

# Cloud ERP System Customization Challenges

Marko Mijač, Ruben Picek, Zlatko Stapić

Faculty of Organization and Informatics

University of Zagreb

Pavlinska 2, 42000 Varaždin, Croatia

{marko.mijac, ruben.picek, zlatko.stapic}@foi.hr

**Abstract.** *Customization is one of the known challenges in traditional ERP systems. With the advent of Cloud ERP systems, a question of determining the state of such systems regarding customization and configuration ability arises. As there are only a few literature sources partially covering this topic, a more comprehensive and systematic literature review is needed. Thus, this paper presents a literature review performed in order to give an overview of reported research on “Cloud ERP Customization” topic performed in the last 5 years. In two search iterations, a total of 32 relevant papers are identified and analyzed. The results show that several dominant research trends are identified along with 12 challenges and issues. Additionally, based on the results, the possible future researches are proposed.*

**Keywords.** Cloud computing, Customization, Configuration, Multi-tenancy, SaaS, ERP

## 1 Introduction

Enterprise Resource Planning System (ERP) is integrated information system with centralized database, which supports main business processes across organization. These systems are among most complex software solutions ever built, and are equally expensive. Traditional SaaS (Software as a Service) on-premise ERP systems imply significant investment in hardware infrastructure and purchasing software licenses. In addition, implementing ERP system in enterprise is costly, time-consuming, high risk process that extends over several years, and it often requires business process reengineering. This is supported by [1] [2] [3] as they summarize key reasons not to implement ERP: high cost, high capital investment, long implementation, limited IT resources, concerns over integration with legacy application, inflexible licensing model, upgrading issues, long ROI time, and success is never assured. These are some of the reasons why during years such systems were accessible only for large companies, and even they treated ERP acquiring as a top priority capital

investment. Small and medium enterprises (SMEs) simply didn't have such vast financial, time or human resources to take advantage of ERP benefits. However, with the advent of Cloud computing technology and accompanying service business models, chances are finally shifting towards SMEs.

Although ERP systems in cloud environment solve or mitigate a number of previously mentioned ERP issues, some issues still remain. One of these issues is customization issue. Preliminary research on Cloud ERP systems topic showed that customization in cloud potentially presents even a greater challenge than in on-premise ERP systems, and that it makes sense to conduct further research. Therefore, the goal of this paper is to reveal perceived challenges regarding customization of Cloud ERP systems by conducting a literature review.

Thus, the paper is organized in sections as follows. The second section covers Cloud ERP systems in general, its advantages and disadvantages, and gives an introduction to customization issue. The third section presents a scientific method used to review existing literature on this topic. The fourth section in details presents the findings and introduces a systematized report containing a glance overview on all papers found to be relevant. Finally, the last section answers the review question, gives hints on possible future research directions and concludes the topic.

## 2 ERP systems in Cloud

Offering various services at different layers is a major determinant of Cloud systems. This paradigm has gained such a momentum in a past few years, a term “anything as a service” or “everything as a service” (also known as XaaS) is coined. However, there are three types of services that are dominant in offerings of Cloud providers: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [4] [5] [6] [7]. Since Cloud providers offer ERP systems as ready software solutions to their clients, this falls into SaaS category of Cloud services.

## 2.1 Advantages and disadvantages

Cloud and SaaS technology brought new possibilities which began to be utilized to overcome some of traditional ERP system drawbacks. Enhanced business model (“pay-per-use”, “pay-as-you-go”) with significant cost reductions, technical breakthroughs (virtualization, XaaS, responsive Web applications...) that are improving performance, availability, user experience, are some of these possibilities. According to [7] [8] [9], reasons for adopting cloud technology in enterprise are: reduction of hardware and license costs, lower TCO (Total Cost of Ownership), transforming capital expenses (CapEx) to operating expenses (OpEx), no entry barrier, easier upgrading, easier implementation, increased scalability and maintainability, reduced requirement for IT department and IT personnel. McCrea [1] also states several reasons for going with a SaaS solution: leaner operation, greater utilization of resources, scalability, improved communication with customers and vendors, and cost structure. Saeed et al. [10] state several motives for Cloud ERP adoption: flexibility for business innovation, faster time to market, allows users to concentrate on core business, scalability, low capital expenditure, better support, reduced IT costs, automatic updates etc.

Beside obvious advantages of Cloud ERP systems, there are also several potential drawbacks and challenges. According to Salleh et al. [8] these are: security issues, flexibility, customization, ownership of data, provider “lock In”, etc. McCrea [1] also states several drawbacks to a cloud-based ERP: security concerns, system reliability, and system performance. Hofman [9] reports several trade-offs to which customers should be prepared: security, interoperability and lock-in, Absence of Service-Level Agreements, Performance Instability, Latency and Network Limits, No Scalable Storage, Innovation impairment.

