Software Deployment Activities and Challenges – A Case Study of Four Software Product Companies

Mika V. Mäntylä and Jari Vanhanen
Software Business and Engineering Institute (SoberIT),
Department of Computer Science and Engineering, School of Science, Aalto University,
Helsinki, Finland
mika.mantyla@tkk.fi, jari.vanhanen@tkk.fi

Abstract— **Software deployment, including both clean installs and updates, is a crucial activity for all software vendors. It starts with a customer's order of a new release and incorporates all steps taken until the customer is satisfied with the deployed product. Using interviews as the main data collection method, we conducted a case study of four companies to discover their software deployment activities and challenges. The studied products were more complicated than pure COTS products. We noticed three product characteristics that make deployment more challenging: 1) the product is tightly integrated to other customer systems, 2) the product offers various configuration options to support different ways of working, and 3) the product requires a pre-created, complex, real-world data model to be usable. We also noticed that software deployment is multifaceted, involving activities related to customer interaction; making integrations; and configuring, installing and testing the products.**

Keywords-software deployment process; industrial case study; installablity; deployment activities;

I. INTRODUCTION

Every software vendor must enable its customers to use the developed software and usually this is achieved with a process called software deployment. For example, if a vendor omits an architectural description, it can still be successful, but if it fails to deliver, it will quickly be out of business. Mockus et al. [1] propose that quality of software deployment is a key factor contributing to the customer's perception of software quality. Jansen and Brinkkemper [2] suggest that smooth deployments are essential for increasing the customer base of software product companies. Standards and other works have often seen software deployment as a part of the release management process [3, 4]. Several technical solutions to improving software deployment, and in particular component deployment, have been proposed [5-7], and a brief summary of typical industrial deployment technologies can be found in a paper by Dearle [8]. However, despite its importance for software businesses, software deployment activities have received only little attention if one excludes the proposed technical solutions. In particular, we are lacking industrial case studies that can shade a light on the state of the practice and thus help researchers focus on the most relevant problems.

The small number of studies on software deployment might be due to the fact that it is a difficult topic to examine. Software deployment is at the intersection of interactions between the customer and the vendor, it creates both technical and management problems, and its wide scope may interest researchers studying both software engineering and management.

We discovered the importance of the software deployment process and related product characteristics, such as ease of installation and updating, when studying quality goals in four different software companies [9]. Based on this experience, we decided to investigate the topic further by making a case study of the same companies in order to understand their software deployment activities and challenges.

This paper is structured in the following way. The next section describes related work and positions our work in the literature. Section III describes the study methodology. Sections IV-VI present the results of the study. Discussion of the work is provided in Section VII, and Section VIII presents the conclusions.

II. RELATED WORK AND TERMINOLOGY

This section presents information about relevant prior work, positions our study within that body of work, and presents the terminology used in this study.

Notable among research concerning software deployment are the studies by Jansen et al. [2, 10, 11]. Their framework regarding activities necessary when updating software products [2] lays a foundation for our work.

Kajko-Mattson and Meyer [3] propose a release management model called EM3 and compare it to the release management process of a single company. They see software deployment as one step included in a larger release management process.

Coupaye and Estublier [12] propose a set of activities needed in software deployment from three interconnected viewpoints of the producer, enterprise, and user. The authors provide some insight about how they think the activities should be handled based on their industrial experiences.

Naturally, there are standards and frameworks, such as CMM [13], ITIL [14], and SWEBOK [4], which provide general guidelines for software deployment. However, the descriptions are often quite general, and they omit insight into the challenges and solutions experienced by the practitioners. Finally, there are technical approaches for software deployment [5-8] that are important for advancing

the current state of software deployment, but they reveal little about the activities and challenges of the practitioners.

Our work is different from the studies mentioned above. First, the prior studies of software deployment activities [2, 3, 12] propose that certain activities need to be performed in a certain order by certain parties. In other words, they propose a model of software deployment activities. As our work is a descriptive case study, we make no such assumptions and study software deployment in vivo. We do not claim that the activities we have identified in this paper are in a complete set of deployment activities. Rather, we describe the activities and the challenges we observed in companies we studied. Second, we study both clean install and update deployments whereas Jansen et al. [2] considered only the latter. Third, we provide a larger set of industrial experiences than those included in research by Kajko-Matsson et al. [3] and Coupaye et al. [12] who are more focused on describing their models of the software deployment process than industrial cases.

