

*Open source alternatives for ERP (LX Office,
SQL Ledger, and ERP5)*

Author: Ahmad Jawed Samsor

Date: 05/16/2013

Table of Contents

Part One:

1. Introduction
 - 1.1. Abstract
 - 1.2. Definition of ERP System
 - 1.3. Definition of Open Source
 - 1.4. Definition of Small and Medium Enterprises

Part Two:

2. Evaluation Criteria
 - 2.1. Functional Fit
 - 2.2. Flexibility
 - 2.3. Support
 - 2.4. Continuity
 - 2.5. Maturity

Part Three:

3. Overview of Open Source ERP Systems
 - 3.1. Comparison Table
 - 3.2. LX Office
 - 3.3. SQL Ledger
 - 3.4. ERP5

Part Four:

4. Conclusion

1. Introduction:

1.1. Abstract

Open source ERP systems are often targeted to enterprises whose requirements are not covered by standard software. Similar circumstances apply to organizations that need continuous adaptation of the software to changing processes and needs. In this paper the suitability of current open source ERP systems for these enterprises will be examined.

It provides sufficient information for a small or medium enterprise to choose a flexible and adaptable open source ERP system. Starting from the question which opportunities a company has to support its processes with IT, the advantages of flexible systems are elaborated. Besides the focus on flexibility, open source specific criteria for support, continuity and maturity are worked out. Then selected open source ERP projects are reviewed and classified according to these criteria. The results are a criteria catalog and a classification of selected open source ERP systems.

1.2. Definition of ERP System

An enterprise resource planning (ERP) system integrates internal and external management of information across an entire organization including finance/accounting, manufacturing, sales and service, customer relationship management, etc. ^[1]

These activities are automated with an integrated software application by ERP to facilitate the flow of information between all business functions inside the boundaries of the organization. To further distinguish ERP systems from general application frameworks and other standard software, accounting functionality is required.

1. Bidgoli, Hossein, (2004). The Internet Encyclopedia, Volume 1, John Wiley & Sons, Inc. p. 707.

1.3. Definition of Open Source

In this paper the open source definition of the Open Source Initiative (OSI) ^[2] is used. According to OSI a software must comply to the following conditions (shorted):

1. Free redistribution, including selling or using as component without fee.
2. The Source code of the program must be available in readable form and must allow distribution.
3. Derived work must be allowed under the same terms as the license conditions.
4. Integrity of the author's source code (licenses may require that modifications are redistributed only as patches).
5. No discrimination is allowed against any person or group.
6. No discrimination against fields of endeavor.
7. Distribution of license (license applies to all whom the program is redistributed to, closing up software is forbidden).
8. License must not be specific to a product.
9. License must not restrict other software.
10. License must be technology neutral.

Licenses that conform to the above definition can get certified by OSI and may use its certification mark. The availability of the source code reduces investment risk as the

development cannot be abandoned easily. Furthermore you have the possibility to adapt the software to your needs.

2. <http://opensource.org>

1.4. Definition of Small and Medium Enterprises

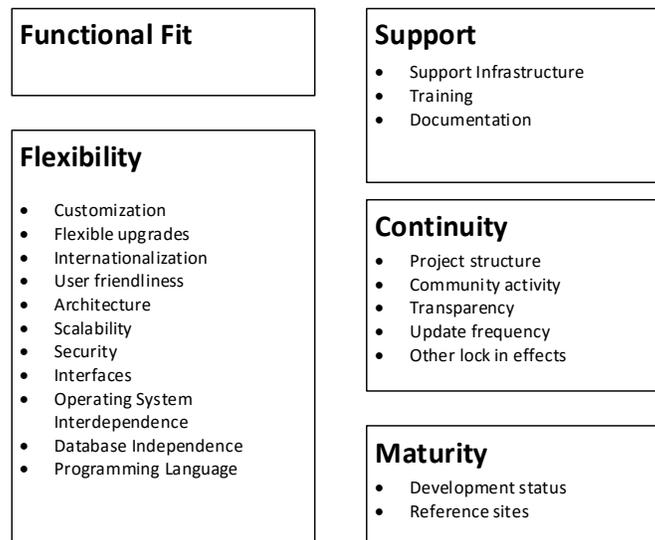
This situation reflects the relative nature of the “small” and “medium” size classifications, which can apply differently to firms in the manufacturing, agricultural, and service sectors. In this paper the definition of U.S. Department of Commerce (Commerce), the U.S. Small Business Administration (SBA), and the U.S. Department of Agriculture (USDA), is used ^[3]:

