RESEARCH REPORT

VTT-R- 05936-10

Challenges and Alternative solutions for ERP’s

Authors:	Juho Eskeli, Samuli Heinonen, Tapio Matinmikko, Päivi Parviainen, Pasi Pussinen
Confidentiality:	Public

Report's title Challenges and Alternative solutions for ERP's	
Customer, contact person, address VTT	Order reference
Project name Yrtti	Project number/Short name Yrtti
Author(s) Juho Eskeli, Samuli Heinonen, Tapio Matinmikko, Päivi Parviainen, Pasi Pussinen	Pages 55/55
Keywords ERP systems	Report identification code VTT-R- 05936-10

Summary

This document addresses the current state of enterprise resource planning systems (ERP's), including the functionality, acquisition process and the challenges related to taking the systems into use. The work done included study of existing published material (publications, internet) and interviews of six different kinds of organisations. The purpose of the work was to find out the current practices in ERP acquisition and roll-out, as well as improvement possibilities relating to the ERP systems themselves and the processes related to ERP acquisition, roll-out and use.

Confidentiality	Public

Oulu 15.7.2010

Written by	Reviewed by	Accepted by
[illegible]	[illegible]	[illegible]
Pasi Pussinen Research Scientist	Tapio Matinmikko Senior Research Scientist	Matias Vierimaa Technology Manager

VTT's contact address
Kaitoväylä 1, 90571 Oulu

The use of the name of the VTT Technical Research Centre of Finland (VTT) in advertising or publication in part of this report is only permissible with written authorisation from the VTT Technical Research Centre of Finland.

Contents

List of abbreviations

CRM	Customer Relations Management: refers to the methodologies and tools that help businesses manage customer relationships in an organized way.
ERP	Enterprise Resource Planning: attempts to consolidate all of a company's departments and functions into a single computer system that services each department's specific needs.
SaaS	Software-as-a-Service: is a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own web servers or download the application to the consumer device, disabling it after use or after the on-demand contract expires. The on-demand function may be handled internally to share licenses within a firm or by a third-party application service provider (ASP) sharing licenses between firms.
SCM	Supply Chain Management: is the management of a network of interconnected businesses involved in the ultimate provision of product and service packages required by end customers (Harland, 1996). Supply Chain Management spans all movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption (supply chain).
DSM	Domain-Specific Modelling: Domain-Specific Modeling raises the level of abstraction beyond programming by specifying the solution directly using domain concepts. The final products are generated from these high-level specifications. An example of a Domain-Specific Modeling Language (DSML) is SysML, which is a subset of UML that is customized for systems engineering applications.

1 Introduction

The current situation in the enterprise resource planning and management (ERP) systems is that the market is dominated by systems that are perceived expensive and inflexible. The Yrtti - project aimed to find out possibilities to challenge the monopoly position of the existing commercial, license-based ERP systems. This document describes the work done and the main findings.

1.1 ERP Systems

The ERP systems as a whole are meant to support enterprises in various phases of the business, e.g., logistics, billing, sales and supply chain management. ERP systems include software applications that are typically integrated via shared data warehouse. The application software typically includes warehouse management, material management, production planning and management, people management, purchase management, customer relations management (CRM), and supply chain management (SCM). Figure 1 illustrates the ERP systems as a whole.

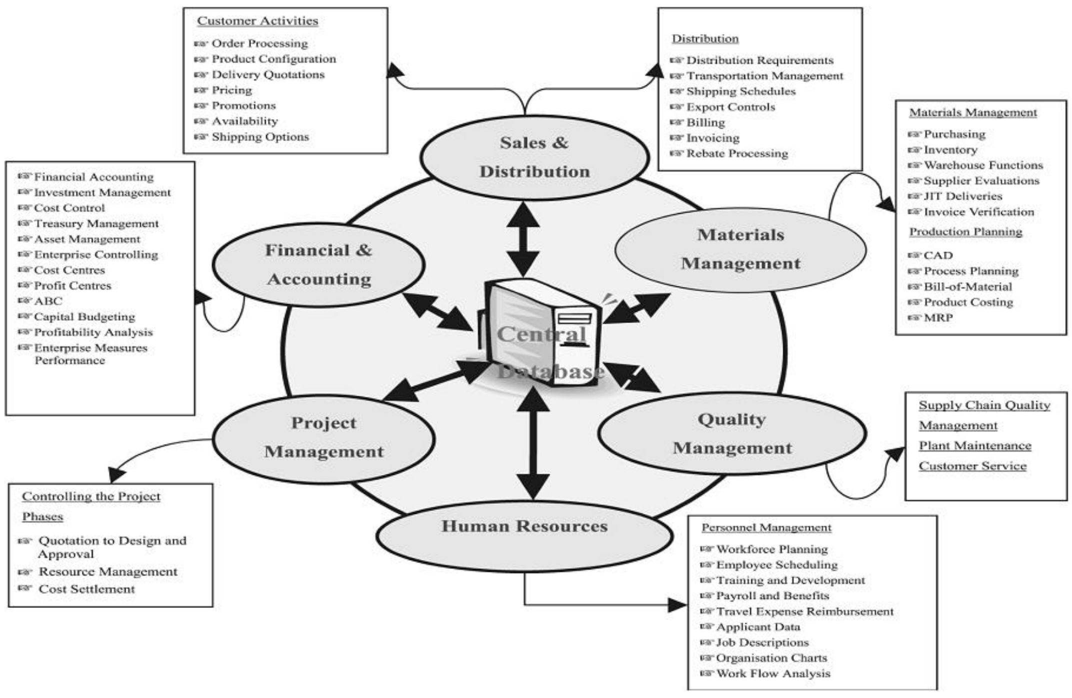


Figure 1. ERP Systems[1]

1.2 The Yrtti project

The Yrtti project's aim was to study the potential of using cheaper license-based and/or open source ERP systems. The final aim of the work was to find possibilities to improve the quality of the systems, enabling enough supplier independency and lowering the purchasing and running costs of the systems.

[1] Shehab E.M, Sharp M.W., Supramaniam L. & Spedding T.A. 2004. Enterprice Resource Planning – An integrative review. Business Process Management Journal. Volume 10, Issue 4. 359-385.

Open source software has significant potential to respond to the current challenges of information systems in enterprises as well as in public sector. These challenges are related to, e.g., dependencies to the suppliers/vendors (vendor lock causing the high costs of acquiring, tailoring and maintenance of the systems). There is a multitude of open source software and tools, and they are also already in use in enterprises, e.g., in the operating system layer (e.g., Linux), application service layer (e.g., Apache) and database layer (e.g., MySQL). However, open source operational systems, such as ERP, are not yet in wide spread use.

In addition to the open source alternatives, an option can also be partly or completely license-based models, but they need to be more flexible than the current market leaders.

The project aimed to:

- identify the factors that make the current ERP systems inflexible and difficult to use from the users viewpoint
- evaluate the applicability of open source tools and way of thinking also in operational systems
- present potential solutions in moving towards more flexible and more open systems and models to work with those systems.

This document is structured as follows: Section 2 describes the market status of ERP systems in use and gives an overview of the costs related to ERPs. Section 3 discusses the main findings of interviews held in order to capture the challenges faced in practise when taking into use an ERP system, and the practitioner’s views of potential improvements with the systems. Section 4 describes alternatives for the currently used systems. Section 5 provides feature comparison between two alternative systems, Severa (that is provided as Saas) and OpenERP and open source alternative ERP. The appendices of the document describe detailed evaluation results of four selected open source or SaaS tools.

2 ERP-systems in use

2.1 Proprietary ERP’s

In proprietary ERP systems the global market leader is SAP AG, in 2007 it had a 28.7% share of the markets. Other well known vendors for 2007 were[2]:

- Oracle (10.2%)
- The Sage Group (7.4%)
- Microsoft (7.2%)
- SSA Global Tech (2.8%)
- Others (47.2%).

Similarly to these numbers, according to the marker research company Gartner, the market share for SAP in 2007 was 27.5%[3]. For the year 2008, SAP announced in their own annual report that market share was 32.8% with yearly growth of 4.4%. SAP’s annual report also acknowledged that biggest competitor was Oracle with a 17.5% market share, these numbers are pointed in Figure 2.

[2] http://www.accountingsoftwarereview.com/ERP-Software-Market-Share-Analysis.html
[3] http://www.sap.com/solutions/business-suite/scm/newsevents/press.epx?pressid=9913

Even thought market shares vary according to the source, it can be said that SAP controls roughly one third of the ERP market.[4]

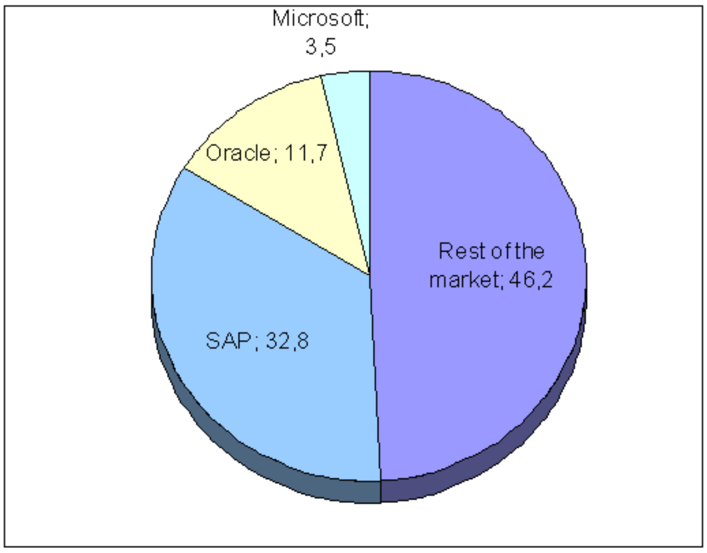


Figure 2.ERP market shares according to SAP Annual Report 2008

The majority of the market seems to be divided between small local vendors, which explain the market share for “others” in Figure 2 or the “rest of the market”. This is most likely because the systems, which market leaders are offering, are too big and complex for small and medium sized companies. Only large corporations can benefit from those.

2.2 ERP in Finland

In Finland, the ERP markets are divided with local vendors providing ERP systems for SME’s and large organizations using international brands. For small companies (less than 100 employees) the market shares in 2007 are presented in Figure 3.

[4] http://www.sap.com/about/investor/reports/annualreport/2008/pdf/SAP_2008_Annual_Report.pdf

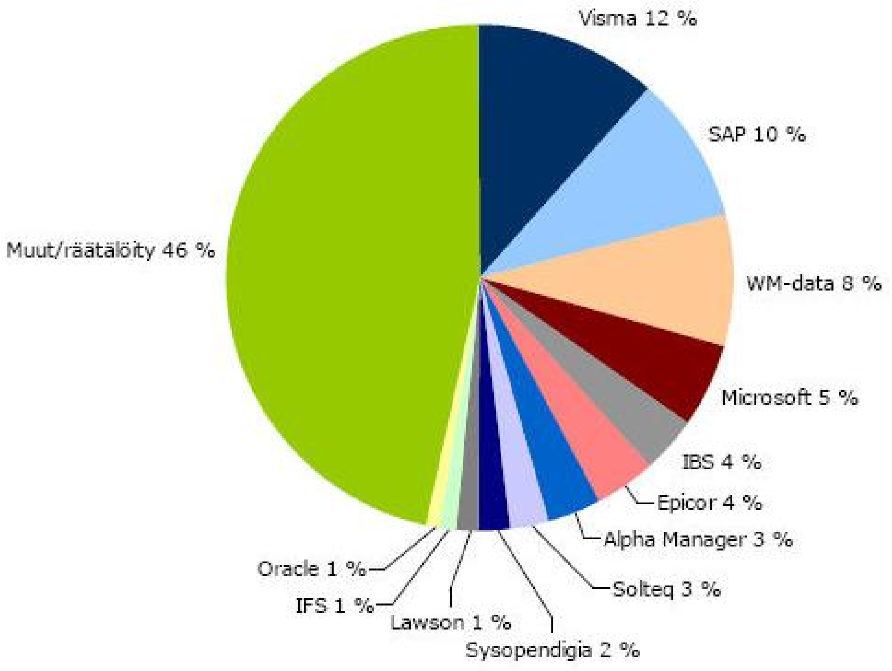


Figure 3 Market shares of ERP systems for small companies in Finland[5]

For medium sized companies (100 – 500 employees) the market shares in 2007 are presented in Figure 4

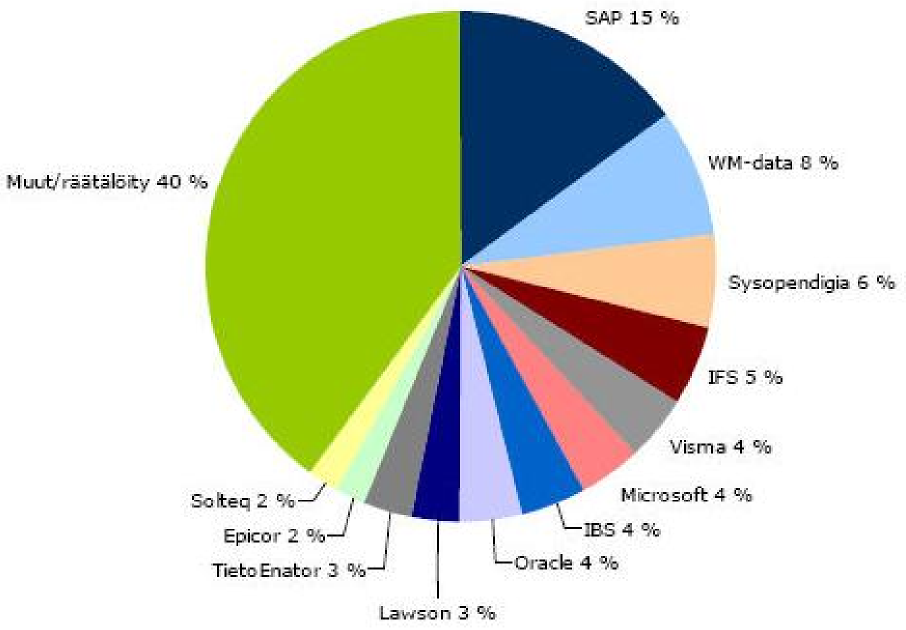


Figure 4. Market shares of ERP systems for medium sized companies in Finland[6]

For large organizations the ERP-market in 2007 was clearly owned by SAP, as presented in Table 1.

[5] http://www.digitoday.fi/bisnes/2008/03/04/sap-kasvu-pohjoismaissa-12-kertaa-markkinaa-nopeampi/20086616/66
[6] http://www.digitoday.fi/bisnes/2008/03/04/sap-kasvu-pohjoismaissa-12-kertaa-markkinaa-nopeampi/20086616/66

Table 1. Market shares of ERP systems for large organizations in Finland[7]

Company	Market share
SAP	48%
Logica	4%
Tietoenator	4%
SysopenDigia	4%
Oracle	4%
Lawson	4%
IFS	3%

In Finland the market leaders for ERP related services (implementation and customizations) are Tieto, WM-Data and Fujitsu Services.[8]

2.3 Costs of ERP

Annual ongoing costs related to ERP system can be divided into three different groups:[9]

1. Internal Costs: How much of the company's own resource are used in the adoption of an ERP system (training, IT support, financing…)
2. External Costs: Costs from using 3rd party services (consultancy, IT vendors, contractors used to modify ERP)
3. Maintenance and support: Fees paid to annually to application vendor.

An average distribution of these costs, according to the study by CFO Research Services and Agresso, is as follows:

• Internal costs:	475 500$	(39% of total costs)
• External costs:	339 250$	(27% of total costs)
• Maintenance and support:	424 750$	(34% of total costs)

Other study indicates that the share of license fees from the total cost of ERP is around 16%--33%.[10] A French study on open source ERP's[11] estimates the share of license fees to be around 25% - 50% of the annual costs. The high share of license fees can be explained for small and medium sized companies, who do not have to use so much external services or train separate IT-personnel. Therefore for SME's, the license fee is usually a larger share of the total annual costs.

[7] http://www.digitoday.fi/bisnes/2008/03/04/sap-kasvu-pohjoismaissa-12-kertaa-markkinaa-nopeampi/20086616/66
[8] http://www.itviikko.fi/talous/2007/10/16/erp-markkinoiden-jakautuminen-yllatti-aaran/200725623/7
[9] http://www.agresso.com/files/CFOResearch_Cost_of_Change_Report.pdf
[10] ERP systems and open source: An Initial review and some implication for SME's. Johansson & Sudzina. Center for Applied ICT, Copenhagen Business School. 2008.
[11] Valyi: Livre blanc: ERP Open Source (2008). Smile. http://www.smile.fr/publications/livres-blancs/erp-open-source

ERP systems are known to require modification according to the business needs, this explains the amount of the external costs and some of the large share of internal costs required to upkeep an ERP system. Since the ERP systems are customized and lots of resources are used to do this, there is a lock-in to an ERP system. It is not easy to switch to another supplier; it might even not be technically possible. The amount of licensing fees can be raised to generate more profits for the vendor, and customers can not save costs by switching to another vendor.

3 Empirical evidence

For this research, six different organizations were interviewed. The interviewees were IT management directors of one SME from manufacturing industry, two large public authorities and three large companies from manufacturing industry. The interviews were done as semi-structured interviews and questions are added as “Annex D” to this document.

Purpose of these interviews was to discover:

- How the companies have prepared to the acquisition of an ERP
- What were the factors that required special attention
- What kind of ERP’s were evaluated
- How did the implementation of ERP-system succeed
- How did the companies achieve optimal use of an ERP-system
- What could be done better in the ERP project

In this section we focus on the results from these interviews and present opinions stated in those interviews.

3.1 Greatest challenges

The single greatest challenge with the acquisition of an ERP system is to match business processes with those of an ERP system. In most cases of the interviews, business processes within the companies were not clearly defined or outlined. This will implicate trouble in the ERP project, as the functionality of an ERP system is based to the use of these processes. Customizations are done to the ERP systems to accommodate the system for a specific organization, but it’s never a complete match and some business process re-engineering is required. Usually it is the combination of both: ERP system customization and business process re-engineering that are required to achieve a successful ERP implementation. The interviewees were unanimous about the fact that a company which has all of their business processes clearly defined has the best chances to successfully implement an ERP system.