We have used the existing terminology from the prior works when possible. However, among the prior studies, there are differences in vocabulary and semantics, so we needed to decide on definitions. By software deployment activities, we mean the tasks performed after the vendor has made a product release and a customer has ordered it, until the point when users and customers have the deployed product and are ready to use it and the vendor's deployment personnel have taken care of necessary deployment aftermath, e.g., immediate support requests after the deployment and deployment report. Words 'deployment activities' or 'activities' refer from this point forward to the aforementioned software deployment activities unless otherwise noted. Deployment type refers to the actual product installation, and it can either be a clean install or update.

III. METHODOLOGY

This section presents the research questions, describes the organizations where the case study was undertaken, and explains the research and data analysis methods used.

A. Research questions

We had two research questions. First, we wanted to understand what software deployment activities exist in the studied cases and how they were performed. Thus, our first research question is:

RQ1: What deployment activities exist, and how are they performed?

Second, to enable us to direct improvement efforts in the case organizations and suggest future research in the software engineering community, we were interested in the areas that are most suitable for improvement. Such areas are typically defined by three attributes: important for the vendor or customer, poor current state, and high cost to perform. All of these attributes indicate that the activity is challenging. Thus, our second research question is:

RQ2: What are the main goals and challenges related to the deployment activities?

B. Cases

We focused on cases in four different companies. All of them were successful small- or medium-sized software product companies. These companies were successful in that they showed high profit margins of 20% or more from 2004 to 2008, grew from 45% to 600% during the four-year period, and had no debt. Three of them had an AAA credit rating.

The companies were selected based on accessibility through our research partnership. In companies B and C there were two divisions, one where deployment was not considered problematic and another in which deliveries were causing some frustration. In both cases, the division that considered deliveries more problematic was selected for the study. In both cases, the other division primarily operated in a COTS fashion, whereas the studied divisions had a more complex product or environment that impacted their deliveries.

Information on the case organizations and the studied products is summarized in TABLE I. In the table and in this description, some information is purposefully unspecific to ensure the anonymity of the cases. All the products are highly configurable and have a long history. The total number of customers varies from a few dozen to a thousand.

Update frequency for a customer varies among cases. In Cases B and D, it is linked to the release cycle of the product, and in Case A, frequent (yearly) updates are recommended in order to keep the installed products in sync with the main version. In Case C, the updates are done only if really needed, i.e., the deployed product has a fatal defect. In this case, the customer's system including the updated product installation needs to be recertified by a third party, which makes it unlikely the customer and vendor will update the product.

In Cases B and C, deployment teams were responsible for deliveries to all customers, and in other cases, the deployment personnel were part of a customer-specific project team. In Case C, the deployment responsibility is divided among members of the deployment team based on the product. In Case B, the responsibility is divided according to the complexity of the deployment. In Cases B and C, each customer is assigned a certain person who is responsible for the installed product for its lifetime. In Cases A, B and C, customer resources or third-party agents might perform easy updates. Clean installs are typically more complex, so more competent resources are needed for those.

.

TABLE I. DESCRIPTION OF THE CASES

	Case A	Case B	Case C	Case D
Personnel	>110 employees (single company)	>60 in the studied division (>300 in the whole company)	>30 in the studied division (>100 in the whole company)	~100 employees (single company)
Customers	>200	>80 for the studied products	~1000	Several hundred
Studied Product	-single product -business software for specific industry -integrated directly with the customers' other business systems -many customization opportunities	-three products for engineering in different fields -the products share a common technological core architecture -integrated directly with the customers' other business systems	-three products (C1, C2, C3) for operating engineering product -products share common core and integration framework -integrated with several other systems	-versatile software product family that supports the operations of service organizations -several customization options - integrated tightly to customer's systems
Product age	>10 years	>20 years	>20 years	5-10 years
Releases	-a release once a month	-major release every 6 months -minor release every 2 months (no database modifications)	-major release every 12 months of the product line (All) -minor release every 12 months (new functionality and fixes) (All)	-major release every 18 months (mainly packages the changes of the minor releases) -minor release every few months
Update policy	-all customers updated at least once a year	-customers typically agree to two products every year	-updates done only if really needed	-all customers are updated at least in every major release.
Deployment volumes	-clean installs: 10-20/year -main line updates: 100-200/year -updates in customer projects 200-400/year	-clean installs: rare -new applications: 0-20/year -updates: 10-30/year	-clean installs: C1 ~100/year, C2 ~20/year, C3 ~10/year -updates: C1 ~120/year, C2 ~4/year, C3 ~2/year.	-clean installs: ~100/year -updates: ~400/year -patches directly to the production environment: sometimes
Deployment organization	Project engineers who are part of project team are responsible for deployments	Technical consultants make clean installs. The customer support team makes minor version updates and major version updates for simple product configurations.	Group of people responsible for deliveries. Responsibility divided based on products. Lifetime responsibility for a deployment.	Technical project personnel deploy the product.