	Manufacturing and non-exporting services firms	Exporting services firms		Farms
		Most	High value	
Number of employees	< 500	< 500	< 500	< 500
Revenue	Not applicable	≤ \$7 million	≤ \$25 million	< \$250,000
Defining institution	SBA Advocacy	SBA/SBA Advocacy	SBA/SBA Advocacy	USDA
Data source	U.S. Census	ORBIS	ORBIS	USDA

3. www.usitc.gov/publications/332/pub4125.pdf (page 18-19) viewed 03/20/2013

2. Evaluation Criteria

The criteria introduced in this part of the paper are hierarchically structured and can be used as starting point for personal alteration. Most criteria are not measurable but it is more about getting some criteria for showing the strengths, weaknesses and differences of open source ERP systems. The five evaluation criteria and its sub-criteria serve to compare selected open source ERP systems the next part “Overview of Open Source ERP Systems”.



According to Lori MacVittie in her article, Implementing an ERP Takes Time and Patience, the important factors to an ERP total costs include cost of consulting, analysis, license, implementation, customization, maintenance, training, integration, support, upgrades and continuous adaption to processes.^[4] Functional fit indicates the amount of customization and additional development needed for a close fit to intended processes. Flexibility shows the opportunities to bridge the functionality gap. Support indicates the knowledge transfer needed for implementation and operation. Continuity is about project sustainability and vendor independence. Maturity points out the risk of choosing a system with inadequate quality that is not production ready. [5]

4. <http://www.networkcomputing.com/1206/1206ws2.html> viewed 04/01/2013
5. <http://www.networkcomputing.com/1206/1206ws22.html> viewed 04/01/2013

2.1. Functional Fit

The term functional fit is preferred over functionality. “The closer the ERP software models your business processes, the smoother the integration will be and the sooner you will reap the benefits.”[6] It is the degree the chosen ERP system fits to your business processes out of the box. Functional fit has a high impact on total cost and implementation time. As functional requirements vary widely depending on the business area, there is no general way to measure functional fit. The number of database tables is stated, when available, as a measurable indicator of the functional amount of an ERP system, assuming the data structure is well designed.

2.2. Flexibility

Flexibility allows bridging the functionality gap between out-of-the-box functionality and the close functional fit provided by a customized system. Apart from the opportunity to adapt the system to optimal business processes, flexibility also implies questions of ease of use and administration, and platform independence. It is about technical concepts and software design of the system. A flexible ERP system meets the following criteria:

- Customization: based on the degree of customization, it can be distributed on two different levels.
 - High level: customization through Meta data editing, which means that the system is customizable by editing easily readable and understandable data, instead of doing low-level coding in a programming language.
 - Low level: customization through (use as application framework). For those developers who want to dive further into details and need more flexibility the system should also be usable as a framework for application development.
- Flexible upgrades: an upgrade procedure that doesn't impact the customization (i.e. Meta data and custom code).
- Internationalization: providing translations for the user interface and local accounting schemes in multiple languages.
- User friendliness: a simple task shall not require navigating through many screens. For routine work keyboard shortcuts need to be provided.

- Architecture: important for most flexibility factors is the chosen architecture. The open source solutions have 2-tier or 3-tier architectures (figure 1). The 2-tier or client- server architecture consists of a “fat” client containing GUI, and business logic, communicating directly with the database. In the case of 3-tier architecture the client is just responsible for GUI and simple data validation.