A third party agent is usually used to implement and customize the ERP system, the vendor of the system seldom does that themselves. Their role is important, has they have to have deep technical knowhow of the ERP system in order to successfully do customizations but they also need to know the customer’s field of industry very well. These third party agents received some criticism in the interviews, as their lack of professionalism had caused some delays the ERP project. In the worst case, it had caused the ERP system not to function properly as some modules hadn’t been installed. Most of the interviewees had actually changed this agent and this change was not considered to be an issue. However the change of the entire ERP system (and the original vendor) was considered to be a challenge; most interviewees were actually building a strategic relationship with the vendor. This means that companies were willing to commit to the ERP system and the vendor and even the third party agent. This was because ERP systems were considered to be business critical systems and long-term partnerships were preferred.

The motivation and commitment of the company personnel was considered to be critical. This includes all parts of the company personnel, from the top management to the middle management and to the employees. The more users the ERP system has, the less are the costs per user, the system is used efficiently and it produces real added value. Usually ERP projects face some change resistance from the company personnel, but this can be overcome with proper training and motivation. The commitment of the company personnel was considered to be the responsibility or priority of the top management.

Time schedules were usually too optimistic, and no proper timeframes were given for the ERP projects. Schedules were given from outside to the ERP project and were too tight. The proper amount of both time and resources were considered to be an important factor for ensuring a successful ERP project.

3.2 Can you affect the price or the quality of ERP?

Main drivers for cutting down costs in the ERP project were considered to be efficient project management and minimizing the amount of customizations. High amount of customizations causes a surge in the costs. The customizations can also cause more costs later, when the ERP system is updated and customization do not usually receive support for the update. This means that support for customization updates has to be purchased separately. Also more competition in the ERP systems market could lower the license fees.

The technical quality of ERP systems was considered to be satisfactory, biggest flaws were considered to be in the user interfaces. The consistency of ERP system also required improvement and as well as the adjustment of company specific parameters. One interesting possibility from the interviews was a guarantee for an ERP system.

3.3 Ways to improve the implementation and use of ERP

Some key factors for improving the implementation and use of ERP systems were mentioned in the interviews:

- *Schedule.* There should be enough time for the implementation process, without any pressure from outside. Any pressure from outside will cause disruptions to the implementation process, however securing sufficient amount of time is challenging in today’s business environment.
- *Sequencing.* Successful ERP projects are usually carried out in specific phases or sequences and the first phase is dedicated to basic functionality. This phase has to be done right, or else all other phases will fail as the

grounds for the system are not done right. In later phases more complex modules and supporting functions (like Human Resources) are added to system.
- *Processes.* The business processes of the company acquiring the ERP system must be clearly defined in order to be transferred to the ERP system properly. This means that the processes are actually followed within the organization, i.e., that it actually describes how things are done in the company. If this doesn't happen, the ERP system will not function properly or will produce false information.
- *Commitment and change resistance.* It's important to make the personnel committed to the use of ERP and minimize the change resistance. These are not technical matters, but more leadership and work psychology oriented matters.
- *Training.* The ERP system will not produce any benefits, if the personnel don't learn how to use it. This means different methods of training with the ERP system, but also a reminder of the role in the business process they are a part of. With the big picture in mind, it's easier to train the proper use of the ERP system. This will also help with the change resistance.
- *Usability.* This is often an overlooked matter, but usability was something the interviewees said should have received more importance. An ERP system that is hard to use faces more resistance than a system with smoother user interfaces.

It has to be noted, that the problems with ERP projects are common and well known. Also plenty of information is available about these problems. Still many ERP projects are only a partly successful and many others fail. Seems that the single most important factor with an ERP-project, is to secure that different phases are realized properly. Classic example could be the training of personnel; it is not enough to train people in a classroom to use an ERP system. The personnel must also know the business processes, function according to the process, and know their part in the process and the role of the ERP system. The biggest single thing that can jeopardize this phase is the scheduling. Schedules that are too tight will cause the commitment and training (or any other step) to be incomplete.

4 Alternatives and improvement possibilities

This section describes alternatives and improvement possibilities based on literature and VTT researchers view. This includes open source based ERP's, utilising ERP's provided as SaaS and customizing open source ERP's using DSML (Domain Specific Modelling Language). DSM based customisation is introduced because customizing would become quicker, easier and more understandable for every stakeholder.

4.1 Open source based ERP's

The open source based ERP's are developed by communities in the Internet, with free access to the source code of the software. According to Serrano & Sarriegi the “Open Source” method of developing software is seemed to suit the development of ERP systems especially well because of three main reasons:[12]

- Adaptability: ERP always need an implementation process to match business processes. Having a full access to the ERP source code can facilitate this customization
- Decreased reliance: Businesses that acquire a proprietary ERP are highly dependent on the product builders and distributors. If one or even both disappear, upgrading and maintaining of the ERP can pose significant problems.
- Reduced cost: Proprietary ERP licenses are expensive. A rule of thumb puts them at between one-sixth and one-third of the implementation project costs.[13]

The business around open source ERP is based on services. Most companies in this field of business offer open source ERP’s as “software-as-a-service” (SaaS).

Pricing is based on per users per month. Communities who develop the open source ERP are usually separated from the companies utilizing the software, but companies usually do advertise the open source software as well.

There are several open source based ERP’s available. Based on the number of downloads, the top 5 of open source ERP’s from Sourceforge are presented in Table 2.

Table 2. Top Sourceforge open source based ERPs downloads

	Open Source based ERP	**Downloads**
1	Compiere ERP + Business Solution	1,496,255
2	OpenBravo ERP	1,013,677
3	OpenTaps open source ERP+CRM	455,812
4	TUTOS	444,636
5	Adempiere ERP Business Suite	417,622

The two clear leaders are Compiere and OpenBravo. However the number of download isn’t the best indicator for the suitability of open source software, other metrics is needed as well. For this project, Compiere and OpenBravo were chosen as candidates.

Other development platform than Sourceforge, is Launchpad. Both of these platforms are popular with open source projects, and from Launchpad an open source ERP called “Open ERP” was chosen as a candidate. Evaluation of Open ERP is discussed in detail in appendix A.

4.2 ERP as a Software-as-a-Service (SaaS)

One new alternative is the possibility to select ERP as a SaaS. Usually SaaS – based applications are run in provider’s server and customers use the applications

[12] Johansson & Sudzina: ERP Systems and open source: an initial review and some implications for SMEs.Center for Applied ICT, Copenhagen Business School, Denmark.(2008)
[13] http://www.cio.com/article/32101/Open_Source_ERP_Gains_Users?page=1

on demand over network connection. The applications used run on servers, which are physically located somewhere else than in the customers premises. This means that SaaS based solution can free the client from hardware acquisitions and server maintenance, system updates and other functions that are needed to support the system. Hence the benefits of SaaS based solution for customer include lower costs and ease of use. As it is the common sales pitch of SaaS based systems, customers can focus only on the use of the system and enjoy the benefits - the SaaS - provider takes care of everything else. Some ERP providers using the SaaS – concept do exist like NetSuite and E2open. In Finland some examples are Manu Online and Severa, see annex B for more information about Severa.

Even though SaaS is becoming more mainstream, it also has some challenges. In a research done by Gartner in 2009 with companies from the U.S and U.K, the top three reasons for deploying a SaaS based solution were: 1) Meeting technical requirements, 2) Security / privacy / confidentiality and 3) Ease of integration and functionality needed for business unit owners. However those who hadn’t chosen to deploy a SaaS based solution said their top three criteria consisted of: 1) High cost of service, 2) Difficulty of integration and 3) Technical requirements. These contradict the traditional promise of SaaS based solutions.

SaaS based solutions have become popular in CRM – systems, as Salesforce.com has gotten more market share. SaaS based solutions were seen as a possibility in the interviews, but it was questioned if large systems such as the whole ERP system for manufacturing company can actually be offered as a SaaS based solution. Solutions currently on SaaS based ERP market cover light functionality and are best suited for companies that only need light ERP modules, for example consulting companies. Light functionality means i.e. project management, customer relationship management, human resource management and finance. For companies which need more robust modules, for example in manufacturing where sub-contractors need to be connected to the ERP system, the SaaS based ERP’s are not yet an option.

Some of the ERP systems provided as SaaS actually use an open source based ERP, which is offered as service to customers (like OpenBravo). This follows the traditional business model of open source software, where business focuses on the services related to open source based software. It can also be viewed so that open source based software has enabled this kind of business activity and therefore generates more competition to the ERP market. Even though open source based solutions are still faraway from the functionality of proprietary ERP systems, it is likely that open source based ERP’s will get more market share in the future, as they become available as SaaS to companies in need of a light ERP.

4.3 Open Source ERP customization with DSM

This chapter briefly discusses about the possibility of utilizing DSM (Domain-Specific Modelling) for customizing open source ERPs. First the motivation of DSM-based customization is discussed. After that, the challenges are clarified. Finally, the subject is concluded by discussing what is required for successful DSM utilization within this problem area.

"Domain-specific modeling (DSM) is a software engineering methodology for designing and developing systems, such as computer software. It involves systematic use of a graphical domain-specific language (DSL) to represent the various facets of a system. DSM languages tend to support higher-level abstractions than General-purpose modeling languages, so they require less effort and fewer low-level details to specify a given system." [8]

Domain-Specific Modeling raises the level of abstraction by specifying the solution directly using domain concepts. Usually the DSM language includes the code generator too, so the final products can be generated directly from the high-level graphical specifications. It is stated that industrial experiences of DSM indicates it being even 5-10 times faster than traditional development practices are. The automation of code generation is possible because both the language and generators need fit the requirements of only one specific domain. [9]

For more explanation about DSM see http://en.wikipedia.org/wiki/Domain-specific_modeling and http://www.dsmforum.org/

4.3.1 Motivation

Utilizing the DSM for open source-based ERP would be a good idea, because customizing would become quicker, easier and more understandable for every stakeholder. If the implementation of DSM is good, it should be possible for almost everyone to customize and configure the open source ERPs' with DSM. Another advantage in DSM utilization is that defective implementations can be forbidden by using the DSM constraints.

Setting the Variability Space is one approach for defining a DSM language. Such cases are typical in product families, where language can be applied for variant design. The variability space is captured in the language concepts, and the modelers' role is to concentrate on the issues which differ between the products or configurations. Languages for pure static variability, like configuration, should be relatively easy to create. [1]

4.3.2 Challenges

The difficulty in DSM language definition lay in behavioural variability and it comes up with a language that supports building almost any kind of new feature based on services of the platform. In those cases, the success of the language creation is dependent on the product expert's knowledge. On the other words, the language developer needs to be real professional with the domain and the product to be successful in language definition. [1]

It is not easy task for DSM language developer to forecast what and how the product will be customized in the future. By nature, the customization is an activity in which the change is usually unique. DSM does not easily suit in that, but conversely. It is neither wise nor even possible to implement the all-embracing DSM language.

It is a known fact that the advantage on using the DSM comes from the repeatability and raising the level of abstraction. When making a decision about utilizing the DSM in the development, at least these issues should be estimated.

4.3.3 Conclusions

Developing a DSM language requires that the domain in question is very well known by the developer in order that it is possible to develop a good and functional DSM language. Moreover, the DSM language itself should be designed in a way that it would be easy to extend with the unexpected change and customization requests.

If the goal is doing just a couple of customizations for the ERP, it is not meaningful to implement and define a DSM language for that. But then, if ERP will be configured and customized and administered in a larger scale in the future, it is good bet to implement a DSM language for customization purposes. However, it should be kept in the mind that developing a DSM language is a quite big effort, so the clear outlook for its exploitation in the future must exists.

To be successful, the DSM language should become wide-ranging enough, in order that general customization requests can be smoothly implemented. Moreover, DSM language should be easy to extend in order that unexpected requests can be implemented without having to use too much effort for that.

5 Open source and SaaS alternatives evaluation

Four Open source and SaaS alternatives were chosen for detailed study. The tools were chosen based on their popularity and potential (based on their descriptions in the internet).

5.1 Overview of selected tools

The following tools were selected for detailed evaluation (detailed results are described in appendices):

- Open ERP (http://www.openerp.com/): The software is claimed to be a complete ERP and CRM system. It has separate client and server components. The software is open source and is released under the GNU General Public License. Open ERP was chosen for trial because it was the most promising one based on the open source ERP comparison.
- Severa (http://www.severa.com/fi/index.aspx) is offered as SaaS (hosted solution) allowing customers to use it through web browser without installation. All customers benefit from servers, security and backup systems. Severa is used through web browser, thus even client-side installation is not needed.
- Pupesoft (http://www.devlab.fi/pupesoft) is a GPL-licenced ERP-system for small- and medium-sized enterprises. It is an open source-based ERP, which is offered by Devlab Oy with comprehensive supporting services.
- Openbravo (http://www.openbravo.com/) is a web-based, open source ERP business solution for small- and medium-sized companies, and it is

released under the Openbravo Public License, based on the Mozilla Public License. Openbravo provides a web-based interface, where the user can view production information, inventory, customer information, order tracking, and workflow information.

5.2 Feature comparison between OpenERP and Severa

In this chapter, OpenERP’s and Severa’s features are compared to the general requirements in a table format. See the Table 3 for detailed information.

Table 3. OpenERP and Severa features compared to general requirements.

	General requirements	OpenERP	Severa
Supplier requirements	Supplier's location	France	Finland
	Supplier's size	Community	32 employees
	Introduction project	Chargeable, price about 2500€, in English	Chargeable, price about 350€, in Finnish
	Technical support	Available from community, also possible to buy trainings	Available from wiki, via mail and phone
Subcontractor requirements	Subcontractor location	For example: Voltage.fi, Turku, Finland	Not needed
	Subcontractor size	Established in 2009, quite small organization	Not needed
	Introduction project	Possible, price depends on services	Chargeable, price about 350€, in Finnish
	Technical support	Chargeable, price +40€/month	Available from wiki, via mail and phone
System requirements	Future Development views	Main developer and community is active	Clear roadmap presented for future releases
	Amount of user organizations	700+ installations/day	6000 active organization
	Documentation	Very extensive, but partly outdated	Very good
	Modularity	Modularity is the issue in OpenERP	Some addon modules can be bought separetaly
	Training	Available against payment, prices 600-2500€	A short introduction is free, more detailed trainings costs 280€ or more
	Main user organizations	All-sized organizations	Small- and Medium-sized organization
	References	Extensive list of references	Quite extensive list of references
	Integrability	Integrations through XML-RPC or certain modules	No open interfaces for integrations
	Multi-Currency and Languages	Widely supported	Supported
	Customizability	Easy to customize with the help of extensive	Not possible

		documentation	
	Reporting	Many possible reporting features	Many possible reporting features
	Time management	Widely supported	Supported
	Implementation language	Python and XML	Not mentioned
	User interfaces	Installable thin client and web UI	Web UI through Flash supported browser
	Hardware reqs	No specific requirements for hardware	No specific requirements for hardware
	Operating system	Server: Windows, Linux or Mac	No installation needed anywhere
	Databases	PostgreSQL	Not mentioned
	Security	Secured XML-RPC connection	Support for SSL connections
	Error recovery	Not mentioned	Guarantee for error detection is 15min, for hardware problems 1h
	Access rights	Widely supported	Supported
	Data imports/exports	CSV data import/export, Microsoft COM mechanism for Excel exports	Exporting reports to Excel is supported
	Interface for other systems	Google (maps, calendar), document management systems, Magento and other eCommerce tools	Google, LinkedIn, Facebook, public enterprise registers
	Interface for email and calendar	Outlook, Thunderbird	Calendar synchronization supported as a supplementary service
	Operational area	Supports many kind of industries and customers	Manufacturing industry not supported, support for expert organizations

5.3 Summary of available alternative tools

Alternative solutions for commercial ERPs seem to be promising. They do not have as much features as commercials ERPs have, but the main features are certainly covered. In fact, it can be said that OpenERP and OpenBravo are extensive enough for almost every small- and medium-sized organization, for which they are developed. It should be noted that e.g. SAP is meant to be used in large organizations too, so these alternative tools are not directly comparable against it.

Severa is fairly compact SaaS tool being not a thoroughbred ERP. If anything, it is more a CRM and Invoicing tool which does not support e.g. warehouse management or manufacturing processes at all. On the other hand, it is a really good product for those purposes it is developed, like engineering- and advertisement offices, and service-oriented expert organizations in general.

Configuring the OpenERP was quite straightforward. A tiny modification was made successfully by utilizing the extensive documentation. The problem was, however, the documentation: it was extensive but a little bit outdated and also somehow garbled. Finding specific information from the documentation was sometimes frustrating. This was also perceived while installing OpenERP in the server. The structure of the OpenERP's architecture was very clear and needed files for modifications were easy to find from file system.

6 Conclusions

The Yrtti project focused on studying the current state of ERP systems and potential alternatives for the current market dominators. The work done consisted of survey of literature (publications and internet sources), interviews of representatives from six different companies using ERP systems, and more detailed evaluation of four open source and SaaS –based ERP system alternatives.

The main conclusions that can be drawn from the work done include that the main challenges in successful ERP project are suitability of the tool to company's processes, resourcing the project well (including time), ensuring the motivation and commitment of the company personnel, and capabilities of the integrator. The currently used ERP systems are technically working well, the main problems are related to the usability of the systems. There are no barriers in principle to take into use the open source or SaaS alternatives, if they fulfil the same technical criteria as the currently used tools. However, there needs to be credible providers and references of use of these in companies. Thus, introducing them in SMEs first seemed to be the most potential way forward.