C. *Case study method and data analysis*

This research was conducted as a case study [15] of four Finnish software product companies. It is considered an embedded case study with four units of analysis, i.e., the four case organizations. The research was conducted in the following manner. Initially, an interview guide consisting of focus areas and open questions was prepared. The guide was based on prior work concerning the software deployment process [2] and our experience at quality goal workshops with the companies [9]. Then we interviewed two persons from each organization who had responsibilities in software deployment. During the interview, one researcher was mainly responsible for asking questions (first author) while the other researcher made notes (second author). Occasionally, the researcher making notes would ask questions if he felt that a topic was not properly covered.

After the interview, the following six steps were taken. First, immediately after the two interviews in a case organization, the first author wrote a memo about that case based on his fresh recollections and notes. This memo was reviewed by the second author. Second, after the interviews had been transcribed by a third-party service, the researchers did a thematic coding of the interviews using ATLAS.ti software. This coding formed an initial set of activities, goals, challenges and context variables. Third, complete case summaries were made and presented to the interviewees. Fourth, a spreadsheet was created to summarize all the activities and context variables of each case. Fifth, a joint session was organized with one or two participants from each organization to discuss the most important themes named by the participants. This session was also recorded, and it provided additional insight about the interview data.

IV. FRAMEWORK FOR ANALYZING DEPLOYMENT ACTIVITIES

We identified four viewpoints for analyzing the deployment activities. This section presents the viewpoints, and their utilization is covered in Section V.

First, based on the analysis of the data we propose the following descriptive classifications of the different activities: *stakeholder communication*, *installation preparations*, *installation*, and *testing deployment*. The activities are discussed according to these classifications in Section V.

The second viewpoint considers who performs the activity. Typically, it is the vendor or the customer, but in some cases the vendor may utilize third-party agents.

The third viewpoint, the deployment type, is either a clean install or an update. It is also possible that the activity

is related to both types of deployment but it has a different nature in them.

The fourth viewpoint characterizes whether the activity is currently addressed with technical or managerial solutions. Technical solutions include means such as developing an installation program, and managerial solutions include means such as a process for product configuration. This viewpoint could also be characterized as practices vs. tools or automation vs. manual. This viewpoint is not relevant for all activities and is discussed below only if it is relevant.

V. DEPLOYMENT ACTIVITIES

This section presents the deployment activities in each of the four classes described above: stakeholder communication, installation preparations, installation, and testing deployment. A summary of the activities can be seen in TABLE II.

A. Stakeholder communication

Transferring knowledge about the software product, its deployment, and changed functionality internally on the vendor side as well as between the customer and vendor is critical for successful deployment. This section describes activities related to this topic.

Informing stakeholders of deployment contents. The goal of this activity is to keep the stakeholders informed about the content of the deployment. The stakeholders include the vendor's personnel, customers, and end users. In all cases, information about the changes in new releases and what the changes mean for individual customers and their product installations is provided by the vendor for each customer. This information is provided both prior to and after the update. Based on the information, customers will decide whether they want the update or not. The organizations we studied reported that if the information is not accurate or missing, customers will be annoyed. This is especially true if the new release forces the users to change the way they work. Case D organizes formal internal training about new applications, new features, and changes in major versions when the release is in internal testing. It allows the vendor personnel to provide more accurate information to the customers. However, according to the interviewees, more advanced solutions for composing accurate release notes are required. Case A and D, had implemented systems allowing release notes to be automatically collected from developers' comments in the version management system. Informing customers is important both for clean installs and updates. However, for clean installs, providing such information is usually done during the sales process. For updates, a smaller amount of information is needed, but the information is more fragmented, and it needs to be gathered for every release.

Training the users. In all cases, the users were given training for using the product. In all cases, a manual exists that enables the user to learn about the product. However, Case C planned to use recorded video clips to train the users to use the new features, so some technical solutions may exist for this activity as well. Training is much more important in clean installs, but it cannot be completely neglected in updates when new features are included in the deployed software.

User support. In all cases, the vendors have help desks to support customers. Here we describe only user support related to the software deployment. Often the user support is noted regarding deployment, so they can prepare for more requests. In Case D, the user support group is responsible for making simple updates. In all cases, the user support can also be provided by the consultant or project engineer who made a particular deployment. Who takes care of which support request varies even within cases. The amount of user support needed is naturally dependent on the quality of user training and informing stakeholder activities.

B. Installation preparations

Before the product can be installed at the customer's site, certain activities are required in order to make sure the actual installation goes smoothly. In this section, such activities found in the cases we studied are described. Typically, these activities are performed in the vendor's site. In all cases, installation preparations required the largest effort of all the activities in clean installs but were simpler to perform in updates.

