



Expecting the unexpected during ERP implementations: a complexity view

Guy Janssens

Open Universiteit, Faculty of Science
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
guy.janssens@ou.nl

Rob Kusters

Open Universiteit, Faculty of Science
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
rob.kusters@ou.nl

Harry Martin

Open Universiteit, Faculty of Science
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
harry.martin@ou.nl

Abstract:

Implementing an ERP (Enterprise Resource Planning) system is a complex, risky, time-consuming, and very expensive affair. Unfortunately, ERP implementations are often still over budget and time, and below expectations. Ticking off critical success factors (CSFs) and risks is supposed to take care of all intricacies during an implementation. However, complexity theory suggests no perfect foresighted knowledge can exist and one should always be prepared for new and unexpected events happening (“unknown unknowns”). Currently, ERP research does not explicitly address this unexpected behavioral aspect of complexity. Therefore, it seems relevant to explore whether this unexpected complexity aspect of ERP implementations can be observed in actual ERP implementations. We demonstrate through an in-depth and structured case analysis that a normal, well-planned, and managed ERP project shows indeed unexpected behavior. That is to say, totally unforeseen major problems appear. From our observations, it is evident that ERP implementations can show significant unexpected behavior despite the best of knowledge, proper preparation, and project management practice. It seems relevant to perform more research into the relevance of appropriate control mechanisms based on acceptance of the inherent complex, i.e. unpredictable nature of ERP implementations. This awareness should complement existing mechanisms as CSFs and risks.

Keywords:

ERP; ERP implementation; ERP implementation complexity; ERP case research; complexity.

DOI: 10.12821/ijispm080404

Manuscript received: 12 March 2020

Manuscript accepted: 8 September 2020

1. Introduction

Despite some 20 years of practice and research, ERP (enterprise resource planning) implementations are often still over budget and time, and below expectations of stakeholders [1]. According to Amid, et al. [2]: *“It is said that about 70% of ERP implementations fail to deliver anticipated benefits and three-quarters of these projects are unsuccessful. These projects are, on average, 78% over budget, took 2.5 times longer than intended and delivered only 30% of promised benefit”*. More recently a report from Panorama consulting [3] shows that in 2019 58% of the cases the project took longer than expected, 45% exceeded budget.

Therefore, the search for means and methods that can make implementations of ERP systems more successful continues to be relevant. A quote from Donald Rumsfeld (a retired American politician) inspired the ERP research in this article: *“Reports that say that something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say, we know there are some things we do not know. But there are also unknown unknowns- the ones we don't know we don't know”*. The concepts “known knowns”, “known unknowns” and “unknown unknowns” might prove to be of value in ERP implementation research approaches. A well-known approach for IT projects is research into critical success factors (CSFs) [4]. CSFs can be considered as ‘known knowns’ [5] for ERP implementation projects. The basic idea behind these ‘known knowns’ factors is that CSFs are expected to provide focus points that need to be addressed thoroughly to achieve a positive effect on the outcome and success of an ERP implementation. Using lists with critical success factors is widely popular in many areas, in particular in project management [6].

Also, considerable research has been performed into risks for ERP implementations. A project risk is an uncertain event or condition that, if it occurs, has a positive or negative effect on a project’s objectives [7]. Risks can be identified in a project, but it is unknown whether or when an event caused by this risk will ever occur and to what extent. Therefore, risks can be considered ‘known unknowns’.

Given all validated and exhaustive ERP CSFs and risks lists: why are ERP implementation projects still not more successful? Could it be that practice simply ignores existence and influence of ‘unknown unknowns’. Is it possible that a third category, i.e. ‘unknown-unknowns’, as a generally ignored category, may have some effect in the success of ERP implementations? However, before research into ‘unknown unknowns’ can start, we should answer a fundamental question, i.e. do ‘unknown unknowns’ significantly influence ERP projects? To our knowledge, there is no ERP research which clearly proves the existence of these ‘unknown unknowns’ and their influence on success in ERP implementation projects. We surmise that in an ERP project ‘unknown unknowns’ manifests themselves as emerging unexpected troublesome issues.

Unexpected behavior of a system (e.g., an ERP implementation project) is a fundamental and well-identified phenomenon in complexity theory [8, 9]. Complexity theory suggests that in complex systems, no perfect foresighted knowledge exists and one should always be prepared for new and unexpected events (a.k.a. the “unknown-unknowns”) [10, 11]. It therefore seems sensible to apply complexity theory as guidance for research into ‘unknown unknowns’, i.e. the unexpected behavior of ERP projects. Therefore, we will explicitly address the complexity of ERP implementations, and try to identify unexpected behavior (i.e. the ‘unknown unknowns’) in actual ERP implementations.

To do that, we will demonstrate through an in-depth and structured case analysis that a well-managed ERP project does indeed show these ‘unknown unknowns’, causing significant issues. Given our findings, we expect similar issues may be observed in other ERP implementation projects as well, and that there is sufficient reason to upgrade current control practices accordingly.

Following, first we present the concepts of ERP implementation, complexity of ERP implementations and motivations for research into unexpectedness of ERP implementations. Next, is presented the case study. After that, are presented the obtained results. Finally, are presented the conclusions as well as suggestions for future research.

