

A Comparison of Open Source ERP Systems

Institute of Information Systems and Operations,
Department of Business Management and Information Systems
Vienna University of Economics and Business Administration

Institute of Software Technology and Interactive Systems,
Business Informatics Group
Vienna University of Technology

Assessor: Univ.Prof. Dr. Dr. h.c. Hans Robert Hansen
Univ.Prof. Dipl.-Ing. Mag. Dr. Gerti Kappel

Author: Thomas Herzog

Vienna, June 2006

WIRTSCHAFTSUNIVERSITÄT WIEN

DIPLOMARBEIT

Titel der Diplomarbeit:

A Comparison of Open Source ERP Systems

Verfasserin/Verfasser: Thomas Herzog

Matrikel-Nr.: 9307656

Studienrichtung: Wirtschaftsinformatik (Universität Wien)

Beurteilerin/Beurteiler: Univ.Prof. Dr. Dr. h.c. Hans Robert Hansen

Ich versichere:

dass ich die Diplomarbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

dass ich dieses Diplomarbeitsthema bisher weder im In- noch im Ausland (einer Beurteilerin/ einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Datum

Unterschrift

Contents

1 Introduction.....	7
1.1 Abstract.....	7
1.2 Definition of ERP System.....	7
1.3 Definition of Open Source.....	8
1.4 Definition of Small and Medium Enterprises.....	8
1.5 Structure.....	9
2 Methods to Implement an ERP System.....	10
2.1 Select an ERP Package.....	11
2.1.1 Re- Engineering.....	11
2.1.2 Customization.....	11
2.2 Develop ERP Software Individually.....	12
2.3 Integrate Best of Breed Choices.....	13
2.4 Advantages of Using a Flexible ERP Package.....	14
3 Evaluation Criteria.....	18
3.1 Functional Fit.....	19
3.2 Flexibility.....	19
3.3 Support.....	26
3.4 Continuity.....	27
3.5 Maturity.....	32
4 Overview of Open Source ERP Systems	33
4.1 Comparison Table.....	33
4.2 SQL Ledger.....	38
4.3 LX Office.....	41
4.4 Tiny ERP.....	44
4.5 GNU Enterprise (GNUe).....	47
4.6 ERP5.....	51
4.7 Opentaps - OfBiz.....	57
4.8 Compiere.....	63
5 Conclusion.....	71
Appendix A: Glossary.....	72

Appendix B: Bibliography.....	76
Appendix C: Surveys.....	83
C.1 Decision Making Criteria: A Survey.....	83
C.2 Key Advantages of Open Source ERP: An Online Poll.....	84
Appendix D: Challenges for Open Source ERP Systems.....	86
Appendix E: Model Driven Architecture MDA.....	88
Appendix F: Other Open Source ERP Systems.....	89
Appendix G: GNU Enterprise Developer Tools.....	92

Illustration Index

Figure 1: Evaluation criteria composition.....	9
Figure 2: Modules in an ERP based integration approach.....	10
Figure 3: Stages of an ERP project taken from Christian Nebenführ.....	11
Figure 4: A flexible ERP system can also be used as development framework, saving integration costs by focusing on one technology.....	15
Figure 5: Meta data customization concept.....	16
Figure 6: Evaluation criteria.....	18
Figure 7: Multiple site support.....	21
Figure 8: N-tier architecture.....	22
Figure 9: Closed source and open source vendor strategies.....	28
Figure 10: GNU Enterprise architecture.....	49
Figure 11: Trade module screen shot: multi-organization, multi-variation analytical stock management.....	52
figure 12: Opentaps/OfBiz service oriented architecture.....	60
Figure 13: Compiere Application Dictionary GUI elements, taken from [Lomb05]	65
Figure 14: Compiere GUI, taken from [Pink05].....	66
Figure 15: Compiere architecture.....	67
Figure 16: Survey form.....	84
figure 17: Survey result.....	85
Figure 18: Class diagram of address.gcd GNUe Class Definition.....	96
Figure 19: Initial form generated from "address.gcd " GNUe Class Definition...98	
Figure 20: Refined form defined in "address-En.gld" GNUe Language Definition	99

Index of Tables

Table 1: Definition of small and medium enterprises of the EU.....	8
Table 2: ERP decision making criteria of SMEs.....	83
Table 3: GNUe Developer Tools and Versions.....	92
Table 4: GNUe trigger levels and conditions.....	93
Table 5: GNUe Application Server Abstraction Layers.....	95
Table 6: GNUe Class Definition terms.....	96

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 adaption of the software to changing processes and needs. In this work 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 is an attempt to create an integrated product that manages the majority of operations in a company. What is different about ERP systems, is that they integrate across functions to create a single, unified system rather than a group of separate, insular applications” [\[Alsha04, p. 454\]](#). To further distinguish ERP systems from general application frameworks and other standard software, accounting functionality is required.

1.3 Definition of Open Source

In this work the open source definition of the Open Source Initiative (OSI)¹ is used. According to OSI this means that software must comply to the following conditions (shorted):

1. **Free redistribution**, including selling or using as component without fee.
2. The **Source code** must be **available** in readable form.
3. **Derived work** must be **allowed** under the same license conditions.
4. Integrity of the author's source code (licenses may require that modifications are redistributed only as patches).
5. **No discrimination** against persons or groups.
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. Licence 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 can not be abandoned easily. Furthermore you have the possibility to adapt the software to your needs. For details please see chapter "[3.4 Continuity](#)".

1.4 Definition of Small and Medium Enterprises

In this paper the definition of the European Union is used²:

Enterprise category	Headcount	Turnover	or	Balance sheet total
medium-sized	< 250	≤ € 50 million		≤ € 43 million
small	< 50	≤ € 10 million		≤ € 10 million
micro	< 10	≤ € 2 million		≤ € 2 million

Table 1: Definition of small and medium enterprises of the EU

¹ <http://www.opensource.org/>

² http://europa.eu.int/comm/enterprise/enterprise_policy/sme_definition/index_en.htm, as viewed on 2005-09-12

1.5 Structure

This is a short description on how the next chapters are related.

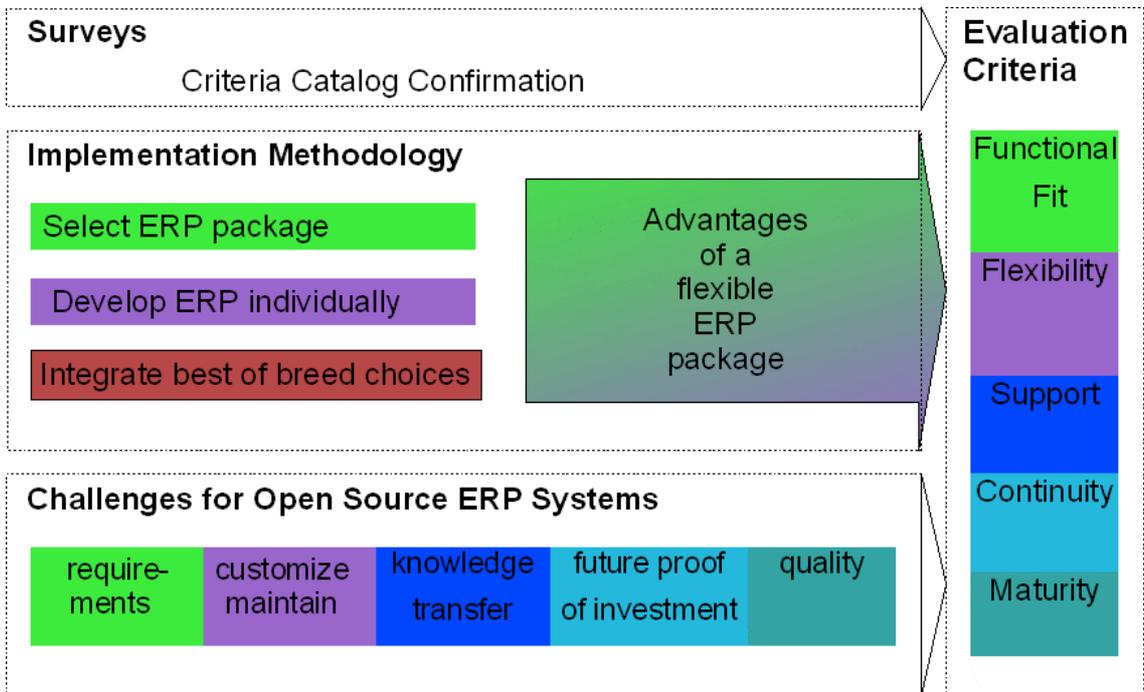


Figure 1: Evaluation criteria composition

Chapter “[2 Methods to Implement an ERP System](#)” shortly discusses different implementation methodologies and works out the advantages of flexible systems. It addresses general technical requirements for ERP systems. Chapter “[3 Evaluation Criteria](#)” develops a criteria catalog based on the previous chapter. The appendix chapters “[8.2 Challenges for Open Source ERP Systems](#)” and “[8.3 Surveys](#)” provide further input for this criteria catalog. Chapter “[4 Overview of Open Source ERP Systems](#)” applies the evaluation criteria to selected open source ERP systems and classifies them. Chapter “[5 Conclusion](#)” recapitulates this work.