7 References

[1] Tolvanen J-P, Kelly S. Defining Domain-Specific Modeling Languages to Automate product Derivation: Collected Experiences. SPLC 2005, pp. 198-209, 2005
[2] http://www.logilab.org/card/pylint_tutorial (2009-11-17)
[3] http://www.logilab.org/card/pylint_manual#pylint-output (2009-11-17)
[4] http://script.wareseeker.com/ASP/openbravo-erp.zip/329664c346 (2009-11-24)
[5] Openbravo web pages, http://www.openbravo.com/
[6] Wikipedia: http://en.wikipedia.org/wiki/Openbravo
[7] http://wiki.openbravo.com/wiki/ERP/2.50/Developers_Guide/Openbravo_Main_Development_Concepts#ERP_application_framework (2009-11-26)
[8] http://en.wikipedia.org/wiki/Domain-specific_modeling (2009-12-11)
[9] http://www.dsmforum.org (2009-12-11)

APPENDIX A: Open ERP

Open ERP introduction

Open ERP was chosen for trial because it was the most promising one based on the open source ERP comparison. The point of trial was to see how easy (or difficult) it is to setup an open source ERP system. In addition, we wanted to know what kinds of mechanisms are provided for configuration and customization of the system. The trial also provided hands on experience on usability of the Open ERP.

The following things are going to be inspected in this trial: Open ERP architecture, customization, and quality assurance of customizations. Open ERP architecture is inspected in terms of system structure (i.e. modular structure, database isolation), and by comparison to legacy systems. In terms of customization the interesting things are what are the provided means of customization (also, what are the levels of customization available), what is the suggested approach in doing customizations (who does, how does), what happens to the customizations when the system is updated (i.e. how are the customizations protected), and what is the practice of returning customizations back to community and how is the quality maintained.

Modes of implementation

The Open ERP manual suggests several alternative scenarios for ERP implementation:

a) SaaS (Software as a Service) or on Demand offer which includes the equipment, the hosting, the maintenance and the support on a system configured to your needs in advance. For more information on running Open ERP as SaaS, see http://ondemand.openerp.com.
b) An internal installation, that you manage yourselves or have managed by an IT services company such as an Open ERP partner.
c) Hosting by a server supplier on which Open ERP is installed, which enables you to proceed to add adaptations on your server.

[http://doc.openerp.com/book/8/8_21_Implem/8_21_Implem_deployment.html]

OpenERP architecture

This chapter tries to answer how Open ERP architecture is structured. An Open ERP system is formed from three main components:

- *PostgreSQL database server*, which contains all of the databases, each of which contains all data and most elements of the Open ERP system configuration
- *Open ERP application server*, which contains all of the enterprise logic and ensures that Open ERP runs optimally,
- *Web server*, a separate application called the Open Object client-web, which enables connecting to Open ERP from standard web browsers. It is not needed when connection is made using a GTK client.

[http://openerp.com/books/openerp-book.pdf]
The Open ERP architecture can be seen in the following figure.

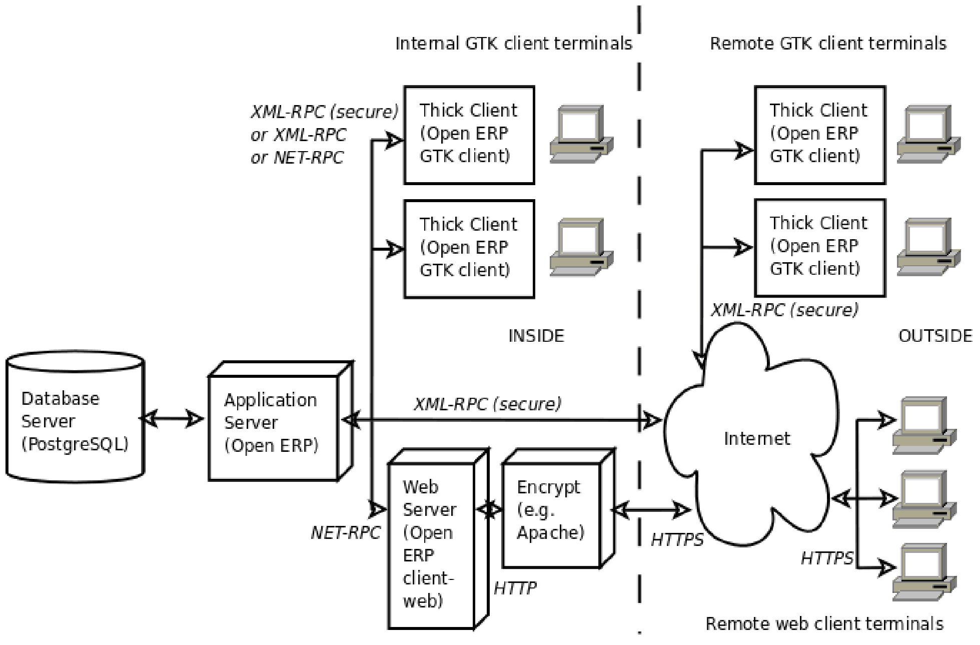

Internal GTK client terminals
Remote GTK client terminals
XML-RPC (secure) or XML-RPC or NET-RPC
Thick Client (Open ERP GTK client)
Thick Client (Open ERP GTK client)
Thick Client (Open ERP GTK client)
Thick Client (Open ERP GTK client)
XML-RPC (secure)
INSIDE
OUTSIDE
Database Server (PostgreSQL)
Application Server (Open ERP)
XML-RPC (secure)
Internet
Web Server (Open ERP client-web)
Encrypt (e.g. Apache)
NET-RPC
HTTP
HTTPS
HTTPS
Remote web client terminals

Figure 5. Open ERP architecture

PostgreSQL is relational and object database management system. It’s a free and open-source high-performance system that compares well with other database management systems such as MySQL and FirebirdSQL (both free), Sybase, DB2 and Microsoft SQL Server (all proprietary). It runs on several types of Operating System, from Unix/Linux to the various releases of Windows, on Mac OS X, Solaris, SunOS and BSD. [http://openerp.com/books/openerp-book.pdf]

Open ERP application and web servers can be run in Windows and Linux environments. Open ERP is built using Python programming language. Python is a dynamic, non-typed language that is object-oriented, procedural and functional. It comes with numerous libraries that provide interfaces to other languages and has the great advantage that it can be learnt in only a few days. For more information on Python, explore http://www.python.org. [http://openerp.com/books/openerp-book.pdf]

Furthermore, the application server has been built on top of Open Object rapid application development (RAD) framework. The main features of the core framework are ORM (based on SQL-Alchemy), a workflow engine, a report designer, a business intelligence interface, an OLAP cube engine. [http://openobject.com/index.php?option=com_content&view=article&id=46&Itemid=53]

Open ERP functionality is built on top of the Open Object framework as modules which results in a cleanly structured application. With this approach each of ERP functionalities has its own module and it’s separated from others. The risk of breaking whole system is much smaller in case of modifying single functionality/module.

By choosing a specific set of modules, the company in question can configure the system to suit their needs. All the official modules are listed and documented in the Open-ERP documentation. Open ERP documentation has detailed description on the module purposes and structure, including version number, dependencies on other modules, and what the module provides: reports, menus, views, objects with fields and their type descriptions (e.g. customer record which contains customer name). [http://doc.openerp.com/technical_guide/document.html#module-document]

Open Object / Open ERP implements MVC architecture. Basically this means that the modules consist of Python code (controller), views are defined in XML and model is defined in database (PostgreSQL) through Object Relational Mapping (ORM). With ORM it is not necessary to manipulate the database directly, but rather through the ORM interface, which encapsulates the database queries and takes care of constraint checking etc. Furthermore, the database tables do not have to be written or modified by hand because the ORM is able to automatically handle them. The whole concept is shown in the following figure.
[http://doc.openerp.com/developer/1_2_module_development/1_server_module.html]

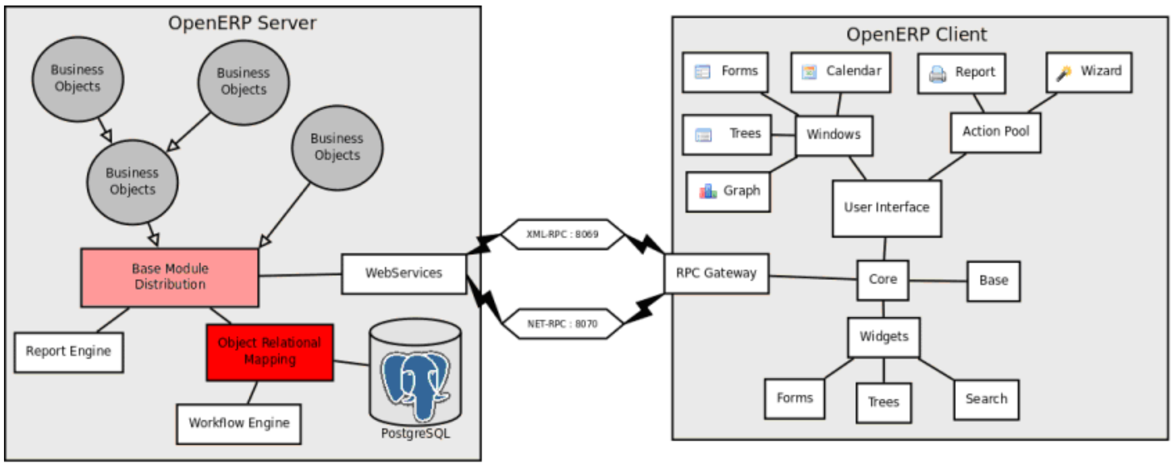


Figure 6. Server-client architecture of Open ERP.

In the above figure, it can be seen that Open ERP is a based on a client/server architecture. The server and the client communicate using the XML-RPC protocol. XML-RPC is a very simple protocol which allows the client to do remote procedure calls. The function called, its arguments, and the result are sent HTTP and encoded using XML. Another protocol called NET-RPC is also supported.

The logic of Open ERP is entirely on the server side. The client is very simple; its work is to ask data from the server and to send them back. With this approach, nearly all developments are made on the server side.

This section explains the meaning of Open ERP server components.

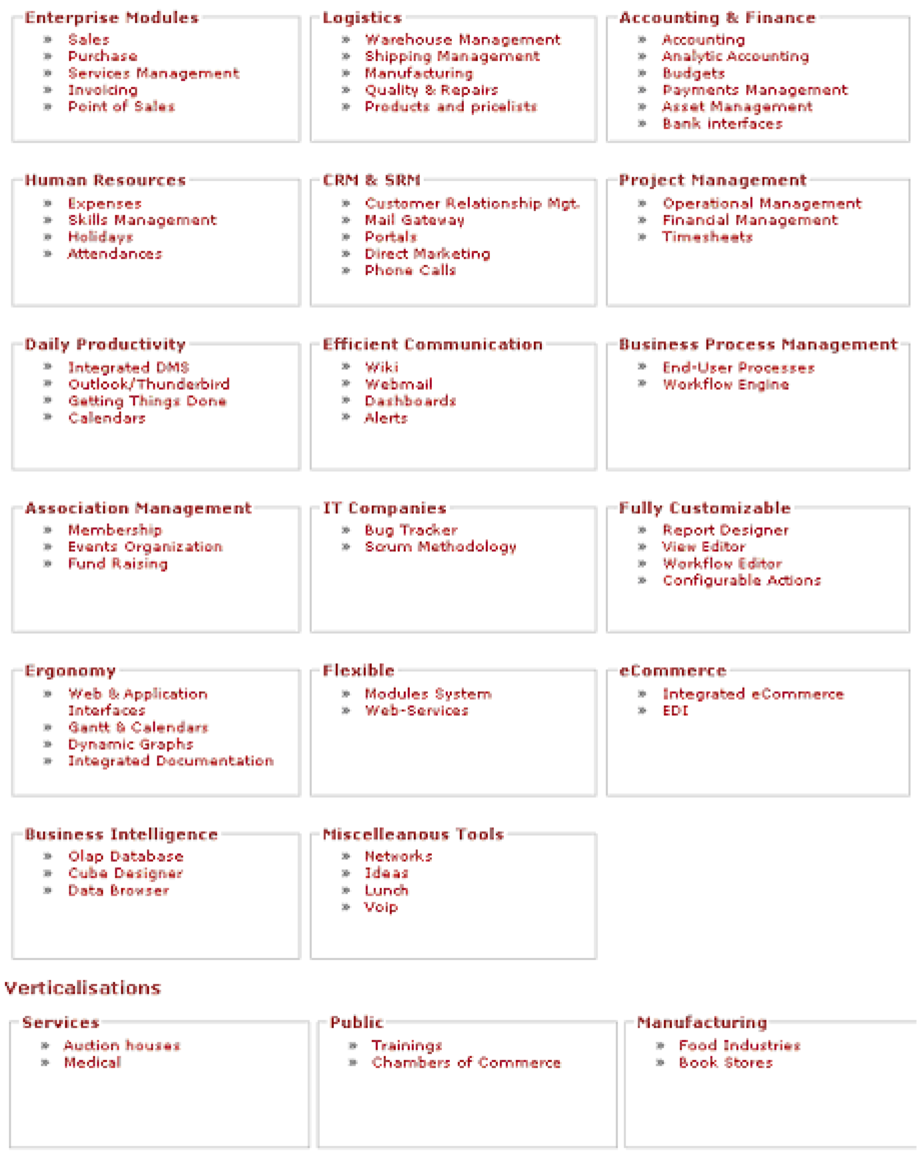
- The **“base module distribution”**-component implements distributed communication mechanism inside the Open ERP server. It supports most common distributed patterns: request/reply, publish/subscribe, monitoring, triggers/callback, etc.
- The “**Web-services”**-component implements communication with the client via supported protocols (e.g. XML-RPC).

- The **"workflow engine"**-component handles workflows in Open ERP, e.g. a sales order generates an invoice and a shipping order.
- The **"report-engine"** component handles the rendering of reports. There are different ways to render reports: custom reports (represented by business objects in the figure) that can be can be directly created via the client interface, personalized reports using openreport, hard coded reports, and OpenOffice Writer templates. The reports are generated in PDF-format.
- **Business objects** describe all data of the program (workflows, invoices, users, customized reports, etc.). They are described by using the ORM module. They are persistent and can have multiple views (described by the user or automatically calculated).

[http://doc.openerp.com/developer/1_2_module_development/1_server_module.html]

Features

Open ERP claims to support a very broad range of features. The following figure lists those features.



Enterprise Modules
- Sales
- Purchase
- Services Management
- Invoicing
- Point of Sales

Logistics
- Warehouse Management
- Shipping Management
- Manufacturing
- Quality & Repairs
- Products and pricelists

Accounting & Finance
- Accounting
- Analytic Accounting
- Budgets
- Payments Management
- Asset Management
- Bank interfaces

Human Resources
- Expenses
- Skills Management
- Holidays
- Attendances

CRM & SRM
- Customer Relationship Mgt.
- Mail Gateway
- Portals
- Direct Marketing
- Phone Calls

Project Management
- Operational Management
- Financial Management
- Timesheets

Daily Productivity
- Integrated DMS
- Outlook/Thunderbird
- Getting Things Done
- Calendars

Efficient Communication
- Wiki
- Webmail
- Dashboards
- Alerts

Business Process Management
- End-User Processes
- Workflow Engine

Association Management
- Membership
- Events Organization
- Fund Raising

IT Companies
- Bug Tracker
- Scrum Methodology

Fully Customizable
- Report Designer
- View Editor
- Workflow Editor
- Configurable Actions

Ergonomy
- Web & Application Interfaces
- Gantt & Calendars
- Dynamic Graphs
- Integrated Documentation

Flexible
- Modules System
- Web-Services

eCommerce
- Integrated eCommerce
- EDI

Business Intelligence
- Olap Database
- Cube Designer
- Data Browser

Miscelleanous Tools
- Networks
- Ideas
- Lunch
- Voip

Verticalisations

Services
- Auction houses
- Medical

Public
- Trainings
- Chambers of Commerce

Manufacturing
- Food Industries
- Book Stores

Figure 7. List of claimed Open ERP features [http://openerp.com/discover/features.html]

Additionally Open ERP provides following integrations to other systems.

- LDAP integration. LDAP enables management of common directories for various different resources through standard TCP/IP network. This enables users in the company to have the same username and password to access all their applications (such as email and intranet).
- XML-RPC interface provides a way of interfacing with the Open ERP server. Using of such standard allows development of integrations in any language as long as XML-RPC is supported. For example, PHP, RoR, Python, Java, etc. could be used.

Setup steps

Open ERP was installed internally on to a PC running Linux (Ubuntu), therefore the chosen implementation approach in this case is b) (i.e. an internal installation). Web UI was used for clients in this installation.

The installation documentation provided adequate guidelines on how to get Open ERP up and running. However, due to different version of Linux distribution used in the installation and in the documentation, some deviations from the default installation procedure were necessary. With the help of user comments in the installation documentation these problems could be ultimately resolved. However, the official documentation should be updated to reflect these guidelines as well.

Open ERP configuration starts with the initializing of the ERP database. The system prompts user on which kind of profile is best suitable for the company in question. The amount of profiles is minimal and the suggested way of doing things is to choose a minimal profile which provides the core set of functionality, followed by manually enabling the needed modules. The database can be populated with demonstration data, which may be helpful during the early phases of installation.

After the initialization, company information can added in to the system. This information can also be modified later. In the next phase the installation wizard will allow creation of new users. The users can also be added later. Finally, it is possible to log in to the system with administrator account or if new users were added during installation they can be used also.

Configuration and data migration

Issues in this chapter address the configuration of the system, which means e.g. how to enable invoice functionality for the system. However, the difference between configuration and customization (according to Open ERP) is first explained.

“*Customization* generally refers to something that requires a bit of technical effort (such as creating specialized code modules) and creates a non-standard system. *Configuration* is less radical – it’s the general process of setting all the parameters of the software to fit the needs of your system (often called *parameterization* or *setup*). Configuration is also, by convention, the name of the sub-menu below each of Open ERP’s top-level menus that is accessible only to the administrative user for that section. *Personalization* is just that subset of configuration options that shapes the system to the particular operational and/or stylistic wishes of a person or company.“ [open_erp]

Users- and groups configuration is done through the UI. Open ERP allows creation of user groups and placement of users in the groups. Fine grained access rights can be defined for users & groups. Partners can be added in to the system and categorized if needed.

New modules can be added by copying modules (modules could originate from an official repository, 3rd party, or own development) to the Open ERP server “addons” directory. The repository scanning mechanism allows for checking of new and updated modules. It is thus possible to add, remove and upgrade modules easily via this mechanism (without manual copying). Open ERP does not guarantee that by installing and removing a new module the database will return to state that it was prior to the installation. Therefore it is suggested that backups are made of the database if a module is modified.