Importing initial customer data was part of clean installs in all the cases. In Cases B and C, the data modelled complex physical structures, and creating and testing it was a laborious and long process that could take months. In all cases, the vendors were responsible for creating the data or at least converting some existing data to be used in the new system. Occasionally, conversions suffered because there was incomplete or incompatible data. Case A considered contractual means for increasing the customers' responsibility and effort in providing valid data. Generally, the vendors relied on person dependent experience and informal processes for arranging access to different data sources and for ensuring that all data had been gathered and was correct and approved by all parties.

Configuring the product for the customer was required in all cases to make sure the product behaves as a customer wants it to. The configuration possibilities were typically difficult for the customer to understand. Setting up options may require editing the database or configuration files directly or using tools. Naturally, technical solutions (tools) can help in the configuration process, but replacing humans completely is not plausible because the configuration is related to training the customers. They need to understand what the different product configuration settings mean in practice. Both the vendor and the customer are needed when configuration occurs, as it is a mutual learning process in which the customer learns from the product and the vendor learns from the customer's processes, environment, and needs.

The fact that all cases manage to produce customer-specific versions of the software without code modifications is a technical achievement that has helped them succeed. Case D, had managed to standardize the product configuration for a large customer segment, and this simplifies the deployment considerably.

TABLE II. DEPLOYMENT ACTIVITIES

Class	Activity	Who	Install or Update
Stakeholder Communication	Informing stakeholders of deployment contents	Vendor	Both
	Training the users	Vendor	Both
	User support	Vendor	Both
Installation Preparations	Importing initial customer data	Vendor	Install
	Configuring the product	Both	Both
	Integrating the product	Vendor	Install
	Scheduling a deployment date	Vendor	Both
	Creating the deployment package	Vendor	Both
Installation	Pre-install checks	Vendor	Both
	Making rollback possible	Vendor	Update
	Installing the product	Both	Both
	Transferring the product from test to production environments	Vendor	Both
	Maintaining information about deployed products	Vendor	Both
Testing	Testing at vendor site	Vendor	Both
	Testing at customer site	Both	Both

Using the existing configuration as a basis for a new configuration was a typical way to start the work in the cases we studied, but the vendors did not see this as the optimal approach. Vendors felt that more formal processes could provide increased visibility for the product configuration process and ensure that all product configurations are created using the same steps. Product configuration is important in both the clean installs and the updates because during updates it is possible that the new product version has new features, and configuration options may need to be tweaked as well.

Integrating the product to a customer's other systems was necessary in all cases. Creating the interfaces required considerable work, and vendors viewed the integrations as fragile, so they exhibited a strong "if it ain't broke don't fix it" mentality. From management's viewpoint, there were no processes, and from the technical perspective, there were few existing tools for integrations. In Case C, the final integration at the customer site could be finalized only when installing the product. Case C was the only organization that had created a middleware that connected all the company's products to the environment. Even in that case, the people responsible for the deployment saw the middleware as fragile and patchy rather than a competitive advantage enabling smooth deployment. However, they admitted that the middleware was an improvement over what they had done in the past. In all cases, vendors performed the integration with the help of the customer or a third-party system expert. Integrations were created during a clean install, and in all cases there was agreement that they should not be touched in an update.

Scheduling a deployment date for either a clean install or an update was done in cooperation with the customer. However, scheduling for clean installs is more flexible because there are no users yet. In all cases, the updates need to be done outside of office hours to minimize users' annoyance due to the system downtime. The date needs to be communicated to the users in advance. In Cases B and C, special times were required. For example, sometimes the weather would not allow a system to be down or it would otherwise be impossible to deploy the system. In Case C, clean installs also require that third-party vendors are present and assisting during a deployment. Getting several parties to the deployment site at a specific time requires extensive communication. In all cases, customers are able to perform some updates themselves, which removes the need for the vendor to do scheduling.

Creating the deployment package. A tool is utilized in cases A, B and C, for creating the deployment package. However, some manual work is still needed in all cases, e.g., for collecting files to correct directories. For updates, this step could be completely automated, and Case B has mostly succeeded in this. For the clean installs, complete automation is not possible due to customer-specific data models. In all cases, general instructions existed on how to create the package. However, as the people responsible for the package creation already knew what they needed to do, they rarely used the instructions. This created a problem because the instructions were often somewhat outdated as updating them was not considered very important. This activity was required in both installation types, but in updates, the package creation was easier as the customer data model and configurations were already in place.

