The Waterfall Approach and Requirement Uncertainty: An In-Depth Case Study of an Enterprise Systems Implementation at a Major Airline Company.

The Waterfall Approach and Requirement Uncertainty:
An In-Depth Case Study of an Enterprise Systems Implementation at a Major Airline Company

Huub J. M. Ruël, University of Twente, The Netherlands

Tanya Bondarouk, University of Twente, The Netherlands

Stefan Smink, Sodexo Altys, The Netherlands

ABSTRACT

The Waterfall approach has been the dominant approach for enterprise systems (ES) implementation since the 1970s. It offers ES project managers a simple, step-by-step way to make ES projects manageable and minimize drawbacks. The main criticism of this approach centres on its inflexibility regarding requirement uncertainty. In this article, the authors challenge this criticism. By means of an in-depth case study of a Waterfall approach-based ES implementation project within the maintenance department of one of the world's biggest airline companies, this article will illustrate how it deals with requirements uncertainty and required flexibility in practice.

Keywords: *Agile Approach, Enterprise Systems, In-Depth Case Study, Project Management, Requirements Uncertainty, Waterfall Approach*

INTRODUCTION

Enterprise systems (ES) can be defined as configurable, off-the-shelf software packages that provide an integrated suite of systems and information resources for operational and management processes across a broad range of business activities (Ward et al., 2005). They are intended to support business in the contemporary knowledge-based global economy (De Carvalho & Tanaka, 2008). Enterprise Systems (ES) cover a plethora of subjects that range from Enterprise Resource Planning (ERP), Enterprise Content Management (ECM) and Customer Relationship Management (CRM), to Decision Support Systems and Business Intelligence. It is acknowledged that developing and managing these systems involve dealing with the dynamics of contextual forces (Nandhakumar et al., 2005).

DOI: 10.4018/jitpm.2010040103

In their review of the studies of enterprise systems implementation, Shanks et al. (2000), Somers and Nelson (2001), Nah et al. (2001), and Umble et al. (2003) show project management, balanced project team, clear goals and objectives, change management, minimum customization, and project champion to be the main critical success factors. All of them stress the importance of ES project management issues as one of the major success factors, a conclusion widely debated in the academic literature on ES implementation and information systems (IS) implementation (Austin & Devin, 2003; Brown, 2004; Kim & Pan, 2006).

ES projects are notorious for their failure rates (Barker & Frolik, 2003; Mendel, 1999; Umble & Umble, 2002), and the question remains, why is project management of ES implementations more difficult than that of other types of IS projects? Jurison (1999) explains that the difficulty is in the nature of the 'product'. The most frequently cited aspects that make managing software projects more difficult are: intangibility of the 'product', complexity of the 'product', and volatility of the requirements. Or in other words, software is invisible, it is difficult to comprehend, and its requirements are under constant pressure to change, making ES project success hard to achieve.

Many project management methodologies and tools have been developed throughout the years that claim to contribute to ES project success, and the importance of project management is fully acknowledged in the literature (White & Fortune, 2002; Somers & Nelson, 2004). Project management is considered a series of activities associated with carrying out a project as effectively as possible (Jurison, 1999, p. 6). Project management aims to anticipate as many of the dangers and problems as possible and to plan, coordinate, and control the complex and diverse activities of projects to ensure successful completion despite the risks (Lock, 2007). Project management has a long history, but in the modern management literature it was Henry Gantt (1861-1919) who first proposed that an organized approach was needed to manage the complex interrelationships among an enormous number of different tasks performed by many different specialists. He developed the Gantt chart, a way of ordering operations and work which is still widely in use by software project managers to track the progress of projects (Jurison, 1999; Lock, 2007). With the enormous growth in information technology use since the 1970s, a new type of project manager emerged: the IT or software project manager. Unfortunately, this type of project manager need not have project planning or scheduling experience. New project management approaches emerged based on successful manufacturing techniques of mass production, of which the Waterfall approach by Winston Royce (1970) has become a prominent exponent (Lock, 2007). In software project development studies, the Waterfall approach is the one referred to predominantly (Huo et al., 2004; Jiang & Eberlein, 2008).

However, in response to growing environmental uncertainty and flexibility, the Waterfall approach is being criticized for its rigid character (Nerur & Balijepally, 2007). Although we understand the roots of this criticism, we cannot fully agree with it. We still see a lot of potential in the "old" approach and argue that if managed well, this approach can greatly contribute to EIS implementation.

This article aims to contribute to this debate by starting from the assumption that the Waterfall approach in practice is not as ill-suited to the dynamics of ES projects as its critics claim, since it is still the most widely used approach for ES implementation projects (Laplante & Neill, 2004). The leading research question therefore is: how does a Waterfall approach-based ES project cope with requirements uncertainty?

The remainder of the article is organized as follows. First we elaborate on the discussion on the origins and advantages / limitations of the Waterfall approach. After that, we present its main 'rival', the Agile approach, and assess its assumed strengths and weaknesses. Then we introduce our in-depth case study, a Waterfall approach-based ES implementation project in one of the biggest airline companies in the world, present our findings and draw conclusions.

The value of our contribution is two-fold: firstly, this article provides an in-depth case study of a Waterfall approach-based ES project, something that is lacking in the literature. Such an in-depth case study in itself can help to elucidate ES projects for a broader audience. Secondly, this article stimulates the Agile approach supporters to clarify and sharpen the relevance of their criticism towards a plan-driven approach such as the Waterfall approach as we aim to present convincing evidence that the Waterfall approach is well-suited to the dynamics of ES projects.

THE WATERFALL APPROACH TO ENTERPRISE SYSTEMS IMPLEMENTATION: A REVIEW AND ALTERNATIVES

The Waterfall approach was introduced by Winston Royce in 1970, adopted by software project managers and further developed through lessons learned from software projects (Harrison, 2003). Modern project management methodologies, such as PRINCE2 and PMBOK, evolved from the Waterfall approach (Harrison, 2003) and are the most widely used project management methodologies in Europe and North America. The Waterfall approach treats a project as a linear process consisting of a series of basic sequential stages, each of which needs to be formally validated before moving to the next stage, thus reducing the complexity of an ES implementation process (Jurison, 1999; Khalifa, 2000; Huo et al., 2004).