## 2.2 Customization issue

Among other issues, the issue of customizing Cloud ERP system to meet client’s specific needs arises. Cloud ERP providers offer systems that are based on standard workflow, business best practices or most common way of doing business. Although this standardization has its own benefits, these are primary the means by which providers are increasing the number of potential customers. However we must face the fact that every enterprise is unique, and so are the requirements they have for software that supports their business processes. Johnson [11] supports this claiming that many companies customize their business processes trying to achieve better performance and efficiency. Attempting to roll back these processes to conform to standards can result in loss of competitive advantage. Although most Cloud ERP providers claim better innovation as one of

benefits of Cloud ERP systems, Hoffman [9] argues that cloud can actually impair the ability of enterprises to innovate. He claims that in order to innovate, enterprises must tailor ERP systems according to their unique needs. In the end, you achieve a competitive advantage by being different and better than your competitors, and not by conforming to same workflow.

## 3 Literature review and research question

Preliminary research on Cloud ERP systems topic showed that customization in Cloud ERP systems does present a challenge and controversy, and that it makes sense to conduct further, more detailed review of literature on this topic. However, considering this is quite new research field, conducting a full scale systematic literature review was not suitable. Nevertheless, in order to achieve more systematic and unbiased review of existing literature on Cloud ERP Customization topic, guidelines from systematic literature review method according to Kitchenham [12] were partially used. Also a following research question was posed:

*What are reported challenges in customization of Cloud ERP systems?*

### 3.1 Planning the review

In order to obtain recent relevant papers on the topic, several scientific databases were used: *ACM Digital Library, IEEE Xplore Digital Library, IEEE Computer Society Digital Library, Science Direct, Scopus, Web of Science, SpringerLink.*

The topic of the paper contains following keywords: *Cloud, ERP* and *Customization*. Also following synonyms or similar words were used: *Cloud – SaaS, ERP – “Enterprise Resource Planning”, customization – configuration*. Based on the keywords and their synonyms following generic search query was built: *(Cloud OR SaaS) AND (ERP OR „Enterprise Resource Planning“) AND (Customiz\* OR Configur\*)*.

Given that words “*customization*” and “*configuration*” can appear in different forms, *operator \** was used to take into account all relevant forms. Only papers published in the field of computer science and between year 2008 and present were taken into consideration. The search query was intended to run upon data contained in *metadata fields, abstracts, titles, and keywords*. However, due to slight differences in search engines of different databases, in some cases query adaptations were necessary. To further standardize search query upon different databases, *command search* was used wherever possible.

### 3.2 Conduction of the review

By executing queries on selected databases total of 34 papers were obtained. After evaluating given papers by *title* and *abstract*, more than half of them were discarded for being too general, and/or for covering irrelevant topics. Total of 13 papers were found partially relevant, but since some papers appeared in more than one database, after removing duplicates 10 papers left. In the following table detailed information about paper sources and obtained results are presented.

Table 1 Results of the first search iteration

	Database	Papers (Relevant / Found)
1.	ACM Digital Library	2 / 5
2.	IEEE Xplore Digital Library	4 / 6
3.	IEEE Computer Society Digital Library	0 / 0
4.	Science Direct	0 / 1
5.	Scopus	6 / 9
6.	Web of Science	0 / 0
7.	SpringerLink	0 / 13
	TOTAL	13 / 34
	<b>TOTAL (duplicates removed)</b>	<b>9 / 34</b>

However, by examining resulting papers, it was clear that no sufficient number of papers were obtained in the first search iteration. Also, more importantly, the papers did not adequately cover the topic. For this reasons, additional search was conducted. In the second iteration, aforementioned scientific databases were used along with Google Scholar search engine. This time, the queries were constructed from the same previous keywords pool, but keywords were permuted in different ways trying to obtain better results. Queries were often less restrictive than in the first iteration, aiming to get papers on Cloud ERP topic, and then manually examining the content of papers in search for customization topic. The second search iteration gave over 15 additional papers.

There are several possible limitations of this literature review that should be considered. The second search iteration was conducted in a less systematic manner than the first one, trying to obtain a larger number of relevant papers. Also, because of general lack of the literature on the Cloud ERP (especially customization) topic, a few non-scientific (but expert) sources were used (blog posts: Johnson [11] [13] and Kimberling [14] [15]).

