) and (Kääriäinen and Välimäki, 2011). In these papers, the authors documented and analyzed the case company's ALM solutions.

The above studies have been mainly carried out using examples but have not been carried based on a systematic case study research. In addition to these papers, we can also identify few white papers and books (Pampino, 2011; Rossberg, 2014; Gousset et al., 2012) discussing the details and adoption of a particular ALM platform and the guidelines for the transition. An important performance indicator in these white papers (Lipsitz Jonathan, 2013) is the focus on return on investment (ROI). In our study, we explicitly discuss the state before and after the ALM adoption to discuss the benefits.

In general, there are a few action research studies in software engineering, the ones related to our study are the ones that are mostly Software Process improvement studies (Iversen et al., 2004; Ianzen et al., 2013; Serour and Henderson-Sellers, 2005; Bjarnason et al., 2012; Grant, 2012).Serour and Henderson-Sellers (2005) investigated the effect of human behavioral patterns during the organizational transition to object-oriented software development process covering a two-year period. The authors reported two transition projects: the first project was terminated due to the resistance in the organization, whereas the second project overcame the resistance by using the lessons learned from the project to success.Lanzen et al. (2013) reported a software process improvement study in a financial organization by using action research. The action research study reports two improvement cycles where the software development processes in the organization are analyzed and improved based on the analysis findings. Bjarnason et al. (2012) and Grant, (2012) focused on improving the requirements engineering phase of the Software Development lifecycle. Bjarnason et al. (2012) proposed a method for supporting project retrospective with evidence-based timelines, whereas in Grant, (2012) used the best practices of joint application design (JAD), unified modeling language (UML), group decision support systems (GDSS), and computer aided software engineering (CASE) to reduce the time spent on requirements engineering phase.

The systematic review by Dos Santos and Travassos (2011) summarizes existing action research studies conducted in the software engineering context. The study concludes that genuine action research is rare, and in the few action research studies it appears that reporting is often incomplete. Dos Santos and Travassos (2011) further states that most of the action research studies often do not explicitly report on the research cycles and the study length. The length of the action research studies varied from 2 months to 5 years (mean time was 21 months and the standard deviation was 16 months). In our work, we have provided a detailed account of ALM transformation using two action research cycles for a 5-year period. Hence our study could be considered as a relevant illustration of the adoption of action research.

8. Conclusion

In this paper, we have carried out an action research study on adoption of ALM within an industrial context. The study can be considered unique from both the focus of the research, that is, the adoption of ALM, and the adopted research methodology, that is, adopting action research for evaluating the adoption of ALM.

To the best of our knowledge this is the first action research study to the adoption of ALM. Very often, speculative statements about ALM are provided which are not justified based on an empirical analysis. Our study provides such a thorough empirical analysis and provides an in-depth reflection on the application of ALM. Our study provides the results of an action research considering the period between 2010 and early 2017 within a large-scale software company. The results are not only important for the company in which the action research has been carried out but also provide important general lessons for similar companies. The derived lessons are in the first place related to the topic of the action research, that is, the adoption of ALM. In this study, we have achieved important lessons and insight related to different versions and applications of ALM. Our first attempt to adopt ALM 1.0 actually did not complete due to several reasons. We observed several non-technical problems such as the resistance of the projects to change their tools and process execution habits, and the lack of alignment with the organizational structures. However, the most important problems still appeared to be the technical problems of ALM 1.0 and its inability to provide a sustainable solution for the hard problems of tool and process integration. ALM 1.0 primarily seems to be based on the selection of best of breed tools, which provides short term solution only, but soon leads to tool and process integration problems and the maintainability overhead.

Our main conclusion in the initial action research was that we had to focus on a sustainable solution and not only provide temporary patch solutions. The integration of the process and tools was necessary but a piecemeal solution such as ALM 1.0 did not provide effective solutions for these. Hence, we have decided to start the second action research cycle which would build on the experiences from the first action research cycle. In the second action research cycle, we focused on the application of ALM 2.0 which adopts a more centralized approach for solving the process and tool integration problems. This second action research was carried out completely with satisfactory solutions. The adoption of the selected ALM 2.0 platform was not only a success for the 5 pilot case studies but were also soon applied for dozens of other projects within the company. With the study, we have identified the solutions that worked and learned on how to integrate this in a company.