The strength of the Waterfall approach and project methodologies that evolved from it mainly involves the management of each definable stage: planning, executing, testing, and closing. Although the labels for the separate stages of the Waterfall approach can differ per author, the basics still follow the original approach introduced by Royce. The recent literature agrees that ES projects based on this approach typically flow through five stages (Sommerville, 2006; Goedecke, 2007). These are: requirements definition (defining the functionalities), design (translating requirements into an executable format for a software system), coding (turning design into an actual, testable product, i.e. actual programming), system testing (testing whether the actual systems works in accordance with requirements and is accepted by the end-user), and operation (installing the software, training the end-users, setting up support and fixing bugs).

Drawbacks of the Waterfall Approach to ES Projects

Although it is the most mature and widely used form, the Waterfall approach has been increasingly criticised over the years. In sum, the criticism boils down to the following: the Waterfall approach is highly formalized and consists of sequential stages and therefore is not able to control uncertainty and changes in requirements. Requirements uncertainty and change are inherent to ES projects, however, as it is virtually impossible for end-users to list the requirements of the system precisely in advance (Beck, 1999; Khalifa & Verner, 2000; Middleton, 2000; Highsmith & Cockburn, 2001; Huo et al., 2004; Hass, 2007; Nerur & Balijepally, 2007). McConnell (2004) in this respect uses the term *the wicked problem*, a problem for which the requirements of the solution cannot be entirely known before completion. The Waterfall approach's focus on planning incorporates the risk that by the time a system is built, the problem it was supposed to solve has changed; the final result might be in accordance with the initial requirements with hardly any shortcomings, but when the job is done, it turns out to be the 'wrong' software (Austin & Devin, 2003). Parnas and Clements (1986) suggest that even if end-users could know all the requirements at the beginning of a project, there are many other factors that need to be known in advance to build a software system. In addition, 'freezing' the requirements at the beginning of a project, as desired by a Waterfall-based ES project management approach, is an excellent opportunity for end-users to ask for everything they think might be useful, as they realize that they might only get such an opportunity once (Poppendieck & Poppendieck, 2003). Overall, it can be said that a Waterfall-based ES project is mostly concerned with software development; it is less concerned with implementing a product that is accepted by the end-user (Becker, 1999).

An Alternative to the Waterfall Approach: The Agile Approach

As an alternative to the Waterfall approach, a new form of ES project management has emerged since the late 1990s, called the Agile approach (Huo et al., 2004). The first Agile approach-based project management methodologies were introduced by Kent Beck. It took until 2001 before the concept of the Agile approach was formally born. In that year a group of prominent practitioners agreed upon the basic values of the Agile approach as laid down in the *Manifesto for Agile Software Development* (Beck et al., 2001; Highsmith & Cockburn, 2001; Hass, 2007). It basically says that the highest priority is to satisfy the customer through early and continuous delivery of valuable software (Beck et al., 2001). The need for an alternative to the Waterfall approach emerged as various researchers and practitioners believed that nowadays the lessons on which the Waterfall approach is based no longer apply (Harrison, 2003; Olsson, 2008; Veenswijk & Berendse, 2008).

The main philosophy of the Agile approach is well expressed in a quote by Austin and Devin (2003): "The difference between a good and bad system is not how well it meets the requirements you know in advance. Meeting requirements is a necessary but insufficient condition for producing an excellent system. What makes a system great, is details that are not specifiable in advance - aspects that must evolve in the making" (p. 93).

The Agile approach accommodates the volatility of requirements and focuses on collaboration between developer and end-users (Demirkan & Nichos., 2008; Garcia-Crespo et al., 2009; Highsmith & Cockburn, 2001; Huo et al., 2004). According to the Agile *manifesto*, individuals and interactions are valued over processes and tools, and responding to change is valued over following a plan (Beck et al., 2001). The Agile approach deals with unstable and volatile requirements by using the following notable techniques: 1) simple planning, 2) short iteration, 3) earlier releases, and 4) frequent customer feedback (Beck, 1999; Huo et al., 2004). The most commonly used Agile approach-based project management methodologies are SCRUM and eXtreme Programming (XP) (Maurer & Melnik, 2006; Frye, 2008). In conclusion, the Agile approach deals with the drawbacks of the Waterfall approach by focusing on responding to change.

Drawbacks of an Agile Approach to ES Implementation Projects

Obviously, the Agile approach is not immune to criticism either. Opponents, mainly from the Waterfall 'supporters' camp, argue that the Agile method is an attempt by software engineers to legitimize software developers' behavior, which is immediately slinging code and producing something that works instead of delivering something that meets all written requirements (Rakitin, 2001; Austin & Devin, 2003). It is culturally embedded within ES implementation project management that in order to develop quality software you need to progress through a sequential, phased life cycle; therefore, the Agile approach will not be honoured (Harrison, 2003).

According to Agile critics it is impossible to develop realistic estimates of the work effort needed to provide a quote because at the beginning of the project, no one knows the entire scope. Moreover, the risk of scope creep (uncontrolled changes in a project's scope) increases significantly due to the lack of detailed requirements documentation (Stephens & Rosenberg, 2003).

In sum, supporters of the Waterfall approach cannot see how users of the Agile approach expect to build anything that satisfies the customers' or end-users' actual demands, without planning up front and carefully analyzing requirements. Supporters of the Agile approach, however, cannot see the point in spending a large amount of time analyzing requirements because they will change anyway (Austin & Devin, 2003). Some refer to this ongoing debate as the methodology war (Jiang & Eberlein, 2008).

As mentioned in the introduction, this article aims to contribute to this debate by starting from the assumption that the Waterfall approach in practice is not as ill-suited to the dynamics of ES projects as its critics claim, and therefore it is interesting to study how a Waterfall approach-based ES project does cope with requirements uncertainty.

