Systemic Differences between SaaS- and On-Premise-ERP: An Overview of a Qualitative Option Calculation Scheme

The Mainframe Is Dead.
Long Live the Mainframe!
The Mainframe Is Dead. Long Live the Mainframe!

Glen Sagers, Kathleen Ball, Bryan Hosack, Doug Twitchell, David Wallace

Page 4

Systemic Differences between SaaS- and On-Premise-ERP: An Overview of a Qualitative Option Calculation Scheme

Björn Link, Andrea Back

Page 11
Many have predicted the death of the mainframe over the last 10 to 20 years, yet it still is a large part of enterprise computing today. Companies not only use the mainframe for “legacy” applications, but are also developing new applications for mainframes, resulting in a rise in mainframe sales. These increases, along with an aging mainframe workforce and academia’s move away from mainframe-related curricula and courses, has resulted in a shortage of workers trained in mainframe applications. In this paper we report on a U.S.-based survey of industry and academia that confirms the existence of this problem. We conclude with some possible future research directions to explore as possible solutions to this potential dilemma.

1. Introduction

Over the last two decades, articles have declared the death of the mainframe computer [3], [22], [27], its extinction in large enterprise computing, and academia’s purge of mainframe related courses. But, is mainframe computing really dead? Should companies do away with these “Legacy” systems that can act as a repository of business processes? [9] Furthermore, should educational institutions really be ridding themselves of mainframe-related curricula in favor of other enterprise technologies? In this paper we report the results of a U.S.-based study designed to ascertain the state of the mainframe in enterprise computing in industry and academia’s ability to supply graduates with the knowledge and skill to run these systems. Based on the results of this study, several proposals for action are given for consideration.

Large enterprise computing includes two primary computing architectures. The first uses one or a few very large servers that provide batch and online transaction processing to hundreds or thousands of clients. These servers have historically been called mainframes and were mainly built by IBM, though other companies such as Hitachi and Tandem built similar machines. Companies such as Sun Microsystems and Hewlett Packard built smaller servers generally not classed as mainframes, but which provide analogous services for smaller enterprise subsystems. The second and more recent architecture is the clustering of many commodity servers in parallel and tasking them to process the transactions of these hundreds or thousands of clients in a distributed fashion. Although the debate about which architecture is better rages on [11], [8], both architectures typically provide the high-performance, high-availability computing needed for large enterprises, and both can be considered large enterprise computing. The rise of Cloud computing and virtualization has further muddled the debate of whether hardware needs to be housed within an organization [6].

While clusters have been touted as the logical successor to mainframes for the past several years, for reasons of economy, scalability, and fail-over capability, IBM has seen increasing sales of their mainframe hardware and software [24], [25]. The often-predicted end of mainframes has still not materialized, and indeed, the trend in some industries is to continue or even increase investment into mainframe architectures. The scope of the current research is not to analyze whether clusters or mainframes are better for a given application or industry, rather this paper focuses solely on the role of mainframes in industry enterprise systems.

To utilize mainframes as a viable part of an organizations enterprise platform, and integrate it with other systems re-
quires specialized training. The knowledge and skills required for this work includes skills in setting up, initializing, managing, and running a mainframe; as well as a knowledge of control or programming languages used in mainframes. Also included in this repository of knowledge are skills in integrating the mainframe with the rest of an organization’s computing equipment, such as initializing data communications and integrating the middleware necessary to operate in a multi-platform environment.

Surveys [23], [7] suggest that the vast majority of mainframe knowledge in industry will disappear as the older information systems (IS) managers retire over the next three to ten years. The graduates to replace these people are being trained with little or no knowledge of mainframes. The resulting problem will be a lack of mainframe-educated workers to replace the retirees. According to Light [19], a risk of losing vital business information that has been stored within these systems as they have evolved is a real consideration for organizations to consider. Other factors, such as the speed or embeddedness of the system, should be considered when making discontinuance decisions [13]

This problem needs to be addressed because it seems that although those directly involved with large enterprise computing believe there is an imminent crisis in available skills, the educational system apparently either does not believe a problem exists or believes everything mainframe is dying and that the legacy should be buried. Legacy programs and mainframe systems are still widely used and even dominate certain industries (e.g. financial), and an even greater need is emerging in integrating mainframe systems with newer information technologies like web and client/server applications in mixed-platform environments. Indications from industry suggest that there is a need for academia to continue or reinstate education in large enterprise systems.

Previous research on the state of mainframes in industry and academia is largely limited to trade journals, as academics apparently have abandoned it as an unfruitful area of study. Most studies indicate that the number of students being prepared for large enterprise computing is declining [23] and that IS managers still see a need for this knowledge base [20], [7]. Financial institutions depend on mainframe processing to the point that COBOL handles nine out of ten ATM transactions within a mainframe environment [10]. Popular press reports the knowledge base for large enterprise computing is leaving the workforce [21], and indicates an exacerbation of the problem, since “few colleges [are] offering mainframe courses, most young people aren’t prepared for the complexity of mainframes” [5], thus new graduates are unable to just walk in and function in their job roles without further training.

If academia does not successfully educate enough people to replace retirees, industry could be placed in a similar position as they were with Y2K. In the Y2K situation, expertise to work with mainframes and COBOL code was outsourced at exorbitant costs. If the current mainframe systems are not adequately maintained to meet the challenges of changing marketing conditions, companies may be forced to patch their operations together resulting in lost time and effort. Also, their systems might not be integrated properly so that redundant systems may exist across platforms, causing reliability, data interchange and maintenance problems.

Applications on the mainframe are not disappearing, so they need to be maintained and integrated into the organization’s information system structure. The potential impact of upcoming retirements is indicated in an IBM study, which shows that a majority of the workforce with experience in IBM mainframe systems is nearing retirement age as illustrated in Figure 1 [15], [16]. This leaves companies faced with hiring under-qualified replacements and training them on the job, or removing mainframes from their enterprise computing capacity.

1.1. Rewrite or Integrate?

General Mills, a company with reported revenues of $11 billion in 2006, and ranked 206th on Fortune Magazine’s list, made the decision in the early part of this century to do away with mainframes [14]. This decision was prompted in large part by the inability to find replacements for their retiring mainframe experts. Mission Linen Supply, Inc. is another company that converted away from mainframes as a result of losing 50 percent of their mainframe experts and being unable to find replacements [14]. Garvey [14] also cites a Meta Group study indicating that 60 percent of workers in data centers housing mainframes are 50 years old or older. This aging workforce is taking their knowledge with them upon retirement.

Carr and Kizior [7] showed some interesting trends relating to COBOL, the historically dominant language used on mainframes. In a study of 208 information systems (IS) managers, 56 percent reported current COBOL code development and

![Figure 1 Years of z/OS or OS/390 Experience](image)
maintenance. Of the organizations using COBOL, nearly 60 percent reported in excess of five million lines of code in use while thirty percent reported 20+ million lines of code in use. Thirty percent of all the respondents’ programming was an effort to maintain COBOL code and 10 percent of the programming effort was geared toward new applications using COBOL. These numbers are down compared to their 1999 study; nevertheless, the study shows that a large amount of code remains.

If organizations are moving away mainframes, then they are also very likely replacing legacy languages such as COBOL. Thus, a logical question is whether it is more efficient to rewrite the legacy code in another language on another platform, or maintain and extend the current code base. In order to rewrite the legacy code, organizations must not only convert these legacy programs to another language, but must find other utilities that provide support to those languages such as sort utilities and common file management operations. This requires a major restructuring of their information systems infrastructures without a disruption in their primary business functions. Legacy code cannot simply be rewritten line by line, converting from one language to another. Applications must go through all the analysis and design phases of the software development life cycle (SDLC) requiring large investments in both time and money. The cost/benefit analysis must demonstrate a significant gain from the replacement in order to justify the expense of rewriting. It is not sufficient to say that the newer technologies are better for the organization simply because they are new. Kelley et al. [18] outline many factors that impede the conversion of legacy systems that continue in use, such as stability of some systems having been built and maintained over long periods of time as well as the interwoven connections between the technology and business processes. They also highlight factors that are contributing to the need to revise such systems due to increasing inefficiencies and deviation of the core business processes from those embedded in a system.