2. Related work

Complexity is a concept that requires further explanation, as it is usually only used as a synonym for the concepts difficult or complicated in day-to-day language. However, the construct of complexity in research has its own value and meaning. Therefore, in this section we will first discuss the link between the construct of complexity and the topic ERP implementation. Next, we will review current literature regarding the unexpected aspect of complexity of ERP implementations.

2.1 What is complexity of ERP implementations?

ERP implementations introduce an entire ERP information system or parts of it into an organization. Although literature clearly describes and defines what an ERP information system embodies, we could not find an explicit definition regarding an “ERP implementation”.

In most cases, ERP implementations are discussed and handled as projects, as is indicated by a vast amount of research into ERP projects as a subject [12-14]. Hence, we turn to project management as a base for our definition. In project management the building blocks or elements of projects are well known. A project always comprises activities. These activities need resources (human and non-human) which perform these activities. The results of these activities are products (and/or services). Products which are required by or are of interest to stakeholders. These activities, products and stakeholders are interconnected to each other and can influence each other [15]. Besides these project aspects of an ERP implementation, a fundamental aspect is its organizational impact [16, 17]. In most cases, using an ERP system changes the way an organization operates by altering its business processes. Research shows that organizational change [18, 19], and as a consequence change management, is a very important aspect of an ERP implementation in order to successfully implement an ERP system [20-23]

Thus, we propose to define an ERP implementation:

ERP implementation

All activities undertaken, resources needed, (sub)products produced, stakeholders, and their interrelationships to introduce (parts of) an ERP information system in an organization and the associated necessary organizational changes.

Our research aims to enhance the interpretation of complexity research in relation with ERP implementation. To do that, we also need to define ERP implementation complexity.

Edmonds profoundly explored the construct of complexity [8] and defined it as: *‘That property of a model which makes it difficult to formulate its overall behavior in a given language, even when given reasonably complete information about its atomic components and their interrelations.’*

Based on the definition of complexity by Edmonds [8] and our research definition of ERP implementation, we combine both into one definition as follows:

Complexity of an ERP implementation

‘That property of an ERP implementation which makes it difficult to formulate its overall behavior, even when given almost complete information about its activities, resources, (sub) products, stakeholders, their interrelations, and the associated necessary organizational changes’.

2.2 Use of the unexpectancy complexity aspect in current ERP implementation research

Since complexity behavior during ERP implementations forms the basis for our research, it is important to know to what extent current ERP research recognizes the impact of complexity in the topic of ERP implementation.

We performed a detailed, specific, and methodical search to retrieve a focused literature set on the impact of complexity in the topic of ERP implementation. We aimed at discovering literature showing a firm grasp of using “complexity” as a

key term in complexity theory as opposed to its day-to-day language use and how complexity is linked with ERP implementation. For instance: is it discussed methodically or merely mentioned descriptively?

We formulated the following assumptions about papers we were interested in:

1. A paper mainly discussing ERP will have the string “Enterprise Resource Planning” in its abstract.
2. A paper discussing ERP and complexity can have the strings “complex” or “complexity” in the title and/or abstract, but will always have the strings “complex” or “complexity” in its full-text.
3. A paper discussing ERP, complexity, and the construct of complexity itself probably contains definitions of complexity in the full-text.

In our search strategy, we combined these three assumptions by appropriate keywords and used the following search string:

```
((Abstract:(“Enterprise Resource Planning”)) NOT (Abstract:(“complex”)) NOT (Abstract:(“complexity”))) AND (“define complex”) OR (“define complexity”) OR (“definition of complex”) OR (“definition of complexity”) OR (“what is complex”) OR (“what is complexity”) OR (“complexity theory”) OR (“complex project”) OR (“unknown unknowns”))
```

We searched in the literature databases of the Open Universiteit, which included amongst others: EBSCO, ACM, IEEE, DOAJ, AIS eLibrary, Scencedirect and Springerlink.

At the time of this writing we retrieved in total 138 relevant papers.

The majority of the 138 reviewed papers uses the term *complex* or *complexity* related to ERP, or even related to non-ERP subjects, but do not define its meaning. They use these terms mainly as a qualitative property or characterization related to ERP systems, ERP projects, ERP implementation and ERP environment. Seven papers refer in their text to complexity theory. Of these, only five papers discuss the term *complexity* explicitly related to ERP implementation, as intended in our research. Of these, three [24-26] did not discuss the complexity of ERP implementation projects itself. Only two papers [27, 28] discussed complexity related to ERP implementation projects. Ghosh and Skibniewski [24] concentrated on indicating what complexity of an ERP project is, but not how to approach this complexity in ERP implementation research.

Fontana and Neto [27] discuss the change of complexity of organizations resulting from ERP implementations. Although they discuss complexity theory more extensively, they did not address the issue of unexpectedness in ERP implementations. All five papers we retrieved consider using complexity theory for research into ERP implementation to be useful in some way. Nevertheless, most of the retrieved ERP implementation research uses complexity descriptively rather than analytically.

As complexity theory suggests, we consider the unexpectedness of ERP implementations an important aspect which needs to be managed during an implementation project. However, we could not find any ERP research which clearly recognizes this. Therefore, it seems relevant to explore whether this unexpected complexity aspect of ERP implementations exists at all in practice.