RESEARCH METHODOLOGY

In answering the research question, we chose a case study methodology and, in particular, an in-depth case study. We wanted to obtain detailed insights into how a Waterfall approach-based ES project deals in real life with requirements uncertainty throughout a project. For that reason we selected a typical case (Yin, 2003), an ES project at one of the world biggest airline companies referred to as AirRoyal in this article.

AirRoyal implemented an ES using a Waterfall approach from March to November 2008. We spent eight months on data collection and having access on a full-time basis to the ES project, to project documents, meetings, informal discussions, panels, and round tables. In this way we conducted an in-depth case study, interviewing all project managers and team members. The overwhelming amount of empirical data enabled us to study many different aspects of the EIS implementation in relation to each other and to view the process within its total environment. Consequently, our case study research provided us with a greater opportunity than other available methodologies to obtain a holistic view of a specific research phenomenon (Gummersson, 2000).

The AirRoyal ES implementation project was identified as a typical case because the company itself, the project, and the conditions of the project were representative of ES implementation projects in large organizations. Secondly, it concerned the implementation of an off-the-shelf ES supplied by a third-party supplier, who also had other large airlines as customers. Therefore, the type of project we selected is very likely to be comparable to projects being carried out at large organizations.

As preparation for the actual collection of data, we developed a case study protocol and a database in order to organize and document the raw data. The primary data collection was conducted through semi-structured interviews as well as by direct observation during meetings and interactions with project team members and stakeholders throughout the project. The individual semi-structured interviews with the different project managers were used to obtain perceptions and opinions about the project from different angles as the project unfolded. These data were used for verification and to fill gaps where secondary data about the main features of the methodology used by AirRoyal to execute and manage the ES implementation project were incomplete. All observation notes, interviews and gathered documentation were coded and electronically stored.

In sum, during our eight-month study of the ES project at AirRoyal, we used multiple data collection techniques: document analysis, semi-structured interviews, and direct observations. In this way we adhered to the tactic of Yin (2003) to explore multiple sources of evidence during the data collection process. The use of several data collection methods offers the opportunity for (data) triangulation (Benbasat et al., 1987; Saunders et al., 2007). Or as Yin puts it: “A finding in a case study is likely to be much more convincing and accurate if it is based on several different sources of information” (Yin, 2003, p. 98).

Case Introduction

AirRoyal is an international airline operating worldwide with a fleet of almost 600 aircrafts and more than 90 000 employees. AirRoyal carries passengers and airfreight to more than

250 destinations worldwide, either non-stop or via another airport.

Aircraft maintenance is crucial to AirRoyal, and a sophisticated software system is used in that process, so-called Engineering & Maintenance (E&M) software. In 2006 the business case for purchasing a new E&M software system was approved, a system typically fitting the definition of an ES, and its implementation was planned as a step-wise process requesting sub-implementations for every type of aircraft separately. Our research focused on the first ES project that took place in 2008.

AirRoyal decided not to adopt the ES supplier's project methodology but to use their own methodology, based on the Waterfall approach. AirRoyal planned the ES project as a linear process with sequential stages based on the principles of PRINCE2 and PMBOK. There were two differences in the ES project management approach at AirRoyal compared with the definition of the Waterfall approach taken from the literature. First, the design and coding stages were combined in one, while the literature clearly distinguishes these as two separate stages. Second, a global design was prepared during the requirements definition stage and a detailed design during the design stage, while the literature suggests designing both during the design stage.

FINDINGS

The data gathered from the semi-structured interviews, document analysis, and observations, put down in transcripts, summaries, and field notes, during our eight months of full-time access to the ES project at AirRoyal were analysed step-by-step in such a way that we could describe the full project from day-to-day. We triangulated the data, which enabled us to reconstruct the 'story' of the project as presented in the following section, though very much compressed and therefore focusing on the main 'events' per stage.

The Starting-Up and Initiating Project Stage

During the kick-off, all of AirRoyal's ES project stakeholders were present: the project managers, the AirRoyal Information Services department, the AirRoyal Engineering & Maintenance (E&M) business analysts (responsible for defining current requirements), AirRoyal E&M business architects (responsible for designing the final IT landscape) and AirRoyal ES users. From the supplier's side nobody was present at the kick-off. During this event, the participants were informed about the goals and scope of the project, the roles and responsibilities of the participants and the global planning for the project.

After the kick-off, the objectives and scope of the project were evident to the project team members, but a clear view of their responsibilities and the detailed planning of the first stage was lacking, as one of the project team members said:

> *I had no idea what was expected from me and what the approach would be to achieve the objectives of the first stage. [A project team member]*

It was also clear that the project used a Waterfall approach to achieve its objectives (Table 1). The project manager was aware of some of the pitfalls of a Waterfall approach, because during the kick-off the project manager explicitly pointed out the importance of defining realistic requirements within the scope of the project:

> *Be realistic in requirements; strive for an optimum between dreams and realism in the given timeframe. [The project manager]*

The Requirements Definition Stage

The start of the project was the transition from the initiating stage to the requirements defini-

Table 1. Summary of the starting up and initiating stage

Stage of ES project at AirRoyal	*Summary of main 'events'*
Starting up and initiating the project	• Initiating document set-up to enable a clear and controlled start of the project and to plan the overall project. • Sharing the project kick-off with all project team members. • Roles and detailed planning of the first stage were not clear to project team members • Project member warned the team of pitfalls of the approach applied [Waterfall].

tion stage (Table 2). It was clear to the project team members that a number of workshops would be organized to arrive at a final list of requirements, but they did not know what needed to be prepared for these workshops nor their agenda.