A more prudent course for some organizations might be integration. Kanter and Muscarello [17] investigated the time it took to web-enable legacy systems. They compared the time required to adapt a COBOL/CICS system for web access with the time it took to rewrite the entire application in Java. The SDLC steps they used for this conversion were requirements analysis, specification, design, development, testing, and performance/value issues. The specification and performance/value issue phases were not needed for the COBOL revision. The total time to adapt the COBOL/CICS application was 29.5 minutes whereas the total time to rewrite in Java was 1275 minutes (21.25 hours) [17]. Kanter and Muscarello [17] put this in terms of money based on an average salary of $54,000. The cost of the revision was $26,30 and the cost of the rewrite was $1,134.80. In percentage terms, the revision took only 2.3% of the time and cost of the rewrite. This was obviously a very small project but if these costs extend linearly to a large enterprise systems project requiring six months and 200 workers, the cost of a rewrite would be $11,232,000 compared to the cost of integration of $258,336. The cost and time benefits are obvious but integration is only a viable solution if there are curricula that teach large enterprise systems, including the components with which they interact and integrate.

1.2. Curriculum

The information systems (IS) curriculum as set forth by the Computing Accreditation Commission (CAC) of the Accreditation Board for Engineering and Technology (ABET) emphasizes the need students to be prepared to effectively function in an IS environment as an IS professional [1]. The criteria indicates that an IS curriculum must also include principal coverage in many areas including hardware and software, a modern programming language, networking and telecommunications, analysis and design, data management, and the role of IS in an organization. One interpretation of modern language could mean that it is widely used in industry. COBOL, within a mainframe environment, is widely used in industry and would fall nicely into that category. COBOL is used in 75 percent of all production transactions on mainframes and 95 percent of finance/insurance data processing, and 60 percent of all web-access data lives on a mainframe [4].

IS managers have indicated the desire to incorporate the integration of large enterprise technologies with object-oriented and web-based technologies into the academic curriculum [23], [7]. Corporations depend on the speed, accuracy, and stability of their large enterprise legacy systems. It is an asset they have invested in heavily. In addition to the mainframe IS staff decline in industry, faculty interest in teaching large enterprise languages such as COBOL has decreased [23], likely due to the fact that many of the educators capable of teaching large enterprise systems and associated programming languages are nearing retirement age.

The current large enterprise systems workforce will be of retirement age very soon. Companies such as General Mills and Mission Linen Supply [14] have already felt this human resource shortage. As a result of the lack of qualified talent, some companies have chosen to replace their mainframes with newer technologies, leading to a large investment in terms of time and money in rewriting legacy programs based on a mainframe into a language that will interact with the chosen technology. If large companies as large as General Mills cannot find adequate staff to replace their mainframe experts, but IBM’s mainframe sales keep increasing, this may indicate a continuing market for trained individuals. If there is a shortage of talent, perhaps there is a need for academic curricula in large enterprise systems.
2. Methodology

Two surveys were conducted, one in industry and one in academia. Both were sent out as interactive PDF documents that users returned via email. The industry survey contained questions pertaining to the number of new hires for large enterprise systems positions they expect over the next five years, the number of large enterprise employees retiring during that same time frame, and the number of positions in large enterprise systems currently open, as well as basic demographic questions. Questions were also asked about the languages, environments, and databases used in the large enterprise systems. The survey to academia had two parts. The first part consisted of questions relating to the number of large enterprise courses offered, the enrollment in such courses in the Spring and Fall semesters of 2006, the large enterprise languages offered and required, and an enumeration of courses constituting their large enterprise curriculum, if they had one. The second part asked questions about the non-large enterprise languages offered, which three languages had the highest enrollment in the Spring and Fall of 2006, and asked about the likelihood of large enterprise related courses being required or offered as electives.

The population for the industry survey consisted of Fortune 1000 companies for 2006 [12]. Sampling this population was difficult. The researchers had a list of contacts at various companies, furnishing a starting point, and others were distributed at various industry meetings the researchers attended. The sample of the population ended up being a networked list, making the sample less than the random ideal. A total of 222 surveys were distributed to these firms.

The population for the academic survey was all universities and colleges that have computer science or information systems programs that were accredited by the Computing Accreditation Commission (CAC) of the Accreditation Board for Engineering and Technology (ABET) as listed on their website as of January 18, 2007. The survey and cover letter were sent to all 259 institutions accredited by CAC/ABET [2].

3. Results

The results of this preliminary study indicate that there is an industry need for new hires in large enterprise computing but there is a shortage of academic institutions offering a skill set to fill this void. The discussion of the results is split into two parts: industry and academia, after which a synthesis of these findings is presented.

3.1. Industry

The level of analysis is at the organizational level with fourteen companies responding to the survey. The industries represented in this preliminary study include financial, computing, insurance, utilities, communications, construction and manufacturing, among others, with a mean number of employees per organization of about 88,000. Almost all of the organizations had IT departments greater than 200 employees in size (13 out of 14) with 100 or more employees working in the area of large enterprise computing (13 out of 14). Most companies (12 out of 14) indicated that languages such as COBOL, JCL (Job Control Language on the mainframe) and REXX (another mainframe language) were a large part of IT employees’ jobs, as well as large enterprise databases on the mainframe like DB2 and IMS.

The industry survey indicates a growing need for individuals with mainframe expertise over the next five years as large enterprise computing employees begin to retire. Eleven out of fourteen organizations indicate a need to replace anywhere from six to 2000 large enterprise computing employees with a mean of 231 across organizations. Factors such as the cost of training, which averaged almost $32,000 (based on eight responses) and difficulty finding employees to hire (see Figure 2) were cited as impediments to finding qualified personnel, despite competitive starting salaries (an average of almost $43,000 for those organizations reporting salary figures) being offered. The issue in the next decade as large enterprise computing employees begin to retire will be finding individuals to fill their shoes. Will academic institutions provide sufficient well-trained students who are able to step into these positions?

3.2. Academic

Among the 40 academic institutions surveyed, only 6 offer large enterprise computing courses and only 1 of these 6 has a large enterprise computing curriculum. Of the remaining 34 institutions, only 3 expect to require such courses in the future, and only 12 foresee any likelihood of making such courses available as electives (see Table 1).
Enterprise systems are an area where outsourcing may not be an option for industry because of the critical nature of the systems involved. Continued partnerships allowing communication between industry and academia are necessary to ensure that universities are producing students that are able to fill available positions and universities are receiving the resources necessary to continue to attract students and develop large enterprise computing curricula.

3.4. Limitations

While the data in the study was collected in 2006 and analyzed in 2007, this issue is still very relevant in industry according to a recent study conducted by TheInfoPro, Inc., in September and October, 2008 [26]. According to this survey, 72% of responding organizations had mainframe staff already eligible for retirement. Further, those organizations using mainframes are expected to increase mainframe-related spending over the next two years, and ranked hiring and training new staff as their highest priority action to offset projected staff shortages.

This survey experienced difficulties in that the PDF survey, designed to be universal, proved to be unusable for some respondents due to enterprise security settings disabling the functionality of the emailed surveys. Therefore, the survey had a small number of respondents, curried from a convenience sample of industry. However, the firms that responded represented some of the giants of their respective industries, and their needs likely reflect the needs of many other large companies in this regard. An additional problem was that the survey to industry asked certain questions that were determined by the various legal departments to be unanswerable. The survey did not give an option for choosing not to answer a particular question. It was indicated in a few emails that when the particular firm was unable to answer, they would respond with either an “unknown” or a “0” (zero). It is beyond the knowledge of the authors which responses of zero represented an actual zero and which represent an unknown. It was assumed in the data analysis that all zeros were indeed zeros.

There were several organizations that chose to complete the survey as a committee effort. This gave a corporate-wide view, which is likely to be more accurate than one person’s view. In the event that duplicate surveys were received from a single firm or a single academic department, the responses were averaged into one response for purposes to achieve the specified organizational level of analysis.