C. Installation

This section describes the activities needed when performing a product installation that is either the clean install or update. Typically, these activities are performed in sequential order at the customer's site.

Pre-install checks. Prior to installation, various checks should be performed [2], e.g., external software components, environmental checks (disk space), or the integrity of the installation package. External software components were checked most often, and the integrity checks were most often omitted. The companies viewed the checks to be simply installation program features that they had not yet implemented. The checks were usually performed by the vendor's staff, but occasionally the customers did some of them.

Making rollback possible. In Cases B, C and D, a backup of executable binaries and the database is taken just before an update. In case A, regularly taken updates of the database are considered enough because making an additional backup would take several hours during which time the system is down. In three cases, backups were very rarely needed, and in Case C, the deployment personnel did not remember that they had ever been used. This activity is done by the vendor and is automatically done by the product either regularly or during product updates.

Installing the product. All clean installs and certain types of updates are done by the vendor's personnel or by third-party agents at the customer's site. Simpler updates are done in all cases through the Internet by the vendor, and in case C, it is sometimes accomplished by sending a CD the customer uses to take care of the actual installation. In case D, the vendor hosts the product for a large number of customers with similar configurations, which simplifies updates. In Cases A, B and D, the database structure also often needs to be updated during an update. Installer tools or scripts are used to make the changes, but technical expertise is needed to interpret the possible warnings and error messages and to decide which actions, if any, are necessary.

In all cases, some instructions for the installation work exist for use by the vendor's personnel. All companies had a difficult time deciding whether to have installation instructions or not and at what level. Experts do not need the instructions as they already know what to do, so deployment perssonel are reluctant to update them. On the other hand, the level of detail in the instructions needed by new employees would be too excessive to create and maintain. A similar challenge existed with regard to instructions for creating deployment package activity.

Transferring the product from test to production environments. In three cases, transferring the product between environments is relatively easy. In case C, transferring is not very important as there is no customer test environment. However, in Cases A and B, it is not possible to simply copy the product installation from the test environment to the production environment. Still, the installation in the test environment simplifies the installation in the product environment as possible adaptations for the installation are known.

Maintaining information about deployed products. In all cases, the companies maintain at least some documentation regarding the deployed products including product version, product configuration, interfaces, and information on what was done at the customer's site and what was problematic. Currently, this information is usually gathered manually. In Case C, a report is written for each deployment, and in other cases, the information is maintained in a spreadsheet. Some automation also exists. In Case C, company personnel can get version information of all components installed. This company also copies the entire hard drive content to a memory stick when leaving the customer's site to produce an exact copy of the environment at the vendor's site.

This activity could also benefit from advanced technical solutions. For example, products would report their information to the vendor directly. However, it should be noted that in many cases, a large amount of information is needed about the product, its configuration, and the environment in which it exists. In all cases, the vendor is responsible for gathering this information, not the customer.

D. Testing the installed product

Even though all releases are tested during the software development process, certain testing activities are still needed during deployment. Below we describe the deployment-specific testing activities.

Testing at the vendor's site. The vendor's personnel test customer-specific aspects of the product and install the product at the vendor's site**.** In all cases, it is impossible to completely test all customer-specific parts at the vendor's site because other systems used by the customer are not available. In Case B, no customer-specific testing is done at the vendor's site due to highly complex customer environment. In all cases, some testing is done when the vendor personnel are installing the product.

Testing at a customer's site The amount of testing at customers' sites varies. The responsibility for testing is split between the customer and the vendor, and it also varies. Vendors typically perform smoke testing at a customer's site as they have already extensively tested the software. Some customers want to test the new release prior to moving it to the production environment. However, in many cases, the customer does not have resources to perform the testing, and bugs from the test environment are moved to the production environment. The customers emphasize testing more during clean installs than updates.

VI. GOALS AND CHALLENGES OF DELIVERIES

In addition to the deployment activities, we also studied the main goals of the vendors and customers during software deployment and the most striking challenges that occur. The vendors' most important goals were 1) decreasing the deployment effort and 2) decreasing the customers' dependency on individual experts.

There were four key challenges related to decreasing the deployment effort. First, extensive time was needed for the product configuration, and vendors considered the ad hoc nature of the product configuration to be a problem.

Second, vendors whose products had a large number of configuration options felt that a formal process could help alleviate that problem, but they recognized that the product configuration is a collaborative activity during which the customer learns about possibilities and needs from the product. Thus, a more formal approach could add visibility to the configuration process but might not reduce the effort and lead time necessary during the process.

Third, building the product integration was a laborious task, but the vendors saw few avenues for improvement. In Case C, the company had created middleware to reduce the effort in creating the interfaces, but it was viewed as patchy and fragile.