To agree on the requirements was a hard and painful process at AirRoyal. After defining the requirements, team members expected them to be evident in the solution, and as such, they found it difficult to accept that some of their requirements were not (yet) included. At the same time, as the project manager remarked, in this way the project team members were involved in the project:

During this stage there were many endless discussions which were time consuming and

Table 2. Summary of the requirements definition stage

Stage of ES project at AirRoyal	*Summary of main 'events'*
Requirements definition stage	• Workshops organized to determine requirements involving the project management, supplier's consultants, AirRoyal's E&M business analysts and architects, and AirRoyal E&M end-users • Business processes as mapped for the 'standard' ES were used to identify the requirements for AirRoyal's ES. • AirRoyal did not prepare any lists of requirements prior to the workshop • Participants perceived this stage as the hardest - to determine the requirements for the business processes of the new ES • The proposed requirements exceeded what was actually needed to achieve the ES project objectives • In general, lengthy and tough discussions were needed, and participants found it difficult to determine what was really required • The supplier's consultants were perceived as not partnering well during the workshops • Based on the requirements (not yet approved), the global design was set up by the supplier. This document proposed an initial design presenting on an abstract level how the requirements would be met • The supplier and AirRoyal ES project management jointly reviewed the global design, and based on this review, the design was refined, and after a hard and painful process, agreement was reached on a number of requirements that would be excluded from the design • Initiating documents were refined based on a clear scope of the project • This stage was formally approved by the project board five months after its start (a two-month delay) • Perceived as a very time-consuming and difficult stage after all, but considered necessary to achieve full involvement and commitment of all project team members

even causing irritation. I let this happen because these discussions were required to keep all the project team members involved in the project and to get user acceptance in the end. If these people would feel ignored, they would do everything to stall the project and they will spread harmful rumours about the ES which will affect the end-user acceptance. [AirRoyal E&M project manager]

Originally, three months were scheduled to complete this stage, but the definition of the requirements turned out to be a more difficult exercise than expected. One of the project team members noted that some perceived this stage as the most difficult one:

Based on my experience from the previous project I think it was not realistic of the project management to schedule only three months for the requirements definition stage, because we experienced the most difficulties during this stage in the previous project. [A project team member]

The Design and Coding Stage

During this stage the approved global design or solution overview was elaborated, which was based on the AirRoyal E&M requirements and intended for building by the supplier (coding). The kick-off meeting for this design and coding stage (Table 3) took place at AirRoyal, again for all stakeholders. The kick-off was very clear, and the participants were ready to start with this stage, as one of the AirRoyal's ES project team members said:

It is a tight schedule, but the approach feels like a very good way to achieve the desired result. [ES project team member]

The project manager and the project team members were truly satisfied with the translation of their requirements into the detailed design. Their requirements were incorporated as agreed upon in the solution overview, and during the first review it was felt that the design was sufficient to enable them to perform their work effectively and efficiently. When asked why they flagged many issues during the previous workshops although they were satisfied with the

Table 3. Summary of the design and coding stage

Stage of ES project at AirRoyal	*Summary of main 'events'*
Design and coding stage	• • The supplier and the ES project management jointly elaborated on the global design, and this elaborated design was built during this stage • Different elements of the global design were elaborated and built in parallel through a number of iterations • First iteration: the supplier builds a prototype of the ES based on the global design for use in the first review session • The supplier and the ES project management jointly reviewed the prototype and evaluated the global design based on this prototype • The ES project management was satisfied with the quality of the first iteration, although many issues were identified on a detailed level • Team leaders were selected to manage the designing and building (resulting from the review sessions) of their assigned element(s) of the entire design • Some of the supplier's team members stayed with the project during designing and building in order to improve communications • The approach towards detailed design worked very well; perfect cooperation between the stakeholders • The second joint review session resulted in an approval of the detailed design, thus the ES was ready to be tested as an integrated system

incorporated requirements, the project manager had an interesting answer:

On a high level (business process) the list of requirements we formulated was complete and these are incorporated satisfactorily, however the devil is in the detail… [Project manager]

The project manager was sure that once the requirements were implemented on a detailed level, they would find unexpected interrelations and unforeseen issues. He realized that it was almost impossible to identify all potential low-level issues during the requirements definition stage. Therefore, resolving issues was incorporated in the planning of the design and coding stage.

During the final review session, which lasted one week, all the use cases, reports, interfaces and adjusted work instructions were approved (all as separate units), and the team leader Training indicated that the training materials were finished to a previously agreed level. Minor issues were still identified during this session, but not all of them were resolved. Those issues were estimated as representing a marginal risk for the success of the project and could be resolved before system testing started. Based on this information, the project manager wrote the end-stage report, which stated that all the units of the ES were built in accordance with the solution overview, except for some small changes. Based on this end-stage report, the project manager received permission to proceed to the next stage.

The Test Stage

During this stage all designed elements were tested to see whether they worked together correctly and to determine if the system was ready for acceptance by the customer. According to the project manager from the supplier's side:

The system test was done to validate that the ES was ready to be accepted by the customer, it would be poor promotion and the customer's confidence in the ES would drop when numerous no-go items were identified during the acceptance test. [a supplier's side project manager]

The system test took three weeks. Based on the results of this test, the supplier had to carry out revision work to fix all the identified issues up to a level that would be sufficient for the system to undergo the acceptance tests. The criterion for approval to proceed to the acceptance test was defined by the supplier in the test plan. After two weeks of revision, the identified issues were resolved sufficiently, and approval was given to proceed to the acceptance test.

The acceptance test took two weeks, and various no-go items were identified even though it had passed the system test. These items were related to errors in the system (not detected during the system test), but also to missing functionality.