4. Future Research

This study indicates a need for a longitudinal study should be conducted to monitor trends in enterprise computing and mainframe technology, and the integration of mainframes with more traditional technologies. It could be argued that

<table>
<thead>
<tr>
<th></th>
<th>Required</th>
<th>Elective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definitely</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Very Likely</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Likely</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Not Very Likely</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>Absolutely Not</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>34</td>
<td>34</td>
</tr>
</tbody>
</table>

**Table 2: Large Enterprise Computing Courses by Section (size) and Semester**

<table>
<thead>
<tr>
<th>COBOL</th>
<th>JCL</th>
<th>z/OS</th>
<th>COBOL</th>
<th>JCL</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1-25</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>26-50</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&gt;50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

**Table 1: Large Enterprise Computing Expected Future Course Offerings**
mainframe technologies should be allowed to slowly die off
to be replaced with “newer” technology, but questions remain
as to which other technologies can scale to levels required for
processing thousands of transactions. Another possibility is
that the mainframe will be able to reinvent itself to support
modern application delivery requirements.

As part of a longitudinal study the usage trends of the main-
fra mes could be tracked to determine what trajectory it is fol-
lowing. Another possibility may exist in the need for legacy
skills continuing to be taught well into the future, especially
for those students who gain experience in both the “old” and
the “new”, even if only during a multi-year period of transition.
Further study is needed to gather information with regard to
industry solutions for finding qualified candidates. Will compa-
nies simply attempt to attract new hires and train them if tradi-
tional educational systems are not going to provide graduates
with the necessary skills and knowledge for the large enterprise
environment? Will these strategies include individual mentor-
ing programs, internship programs, outsourcing programming
skills, or even a “boot camp” training program similar to what
was used during Y2K? In other words, is there an ongoing need
for academic training in mainframes, or should private indus-
try solutions for finding qualified candidates. Will compa-
nies simply attempt to attract new hires and train them if tradi-
tional educational systems are not going to provide graduates
with the necessary skills and knowledge for the large enterprise
environment? Will these strategies include individual mentor-
ing programs, internship programs, outsourcing programming
skills, or even a “boot camp” training program similar to what
was used during Y2K? In other words, is there an ongoing need
for academic training in mainframes, or should private indus-
try training such as certification programs take this role.

Finally, it would also be beneficial to track organizations that
have moved away from mainframe environments over the
next five years in order to examine their successes or failures.
This could provide some insight into lessons learned or strate-
gies that were successful?

5. Conclusion

While this study is preliminary, it illustrates a continued need
for mainframe-educated workers in industry. While this does
not mean that every institution needs mainframe courses as
part of their enterprise systems curriculum, neither should
mainframes be abandoned. Enterprise systems consist of
much more than mainframes, including cluster computing,
ERP systems, data warehouses, and other enterprise systems,
but mainframes are still a part of enterprise computing, and
likely will be for decades to come. This study showed that
academia is not currently capable of providing enough main-
frame-educated individuals due to the discontinuation of
large enterprise systems education over the last 10-20 years.
Industry has an immediate and a foreseen need five years for-
ward. A few industry respondents even indicated that looking
beyond the next five years the need would be even greater.
These findings, paired with academia responses indicating the
unlikelihood of large enterprise courses, much less an entire
curriculum, does not give an encouraging personnel recruiting
outlook for industry.

Industry respondents specified a need for people with main-
frame knowledge, large enterprise languages, databases, and
systems knowledge, and integrated knowledge. The number of
new hires needed by industry through retirement and attrition
is greater than the number of people being prepared through
academia. Furthermore, the retirement of older, skilled main-
frame personnel over the next few years will exacerbate this
situation as the shortage of graduates to fill mainframe entry
level positions and the managers to supervise mainframe op-
erations becomes critical to the survival of their information
systems.

We propose two possible approaches to this problem. First,
related research supports integration of mainframe and client/
server technology over redesigning and rewriting legacy code
in a new language on a completely different computer plat-
form other than mainframe. Newer mainframe technologies
(such as IBM's WebSphere) adapt mainframes to client/server
and web environments so there is not a need to abandon the
mainframe environment which has offered a stable and secure
platform to allow organizations to meet their goals and objec-
tives. Organizations can achieve the benefits of both environ-
ments by blending mainframe and client/server technologies
into a multi-platform information system that will allow them
flexibility to adapt to new needs, requirements, and trends. This
approach will still require some mainframe educated workers,
but fewer than to run the enterprise entirely on the mainframe.

The second approach is for industry to encourage academia
to develop curriculum that focuses on large enterprise com-
puting. Incentives for development could include, but need
not be limited to, scholarships, internships for students, grant
funding for curriculum development, faculty training pro-
grams, and guest lecture programs for prospective students.
The fact that industry is integrating this technology and will
seek graduates with these skills for an extended period of time
should be an incentive for academia to provide graduates with
these skills. A high placement rate for graduates can improve
the declining enrollment often seen in information technol-
ogy departments throughout the United States. While further
study is needed to confirm these trends and the ongoing need
for graduates, this pilot study should serve as a reminder for
academia to consider the needs of industry when designing
curricula.

References
www.abet.org/.
April, pp.70-72,75.
to Run Decidedly Unsexy, but Vital, Systems. The Boston Globe, August


Keywords
Mainframe, enterprise computing, enterprise systems, education, COBOL

Contact:

Glen Sagers
School of Information Technology
Illinois State University
Normal, IL
Email: gsagers@ilstu.edu
Phone: (309) 438-3741

Kathleen Ball
School of Information Technology
Illinois State University
Normal, IL

Bryan Hosack
School of Information Technology
Illinois State University
Normal, IL
Email: bhosack@ilstu.edu
Phone: (309) 438-8133

Doug Twitchell
School of Information Technology
Illinois State University
Normal, IL
Email: dtwitch@ilstu.edu
Phone: (309) 438-7756

David Wallace
School of Information Technology
Illinois State University
Normal, IL
Email: dcwalla@ilstu.edu
Phone: (309) 438-8049
Systemic Differences between SaaS- and On-Premise-ERP: An Overview of a Qualitative Option Calculation Scheme

Björn Link, Andrea Back

Andrea Back works as a leading Professor at the department of "SoM" School of Management at the institute of business computing at the university of St. Gallen. Her primary focus in research has been Business 2.0, Mobile Uni-Apps and Learning Center. She is the co-editor of newspapers such as "Zeitschrift für e-learning-lernkultur und bildungstechnologie", "Learning Waves-Blog zur Zukunft von Lernen und Wissen", "Blog Business 2.0", "WissensWert Mitmachzeitschrift über Enterprise 2.0", Knowledge Management and E-Learning" and "Fallstudien-netzwerk Enterprise 2.0".

Björn Link is a PhD-candidate at the department of "SoM" School of Management at the institute of business computing at the university of St. Gallen.

A research-based calculation scheme of IT options will be developed based on systemic differences between SaaS- and On-Premise-ERP. The calculation scheme can compute relative cost differences between SaaS- and On-Premise, and thus better determine which ERP operating mode is more financially affordable in the particular case. The systemic differences have been researched through a multiple case study with four ERP producers. The data obtained has been substantiated by ERP literature in general and SaaS literature in particular. The comparative total ERP cost was calculated by applying a total cost of ownership approach, which was modified to sum up only all relative cost differences rather than all absolute costs. This relative total cost approach enables the reader to compare the relative cost differences for each operation mode and includes financing aspects; the fixed costs were discounted and interest rates (by debt and equity) were included to compensate for the investment differences between On-Premise- and SaaS-ERP. The classification and calculation scheme is limited in that the research method is qualitative. A quantification of the adoption factors determined by building strict relations between the systemic differences and the customer's characteristics, as well as standardization and weighting of the importance of the adoption factors is not the focus of this contribution and is left to other publications.

1. Introduction

Relatively complex IT systems such as ERPs could until recently only be operated as licensed products on local servers. The Software as a Service (SaaS) innovation, drawing on existing technology, made it possible for the first time for providers not only to offer a more complex system but also to deliver it over the Internet. Each new operating model allows additional application options; the question for research is, then, which of the operating modes, SaaS, On-Premise, or perhaps ASP or some hybrid form, offers the best long-term value in a particular ERP case. It is then left to each company to select those solutions that offer the lowest cost with the best possible support for their operational processes. An ERP cost calculation method would best support the ERP operation mode-selecting end user, but a search of the extant literature yielded no results pertaining to ERP cost calculation methods or ERP delivery strategy selection.