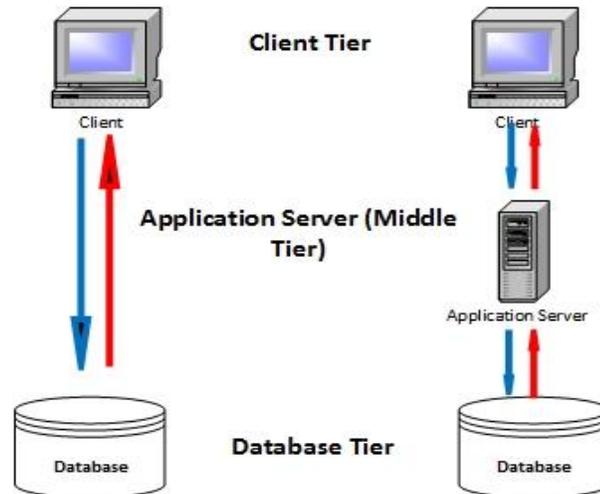


Figure 1: N-tier Architecture

- Scalability: “A system that does not scale to support all your customers is a disaster waiting to happen.”^[6] The system should support high transaction volumes with constant response times. Scalability is highly dependent on the architecture and thus on the application server and database technology used.
- Security: Users are allowed to view and change only the data they need for their work. The granularity can be defined on form, field and row level. Row level security restricts access on data level. For example, a user can only see transactions of the affiliate he is responsible for.
- Interfaces: An interface is a communication boundary of an ERP System. Interfaces are used to connect the ERP system with other systems or to generally exchange data. The former is known as Enterprise Application Integration (EAI) and uses standardized server side interfaces like CORBA (Common Object Request Broker Architecture), XML-RPC (XML- Remote Procedure Call) and SOAP (Standardized Object Access Protocol) to automate the business processes beyond system boundaries. But also integration on database level can be sufficient especially for read only data that does not have to invoke business logic. As this kind of integration is solely database specific, it will not be evaluated here.

- Operating System Interdependence: Operation system independence allows you to run the ERP system on various platforms. It is a necessary feature on the client side, if the users have different operating systems.
- Database Independence: The database has high influence on the scalability of the system. Some prefer open source databases for open source ERP systems. A survey found out that over 90% of the Opentaps interest group prefers open source databases. ^[7] There is a trade-off between database independence and database features, especially the object relational features used. High database independence also implies using a minimal common feature set provided by all supported databases. Some features lost through independence can be provided through the application or used application server.
- Programming Language: The language can be a criterion to leverage available skills for low level customization. The programming languages of the selected ERP Systems are open source scripting languages (Python ^[8], Perl ^[9]) and Java ^[10]. Python is known for its easily readable, concise syntax and its built-in refactoring ^[11] capabilities. Perl is widely used, but requires more developer discipline to get a serviceable code.

6. <http://www.networkcomputing.com/1206/1206ws22.html> viewed 04/05/2013

7. <http://www.prweb.com/releases/2005/11/prweb308560.htm> viewed 04/05/2013

8. <http://www.python.org>

9. <http://www.perl.org/>

10. <http://java.sun.com/> (Java is not open source)

11. Refactoring is the reorganization of the source code to improve internal consistency and clarity.

2.3. Support

Support helps to shorten the implementation time due to the knowledge transfer to the company. It helps to develop internal skills or engage external consultants for implementing and maintaining an open source ERP system.

- Support infrastructure: Reliable and responsive support is important ^[12]. It can be local or online. Most open source ERP projects solve the problems regarding different national requirements through a partner network. A local partner can provide consulting, support, add-on modules and address national requirements like accounting standards, interfaces to public authorities and banks. Apart from knowledge about national requirements partners also have specific industry knowledge. Online support over public, uncensored forums and mailing lists is important, because it offers users and developers the opportunity to read and discuss issues.
- Training: Here the quality and frequency of user and technical trainings or the organization of regular conferences are of certain importance.

- Documentation: Completeness and up-to-datedness of the user and developer documentation are needed. Many projects use a Wiki content management system for collaborative documentation authoring and maintenance.

12. <http://www.networkcomputing.com/1206/1206ws22.html> viewed 04/10/2013

2.4. Continuity

Project continuity ensures that your ERP system expenses are a sustained investment. When you focus on one system you also run the risk that the system will be no longer endorsed.

- Project Structure: The evaluated projects are company or community driven. Company driven means that a company is responsible for the development, provides services and certifies partners for local support. A typical company driven project has the following participants: open source project company, partner companies, customers with support contract, customers without support contract and the users working with the system. A company-driven project can have source code contributions from the user community and partners. Community driven means that the development is cooperative and there is no single company solely responsible.
- Community activity: the number of messages in forums and mailing lists, quantity, qualified answers and response times by the community members. Documentation activity like web site creation and wiki entries form a part of support/documentation.
- Update frequency: A change log document informing about features of a new release shows past update activity. Whereas community activity is about communication, regular updates show development activity.
- Other lock in effects: Besides the lock-in on the project itself, possible side effects can stem from used (e.g. commercial) components, technologies or dependencies on other open source projects. Operation system independence, database independence and programming language, which were discussed in the flexibility criteria section, are also lock-in related criteria.