2 Methods to Implement an ERP System

A small or medium Enterprise (SME) has several possibilities to implement an ERP System as described in the following chapters:

- Chapter [2.1 Select an ERP Package](#) and leverage its capabilities.
- Chapter [2.2 Develop ERP Software Individually](#).
- Chapter [2.3 Integrate Best of Breed Choices](#) of ERP packages, special application software and individually developed modules.

Starting with an ERP package there are many ways to support your processes with IT:

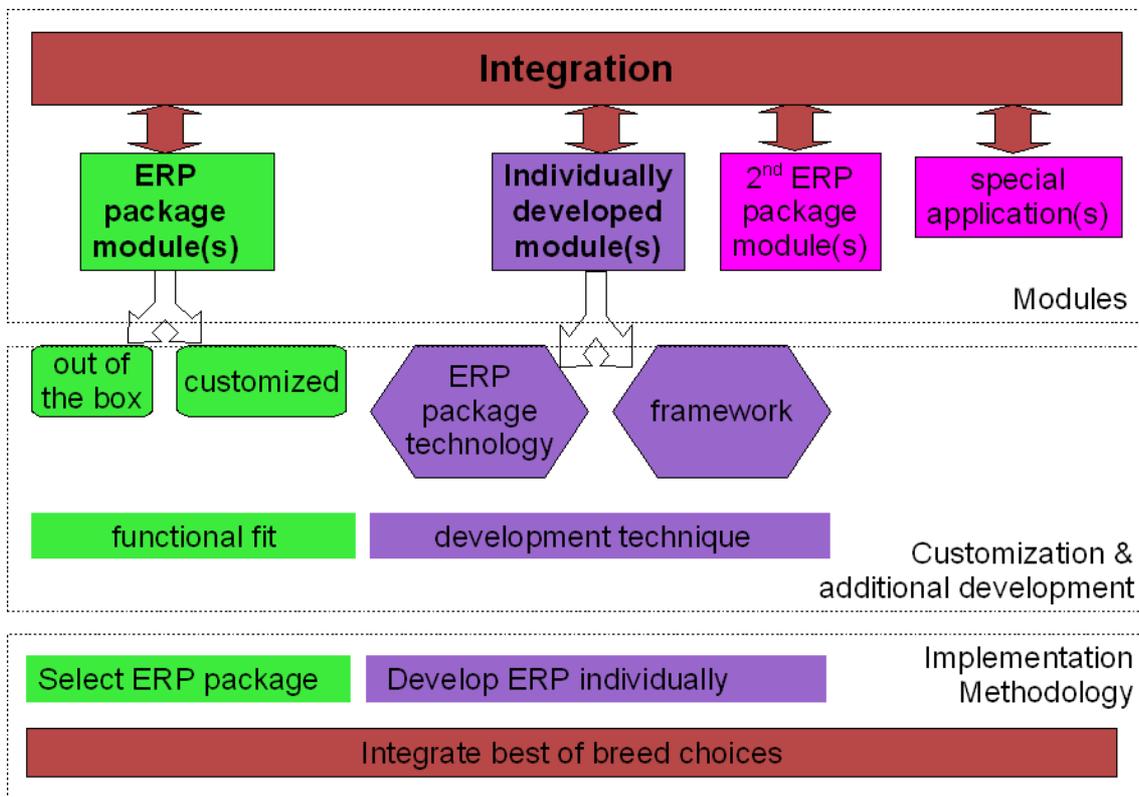


Figure 2: Modules in an ERP based integration approach

This work focuses on open source ERP packages that can be easily customized and used as development framework for custom modules. This is a mixture of the approaches "[2.1 Select an ERP Package](#)" and "[2.2 Develop ERP Software Individually](#)". It is a compromise between flexibility, adaption to processes and saving integration costs. Chapter "[2.4 Advantages of Using a Flexible ERP System](#)" describes the reasons for this choice in detail.

2.1 Select an ERP Package

There are several stages in the selection process of an ERP package³.

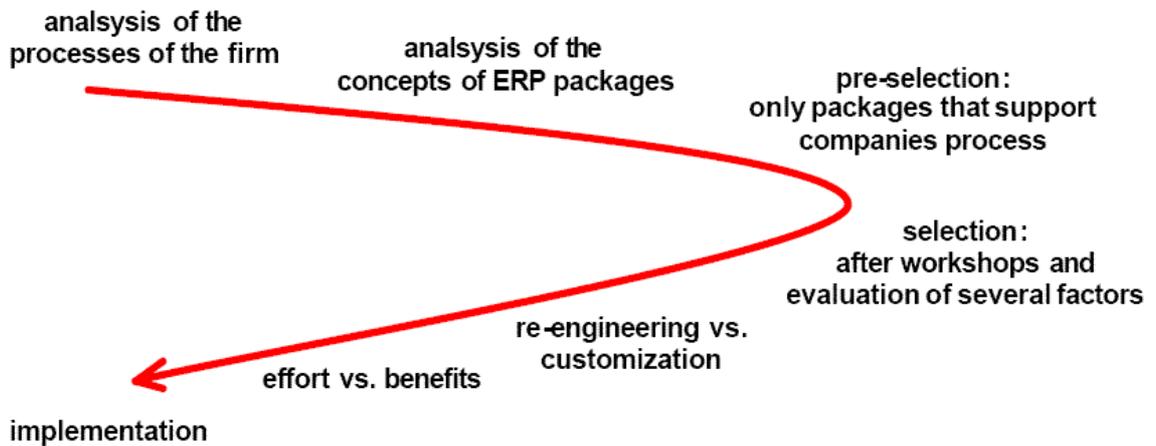


Figure 3: Stages of an ERP project taken from Christian Nebenführ

2.1.1 Re- Engineering

It is especially suitable for small and cost-sensitive enterprises to use an ERP system **out of the box**. Basically a company can adapt its processes to the ERP system, which generally uses best-practice processes, and use it out of the box. This is called the re-engineering approach (=adaption to ERP processes in this context). “Anyone who has attempted to change business procedures even slightly, knows that this is painful” [Vittie01]. If the current processes needs to be changed dramatically, there might occur staff resistance and change management is needed. “ERP implementation is predominantly a change management project” [Kay98 as cited in Telt00].

2.1.2 Customization

Out-of-the-box ERP systems should be used in areas that are not the core competency of an enterprise. In core competency areas your unique processes usually are a competitive advantage and better than best-practice processes

³ The figure is taken from the slides of unit 1 of Christian Nebenführ's course on Enterprise Resource Planning, WU Wien, SS2005

provided by ERP systems. For example, UPS (United Parcel Services), the shipping company, should not use an out-of-the-box ERP to track package inventory, shipments, logistics and routing as this is the area they know best themselves⁴. In this case the ERP package should be customized to adapt to those unique business processes. “Customization is a difficult task” [[Dave98, as cited in Alsha04, p.455](#)]. In addition to the initial customization costs consequential costs are introduced, as the customization might need to be applied again after upgrades of the base ERP package. For a high level discussion of customization see [[Song01](#)] and [[Harr00](#)]. It depends on the flexibility of the chosen software if and to what extent customization is appropriate.