Fourth, creating the data model and getting proper data from the customer required considerable effort in clean installs. Creating the data model manually took considerable

effort in Cases B and C, and in Case A, getting proper input data from the customer was difficult.

The vendors were keen to decrease dependency on experts in deployment through increasing knowledge transfer among their own employees. They considered expertise to be necessary as deployment required detailed product and environment knowledge. For example, during database updates, possible difficulties could emerge that only an expert could solve. However, reducing the dependency on experts through knowledge transfer is a laborious process, and the companies seemed to have accepted that it is just more efficient to have a few individuals who have responsibility for the deliveries, even though it causes difficulties when they are unavailable.

The customers' most important goals, according to the vendor's personnel, were 1) deployment requires little attention on their part, 2) there are no undesired changes in the update (defects or feature changes), and 3) the downtime is scheduled appropriately. The customers are highly displeased if a function they have been using in the previous version is no longer working. Furthermore, even if there are changes in the functionality, like a new layout of the GUI that is not explained beforehand, the customers will be displeased. In general it appears that the customers do not want to change their current way of working, and if they must do so, they must be notified.

VII. Discussion

This section answers the research questions and evaluates the study and its limitations.

A. RQ1: Deployment activities

What deployment activities exist and how are they performed?

As one viewpoint, we proposed a descriptive classification of the different deployment activities. The content of the classes was presented in detail in Section V.

Stakeholder communication activities deal with transferring knowledge or information among people. Furthermore, the changes in one of these activities are likely to affect the other activities; e.g., if information of release notes is extensive and used extensively by customers, there is less need for customer training.

Installation preparation activities must be done before an installation. These activities, such as product configuration, are often laborious in clean installs but are much simpler in updates. These activities are typically done on the vendor's site.

Installation activities are directly related to installing the product for the customer and are typically performed in sequential order at the customer's site. These activities must be scheduled with the customer, and any problems or difficulty in performing them are often directly visible to the customer. Thus, they are likely to have a strong impact on the customers' perceived quality of the system.

Testing activities do not refer here to testing performed during software product development but only to that which is specific to deployment. We identified three types of testing: testing the customer-specific aspects at the vendor's site, testing the installation at the vendor's site, and testing at the customer's site by the vendor and the customer.

We decided to use descriptive classification instead of temporal because the order of activities is context-dependent and is likely to vary. For example, should user training activity precede install product activity or vice versa? Descriptive classes should also be easier to understand than activities which, in some contexts, just happen to follow another. Installation and installation preparation are temporal but they were actually separated from each other as they have different nature. Installation activities are performed in sequential order at the customer's site when installation preparation activities are much more laborious and are performed at the vendor's site.

We studied all the discovered activities from three additional viewpoints. First, we looked at whether the vendor or customer performs the particular activity. We found that in our case study's organizations, most of the activities are performed by the vendor. This is different from what two previous studies reported. Their models of software deliveries suggested that most of the activities should be performed by the customer [12]. Our view is that a company using either approach can be successful. The fact that the vendor performs the updates can be viewed as a competitive advantage when the customers want to focus on their core business rather than maintaining their IT infrastructure. Furthermore, it is often not feasible for a customer to perform the installation when complex integrations or product configurations are required.

Secondly, we looked at whether an activity is related to clean installs, updates, or both. We found that many activities were related to both deployment types. However, they can be different. For example, configuring the product for a customer is typically a lengthy process in clean installs that involve mutual learning. In updates the needed changes for product configuration are quicker to perform as a customer understands the product and the vendor understands a particular customer's needs. Therefore, some activities that were performed in both types of deliveries were more laborious during a clean install. We also found that the activities that required the most effort were usually related to the clean installs, such as product integration.

Thirdly, we examined the division of work that is technical, i.e., tools and technologies, and managerial, i.e., practices and processes. We found that many activities could benefit from either technical or management improvements. For example, building an installation tool (technical approach) for automating the installation work would decrease the installation effort and possibility for errors. Of course the benefits and costs of the installation tool depend on several factors making it difficult to evaluate the overall benefits in advance. Considering the installation tool these factors include, e.g., the number of customers, expertise needed in the manual installation, effort needed and errors caused by the manual installation, effort needed to build the installation tool, and the feasibility of the installation tool.

We found many activities similar to those proposed in prior works [2, 3, 12], which strengthens the shallow research in the area of software deployment. The most

important difference between our work and that of prior researchers is that they propose a model that contains all the necessary activities for deployment. We focus on describing the activities found in the four cases we studied and do not attempt to create a general, complete model based on these cases. Our work should be viewed as an industrial case study of deployment processes in vivo. In addition, some prior works [2] focus on only updates, whereas we also consider clean installs, and others provide limited data on the industrial cases [3, 12].