2.5. Maturity

Here maturity is used in a more narrow context and means the quality of software. Whereas flexibility is about technical concepts and the design of the software, maturity tells you how well and bug free it is implemented and tested.

- Development status: Some open source ERP packages are not ready for production yet. The concept of the development status of Sourceforge is also applied to non Sourceforge hosted open source projects. They can be in the state of planning, alpha, beta or stable. Planning implies that the specifications of a software have been defined and no executable program is available. The first release of a computer program is called alpha version or alpha release. It is likely to be instable and incomplete, but useful for

demonstration purposes and as proof-of-concept prototype that will be further developed. Beta version or beta release is a release of a computer program that is still under development, but published for testing purposes. The functionality has not been fully tested and major bugs might appear. After a beta release has been thoroughly tested and major bugs have been fixed, the program becomes a stable release. Then only minor bugs that do not harm functionality are allowed.

- Reference site: The quality of a stable release can be proofed by implementing and extensive testing of the software. There is the risk that the system turns out to be inadequate. Thus it is better to see the ERP system in practice and discuss implementation and operational issues with a customer who already uses and knows the system ^[13]. Reference sites listed on the project home page and the availability of documented business cases are the relevant criteria.

13. <http://www.networkcomputing.com/1206/1206ws2.html> viewed 04/11/2013

3. Overview of Open Source ERP Systems

The information provided here is based on listed online resources. First a comparison table will be introduced. Later the projects will be examined in detail.

3.1. Comparison Table

In the functionality section only some modules are listed.

Legends:

√ yes	x no
n/a not available	? unknown
+ above average	~ average
- below average	
(average refers to the other evaluated open source ERP systems)	

Evaluation Criteria		Open Source ERP System		
#	Criteria	SQL Ledger	LX Office	ERP 5
SIZE				
	micro	+	+	
	small	+	+	+
	medium	+	~	+
	large			+
FUNCTIONALITY				
1	Number of Tables	45	36	n/a
	e-Commerce	X	√	√
	Accounting	√	√	√
	MRP			√
	POS			√
	Inventory&Warehouse	√	√	√

Evaluation Criteria		Open Source ERP System		
#	Criteria	SQL Ledger	LX Office	ERP 5
FLEXIBILITY				
1	Customization	~	~	+
2	Flexible upgrades	+	+	?
3	Internationalization	+ (multi-site)	-	+ (multi-site)
4	User friendliness	~	~	+
5	Architecture	3-tier web	3-tier web	3-tier web
6	Scalability	+	~	+
7	Security	~	~	+
8	Interfaces CGI, SOAP CGI XML-RPC,	CGI, SOAP	CGI	XML-RPC, SOAP, XML
9	OS independence	√	√	√
10	DB independence	√	X	object db
11	Programming Language	Perl	Perl	Python

Evaluation Criteria		Open Source ERP System		
#	Criteria	SQL Ledger	LX Office	ERP 5
SUPPORT				
1	Support infrastructure	~	~	+
2	Training	X	X	√
3	Documentation	+	-	-
1	Project structure Company	Company + Partners	Company + Partners	Company + Partners
2	Community activity	+	+	-
3	Transparency	~	~	~
4	Update frequency	+	~	+
5	Other lock-in effects migration tool			
1	Development status	Stable	Stable	Stable
2	Reference site	+	~	+

Evaluation Criteria		Open Source ERP System		
#	Criteria	SQL Ledger	LX Office	ERP 5
OTHER				
1	License	GPL	AL,GPL,LGPL	GPL
2	Online demo	√	√	X
3	Sourceforge hosted	√	√	X
4	CVS access	X	√	√
5	Download checksum	√	X	X
6	Project start	2000	2004	2002

3.2. SQL Ledger

SQL-Ledger is an accounting, commercial and production suite for micro and small enterprises or medium enterprises with basic requirements. For the adaption to national accounting standards programming skills might be necessary. Besides accounting it offers basic functionality for purchasing, sales and inventory management. Functionality and user interface can be tested online. Migration scripts for upgrading from Quickbooks and gnuCash are also available.