A major issue was the performance of the ES, however. Based on the result of the acceptance test, the supplier's side project manager said:

We performed the system test at the supplier's site, but when we did the acceptance test at AirRoyal we ran into all kinds of unexpected issues because of the different test environment. Therefore, we decided to execute future system tests at the customer's site. This requires more resources, but then we can be more certain that the ES is ready for customer acceptance and in this way reduce the risk of a possible disappointment by the customer. [Supplier's side project manager]

As mentioned earlier, some missing functionality was identified during the acceptance test. Most of this functionality could not be traced back to the original requirements, or as one of AirRoyal's ES project team members put it:

We simply did not realize that we missed this functionality until we actually carried out all the real-life scenarios with the ES. [AirRoyal's ES project team member]

This functionality was required by AirRoyal, but it did not have a large impact on the project as it concerned some minor changes in the configuration of the ES. (See Table 4)

The Operational Stage

After the ES was accepted, the next and final stage of the implementation was to get the new ES operational (Table 5). This implied the replacement of the old system (so-called site activation). Consequently, the operation stages included two major topics: training of end-users and the transfer process from the old to the new system. The user training process had already been initiated during the requirements definition stage when the initial training development and deployment plans were set up. These plans were included in the overall project planning and recorded in the refined project charter, which was approved by the project board. According to the AirRoyal team leader responsible for training:

Previous projects had significant delays because of training issues, therefore we initiated the training process as early in the project as possible and it was required that the project board was committed to the training plan. [AirRoyal team leader training]

During this project, user training started at the same time as the joint system and acceptance test. The reason for this was according to the AirRoyal team leader responsible for training:

From the start we knew that it would be an immense job to train all these people within six weeks and therefore we decided to take the risk of possible changes in the training material and

Table 4. Summary of the test stage

Stage of ES project at AirRoyal	*Summary of main 'events'*
Test stage	• An installation record was set up to check the versions of the different units built during designing which were integrated into one system • The supplier and AirRoyal defined their test cases (scenarios) and test plan (summary of all the scenarios and the acceptance criteria) which would be used to perform their tests • The system test did not pass the acceptance criteria, and the supplier had to carry out revision work to fix the identified issues • After two weeks the issues were resolved to a sufficient level to pass the acceptance criteria, and approval was given for the acceptance test • AirRoyal performed the acceptance test, this was done in the 'real-life' setting of the ES (including interfaces and 100% accurate data) • Various 'no-go' items were identified during the acceptance test • The 'no-go' items were mainly unexpected issues revealed by the different test environment compared with the system test. Only a few issues were related to missing functionality • A 'no-go' after the acceptance test required a new cycle of system tests and acceptance tests, a decision taken jointly • A month of revision work was required before this joint system and acceptance test could start • The result of the joint system and acceptance test was within the acceptance criteria, and only some minor issues were identified • The ES was accepted by AirRoyal, and approval by the project board to continue to the next stage was given two months behind schedule

Table 5. Summary of the operational stage

Stage of ES project at AirRoyal	*Summary of main 'events'*
Operational stage	• The two major topics of this stage were user training and site activation • The process of user training was already initiated during the requirements definition stage; the training plan was set up then and included in the project planning • Training material was prepared, and trainers were trained during the design stage; after the test stage, the changes were incorporated • During this stage the training was deployed, and 350 users were trained • When a sufficient level of users were trained, site activation (actual transfer from one system to another) was initiated • The supplier and AirRoyal's project management jointly defined the site activation plan, including the site activation strategy, approach, participants, 'go'/'no-go' criteria, support infrastructure and contingency planning • Old and new systems were running in parallel for a while • Running both systems in parallel was done to compare the performance of the new system with the old one • Only a few issues were identified during the parallel run, and they were mainly related to incorrect data entry by users • After a very smooth site activation and a one-week parallel run, the AirRoyal ES project board arranged a 'go-live' ceremony • The 'go-live' sign was given by a AirRoyal Executive and was perfectly on schedule according to the planning of this particular stage

by doing this gained two additional weeks of training. [AirRoyal team leader training]

After the site activation was completed successfully, the project was ready for the 'go-live' decision. The project board arranged a 'go-live' ceremony which also formed the approval to continue to the next stage. During this ceremony an executive of AirRoyal gave the formal 'go-live' signal, and the old AirRoyal ES system was switched off. The project manager of AirRoyal was very satisfied with the result of this stage as expressed through the following statement:

During the previous stages of this project we had to deviate from the schedule because of various issues, while in contrast, this site activation went extremely smoothly and according to schedule. This shows that it was a long and hard road getting there, but all the effort resulted in a high-quality product. [AirRoyal project manager]

The Closing Stage

During the parallel run and directly after go-live, a transition-to-support team was in place to support the users (Table 6). During the transition-to-support period, a transition manager was ready to manage the support process. The project board received reports at least once a week about the status of this process. The project team was very pleased with the site activation process, as one of the project team members said:

The post go-live support was perfectly arranged, there was always someone to assist the end-users, and issues were resolved very quickly by MXI (the supplier). This, together with the performance of the ES, resulted in a very good acceptance of the ES by the end-users. [A project team member]

The IT service desk of AirRoyal had already been tested for their ability to provide end-user support.

Table 6. Summary of the closing stage

Stage of ES project at AirRoyal	*Summary of main 'events'*
Closing stage	• Ongoing end-user support was installedThe plan for implementing the transition-to-support team and the 'project-exit criteria' was jointly developed by the supplier and AirRoyal's ES project team during the previous stage • The transition-to-support was perfectly organized, was on schedule, and the exit criteria were met four weeks after 'going live' • All project team members were de-charged from the project after the ES project board formally approved the 'end project report' • The project was finished five months later than originally scheduled, nevertheless user acceptance was very high

Finally, an 'end project report' was written, which also listed the lessons learned during this project. The project board reviewed the document and gave approval to close the project down. As a final reflection, the AirRoyal ES project manager stated:

The project was closed down five months later than originally scheduled, nevertheless the user acceptance was great, and we have learned various useful lessons for carrying out future projects. [The AirRoyal ES project manager]

Comparison of Results with the Drawbacks of the Waterfall Approach

Before evaluating the Waterfall approach, we must emphasize the high quality of the implementation process of ES at AirRoyal. The discussion above shows that every implementation stage resulted in a positive outcome, ranging from the project leader's satisfaction to approval by the board. We clearly observed the evolutionary growth of the implementation quality results. The starting-up stage resulted in warnings given by the project member about the pitfalls of the applied approach [Waterfall]. Then, during the requirements definition stage, the supplier and AirRoyal ES project management achieved a joint revision of the design and agreement on a number of requirements. While designing and coding, the stakeholders acknowledged perfect collaboration that led to an approval of the detailed design and ES's readiness to be tested as an integrated system. The test stage showed that the acceptance level was within the expected criteria. The ES was accepted by AirRoyal and approved by the project board. The closing stage showed a delay of five months, but very high user acceptance.