3. Research design

If we consider ERP implementations following Edmonds definition [8] and Manson’s complexity typology [11], it can be expected that an ERP implementation, being a social system, despite proper planning and management, will show unexpected behavior. Consequently, even in well managed and planned ERP projects by state-of-the-art knowledge and best practices, unexpected issues may arise and remain potentially undetected, possibly with increasing impact over time.

In our research, we will try to detect whether this unexpected behavior of ERP implementations can be discovered in practice. To do this, we have performed a case study, in which we evaluated an actual ERP implementation. Various project documents were scanned and a variety of stakeholders were interviewed for unexpected anomalies and suspect

events, i.e. issues which cannot be solved within the boundaries, abilities, and authorities of an ERP project itself. Following a deterministic complexity paradigm [11], all issues should be solved within the scope and boundaries of the project with proper planning and use of relevant experience from other ERP implementations. In general, we searched for issues which would need a higher level of involvement/authority in decision making exceeding a projects scope.

Our goal in this study is to discover the existence of unexpected behavior by discovering unexpected issues that are clearly out-of-scope of the implementation project and can only be solved outside the confinement of the project itself.

3.1 Issues and events

To achieve the research objective, we need to identify unexpected issues in practice. For this we will look at observable events. Analysis of these events can lead to the identification of underlying (not readily observable) issues. For each of these issues we then need to judge if their occurrence was 'unexpected'.

For example, an issue could be the refusal of a department to cooperate in an ERP implementation project. This refusal can cause the project to fail. Such an *issue* can be revealed by one or more *events*. Some examples of events are: a project activity is overdue, a resource is lacking, an alert is given that the functionality of the new ERP system cannot support a certain part of an organization, or an angry email is received from a manager stating that his department no longer will take part in the project.

Such vents are signs or symptoms of an underlying issue. For instance, if a project member calls in sick (an event!), the underlying issue could a touch of flu, but could also be a symptom of a poor relationship between the project manager and project member. Events call for a deliberate analysis for what issues have caused the event and to decide, whether and how to act. If the actions or lack of actions do not lead to a solution of an issue, the events may recur, or new events may be triggered. The issue is *recurring* and the chain of decision making, decisions and action recurs as well.

To validate our proposition that unexpected issues can have profound detrimental effects to an ERP implementation project we have to find such unexpected issues in actual ERP implementation projects. However, we expect that a substantial part of these issues in an ERP implementation project cannot be found. Some are never detected. Some are simply quickly forgotten or oppressed in people's memory, or their impact has not been fully comprehended or formally registered. Therefore, we must settle for the evident "big" issues that leave traceable evidence of their existence. With hindsight, the help of stakeholder "witnesses", formal project documentation and some detective work, we may uncover a few of these high-impact issues.

We expect that these events related to recurring issues can be more easily traced in documents and that participants in the ERP implementation project remember these events, so we will focus on detecting these *recurring unexpected issues* as main evidence for the existence of unexpected issues in our case study. We will focus on two characteristics of issues.

First, when events occur, and the underlying issue which causes these events is unclear to the ERP project, there is a probability that ineffective actions have been taken for that issue. These ineffective actions demonstrate that an issue has not been solved definitively, triggering subsequent events on the same issue. On the other hand, if events occur and the underlying issue is clear, we expect a higher probability that proper actions have been taken for that issue, and the issue has been solved. Therefore, we differentiate in issues that are clear and issues that are unclear (i.e. not well understood) to the ERP implementation project stakeholders.

Second, if the authority for solving the issue is unclear or not generally accepted, then we assume that the ERP implementation project has not been properly set up with appropriate authorities to solve these types of issues. If such issues would have been anticipated, proper authority would have been allocated to a project in the first place. Again, subsequent events on similar issues may reoccur. Therefore, we also differentiate in issues for which the decision authority is clear and present, and issues for which the decision authority is unclear or missing.

If we differentiate issues by these two characteristics: the clarity of an issue and the clarity and presence of an appropriate decision authority, then we can distinguish four types of issues as shown in table 1. In table 1 we also showed for every type whether it could be expected that that type of issue will trigger subsequent events.

Table 1. Types of issues

	decision authority clear and present (inside or outside the project boundaries)	decision authority unclear or missing
issue clear	<p style="text-align: center;">Type A</p> <p>In an ideal situation, an issue is clear, and the authority for deciding on taking actions to solve that issue is clearly known.</p>	<p style="text-align: center;">Type C</p> <p>In a situation where an issue is clear, but the authority for deciding on taking actions is unclear, an issue might not be solved.</p>
issue unclear	<p style="text-align: center;">Type B</p> <p>If an issue is unclear, but the authority for deciding on taking actions is clear, then we expect an issue will not be solved right away and keeps on recurring. However, when more events occur related to the same issue, an issue might become clear in a way that appropriate decisions can be made and proper actions performed.</p>	<p style="text-align: center;">Type D</p> <p>In the worst of situations, an issue is not clear and also the authority for deciding on taking actions is not clear.</p>

We labeled these issues type A, B, C and D.

Type C and D are the types of issues:

1. which we expect to generate multiple events;
2. due to unclarity of decision authority we can easily classify as unexpected issues;
3. we are most interested in, because they are easier to detect than other types of issues.

Therefore, C and D types of issues will be looked for in an actual ERP implementation project.