Each module has to be configured when it is enabled. The modules provide a wizard (a kind of installer) which guides the user in the configuration process. The configuration consists of filling out various information and choosing options which depend on the module in question. Configuration settings can be backed up for further usage (such as replicating the settings in another database).

Depending on the modules selected, varying information has to be added in to the system. Due to amount of existing modules, specifics are not addressed. The information can be imported/exported to/from the Open ERP system via CSV files. Open ERP documentation describes in detail the formats and examples for import / export. Other export mechanisms such as Microsoft COM, XML-RPC, and direct database access are also available.

Various languages can be easily enabled from the web UI. Multiple languages can also be used simultaneously, in case a partner needs to work with a different language. However, Finnish is not included in the officially supported set of languages.

Users can each have their own dashboard, adapted to their needs, to enable them to manage their own work effectively. For example a developer using project dashboard can see such information as a list of the next tasks, task completion history and an analysis of the state of progress of the relevant projects. These configuration changes are carried out through the user interface. [open_erp]

Open ERP’s main menu can be entirely reorganized. The management of access rights makes it possible to assign certain functions to specific system users. Roles can be assigned which define the part that each system user plays in the workflows that move system documents from state to state (such as the ability to approve employee expense requests). [open_erp]

Updates and upgrades

According to Open ERP there are four sources of official code change:

- patches supplied by Tiny to correct faults: after validation these patches shouldn’t cause any secondary effects,

- minor updates, which gather the fault corrections together in one package, and are generally announced with a modification of the version number, such as from 5.0.0 to 5.0.1,
- upgrades, which bundle both the fault corrections and the improvements to the functionality in a major release such as from 5.0.3 to 5.2.0.
- new functions generally released in the form of new modules. [open-erp]

It is suggested that offline and online versions of the database are made. Initially the changes caused by upgrades are tested in offline version and migrated to the online version if everything works correctly.

As a partial update specific modules can be updated to newer version(s). Furthermore, upgrade to a newer ERP version is possible by Tiny provided migration scripts. The scripts allow upgrading from official stable release to a new release. Furthermore, migration documentation is provided for each release. Migration between versions can be difficult when the ERP has been (significantly) modified. The migration will then involve a lot of manual work. For example, careful verification should be made that the modules operate correctly after upgrade. The documentation also gives an impression that it is difficult to anticipate what kind of breakages may occur during migration to a newer version if the ERP has been modified.

How to commit own modules or updates

If it is wanted to contribute on OpenERP, the proposed method is as follows:

- Create a branch on launchpad on the OpenERP project.
- Develop your own features or bugfixes in your own branch on launchpad.
- Once code is good enough, propose your branch for merging.
- Work will be evaluated by one responsible of the commiters team.
    - If they accept your branch for integration in the official version, they will submit to the quality team that will review and merge in the official branch.
    - If the commiter team refuses your branch, they will explain why so that you can review your code to better fits guidelines.

Customization

Issues in this chapter address the customization of the system, e.g. how are new modules created or existing ones modified. Guidance on how to build modules is provided in Open ERP documentation.
For example, if a new field (e.g. CEO name) was needed in the company description, there are at least two ways in which the modification can be made. The customization can be done through the web UI or by editing the python and XML files on server. The latter is a better choice, because editing through the web UI is slow.

Let us consider a scenario where a new field is added to the company database table. The modification is not made directly to the database but rather to the object representing the company-abstraction. Secondly, the company form(s) needs to be modified to show the new field in the UI. This can be done by modifying view XML specification(s). The modified modules need to be upgraded afterwards. The database is rebuilt according to information provided in XML files and Python classes. Thus, the new field can be seen in company view for each of the companies stored in the system. If unit testing is needed for modules, such functionality is also available in Open ERP.

Users can make changes to the layout from web interface. These changes are stored per user to database table which contains all the customizations done “on the fly”.

By using the *OpenOffice.org Report Designer* module it is possible change any part of any of the reports produced by the system. The system administrator can configure each report to modify its layout and style, or even the data that’s provided there. No programming is needed when the OpenOffice.org extension is used for creating reports.

Workflows in Open ERP are used to model business processes, e.g. define steps for handling of purchase order and documents produced. Workflows are modelled by using XML files. The language for workflow programming is defined in the developer handbook.

ERP terminology can be substituted to suit the company terminology by using the language translation mechanism. Translations can be made in a CSV file or in the UI.

It should be noted that while installing, updating and removing a new module, Open ERP does not guarantee that the database will return to state that it was prior to the installation. Therefore it is suggested that backups are made of the database if a module is modified.

Customization Experiences

Two simple customizations were implemented to the OpenERP; one selectable datafield’s value were changed and one totally new datafield was added in the Human Resources - module. Customization was implemented by editing the python- and xml-files. It was the easiest part of customization. It was only needed to clarify what (code snippet) to implement and where (folder, code files) it needs to be performed. The more frustrating part was investigating how to enable the update into database according to the modifications. In documentation, it was stated that the database is rebuilt according to information provided in XML files and Python classes. However, it was NOT emphasized in the same chapter, that invoking a database upgrade after the installation of a new version of some module, it is needed to start the server with the **--update=module_name** argument. Finally, the information for this database upgrade command was found from another place of documentation.

The first customization was adding the new value into an employee information form. Initially there was no possibility to select a ‘single parent’ option in marital status-field. This was seen important to be defined in our scenario, so we customized the hr.py python model as follows.

Original:

```
'marital': fields.selection([('maried','Maried'),
('unmaried','Unmaried'), ('divorced','Divorced'),
('other','Other')], 'Marital Status', size=32,
```

Customized:

```
'marital': fields.selection([('maried','Maried'),
('unmaried','Unmaried'), ('divorced','Divorced'),
('single_parent','Single Parent'), ('other','Other')], 'Marital
Status', size=32),
```

The second customization was adding a totally new datafield into the employee information form. That requires updating the xml-file too. The following line was added into hr.py file.

```
'military_rank': fields.selection([([('', ''),
('private','Private'), ('corporal','Corporal'),
('sergeant','Sergeant'), ('lieutenant','Lieutenant')], 'Military
Rank'),
```

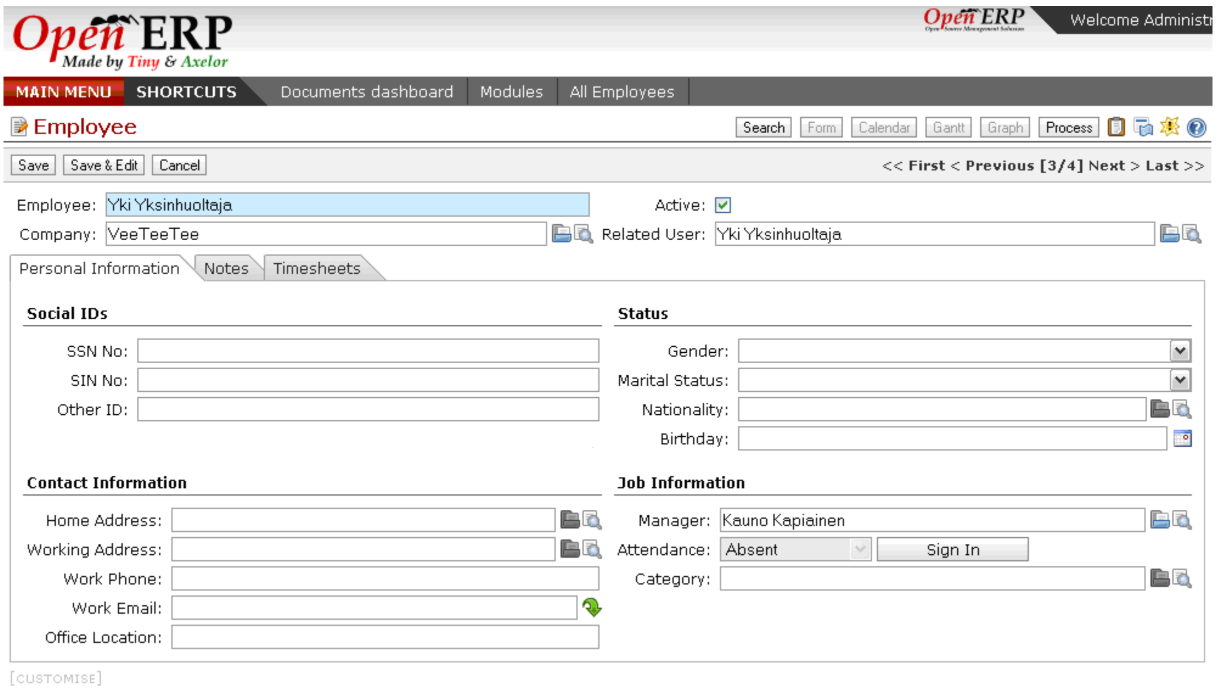
As stated, the xml file must also be modified. The following row was added into hr_view.xml file.

```
<field name="military_rank"/>
```

Finally, when these modifications were made and files saved, OpenERP server startup command with adequate parameters must be ran. During the OpenERP startup routine, a database is updated. In that case, the startup command was as follows.

```
openerp-server.py --update=hr
```

The following figures illustrate the UI layout before and after the customization.



Employee:	Yki Yksinhuoltaja	Active:	✓
Company:	VeeTeeTee	Related User:	Yki Yksinhuoltaja

Social IDs

SSN No:	
SIN No:	
Other ID:	

Status

Gender:	
Marital Status:	
Nationality:	
Birthday:	

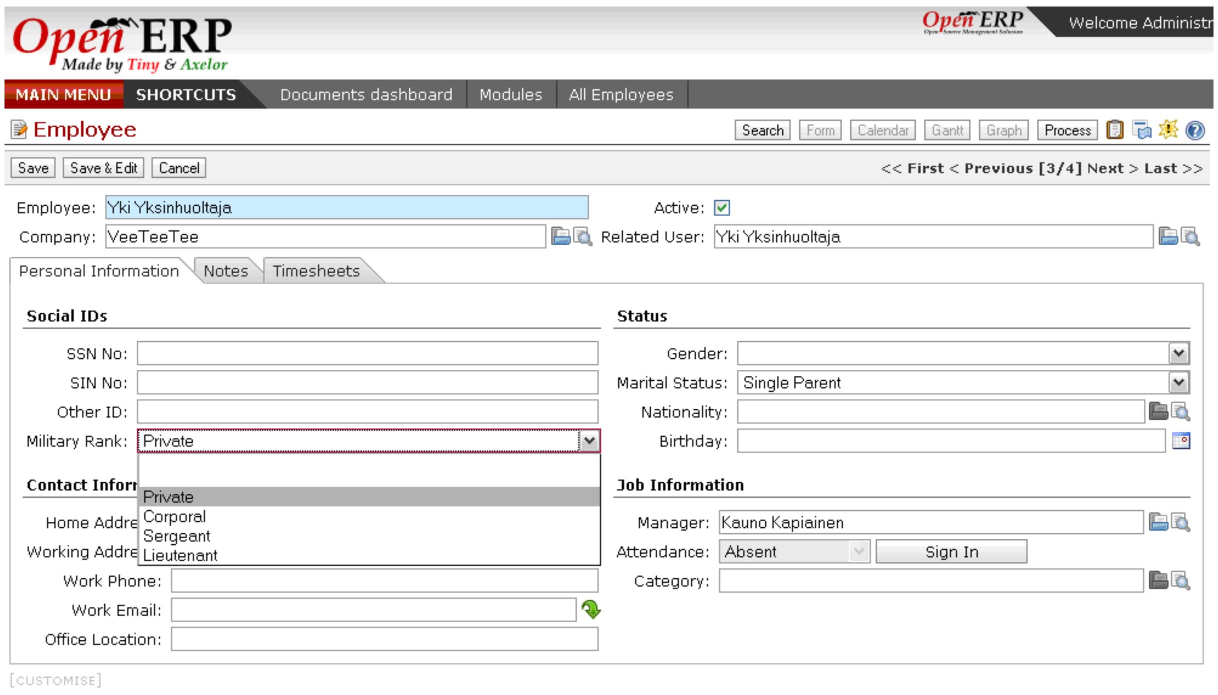
Contact Information

Home Address:	
Working Address:	
Work Phone:	
Work Email:	
Office Location:	

Job Information

Manager:	Kauno Kapiainen
Attendance:	Absent
Category:	

Figure 8. New Employee form before the customization.



Employee:	Yki Yksinhuoltaja	Active:	✓
Company:	VeeTeeTee	Related User:	Yki Yksinhuoltaja

Social IDs

SSN No:	
SIN No:	
Other ID:	
Military Rank:	Private

Private
Corporal
Sergeant
Lieutenant

Status

Gender:	
Marital Status:	Single Parent
Nationality:	
Birthday:	

Contact Infor

Home Addre	
Working Addre	
Work Phone:	
Work Email:	
Office Location:	

Job Information

Manager:	Kauno Kapiainen
Attendance:	Absent
Category:	

Figure 9 New Employee form after the customization.

Source Code Quality Analysis

In this section, the quality of OpenERP server source code is analyzed in a general level. First we take a look to the structure of a file system of OpenERP server. After that, the source code files are analyzed with a couple of examples, and finally, a short summary and a kind statement is presented. In this case, we are only concentrating to the OpenERP server implementation. The architecture of OpenERP is described more detailed in section OpenERP architecture.

File system structure

OpenERP-server – folder includes a folder named *bin* where *openerp-server.py* is lying. That is the main file which is ran in OpenERP application server startup. It imports all the necessary modules and initializes the required services during the startup routine. These modules are e.g. *pooler* connecting to the database, and *addons* including all the installed addons for that specific OpenERP installation. Moreover, during the startup-routinge, the server itself is launched meaning that port, interface, http-daemon and XML-RPC services are launched.

The usage of the modules is the way to extend OpenERP functionality. The default OpenERP installation is organized as a kernel and various modules. The following modules are situated under the bin-folder: *addons, ir, osv, pychart, report, service, tools, wizard* and *workflow*. *Pychart* is a module which takes care about drawing the different kind of graphs and charts and, respectively, *report*-module handles the actions regarding to the imports, reports and reporting interfaces. *Services*-module manages the web services and login procedures to the server. *Tools*-module handles the actions like configurations, convertions, maintenance and pdf-utilizations. In *wizard*-module it is possible to implement addons' installation wizards, and *workflow*-module is for delivering services and logs for workflow-actions.

The *addons*-folder is one of the most interesting folders under the OpenERP application server file system, because it includes almost all the systems functionality under it. If a developer implements the new functionality for the OpenERP, it must be copied in addons-folder as a certain kind of module. The following items must be included in the addon-folder during the module installation: a subdirectory, a module description file: __terp__.py, the Python files containing the objects, and .xml files that download the data (views, menu entries and demo data).

XML files located in the module directory are used to modify the structure of the database. They are used for many purposes like data initialization, and data, views, reports and wizards declarations. The structure of a file system is clear and easily understandable.

Source Code Analysis

As stated earlier, OpenERP consists of both python and XML files. XML is used to model a views and menu entries. With OpenERP, the XML files are used in standard way. There are not so much comments in the code, but the tree structure and naming of elements, like models, fields and variables are very clear and logical, making the code easy to read and understand. In addition, the naming of files is logical and successful. The next code snippet is taken from sale_view.xml – file.

```
<?xml version="1.0" encoding="utf-8"?>
<openerp>
    <data>
        <menuitem icon="terp-sale" id="menu_sale_root" name="Sales
Management"
            groups="group_sale_user"/>
        <record id="view_shop_form" model="ir.ui.view">
            <field name="name">sale.shop</field>
            <field name="model">sale.shop</field>
            <field name="type">form</field>
            <field name="arch" type="xml">
                <form string="Sale Shop">
                    <field name="name" select="1"/>
                    <field name="warehouse_id" required="1" select="1"/>
                    <separator colspan="4" string="Accounting"/>
                    <field name="payment_default_id"/>
                    <field domain="[('type','=','sale')]"
name="pricelist_id" select="1"/>
                    <field name="project_id" select="1"/>
                    <separator colspan="4" string="Payment accounts"/>
                    <field colspan="4" name="payment_account_id"
nolabel="1"/>
                </form>
            </field>
        </record>
```

What comes to the python code, the same trend continues. Commentary is quite scarce, but the code is still quite easy to read and clearly formulated. Source code layout gives a positive image about the OpenERP developers. Another code snipped below is python-code and it is taken from sale.py – file.

```
    def action_invoice_cancel(self, cr, uid, ids, context={}):
        for sale in self.browse(cr, uid, ids):
            for line in sale.order_line:
                invoiced = False
                for iline in line.invoice_lines:
                    if iline.invoice_id and iline.invoice_id.state ==
'cancel':
                        continue
                    else:
                        invoiced = True
                self.pool.get('sale.order.line').write(cr, uid, [line.id],
{'invoiced': invoiced})
        self.write(cr, uid, ids, {'state': 'invoice_except', 'invoice_ids':
False})
        return True
```

In order that code analysis is not based on the subjective evaluation only, a source code analysis tool is used for formal evaluation. We selected a tool called Pylint, which is meant to be used to check for code consistency, following of standards, naming conventions, error prone sections of code, and code that suggested poor reliability. Pylint is a tool that checks for errors in python code, tries to enforce a coding standard and looks for smelling code.

The Python community has formalized some recommended programming styles to help everyone write code in a common, agreed-upon style that makes the most sense for shared code. This style is captured in PEP-8. Pylint can be a quick and easy way of seeing if source code has captured the essence of PEP-8. Pylint will display a number of messages as it analyzes the code, as well as some statistics about the number of warnings and errors found in python files. The messages are classified under various categories such as errors and warnings. Finally, the code is given an overall mark, based on the number and severity of the warnings and errors. **Error! Reference source not found.**, **Error! Reference source not found.**

The following python files from openerp-server/bin/addons/ - folder were analysed by Pylint: *account.py*, *crm.py, hr.py*, *product.py* and *sale.py*. Moreover, following main modules' python-files are also analysed*: openerp-server-py, pooler.py*, *secure.py*, *web_services.py* and *osv.py*.