This illustrates the success of the implementation process at AirRoyal that we definitely attribute to the Waterfall approach the company used.

According to the literature, one of the drawbacks of the Waterfall approach is that it is not able to deal with uncertainty and changes in requirements (occurring from *the wicked problem*; McConnell, 2004) due to its high level of formalization and its sequential stages.

The case of AirRoyal showed that *the wicked problem* does occur in practice, and that requirements do change during an ES project. However, the Waterfall-based ES project was able to deal with *the wicked problem* in practice and with uncertainty and changes in requirements throughout the project. Furthermore, the impact of these requirements changes was not as severe as discussed in the literature. We explain this result as due to the fact that in practice ES projects are not 'built from scratch', and industry standards are available through the supplier that only need some specific adjustments. Therefore, in the AirRoyal case there was some sort of 'back-up' to rely on, a type of knowledge base from earlier cases about how to deal with requirement adjustments.

Another interesting difference between the Waterfall approach as described in the literature and the approach in practice is that a major part of the designing is already included in the requirements definition stage. A rudimentary design is established that describes the final design up to a low level of detail, which is sufficient to convey to the end-users how the requirements will be met. This rudimentary design provides end-users with a better idea of the completeness and correctness of the requirements and allows the incorporation of change in the requirements prior to stage approval. Further design work is done in the coding stage. Therefore, the AirRoyal case study showed that there is space for joint design and coding, and that the design goes through a number of early iterations where adjustments to the requirements can be made, providing end-users with an overview of an early release of the ES.

Another acknowledged drawback of the Waterfall-based ES project management approach is that by freezing requirements at the beginning of the project, end-users tend to ask for everything they think they might need, especially if they believe they will only get 'one shot at it'. Or in other words, there is a risk of an increased scope of a project, well beyond what is necessary to meet a project's overall target. This drawback was observed in the AirRoyal ES project as well. However, in practice it seemed to have a limited impact on the project's success. The explanation for this is again that requirements were prepared in practice on an existing industry standard-based ES from the supplier and not from scratch, something which is not always considered in descriptions of the Waterfall approach in the literature.

The third drawback of the Waterfall approach is that its orientation toward planning means that by the time that building of the system has finished, the problem it is supposed to solve has changed; one might build software with few defects this way, but when it is ready it turns out to be the wrong software. In the AirRoyal ES project this drawback did not emerge. It was not a very smooth process to get the ES implemented, but the acceptance of the ES was very high since it met the end-users' expectations. Moreover, the ES dealt with all the problems it needed to solve.

Overall, the Waterfall approach-based ES project at AirRoyal was able to deal with requirements uncertainty and changes. By including the initial part of the design stage (rudimentary design) in the requirements definition stage and the remaining part of the design stage (detailed design) in the coding stage, the Waterfall approach includes some of the features that are said to be so special of the Agile approach.

The design process was done in two iterations. After each design step the ES was actually built according to this design, and therefore the building of the ES occurred in two iterations as well. The two design and building iterations resulted in an early release of the ES. Before the design was formally approved, the end-users had already had an opportunity to actually work with a prototype of the ES. The two design and building iterations and the early releases resulting from this allow for more frequent end-user feedback. The initial design part carried out during the requirements definition stage in AirRoyal's ES project permitted more frequent end-user feedback as well.

It can be concluded that a Waterfall approach-based ES project, if well managed, has features that its opponents claim it lacks. By applying these features, a Waterfall approach-based ES project keeps its original focus on planning but is flexible in responding to changing requirements.

CONCLUSION AND DISCUSSION

The main research question in our study was: how does a Waterfall approach-based ES project deal with requirements uncertainty and changes? By means of an in-depth case study, we collected data about an ES project at a major airline company during eight months of full-time access to project meetings, project documentation, and direct observations and by conducting semi-structured interviews with project managers and team members. Through data triangulation and interviewing different stakeholders, we were able to reconstruct the 'story' of the ES project.

The answer to the research question is that a Waterfall approach-based ES project deviates in practice to certain extent from its features as presented in the literature. These deviations include splitting the design stage; one major part is included in the requirements definition stage, and the other part is included in the coding stage. Based on the requirements, a rudimentary design is set up in the former stage that describes the design with a low level of detail, which is sufficient to convey to the end-user how the requirements will be met.

The other part of the system design (detailed design) is included in the coding stage. Thus, in practice a Waterfall approach-based ES project includes a joint (developers plus end-users) design and coding stage, and involves a number of iterations. In this way, adjustments to the requirements can be made and changes included before a final design is approved.

Furthermore, *the wicked problem* does occur in practice, and thus requirements are subject to change in an ES project. However, a Waterfall approach-based ES project does not turn out in practice to be as 'simplistic' and inflexible as opponents of the approach claim. It does include features typical of an Agile approach, such as iterations, early releases, and frequent end-user feedback, though in a more moderate way.

By doing so, a Waterfall approach-based ES project keeps its original focus on planning, but is more flexible in responding to requirements uncertainty and changes. In the case of AirRoyal this turned out to be sufficient and resulted in successful user acceptance of a new ES.

Interestingly, the impact of requirements uncertainty and changes in practice was less than the literature suggests.

The main conclusion of our research is therefore that the 'demise' of the Waterfall approach for ES implementation projects is not imminent. In practice, this approach seemed to be an appropriate one to successfully implement an ES.

Discussion

The question emerges: where do we go from here? If the Waterfall approach seems suitable for ES development and implementation, then what remains of the criticism brought up by its opponents? Abrahamson et al. (2009) state that "Agile system development methods emerged as a response to the inability of previous plan-driven approaches to handle rapidly changing environments". Our in-depth case study presents evidence, however, that a so-called plan-driven approach like the Waterfall approach is more suited for changing environments than claimed by the Agile community.

Perhaps the literature has contributed to the division between the two camps by using labels that do not fully represent reality. Abrahamson (2009) at least acknowledges that there are shortcomings in Agile systems development research, such as a lack of understanding of what constitutes 'agility'.