To develop the ERP cost calculation method it is necessary to identify the systemic differences between SaaS- and On-Premise-driven ERP systems, so that a calculation of the two options can provide decision support for or against one of the two operation modes. The classification of the systemic differences, which has been researched and published in a previous article, will hence be the point of departure of the option scheme to be constructed. This paper, therefore, lays as a qualitative foundation in section 3 a comparison of the main differences between SaaS- and On-Premise-ERP and an option scheme, which will be used for calculation in section 4. The 5th section states some management strategies based on the previous calculation scheme, and the last section concerns the outlook and limitations.

To step into the field, the related work and the research method are briefly discussed in the next section.

2. Related Work and Research Method

2.1 Related Work

The extant literature on ERP systems and the SaaS operation mode in general, as well as on SaaS-ERP in particular, was the starting point of the research. Little scientific literature about calculation-based models of IT options with respect to operation modes was found, because most of the papers on ERP...
systems are concerned with the strategic and functional selection processes, e.g. 1.8, 1.8. With regards to the economics of SaaS-ERP systems, there is no indicator at all in the extant literature for such a comparative calculation between SaaS- and On-Premise-ERP (c.f. 1.8 with further references). There is an approach for calculating the value added by the IT in the case of service-oriented architecture (SOA, 1.8, 1.8, 1.8) or in the case of value-oriented process modeling 1.8. This approach enables the user to calculate the in- and out-payments of a specific process and to compare the different IT options under consideration.

Another approach to investigating SaaS is the cost and the pricing of the service. In Katzan Jr. and Dowling the operation mode and characteristics of SaaS, as well as, inter alia, the costs of a SaaS-ERP system are presented 1.8. Studies that look intensively at pricing models can also be found: 1.8, 1.8, 1.8. Choudhary takes a comparatively detailed look at SaaS and On-Premise in order to use pricing models to estimate the differences in quality 1.8. While common cost calculation criteria for a SaaS-software can be transferred to a cost effectiveness calculation between SaaS- and On-Premise-ERP, price modeling has no place in a cost effectiveness calculation, since the conditions and bases of calculation are specified by the ERP producer.

2.2 Method

A two-step approach was used to construct the option calculation scheme. First, a classification of the systemic differences enabled the identification of all the general differences between the two extremes of ERP operation modes to be found. Second, the classification was used to build the option calculation scheme by relative comparison. In detail:

2.2.1 Data acquisition, data sampling and classification of systemic differences.

A “rigorous literature review” as described by vom Brocke et al. had been conducted to gain insight into what already exists 1.8. This literature review enabled a detailed exploration of all current general operation mode differences between cloud computing and On-Premise, which may be applied to the more specific domain of ERP operation modes and further identifying the remaining research gap with respect to the more specific ERP operation mode differences 1.8. The data from the literature were analyzed using both open coding and operation mode-contrasting meta-matrices 1.8, 1.8. The matrix thus obtained, when analyzed using pattern coding, revealed 6 main pattern clusters and many systemic differences.

The applicability of the general differences found in the literature was investigated and verified through case study research at ERP producers’ premises. The inappropriate differences were discarded and the general operation mode differences were supplemented with further ERP-specific systemic differences. These case studies, moreover, provided more background information, allowing explanations or a better understanding of the contexts of the systemic differences to be found. In total, 15 interviews with 4 different ERP producers were conducted and transcribed.

The data collection was supplemented by document analysis (websites, informational material, pricing lists, internal documents, etc.), researcher’s notes and real artifacts (ERP systems, test accounts, instructional videos; c.f. 1.8, 1.8, 1.8). These case data were analyzed using selective and open coding and were structured into a contrasting meta-matrix where operation-mode specific explanations and contextual information had been assigned to the respective systemic difference criteria 1.8, 1.8. The meta-matrices for each case had been condensed to an aggregate contrasting meta-matrix; the most important systemic differences are available in 1.8.

The main meta-matrix, which contains all identified ERP operation mode differences, was used as the starting point for constructing the option calculation scheme. All classified systemic differences can be assigned as an advantage of either SaaS- or On-Premise-ERP, which may reduce the overall cost of the ERP system in a particular case 1.8. This previous research enables a calculation scheme to be built using differential calculation methods, thereby eliminating absolute cost raising, which is usually perceived as too time-consuming.

2.2.2 Option calculation scheme construction method.

According to vom Brocke et al. alternative process designs can be compared in two different ways: using a total or a differential calculation 1.8. In the total calculation, each payment amount is assessed and calculated independently, whereas in the differential calculation only the additional payments that are relevant to the comparison of two alternatives are considered. In principle the latter approach could be transferred to the comparison of ERP operation modes, even though the ERP operation modes are not themselves process designs. The operation modes have an impact on the business processes and their valuation, e. g., maintenance cost differences or assessment of the flexibility and adaptability to the lived processes. So there is some evidence supporting the use of this approach for comparing ERP operation modes. The calculation method of vom Brocke et al. is based on the assessment of Event-driven Process Chains at the operational level, Visualization of Financial Implications (VOFI) at the budgeting level and the Total Cost of Ownership (TCO) and Return on Investment (ROI) at the corporate level 1.8. This 3-stage model, which is strictly aligned to evaluating process designs, can be simplified when used to evaluate operation modes. On the one hand, several cost factors are exogenous
and therefore already known on the corporate level, e. g. the license or the subscription costs. So no further investigation of these cost factors will be necessary on the operational and budgeting level. On the other hand, ERP systems that are the same from a functional perspective have the same supporting functions; therefore when comparing such systems, one can estimate that the time and money savings will be the same. The in-payments from applying the ERP system, defined as savings of money and time, can therefore be excluded from the TCO balance, when comparing using the differential calculation. In the exceptional case of the SaaS- and On-Premise-ERP systems not being identical, e. g., with SAP Business by Design vs. SAP Business One, then the user must investigate all the differential functional in-payment factors as well and should include them as in-payment amounts in the cost calculation scheme.

In contrast, internal costs to keep the ERP system operational, e. g. maintenance or updating, are typically not directly known at the corporate level. These internal costs need to be registered at the operational level and have to be budgeted for each year to calculate the total costs over the whole expected lifetime of the ERP system.

Measuring the gains in flexibility typically associated with SaaS in the event that the ERP system must be adapted to a changed situation is more difficult. These flexibility gains will indeed arise insofar as optimal resource management is possible, with costs minimized by renting neither too many nor too few modules, user accounts or infrastructure capacity. Further, the effect size with flexibility gains is directly dependent on the probability or frequency of the expected business change 1.8. The cost amount in the calculation scheme therefore has to change depending on how often the business will change and how much flexibility is required. The respective probabilities therefore must be estimated in advance for each year and budgeted over the whole expected ERP lifetime.

The method applied in constructing the operation mode option calculation scheme will be a differential total cost of ownership approach at the corporate level, which refers to data similar to that investigated in the first part with regards to the systemic differences of ERP operation modes. The operation mode option calculation scheme must include all the differential costs of acquisition, operation, and use. Ellram 1.8 and Ellram and Siferd 1.8 declare the TCO method to be best suited, inter alia, for outsourcing decisions and supplier selections, especially when a high monetary value is at stake, although Ellram 1.8 declared that the main difficulty of the applied method is the complexity of taking all costs into account for the respective object. The applied differential approach eliminates exactly this difficulty by looking only at all cost factors which differ between the two operation mode options. This reduces the complexity considerably and matches best with the present study’s aim of determining the operation mode cost differences in a particular ERP case. Furthermore, this method enables ERP-selecting end users to compute operation mode related cost differences with little effort and without much prior know-how.

3. Option Calculation Scheme

The option calculation scheme is limited to general differences identified by the literal replication logic 1.8. So neither producer-specific differences, nor customer-specific characteristics that induce specific needs are considered here. When applying this scheme, the user should add in any additional case-specific differentiating out-payments, especially with respect to functional differences, before calculating the results as described in the next section. As already mentioned, the option calculation scheme is intended to compare all out-payments exceeding those of the respective other operation mode, so no absolute option scheme cost calculation is provided here.

The option calculation scheme is subdivided into the three following subsections: First, the general nonrecurring differentiating costs are presented; then, the recurring differentiating cost factors are described. The last subsection looks at event-driven differentiating cost factors.