## 4 Results of the review

A literature review that was conducted did not show a large number of papers dedicated solely to Cloud ERP

customization topic. However, a number of papers partially covered this topic, or at least addressed customization as a potential issue in Cloud ERP systems. The lack of scientific coverage of Cloud ERP systems topic is also reported by Elragal and Haddara [16] as they state that cloud computing is rarely discussed in ERP context.

In the following table papers obtained in first and second search iteration are presented and categorized according to content:

Table 2 Obtained papers enumerated by topics

Topics	Papers
Multi-tenant SaaS systems	[10], [13], [17], [18], [19], [20], [21], [22], [23], [24]
SaaS customization approaches	[17], [18], [19], [20], [21], [22], [23], [24] [25], [26], [27], [28],
Cloud ERP Customization issues and barriers	[3], [4], [6], [10], [11], [14], [15], [23], [28], [29], [30], [31]
Cloud ERP advantages and disadvantages	[1], [2], [3], [5], [6], [7], [8], [9], [10], [16], [22] [29], [30], [31], [32], [33],

### 4.1 The need for customization in Cloud ERP systems

Cloud ERP systems are slowly but surely penetrating into enterprises and are taking their part of the enterprise software market. McCrae [1] reports results of a survey, where over 200 manufacturers were inquired about adopting cloud platform for ERP system. 5 percent of respondents said that they already adopted SaaS software, 25 percent are evaluating the possibilities, while the remaining respondents were not sure how SaaS software could fit into their business strategy. Kimberling [14] argues customization to be one of the most controversial topics surrounding ERP software. According to his report only 23 percent of organizations implement plain vanilla ERP software with little or no customization. Following table shows average rate of ERP customization for four well known ERP providers.

Table 3 Rate of ERP customization [14]

	Heavy	Moderate	Vanilla
SAP	38,40%	40,60%	21,00%
Oracle EBS	34,40%	40,00%	25,60%
Microsoft Dynamics	32,80%	42,20%	25,00%
Tier II ERP	23,50%	48,10%	28,40%

This report clearly shows that customization of ERP systems is common and important feature. The data presented in this report refers to traditional ERP systems; however a question arises if the

customization needs of enterprises will significantly differ in Cloud ERP systems. Sun et al. [28] showed dependency between configuration and customization demand level and various SaaS solutions. Their results show that the increase in complexity of software also increases the demand for customizations. Considering ERP systems are among most complex software deployed as SaaS, it is reasonable to expect very high demand for customization.

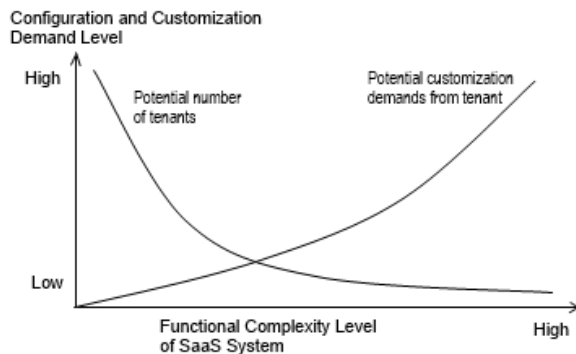


Figure 1 Dependency between customization demand level and complexity of SaaS solutions [28]

Sun et al. [28] state several causes of requirements variance among different customers: industry focus differences, customer behavior differences, product offering differences, regulation differences, culture differences and operation strategy differences. ERP systems usually go through substantial customization process to make them fit to the needs of particular company, and the feasibility of such adaptations in cloud systems should be addressed before the decision of using cloud based ERP systems is taken [6].

Wortmann et al. explore the analogy between delivering physical products in automotive industry and delivering enterprise information systems. Their research shows significant correlation in delivery stages of physical and service product. They also emphasize the concepts of componentization, cloud services, and use of single-code based configurable software in achieving enterprise software customization [27].

Lechesa et al. state that flexibility, achieved by configuration and customization, is necessary in ERP SaaS in order to increase adoption rate and to allow clients to differentiate their business from competition [29].

Conversely, some authors are skeptical about customization of Cloud ERP systems. Ried et al. [4] state that customization results in complex implementation projects and unpredictable behavior from badly configured application packages. Also, they claim most successful cloud vendors to be those that convince customers to adapt standardized application logic. Finally, Kimberling [14] as well

poses three reasons for ERP customization controversy:

- Customization increases complexity and risk of an implementation, which also leads in difficulty in upgrading software.
- Customization undermines best practices built into the software.
- Customization is often a symptom of bigger problems, such as solution's mismatch with company requirements, purchase of inadequate ERP product etc.