PyLint message types are following:
- [C]onvention for coding standard violation
- [R]efactor for a "good practice" metric violation
- [W]arning for stylistic problems, or minor programming issues
- [E]rror for important programming issues (i.e. most probably bug)
- [F]atal for errors which prevented further processing

Global evaluation rate maximum value is 10 points. Other values in the **Error! Reference source not found.** are the numbers of message types.

Table 4. Pylint python source code analysis.

File name	Statements	Convention	Refactor	Warning	Error	Global rate
account.py	1212	854	66	182	144	-5,03
crm.py	455	281	29	70	61	-5,05
hr.py	96	63	11	21	0	0,1
product.py	343	222	15	63	45	-4,87
sale.py	609	326	35	100	82	-4,3
openerp-server.py	91	32	0	8	0	5,6
pooler.py	39	14	0	2	0	5,9
security.py	45	21	0	3	0	4,67
web_services.py	548	191	34	63	39	1,19
osv.py	184	64	6	16	9	2,88
AVERAGE	**362,2**	**206,8**	**19,6**	**52,8**	**38**	**0,109**

Statement of code analysis

What Pylint analysis says is not meant to be taken as gospel, because it tries to detect things that may be dangerous in a context, but maybe not in others. Or because it checks for some things that developer doesn’t care about. Generally, developers shouldn't expect Pylint to be totally quiet about his code, so he doesn't necessarily be alarmed if it gives him a lot of error messages.

If we look at the **Error! Reference source not found.**, it can be immediately seen that there are not any fatal errors in the python code. That is the most important thing for a product. However, the level of Error messages is surprisingly high in OpenERP. The average number of all statements compared to the average number of statements causing an Error messages says that even 10% of all statements have a kind of error. Furthermore, 15% of statements cause the Warning message, 5% of statements cause the Refactor message and 57% of statement creates the Convention message.

It should be noted, that OpenERP have a hundreds of python code files, and hundreds of XML files too. Because Pylint can not analyse the XML files, the importance of Pylint-results in this source code analysis is not very high. If anything, its results are more suggestive than the absolute truth about the quality of code. Under the circumstances, it seems that the quality of OpenERP source code is in a pretty good level, but it could be even better if the formal errors could be repaired and the commentary would be richer.

Conclusion

The installation of Open ERP provided hands on experience on what it requires to get an open source ERP up and running. The results show that installation is easy and quite straight forward. Also the initial expression on usability & look and feel of the system is professional. However, due to limited scope of the trial, several points need to be addressed in more detail: how easy is it to integrate Open ERP to company systems and how much customization is needed before the system can be used in daily operations.

We got some (indirect) answers to these questions based on our study, Open ERP seems to provide rather good mechanisms and interfaces for integration and customization, but these mechanisms were not tried out in practice very deeply. Only a quite lightweighted customization trial was made. After brief examination of the source code, the coding style in Open ERP is well intended (as a result from using Python), clear, understandable. Comments in the source code seem sparse. Module descriptions provide some additional information, but the amount of information seems to vary greatly. Overall, the documentation for installation and development is good for an open source project.

As ERP systems are used by even hundreds of personnel simultaneously, the scalability of the system is an important issue. Scalability and performance of the system was not measured. However the design of Open ERP makes it possible to increase performance by adding more machines on the problem; distributed database (PostgresSQL feature), load balancing of the web frontend (multiple running web frontends & apache load balancing). Open ERP server backend may act as a bottleneck because it is not clear if the server software can be balanced between multiple machines.

APPENDIX B. Severa

Introduction

Operations model
Severa is offered as SaaS (hosted solution) allowing customers to use it through web browser without installation. All customers benefit from servers, security and backup systems. Severa is used through web browser, thus even client-side installation is not needed.

Severa provides user manual and support forum for all users. For personal service Severa provides consulting and training services that can be ordered online.
[http://www.severa.com/uk/tech_specs.aspx]

User Organizations

The biggest group of users are advertisement- engineering- and architecture offices as well as management consults and IT-service companies. Severa seem to suiting best for small- to medium-sized businesses. For larger companies, the licence fees would rise too high.

Amount of simultaneous users
Not mentioned exactly in documents, but if looked at the technology in the background, it can be said that the amount of users could be massive. Severa is being marketed as supporting about 130,000 ongoing projects with over 6,000 customers.

Pricing
All features are provided at no cost for one user. Additional users cost 25€ per user per month including 24/7 unlimited use, support, free maintenance and security. The more users per organization, the lower payment per user. See the table below for more detailed information about pricing.

Table 5. Severa pricing

Users	**Monthly pricing**	**Additional users**
1 user	FREE	25 €
10 users	220 €	18 €
20 users	400 €	12 €
50 users	750 €	10 €
100 users	1250 €	10 €
500+ users	5250 €	10 €

[http://www.severa.com/uk/pricing.aspx]
User Introduction
New users could be introduced with Severa by arranging a demo event through internet-based netmeeting. During the demo event, the most common features are discussed and introduced. There is also space for open questions. In addition, it is possible to buy detailed training sessions from Severa, like Admin training, New user training, Intermediate training and Consulting workshop. Each training costs about 280€ - 490€.

Technology in the background

Performance. Severa is build on 20 Intel Xeon processors and with 9 independent network providers delivering about 6.3 Gbps bandwidth. Severa utilizes Cisco Powered Network with100% Network Uptime Guarantee. They promise uninterruptible power with UPS and 24/7 server management quarantee. [http://www.severa.com/uk/download/pdf/security.pdf]

Hosting partner. Severa uses Rackspace’s UK data centre (World's largest hosting provider by revenue in 2007) that provides certain Service Level Agreements. Thus Severa’s services are protected against data loss, malicious code and hardware failures.

[http://www.severa.com/uk/download/pdf/security.pdf]

System requirements. Computer with network access and internet browser is required for Severa users. Mozilla Firefox is recommended for web browser, but Severa also supports other newest web browsers with Flash and any operating systems (Windows, Linux and Mac).

Security

Physical security. The server side Data center is certified with 24/7 onsite security personnel and biometric scanners. Data center is engineered with connectivity, power, heating and ventilating systems to avoid failures. Severa utilizes Automated Security Patching, Cisco firewalls, Anti-Virus Software, vulnerability scanning and DDoS prevention program.

Network security. 100% Cisco Powered Network is audited by Cisco, which should ensure a good security protection. The network also incorporates a patented Intrusion Detection System (IDS) and transmissions are SSL encrypted.

Data backups are taken daily basis to secure locations. The daily backups are stored for 30 days and monthly backups for 12 months. User can also create own data backups directly from web user interface in Excel format.
[http://www.severa.com/uk/download/pdf/security.pdf]

Features

Customization
Pages and drop-down menus are partially customizable so it is possible to remove those you don't need or add more options. Severa ensures that its forms display key information and limit the number of required fields.
[http://webworkerdaily.com/2009/05/13/severa-easy-one-click-crm-project-management-and-invoicing/]

Based on hands-on experience, the customization possibilities are confined to the user interface customization. In addition, it is quite limited set of view which can be customized. Thus, it can be said that the right word in that case may by personalization.

Integration possibilities
Severa manages customer process from sales, to project management, time tracking and billing. Typically users want to export work hours for payroll and invoices for accounting.
System is integrated to Google Maps, Google Search, LinkedIn and Facebook. With add-ons, user can also import contacts to customer register. There would also be add-ons for more automated integrations. [http://www.severa.com/uk/tech_specs.aspx]

More explanations about the integrations can't be found from the specifications. However, data import and export functions are provided, so it is possible to transfer data between existing system and Severa by Excel documents. Utilizing these beforementioned add-ons can not be performed, so, at the moment it is not clear how they work and how valuable they can be.

Documentation
Quite extensive documentation is available for users on the web. The whole functionality of the tool seems to be covered by documentation. Severa's help page contains FAQ, wiki-based help concerning to the functionality, and a couple of video presentations. In addition, a glossary with terminology is provided on the web site.

During the use, a tiny quick help window is available all the time. It is handy to see some help while doing some new action with the tool.

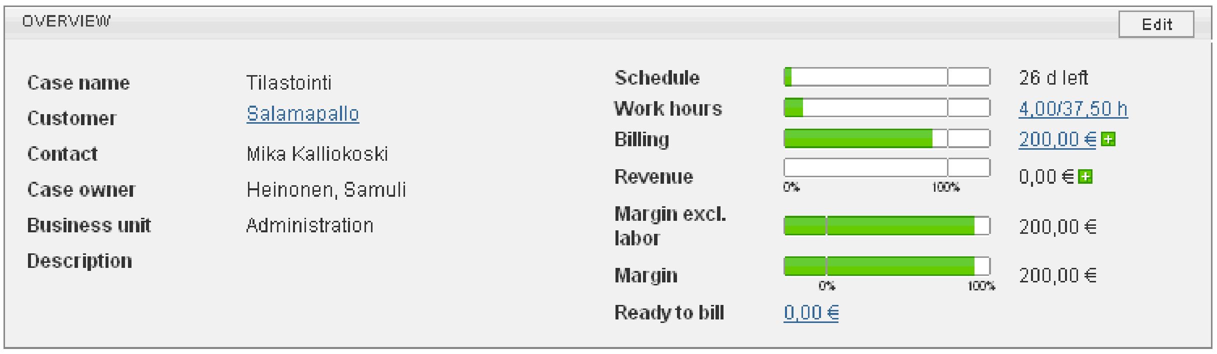
Access rights
User management is an essential part of Severa. Only authorized users are allowed to access the system and user privileges are controlled with access right profiles. The system comes with industry-specific access right profiles for specific roles. These access right profiles can be modified freely. All actions related to user management require sufficient access rights.

Access right profiles define the access rights to the following sections of the system: accounts, cases, users and administrators. User cannot edit the sections or their access right levels, but editing the profiles to which they are linked is possible.
[http://support.severa.com/wiki/doku.php/user_management]

Time tracking
Severa is a work time reporting tool that allows reporting work hours to both sales and internal cases. All entries are tied to specific cases and work or travel types. This feature makes it possible to monitor case costs accurately. [http://support.severa.com/wiki/doku.php/work_time]

Work time entry seems to be quite straightforward and easy to use in Severa. User just selects the case and task from the drop-down menu, and adds some hours to it. There is also a field for some explanation for the task in question. Days that no longer need users' attention, can be saved and locked.

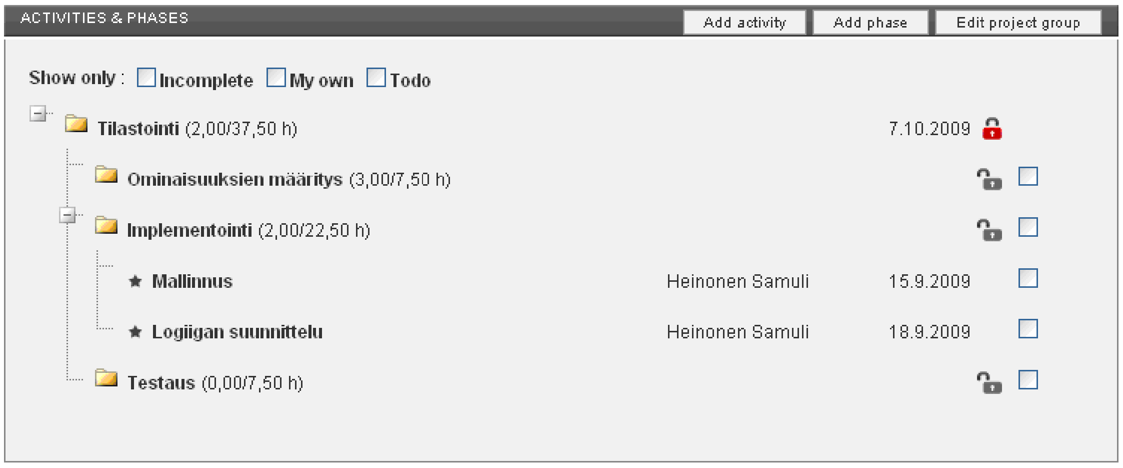
Project management
Project management with Severa is a bit lightweighted. It is more concentrating to the billing issues. However, different kind of activities and phases can be planned and timetables for them can be planned, too. For example, this kind of overview is available for every case (project).


OVERVIEW

Case name	Tilastointi	Schedule		26 d left
Customer	Salamapallo	Work hours		4,00/37,50 h
Contact	Mika Kalliokoski	Billing		200,00 €
Case owner	Heinonen, Samuli	Revenue		0,00 €
Business unit	Administration	Margin excl. labor		200,00 €
Description		Margin		200,00 €
		Ready to bill	0,00 €	

Figure 10. Project overview.

Another, a bit detailed viewpoint for the tasks can be seen here:


ACTIVITIES & PHASES

Show only : Incomplete My own Todo

Tilastointi (2,00/37,50 h)		7.10.2009	
Ominaisuuksien määritys (3,00/7,50 h)			
Implementointi (2,00/22,50 h)			
Mallinnus	Heinonen Samuli	15.9.2009	
Logiigan suunnittelu	Heinonen Samuli	18.9.2009	
Testaus (0,00/7,50 h)			

Figure 11. Project view in tasks level

Realization

For example, last month's invoices, open invoices and this month invoices can be seen from Invoices-menu. Realized working hours according to the cases and employees can be listed, too.

Invoicing

This is one of the strongest features in the Severa. Creating and generating invoices from the spent efforts is really easy. Invoices can be generated based on the completed working hours or just by creating an invoice which includes freely specified fixed fees. The combination is also possible. Editing the details of invoice before sending it is simple and easy to do.

CRM (Customer Relationship Management)

Based on the hands-on tests, it can be said that Severa has a basic set of features on the area of CRM. It might be enough for most of users, but not for those who like to use beforementioned features. However, it seemed that Severa is missing some features like contract management, asset management, document sharing, knowledgebase and customer web portal

[One user comment in http://webworkerdaily.com/2009/05/13/severa-easy-one-click-crm-project-management-and-invoicing/]

User interface

Based on the initial tests, it is easy to get familiar with Severa. All actions are available quickly and they are designed well. If Severa is compared to the OpenERP, it feels that Severa's UI is slightly more intuitive. They both have a web-based user interface, so the comparison is quite easy to perform. On the other hand, OpenERP has clearly more functionality and customization possibilities than Severa, making OpenERP a bit complex and tangled.

User interface Functionality

This chapter presents the features of Severa. Functionalities are divided in the following groups, and are accessible from the tool's menu bar:

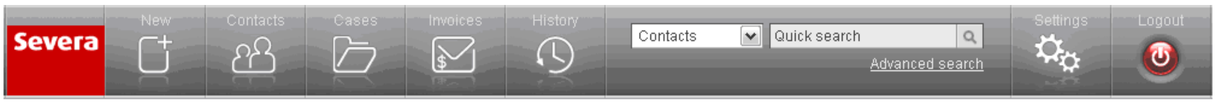
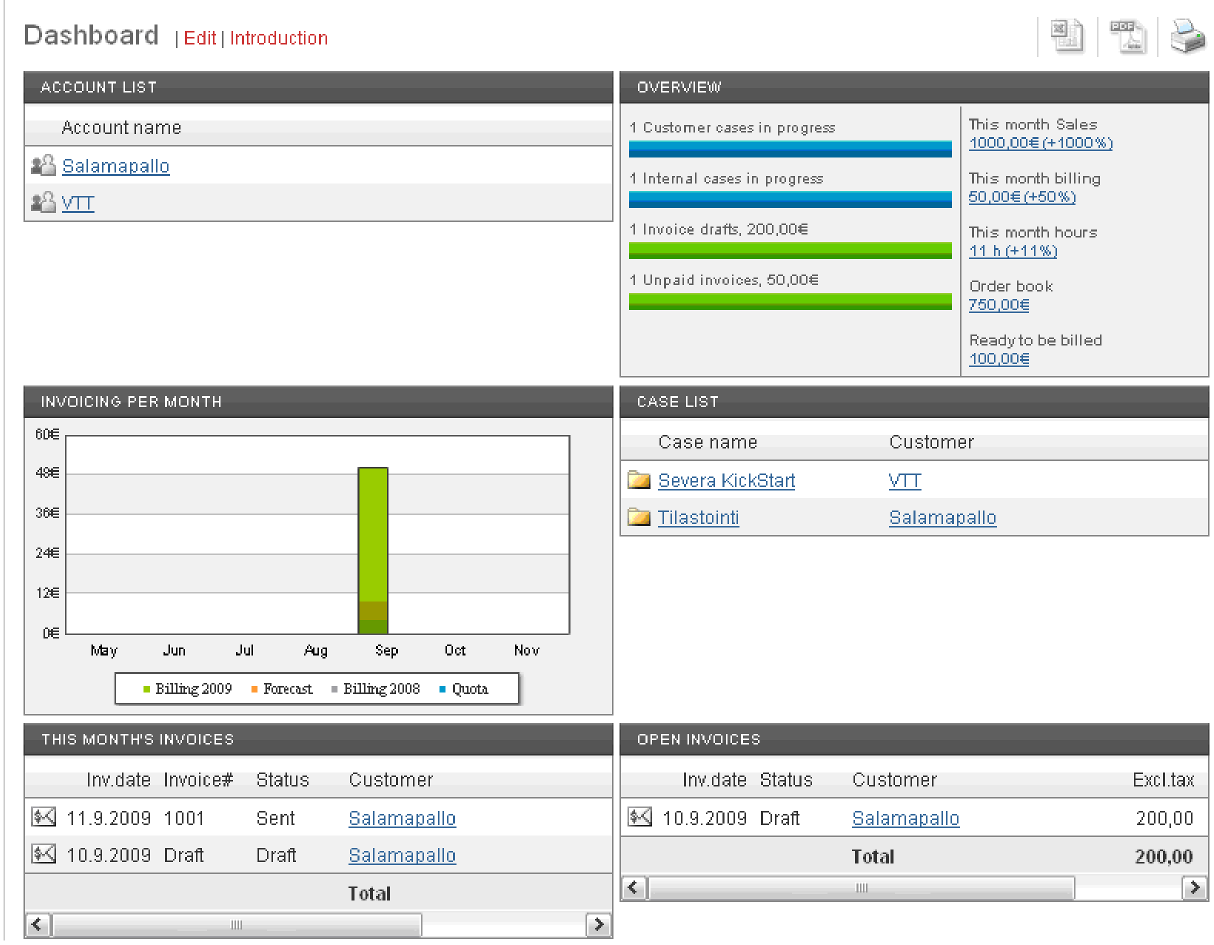


Figure 12. Tool bar menu.