2.2 Develop ERP Software Individually

For specific requirements not covered by the ERP system individual modules are developed and integrated. The development basis is either the technology stack of your chosen ERP system or your favorite framework. A framework is a software library that makes up reusable design for a specific class of software [[Gamma95 pp.26-28](#)]. For productivity reasons ERP software is generally not developed from scratch. A framework, either open source, closed source or self developed, is used.

Every software that is designed for reuse and change can also be used as a framework in its intended application area, provided you have the source code or good interface documentation. This makes open source ERP systems interesting from a developer's and customization perspective, because you have the opportunity to adapt the framework and easily share improvements or develop add-on modules. This should lead to user driven adaption of the system to market needs. There is a smooth transition between flexible ERP packages that can be customized and frameworks, especially in the area of open source ERP systems. For a generative approach to software development please have a look at the appendix chapter “[8.1 Model Driven Architecture MDA](#)”.

⁴ <http://www.tc.umn.edu/~hause011/article/ERPReasons.html>, viewed on 2005-11-17, Steven Hauser: Reasons to get an ERP

2.3 Integrate Best of Breed Choices

This chapter briefly describes the broad concept of software integration in the context of ERP. For an overview please have a look at figure 2 “Modules in an ERP based integration approach“ on page 10.

Basically you have endless choices in supporting your processes with IT. Here the use of an ERP system and the integration of other systems constitutes the basis for further thoughts. The smaller your company is and the more budget restrictions you have, the more you should keep to standard approaches like out-of-the box solutions and easily maintainable customizations. “Executives in the industry say that generic products require longer implementations, more workarounds, and increase the complication of add-on solutions” [\[Rama00as cited in Alsha04, p. 455\]](#).

You can also use the best modules of different ERP systems. To save integration costs, this approach is only recommended for medium to large organizations. A successful case study in mixing SAP R/3 suite with Oracle Applications 11i HRMS to save customization cost is described in [\[Alsha04\]](#). One should consider carefully to choose a flexible Enterprise Application Integration (EAI) middleware supporting all systems. Mixing of ERP systems requires open, standardized interfaces and introduces subsequent costs due to increased complexity. On transaction level most systems do not support distributed transactions, which take into account multiple points of failure and allow transactions to be rolled back over all affected systems in case of an error. For reporting issues the distributed data should be integrated into a common reporting system, for example a data warehouse. Data quality may suffer when data is partly duplicated in multiple systems and distributed transactions are not supported. It is also more difficult to estimate change impacts for other systems when upgrading or adapting one system. User account and user rights management is also more complex. Some users might have to deal with multiple applications concurrently.

In addition to mixing ERP systems the integration of individually developed modules and software for special applications should also be considered here. It depends to a large degree on the specific company, how many systems should be used and to what extent they should be customized or individually

developed. But it is generally a good idea to reduce complexity. You have to find the balance between the number of systems, the degree of customization, adaption to processes, complexity, cost and maintainability.

If flexibility is your priority, you have to look carefully at the technical concepts of the chosen ERP system or framework. Mixing systems can decrease flexibility as changes might affect several systems, especially if a business process is covered by several systems. For small and medium enterprises integration costs can be too high.

2.4 Advantages of Using a Flexible ERP Package

After giving an overview of different technical approaches for the implementation of ERP system, I want to sum up the advantages of a flexible ERP package, that can easily be customized and be used as development framework for creating custom add-on modules.

As described above the integration of different software pieces is complex and cost intensive in implementation as well as in operation. Developing the whole ERP system means reinventing the wheel to some part. Out-of-the-box solutions do not address the need for flexibility and adaption to unique processes.

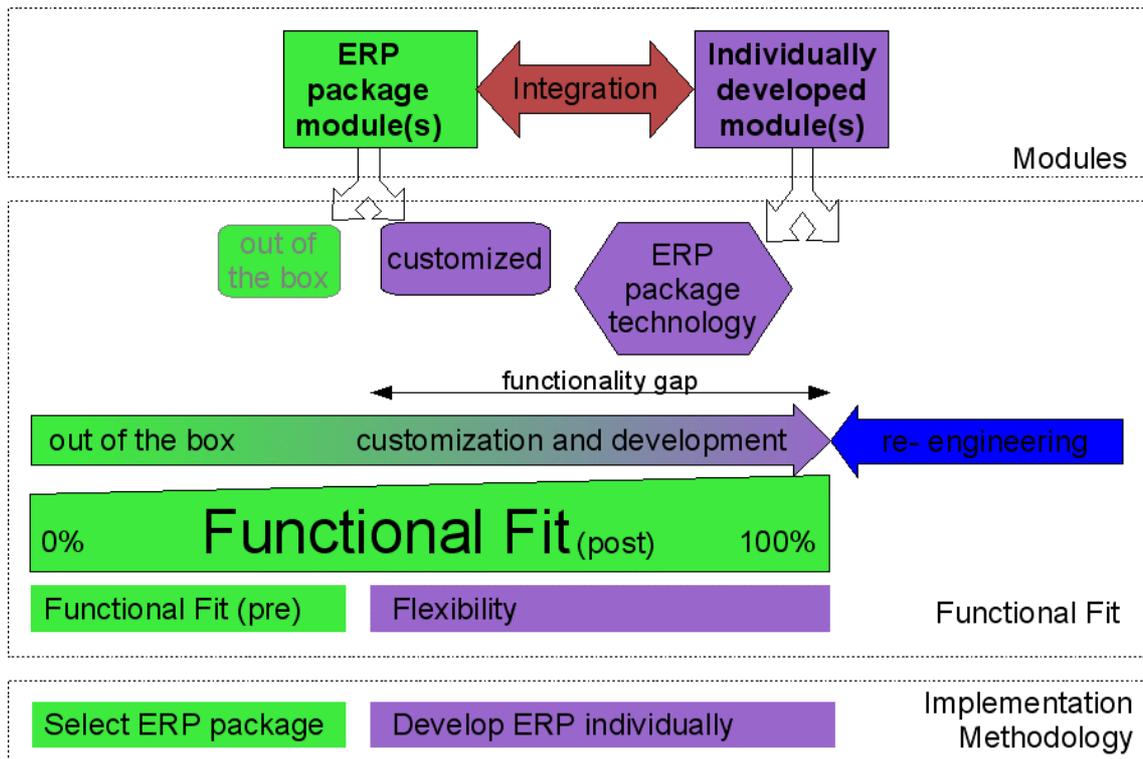


Figure 4: A flexible ERP system can also be used as development framework, saving integration costs by focusing on one technology

It is important that the chosen ERP package has a good **functional fit** with the company's business processes out of the box as this saves customization costs and shortens implementation time [\[based on Vittie01\]](#). But if you want close fit and the ability of continuous process improvement, then **flexibility** and customization are a must. A close fit of the system to the company's processes and needs, also leads to generally lower operational costs.

Flexibility is the reason why the technological concepts and the software architecture of the chosen ERP system are very important. This flexibility allows to **focus on one system** or fewer systems and technologies. Apart from the customization opportunities flexibility here also means that you can develop add-on modules and interfaces based on the technology of the chosen ERP package. This means saving integration costs as functionality of other (e.g. 2nd) ERP packages and special applications can be achieved through customization and individual development. This focus on know how also saves administration and training costs. An appropriate ERP package supports many processes out of the box and additionally allows flexibility.

A flexible system allows unique, adaptable processes and supports a

diversification strategy, in which IT is a resource for differentiation from competitors. The company's internal knowledge about the system further strengthens the flexibility aspect.

Advantages of a Meta Data Customization Approach

Meta data is data about data. A flexible ERP System allows high-level customization through meta data editing and low-level customization through coding. The advantage of high-level customization is that you do not need to know the programming language. This is possible through the abstraction of parts of function, structure and behavior from the ERP systems into meta data [based on Mill03, p. A-2]. This means the ERP system should provide a core or service engine that reads the meta data to generate forms, data structures and workflow between forms. In this case meta data defines the model of the application. The ERP engine executes this model.