SQL Ledger		
#	Sub-Criteria	Description
FLEXIBILITY		
1	Customization	low level code customization, layout templates
2	Flexible upgrades	for simple customizations
3	Internationalization	38 languages, 33 accounting schemes
4	User friendliness	intuitive Web GUI
5	Architecture	3-tier web architecture
6	Scalability	very good
7	Security	coarse grained access control, specified for each user separately
8	Interfaces	CGI, SOAP, LaTeX
9	OS independence	Windows, Linux/Unix, Mac OS X; web client
10	DB independence	PostgreSQL (open source), Oracle, IBM DB2
11	Programming Language	Perl
SUPPORT		
1	Support infrastructure	partner network, mailing list, contracts
2	Training	n/a
3	Documentation	sufficient user and developer documentation
CONTINUITY		
1	Project structure	company driven, international partner network
2	Community activity	150-300 mails a month
4	Transparency	roadmap; no public CVS access and bug tracking
5	Update frequency	regular
6	Other lock-in effects	none
MATURITY		
1	Development status	stable
2	Reference site	many reference sites, worldwide distributed
SQL Ledger features [14][15][16]		

Customization

It can be customized with external Perl modules and used like a framework. The code has to meet defined calling conventions and the main core needs not be modified. External module customizations apply on global or on user level. Sophisticated meta data customization without special programming skills is not provided. The layout for GUI and reports can be changed with templates.

Flexible upgrades

It depends on the degree of customization. When using the external script approach, updates are no problem.

Internationalization

The language preference, comma and date formats can be configured on user level.

Security

Access is controlled on user level. The administrator can grant several functionality groups per module to a user.

Documentation

A 289-page reference manual containing user and some developer information is sold separately. More developer information including ER diagrams, customization and integration samples as well as access to the current development version is provided by a commercial Development Support contract. The project is often reviewed in the media.

Project structure

The system is run by DWS Systems Inc., Canada, and is backed by several partners that provide customization and integration services. The partners are located in Australia, Belgium, Italy, Switzerland, Netherlands, U.K and U.S.A. There are about 100 contributors listed on the homepage. Very few of them contributed Perl code. A fork of this project, driven by two German enterprises, is LX Office.

Transparency

Planned features and their progress are documented. The bug tracking and versioning systems are not public. Developer support is offered on a commercial basis

14. <http://www.sql-ledger.com/cgi-bin/nav.pl?page=feature/index.html&title=Features> viewed 04/15/2013
15. <http://sourceforge.net/projects/sql-ledger> viewed 04/15/2013
16. <http://en.wikipedia.org/wiki/SQL-Ledger> viewed 04/15/2013

3.3. LX Office

This is a fork of SQL Ledger which targets the needs of the German speaking market and offers commercial support to its customers. Only the differences to SQL Ledger are described here. According to the SQL Ledger project leader Dieter Simander, there were no functional reasons for this fork. It offers different functionality in the areas of accounting, reporting, security and special interfaces for Germany (DATEV, preliminary turnover tax return). A CRM and a groupware module are provided together with commercial add-on modules for point of sale and

interfaces to online shop systems. A main difference lies in the handling of taxes. They are defined within the accounting schema as opposed to SQL Ledger, where the taxes depend on the customer, supplier, service or product.

LX Office		
#	Sub-Criteria	Description
FLEXIBILITY		
1	Customization	
2	Flexible upgrades	
3	Internationalization	GUI is translated into German, English, French
4	User friendliness	
5	Architecture	
6	Scalability	good, limited to PostgreSQL
7	Security	
8	Interfaces	additional DATEV interface for Germany
9	OS independence	
10	DB independence	PostgreSQL (open source)
11	Programming Language	Perl, PHP
SUPPORT		
1	Support infrastructure	partner network, mailing list, forum, contracts; (communication is in German)
2	Training	
3	Documentation	Wiki system, less compared to SQL Ledger
CONTINUITY		
1	Project structure	company driven
2	Community activity	~300 messages per month
4	Transparency	roadmap, change log, bug tracking and subversion access
5	Update frequency	regular, less compared to SQL Ledger
6	Other lock-in effects	
MATURITY		
1	Development status	stable
2	Reference site	several sites in the German speaking market

Documentation

A Wiki System is used for developer and user documentation. An ER- Diagram of the ERP module schema is available. Compared to SQL Ledger, less documentation is available.