B. RQ2: Deployment goals and challenges

What are the main goals and challenges related to the deployment activities?

We identified the main goals and challenges in software deployment in order to find suitable targets for improving software deployment in the companies we studied. The results related to this research question were presented in Section VI.

All the vendors were keen to reduce the deployment effort. The most laborious activities for the vendor took place during clean installs and occurred during the installation preparation activities. System integration and product configuration were the most laborious activities, but the creation of the data model was also described as quite laborious and occasionally difficult because proper input data might not be provided by the customer. The preparation activities required much less effort in updates.

Companies wanted to decrease their dependency on experts in deployment, and transferring knowledge among vendors' employees was another challenge. We also found that customer communication was important, but company representatives we interviewed did not perceive this to be very challenging when executed properly. All vendors noted that any changes in functionality need to be explained to the customer if they affect the customer's use of the software. Overall, customers are not keen on changing their work practices unless doing so brings them a dramatic improvement.

We are not aware of prior studies that have listed deployment challenges based on industrial cases as we have done. However, the challenges themselves are not new, and other scholars have addressed these problems. Although the studied products are highly domain specific and cannot be considered to be general purpose, i.e., Enterprise systems such as ERP or CRM, the vendors could undoubtedly learn something from the studies on the integration solutions used in enterprise systems [16, 17].

The problem of product configuration has been addressed from a technical perspective [18, 19]. However, we are not aware of any empirical studies of the product configuration process done jointly with customers and vendors and involving customers learning about the product and its possibilities. As the companies felt that this area was problematic, perhaps future works should look into it.

Knowledge transfer has been extensively studied in the area of management sciences [20] and also in the software engineering arena, using what is known as the experience factory [21].

C. Limitations and evaluation of study

This section presents the main limitations of the study through an assessment of its internal, construct, and external validity. Internal validity needs to be assessed when causal relationships are studied [15]. The purpose of the study was not to establish causal relationships. Some of our findings have a causal nature and it is possible that unknown affecting factors exist although we had thorough understanding of the cases.

Construct validity is concerned with the design of a study and whether the studied artifacts really represent what the researchers had in mind [15]. We examined more than one company, so triangulation of companies is present. We had limited triangulation of the data collection methods. We relied mostly on the two interviews performed at each organization. Additionally, we benefitted from our previous experiences with the companies [9], and from the informal communication that was used to clarify the information obtained during the interviews. Due to the research collaboration, we examined the cases and their contexts over a longer period, which strengthens the results. All of the research was done by two people, which limited researcher bias to some degree. However, as the two people were working closely together rather than independently the researcher bias is not completely eliminated.

External validity is concerned with whether it is possible to generalize the results [15]. Generalizing results is difficult in software engineering because of the effect of context [6, 7, 13]. As a case study offers a limited possibility of generalization, our case selection has undoubtedly affected the results. As discussed in Section III.B, two vendors also had divisions providing COTS products, but they were not selected for the study as they had no problems with deliveries. Although this limits one's ability to generalize the findings, we can also interpret this as a tentative result indicating the impact of the software's operational environment to the deployment process. It seems very intuitive to think that standard COTS products result in fewer deployment-related difficulties compared to products closer to Enterprise systems.

VIII. Contribution and Conclusions

The contribution of this work is distilled and increased understanding about the process of software deployment and the difficulties associated with it based on experiences of four software companies. This information can be useful for both practitioners and academics. Practitioners can learn from the experiences of this case study and benchmark their software deployment process against the four industrial cases that combine over 50 years of experience from deploying over 1000 systems to customers. Academics can benefit from this industrial case study by steering their research towards industry relevant problems.

Based on our study we make two conclusions. *Three product characteristics make deployment challenging,* based on our case study. The characteristics are as follows: 1) the product is tightly integrated to other customer systems, 2) the product offers various configuration options to support

different ways of working, and 3) the product requires a pre-created, complex, real-world data model, e.g., a model of an airplane, to be usable. Thus, deployment seems to be a challenging problem only to a subset of all the software deployed. It is unlikely that the deployment of word processor software, which holds none of the above-mentioned attributes, would be particularly challenging.

Software deployment is a multifaceted topic, consisting activities such as customer interaction, making integrations to other systems, product configuration, and testing. Thus viewing deployment as a purely technical topic, with features of an installer tool, for example, offers a limited point of view. Many activities presented in this paper are likely to require different skills and tools. We hypothesize that many problems related to software deployment come from its broad and multifaceted scope. However, we acknowledge that the research of individual activities of software deployment may not be very mature. Therefore, it is no wonder that companies are experiencing difficulties with software deliveries.