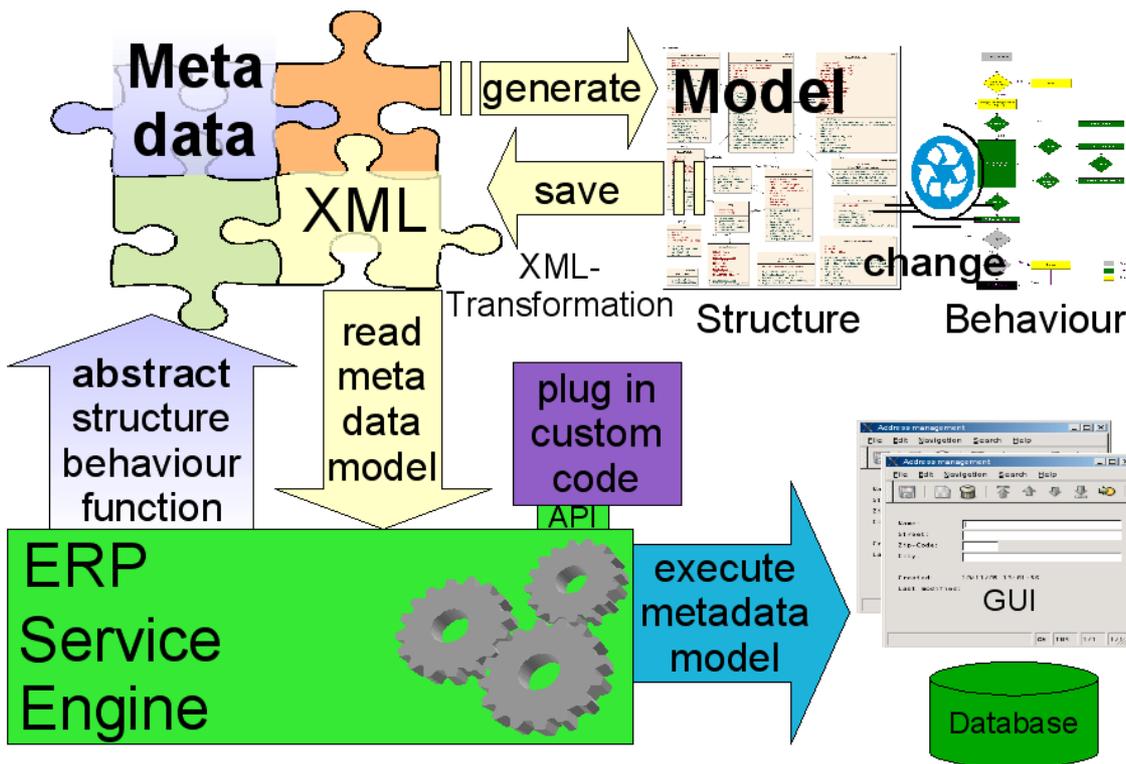


Figure 5: Meta data customization concept

Meta data can be stored in files or in the database (common names are meta data repository or application dictionary). XML (eXtensible Markup Language) is a widely used format for storing meta data. The XML format has the advantage that its structure can be verified and XML data can easily be transformed.

The next higher level of ERP customization is doing most customization (meta data editing) graphically and supported by tools. For example you can use common UML (Unified Modeling Language) tools to define a model and transform this model to ERP system meta data (ERP domain specific model) so that it can be run by the ERP engine. The round trip ERP customization vision is to generate UML models from ERP meta data, edit them with the help of an UML tool and transform customizations back to meta data executed by an ERP service engine.

In addition to the abstraction of structure and behavior, another advantage of the meta data customization approach is that meta data can be generated with common software developer tools and can be used to automatically generate documentations.

3 Evaluation Criteria

The criteria introduced now are hierarchically structured, and can be used as starting point for personal adaption. Most criteria are not measurable, but if you need a formal evaluation process, you can still apply subjective weighing like suggested in [\[Chun04\]](#). 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 in chapter "[4 Overview of Open Source ERP Systems](#)".

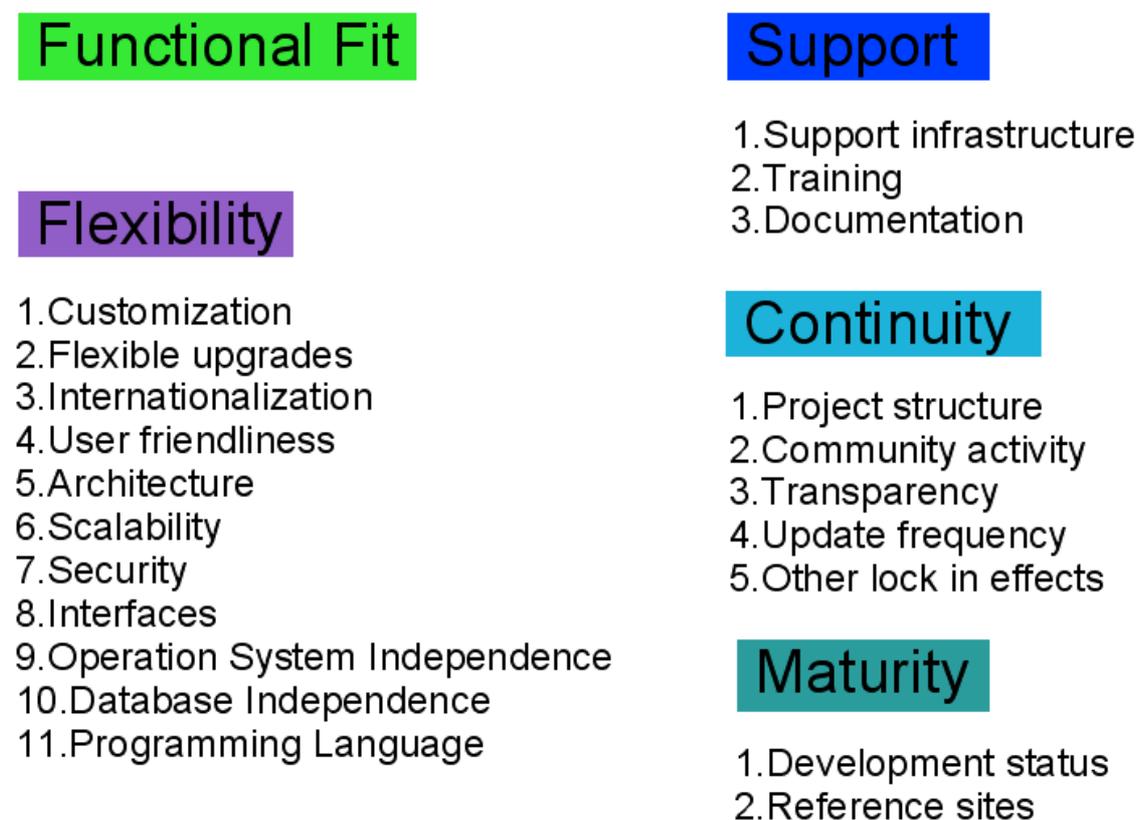


Figure 6: Evaluation criteria

Important for ERP systems are the **total costs** which include cost of consulting, analysis, license, implementation, customization, maintenance, training, integration, support, upgrades and continuous adaption to processes [\[based on Vittie01, Stoy05\]](#). All evaluation criteria have cost influence. As shown on figure 4 on page 15, functional fit out of the box shows 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.

Open source licensing issues are not addressed here and thus do not form part of the criteria catalog. For a detailed discussion see [\[Laur04\]](#).