Dashboard

Dashboard pulls together the data from all projects and accounts. It is largely customizable, so user can decide what (s)he wants to be shown in it.



Dashboard | Edit | Introduction

ACCOUNT LIST

Account name
Salamapallo
VTT

OVERVIEW

1 Customer cases in progress
1 Internal cases in progress
1 Invoice drafts, 200,00€
1 Unpaid invoices, 50,00€

This month Sales
1000,00€ (+1000 %)
This month billing
50,00€ (+50 %)
This month hours
11 h (+11 %)
Order book
750,00€
Ready to be billed
100,00€

INVOICING PER MONTH

CASE LIST

Case name	Customer
Severa KickStart	VTT
Tilastointi	Salamapallo

THIS MONTH'S INVOICES

Inv.date	Invoice#	Status	Customer
11.9.2009	1001	Sent	Salamapallo
10.9.2009	Draft	Draft	Salamapallo
			Total

OPEN INVOICES

Inv.date	Status	Customer	Excl.tax
10.9.2009	Draft	Salamapallo	200,00
		Total	**200,00**

Figure 13. Dashboard.

Typical data in the Dashboard could be:

- Account list window showing all customers
- Overview window showing a progress bars, invoices, sales and billing info

- Invoicing per month in graphical format
- Case list with case names and customers
- This month invoices, which presents invoices in detail during the current month
- Open invoices showing the upcoming revenues

New – menu button is used to create a new expression from one of the following things: *Account, Case, Work time entry, Invoice, User* or *Activity*. In a nutshell, these things tell a lot from the Severa’s limitations. It is a tool exactly for small- or medium-sized organizations.

From **Contacts** – menu user can list *Accounts, Most profitable customers, Contacts, Neglected contacts* or *Modify list*. Selecting *Modify list*-action makes it possible to manage these saved searches list, which are straightly selectable from the **Contacts**-menu.

By clicking a **Cases -** menu button user can select one of the following actions: *Case list, Case progress report, most profitable cases, Overdue cases, Uninvoiced cases, Work time per case, Work time entries* and *Modify list*. Again, Selecting Modify list-action makes it possible to manage the list of saved searches, which are straightly selectable from the **Cases**-menu.

An **Invoices** menu includes *Last month’s invoices, Open invoices, this month’s invoices* and *Modify list* menu items.

History menu includes a links to the historical data. It provides a quick way to navigate in the system to users most recently viewed items. There exist direct links to the *Cases*, *Accounts*, *Invoicies* and *Activities*. By clicking a certain link brings user to the page which shows e.g. *a case overview* window, *activity and phases* - window and *account or contact information* - window.

In **Settings** tab user can configure information regarding to the *Users, organization, Accounts and contacts, Case settings, Work time & travel* and *Invoice*. For each of then there are a set of parameters which can be modified, like Personal settings, Business units, Contact roles, Activity types, Work performances and Tax per cent. From **Settings** tab, user can create new users, modify pricelists, worktypes and things like that.

Calendar is not a menu item like the other, but it is a kind of widget on the right side of a screen. It is available all the time in spite of what action is currently ongoing. From the calendar, user can rapidly see the most important deadlines and Todo-appointments at a glance. By clicking the mark on the calendar, a more detailed view is opened in the screen.

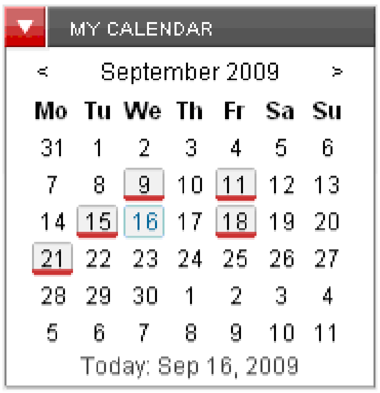
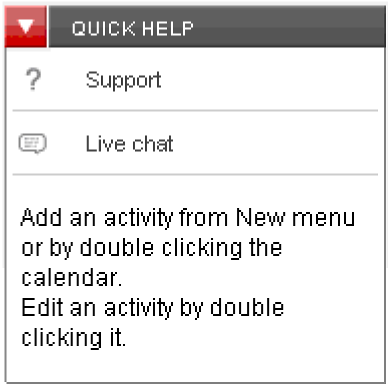


Figure 14. Calendar.

Quick Help is another widget which is available all the time. It provides a short help text regarding to the current action. For example, when user moves the cursor on the button, the help window informs user how to use the current action.



Add an activity from New menu or by double clicking the calendar.
Edit an activity by double clicking it.

Figure 15. Quick help.

Conclusions

Severa is offered as a SaaS allowing customers to use it through web browser without any installation. Price is composed of licence fees, every user must pay a licence fee by monthly basis. The more users the lower price per user.

The most important features are Sales management, Project management, Time tracking, Billing and Reporting. Severa is not too large, it includes the most important functionality for small- to medium-sized businesses. The biggest group of users are advertisement- engineering- and architecture offices as well as management consults and IT-service companies.

The user interface in Severa is pretty clear and intuitive. It is very easy to get in touch with it. User can immediately use the basic set of features naturally, which is a clear advantage against the larger ERP systems. On the other hand, Severa's functionality is quite constricted at the moment, meaning that it does not necessarily suit well for large organizations.

Customizing and tailoring is not possible, only personalization is allowed. Severa is not delighted to implement customer-specific features, but it is possible to propose a certain features for their roadmap.

All updates are immediately on hands for every customer. Incoming updates can be seen from the roadmap, which is openly available for customers in the Severa's wiki-pages. Customer support is mainly offered through wiki-pages and Support Community forum.

APPENDIX C. Pupesoft

Introduction

Operations model
Pupesoft is a GPL-licenced ERP-system for small- and medium-sized enterprises. It is an open source-based ERP, which is offered by Devlab Oy with comprehensive supporting services. Devlap Oy has developed open source products from year 2001.
[http://www.devlab.fi/etusivu]

User Organizations
Small- and medium-sized organizations.

Pricing
Pupesoft is based on the open source GPL licence, so the software itself is free. However, the turnkey installation with desired customizations costs about 2000-5000€ plus required server hardware. In addition, Devlab offers user support for 400€/month/10user. [Telephone conversation on 2009-11-05, http://www.devlab.fi/toiminnanohjaus]

User Introduction
Devlab offers users introduction training against the payment. User instruction can be found from web.
http://www.devlab.fi/wiki/index.php/K%C3%A4ytt%C3%B6ohjeet

Technology in the background

Pupesoft runs on Fedora Core Linux distribution in the organizations' server. The software itself is implemented by PHP and it uses MySQL database as data storage. Pupesoft is used through internet browser over HTTP protocol.

Features

Pupesoft is an invoicing application including an extending set of billing features. That seemed to be the all it offers; there is no working hour's entries, no project management functionalities, no one thing and another. Anyway, a stockpile management is supported. It can be said that Pupesoft is quite lightweighted 'ERP' at the moment, although it is under construction all the time. The list of Pupesoft features is here:
General features

- Support for multiple organizations
- Support for group of companies
- Multi-currency support
- Marginal taxation
- Multilanguage documentation
- Multilanguage user interfaces
- User profiles
- Serial number follow-up

Purchasing

- Importing purchase orders from files
- Support for electrical purchase orders (supplier-specific)
- Registering the product into stockpile database automatically from the web invoice
- Effective and easy-to-use manual stockpile management
- Electrical customs reporting and management

Purchasing reskontra

- Web invoices
- Bar codes
- Support for invoice reuse
- Default posting from enrerprizes and suppliers
- Travel invoices

Stockpile management

- Support for multiple stockpiles
- Priorizititing the stockpile printing
- Automatic expenditures according to weight and price of specific order
- Support for different kind of consignment notes
- Combining the consignment notes of one customer
- Carried forward from stockpile to another

CRM

- Contact and contact person management

Reporting

- Dynamic reporting
- Support for generating SQL and ODBC reports

Sales management

- Special offers
- Support for importing offers from file
- Support for electrical offers
- Real-time saldo inquiries from suppliers
- Past deliveries
- Product families
- Substitute products
- Cash desk selling
- Work orders
- Automatic posting and reference management
- Manual adjustments

- Part payment

Manufacturing

- Manufacturing to the stockpile or directly to the customers
- Product recipes

Accounting

- Real-time accounting
- Automatic registration of all business transactions
- Budgets
- Dimensions, cost pools, subjects and projects
- Change logs

Intranet

- Calendars
- Inhouse news

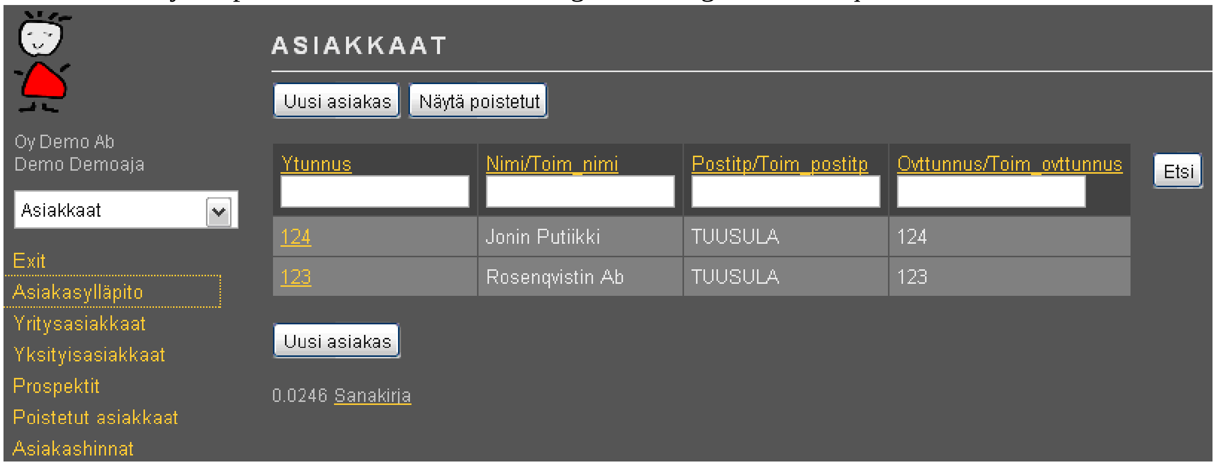
Extranet

- User management
- Ordering from file
- Support for browsing an orders and deliveries
- Price list updates
- Extranet news

[http://www.devlab.fi/toiminnanohjaus]

User interface

According to the hands on tests, Pupesoft seem to be unfinished product. It is a lightweight form-based tool with a very simple web interface. The feeling with using the tool is quite coarse.



Ytunnus	Nimi/Toim_nimi	Postitp/Toim_postitp	Ovttunnus/Toim_ovttunnus
124	Jonin Putiikki	TUUSULA	124
123	Rosenqvistin Ab	TUUSULA	123

Figure 16. Screenshot from Pupesoft general view.

Conclusions

Pupesoft is an open source server-based ERP-tool for small- and medium-sized businesses. It is concentrating to the economic and financial issues like sales management and invoicing and it does not support project management functionalities at all. For example, inserting working hours is not possible.

Pupesoft need to be installed in organizations web server. Devlap recommends using Fedora Core Linux distribution for Pupesoft server. It is offered as a kind of SaaS; they promote the turnkey delivery against the payment, which is some 2000-5000 euros per installation, including parametrization, customization and some kind of user instructions. In addition, they offer a user support for 10 users with 400€/month. The payment is higher for more users.

The user interface is quite rough-grained. User interface is based on the same form-based layout, which causing a feeling that Pupesoft is still incomplete. User instructions, installation instructions and FAQs are available on www.devlab.fi/wiki - pages.

APPENDIX D. OpenBravo

Introduction

Operations model
Openbravo is a web-based, open source ERP business solution for small- and medium-sized companies, and it is released under the Openbravo Public License, based on the Mozilla Public License. Openbravo provides a web-based interface, where the user can view production information, inventory, customer information, order tracking, and workflow information. **Error! Reference source not found.**

Openbravo has an extensive set of certified parters who can implement and customize Openbravo products in companies of different sizes and across a wide variety of sectors and activities over the world. Openbravo is meant to be installed in organizations intranet server, in which users take a connection through a web browser. **Error! Reference source not found.**

Pricing
OpenBravo basic version is available at free of charge, but it is also delivered as a chargeable, more extensive version. Moreover, as stated earlier, Openbravo has a set of certified parners, which are able to install, customize and provide a support and maintenance for Openbravo ERP user organizations. The price depends on what services are bought from the supplier. Annual subscriptions are 400€-750€ per concurrent user based on volume and depending what version about the system is choosed.

Technology in the background

Openbravo's architecture focuses on two development frameworks:

- MVC is a web applications development framework, which helps to decouple the database, user interface elements, and business logic. The separation of these elements into different files results in a more structured code, facilitating development and maintenance.
- MDD is a software design approach that relies on metadata stored in a dictionary to model the behavior of the application. This results in a reduction in manual coding and fewer bugs, allowing business experts with little coding experience to configure the application to suit the needs of each enterprise.

These two models allow for integration with other programs and for a simple interface. Openbravo ERP uses modern but proven technologies to meet the strict performance and scalability requirements of enterprise grade environments:

- Java and Javascript
- SQL and PL/SQL
- XML and XHTML

Key architecture components of Openbravo can be seen in the following figure (**Error! Reference source not found.**):

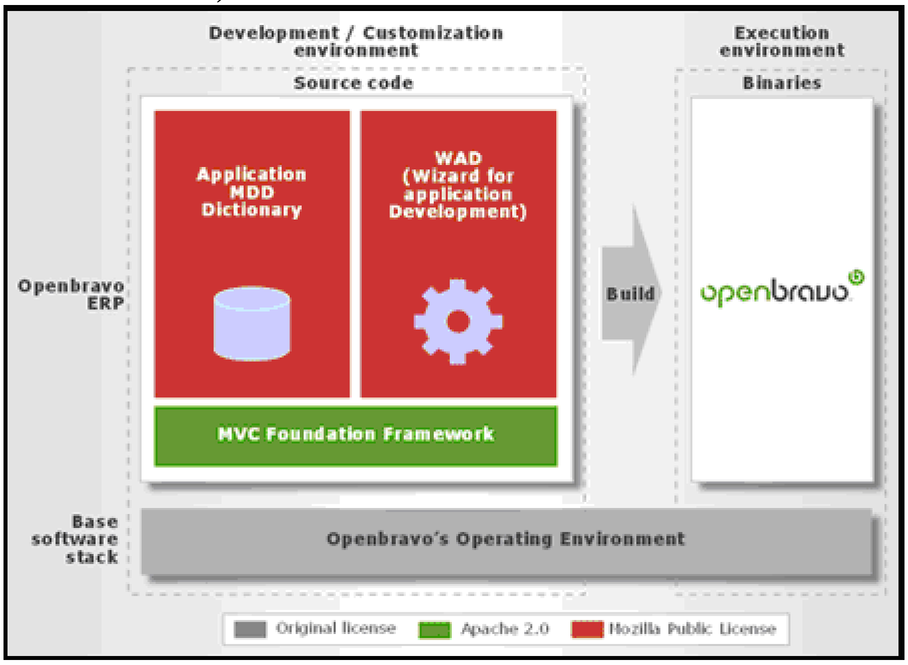


Figure 17. Architecture components of Openbravo

WAD (Wizard for Application Development). The engine, built by Openbravo, automatically generates the application binaries from the MDD dictionary. The files generated by WAD are compliant with the MVC standard. **Error! Reference source not found.**

Application MDD Dictionary Stores the metadata which describes each element of the application and its behavior. **Error! Reference source not found.**

MVC Foundation Framework is a set of programming utilities, either selected from the open source candidates available or built by Openbravo when no candidates are available. These utilities facilitate web-based model-view-controller (MVC) application development. **Error! Reference source not found.**

Operating Environment is composed of well-known third party applications such as Apache http Server and Tomcat, and a PostgreSQL or Oracle database, which can be installed in a many of Operating Systems, including GNU/Linux or Microsoft Windows. **Error! Reference source not found.**

Openbravo ERP follows a Model Driven Development (MDD) approach. This means that Openbravo uses a technology agnostic model to define application components, such as windows and processes. Based on this application model, java code and other software artifacts are generated. Openbravo model information, the metadata, is stored in the Openbravo Application Dictionary. The process to generate the code from metadata is called Wizard for Application Development (WAD). **Error! Reference source not found.**

Modularity is a new capability introduced in the Openbravo, which allows defining and packaging additional functionality and configurations as extension modules. Modularity means that, instead of customizing the core code to fulfill user requirements, it is possible to externally extend Openbravo ERP functionality and to configure it by an independent module. **Error! Reference source not found.**

Features

Procurement Management includes rates, purchase orders, goods receipts, invoice registration and accounting, purchase planning, etc. Openbravo's handling of the flow of supply guarantees the integrity, tracking, and homogeneity of the entire process. Each document in the supply process is based on the information contained in the previous document, so that repetitive introduction of data and human errors are avoided. **Error! Reference source not found.**

Warehouse Management processes built into Openbravo allows the inventory to always be up to date and correctly valued. The possibility of defining the warehouse structure to unit level (storage bins) facilitates the exact localization of a stock at any time. The warehouse management includes warehouses and bins, warehouse units, lots, serial numbers, packages, labels, receipts and deliveries, movements between warehouses, inventories, stock valuation, transport, etc. **Error! Reference source not found.**

Project and Service Management functionality is orientated towards companies whose activities are based on the delivery of projects and services. With relationship to projects, Openbravo allows for the management of budgets, phases, tasks, expenses and purchases related with each individual project. These projects may be related to monitoring construction projects or even sending out and sales and purchase related requests. **Error! Reference source not found.**