It is important to note that a detected list of events/issues is merely a minimal list and not the complete list of issues/events in this ERP implementation project case because:

- Not all events/issues may have been recorded in the documentation;
- Not all documentation may have been retrieved or made available;
- Not all events/issues may have been detected by the researcher in the documentation.

3.2 Case selection, information sources and research steps

We performed an intensive case study encompassing an actual ERP implementation project. In this case study, we gathered information from project participants and relevant documents in several steps. Arguing from this information we tried to determine whether this ERP implementation project had unexpected recurring issues.

For our appropriate research case we aimed for an ERP implementation with the following characteristics:

1. The ERP implementation must have taken place in a professional organization with a professional project management organization and up-to-date skilled and experienced project managers;
2. The project, as ERP projects mainly are, should be considered an important, large, and costly project, i.e. the stakes are high, ensuring significant management attention and a sound project management setup is required;

3. The ERP implementations has been completed fairly recently to ensure good recollection with the stakeholders (preferably no longer than three years ago);
4. The organization should be transparent and interested in this type of research, with no restraints or taboos;
5. We expect that for this type of ERP implementation certain information sources should be available to obtain information about issues;
6. Stakeholders with knowledge of and experience from that particular ERP implementation;
7. Documents which hold information about this particular ERP implementation: Project definition reports, management reports and other relevant reports dealing with the ERP implementation project.

Performing a case study as a primary research approach requires considerable effort and a sound case study selection is justified. Any selection process raises issues concerning external validity. In this research project we opted for purposive sampling [29] as the key motivation in case selection. Consequently, the external validity is bound to the characteristics of the selected case study. In turn, most of the characteristics are “purposively” chosen to maximize the chances to identify “known-unknowns”, in a rather common and realistic context. Given the summary of characteristics, this means that we can claim external validity for contexts that can be characterized by the characteristics (1, 2, 4, 5, 6).

To retrieve information from the ERP project participants and documentation, we divided this case study into several steps. These steps are shown in figure 1.

Internal Validity

We used triangulation in our research design to assure that our findings are genuinely based on a critical investigation of all available data and do not depend on a few “convenient” examples. We analyzed all available ERP project related documents and inquired within the interviews after all issues of the project. As we wanted not to reveal the purpose of our research to the interviewees, we did not specifically ask for recurring issues, but we asked for issues/problems in general. For every potential recurring issue, we tried to detect objectively as possible (by triangulation) whether it was an issue which unexpectedly occurred indeed. By discussing every potential recurring issue in separate interviews in depth and documenting and verifying the results, we transparently underpinned the conclusions, i.e. whether unexpected issues existed in this ERP project.

We tried to enhance internal validity by first selecting an ERP implementation which consisted of several sub-projects, with their specific project leaders, steering committees, scopes et cetera, to increase the likelihood of finding events and issues. We then retrieved all issues (A, B, C and D issues) and only after applying the definition of recurring issues to it; we typified an issue as relevant or not.

A final check was performed by our contact person, independently from the researchers. Also, our findings and any intermediate steps were recorded for verification by others. To maximize a sound designation of issues to the types A, B, C or D, we made use of several well-informed persons as a source of triangulation.

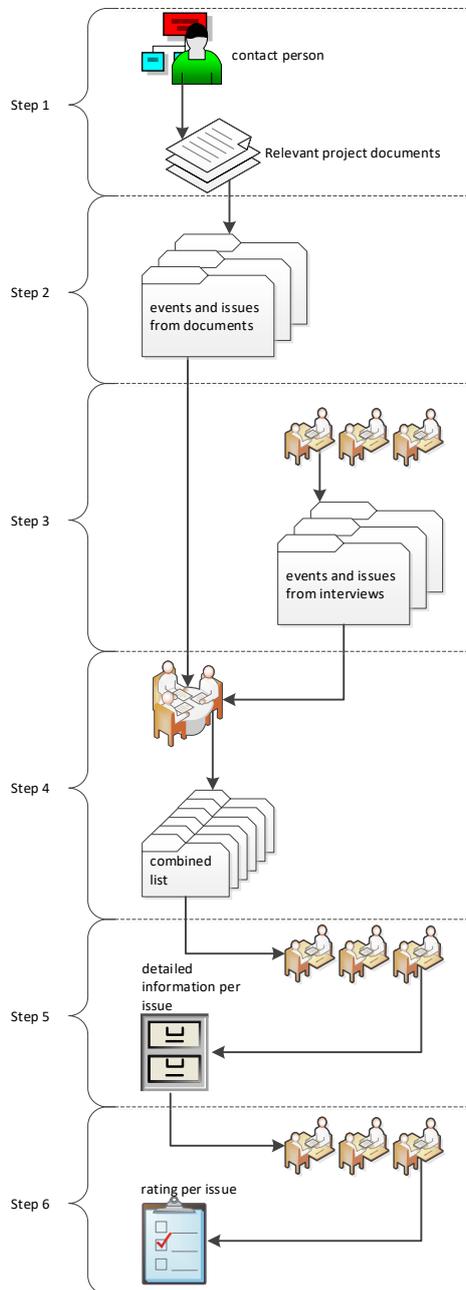
External Validity

We carefully selected a case by purposeful sampling [29], to be able to apply our findings in other comparable situations (see section 1.4). But of course, we are aware that our results might not apply to situations which considerable deviate from our selected case type. In that case, more research into these deviating types should be performed.