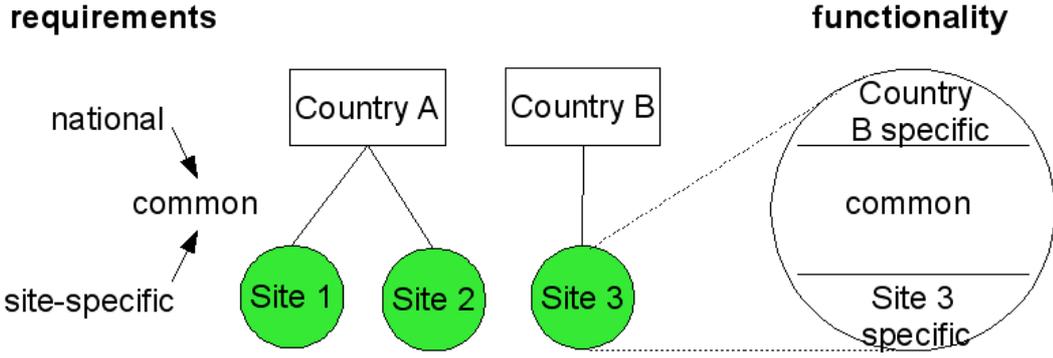
3.1 Functional Fit

Functional fit is used for the company's perspective, functionality for the ERP system's perspective. The term functional fit [\[Vittie01\]](#) is preferred over functionality. Fit implicitly means that functionality you do not need is useless. It is the degree the chosen ERP system fits to your business processes out of the box [\[Vittie01\]](#). The higher the fit is, the lower are the customization and implementation costs. 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. Thus functionality is not the focus of this paper. 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.

3.2 Flexibility

After the description of the advantages of flexible ERP systems in chapter "[2.4 Advantages of Using a Flexible ERP Package](#)", the criteria for flexibility will be defined now. Flexibility allows to bridge 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**:

FLEXIBILITY Criteria	
#	Sub-Criteria
1	Customization
<p>Depending on the degree of customization needed and the skill levels of the ERP system experts, different levels for customization should be provided. Thus the customization effort can be distributed among a larger number of participants [based on Fisch04, p.37].</p> <p>High level customization through meta data editing. In this context that means that the system is customizable by editing easily readable and understandable data, instead of doing low-level coding in a programming language. A business matter expert should be able to customize the system without having detailed programming knowledge. The goal is to reduce the learning burden while providing powerful facilities to address a wide range of problems [Fisch04, p.34]. Powerful high-level customization possibilities constitute an important productivity factor for shortening the implementation time and allow the continuous adaption to processes.</p> <p>Low level customization (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. Here the ERP system defines the software architecture and allows the addition of custom operations. This custom code must meet the frameworks Application Programming Interface (API) specifications. Coding is called low-level customization.</p> <p>The next lower customization/development level would be the adaption or extension of the framework, for example to provide additional high-level customization facilities.</p>	
2	Flexible upgrades
<p>As customizations are defined as meta data and custom code, which must meet the conventions of the framework, it is possible to provide an upgrade procedure without impact on customizations. Due to this clear separation and the service engine approach, upgrades of the core system do not induce new adaptations of customizations.</p>	

FLEXIBILITY Criteria	
#	Sub-Criteria
3	Internationalization
<p>The system supports multiple languages, multiple accounting- and costing schemes, as well as multiple sites [Vittie01, Stoy05]. The simplest form of internationalization is providing translations for the user interface and local accounting schemes. The language is selected on user level. You can differentiate between simple translation of static graphical user interface (GUI) parts (e.g. menus, field labels), translation of dynamic GUI parts (e.g. workflow states), and content (e.g. product descriptions). Legal national requirements especially in accounting often demand custom workflow or business logic. This means that good customization possibilities (flexibility criteria 1. and 2.) are a precondition for internationalization.</p> <p>It is very important for open source ERP systems, even simple ones intended to be used solely locally, to provide the flexibility to support many nations in order to gain a broader international user base and reduce the risk of project forking due to limited internationalization support. Project forking means splitting the code base, which leads to two separated projects and thus community fragmentation and less collaboration.</p> <p>Multiple site support implies serving several internationally distributed sites by using different accounting and costing schemes within one ERP system⁵.</p> <p>requirements</p>  <p>functionality</p>	
<p>Figure 7: Multiple site support</p> <p>To manage site and national specific functionality, the ERP system should allow the generalization of common functionality. The sites can be served with one central system having reliable connectivity to all involved sites or with distributed and synchronized systems.</p>	

⁵ For a possible solution to this problem see <http://www.compiere.org/technology/multi/org.html>, as viewed on 2006-01-07

FLEXIBILITY Criteria

#	Sub-Criteria
---	--------------

4	User friendliness
---	-------------------

The user interface should be designed according to the information needed for a task. A simple task shall not require navigating through many screens. It is part of customization to adapt the ERP system to processes. For routine work keyboard shortcuts need to be provided. Some ERP systems only support few graphical user interface (GUI) elements. User friendliness has a bearing on customization possibilities, user acceptance, training costs and operations cost.

5	Architecture
---	--------------

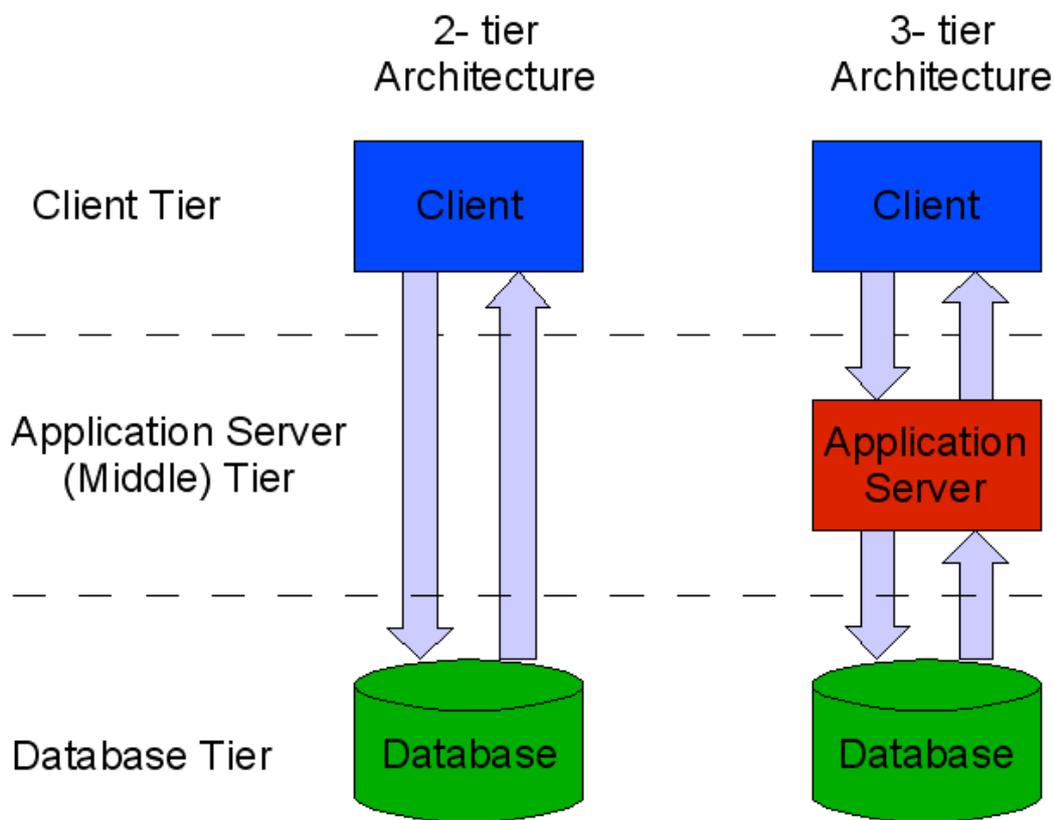


Figure 8: N-tier architecture

Important for most flexibility factors is the chosen architecture. The open source solutions have 2-tier or 3-tier architectures. 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 a 3-tier architecture the client is just responsible for GUI and simple data validation. All the logic is

FLEXIBILITY Criteria	
#	<i>Sub-Criteria</i>
	<p>capsuled in the application server. The database is responsible for storing persistent data. Usually in the case of 3-tier architecture the “thin” client is a web browser, and the application server is a web application server. Some architectures leverage the functionality of a “standard” general purpose application server (J2EE application server for Java, Zope for Python), others use a proprietary server or a basic web server. Advanced architectures allow many different or nearly any client types to be run against the application server. These clients can be web based, terminal based as well as rich GUI based, running on a mobile device or a personal computer. This is possible due to a multi-layer design. The middle tier is further split horizontally into data, business logic and presentation layer. So only the presentation layer needs to be changed to support various client types. Additional flexibility is possible by vertically splitting the system into services which are connected with flexible workflows. For integration with external system these services can be published as web services. Workflow is the automation of a business process, during which information is passed along the system according to a set of rules⁶. It can be specified in standardized XML Process Definition Language (XPDL)⁷. This corresponds to the behavior model in figure 5 “Meta data customization concept“ on page 16.</p>
6	<p>Scalability</p> <p>“A system that does not scale to support all your future users is a disaster waiting to happen” [Vittie01]. 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.</p>
7	<p>Security</p> <p>User or better role-based security mechanisms allow the definition of different levels of access rights. 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.</p>

⁶ This definition is based on http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf- page 8, emphasizing its technical aspects.