3.1 Initial and Nonrecurring Costs of ERP Operation Mode Options

License and hardware costs: The most obvious and important difference between SaaS- and On-Premise-ERP is that with SaaS, no instance is installed and maintained by the customer. This has the fundamental consequence that with SaaS, no licenses have to be bought, nor does hardware have to be provided (cf. 1.8, 1.8, 1.8, 1.8, 1.8). That means, no initial ERP and operating system installation costs arise in a SaaS-ERP and no IT professionals have to be hired to install or maintain the systems (cf. 1.8, 1.8, 1.8, 1.8, 1.8). The operation and provisioning of the SaaS-system is settled by a monthly subscription fee, allowing fixed costs to be replaced by variable recurring costs. In contrast, On-Premise offers entail high initial and nonrecurring license and hardware costs.

Preliminary project costs: The selection, initiation and implementation stages seem to be similar for the two operation mode options, with minor differences. The similarities arise from the fact that SaaS solutions are not less complex than On-Premise solutions and both systems must be implemented to reproduce and comply with the company’s business processes 1.8, 1.8, 1.8. A preliminary project therefore has to be carried out for SaaS as well. But the scope of the preliminary project may be smaller in a SaaS-ERP than in an On-Premise. To check the functionalities of the ERP system, SaaS customers needs only to open a test account to try the features of the ERP
system, and have all their questions about the features and functions answered immediately; this facilitates the process of selecting all required modules. However, if the systems are identical, the SaaS test account with its advantages of immediate access and the possibility of testing the ERP functionalities may also be used by those selecting On-Premise.

After the selection of the ERP system, SaaS requires no installation, but only the creation of an account. Furthermore, SaaS is preconfigured and typically has self-configuration mechanisms enabling the user to modify the configurations of the system himself, e.g. with a wizard or with user-friendly tables. While this does require learning how to change the configurations, it increases the flexibility and reduces both the service costs and the dependence on the ERP partner. In an On-Premise system, the partner has to understand the business needs first, to get to know exactly what settings have to be changed. There will be some hermeneutic circling between the On-Premise ERP customer and the ERP partner before the configuration settings match the needs of the customer exactly. Hence preliminary project costs and time expenditures may be higher with On-Premise-ERPs than with SaaS, meaning that the same cost categories arise with an On-Premise-ERP as with a SaaS, but not the same costs. All categories, with their respective costs if known or estimable, should be included in the option calculation with respect to module selection, configuration, data migration and costs of conducting a pilot. Hence, ERP selecting customers are strongly encouraged to ask their ERP partners for the respective preliminary project costs and time expenses for each ERP operation mode. This sensitizes the customer to the respective cost factors and may help to fix the right price for the preliminary project service. If the prices for the preliminary project are fixed at the same amount for both ERP operation modes, then the customer can delete these costs from the option calculation scheme.

**Training:** Training is required for any ERP system, irrespective of the operation modes (cf. 1.8). Any expenditures will be directly dependent on the learning forms provided. The singular instance approach of SaaS facilitates the implementation of self-learning sections such as Web learning directly into the ERP system. These new learning methods and self-explanatory mechanisms in SaaS-ERP systems may reduce the costs for conventional classroom learning lessons or even make them obsolete. Additionally, the new learning methods enable immediate training, while classroom learning sessions must be scheduled in advance. Similar mechanisms might also be available in the future for On-Premise ERPs. The trainee’s effort will often be similar irrespective of the training concept. The ERP implementing customer should ask his partner which different training concepts exist and what costs they entail.

**Customization:** If the ERP system has to be customized, there will be different cost factors to be taken into account in the cost calculation. SaaS has only limited customizability compared to On-Premise, because SaaS-ERPs can only be operated in the standard. Thus, the adaptability of SaaS-ERP is limited to the configuration options and interfaces that have been thought of in advance 1.8, 1.8. Further changes can only be made by the ERP provider when including the changes to the standard at an agreed price or with interconnected third-party systems, which must be integrated to the adaption and translation layers, if any. On the other hand, almost everything can be modified in an On-Premise-ERP by changing the source code, but changing the source code requires much know-how and time, and also incurs high costs. Moreover, the customized part is not always supported, especially when major adaptations are carried out. Customization may result in higher maintenance costs than with the SaaS-ERP working on the standard. So SaaS enjoys the advantage of a more reliable system and a guaranteed maintainability in the future, as well as lower customization cost, but lacks unlimited customization capability. Additionally, extensive configuration options are very complex; the customer will need guidance from the ERP partner. SaaS is therefore best suited for low-level customization, whereas a customer who needs a highly specific ERP program is, according to the transaction cost theory, best served by On-Premise 1.8, 1.8. Taking this into account, customization cost differences are very difficult to estimate, because the real costs are rarely foreseeable. In SaaS all estimated wages and service costs for configuring the system, for the ERP provider to adapt the standard and for programming the interface- or third-party system should be considered in the option calculation. Customizing an On-Premise ERP may incur programming and adaption costs to change the source code, to integrate the new function into the system, to test the section and adapt the interfaces. All wages of the internal personnel and all service costs should be considered. The service costs for all adaptations and programming services should be negotiated with the ERP partner; requesting a quote and fixing the price for the necessary adaptations are strongly recommended. With the quotes in hand, the customer should compare the additional costs of the adaptations with the additional recurring costs due to a higher labor effort in case of non-customization (see recurring cost section). These additional labor costs will add up to a significant amount over the ERP’s lifetime, if the lack of customization means that important internal processes are

---

1 Smith et al. categorized training as operations costs, which certainly seems to be evident in the case of general IT, where new software or updates require training each year 1.8. This does not occur in the same way with respect to ERP systems, where the operating principles will be kept constant over the years in most cases. Some recurring training costs arise when new employees are trained and occasionally a new release makes it necessary to retrain the employees. These costs may be included in the next sections as recurring training costs.
misaligned. The investigation of all these customization options and their concomitant costs is necessary to the development of the right customization strategy.

3.2 Variable ERP Operation Mode Costs

Maintenance fee vs. subscription: In addition to the higher initial and nonrecurring costs of an On-Premise-ERP system, recurring costs, which must not be underestimated, will arise as maintenance and updating fees 1.8, 1.8. Predictable maintenance fees of 12% to 22% in respect to the license costs have to be paid to the ERP producer each year. Furthermore, less predictable maintenance services or wages of internal IT professionals have to be taken into consideration. All costs or expenditures for maintaining, backing up, and updating the ERP system and the infrastructure have to be estimated and should be included in the calculation. Customers who are replacing their existing outdated or obsolescent ERP systems may separate the current ERP costs from the other IT costs, obtaining a clearer and more business-related comparison between the operation modes. Separating the internal ERP costs from the other IT helps to get a clear statement about the current and future costs for the internal operation and maintenance. Unfortunately, most of the companies studied for this paper do not separate the ERP from the other IT costs. One way to overcome this problem would be to estimate the proportion of the costs of the obsolescent ERP out of the total IT costs, without having a strict separation by using different accounts. The proportion may be estimated by recording all the work done for a short period of time and extrapolating for a longer period.

SaaS-ERP customers, in contrast, pay only a clear and definable subscription fee, which includes all maintenance, backup, and infrastructure costs. The subscription fee is often dependent on the number of users, sometimes with a required minimum; this simplifies the estimation of the amount to be paid. But the subscription costs may add up to a not-insignificant amount each year.

Service and support: In On-Premise-ERP systems, service contracts may include all additional services and support by the ERP partner up to an agreed limit (on top of the maintenance and update contract). This may reduce the service and support costs, which otherwise would be charged by effort. In a SaaS-ERP, a service contract is not necessary and the support

---

Table 1: Option calculation scheme; first part: nonrecurring costs

---
is always included, but may be limited as well. So different cost calculation values may be applicable, which should be considered in the calculation scheme.

**Customization:** The variable costs of customization are driven by the additional maintenance expenses that occur when the standard system is adapted. Especially when the system has to be updated, the individual source code may produce incompatibilities. So in On-Premise systems, further programming or problem-fixing costs may come along with the customization of the ERP system. The customer should therefore ask his ERP partner how often such incompatibilities typically arise, because the additional maintenance fee for the customized part will typically not cover major problems. Further, it is necessary to get information about the additional maintenance fee, the additional rates per hour for cases which are not covered by the maintenance contract and how much time is typically estimated to fix the major problems. This information makes calculating the average additional recurring costs of customization for On-Premise-ERP systems possible.