Reliability

To improve a correct representation of the information we retrieved from the interviewees, we recorded every interview. Based on this recording, we carefully prepared a written summary for the interviewee which the interviewee confirmed and/or enhanced. Also, the selected issues by the researcher from the documentation was by the contact person as being a representative list of issues belonging to the ERP implementation project.

An important aspect of reliability is transparency of the research protocol. Therefore, we discussed the research steps in a previous section. Furthermore, the results from every step are available (privacy guidelines permitting) through the first author.

**Step 1: Gain access to a contact person and assuring confidentiality**

After selection and admittance to the case, in this step a main contact person within the case was requested. This contact person was well informed about the ERP project and had access to all managers, project participants and documents.

Step 2: Extraction of events and issues from project documentation

By studying the content of the provided documents, the researcher valuated these for potential events and issues. The researcher created a list of retrieved events/issues. The contact person from the case organization revised and validated this list.

Step 3: Extraction events and issues by interviews

Confirmation of this list and additional issues/events was obtained from ERP implementation participants. In consultation with the contact person relevant participants were selected with an extensive overview of the project.

This step served two purposes:

1. Discovery of events/issues not documented or missed in the documentation.
2. Confirmation of events/issues already discovered in step 2.

Step 4: Integration of events/issue lists

The main researcher combined the lists from step 2 and 3 into one list, which could be considered a fair representation of events/issues from the case. The research team and the contact person of the case checked the list.

Step 5: Profound potential C/D issue information retrieval

The information from the interviews and information from the documents was combined in a structured file per issue:

1. Background of issue;
2. Description of issue and events;
3. Description of settlement issue by decision-making processes, decisions, actions and if the issue was solved.

Step 6: Issue rating

Based on the gathered detailed information about the issues in the file from step 5, the research team classified the issues in A, B, C or D. The contact person and if a well-informed ERP stakeholder verified the definitive classification of the issues in A, B, C or D.

Fig. 1. Research steps

4. Research results

4.1 Case description

We were very fortunate to have obtained approval and cooperation from a large government agency in the Netherlands, which fits our pre-set requirements. This government agency had implemented and still was implementing ERP by means of a professional project organization. The selected case is an ERP implementation that affected large parts of the government agency itself.

This government agency already used Oracle's PeopleSoft ERP software [30] for financial management. The organization started with projects for implementing more PeopleSoft applications. The goal was to phase out several expensive legacy systems and create an integrated, cost and support efficient information system.

The government agency had a department dedicated to the implementation and support of ERP systems. This department contained several experienced IT project managers. Also, the government agency had a program department where every IT related project and subproject was assessed and monitored. Project management was performed in accordance with and by the standards of the PRINCE2 methodology [31]. The project managers had to be PRINCE2 certified.

Also, they contracted a consultancy firm to advise and assist for parts of the ERP implementation where they lacked knowledge and experience with PeopleSoft applications. The project managers were internal employees. The general managers of the staff services initiated and supported this ERP implementation.

Considering all these characteristics of the case project organization, project manager's profiles and in place project management standards, all case requirements were met. Consequently, we have ascertained that the case organization had a professional project organization and worked according to professional standards, and is suitable for our research goal.

The government agency was very much interested in our research and broadly recognized the problems and risks associated with large ERP implementations. They agreed to fully assist in the case study.

4.2 Results from main research steps

Step 1: Gain access to a contact person and assuring confidentiality

A suitable contact person was assigned to our research.

Step 2: Extraction of events and issues from project documentation

The researcher gained access to 129 documents, which were all electronic files. These documents consisted of the following types: agenda, audit, decision document, decision-making list, communication plan, leaflet, mail, memo, minutes, design document, presentation, program plan, project handbook, progress report and plan.

From these documents, 72 possible events or issues were distilled.

Step 3: Extraction events and issues by interviews

To get a clear picture of the project goals and settings, four authoritative participants (Table 2) were asked to provide an extensive overview of the project. All participants had at least a bachelor educational level.

Table 2. Respondents

Position	Experience in ERP implementations (years)	Assigned to current position (years)
Director of facility management and purchasing	2.5	5
Manager ERP competence center	16	2
Manager project professionals	15	4
Project manager ERP	2	2

A comprehensive list of events/issues was compiled via extensive interview sessions with the help of these participants.

Step 4: Integration of events/issue lists

The 72 distilled events by scanning the official documentation were also verified by this group of participants. In the end, an additional 42 issues/events were captured, which produced a list of 114 issues.

Step 5: Profound potential C/D issue information retrieval

Initially, it was intended to analyze every retrieved issue by performing interviews in detail to indicate which issues were from C or D type. However, the list of 114 discovered issues from step 4 was simply too big for detailed interviews. Therefore, a decision was made to add an intermediate step. We aimed at verifying the very existence of unexpected high impact issues that remained unsolved and not analyzing all 114 issues in detail. Therefore, a preliminary selection of 11 potential C or D issues was selected by the research team and discussed and verified with the contact person from the organization.