ACKNOWLEDGEMENTS

We would like TEKES for funding our work, and anonymous representatives of the case companies for participating in our research.

REFERENCES

[1] A. Mockus, P. Zhang, P.L. Li and A. Res, "Predictors of customer perceived software quality," *Proceedings. 27th International Conference on Software Engineering (ICSE), 2005.* 2005, pp. 225-233.

[2] S. Jansen and S. Brinkkemper, "Definition and Validation of the Key process of Release, Delivery and Deployment for Product Software Vendors: turning the ugly duckling into a swan," *22nd IEEE International Conference on Software Maintenance, 2006. ICSM'06,* 2006, pp. 166-175.

[3] M. Kajko-Mattsson and P. Meyer, "Evaluating the acceptor side of EM 3: release management at SAS," *Empirical Software Engineering, 2005. 2005 International Symposium on,* 2005, pp. 10.

[4] A. Abran, J.W. Moore, P. Bourque, R. Dupuis and L.L. Tripp, *SWEBOK: Guide to the Software Engineering Body of Knowledge,* Los Alamitos, California: IEEE Computer Society, 2004.

[5] E. Dolstra, E. Visser and M. de Jonge, "Imposing a Memory Management Discipline on Software Deployment," *ICSE '04: Proceedings of the 26th International Conference on Software Engineering,* 2004, pp. 583-592.

[6] R.S. Hall, D. Heimbigner and A.L. Wolf, "A cooperative approach to support software deployment using the software dock," *Proceedings of International Conference on Software Engineering,* 1999, pp. 174-183.

[7] J.M. Bishop Ed., *Component Deployment, Proceedings of IFIP/ACM Working Conference on,* Berlin, Germany,: Springer, 2002.

[8] A. Dearle, "Software deployment, past, present and future," *2007 Future of Software Engineering,* 2007, pp. 269-284.

[9] J. Vanhanen, M.V. Mäntylä and J. Itkonen, "Lightweight Elicitation and Analysis of Software Product Quality Goals – A Multiple Industrial Case Study," *Proceedings of the third International Workshop on Software Product Management (IWSPM),* 2009, pp. 42-52.

[10] S. Jansen and S. Brinkkemper, "Ten Misconceptions about Product Software Release Management explained using Update Cost/Value Functions," *Pre-publication Proceedings of the First International Workshop on,* 2006,

[11] S. Jansen, G. Ballintijn, S. Brinkkemper and A. van Nieuwland, "Integrated development and maintenance for the release, delivery, deployment, and customization of product software: a case study in mass-market erp software," *Journal of Software Maintenance and Evolution: Research and Practice,* vol. 18, no. 2, 2006, pp. 133-151.

[12] T. Coupaye and J. Estublier, "Foundations of enterprise software deployment," *Proceedings of the 4th European Conference on Software Maintenance and Reengineering (CSMR2000),* 2000, pp. 65-73.

[13] CMMI Product Team, *CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development,* CMU/SEI, 2001.

[14] Office of Government Commerce (OGC), *ITIL Version 3 - Service Transition Book,* TSO (The Stationery Office), 2007.

[15] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering,* vol. 14, no. 2, 2009, pp. 131-164.

[16] H.J. Rognerud and J.E. Hannay, "Challenges in enterprise software integration: An industrial study using repertory grids," *Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on,* 2009, pp. 11-22.

[17] M. Themistocleous, Z. Irani, R. O Keefe and R. Paul, "ERP Problems and Application Integration Issues: An Emperical Survey," *Proceedings Annual Hawaii Int'l Conf on System Sciences,* 2001, pp. 278-287.

[18] J. Tiihonen, T. Soininen, T. Männistö and R. Sulonen, "State-of-the-practice in product configuration–a survey of 10 cases in the Finnish industry," *Proceedings of the First Knowledge intensive CAD: IFIP WG 5.2 Workshop,* 1996, pp. 95.

[19] M. Raatikainen, T. Soininen, T. Männistö and A. Mattila, "Characterizing configurable software product families and their derivation," *Software Process: Improvement and Practice,* vol. 10, no. 1, 2005, pp. 41-60.

[20] L. Argote and P. Ingram, "Knowledge transfer: A basis for competitive advantage in firms," *Organ.Behav.Hum.Decis.Process.,* vol. 82, no. 1, 2000, pp. 150-169.

[21] V.R. Basili, "The experience factory and its relationship to other improvement paradigm," *Lecture Notes in Computer Science,* 1993, pp. 68-83.