⁷ <http://www.wfmc.org/standards/XPDL.htm>

FLEXIBILITY Criteria	
#	<i>Sub-Criteria</i>
8	Interfaces
<p>An interface is a communication boundary of an ERP System. The user interface was discussed under flexibility criteria 4. Other interfaces are described here. They 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. For a case study on this topic please see [Alsha04].</p> <p>Sending and receiving emails and handling of email attachments is important for CRM related communication and user notification. On the client side attaching files to ERP data like CAD documents, receipt scans or product pictures should be supported. Other often manually used interfaces are office integration, CSV export and import and general reporting.</p> <p>Local interfaces to public authorities and banks will be addressed when they form a part of the system. In general, they are provided by local support partners.</p>	
9	OS independence
<p>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.</p>	

FLEXIBILITY Criteria	
#	Sub-Criteria
10	DB independence
<p>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 prefer open source databases⁸.</p> <p>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.</p>	
11	Programming Language
<p>What is open source without knowing the language of the source? The language can be a criterion to leverage available skills for low level customization.</p> <p>The programming languages of the selected ERP Systems are open source scripting languages (Python⁹, Perl¹⁰) and Java¹¹. Python is known for its easily readable, concise syntax and its built-in refactoring¹² capabilities. Perl is widely used, but requires more developer discipline to get a serviceable code. Java has strong industry support and many software engineering tools are available. For a practical comparison between Python and Perl based on examples see [Raym00]. [Ferg04] compares the language concepts of Python and Java, which are both object oriented languages.</p> <p>Counting lines of code is a bad indicator for functionality for the following reasons: High-level scripting languages need fewer lines of code. Flexible meta data based design approaches also need fewer lines of code and meta data can be defined in the program code and/or externally.</p>	

⁸ <http://www.prweb.com/releases/2005/11/prweb308560.php> , viewed on 2005-12-2; sample: 1760, there is no description how the survey was conducted.

¹² Refactoring is the reorganization of the source code to improve internal consistency and clarity.

¹¹ <http://java.sun.com/>, Java is not open source

¹⁰ <http://www.perl.org/>

⁹ <http://www.python.org>

3.3 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 Criteria	
#	<i>Sub-Criteria</i>
1	Support infrastructure
	Reliable and responsive support is important [Vittie01 , Stoy05]. 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.
2	Training
	Here the quality and frequency of user and technical trainings or the organization of regular conferences are of certain importance.
3	Documentation
	Completeness and up-to-dateness of the user and developer documentation are needed. Many projects use a Wiki content management system for collaborative documentation authoring and maintenance.

3.4 Continuity

Project continuity ensures that your ERP system expenses are a sustained investment. When you focus on one system, as suggested in Chapter [2.4 Advantages of using a flexible ERP package](#), you also run the risk that the system will be no longer endorsed. This problem can be diminished with:

Independence of vendor strategy (no vendor lock-in)

For a detailed discussion on IT-based vendor lock-in please see [\[Mata95 p.489\]](#). He discusses the create-capture-keep paradigm (vendor lock-in) as a possible IT-based source for sustained competitive advantage of an IT service provider. Here possible advantages of open source ERP systems regarding vendor lock-in will be briefly discussed.

The consolidation in the ERP market and the rapid technology changes might force customers to follow the vendor product strategy and thus the possible up-selling or costly migration proposals of ERP vendors. There is a risk of the system being discontinued due to vendor take-over, vendor bankruptcy or a technology change. Open source software reduces the investment risk as development can not be abandoned easily. You have the opportunity to maintain the software yourself, but to gain scale advantages it is important to have the system backed by companies and an active community that keep the ERP package up to date. For painless upgrades of a heavily customized ERP system a flexible software design is needed. On the other hand, when the project is driven by one company only, there is a risk of new versions being published under a different license. Even in this case the open source company has less power for undertaking an “unfriendly” strategy change, because there is still the risk of possible project forking, when the product strategy is changed in a way customers do not like. [\[Gold05\]](#) discusses the importance of the user community and possible up-selling strategies in open source projects. Open source companies are highly dependent on a vibrant user community as only a small share of the users is interested in buying additional services. Up-selling strategies like shielding developers from the community and holding back features of open source versions together with a strong focus on selling a commercial version may harm community growth. A small community in turn

hinders selling services like additional documentation, training, consulting and partner certification.

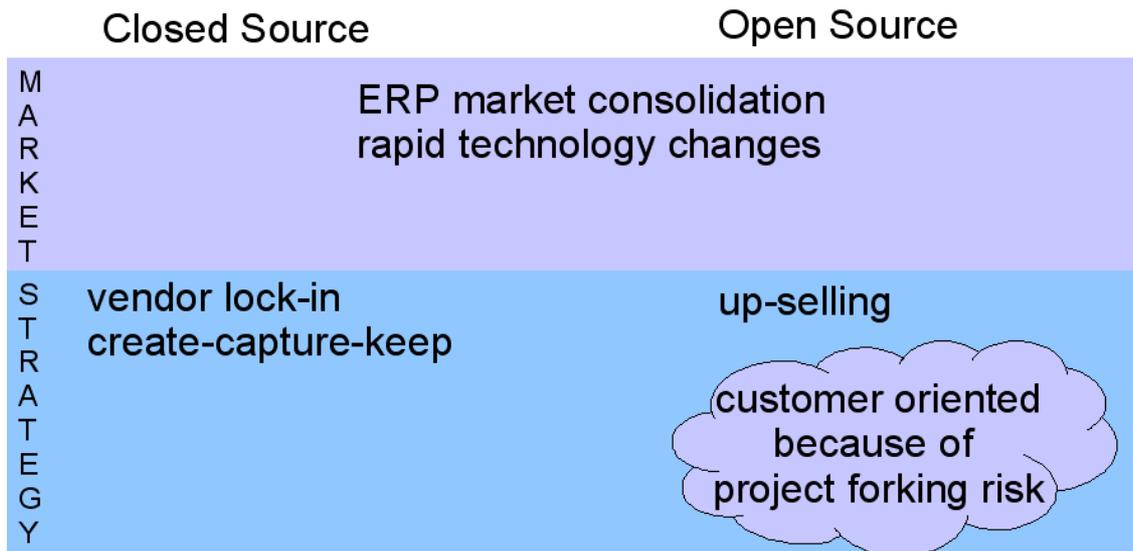


Figure 9: Closed source and open source vendor strategies

Community participation and size

[Chan04 pp.11-20] classifies different online community members and community participation models. Applied to open source ERP systems there are 4 community member categories: Virtual users are active in forums, beta testers provide bug descriptions, content creators create documentation and requirement specifications and developers enhance the system.

The larger and more active the community of an ERP project is, the lower is the risk that the project gets abandoned. There is no way counting the number of customers using open source ERP systems as an indicator for continuity, because customers need not give feedback to the project leaders.