The same could hold true for research on plan-driven approaches as well, such as what does a term like 'plan-driven' mean? A possible common sense response could be that plan-driven approaches are still designed by humans, and still need human interaction and collaboration for their execution. Therefore, an intriguing question is whether in reality the dividing line between the plan-driven Waterfall and user-oriented Agile approach is more of a sliding scale.

Limitations and Future Research

To ensure the quality of the findings of this research, we carefully followed the prescribed tactics for single case study research. To improve the validity and reliability of this research, the single case was selected very carefully, multiple sources of evidence and multiple viewpoints on the phenomenon were used, and a research protocol was followed. However, there are some limitations that should be noted. The most significant one concerns generalization of the research findings (the external validity), as single case studies form a limited basis for generalizing. In order to strengthen the findings of this research and generalize them, further case studies should be undertaken. Regardless of this limitation associated with the methodology used, this research still provided various useful findings of interest to both academics and practitioners. Although we were able to generalize to the level of a theory as Yin suggests, future cases would be helpful to understand the contingency factors of the Waterfall approach across different sectors and types of organizations.

This research was limited to the investigation of a Waterfall approach-based methodology for ES implementation projects. Future experimental research design might include an Agile approach-based methodology for ES implementation project management. This would provide the opportunity to investigate the effects of an Agile approach on ES implementation success, and would allow for comparisons.

Finally, this research was driven by our curiosity regarding the high failure rate of ES implementation projects. During the literature review, project management was identified as one of the most significant critical success factors. The proposed modifications to the Waterfall approach-based ES implementation methodology as an outcome of this research should result in greater ES implementation success. For that reason, future research is required to verify that the modified approach towards ES implementation project management will actually result in a greater ES implementation success rate.

REFERENCES

Abrahamsson, P., Conboy, K., & Wang, X. (2009). 'Lots done, more to do': The current state of agile systems development research. *European Journal of Information Systems*, *18*, 281–284. doi:10.1057/ejis.2009.27

Austin, R., & Devin, L. (2003). Beyond requirements: Software making as art. *IEEE Software*, *20*(1), 93–95. doi:10.1109/MS.2003.1159037

Barker, T., & Frolick, M. N. (2003). ERP implementation failure: A case study. *Information Systems Management*, *3*, 43–50. doi:10.1201/1078/43647.20.4.20030901/77292.7

Beck, K. (1999). Embrace change with extreme programming. *IEEE Computer*, 70-77.

Beck, K. (2001). *Manifesto for agile software development*. Retrieved August 06, 2008 from http://www.agilemanifesto.org

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *Management Information Systems Quarterly*, *11*(3), 369–385. doi:10.2307/248684

Brown, W. (2004). Enterprise resource planning (ERP) implementation planning and structure: a recipe for ERP success. In J. S. Whiting, J. Ashworth, & D. Mateik (Eds.), *Annual ACM SIGUCCS Conference on User Services 2004* (Vol. 32, pp. 82-86). Baltimore: AMC.

De Carvalho, R. A., & Tanaka, A. K. (2008). Editorial message to a special track on enterprise information systems. In *Proceedings of the Annual ACM symposium on Applied Computing* (Vol. 23). Ceará, Brazil.

Demirkan, H., & Nichos, J. (2008). IT services project management: lessons learned from a case study in implementation. *International Journal of Project Organisation and Management*, *1*(2), 204–220. doi:10.1504/IJPOM.2008.022192

Frye, C. (2008). Software *development groups take many routes to agile*. Retrieved August 05, 2008 from www.SearchSoftwareQuality.com

Garcia-Crespo, A., Colomno-Palacios, R., Gomez-Berbis, J. M., & Ruano-Mayoral, M. (2009). A project management methodology for commercial software reengineering. *International Journal of Project Organisation and Management*, *1*(3), 253–267. doi:10.1504/IJPOM.2009.027538

Goedecke, D. (2007). The role of the software practitioner in the development of public safety software-intensive systems. In T. Cant (Ed.), *Australian Workshop on Safety Critical Systems and Software and Safety-related Programmable Systems* (Vol. 86, pp. 13-19). Adelaide, SA Australia: Australian Computer Society, Inc.

GummerssonE. (2000). Qualitative methods in management research (2nd ed.). London: Sage Publications Inc.

Harrison, W. (2003). Is software engineering as we know it over the hill? *IEEE Software*, *20*(3), 5–7. doi:10.1109/MS.2003.1199629

Hass, K. B. (2007). The blending of traditional and agile project management. *Project Management World Today*, *9*(5), 1–8.

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *IEEE Computer*, *34*(9), 120–122.

Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004). Software quality and agile methods. In *Proceedings of the Annual International Computer Software and Applications Conference (COMPSAC 2004)* (Vol. 28, pp. 520-525).

Jiang, l., & Eberlein, A. (2008). Towards a framework for understanding the relationships between classical software engineering and agile methodologies. In P. Kruchten & S. Adolf (Eds.), *The 2008 International Conference on Software Engineering* (Vol. 30, pp. 9-14). Leipzig, Germany: ACM.

Jurison, J. (1999). Software project management: The manager's view. *Communications of the Association for Information Systems*, *2*(17), 1–57.

Khalifa, M., & Verner, J. M. (2000). Drivers for software development method usage. *IEEE Transactions on Engineering Management*, *47*(3), 360–369. doi:10.1109/17.865904

Kim, H.-W., & Pan, S. L. (2006). Towards a process model of information systems implementation: the case of customer relationship management (CRM). *The Data Base for Advances in Information Systems*, *37*(1), 59–76.

Laplante, P. A., & Neill, C. J. (2004). The demise of the waterfall model is imminent and other urban myths. *ACM Queue; Tomorrow's Computing Today*, *1*(10), 10–15. doi:10.1145/971564.971573

LockD. (2007). Project Management (9th ed.). Hampshire, UK: Gower.