## 4.2 The lack of customization ability in Cloud ERP systems

As already stated, Cloud ERP vendors tend to utilize economy of scale by offering standardized solution to reach larger number of potential customers [10] [28] [21]. An issue arises when customers expect Cloud ERP systems to be tailored according to their specific needs, providing the same level of customization as on-premise ERP solutions. There are valid concerns that Cloud ERP systems provide less flexibility and customization options than traditional ones.

Saeed et al. [10] argue customization to be difficult in Cloud ERP systems, and consider it a technical barrier to Cloud ERP adoption. Such system is not under customer's control as they do not own, but only rent it. Also ERP system is deployed in cloud environment, which is much stricter than on-premise proprietary environment. However, some authors note that there is a discrepancy in this matter between ERP vendors' claims and academic reports, which needs to be further clarified. Jia [3] claims ERP systems delivered as SaaS do not provide enough ability to develop customization, and therefore present a challenge for customers with more demanding requirements. According to Schubert et al. [6] SaaS is only suited for software "out of the box", that does not require much customization or integration with other applications.

Muhleman et al. [32] in their paper discuss motives to adopt cloud enterprise solutions, and they give an advantage to in-house solutions in case of dealing with larger companies. As one of the reasons they state that in-house ERP solutions have increased customization ability, while with cloud-based SaaS solutions this is typically limited to the vendor.

Bibi et al. performed a SWOT analysis for migrating business software to cloud, and positioned limited customizability and limited configurability as weaknesses. They stated that on-premise software development focuses on customization as a means to market innovations, whereas cloud-based development restricts it to keep the total costs of operations low [33].

Kimberling in his post covers the question of suitability of SaaS ERP solution to one's organization. As one of the criterion he states level of

complexity and uniqueness of company's business. According to him, SaaS ERP solutions are still not as flexible as on-premise solutions, and it is more difficult to change SaaS offerings to fit your specific needs. If a company has unique and complex business processes, and requires heavy customization, on-premise ERP systems might present a better solution [15].

Jiang et al. [18] claim SaaS applications have pre-determined business processes, and customization is realized by also pre-determined parameter configurations, which can be inflexible to users' specific needs. Johnson [11] poses several characteristics of software that is "Easy to Customize":

- Software has well documented APIs,
- Software is written in standard or common programming language and platform,
- Software has SDK (Software Development Kit),
- Customizations are managed separately from core logic,
- Customizations occur at any time.

Although Johnson argues that no SaaS software satisfies those criteria, he also raises the possibility that customization is not critical for one's business, and that specific needs can be satisfied simply by configuring some aspects of ERP software [11].

Because of multi-tenant approach, Makkar and Meenakshi [30] also report customization to be complex and one of the main problems of Cloud ERP systems. They state that this kind of architecture constrains user's ability to customize application logic and interface according to their specific requirements.

Limited customization is reported to be one of the main barriers to adoption of ERP SaaS according to Lechesa et al., because ERP SaaS is often perceived as rigid. In their paper, they interviewed five highly positioned individuals from five different companies. In terms of customization respondents hold a view that ERP SaaS is more appropriate for standardized applications, with less customization demand. [29]

Iqbal et al. in their paper examine challenges in Cloud ERP implementation, using Markus and Tanis' ERP life cycle model. They conducted series of interviews with users, consultants and providers of Cloud ERP systems. Research findings recognized customization as one of the challenges in second phase of Cloud ERP life cycle. However, respondents also stated that Cloud ERP systems are customizable provided you choose appropriate vendor. [31]

### 4.3 Two approaches to Cloud ERP adjustment

Adjustment of Cloud ERP system to better suit your specific needs can be approached by *configuration* and *customization*. It is essential to distinguish between these two concepts. According to Kimberling [14], configuration is normal set-up of the software,

usual part of any implementation, and does not require changes to the source code. The same author implies that unlike configuration, customization requires changes to the source code, i.e. does not come as a prebuilt option.

Similar views on these concepts offer Sun et al., stating that configuration does not involve source code change; instead it supports change of application functionality within predefined scope such as adding data fields, changing field names, modifying UI, adjusting business rules. Conversely, customization requires change or upgrade of source code, to create functionality that is beyond configurable limit [28].

It should be said, however, that in most cases configuration is preferred of the two options. As Sun et al. state, customization is much more expensive for both provider and customer. On one side, customer pays large sums of money to provider or some third party consultants to customize solution; on the other side provider must deal with increasing complexity of software. However, the ability to adjust Cloud ERP with configuration depends on how much flexibility and different variants have providers initially incorporated into software. Anything that goes beyond that predefined scope must be solved by customization [28].