These issues were:

1. Too wide scope subproject A;
2. Failure to start 'Time registration project';
3. Part of organization does not want to change its ordering process in accordance with processes designed in subproject A;
4. Blueprint Phase for subproject A was very costly and time-consuming;
5. Additional costs and delay of activities within subproject A by performing line activities within this subproject;
6. Project B expires over budget and time because the short-term solution chosen is not optimal;
7. Low acceptance rate and not using PeopleSoft leave module;
8. Transfer of e-HRM to management failed;
9. Dutch railways business card solution too complex for support department;
10. Missing functionality in course administration;
11. single sign-on fails.

Example of a C/D issue (issue 7)

In this case the introduction of the timesheet module of their ERP system was considered a replacement of an in-use standard Excel spreadsheet. Managers and employees used their individual copy of this Excel spreadsheet to keep track of hours worked and leaves. Only during transferal of this in essence simple functionality to the ERP system, the project discovered the spreadsheet was also used as an informal rewarding system for managers to their subordinates. This informal and flexible system clearly was not possible in their chosen ERP system because of authorizations and the complete integration of data with other modules. Implementing this functionality would have led to inconsistencies. As a result, this hidden and informal but yet significant business process for rewards became a critical obstacle to acceptance of this part of that ERP module. The project management team did not have the authorization to change or end this informal reward method, or any means to ensure that the new ERP system would support this reward method.

This issue was regarded in the project as a serious problem with consequences for the project. Some departments refused to use this part of the system.

This list shows a short title for every issue. A detailed description per issue (in Dutch) can be requested from the first author.

These 11 potential C/D issues were analyzed in four interviews for final verification. Four individuals to which the contact person attributed detailed knowledge about these issues were interviewed. To increase reliability, we assured that at least two individuals could provide information about every issue.

The information from the interviews and information from the documents was combined in a separate structured file on per issue.

Step 6: Issue rating

Based on this detailed filed information of the 11 issues in the file, the definitive classification of the potential C/D issues in A, B, C or D was performed.

The researcher first performed the classification based on the issue information. After that, the two co-researchers individually and independently checked the interpretations made and checked the consistency of the data based on the available filed information.

As a last check, a contact person at the case organization also performed independently the classification. This contact person was a former leading consultant of the government agency. The consultant advised during the implementation and therefore, had a complete overview of the project.

An issue was only designated positively as a C/D type issue if **all participants were in complete agreement**.

This is shown in Table 3.

Table 3. Results from issue rating

	Issue										
	1	2	3	4	5	6	7	8	9	10	11
Researchers	y	y	y	y	n	y	y	n	y	n	n
Former consultant	y	y	y	y	n	y	y	n	y	y	n
Contact person	y	y	y	y	y	y	y	y	y	y	y
Final rating	Y	Y	Y	Y	N	Y	Y	N	Y	N	N

In the end, rating resulted in **7 issues of type C/D (Issue 1, 2, 3, 4, 6, 7 and 9)**

5. Discussion and conclusion

5.1 Conclusions

Our goal in this study was to detect the existence of unexpected issues that are clearly out-of-scope of an ERP implementation project. In our research we performed a case study to detect if such unexpected behavior of an ERP implementation exists.

Even though the scope of the sources of information has been limited for practical reasons, we discovered seven unexpected serious problems in this ERP implementation project. Seven issues, each of which had a significant impact. These issues could never have been avoided or detected in advance, for example, by using CSFs.

In addition, the current knowledge base of the CSF could easily be expanded to include additional issues and related discoveries ("known knowns"), but a list of CSFs naturally always lags behind new but undiscovered issues. This also applies to risks ("known unknowns"). In other words, by definition, CSFs and risks will always be incomplete in complex environments such as ERP implementations. To avoid unexpected issues, it is necessary to prepare for the unexpected for example by:

- Installing a thorough review process of issues discovered, especially if they seem to recur.
- Accepting that issues may be rooted in deeper or external out-of-project scope issues, not directly visible by a project team.

- Recognizing and accepting that a project manager can escalate such issues to higher managerial authorities, who need to take actions accordingly. Such an escalation should be professional project management practice, rather than a sign of a project managers incompetence.
- Recognizing that the world is dynamic. This always requires an open mind to expect truly the unexpected.

5.2 Discussion

We performed our case study in a large organization with a significant internal IT and other support staff. This IT and support staff were very experienced with implementations in a project structure. Also, the IT and support staff have been trained well in project management and in subject matter topics. The key players have an academic background and are assumed to be proficient in analyzing project issues. Therefore, we have selected a typical ERP implementation, which supports the external validity of our results.

Our results also show that there are recurring issues within several of the sub-projects in this ERP implementation, which supports the internal validity of our results.

In the Netherlands, government agencies must be transparent by law. The event detection depended on the availability of formal documents and the recollection of individuals who each had a specific role and perspective on the entire implementation project. Despite the required transparency of government agencies and good intentions, it cannot be avoided that some information may be lost or never recorded properly, because it was not part of the daily routine, forgotten or simply regarded as insignificant at the time. In our research we focused on negative events to detect issues. However, also positive events can exist, for instance: being ahead of schedule, or becoming aware that the functionality of the ERP software also can support other processes in the organization. These may also relate to unexpected issues. Despite of these limitations in detecting unexpected issues, we discovered seven very challenging issues which could have easily jeopardized the entire implementation. Therefore, it is fair to assume we may have just discovered the rather obvious ones, the so called “tip of the iceberg”. Potentially, if we were able to access more information and people we could have detected even more hidden detrimental issues, only further strengthening our conclusion.