Maurer, F., & Melnik, G. (2006). Agile methods: moving towards the mainstream of the software industry. In [). Shanghai, China.]. *Proceedings of the International Conference on Software Engineering*, *28*, 1057–1058.

Mendel, B. (1999). Overcoming ERP project hurdles: experts offer tips on avoiding 10 problems that plague many ERP implementation projects. *InfoWorld*, *21*(29), 87.

Middleton, P. (2000). Barriers to the efficient and effective use of information technology. *International Journal of Public Sector Management*, *13*(1), 85–99. doi:10.1108/09513550010334506

Nah, F. F., Lau, J. L., & Kuang, J. (2001). Critical Factors for Successful Implementation of Enterprise Systems. *Business Process Management Journal*, *7*(3), 285–296. doi:10.1108/14637150110392782

Nandhakumar, J., Rossi, M., & Talvinen, J. (2005). The dynamics of contextual forces of ERP implementation. *The Journal of Strategic Information Systems*, *14*, 221–242. doi:10.1016/j.jsis.2005.04.002

Nerur, S., & Balijepally, V. G. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, *50*(3), 79–83. doi:10.1145/1226736.1226739

Olsson, N. O. E. (2008). External and internal flexibility – aligning projects with the business strategy and executing projects efficiently. *International Journal of Project Organisation and Management*, *1*(1), 47–64. doi:10.1504/IJPOM.2008.020028

Parnas, D. L., & Clements, P. C. (1986). A rational design process: how and why to fake it. *IEEE Transactions on Software Engineering*, 1–12.

Parr, A., & Shanks, G. (2000). A model of ERP project implementation. *Journal of Information Technology*, *15*, 289–303. doi:10.1080/02683960010009051

PoppendieckM.PoppendieckT. D. (2003). Lean Software Development: An Agile Toolkit (1st ed.). Boston: Addison-Wesley Professional.

Rakitin, S. (2001). Manifesto Elicits Cynicism. *IEEE Computer*, *34*(12), 4.

SaundersM.LewisP.ThornhillA. (2007). Research Methods for Business Students (4th ed.). Harlow, UK: Pearson Education Limited.

Shanks, G., Parr, A., Hu, B., Corbitt, B., Thanasankit, T., & Seddon, P. B. (2000). Differences in critical success factors in ERP systems implementation in Australia and China: A cultural analysis. In H. R. Hansen, M. Bichler, & H. Mahrer (Eds.), *The European Conference on Information Systems* (Vol. 8, pp. 537-544). Wienna: Wirtschaftsunivsitat Wien.

Somers, T., & Nelson, K. (2001). The impacts of critical success factors across the stages of enterprise resource planning implementations. In []. Washington, DC: IEEE Computer Society.]. *Proceedings of the Hawaii International Conference of System Sciences*, *8*, 1–10.

SommervilleI. (2006). Software engineering (8th ed.). Harlow, UK: Pearson Education.

Stephens, M., & Rosenberg, D. (2003). *Extreme Programming Refactored: The Case Against XP*. USA: APress.

Umble, E. J., Haft, R. R., & Umble, M. M. (2003). Enterprise resource planning: Implementation procedures and critical success factors. *European Journal of Operational Research*, *146*, 241–257. doi:10.1016/S0377-2217(02)00547-7

Umble, E. J., & Umble, M. M. (2002). Avoiding ERP implementation failure. *Industrial Management (Des Plaines)*, *44*(1), 1–25.

Veenswijk, M., & Berensde, M. (2008). Constructing new working practices through project narratives. *International Journal of Project Organisation and Management*, *1*(1), 65–85. doi:10.1504/IJPOM.2008.020029

Ward, J., Hemingway, C., & Daniel, E. (2005). A framework for addressing the organisational issues of enterprise systems implementation. *The Journal of Strategic Information Systems*, *14*, 97–119. doi:10.1016/j.jsis.2005.04.005

White, D., & Fortune, J. (2002). Current Practice in Project Management - an empirical study. *International Journal of Project Management*, *20*, 1–11. doi:10.1016/S0263-7863(00)00029-6

YinR. K. (2003). Case study research Design and Methods (3rd ed.). London: Sage Publications Inc.

Huub J.M. Ruël is an assistant professor of International Human Resource Management at the University of Twente (The Netherlands), before that – at the American University of Beirut (Lebanon), Kuwait-Maastricht Business School (Kuwait), and the University of Utrecht (The Netherlands). He holds a PhD in Business Administration/Human Resource Management. His thesis focused on implementation of IT's in office-environments. His main research focus is Human Resource Information Systems/e-HRM in global companies, combining the IT, HRM and International Management field. In 2004 he published a book e-HRM: innovation or irration? together with dr. Tanya Bondarouk, in which the results of e-HRM implementation in five large international companies were presented. Huub J.M. Ruël has authored several follow-up publications on this topic since then.

Tanya Bondarouk is an assistant professor of Human Resource Management at the University of Twente, the Netherlands. She holds two PhDs: in Didactics (1997) and Business Administration/HRM (2004). Since 2002 she has been busy with the emerging research area of Electronic HRM. Her main publications concern an integration of Human Resource Management and social aspects of Information Technology Implementations. Her research covers both private and public sectors and deals with a variety of areas such as the implementation of e-HRM, management of HR-IT change, HRM contribution to IT projects, roles of line managers in e-HRM, implementation of HR Shared Service Centers. She has conducted research projects with the Dutch Ministry of Interior and Kingdom Relations, Dow Chemical, Ford, IBM, ABN AMRO bank, Shell, Unit4Agresso. Among her current research projects are Implementation of HR

Shared Service Centers at the Dutch Ministry of Defense, Large Non-academic Hospital, and the Belgian Federal Public Health Service. Since 2006 she is involved in organizing European Academic Workshops on e-HRM, and International Workshops on HRIS.

Stefan Smink is a project management trainee at Sodexo Altys, Amsterdam Airport, the Netherlands. He holds a professional bachelor degree (BEng) from the University of Applied Sciences Windesheim, Zwolle, the Netherlands and a Master of Science degree in Business Administration (MScBA) from the University of Twente, the Netherlands.