The limited customizability of SaaS-ERP, in contrast, leads to fewer special codes and thus to lower additional maintenance expenses. But sometimes essential customizations are not possible, making necessary a compromise that will lead to higher labor expenses. These expenses may be included in the calculation and have to be compared to the additional On-Premise maintenance costs.

### 3.3 Change of Requirements and Stability of the ERP Systems

Comparing ERP operation modes make it necessary to contrast and evaluate the differences in flexibility. Most customers who face high volatility and who require fast changes, growth, or flexibility, or are project-driven, are best served by the SaaS operation mode. The flexibility may allow cost reductions in the future, so the customer should assess the value of the additional flexibility and should take these reductions into account in the option scheme calculation. But the key consideration with respect to the cost calculation is that the costs of change arise only when the changes are put into effect. So each criterion mentioned in this section will be included in the cost calculation only with its respective probability of occurrence. The ERP customer should therefore estimate how likely each of these change criteria below is and can as a result omit all irrelevant criteria. Whether each cost is added to the recurring or the nonrecurring section can be seen in Table 3.

**Functional and infrastructure change:** On-Premise systems are generally more fixed, especially in the sense of the scalability of infrastructure and modules. Modules can be increased by paying the license and recurring maintenance fees for the additional module, but not decreased. The module increase sometimes comes along with a change of the package, necessitating the reinstallation and remigration of the ERP system, and incurring high costs of change. Further, increasing...
modules triggers implementation costs, but as these are identical between the ERP operation modes, they are not considered here.

In the case of a module reduction only the maintenance contract may be reduced - if at all. But the option of reducing the maintenance contract to save money over a limited period is not made available. Most often, extra reactivation fees will be charged to catch up to the latest version. In a SaaS-ERP, modules can always be decreased or increased at least monthly among all the subscription bundles offered by the provider. Companies with a typically volatile business, e.g. seasonal or project-oriented, can therefore benefit most from SaaS-ERPs 1.8, 1.8. When modules are added, the subscription cost will be increased, but no additional costs are incurred relative to the On-Premise-ERP. But with the elimination of a module, data migration costs will accrue, because the reduction means that the data history of the module can no longer be accessed.

**Peak-loads, capacity change and scalability:** The load of an ERP system is directly dependent on the number of users and on the course of business. So when the business works well, the load of the ERP system will be higher. The On-Premise system always has to be aligned with the highest level of load, even when the loads will last for only a short time, e.g. Christmas season or during stocktaking. Exceeding the infrastructure limits will result in slow response times, which can only be overcome by expanding the infrastructure and migrating the system to the new infrastructure. In contrast to this scenario, SaaS has unlimited scalability, because a professional provider will have enough hardware available to serve their customers. Furthermore, peak-loads can be balanced, because customer peaks often arise at different times 1.8. Indeed, the more user accounts are rented, the more expensive the SaaS system is, but the number of user accounts can be rented on a monthly basis, with the unused user accounts reduced after the season is over. This frees up unused resources and reduces the cost to the customer. Customers who wish to change the number of users frequently could insert the average number of users instead in the subscription cost of the variable cost section.

**Dependence on the provider:** SaaS-ERP systems are generally more dependent on the ERP partner, especially on the provider 1.8, 1.8, 1.8, 1.8. Should the provider discontinue the service, whether the discontinuation is planned or unplanned

### Table 3: Option calculation scheme; third part: change of requirements

<table>
<thead>
<tr>
<th></th>
<th>SaaS-ERP</th>
<th>On-Premise-ERP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flexibility, Changeability</strong></td>
<td>Module Increase with $P_{\text{module inc}}$:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subscription fee increase $P_{\text{MI}} \times \text{number of modules}$</td>
<td>RC</td>
</tr>
<tr>
<td></td>
<td>installation costs $P_{\text{MI}} \times \text{hours}$</td>
<td>NRC</td>
</tr>
<tr>
<td></td>
<td>system migration costs $P_{\text{MI}} \times \text{hours}$</td>
<td>NRC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Module Decrease with $P_{\text{module dec}}$:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>subscription fee decrease $P_{\text{MD}} \times \text{number of modules}$</td>
<td>RC</td>
</tr>
</tbody>
</table>
|                       | data migration costs $P_{\text{MD}} \times 
|                       | \text{hours, wages, service}$ | NRC                                          |
|                       |               |                                          |
| **Scalability with $P_{\text{scal}}$:** | change of subscription fee $P_{\text{Sc}} \times 
|                       | \text{user, space, increase, decrease}$ | etc. RC                                        |
|                       | \text{etc.}$ |                                          |

<table>
<thead>
<tr>
<th><strong>Dependence and Stability</strong></th>
<th>SaaS-ERP</th>
<th>On-Premise-ERP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependence on Provider with $P_{\text{provider Charge}}$:</strong></td>
<td>residual value $\Sigma (\text{NRC}_{\text{mod}})$</td>
<td>NRC</td>
</tr>
<tr>
<td></td>
<td>total cost of ERP exchange $P_{\text{TC}} \times \text{Implementation cost, migration, etc.}$</td>
<td>NRC</td>
</tr>
<tr>
<td><strong>Dependence on Internet Provider with $P_{\text{change line}}$:</strong></td>
<td>singular connection: $P_{\text{OL}} \times \text{time}$</td>
<td>NRC</td>
</tr>
<tr>
<td></td>
<td>OR multiple connections: $P_{\text{OL}} \times \text{time}$</td>
<td>NRC</td>
</tr>
<tr>
<td></td>
<td>Second internet line fee $P_{\text{OL}} \times \text{time}$</td>
<td>RC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Dependence on ERP Partner with $P_{\text{Provider Charge}}$:</strong></th>
<th>continuous use without support: $\Sigma (\text{RC}_{\text{OnPremise}})$</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\Sigma (\text{RC}_{\text{OnPremise}})$</td>
<td>RC</td>
</tr>
<tr>
<td></td>
<td>$\Sigma (\text{RC}_{\text{OnPremise}})$</td>
<td>RC</td>
</tr>
</tbody>
</table>

\[ \Sigma \text{NRC}_{\text{Change SaaS}} = \Sigma \text{RC}_{\text{Change SaaS}} \]

\[ \Sigma \text{NRC}_{\text{Change OnPremise}} = \Sigma \text{RC}_{\text{Change OnPremise}} \]

Table 3: Option calculation scheme; third part: change of requirements
(e.g. due to bankruptcy), then all the customer can do is to change the system and export the data to a common format (e.g. Excel tables or SQL DB). This incurs all the fixed costs of replacing an ERP.

The On-Premise system, on the other hand, may become outdated and therefore not supported anymore, but this does not imply an immediate requirement to change the system. The system can still be used without support from the ERP partner until it no longer meets the requirements of the accounting standards or the company’s own needs. Proceeding with the outdated system saves all further contractual maintenance and service costs, but incurs further internal continuation and maintenance costs for the work that had been done by the provider before the system was outdated. Otherwise, in the case of a premature change, the residual value of the old system which is written off has to be considered in addition to all the fixed costs incurred in replacing the On-Premise system. A SaaS system, in contrast, will never be outdated and will always run on the newest version, because a service and not a system is sold. There is no need to hold new features back for new ERP software versions as in an On-Premise ERP. Just the opposite is true: the new feature can only be rented, e.g. as a new module, when it has been made available in the SaaS-ERP system.

So, on one side, there is the replacement of the SaaS system if it is discontinued and on the other side there is the On-Premise becoming outdated, with two options: proceeding with the system or writing it off and replacing it. In both operation modes, any incurring costs may be estimated and multiplied by the respective probability of their occurrence. A risk assessment of the discontinuation or the outdated may help to get a more precise estimation of the likelihood of their occurrence.

Dependence on the internet: SaaS-ERP is, in addition, more dependent on the internet provider than an On-Premise-ERP, causing higher downtime costs when the internet service is interrupted. Highly downtime-critical customers can compensate by using multiple connection lines (fixed or mobile) from different providers. This diminishes the probability of downtime, but incurs the extra costs of two internet lines. On-Premise-ERP is only dependent on the internet for offices or plants outside its premises, reducing the internet downtime costs of the main site to zero. So when comparing SaaS- vs. On-Premise-ERP, the difference in downtime costs as triggered by an internet outage should be included, provided that the downtime costs are high, internet outages happen, and the probability of internet outage is known.