We have executed a single case study while explicitly taking measures to counter the standard research limitations inherent to such a study. Key issues are related to the subjective nature of data acquisition and analysis. To handle this, we have used triangulation for both data gathering and analysis, as described above. This, coupled with an explicit awareness to the issue is expected to reduce any bias issues that might arise. Similarly, generalization of results is often an issue with this type of research. After all, the conclusions are based on a single case study, albeit meeting broad selection criteria. However, given the relatively large number of issues found, we expect that similar ‘unknown unknowns’ will occur elsewhere. How widely is of course outside the scope of this study and requires further research?

A question that might arise is: if we see enough occurrences of ‘unknown unknowns’, can this then be used to enhance e.g. risk or CSF-approaches. By e.g. transferring them into ‘known unknowns’. In essence, this is how these approaches were developed in the first place and resulted in useful contributions. However, in our opinion there is a limit to how far this can be stretched. The examples found in the case study were very specific and truly unexpected. Looking at them from a higher level of abstraction, it is surely possible to classify them under an already existing and recognized CFS or risk. But that did not help the organisation which looked at CSFs and risks before starting the project. The higher level of abstraction hides the specificity of an ‘unknown unknown’. Hiding it in plain sight as it were. There are probably just too many things that can go wrong in such an extensive project. That is why imagining them in advance is an inadequate remedy.

5.3 Further research

In this research, we only searched for negative issues which disturb an ERP implementation. It seems worthwhile to perform also research into positive issues which as discussed previously also can disturb an ERP implementation. It might be relevant to determine whether the management mechanisms for solution for these positive issues should differ from the negative issues.

Performing research into management mechanisms which can better manage this type of unexpected behavior seems obvious. Possibly other control mechanisms than the ones from project management are more suited for dealing with these unexpected issues. For instance, program management owns control mechanisms that might cope with this unexpected behavior. As is discussed in “Gower handbook of programme management” [32] an aspect of programs is: “*Exist in a world that is constantly changing. These changes need to be constantly monitored and their impact on the programme and its projects controlled and managed*”. Ribbers also gave the suggestion that program management might be a suitable control mechanism for ERP implementations [33]. Therefore, it seems relevant to perform more research into the relevance of appropriate control mechanisms based on acceptance of the inherently complex nature of ERP implementation.

To maintain an attitude towards “expecting the unexpected” is no doubt difficult. Research into support for such an attitude is therefore required.

Given this relevance of unexpectedness it seems relevant for top management to be involved more hands-on with this type of projects. Therefore, research into what top management support looks like and how this can be obtained and executed, seems to be worthwhile.

References

- [1] D. Aloini, R. Dulmin, and V. Mininno, "Risk assessment in ERP projects," *Information Systems*, vol. 37, pp. 183-199, 2012, doi: 10.1016/j.is.2011.10.001.
- [2] A. Amid, M. Moalagh, and A. Z. Ravasan, "Identification and classification of ERP critical failure factors in Iranian industries," *Information Systems*, vol. 37, no. 3, pp. 227-237, 2012, doi: 10.1016/j.is.2011.10.010.
- [3] P. C. Solutions, "2019 ERP Report: People | Process | Technology," Panorama Consulting, Greenwood Village, 2019.
- [4] C. Iriarte and S. Bayona, "IT projects success factors: a literature review," *International Journal of Information Systems and Project Management*, vol. 8, no. 2, pp. 49-78, 2020, doi: 10.12821/ijispm080203
- [5] C. Chapman and S. Ward, *Project risk management: processes, techniques and insights*. John Wiley, 1996.
- [6] M. L. Todorović, D. Č. Petrović, M. M. Milić, V. L. Obradović, and S. D. Bushuyev, "Project success analysis framework: A knowledge-based approach in project management," *International Journal of Project Management*, vol. 33, no. 4, pp. 772-783, 2015, doi: 10.1016/j.ijproman.2014.10.009.
- [7] PMI, "Risk," in *PMI Lexicon of Project Management Terms*, ed. Pennsylvania USA: PMI Publications, 2017, p. 18.
- [8] B. Edmonds, "Syntactic Measures of Complexity," PhD, Department of Philosophy, University of Manchester, Manchester, 1999.
- [9] M. Hertogh and E. Westerveld, "Playing with Complexity: management and organisation of large infrastructure projects," ed. Rotterdam: Erasmus Universiteit Rotterdam, 2009.
- [10] B. Edmonds, "Complexity and Context-Dependency," *Foundations of Science*, journal article vol. 18, no. 4, pp. 745-755, 2013, doi: 10.1007/s10699-012-9303-x.
- [11] S. M. Manson, "Simplifying complexity: a review of complexity theory," *Geoforum*, vol. 32, pp. 405-414, 2001.
- [12] A. Fadlalla and F. Amani, "A keyword-based organizing framework for ERP intellectual contributions," *Journal of Enterprise Information Management*, vol. 28, no. 5, pp. 637-657, 2015, doi: doi:10.1108/JEIM-09-2014-0090.
- [13] E. Nazemi, M. Tarokh, and G. Djavanshir, "ERP: a literature survey," *The International Journal of Advanced Manufacturing Technology*, vol. 61, pp. 999-1018, 2012.
- [14] B. R. Schlichter and P. Kraemmergaard, "A comprehensive literature review of the ERP research field over a decade," *Journal of Enterprise Information Management*, vol. 2, pp. 486-520, 2010.
- [15] J. R. Meredith and S. J. Mantel Jr, *Project management: a managerial approach*, Eighth ed. John Wiley & Sons, 2011.
- [16] S. V. Grabski, S. A. Leech, and P. J. Schmidt, "A Review of ERP Research: A Future Agenda for Accounting Information Systems," *Journal of Information Systems*, vol. 25, no. 1, pp. 37-78, 2011, doi: doi:10.2308/jis.2011.25.1.37.