Project structure

The system is driven by the companies LINET Services and Lx-System, which employ 5 developers. A partner network providing commercial support is available for Germany and Austria.

Community activity

Project communication is in German. The forum and the Sourceforge hosted mailing lists are very active. The community is smaller than in the case of SQL Ledger. Activity of the relevant communication channels:

<u>Communication channel</u>	<u>messages per month</u>
Forum	~150
lx-office-devel	~150
lx-office-erp-users	~30

Transparency

The planned functionality for the next release is documented. A change log describes the introduced functions and bug fixes of a new release. The release dates are missing in the change log. They use a professional public bug tracking system. The version control system (subversion) is public, logs are available as mailing list, and developers are invited to participate on the homepage.

Update frequency

Compared to SQL-Ledger the frequency is slower and less new functionality is introduced. The latest ERP module was released 10 months ago. The current development effort is on CRM and the integration of other systems.

17. <http://www.lx-office.org> viewed 04/20/2013
18. <http://de.wikipedia.org/wiki/Lx-Office> (Translated to English through Google translate)
19. <http://sourceforge.net/projects/lx-office> viewed 04/20/2013

3.4. ERP5

ERP5 was originally developed as apparel industry solution for an organization with more than 300 employees on 5 internationally distributed sites. The special requirements of the apparel industry are mass customization of products and hence the need for handling unstructured and multimedia data. Mass customization means that many variants of a product are possible. The original modules are now generalized as business templates. For demonstration purposes a live CD image can be downloaded which starts the software from a bootable Linux CD. Many modules are provided, some are not production ready:

- Trade provides purchase, sale, order and warehouse management functionality.
- PDM (Product Data Management) allows product definition, variations, categorization, bill of materials (BOM) and a multimedia catalog. There is also a special PDM module for the apparel industry available, which supports all document types required for the handling of fabrics, models, sizes, CAD files, etc.
- MRP (Manufacturing Requirements Planning)

- CRM (Customer Relationship Management)
- Accounting
- HR (Human Resources)
- The E-Commerce online shop supports XML based synchronization with a backend server.
- CMS (Content Management System) to store, index and classify all documents and unstructured data.

ERP5		
#	Sub-Criteria	Description
FLEXIBILITY		
1	Customization	business templates, Zope ^[22] based concepts
2	Flexible upgrades	not documented
3	Internationalization	3 languages; multiple internationally distributed sites support
4	User friendliness	intuitive Web GUI, many GUI elements
5	Architecture	3-tier web architecture, based on Zope
6	Scalability	excellent (clustering, load-balancing, fail over and distribution features)
7	Security	uses the fine grained Zope security model
8	Interfaces	HTTP, XML-RPC, SOAP
9	OS independence	Windows, Linux/Unix, Mac OS X; web client
10	DB independence	Zope object database is used
11	Programming Language	Python
SUPPORT		
1	Support infrastructure	partner network, mailing list, contracts
2	Training	developer conferences
3	Documentation	little documentation is available
CONTINUITY		
1	Project structure	company driven, partner network
2	Community activity	~20 mails a month ^[23]
4	Transparency	subversion access, no roadmap
5	Update frequency	regular
6	Other lock-in effects	none
MATURITY		
1	Development status	stable, some modules are beta
2	Reference site	solutions for apparel industry, aerospace, banking, hospital and government ^[24]

Customization

New modules can be stored and distributed as business templates. Business templates are a concept to simplify the installation, customization and sharing of ERP5 modules. For the customization and implementation of ERP5 Python, Zope and XML skills are needed in addition to detailed ERP5 knowledge. The Zope based customization concepts are explained in the architecture section.

Internationalization

The user interface is translated into French, English and Portuguese (Brazil). Apart from multiple languages, which are provided by the Zope application server, the model of ERP5 supports multiple internationally distributed sites. It is based on a model that allows grouping anything into a category. This feature is very useful to manage a group of companies which belong to a common holding or to manage so-called "factoring" partnerships or networks, that is a partnership where one company asks a few other companies to cooperate and build together a given product.