Production Management and plant management in Openbravo allow a complete shaping of the productive structure of each organization as well as the relevant data for production: production plans, and products used to make one another. The functionality provided by Openbravo is orientated towards covering the usual necessities of a discrete production environment: production planning and requests related to procurement using MRP, creation of manufacturing orders, job reports, calculating costs of production, notification of job incidents and maintenance reports. **Error! Reference source not found.**

Sales Management and Customer Relationship Management (CRM) module is expressly designed with the objective of allowing flexibility and adaptability in its execution, needed in commercial processes. It is possible to link documents, like orders, shipments and invoices, in any order that the company requires or even disregard any one of these that is not necessary. CRM provides functionality for prices, rates, varying quantity sales orders, shipments, invoicing, volume discounts, commissions, CRM, etc. **Error! Reference source not found.**

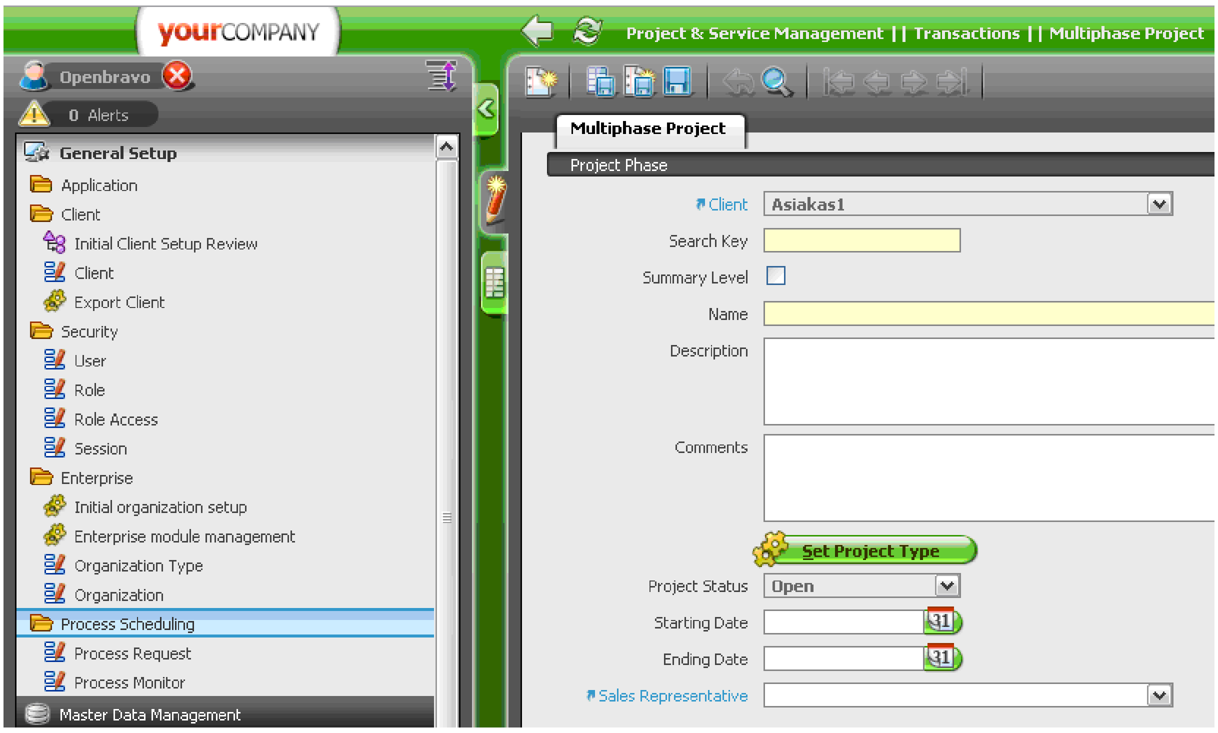
Financial Management functionality provided by Openbravo is designed to minimize manual data input on behalf of the user, thereby freeing them from routine tasks and allowing greater focus on other, more value added tasks. This module includes chart of accounts, accounts, budgets, taxes, general accounting, accounts payable, accounts receivable, bank accounting, balance sheet, P&L, fixed assets, etc. **Error! Reference source not found.**

Business Intelligence (BI) component, which is integrated into the management system, will help monitoring the state of a company, providing you with the relevant information for decision-making. The predefined balanced scorecard will allow you to verify, through the monitoring of a series of key indicators, if the defined strategy is being correctly implemented in your organization. **Error! Reference source not found.**

User experiences

According to the hands on tests, the first impression of Openbravo's user interface is quite confortable and menu options are easily accessible. The user can view production information, inventory, customer information, order tracking, and workflow information all from a Web browser. Clicking on the option opens up all the associated tasks in a drop-down. For example, transactions such as sales and shipment orders are accessible under the Sales Management menu. Each task window is icon-driven, and the icons are pictorially easy to understand. In addition, reports seem to be easy to create and there are several templates, as well as the ability to create customized ones. **Error! Reference source not found.** represents the user interface of Openbravo.

On the other hand, the functionality is not always so clear that was expected. For example, when inserting an initial data like organization's parameter, clients and users, it was quite confusing that user-item can be found under Security-folder. Why is that? Another confusing thing is the naming policy of items. How entity, role, customer and client differ from each others, or what kind of relationship they have with each others? While using the tool, somehow the feeling is like being lost. Fortunately, user manual is clear and very helpful in problematic or unclear situations.



Client: Asiakas1
Search Key:
Summary Level:
Name:
Description:
Comments:
Project Status: Open
Starting Date:
Ending Date:
Sales Representative:

Figure 18. Screenshot from Openbravo UI.

Conclusions

Openbravo is quite well-known open source ERP system. It is quite widely used in small- and medium-sized organizations around the world, so the expectations about it were high. The feature list is persuasive and the user interface looks professional, so everything seems to be in a good condition. Moreover, it is modular and should be tolerably easy to customize and expand, because it is developed on MDD framework.

However, a short-term hands-on test gave a little bit contradictory feeling about Openbravo. Even if the UI looks pleasant, the usability was sometimes more or less crappy. Relations between the clients, projects, roles and users were a bit unclear. But then, there were also many things which were implemented well. For example, forms and tables looked wery good and were mainly easy to use and handle. However, it should be noted that Openbravo hands-on tests lasts only one day, so it did not carry weight so much. Under the circumstances, training sessions with an extensive excersises are recommended, in order that system could be used more efficiently.

APPENDIX E. Interviews (in Finnish)

Johdanto

VTT:n tutkimusprojektissa ”YRTTI” (Yritystietojärjestelmien avoimet vaihtoehdot) selvitettiin yritysten toiminnanohjausjärjestelmien (ERP -järjestelmien) hankitaan ja käyttöön liittyviä haasteita sekä parhaita toimintatapoja. Tutkimuksessa haastateltiin kuuden erilaisen organisaation tietohallintojohtajaa. Haastateltuihin organisaatioihin sisältyi yksi valmistavan teollisuuden P&K –yritys, kaksi suurta julkista organisaatiota sekä kolme suurta valmistavan teollisuuden yritystä.

Haastattelut suoritettiin puolistrukturoituina haastatteluina (haastattelurunko on liitteenä). Haastatteluiden tarkoituksena oli selvittää yritysten toimintatapoja toiminnanohjausjärjestelmiä hankittaessa seuraavilla pääosa-alueilla:

- miten hankitaan oli valmistauduttu
- mitä asioita pyrittiin huomioimaan
- miten erilaisia ERP -järjestelmiä arvioitiin
- miten valitun ERP -järjestelmän implementointi onnistui
- miten valittu ERP –järjestelmä on saatu tehokkaaseen käyttöön
- miten ERP-järjestelmän vaihto onnistuisi
- mitä voisi tehdä toisin käyttöönottoprojektissa sekä järjestelmän valinnassa

Lisäksi haastatteluissa kysyttiin yritysten edustajien näkemyksiä markkinoilla nouseviin ERP –järjestelmiin, eli SaaS (Software as a Service) - pohjaisiin sekä avoimeen lähdekoodiin perustuviin ERP-järjestelmiin.

ERP-järjestelmän hankintaprosessi

ERP -järjestelmän valinta suoritetaan arvioimalla olemassa olevia järjestelmiä itselle tärkeiden ominaisuuksien pohjalta. Usein järjestelmää hankkiva yritys perustaa projektiryhmän, jonka jäsenet arvioivat vaihtoehtoiset järjestelmät yhteisten arviointiperusteiden pohjalta. Esimerkkejä käytetyistä arviointiperusteista ovat:

- Kansainvälisyys
- Järjestelmän toimittajan ja implementoijan uskottavuus
- Käytettävyys ja joustavuus
- Toiminnallisuus
- Kokonaiskustannukset
- Rajapinnat
- Käyttöympäristö

Kansainvälisyys osoittautui haastatteluissa tärkeimmäksi kriteeriksi lähes kaikille haastatelluille; erityisesti valmistavassa teollisuudessa valmistuslaitokset sijaitsevat useassa eri maassa ja niiden kokonaisvaltainen hallinta halutaan yhden järjestelmän taakse. Kansainvälisyys tarkoittaa tässä yhteydessä myös sitä, että järjestelmä toimii eri maiden lainsäädäntöjen mukaisesti esimerkiksi kirjanpidossa. Liiketoiminnan kansainvälistyminen usein laukaiseekin ERP-järjestelmän hankintaprosessin yrityksessä.

Toimittajan (ERP-järjestelmän valmistaja) ja **implementoija** (ERP -järjestelmän asentaja, kouluttaja ja kustomoija) **uskottavuus** olivat tärkeitä valintakriteerejä. Valitsemalla näihin tehtäviin uskottavat ja luotettavat partnerit pyritään alentamaan ERP -projektin epäonnistumisen riskiä. Osa haastatelluista korosti implementoijan roolia ja totesi että roolia olisi pitänyt korostaa jopa enemmän, koska implementoijan toimiala- ja järjestelmäosaaminen on osoittautunut merkittäväksi tekijäksi ERP järjestelmän käyttöönoton onnistumisessa.

Käytettävyys ja joustavuus olivat tärkeimpiä prioriteetteja kahdelle yritykselle. Käytettävyydestä oltiin kuitenkin valmiita luopumaan, mikäli näin pystyttiin hankkimaan teknisesti parempi järjestelmä. Yleisesti käytettävyyteen olisi pitänyt haastateltavien mielestä panostaa enemmän valintaprosessissa, koska vaikeakäyttöinen järjestelmä aiheutti käyttäjien keskuudessa muutosvastarintaa sekä erilaisia käyttövirheitä.

Toiminnallisuus oli myös merkittävä tekijä järjestelmää valittaessa. Osa haastatelluista oli määritellyt laajat toiminnalliset vaatimukset järjestelmälle, rajaten mahdollisten järjestelmien määrän muutamaan vaihtoehtoon, joiden välillä toiminnallisuudessa ei ollut suuria eroja. Kaikissa ERP järjestelmissä oli lähes samat toiminnallisuudet eikä toiminnallisia puutteita ERP järjestelmissä käytännössä ole, varsinkaan valmistavan teollisuuden piirissä. Muutamia puutteita tunnistettiin palveluorganisaatioissa ja ne kohdistuivat projektinhallintaprosessien toiminnallisuuteen.

Kokonaiskustannukset olivat merkittävä tekijä järjestelmän valinnassa, mutta ei yleensä tärkein kriteeri varsinkaan suurissa yrityksissä.

Rajapinnat ja raportit eli mahdolliset liitynnät muihin järjestelmiin olivat tärkeä kriteeri osalle organisaatioista. Näihin myös toivottiin parannuksia. Erityisesti toivottiin avoimia ja standardin mukaisia rajapintoja, joita voitaisiin hyödyntää tietojen viennissä toisiin järjestelmiin ja raportointityökaluihin tai tietovarastoihin. Monet yritykset käyttivät erityistä raportointityökalua ja tietovarastoa, joista tuotettiin raportteja päätöksenteon tueksi ja siksi rajapinnat koettiin tärkeiksi.

Käyttöympäristö eli ERP järjestelmän teknisen suoritusympäristö ei ollut merkittävä valintakriteeri, koska kaikki ERP järjestelmät toimivat yleisesti käytetyissä käyttöjärjestelmissä sekä sovelluspalvelimissa ja tietokannoissa.

ERP järjestelmien kustannusjakauma

ERP-järjestelmän hankintaan liittyy useita eri kustannuksia, jotka voidaan jakaa seuraaviin osiin: **lisenssimaksuihin** (sisältää perusylläpidon), **sisäisiin kustannuksiin** (henkilöstön kouluttaminen ja tukihenkilöstö) sekä **ulkoisiin kustannuksiin** (konsultit, varsinaisen asennustyö ja kustomoinnit).

Haastattelujen perusteella lisenssimaksujen osuus vaihtelee 10% - 25% kokonaiskustannuksista. Osa haastatteluissa piti lisenssimaksuja varsin kohtuullisina ja koki saavansa rahoilleen riittävää vastinetta. Toinen osa haastatelluista piti kyllä nykyisiä lisenssimaksuja melko kohtuullisina, mutta toimittajan ehdottamat lisenssimaksujen korotukset koettiin liian suuriksi.

Suurimman kustannuserän luovat ulkoiset kustannukset, eli konsulttien käyttö sekä tarvittavien muutosten ja rajapintojen toteuttaminen, käyttöönottoon liittyvät datakonversiot sekä eri ympäristöjen ylläpitäminen (esimerkiksi testi-, kehitys- ja koulutusympäristöt). Järjestelmään tehtävät muutokset toteutetaan yleensä kolmannen osapuolen toimesta, vain muutama suomalainen ERP -toimittaja toteuttaa itse myös kustomoinnit ja asennuksen. Sisäisiä kustannuksia ei erityisesti korostettu, mutta kaikki tunnistivat että niitä syntyy, varsinkin jos ERP projekti viivästyy tai käyttöönoton koetaan muuten epäonnistuneen. Tällöin sisäiset kustannukset kasvavat merkittäviksi, eikä niitä ole enää helppoa hallita.

Haasteet ja ongelmat

Suurin yksittäinen haaste ERP-järjestelmän hankinnassa on yrityksen omien prosessien sovittaminen ERP -järjestelmään tai ERP järjestelmän räätälöinti vastaamaan yrityksen prosesseja. Useissa tapauksissa yrityksen omat liiketoimintaprosessit eivät ole selvästi kuvattuja tai kuvattuja prosesseja ei oikeasti noudateta yrityksen sisällä. Tästä seuraa ongelmia, sillä ERP -järjestelmän toiminta perustuu prosessien seurantaan ja toteutumiseen. Joskus ERP -järjestelmää räätälöidään vastaamaan yrityksen prosesseja, mutta täydellistä yhteensopivuutta on vaikea saavuttaa. Yrityksen on siis käytännössä pakko ainakin osittain sovittaa omia prosessejaan järjestelmän mukaisiksi. Haastateltavat olivat yksimielisiä siitä, että yrityksellä, jolla on kypsät liiketoimintaprosessit, on parhaimmat mahdollisuudet ottaa ERP -järjestelmä onnistuneesti käyttöön.

ERP-järjestelmän kustomoinnin ja asentamisen hoitaa yleensä implementoija. Implementoijan rooli on suuri, sillä heidän on hallittava sekä tuotteen kustomointi että asiakkaan toimialaan liittyvien tarpeiden huomiointi. Käytännössä implementoijan osaaminen ei aina ole ollut riittävää. Mikäli implementoijalla ei ole riittävää osaamista ERP -järjestelmän jalkauttamiseen, järjestelmää ei välttämättä saada oikein tai kokonaan käyttöön. Tällöin kaikkia järjestelmien moduuleita ei saada käyttöön, ja todellisuudessa vain murto-osa ominaisuuksista voi olla käytössä. Uusien ominaisuuksien käyttöönotossa on myös ollut hankaluuksia, yritys voi joutua ottamaan käyttöön (ostamaan) tarpeettomia ominaisuuksia samalla.

Järjestelmän implementoijan vaihtoa ei koettu ongelmalliseksi ja suuri osa haastatelluista olikin vaihtanut implementoijaa. Implementoijan vaihtaminen oli myös parantanut palvelun laatua. Sen sijaan ERP - järjestelmän vaihto on todella hankalaa ja tarkoittaisi käytännössä uutta ERP projektia. Järjestelmän vaihtomahdollisuutta ei tosin pidettykään erityisen tärkeänä vaan pyrittiin sitoutumaan yhteen järjestelmää ja yhteen implementoijaan.

Johdon ja henkilöstön sitoutuminen oli myös avain asemassa: parhaiten onnistuneet ERP -projektit olivat onnistuneet sitouttamaan henkilöstön ja johdon (myös keskijohdon) ERP:n käyttäjiksi menestyksekkäästi. Tällöin ERP -järjestelmä oli myös otettu käyttöön tehokkaasti. Parhaaksi keinoksi alentaa kokonaiskustannuksia tunnistettiin järjestelmän tehokas käyttö organisaatiossa: mitä enemmän käyttäjiä ERP -järjestelmällä on, sitä alemmat ovat sen käyttökustannukset per käyttäjä, ja järjestelmä alkaa oikeasti tuottaa synergiaa. Myös muutosvastarinta todettiin haastatteluissa ongelmalliseksi ERP:n käyttöönotossa, mutta henkilöstön kouluttamisella tähän voitiin vaikuttaa. Henkilöstön sitouttaminen muutokseen koettiin haastatteluissa yrityksen johdon tehtäväksi.

Järjestelmän valintaan ja käyttöönottoon oli usein määritetty liian tiukka aikataulu. Aikataulu tulee usein ulkopäin eikä siihen pystytä vaikuttamaan. Asianmukainen projektiryhmän resurssointi ja aikataulutus onkin tärkeä tekijä ERP – projektin onnistumiselle.

Vaihtoehtoja ERP:n hinnan ja laadun parantamiseksi

Kilpailun lisääntyminen markkinoilla voisi laskea erityisesti lisenssipohjaisia kustannuksia. Lisäksi räätälöintien määrän pitäminen mahdollisimman vähäisenä auttaa kustannusten hallitsemisessa, eli kannattaa valita järjestelmä, joka sopii mahdollisimman hyvin yrityksen liiketoiminnan luonteeseen ja prosesseihin. Lisäksi olisi hyvä eristää räätälöinnit niin, että ne voitaisiin sisällyttää versionvaihtoihin. Tämä auttaisi vähentämään kallista räätälöintityötä.