If the project is hosted by <http://www.sourceforge.net>, a widely used platform for open source projects, then you can use the provided statistics as an indicator. For some statistical findings on open source ERP projects hosted by Sourceforge, regarding the number of developers, life- span, CVS (concurrent versioning system) activities and the number of downloads, see [Hyos05]. [Matus03 pp.27-31] describes sourceforge project characteristics and measures. For a detailed general analysis on mining Sourceforge projects see [Howi05]. The usability of Sourceforge statistics without detailed project analysis is questionable. Some reasons are: Projects do not use or just partly use the services offered. The source forge website is not maintained. For example, the

developers switch the versioning system from CVS offered by Sourceforge to the newer subversion technology without removing the old CVS data. Projects claim to be an ERP System but are not, even when taking the definition loosely. The download is available on several locations or the software is also available on Linux distributions.

The **number of messages** in mailing lists or forums is a measurable indicator. More important for the estimation of continuity is the content of the user and developer communication. As a side effect you also get a hint on the maturity of the project. Another indicator for continuity is the product itself. A good and usable system will hardly be dropped by the community.

CONTINUITY Criteria	
#	<i>Sub-Criteria</i>
1	Project structure
<p>The evaluated projects are company or community driven.</p> <p>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. The business model and the size of the involved companies are continuity indicators. Financial information about the involved companies is not available for the evaluated projects. A company-driven project can have source code contributions from the user community and partners.</p> <p>Community driven means that the development is cooperative and there is no single company solely responsible. [Matus03 p. 24] describes the organizational structure of community-driven projects. The degree and areas of community contribution are also listed here.</p>	
2	Community activity
<p>As the community size is not measurable, its communication activity is measurable in certain communication channels. Here the number of messages in forums and mailing lists is used. Besides quantity, qualified answers and response times are important. Documentation activity like web site creation and wiki entries form a part of support/documentation.</p>	

CONTINUITY Criteria	
#	<i>Sub-Criteria</i>
3	Transparency
<p>This is about entry barriers for developers and possibilities for the community to contribute to and influence the process, the quality of project management as well as the documentation of the development process. A reliable, documented roadmap helps to estimate the current focus and future direction of the project. A reason why some projects do not have a detailed roadmap with time schedules is that new functionality needs to be sponsored [Chalif04a, Chalif04b also cited in Mend04]. As developers are professionals, a customer needs to pay them for implementing certain functionality, except it is seen as essential for the project. A public issue tracking system informs about bug details and the time it takes to fix them, planned features and their prioritization.</p> <p>Especially when the project is company driven, it is interesting to know if and in what form active development participation is intended. The degree of community involvement in the development process constitutes another factor for vendor independence, in this case independence from the open source company or project leaders. With access to the code versioning system together with technical documentation you can estimate the chances for active participation in the development process. The source code needs to be readable and documented. The source code versioning system needs to have a detailed change log and understandable tagging of forks and release versions, to be usable. The documentation of development tools and build procedures helps new developers to get quickly involved into the project. Code contribution management is especially important for tightly coupled ERP systems aimed to cover widely varying requirements. Apart from the control of code quality and its fit into the software architecture it ensures that new functionality is of general community use and not too customer specific [Chalif04b].</p>	

CONTINUITY Criteria	
#	<i>Sub-Criteria</i>
4	Update frequency
<p>The continuous introduction of new functionality and the fixing of bugs are a proof of development continuity. 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.</p>	
5	Other lock-in effects
<p>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.</p>	

3.5 Maturity

[\[Gold04\]](#) introduces the Open Source Maturity Model as a general process for selecting, assessing and implementing open source products. Here maturity is used in a more narrow context and means the quality of a 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.

MATURITY Criteria	
#	<i>Sub-Criteria</i>
1	Development status
<p>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.</p>	
2	Reference site
<p>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 [Vittie01]. Reference sites listed on the project home page and the availability of documented business cases are the relevant criteria.</p>	

4 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. Please read the evaluation criteria sections in the context of their definitions in the previous chapter. Relative terms always refer to the other evaluated open source ERP systems.

4.1 Comparison Table

In the functionality section only some modules are listed. For details please read the text. GNU Enterprise is not production ready now and does not have any ERP module.

Legend:

√	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						
#	Sub- Criteria	SQL Ledger	LX Office	TinyERP	GNUe	ERP5	Opentaps	Compiere
SIZE								
	micro	+	+	+				
	small	+	+	+		+	+	+
	medium	+	~	~		+	+	+
	large					+	+	~
FUNCTIONALITY								
1	Number of Tables	45	36 ¹³	162	n/a	n/a ¹⁴	763	385 ¹⁵
	e-Commerce	x	√ ¹⁶	√ ¹⁷	x	√	√	√
	Accounting	√	√	√	x	√	√	√
	MRP			√	x	√	√	
	POS			√	x	√	√	
	Inventory&Warehouse	√	√	√	x	√	√	√

¹³ LX Office tables: only ERP module without CRM is counted

¹⁴ As ERP5 uses ZOPE application server (Z Object Persistence Engine) as object storage there are no tables to count

¹⁵ The database has 499 tables. 385 are used for functionality and 114 constitute the meta data application dictionary.

¹⁶ LX Office sells a module for osCommerce integration, www.oscommerce.org

¹⁷ Tiny ERP has EzPublish integration module, <http://ez.no/>

#	Sub- Criteria	SQL Ledger	LX Office	Tiny ERP	GNUe	ERP5	Opentaps	Compiere
FLEXIBILITY								
1	Customization	~	~	+	+	+	+	+
2	Flexible upgrades	+	+	+	?	?	~	+
3	Internationalization	+	-	+	+	+	+	+
		multi site		multi site		multi site	multi site	multi site
4	User friendliness	~	~	+	?	+	~	~
5	Architecture	3-tier web	3-tier web	3-tier rich	2 or 3 tier rich/web	3-tier web	3-tier web	2 and 3- tier ¹⁸ fat
6	Scalability	+	~	~	?	+	+	+
7	Security	~	~	+	?	+	+	+
8	Interfaces	CGI, SOAP	CGI	XML-RPC, Office	XML-RPC, Corba, LDAP	XML-RPC, SOAP, XML	SOAP, CSV, XML	CSV
9	OS independence	√	√	√	√	√	√	√
10	DB independence	√	x	x	√	object db	√	x ¹⁹
11	Programming Language	Perl	Perl	Python	Python	Python	Java,Scripting	Java

¹⁸ Compiere's accounting module uses 3-tier architecture, the rest is 2-tier client/server architecture

¹⁹ Compiere supports Oracle database, Sybase is beta

#	Sub- Criteria	SQL Ledger	LX Office	Tiny ERP	GNUe	ERP5	Opentaps	Compiere
SUPPORT								
1	Support infrastructure	~	~	~	-	+	+	+
2	Training	x	x	√	x	√	√	√
3	Documentation	+	-	~	~	-	+	+
CONTINUITY								
1	Project structure	Company +partners	Company +partners	Company +partners	Community	Company +partners	Companies +community	Company +partners
2	Community activity	+	+	+	~	-	+	+
3	Transparency	~	~	~	+	~	+	+
4	Update frequency	+	~	+	~	+	+	+
5	Other lock-in effects							migration tool
MATURITY								
1	Development status	Stable	Stable	Stable	Planning, beta	Stable	Stable	Stable
2	Reference site	+	~	~	X	+	+	+

#	Sub- Criteria	SQL Ledger	LX Office	Tiny ERP	GNUe	ERP5	Opentaps	Compiere
OTHER								
1	License	GPL	AL,GPL,LGPL	GPL	GPL	GPL	MIT-PL, GPL	MPL based
2	Online demo	√	√	x	x	x	√	x
3	Sourceforge hosted	√	√	x ²⁰	x	x ²¹	√	√
4	CVS access	x	√ ²²	partner only ²³	√	√	√ ²⁴	√
5	Download checksum	√	x	x	x	x ²⁵	x	√
6	Project start	2000	2004	2000	2000	2002	2001 (2005) ²⁶	1999

²⁰ TinyERP is also available on sourceforge, but this page is not maintained, even the files are old versions.

²¹ ERP5 is also available on sourceforge, but this page is not maintained, even the files and homepage link are missing.