User friendliness

The user interface is web based and offers many different GUI elements. It uses the page template technology of the application server.

Architecture

ERP5 uses the open source Python based ZOPE (Z Object Publishing Environment) application server and content management framework as its foundation. The web based Zope Management Interface is used as IDE for developing ERP5 modules. The application server also supports multiple languages, provides a security model, object persistence, and remote object access through HTTP, XML-RPC and SOAP.

Interfaces

The Zope object publishing features allow HTTP, XML-RPC and SOAP access to any ERP5 object. Synchronization between distributed sites over insecure connections is supported. Report generation requires Python scripting knowledge or relational data duplication.

DB independence

The platform independent Zope object database is used together with the relational MySQL database. The relational database is used for reporting. The platform independent Zope application server provides capabilities of object oriented databases but does not provide SQL access. The Zope object database allows very fast object access, but its querying limits are covered with relational duplication of reporting data. It can be viewed as kind of Relational

Training

Classic classroom training is not available, but conferences targeted to the Zope community for knowledge transfer to skilled developers are organized.

Project structure

ERP5 is run by the French consulting company Nexedi ^[26] with partners in France, Luxembourg, Germany, Poland, Brazil, Senegal and India ^[27]. The Brazilian research group is also contributing. Their work is documented in Portuguese on the project homepage. A project also supported by Nexedi, is Z3ERP ^[28] which aims to port ERP5 to the Zope3 platform by sustaining ERP5 backward compatibility. Supporting this project could be a good way of learning ERP5 technologies.

Transparency

No actual project roadmap and change log document are provided. Due to the small amount of project documentation and partly the website structure, it is hard to find out the aim of the current development effort. The CVS versioning system is public.

Update frequency

The CVS is very active, but a change log documenting releases and new functionality introduced is not provided. There is a software status document ^[29] which shows the completion percentages of components and modules, but you cannot see which functionality was introduced and when.

Reference site

Besides the initial apparel solution ERP5 is used by a big consulting company for the implementation of an order management system in the aerospace industry. This system integrates with SAP and other back office systems. ERP5 is also used by a large central bank, a hospital, in the automotive industry and by French government authorities [30].

20. <http://www.erp5.org/HowTo> viewed on 04/25/2013
21. <http://en.wikipedia.org/wiki/ERP5> viewed on 04/25/2013
22. Zope (Z Object Persistence Engine) is an open source application server for building content management systems, intranets, portals and custom applications. www.zope.org
23. <http://blog.gmane.org/index.php?prefix=gmane.comp.python.zope.erp5> viewed on 04/25/2013
24. <http://www.erp5.com/industry> viewed on 04/25/2013
25. <http://www.zope.org/Members/infrae/Formulator> viewed on 04/27/2013
26. <http://nexedi.com/> viewed on 04/27/2013
27. <http://www.erp5.com/consulting/erp5-consulting-partners> viewed on 04/27/2013
28. <http://www.z3erp.org> viewed on 04/27/2013
29. <http://www.erp5.org/sections/download/status/view> viewed on 04/27/2013
30. <http://www.erp5.com/industry> viewed on 04/27/2013

4. Conclusion

Flexible ERP systems allow unique, adaptable processes that support a diversification strategy, where IT is a source for differentiation from competitors. The availability of the source code and its use as application development framework can lead to a higher degree of flexibility and allows a close functional fit of the system to business processes. Company-internal system knowledge allows continuous process improvement and quick reaction to changing requirements. The main reasons for choosing open source ERP systems are cost, flexibility, vendor independence and company-internal knowledge generation. The project structure of the selected ERP systems varies from company-driven to community-driven.

For the purpose of a coarse, subjective summary, some system properties are pointed out: SQL Ledger has accounting focus and is proved internationally. It is best used out-of-the-box or as accounting module in an integrated solution. The SQL Ledger fork LX Office should only be evaluated for the German speaking market. ERP5 has very flexible architectures, following a purely object oriented approach. Further work needs to be done in the areas of functional features documentation and comparison. Some ERP systems still have to prove the flexibility concepts in practice with documented business cases.