Sun et al. [28] propose Configuration and Customization Competency Model to describe SaaS software's variance level:

1. Entry (None) – Highly standardized offering with no configuration and customization support.
2. Aware (Low) – Relatively standardized offering with some predefined variance points.
3. Capable (Medium) – Relatively standardized offering with user defined configuration.
4. Mature (High) – Base offering with programmable environment to enable user preferred customization.
5. World class (Extremely high) – Offering programming platform and tools to enable extremely high customization and even new application development.

### 4.4 Multi-tenant Cloud ERP systems

The situation gets more complex as we adopt multi-tenancy as a desired environment type in SaaS, and therefore cloud-based ERP solutions.

Multi-tenancy is an architectural pattern in which a single instance of the software is run on the service provider's infrastructure, and multiple tenants access the same instance. On the other hand, in single-tenant environment every tenant has his own customized application instance [21].

This delivery model is focused on further exploiting economies of scale, by offering the same instance of an application to as many customers (tenants) as possible [22]. According to Bezemer et al. [21] the benefits of multi-tenancy are twofold for

service providers: (1) by deploying only one application instance instead of hundreds or thousands, system maintenance and deployment becomes easier; and (2) hardware utilization rate is improved as multiple tenants share the same hardware resources. These characteristics of multi-tenancy are allowing cloud providers to reduce overall costs, and to offer customers better deals. However, whether the system is single-tenant or multi-tenant should be completely opaque to customer. To customer must appear as if they are using dedicated resources. Mietzner et al. [22] support this claiming that multi-tenant awareness means that the software must appear to each tenant as if he was the sole tenant of the application.

Harris and Ahmed also present benefits of multi-tenant approach, such as: cheaper and easier updates, amplified consumption of hardware, highly configurable application, ease of maintenance etc. The same authors report following barriers and challenges: the cost of migrating single-instance model to multi-instance model, risk of wrong architectural choice, performance, scalability, security, zero downtime, maintenance concerns [19].

However, Johnson argues that there are many components in a system that can be multi-tenant, including application code, operating system, data storage, and computing resources. Consequentially, there are many varieties in ERP application depending on which of these resources are shared [13].

Mietzner et al. in their report introduce taxonomy of tenancy patterns. They evaluate the properties of single instance, single configurable instance, and multiple instances patterns with regard to several criteria. Single instance pattern assumes no variance in customers' requirements, so all tenants can use single instance of service. Single configurable instance pattern presents a service that is configurable enough to satisfy varying customers' needs, so again all tenants use single instance of service. Multiple instance pattern assumes each customer has very specific requirements, or underlying platform does not support multi-tenancy. Nevertheless, in this case each tenant must have its own service instance [22] [23].

Unfortunately, although multi-tenancy presents some advantages, it greatly increases the complexity of ERP systems implementation. With regard to customizability Mietzner et al. state that in single configurable instance pattern (which corresponds to idea of multi-tenancy) customizability is only partially solved. Tenant non-specific parts of service can be updated at once for all tenants; however, updating of tenant-specific configuration is harder, as it requires redeployment of configuration data for each tenant. Also, sometimes it is not possible to achieve tenant specific behavior by means of configuration, because business logic is completely different for different tenants [23].

As Bezemer et al. [24] state, multi-tenant systems should have high degree of configurability and/or customizability, however, this might introduce

maintenance problems due to additional complexity. In single-tenant systems issues such as customization, configuration and versioning are usually solved by creating branches in the development tree. However, in multi-tenant systems this is no longer possible. Instead features like this must be integrated in application architecture, which inherently increases the complexity of code and difficulty of maintenance. These are some of the reasons why multiple versions of applications may be necessary to coexist [21] [24].

However, despite the increase in code complexity Bezemer et al. believe multi-tenancy to be maintenance dream. To realize that goal, they emphasize the role of proper implementation of such systems, especially layered application architecture [24].

Kabbedijk and Slinger [17] identify three levels of variability in multi-tenant systems:

- Low level – variability in look and feel, visual presentation of system,
- Medium level – feature variability, changes in software workflow and logic tier,
- High level – full variability influencing multiple tiers at the same time, and allowing tenant to run their own code.

According to Bezemer et al. [24] multi-tenant systems should allow following types of configuration:

- Layout Style - The layout style configuration component allows the use of tenant-specific themes and styles.
- General configuration - The general configuration component allows the specification of tenant-specific configuration, such as encryption key settings and personal profile details.
- File I/O - The file I/O configuration component allows the specification of tenant-specific file paths, which can be used for, e.g., report generation.
- Workflow - The workflow configuration component allows the configuration of tenant-specific workflows.