This scheme does not claim to be complete, especially because the criteria mentioned here result from general differences between SaaS and On-Premise. Many producer-specific differences between the operation modes may exist that are not captured here. In particular, to consider all differentiating factors, whenever this option scheme is applied, it should be supplemented by any situational specific differences and by any missing criteria. When applying a cost comparison between different operation modes, e.g. between ASP and SaaS, then the same method of comparison can be applied, but diverse systemic differences may be relevant. Therefore different costs have to be assessed and estimated with different operation modes before the calculation in the next section can be conducted.

4. Calculating the Differential Total ERP Costs

In principle, there are diverse methods of calculating total costs and each method has its advantages and drawbacks. This section uses a more process-related perspective based on the work of vom Brocke & Simons et al., which facilitates the variation of depreciation periods and interest rates, as well as enabling the inclusion of imputed interests 1.8. To get to the total cost, all recurring costs and all yearly depreciation values of the nonrecurring costs will be summed up. In addition to this, interest costs for financing the current residual values of the nonrecurring costs by debts or equity are added. When applying the total cost approach using the absolute costs, no distortion arises from calculating the interest costs. But this is not the case when using the differential TCO approach, except for the case where the same proportions of the nonrecurring costs are financed by equity and debts as in the absolute total cost approach. Hence, to avoid distorting the interest cost amounts, a debt-equity financing ratio that funds the nonrecurring costs by debts and equity in the same relative proportion as in the absolute calculation will be applied here. The ratios will most likely be different for each operation mode.

The yearly differential TCO is calculated by adding up the recurring costs with the nonrecurring costs, which will be distributed by linear depreciation. This will be achieved by discounting the nonrecurring costs over the average ERP lifetime; the residual value, which is the basis for the yearly interest, will decrease each year by an nth part per year 2. Secondly, the interest is calculated on the basis of the residual values and is assigned to the respective yearly recurring costs. Last but not least, all yearly total costs until the end of the average ERP lifetime have to be added up to obtain the differential total ERP operation mode costs.

4.1 Yearly Nonrecurring Cost Proportion and Yearly Recurring Costs

The recurring and nonrecurring costs identified in the previous chapter should be used to compare the ERP operation
mode options. This section calculates the relative additional costs of each operation mode, but the same method could be applied using the absolute total costs. To get the yearly operation mode option costs the following parameters are used:

\[
\begin{align*}
\Sigma NRC &= \Sigma RC + \Sigma NRC_{\text{Change}} \\
\Sigma RC &= \Sigma RC_{\text{Change}} \\
\Sigma F &= \Sigma F_{\text{Change}} \\
EF_{i} &= \text{interest rate of equity financing} \\
DF_{i} &= \text{interest rate of debt financing} \\
DFr &= \text{ratio of debt financing (percentage of all costs financed by debts)} \\
n &= \text{depreciation period; ERP lifetime}
\end{align*}
\]

The total additional recurring costs of each operation mode for each year can be determined by summing up the respective recurring costs. The nonrecurring costs should be depreciated over the whole lifetime of the ERP system, to get the yearly nonrecurring cost proportion (1). Interest rates of the residual nonrecurring cost value will be incurred, along with the depreciation amount to correct for the financing aspect of the nonrecurring cost residuals. The nonrecurring cost amount may be financed by equity or debt, but often a mixture of both financing forms is used. The relative cost accounting does not cover all costs, so it is necessary to keep the equity to debt financing ratio equal to the same proportions by equity and debt as in the absolute total cost accounting (2), (3):

\[
\begin{align*}
\text{Yearly expense of depreciation:} \\
dNRC &= \Sigma NRC / n \quad (1) \\
\text{Interest financed by equity:} \\
I_{E,Y(x)} &= EF_{i} \cdot \Sigma NRC_{T} \cdot (1 - \frac{1}{n}) \quad (2) \\
\text{Additional debt financing costs:} \\
I_{DF,Y(x)} &= (DF_{i} - EF_{i}) \cdot \Sigma NRC_{T} \cdot DF_{i} \cdot (1 - \frac{1}{n}) \quad (3)
\end{align*}
\]

\[\begin{align*}
\text{4.2 Yearly Total Costs}
\end{align*}\]

The yearly relative total cost amount for each operation mode can be obtained by adding up the recurring costs with the nonrecurring cost proportion (depreciation expense) and the interest.

In principle this formula can be applied to calculate the relative cost difference for each particular year and operation mode. Using the yearly total costs makes it possible for the user to change the interest rates or the ratio of debt financing for each year. The debt redemption is often faster than the depreciation period, so the ratio of debt financing may diminish from year to year until the whole ERP system is financed by equity. In this case, a different ratio of financing (DF_{Y(1)} = \frac{n-1}{n}) has to be applied for each year. Furthermore, interest rates, especially for debt financing, can vary over the depreciation period and have to be adjusted. Recurring or nonrecurring costs may change for each year as well. When nonrecurring costs are varied, then the depreciation fixed cost proportion has to be adapted to the new situation. The formula (1), which is the basis for interest calculation in (2) and (3), changes for the remaining years to:

\[
dNRC_{(T \text{ (new)})} = \left( \frac{(n-1) \cdot \Sigma NRC_{T} + \Sigma NRC_{\text{Change (new)}}}{n} \right) / (n-1)
\]

The changes in nonrecurring and recurring costs, as well as interest rates can rarely be foreseen, so in the option calculation the real costs must be estimated. Thus it often makes the most sense to keep as much as possible constant and to adjust these costs only when the predicted course of progression requires adaption to the real situation in the future. Therefore all the parameters will be kept constant in the next section, although the ratio of debt financing, which can be predicted in advance, e. g. with a redemption plan or by the expiration of the loan, could vary.

\[\begin{align*}
\text{4.3 Total Comparative Operation Mode ERP Costs}
\end{align*}\]

The comparative total costs over the whole depreciation period have to be calculated to compare the options between the ERP operation modes. The final result tells the user which of the two operation modes is cheaper. To get the relative total costs per operation mode over the whole period, one needs only to sum up the yearly total costs identified in the last section. So the following formula can be stated:

\[
\text{Total relative cost amount for the year } Y(x):
\]

\[
\begin{align*}
T_{\text{Comp,Y(x)}} &= \Sigma RC_{T} + dNRC_{T} + I_{E,Y(x)} + l_{DF,Y(x)}
\end{align*}
\]

\[
\begin{align*}
\text{3 Total amount financed by equity. Because } DF_{i} > EF_{i} \text{ always, the total amount can be added when only the additional interest } (DF_{i}-EF_{i}) \text{ is used for the additional debt finance costs. See (3).}
\end{align*}
\]
This allows a rearrangement and simplification of the formula in (6):

\[ T_{\text{Comp,Y}}(o) = \sum_{x=1}^{n} \left( \frac{\text{SNRC}_T}{n} + \frac{\text{IEY}_Y(o) + \text{IaDF}_Y(o)}{n} \right) \]

= \sum_{x=1}^{n} \left( \sigma \text{RCT} + \frac{\text{SNRC}_T + \text{IEY}_Y(o) + \text{IaDF}_Y(o)}{n} \right)

= n \sigma \text{RCT} + \Sigma \text{SNRC}_T + \sum_{x=1}^{n} (\text{IEY}_Y(o) + \text{IaDF}_Y(o)) \]

* (\Sigma \text{RCT}; \Sigma \text{SNRC}_T = \text{constant})

This formula cannot be simplified in the case of variable interest rates, variation in nonrecurring or recurring costs, or a change in the ratio of debt financing. Each year has to be calculated by its respective parameters, so that the results of each year can be summed up to the comparative total costs.