Our study provides valuable experiences and lessons learned that can avoid non-optimal steps for solving the faced problems that we have experienced. Several important instruments in this study can be reused by many companies. We have provided the set of criteria for evaluating ALM tools. We have provided an approach for assessing the selected tools. Furthermore, we have defined a systematic approach for preparing a company and the smooth integration of the selected ALM tool using pilot studies.

The adoption of the ALM 2.0 platform required some cost that needs to be considered together with the related benefits. These costs mainly include the cost for initial assessment of platforms, and the cost for customization of the selected ALM 2.0 platform. The benefits include process automation, traceability of development artefacts, and visibility. On its turn, this led to better quality management, the reduction of development time, reduction of the cost of development, and alignment of the processes within the company. Although in the initial phases of the process there was some resistance to the adoption of ALM, later on in the research the adoption of ALM was considered beneficial by many stakeholders including engineers and managers. Thanks to the critical research approach of the adopted action research we also identified the current ongoing problems of ALM 2.0. In particular, the lack of a multi-platform solutions for the popular ALM 2.0 platforms was an important obstacle.

As stated above, the adoption of action research to ALM has not been considered before. Action research is becoming an important research methodology in software engineering that is gaining more attention recently. However, several studies have also acknowledged the lack of maturity in action research studies. Unfortunately, action research papers that include all the steps are still lacking. In this paper, we have provided a detailed action research study that can be used to illustrate the execution of the action research steps.

We can state that the adoption of action research was a very helpful instrument for supporting the empirical evaluation and integration of ALM within the company. In our future work, we will further elaborate on the adoption of ALM 2.0 and initiate new action research for empirical evaluation of other relevant research topics.

Acknowledgments

This work has been carried out at HAVELSAN. We would like to thank all members of HAVELSAN developer productivity tools team.

References

Azoff, M. Decision Matrix : Selecting an Application Lifecycle Management Vendor [Online]. Available:.

Baskerville, R.L., 1999. Investigating information systems with action research. Commun. AIS 2 (3), 4.

Bjarnason, E., Berntsson Svensson, R., Regnell, B., 2012. Evidence-based timelines for project retrospectives A method for assessing requirements engineering in context. In: IEEE 2nd. International Workshop on Empirical Requirements Engineering (EmpiRE), pp. 17–24.

Boehm, B., Turner, R., 2004. Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods. In: Proceedings of the 26th International Conference on Software Engineering, pp. 718–719.

Chappell, D. What is Application Lifecycle Management? Online]. Available: http://www.davidchappell.com/WhatIsALM–Chappell.pdf [Accessed: 19-May-2018].

Chappell, D. Application Lifecycle Management as a Business Process [Online]. Available: http://www.davidchappell.com/writing/white_papers/ALM_as_a_Business_Process_v2.0–Chappell.pdf [Accessed: 19-May-2018].

Davison, R.M., Martinsons, M.G., Kock, N., 2004. Principles of canonical action research. Inf. Syst. J. 14 (1), 65–86.

Denscombe, M., 2010. The Good Research guide: For small-Scale Social Research Projects, 4th ed. McGraw-Hill/Open University Press, Maidenhead, England.

Dos Santos, P.S.M., Travassos, G.H., 2011. Action Research Can Swing the Balance in Experimental Software Engineering. Adv. Comput. 83, 205–276.

Easterbrook, S., Singer, J., Storey, M.-A., Damian, D., 2008. Selecting Empirical Methods for Software Engineering Research. Guide tAdv. Empirical Softw. Eng. 285–311.

Ebert, C., 2013. Improving engineering efficiency with PLM/ALM. Softw. Syst. Model. 12 (3), 443–449.

Gousset, M., Keller, B., Woodward, M., 2012. Professional Application Lifecycle Management with Visual Studio 2012. John Wiley & Sons, Inc., Hoboken, N.J.

Grant, D., 2012a. Shortening requirements engineering: Dual imperative action research study. J. Comput. Inf. Syst. 53 (2), 1–8.

Grant T., 2012. “The Forrester Wave ™TM : Application Life- Cycle Management, Q4 2012,”

“Havelsan Corporate Web site.” [Online]. Available: http://havelsan.com.tr/EN. [Accessed: 22-Oct-2017].

Ianzen, A., Mauda, E.C., Paludo, M.A., Reinehr, S., Malucelli, A., 2013. Software process improvement in a financial organization: An action research approach. Comput. Stand. Interf. 36 (1), 54–65.

Iversen, J.H., Mathiassen, L., Nielsen, P.A., 2004. Managing risk in software process improvement: An action research approach. MIS Q. 28 (3), 395–433.