#### 4.5 Existing customization approaches

Jiang et al. state that SaaS platform can be easily customized as they propose a SaaS Application platform based on Model-driven approach as a solution to customization and integration issues. According to them, customization is done on three layers: user interface layer, business process layer, and data layer. Customization in all three layers is achieved by deriving new, user specific models, which are kept in separate files for each user. Each model is accessible solely by user's ID. Data is kept in shared database with shared scheme, but the user has access only to a data and a scheme that is associated to its unique ID. The main determinant of

this approach is use of model to avoid explicit writing of application code [18].

Zhu and Wang [26] also propose MDA-based approach to customization of software in SaaS. However, their approach emphasizes the use of SOA principles.

Borovski and Zeier state that although adopting SaaS ERP does reduce TCO, it does not reduce customization costs. They claim that SaaS adaption is traditionally done via customer extensions or “*composite applications*” built on top of standard ERP systems. In their paper they propose ERP system architecture that supports development of this kind of *composite applications*. However, since these applications heavily depend on ERP implementation, an issue arises when implementation is changed or upgraded. Authors also emphasize ERP data accessibility as the most important factor in realization of composite applications. They introduce a concept of BOQL (Business Object Query Language) as a way of accessing data in their proper form – business objects. [25]

Kabbedijk and Slinger [17] examine three design patterns used to achieve variability in multi-tenant systems: *Customizable Data Views* pattern, *Module Dependent Menu* pattern, *Pre/Post Update Hooks* pattern. While the first and the second pattern deal with configuring visual presentation, the third one enables changing business workflow by hooking additional modules before or after data update.

Harris and Ahmed propose architectural blueprint for building customizable multi-tenant SaaS solution, based on MVC pattern. They emphasize that SaaS functions should be available to different tenants as a set of well-defined services, leveraging technologies such as SOA [19]. Authors Jing and Zhang [20] also present a solution that relies on SOA. Their Open SaaS Software Architecture (OSaaS) provides tenants with tools (UML, BPEL) to build their own business processes, to share and collaborate with their partners.

## 5 Conclusions

The main goal of this paper was to determine a current research state on Cloud ERP customization topic, and to reveal potential challenges and issues. Literature review performed in this report showed the lack of scientific coverage of Cloud ERP systems. Perhaps, the reason for this can be found in the fact that cloud technology only just began its penetration in ERP systems market. In addition, the occurrence of customization topic is even less common. This can be explained by the fact that cloud technology brought various concerns that at first glance may seem more important to deal with, such as security concerns, performance issues and reliability.

Customization is a known issue and challenge in traditional ERP systems. However, according to review results there is no indication that the situation

with Cloud ERP systems significantly differs, and that the need for customization in Cloud ERP is less than in traditional ones.

Number of authors addressed a customization of Cloud ERP systems as an important issue, and possibly a greater challenge than it was in traditional ERP systems. This seems logical as traditional ERP systems were usually deployed as on-premise systems, which were fully controlled by the customer. In this environment, where ERP was built on proprietary infrastructure, enterprise could engage vendor or a third party consultants to adapt the system to better suit their needs. In no way the needs and requirements of other customers or enterprises were of its concern.

In Cloud – SaaS implementations of ERP systems the situation is quite different. This is especially the case with multi-tenant systems, where all tenants (different customers, enterprises) share the same resources and the same application instance. Here, several issues appear. Preferred way of adapting Cloud ERP system is by configuring various predefined system points. This means that vendor must keep in mind all possible variants of system that customer could request. This introduces additional complexity in already most complex software systems.

Another issue arises when customer wants to implement a feature that doesn't even exist in standard Cloud ERP implementation, or the feature in which the extent of changes is so great that surpasses the capabilities of configuration. Such changes require customization, changes of application code. Some of the questions that inevitably arise here are:

- How open is vendor's application codebase? Who can view and change application code?
- Are all parts of the system customizable?
- How easy is the system customizable?
- How will customer's specific implementation be separated from vendor's standard implementation and other customized implementations?
- How will the compatibility with future versions of system be assured?

A number of authors discussed customizability in multi-tenant SaaS applications, but these researches mainly covered systems such as email clients, office utility software, CRM systems. All of these systems are significantly simpler than a full scale ERP system, which raises a question of suitability of methods presented in these researches for ERP domain.