ERP -järjestelmissä ei koeta haastatteluiden pohjalta olevan suuria teknisiä puutteita tai ongelmia. Suurin vaikutus kustannuksiin voitaisiin saada onnistuneella kokonaisprojektinhallinnalla, jossa yrityksellä itsellään on lähes yhtä suuri rooli kuin ERP-järjestelmän toimittajalla.

Käyttöliittymät koettiin hankaliksi ja niihin toivottiinkin parannusta. Yksittäisinä keinoina mainittiin toimintojen johdonmukaisuuden ja yhtenäisyyden kehittäminen sekä käyttöliittymän parametrisointi, eli yrityksen kannalta tarpeettomien syöttökenttien poistaminen näytöltä sekä yrityskohtaisten tarkistussääntöjen tekeminen syöttökenttiin.

Myös takuun myöntäminen ERP -järjestelmille nähtiin mielenkiintoiseksi mahdollisuudeksi.

Vaihtoehtoja nykyisille ratkaisuille

Haastatellut pitivät ERP -järjestelmien markkinoita kypsinä ja uusien yritysten ja järjestelmien pääsy markkinoille nähtiin haastavaksi. Mahdollisuuksia nähtiin P&K – sektorille mukaan pääsyyn, etenkin uusilla toimintamalleilla (SaaS ja OpenSource). Sitä kautta olisi mahdollista tulevaisuudessa saada jalansijaa myös isompien yritysten ERP -järjestelmistä.

SaaS pohjainen ERP

Ohjelmistopalveluna (Software-as-a-Service - SaaS) toimitettavat ERP-järjestelmät ovat nousseet markkinoille myös Suomessa. Tyypillisesti SaaS-pohjaisia ohjelmistoja tarjotaan kuukausihinnalla käyttäjämäärien mukaan. Tällaiset ERP-järjestelmät koettiin mahdollisiksi, mutta nykyisellään liian kevyiksi erityisesti valmistavaa teollisuutta silmälläpitäen. Yrityskohtaisten räätälöintien, jotka voivat olla huomattaviakin, toteuttaminen koetaan haasteelliseksi SaaS – pohjaisille ERP:lle. Lisäksi toimittajiksi kaivataan luotettavia ja uskottavia toimittajia, joita nuorella SaaS – toimialalla ei vielä koeta olevan. SaaS suuntaus nähtiin kuitenkin varsin mielenkiintoisena ja sen uskottiin lisääntyvän erityisesti palveluliiketoiminnassa, mutta myös valmistavan teollisuuden osalta tietyissä toiminnoissa, esimerkiksi asiakkuudenhallinnassa..

Avoimeen lähdekoodiin perustuva ERP

Avoimeen lähdekoodin pohjautuvat ERP-järjestelmät koettiin haastatteluissa varteenotettaviksi vaihtoehdoiksi nykyisille kaupallisille järjestelmille. Näihin järjestelmiin suhtauduttiin positiivisesti, eikä niiden leviämiselle nähty merkittäviä esteitä, jos ne täyttävät samat vaatimukset kuin suljetut lisenssipohjaiset ERP-järjestelmät.

Avoimeen lähdekoodin pohjautuvia ERP – järjestelmiä on saatavilla useita, joista latausmäärien perusteella suosituimpia ovat OpenERP, OpenBravo ja Compiere. Avoimeen lähdekoodin perustuvan ERP-järjestelmän voi ladata ja asentaa kuka vain, mutta yleensä niiden käyttöön ostetaan avuksi partneri. Usein avoimia ERP - järjestelmiä tarjotaan asiakkaille SaaS - pohjaisina, joissa implementoijan liiketoiminta perustuu asennukseen ja erilaisiin palveluihin kuten esimerkiksi tekniseen tukeen ja räätälöintiin avoimen ERP - järjestelmän ympärillä.

Suomessa avoimen lähdekoodin ERP-järjestelmän toimittajia on vain muutamia ja tämä hidastaa huomattavasti näiden yleistymistä markkinoilla. Luotettavan ja uskottavan partnerin löytyminen ERP - järjestelmän toteuttamiseen on tärkeä kriteeri, josta ei haluta joustaa. On kuitenkin todennäköistä, että avoimet ERP-järjestelmät saavat jalansijan P&K – yrityksistä, joille riittää kevyempi ERP -ratkaisu. Hyvät kokemukset avoimista ERP järjestelmistä auttavat niiden leviämistä edelleen.

”Best of breed” pohjainen ERP

Aikaisemmin käytössä olleet ns. ”best of breed” ratkaisut olivat kalliita, koska jokaisesta moduulista tarvittiin omat lisenssit, toimittajasopimukset, palvelimet yms. Lisäksi tieto oli hajallaan ja usein kopioituna moneen paikkaan. Lisäksi jokaiseen järjestelmään piti kirjautua erikseen ja järjestelmien integrointi oli kallista ja vaati ylläpitoa. Laajat ja hajanaiset BoB ratkaisut eivät tule enää kysymykseen, mutta osa aiemmista ongelmista voidaan nykyään ratkaista, esimerkiksi kertakirjautuminen on ainakin osittain mahdollista ja myös työkalujen integroitavuus on helpompaa. Käytännössä haastateltavilla oli myös käytössä tietyntyylinen BoB –lähestyminen, eli ERP – järjestelmä toimi yhteistyössä yhden tai kahden olemassa olevan järjestelmän kanssa (esim. varastonhallinta, taloushallinto, CRM). Haastattelujen perusteella on vaikea sanoa millä tasolla ”best of breed” tulee etenemään. Isot yritykset tullevat jatkossakin todennäköisesti käyttämään useita järjestelmiä ja keskitettyä tietovarastoa (joka auttaa tietojen hajanaisuuden ongelmaan). Pienemmät yritykset voivat puolestaan sitoutua yhteen tai kahteen pääjärjestelmään.

Haastattelujen yhteenveto

Tutkimuksessa haastatellut yritykset olivat hyvin tietoisia ERP -järjestelmän hankintaan liittyvistä riskeistä ja olivat kriittisiä myös oman organisaationsa valmiuksista ERP -projektiin. Kaikki haastateltavat yritykset olivat tehneet analyysit ERP -järjestelmistä sekä pyrkineet resursoimaan voimavaroja vaativaa projektia varten. Useimmilla haastateltavilla oli ollut käytössään pisteytyskriteerit, joiden pohjalta ERP -järjestelmiä arvioitiin projektiryhmän toimesta. Tästä huolimatta usealle yritykselle tuli yllätyksenä yritysten omien prosessien ja ERP-järjestelmän yhteensopivuuden luomisen vaativuus. Mielenkiintoinen tulos oli, että yritykset olivat varsin itsekriittisiä: useimpien ERP -järjestelmän käyttöönoton ja käytönongelmien syyt nähtiin yritysten omassa toiminnassa eikä juurikaan järjestelmässä. Osa ongelmista toisin johtui puutteista järjestelmän implementoijan toimiala- tai ERP järjestelmän osaamisesta.

Liiketoimintaprosessien yhteensopimattomuus toiminnanohjausjärjestelmän kanssa johtaa lähes aina ongelmiin, jotka näkyvät viivästyneinä aikatauluina sekä kasvavina kustannuksina. Lisäksi syntyy vaara ERP -järjestelmän käyttämisestä väärin, jolloin myös järjestelmä tuottaa väärää tietoa liiketoimintaprosesseista. Tämä taas nähdään usein käyttäjien keskuudessa ERP järjestelmän toimimattomuutena. Toinen yleinen ongelma on liiketoiminnan laajentumisesta ja kansainvälistymisestä aiheutuvat ERP –järjestelmään kohdistuvat vaatimusten muutokset. Vanhoilla vaatimuksilla hankitut ERP-järjestelmät eivät vastaa tämän päivän liiketoiminnan tarpeisiin, ja järjestelmän kehittäminen on joko teknisesti mahdotonta tai koetaan liian kalliiksi. Tämä onkin usein syy uuden ERP järjestelmän hankinnalle.

Haastateltavat toivat myös selkeästi ilmi halun sitoutua ERP-järjestelmän tuottajaan: ERP -järjestelmä on liiketoimintakriittinen sovellus jonka tueksi haluttiin rakentaa pitkäaikaista suhdetta. Toimittajan vaihtaminen sen sijaan koettiin olevan tarvittaessa helpostikin mahdollista, ja osa oli jo näin tehnyt.

Haastatteluissa selvisi myös, että ERP -järjestelmien lisenssimaksut ovat varsin kohtuullisella tasolla, eivätkä haastateltavat kokeneet niitä liian kalliiksi. Suurimmaksi kustannukseksi koettiin järjestelmän räätälöinnistä aiheutuvat kulut, sekä räätälöinneistä aiheutuvat ongelmat jatkossa kun järjestelmiä päivitetään. Varsinainen ERP -järjestelmä koettiin haastatteluissa varsin toimivaksi ja virheettömäksi. Ongelmatilanteet syntyvät usein ERP -järjestelmän väärästä käytöstä, johon suurin yksittäinen syy on järjestelmän vaikeakäyttöisyys.

Tulevaisuudessa avoimien rajapintojen odotetaan yleistyvän, osittain avoimen lähdekoodin ohjelmistojen yleistymisen vaikutuksesta sekä osittain eri toimialojen standardoitumisesta (eli toimialan eri osapuolet sopivat esimerkiksi tietojen välitysformaateista ja rajapinnoista). Haastatteluissa kävi ilmi ennakkoluulottomuus avoimen lähdekoodin ERP -järjestelmiä kohtaan sekä tietoisuus niiden olemassa olosta. Käyttöliittymien sekä rajapintojen räätälöinnin rooli tulee korostumaan tulevaisuudessa, jolloin järjestelmien muutettavuus nousee suurempaan roolin. Avoimen lähdekoodin ERP -ohjelmistot nähtiin toteutuvan lähinnä SaaS – muotoisina palveluina, ensimmäisenä käyttäjäkuntanaan P&K -yritykset.

Haastatteluissa tuli esiin seuraavat kriittiset menestystekijät ERP -järjestelmän käyttöönoton onnistumiseen:

- Projektinhallinta ja projektin johtaminen
- Liiketoimintaprosessien uudelleenorganisointi
- ERP järjestelmään tehtävät räätälöinnit,
- Toimittajan (implementoijan) osaaminen (järjestelmä ja toimiala)
- Johdon tuki
- Muutosjohtaminen
- Käyttäjien koulutus.

ERP järjestelmän valinta oli usein tehty pisteyttämällä mm. seuraavia tekijöitä.

- Toiminnallisuus/sopivuus em. kansainväliseen toimintaan,
- Tuotteen käytettävyys
- Käyttöympäristö (käyttöjärjestelmä yms.)
- Hinta
- Tuotteen tulevaisuus

- Rajapinnat

ERP järjestelmien toiminnallisuus koettiin yritysten keskuudessa riittäväksi, mutta varsin suureksi ongelmaksi koettiin järjestelmien huono käytettävyys (käytönnopeus, toimintojen yhteneväisyys ja loogisuus ja järjestelmän yleinen informatiivisuus).

Mahdollisuuksina kustannusten alentamiseen nähtiin räätälöintien välttäminen ja eristäminen (vähempitöiset päivitykset). Lisäksi avoimen lähdekoodin ja SaaS konseptin mukaiset järjestelmät nähtiin mahdollisina vaihtoehtoina, mutta niiden uskottavuudessa oli vielä kehittämisen varaa.

Johtopäätökset

Haastattelujen ja tutkimuksen perusteella tunnistettiin muutamia oleellisia tekijöitä, joilla voitaisiin parantaa ERP -järjestelmän käyttöönottoa ja käyttöä.

Aikataulu. ERP projektin aikataulu olisi hyvä tehdä ERP käyttöönoton vaatimalla aikataululla ilman ulkoisia paineita (esimerkiksi uusi organisaatio astuu voimaan tai uusi tehdas aloittaa toiminnan tiettynä ajanhetkenä, johon mennessä järjestelmä pitää olla tuotannossa). Nämä ulkoiset paineet aiheuttavat usein liikaa kiirettä, jolloin onnistumisen kannalta oleellisia asioita ei ehditä tehdä kunnolla. Käytännössä riittävän ajan varaaminen käyttöönotolle on kuitenkin haasteellista, koska juuri edellä mainitut asiat usein laukaisevat ERP -järjestelmän hankinnan ja määrittävät aikataulun.

Vaiheistus. Onnistuneet ERP projektit olivat usein hyvin vaiheistettuja, ensimmäisessä vaiheessa pyritään tuomaan järjestelmään perusominaisuudet kuten omat liiketoimintaprosessit valmistuksessa sekä tavarantoimittajat. Myöhemmissä vaiheissa järjestelmää pyritään laajentamaan perustoiminnallisuuden yli lisäämällä siihen uusia moduuleita ja toimintoja, kuten asiakashallinta, henkilöstöhallinta ja projektinjohto. Liian suuren kokonaisuuden hallinta tuottaa usein hankaluuksia vaikka onnistuessaan se voi alentaa kokonaiskustannuksia. Ensimmäinen vaihe on usein hankalin ja se olisi tehtävä erityisen huolellisesti, koska puutteet ensimmäisessä vaiheessa heijastuvat negatiivisesti muihin vaiheisiin.

Toimintaprosessit. Oleellinen asia ERP järjestelmien käyttöönotossa on toimintaprosessien kypsyys. Kypsät ja ERP järjestelmän kanssa yhteensopivat prosessit helpottavat järjestelmän käyttöönottoa huomattavasti. On tärkeää myös varmistaa, että määriteltyä prosesseja noudatetaan yrityksessä, eli kuvattu prosessi vastaa käytäntöä. Usein käyttöönoton tai tuotantokäytön alkuvaiheessa huomataan, että kuvattuja prosesseja ei käytännössä olekaan noudatettu ja tämä aiheuttaa monenlaista sekaannusta.

Organisaation sitoutuminen ja muutosvastarinta. ERP järjestelmän hankinnan ja käyttöönoton kannalta olennaista on johdon ja koko organisaation sitoutuminen järjestelmän käyttöönottoon ja käyttäjien muutosvastarinnan minimoiminen. Nämä asiat eivät ole teknisiä tai teknologisia asioita vaan enemmänkin johtamiseen ja työpsykologiaan liittyviä asioita. Käyttäjien muutosvastarinta ja organisaation kaikkien tasojen puutteellinen sitoutuminen aiheuttaa ERP järjestelmän käyttöönoton epäonnistumisen.

Koulutus. Koulutuksen merkitys on erittäin suuri ERP järjestelmän onnistuneelle käyttöönotolle ja käytölle. Käyttäjien koulutuksen laatuun on myös kiinnitettävä huomiota. Pelkästään järjestelmän käyttökoulutus esimerkiksi koulutustiloissa ei riitä, vaan käyttäjille on koulutettava myös yrityksen toimintaprosessit eli miten yritys toimii, mitä vaihetta kukin käyttäjä tekee kokonaisprosessissa ja mihin kaikkeen oma työ vaikuttaa. Tämä lisää käyttäjien motivaatiota ja he ymmärtävät oman työn huolellisen suorittamisen merkityksen.

Koulutusta ja osaamisen varmistamista on myös suunnattava käyttöönoton jälkeiseen aikaan, jolloin tulee varmistaa että järjestelmää käytetään oikein. ERP –järjestelmää tehokkaasti hyödyntävissä yrityksissä, käyttäjät osaavat aidosti käyttää järjestelmää oikein ja he myös tietävät millaisia ongelmia aiheutuu, jos ERP järjestelmää käytetään väärin tai huolimattomasti.

ERP järjestelmän käytettävyys. ERP järjestelmän helppokäyttöisyys ja käytettävyys (järjestelmän nopeus, toimintojen johdonmukaisuus ja yhtenäisyys) on jäänyt liian vähälle huomiolle sekä ERP järjestelmien kehittäjien kehityspainotuksissa että yritysten valintakriteereissä. Vaikeakäyttöinen järjestelmä aiheuttaa tarpeetonta muutosvastarintaa ja käyttövirheitä myös kokeneiden käyttäjien keskuudessa. Tulevaisuus näyttää hiukan valoisammalta, koska seuraavissa ERP järjestelmien versiossa on keskitytty järjestelmien käytettävyyden parantamiseen.

Mielenkiintoinen johtopäätös tästä tutkimuksesta on se, että ERP projektien mahdolliset ongelmat sekä onnistumisen ja epäonnistumisen kannalta tärkeät asiat ovat kaikkien tiedossa ja tietoa niistä on yleisesti saatavilla. Silti useat ERP projektit koetaan joko kokonaan tai ainakin osittain epäonnistuneeksi. Onkin todettava, että tärkein asia ERP projektin onnistumisen kannalta on tärkeiden ja oleellisten asioiden tekeminen oikein. Ei siis riitä, että yritys tiedostaa ja kohdistaa toimintaa ERP järjestelmän kannalta tärkeisiin asioihin vaan tärkeät asiat on tehtävä oikein ja mahdollisuuksien mukaan vielä varmistettava, että ne on tehty varmasti oikein. Hyvänä ja yksinkertaisena esimerkkinä voidaan pitää koulutusta ja johdon sekä käyttäjien sitouttamista, joissa ei riitä että käyttäjiä koulutetaan luokkatiloissa tai käyttäjiä tiedotetaan, vaan tulee myös varmistaa, että käyttäjät todellakin osaavat käyttää järjestelmää ja tuntevat prosessit sekä toimivat niiden mukaisesti. Suuri yksittäinen syy siihen, että tärkeitä asioita ei tehdä oikein, on liian tiukat käyttöönottoprojektin aikataulut eli asioita ei ehditä tehdä oikein. Toinen merkittävä tekijä on yrityksen kulttuuri eli kuinka hyvin työntekijät ja johto ovat sitoutuneet yritykseen ja yrityksen toiminnan kehittämiseen yleensä.