Jwo, J.-S., Hsu, T.-S., Cheng, Y.U.C., 2013. Jumpstarting Application Lifecycle Management : a New Approach. J. Inf. Sci. Eng. 29 (3), 475–492.

K., J., Välimäki, A., 2009. Applying Application Lifecycle Management for the Development of Complex Systems : Experiences from the Automation Industry. EuroSPI, pp. 149–160.

Kääriäinen, J., Välimäki, A., 2011. Get a grip on your distributed software development with application lifecycle management. Int. J. Comput. Appl. Technol. 40 (3), 181–190.

Kääriäinen, J., Välimäki, A., Mertins, K., Ruggaber, R., Popplewell, K., Xu, X., 2008. Impact of Application Lifecycle Management - A Case Study. In: International Conference on Interoperability of Enterprise. Software and Applications, pp. 55–67.

Lipsitz Jonathan, W., 2013. The total economic impact of microsoft application lifecycle management. Forrester Res. 33.

Murphy, T.E., Duggan, J.. Magic quadrant for application life cycle management [Online]. Available:.

Murphy, T.E., Duggan, J., Wilson, N.. Magic quadrant for application development life cycle management [Online]. Available:.

Pampino, C. Five imperatives for effective application lifecycle management [Online]. Available: [Accessed: 19-May-2018].

Peksens, I., 2013. Methodology of building ALM platform for software product organizations. In: CEUR Workshop Proceedings, 1023, pp. 105–115.

Portillo-Rodríguez, J., Vizcaíno, A., Piattini, M., Beecham, S., 2012. Tools used in Global Software Engineering: A systematic mapping review. Inf. Softw. Tech. 54 (7), 663–685.

Rossberg, J., 2014. Beginning Application Lifecycle Management, 1st ed. Apress, New York.

Runeson, P., Höst, M., 2008. Guidelines for conducting and reporting case study research in software engineering. Emp. Softw. Eng. 14 (2), 131–164 Dec.

Schwaber, C., 2006. The Changing Face Of Application Life–Cycle Management. Forrester Res.

Schwaber, K., Sutherland, J., 2011. The scrum guide. Scrum. org, October 2, 17 July.

Serour, M.K., Henderson-Sellers, B., 2005. Resistance to adoption of an OO software engineering process: An empirical study. European and Mediterranean Conference on Information Systems, EMCIS 2005.

Eray Tüzün. Eray Tüzün is currently a faculty member in the Department of Computer Engineering at Bilkent University. He has over 15 years of experience designing and building software. Prior to joining Bilkent University, he worked 9 years in HAVELSAN in various positions (Productization Lead, Academy Manager, Product Owner, and Software Engineer). He has previously worked as a Software Design Engineer at Microsoft in Microsoft Online Services group, Senior Software Engineer at Howard Hughes Medical Institute and Research Engineer at CWRU Genomics Center. Eray Tüzün received his bachelor’s and master’s degrees in Computer Science and holds a PhD in Information Systems.

Bedir Tekinerdogan. Bedir Tekinerdogan received his MSc degree and a PhD degree in Computer Science from the University of Twente in The Netherlands. Until 2008 he was a faculty member at University of Twente, after which he joined Bilkent University (Turkey) until the end of 2014. Currently he is full professor and chairholder of the Information Technology group at Wageningen University in The Netherlands. He has 25 years of experience in software engineering and information technology research and education.. His current research at Wageningen University concerns smart system of systems engineering, with an emphasis on software engineering and information technology.

Yagup Macit. Yagup Macit received his Bachelor of Science degree from Department of Management Engineering at Faculty of Management at Istanbul Technical University in 1990. He received his postgraduate degree from Department of Information Technology at Institute of Science and Technology at University of Turkish Aeronautical Association. Yagup Macit currently works as a DevOps Infrastructure Manager at HAVELSAN. He has more than 20 years of experience in Software industry. His research interests are System and Software Architecture, Software Design Patterns, Application Lifecycle Management, DevOps, Software Development Productivity, and Empirical Software Engineering.

Kürşat İnce. Kürşat İnce received his BS and MSc degrees in Computer and Information Science from Bilkent University, in 1996 and 1999 respectively. Since 1996, he has been working in various software and systems engineering positions such as team leader, project manager and product owner at HAVELSAN. Currently he works as R&D Coordinator at HAVELSAN Istanbul R&D Center. Kürşat İnce continues his PhD studies at Gebze Technical University.