- [17] D. Schniederjans, "Successful ERP implementation: an integrative model," *Business Process Management Journal*, vol. 19, pp. 364-398, 2013, doi: 10.1108/14637151311308358.
- [18] K.-Y. Kwahk, "ERP Acceptance: Organizational Change Perspective," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06) Track 8*, ed, 2006, p. 10.
- [19] H.-L. H. W. Wei, E. T. G. E. W. Wang, and P.-H. P. J. Ju, "Understanding misalignment and cascading change of ERP implementation: a stage view of process analysis," *European Journal of Information Systems*, vol. 14, pp. 324-334, 2005.
- [20] H. Altamony, Z. Al-Salti, A. Gharaibeh, and T. Elyas, "The relationship between change management strategy and successful enterprise resource planning (ERP) implementations: A theoretical perspective," *International Journal of Business Management and Economic Research*, vol. 7, no. 4, pp. 690-703, 2016.
- [21] C. Marnewick and L. Labuschagne, "A conceptual model for enterprise resource planning (ERP)," *Information Management & Computer Security*, vol. 13, pp. 144-155, 2005, doi: 10.1108/09685220510589325.
- [22] E. W. T. Ngai, C. C. H. Law, and F. K. T. Wat, "Examining the critical success factors in the adoption of enterprise resource planning," *Computers in Industry*, vol. 59, pp. 548-564, 2008.
- [23] L. Shaul and D. Tauber, "Critical success factors in enterprise resource planning systems: Review of the last decade," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 55, 2013.
- [24] F. Bollou, E. Balogun, and I. Usang, "Eradicating complexity in software interface for increased productivity: Increasing Effectiveness of Enterprise Systems," presented at the Proceedings of the International Conference on Software Engineering Research and Practice (SERP), 2012.
- [25] M. Bradford and J. Florin, "Examining the role of innovation diffusion factors on the implementation success of enterprise resource planning systems," *International Journal of Accounting Information Systems*, vol. 4, pp. 205-225, 2003.
- [26] K. J. Spiteri, C. L. Luca, T. Reynolds, and G. Wilson, "Defining a baseline complexity model for ERP Systems over SaaS," *Journal of Internet Technology and Secured Transactions*, vol. 1, no. 3/4, 2012.
- [27] R. M. Fontana and A. I. Neto, "ERP systems implementation in complex organizations," *JISTEM - Journal of Information Systems and Technology Management (Online)*, vol. 6, no. 1, pp. 61-92, 2009, doi: 10.4301/10.4301%2FS1807-17752009000100004.
- [28] S. Ghosh and M. J. Skibniewski, "Enterprise resource planning systems implementation as a complex project: a conceptual framework," *Journal of Business Economics and Management*, vol. 11, no. 4, pp. 533-549, 2010.
- [29] I. Coyne, "Sampling in Qualitative Research. Purposeful and Theoretical Sampling; Merging or Clear Boundaries?," *Journal of Advanced Nursing*, vol. 26, no. 3, pp. 623-630 10/1997 1997, doi: 10.1046/j.1365-2648.1997.t01-25-00999.x.
- [30] Oracle, "Oracle PeopleSoft Applications," <http://www.oracle.com/us/products/applications/peoplesoft-enterprise/overview/index.html> (accessed August, 2019).
- [31] Axelos, "PRINCE2 (PProjects IN Controlled Environments)," <https://en.wikipedia.org/wiki/PRINCE2> (accessed August, 2019).
- [32] G. Reiss, M. Anthony, J. Chapman, G. Leigh, P. Rayner, and A. Pyne, *Gower Handbook of Programme Management*. Gower Publishing, Ltd., 2006, p. 738.
- [33] P. M. A. Ribbers and K.-C. Schoo, "Program management and complexity of ERP implementations," *Engineering Management Journal*, vol. 14, pp. 45-52, 2002.

Biographical notes



Guy Janssens

Guy L.S.G. Janssens is an Assistant Professor at Open Universiteit in the Netherlands. He received his Ph.D. from Open Universiteit. His research focuses on project management, and the complexity and control of implementation of ERP systems.



Rob Kusters

Rob J. Kusters obtained his master degree in econometrics at Tilburg University in 1982 and his Phd. in operations management at Eindhoven University of Technology in 1988. He is professor of 'ICT and Business Processes' at the Dutch Open University in Heerlen where he is responsible for the master of science program 'Business Process Management and IT' and chairs the information systems department. Research focuses on project and process performance, IT-Governance, software quality and software management.



Harry Martin

Harry H. Martin is an Associate Professor at the Open University in the Netherlands. He is professionally interested in various Information Science issues, but his current research and teaching focusses on strategic IT-sourcing and qualitative performance measurement.