In the special case where all parameters are presumed to be constant, and the ratio of debt financing is kept constant until the end of the depreciation period, then the sum of the interest amounts can be rearranged by the Gaussian sum formula:

\[ \sum_{x=1}^{n} \text{IEY}_Y(o) = \text{EF}_1 \times \sum_{x=1}^{n} \Sigma \text{SNRC}_T \times \left( 1 - \frac{1}{x} \right) \]

\[ = \text{EF}_1 \times \Sigma \text{SNRC}_T \times \left( \frac{n+1}{2} \right) \]

\[ = \sum_{x=1}^{n} \text{IaDF}_Y(o) = \]

\[ = (\text{DF}_1 - \text{EF}_1) \times \Sigma \text{SNRC}_T \times \text{DF}_1 \times \sum_{x=1}^{n} \left( 1 - \frac{1}{x} \right) \]

\[ = \frac{\text{DF}_1 - \text{EF}_1}{2} \times \text{DF}_1 \times \left( \frac{n+1}{2} \right) \]

This allows a rearrangement and simplification of the formula in (6):

\[ T_{\text{Comp,Y}}(o) = \sum_{x=1}^{n} \left( \text{SNRC}_T \times \left( 1 - \frac{1}{x} \right) \right) \]

\[ = \Sigma \text{SNRC}_T \times \left( 1 + \frac{n \Sigma \text{SNRC}_T + \text{IEY}_Y(o) + \text{IaDF}_Y(o)}{\Sigma \text{SNRC}_T} \right) \]

\[ = (\text{EF}_1 + \text{DF}_R) \times (\text{DF}_1 - \text{EF}_1) \]

* (\Sigma \text{RCT}; \Sigma \text{SNRC}_T; \text{EF}_1; \text{DF}_1; \text{DF}_R = \text{constant})

5. Cost Comparison: Management Strategies and Conclusions

Functional selection first: The selection of ERP systems does not stop at functional criteria, especially when several ERP operation mode options are available. It is evident that the leading ERP candidate should be selected on the basis of an evaluation of the ERP candidates’ coverage of the functional requirements (cf. 1.8), because the ERP’s embedded structures have to comply with the organization’s embedded structures, so the ERP system has to be aligned to the institutional context of the company 1.8. Given a choice between a SaaS- and On-Premise-ERP system of the same preferred ERP candidate, the main question, answered in this contribution, comes into effect: which of these operation mode options is the best choice from a cost perspective in the long run in this particular case? This answer can be obtained through an option calculation scheme, which is not reduced purely to SaaS subscription costs. Diverse fixed and variable costs, which arise during the implementation and operation of the ERP system, were uncovered.

Internal ERP costs: The first step toward an option calculation scheme is to look at the current internal ERP costs, if the selecting ERP customer has one that is to be replaced. With this procedure the selecting customer is able to estimate all the internal maintenance and updating costs by listing all expenditures in a non-aggregated manner. Another strategy to get more precise cost estimations for the ERP system operation is to visit one of the preferred ERP partner’s customers, especially when the company has never had an ERP system before. During the visit, the customer may ask about the internal maintenance costs and the time spent in keeping the system operational. This information may help the customer get an idea of the cost dimensions. Further internal costs to be collected are the cost of the hardware, server, and server software, as well as the backup system, especially if they are to be renewed as well.

ERP and ERP integration costs: When all the internal costs of an On-Premise system are clear, a meeting should be arranged with the currently preferred ERP partner. In the meeting, the ERP partner should clearly explain the cost differences between the two ERP operation modes, emphasizing the license costs, maintenance fees, SaaS subscription costs, training costs, and implementation or preliminary project costs. Further discussion points should be the implications of customizations on future updates and maintenance and their concomitant costs, as well as the flexibility of each operation mode in light of changing requirements: module or user extension or reduction, bundles and scaling, minimum number of user accounts, waiting time before the next change period, etc. The customer should know as precisely as possible what

---

4 The proof for this rearrangement is given in the Annex.
These diverse soft and non-monetary factors, as well as the reference criteria to the particular company characteristics. Different operation modes, diverse soft and non-monetary factors influenced operation mode. Customer and the ERP partner to mitigate the costs for the pre

ferred operation mode.

sensitizes the ERP customer to all the cost differences, which can be seen as a good basis for negotiation between the ERP customer and the ERP partner to mitigate the costs for the preferred operation mode.

Besides calculating the financial advantage of the preferable operation mode, diverse soft and non-monetary factors should be included to align the more advantageous systemic difference criteria to the particular company characteristics. These diverse soft and non-monetary factors, as well as the cost comparison results are at least necessary to develop an overall ERP delivery strategy. Further information about the more qualitative operation mode selection is available in 1.8.

6. Limitations and Outlook

A method should serve a specific purpose, opting for advantages and disadvantages with the method selection. This method was aimed at ERP selecting end users who have little prior know-how and few human resources. The method is the only one directly aligned to this problem, but is not the only way of calculating ERP operation mode differences. Most of the investment calculation methods, e.g. Net Present Value or Discounted Cash Flow, could be applied as well. These investment methods are based on an absolute comparison, taking all costs into consideration. Using these methods for calculating the ERP operation mode cost differences may result in more accurate and better comparable results, but they are far more expensive and complex to apply. Hence, the main limitations of the proposed option calculation scheme come principally out of the method used. The differential cost approach enables comparisons only of pairs, not of third options. The exclusion of all the similarities carries the risk of overlooking company-specific differences, which are not included in this proposed general calculation scheme. And last but not least, the biggest difficulty remains the often time-consuming assessment or estimation of each single cost factor discovered here as a systemic difference between the operation modes 1.8. The lack of accessible data is one major disadvantage of the total cost of ownership approach 1.8. Neither the total costs nor their components can be determined in general, because the amounts depend on the company characteristics and requirements and therefore differ from case to case. The better the costs can be assessed, the more accurate results this cost calculation scheme can be expected to provide. Data from the experience of the ERP partner or customers working on the same system could help to fill this gap, but to overcome this weakness, research to find generalizable cost amounts in relation to the company’s characteristics is also encouraged.

Furthermore, the total cost of ownership method is not the best method to assess the costs and advantages of flexibility, imputed risks and gains, because the farther in the future the probability estimates are predicted, the rougher the assessments are. The method can therefore provide a feeling about the cost advantage, especially about the value of greater flexibility, but the real cost advantage may not be determined accurately with this method. Non-monetary risk assessment methods may be perhaps more constructive. Further non-monetary criteria such as ubiquity or collaborative gains are also excluded from the cost calculation scheme. Hence, to fill this gap, other methods are required, especially to be able to construct the qualitative part of the ERP delivery strategy, e.g. 1.8.

Another limitation of the calculation method is that the result is not tax-adjusted. An On-Premise system lengthens the balance sheet with the purchase and has thereafter to be taxed as an asset; this is not true for a SaaS-ERP. This cost difference was not considered due to multiple tax laws and rates of each state.

As mentioned, further research has to be qualitative, capturing not only the systemic differences but also the criteria, needs and requirements of the ERP customers by looking at the company’s characteristics. Since each customer will benefit from the systemic ERP operation mode differences only in relation to their company’s characteristics, the systemic differences should be compared with the company characteristics to build general claims about the situations in which each ERP operation mode is preferable. Additionally, this option calculation scheme, as well as all further qualitative findings and criteria will be applied to a concrete case, which will allow this scheme to be validated and refined. Any adjustments and refinements after the appliance are possible; the findings of the case study will form the basis for a system that will provide evidence supporting a decision between SaaS- and On-Premise-ERP on both strategic and financial grounds. For this purpose, more research and several further publications will be required.

Keywords
SaaS-ERP; On-Premise-ERP; Differences between SaaS and On-Premise; Calculation Scheme of IT-Options
References


Systemic Differences between SaaS- and On-Premise-ERP: An Overview of a Qualitative Option Calculation Scheme


Annex: proof for formula conversion

\[
\sum_{x=1}^{n} \left( 1 - \frac{x-1}{n} \right) = \sum_{x=1}^{n} \left( 1 + \frac{1}{n} - \frac{x}{n} \right)
\]

\[
= \sum_{x=1}^{n} \left( \frac{n+1}{n} - \frac{x}{n} \right)
\]

\[
= \sum_{x=1}^{n} \left( \frac{n+1}{n} - \frac{x}{n} \right)
\]

\[
= n \cdot \left( \frac{n+1}{n} \right) - \sum_{x=1}^{n} \frac{x}{n}
\]

\[
= (n+1) - \frac{1}{n} \sum_{x=1}^{n} x
\]

\[
= (n+1) - \frac{1}{n} \cdot \left( \frac{n(n+1)}{2} \right)
\]

\[
= \left( \frac{n(n+1)}{2} \right)
\]