Most of the analyzed research papers showed a certain amount of skepticism towards configuration and customization capabilities of Cloud ERP systems. On the other hand, Cloud ERP systems' providers boast with configuration and customization capabilities of their products. Although we cannot talk about Cloud ERP vendors' perspective as unbiased one, claims of research community are also seldom supported with empirical evidences. Nonetheless, a

discrepancy between vendors and academic community obviously exists.

To sum up and to answer to proposed research question, following challenges and issues in Cloud ERP customization topic are identified:

Table 4 Reported customization challenges and issues in Cloud ERP

Challenges and issues	
1.	Potentially high level of customization requirements.
2.	Inflexibility and inability of Cloud ERP to support all specific customization and configuration needs.
3.	Increase in complexity of Cloud ERP due to implementation of different predefined configurations and ad hoc customizations.
4.	Difficulties in managing the system due to the increasing complexity.
5.	Difficulties in separating configurations and customizations of different tenants.
6.	Difficulties in upgrading software and keeping customizations compatible with future releases.
7.	Absence of proper SDK for easier development of customizations.
8.	Closed or limited Cloud ERP solution's codebase.
9.	Complexity of multi-tenant solutions.
10.	Heavy customizations are expensive and can be a sign of purchasing inadequate Cloud ERP system.
11.	There can be a security issue in implementing innovative business technology as customizations in Cloud ERP.
12.	Customizations can cause stability issues and performance problems.

Since research in this area is quite scarce, a number of further research topics can be identified:

- a clear framework to determine the level of configuration and customization ability of Cloud ERP systems,
- comparisons of different Cloud ERP solutions regarding the level of customizability and configurability,
- an empirical case studies evaluating customizability of specific Cloud ERP solutions,
- suitability evaluation of multi-tenancy implementation approaches for use in Cloud ERP systems.

In the light of these possible future research topics, this paper presents the base in current research state in Cloud ERP customization with identified challenges and issues.

## References

- [1] B. McCrea, "Putting the spotlight on ERP," *Logistics Management*, pp. 32 – 35, Jun-2011.
- [2] S. Shukla, S. Agarwal, and A. Shukla, "Trends in Cloud-ERP for SMB's : A Review," *IJNIET*, vol. 1, no. 1, Jul. 2012.
- [3] S. Jia, "Integrating Conventional ERP System with Cloud Services," Master Degree, Stockholm University, Stockholm, 2009.
- [4] S. Ried, H. Kisker, and P. Matzke, "The Evolution Of Cloud Computing Markets," Cambridge, 2010.
- [5] S. L. Saini, D. K. Saini, J. H. Yousif, and S. V. Khandage, "Cloud Computing and Enterprise Resource Planning Systems," presented at the World Congress on Engineering 2011, London, UK, 2011, vol. 1.
- [6] P. Schubert and F. Adisa, "Cloud Computing for Standard ERP Systems: Reference Framework and Research Agenda," *Fachbereich Informatik*, no. 16/2011, 2011.
- [7] G. Suci, E. Ularu G., and R. Craciunescu, "Public versus Private Cloud Adoption – a Case Study based on Open Source Cloud Platforms," presented at the Telecommunications forum TELFOR, Belgrade, 2012, pp. 494 – 497.
- [8] S. Salleh Maliza, S. Teoh Yen, and C. Chan, "Cloud Enterprise Systems: A Review of Literature and its Adoption," presented at the PACIS 2012 Proceedings, Hochiminh City, 2012.
- [9] P. Hofmann, "Cloud Computing: The Limits of Public Clouds for Business Applications," *IEEE Internet Computing*, vol. 14, no. 6, pp. 90–93, 2010.
- [10] I. Saeed, G. Juell-Skielse, and E. Uppström, "Cloud Enterprise Resource Planning Adoption: Motives & Barriers," in *Advances in Enterprise Information Systems II*, Aalborg, Denmark, 2012.
- [11] D. Johnson, "Customizing ERP on SaaS and Cloud Platforms," *ERP Cloud News*, 27-Oct-2009. [Online]. Available: <http://erpcloudnews.com/2009/10/customizing-erp-on-saas-and-cloud-platforms/>. [Accessed: 05-Feb-2013].
- [12] B. Kitchenham, "Procedures for Performing Systematic Reviews," Eversleigh, 2004.
- [13] D. Johnson, "Multi-tenant versus Single-tenant ERP – a comparison," *ERP Cloud News*, 2010.
- [14] E. Kimberling, "ERP Software Customization: The Ultimate Sin of Enterprise Software?," *Panorama Consulting Solutions*, 11-Dec-2009. [Online]. Available: <http://panorama-consulting.com/erp-software-customization-the-ultimate-sin-of-enterprise-software/>. [Accessed: 06-Feb-2013].



- [15] E. Kimberling, "Is SaaS ERP right for your organization," *360° ERP Blog*, 2011. .
- [16] A. Elragal and M. Haddara, "The Future of ERP Systems: look backward before moving forward," presented at the CENTERIS 2012, Algarve, Portugal, 2012.
- [17] J. Kabbedijk and J. Slinger, "Variability in multi-tenant environments: Architectural design patterns from industry," presented at the 30th International Conference on Conceptual Modeling, Brussels, 2011, vol. 6999, pp. 151 – 160.
- [18] X. Jiang, Y. Zhang, and S. Liu, "A Well-designed SaaS Application Platform Based on Model-driven Approach," presented at the Ninth International Conference on Grid and Cloud Computing, Nanjing, Jiangsu, P.R. China, 2010.
- [19] I. S. Harris and Z. Ahmed, "An Open Multi-Tenant Architecture to Leverage SMEs," *European Journal of Scientific Research*, vol. 65, no. 4, pp. 601–610, 2011.
- [20] J. Jing and J. Zhang, "Research on Open SaaS Software Architecture based on SOA," presented at the 2010 International Symposium on Computational Intelligence and Design, Hangzhou, China, 2010, vol. 2, pp. 144 – 147.
- [21] C.-P. Bezemer, A. Zaidman, B. Platzbeecker, T. Hurkmans, and A. Hart, "Enabling multi-tenancy: An industrial experience report," presented at the IEEE International Conference on Software Maintenance (ICSM), Timisoara, 2010, pp. 1–8.
- [22] R. Mietzner, A. Metzger, F. Leymann, and K. Pohl, "Variability modeling to support customization and deployment of multi-tenant-aware Software as a Service applications," presented at the Principles of Engineering Service Oriented Systems, 2009., Vancouver, BC, 2009.
- [23] R. Mietzner, T. Unger, R. Titze, and F. Leymann, "Combining Different Multi-Tenancy Patterns in Service-Oriented Applications," presented at the IEEE Enterprise Distributed Object Conference, Auckland, 2009.
- [24] C.-P. Bezemer and A. Zaidman, "Multi-tenant SaaS applications: maintenance dream or nightmare?," in *Proceedings of the Joint ERCIM Workshop on IWPSE-EVOL '10*, Antwerp, Belgium, 2010, pp. 88 – 92.
- [25] V. Borovskiy and A. Zeier, "Enabling enterprise composite applications on top of ERP systems," presented at the Asia-Pacific Services Computing Conference APSCC, Shanghai, 2009.
- [26] X. Zhu and S. Wang, "Software Customization Based on Model-Driven Architecture Over SaaS Platforms," presented at the Asian-Pacific Services Computing Conference APSCC, Shanghai, 2009.
- [27] J. C. Wortmann, H. Don, J. Hasselman, and A. Wilbrink, "Enterprise Information Systems as a Service: Re-engineering Enterprise Software as Product-Service System," in *Advances in Production Management Systems. Value Networks: Innovation, Technologies, and Management*, vol. 384, Stavanger, Norway: Springer Berlin Heidelberg, 2012, pp. 496–505.
- [28] W. Sun, X. Zhang, C. Jie Guo, P. Sun., and H. Su, "Software as a Service: Configuration and Customization Perspectives," presented at the Congress on Services Part II, Beijing, 2008.
- [29] M. Lechesa, L. Seymour, and J. Schuler, "ERP Software as Service (SaaS): Factors Affecting Adoption in South Africa," presented at the CONFENIS, Aalborg, Denmark, 2011, pp. 152–167.
- [30] G. D. Makkar and B. MEENAKSHI, "EAAS - ERP AS A SERVICE," *Journal of Information and Operations Management*, vol. 3, no. 1, pp. 141 – 145, 2012.
- [31] U. Iqbal, E. Uppström, and G. Juell-Skielse, "Cloud ERP implementation challenges: A study based on ERP life cycle model," in *Advances in Enterprise Information Systems II*, Aalborg, Denmark, 2011, pp. 389–405.
- [32] R. Muhleman, P. Kim, J. Homan V., and J. Breese-Vitelli, "Cloud Computing: Should I Stay or Should I Cloud?," presented at the Conference on Information Systems Applied Research, New Orleans Louisiana, USA, 2012.
- [33] S. Bibi, D. Katsaros, and P. Bozaris, "Business Application Acquisition: On-Premise or SaaS-Based Solutions?," *IEEE Software*, vol. 29, no. 3, pp. 86 – 93, 2012.