²² LX-office offers subversion access on www.lx-office.org. CVS data on www.sourceforge.org is just a very old initial version.

²³ TinyERP offers subversion access for partners. CVS data on www.sourceforge.org is a very old version.

²⁴ Opentaps itself has no public code versioning system, but the underlying OfBiz framework can be accessed via subversion or weekly builds.

²⁵ A "live CD" containing a bootable Linux with installed ERP5 can be downloaded via BitTorrent peer-to-peer network

²⁶ The OfBiz project started 2001, the Opentaps brand in 2005.

4.2 SQL Ledger

<http://www.sql-ledger.org>

<http://sourceforge.net/projects/sql-ledger>

License: GNU General Public License (GPL)

Status: stable

This section is based on [\[Krumb05 pp. 482- 495\]](#) 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
3	Transparency	roadmap; no public CVS access and bug tracking
4	Update frequency	regular
5	Other lock-in effects	none
MATURITY		
1	Development status	stable
2	Reference site	many reference sites, worldwide distributed

F1 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.

F2 Flexible upgrades

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

F3 Internationalization

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

F7 Security

Access is controlled on user level. The administrator can grant several functionality groups per module to a user²⁷.

F8 Interfaces

[Sarang02] features a chapter describing the development and use of SOAP interfaces for SQL Ledger.

S3 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.

C1 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.

C3 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.

²⁷ <http://www.sql-ledger.org/cgi-bin/nav.pl?page=feature/acs.html&title=Access%20Control>, as viewed on 2006-01-07

4.3 LX Office

<http://www.lx-office.org>

<http://sourceforge.net/projects/lx-office>

License: Artistic License, GNU General Public License (GPL), GNU Library or Lesser General Public License (LGPL)

Status: stable

This chapter is based on [\[Krumb05, pp.495- 498\]](#). 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 fork²⁸.

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.

²⁸ Additionally he claims, that Linet Services tried to steal the sql-ledger.de domain name.

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
3	Transparency	roadmap, change log, bug tracking and subversion access.
4	Update frequency	regular, less compared to SQL Ledger
5	Other lock-in effects	
MATURITY		
1	Development status	stable
2	Reference site	several sites in the German speaking market

S3 Documentation

Non free documentation under development. 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.

C1 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.

C2 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

C3 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.

C4 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.

4.4 Tiny ERP

www.tinyerp.com

www.tinyerp.org

License: GNU General Public License (GPL)

Status: stable

The user interface of TinyERP is comfortable and well designed. Lots of charts and graphics types are available. It is targeted to micro, small and medium enterprises having up to 150 employees in the sectors trade, distribution and service. The modules available are accounting (inspired by Sage 100), CRM, sales& purchases (delivery, purchase, sales management, point of sale), human resources, marketing, MRP (manufacturing resource planning), MRPII, inventory control, project management. Additionally, there are some special purpose modules and a interface to ezPublish²⁹ eCommerce application. Modules can be installed partly in TinyERP.

TinyERP		
#	Sub-Criteria	Description
FLEXIBILITY		
1	Customization	XML defined forms and workflow
2	Flexible upgrades	for high level customizations ³⁰
3	Internationalization	11 languages, 2 accounting schemes
4	User friendliness	well designed; intuitive rich GUI
5	Architecture	3-tier, with rich GUI client
6	Scalability	good, limited to PostgreSQL
7	Security	fine grained access control (groups, roles)
8	Interfaces	server: XML-RPC, e-Commerce client: Excel, CSV, PDF, Open Office ³¹

²⁹ www.ez.no

³⁰ Flexible upgrades are provided when the ERP service engine is not changed.

³¹ Open Office is also used for report design.

TinyERP		
#	Sub-Criteria	Description
9	OS independence	server ³² : Linux, Windows client: Linux, Windows, Mac OS X ³³
10	DB independence	PostgreSQL ³⁴
11	Programming Language	Python
SUPPORT		
1	Support infrastructure	partner network, forums, support contracts
2	Training	user and developer training
3	Documentation	Wiki, user and administration documentation
CONTINUITY		
1	Project structure	company-driven, international partner network
2	Community activity	~200 forum messages per month
3	Transparency	change log, bug tracking; no public CVS
4	Update frequency	regular, fast progress
5	Other lock-in effects	none
MATURITY		
1	Development status	stable ³⁵
2	Reference site	two hotels are listed on the homepage

F1 Customization

Forms, reports and workflow development are supported without the requirement of programming. Forms and workflow are customizable with XML files. Tiny ERP is customizable via the client in Administrator mode. A brief tutorial for developing new modules is available.

The code is very compact and has a tight memory footprint. For low level code customization a forum thread³⁶ explains some of the functionality.

³² The platform is developed and tested on Linux.

³³ The client installation on Mac OS X requires advanced administration knowledge.

³⁴ The object-relational features (inheritance..) of PostgreSQL are used.

³⁵ MRP module is missing documented reference customers.

³⁶ It is the thread "[Finding out how TinyERP works?](http://tinyerp.org/forum/viewtopic.php?t=501&sid=8a39346025c3e52cb273869a96ab14d0)" in the "General" tinyERP Forum:
<http://tinyerp.org/forum/viewtopic.php?t=501&sid=8a39346025c3e52cb273869a96ab14d0>

F3 Internationalization

The translation of languages lies mostly on the server side and to a small part on the client side³⁷. It provides a translation mechanism for the content of form fields showing list values. Server side translations are available in Spanish, Hungarian, French, German, Romanian, Italian, Portuguese, Chinese, Czech, Albanian and Dutch. Some of these translations are not for the current version of TinyERP or are not complete. Without client side translation the GUI is in mixed language. TinyERP supports multiple accounting schemes. Schemes for Belgium and France can be delivered.

F5 Architecture

TinyERP uses a self-developed proprietary server. All logic and data stems from the server. The client is based on the multi-platform GTK+³⁸ toolkit and provides a “fat” client like rich GUI.

S3 Documentation

A Wiki is used for collaborative documentation authoring. The installation (38 pages), user (141 pages) and administration (31 pages) manual are free. A user tutorial (77 pages) is sold. A brief public introduction to module development is available online. The technical concepts and architecture are not documented. Partners have access to additional resources.

C1 Project structure

TinyERP is run by the Belgium company Tiny sprl³⁹ which employs five developers. It is mostly used in French-speaking countries. Besides the many French partners there are also partners in Argentina, Canada, Spain, Switzerland. The founders themselves sell services too. Most contributions concern translation and bug fixing. Partners also do module development.

C3 Transparency

New features are published in the “New Releases” forum when released. Polls on new wanted features are also conducted there. A detailed change log is published in the download section. There is no public roadmap available. Access to the concurrent code versioning system is available only for partners.

³⁷ www.tinyerp.org/download_i18n.php, as viewed on 2006-01-12

³⁸ www.gtk.org

³⁹ www.tiny.be

4.5 GNU Enterprise (GNUe)

www.gnenterprise.org

www.gnu.org/software/gnue/project/project.html

License: GNU General Public License (GPL)

Status: Developer tools: alpha, beta

ERP modules: planning

This chapter is based on [Cater05, GNUe02, Müll05]. GNUe aims to deliver a complete ERP Software Package based on universal database drivers and XML defined forms, reports and workflow. Its architecture is based on open standards and universal best-practice approaches with interfaces to several programming languages. There is no out-of-the-box packaged solution available. All **ERP package modules** are **still in planning stages**.

The Developer Tools projects constitute the basis for the planned ERP package. The Developer Tools sub projects are very active and in alpha or beta state. The GNUe project in general is in an early stage. Your GNUe application currently needs to be designed from scratch.

GNU Enterprise (GNUe)		
#	<i>Sub-Criteria</i>	<i>Description</i>
FLEXIBILITY		
1	Customization	XML-defined with IDE
2	Flexible upgrades	conceptually supported, but not documented
3	Internationalization	XML based GUI translation
4	User friendliness	average
5	Architecture	2-tier, or 3-tier operation with proprietary application server and a forms client
6	Scalability	database scales, server needs proof
7	Security	role based access control
8	Interfaces	XML-RPC, CORBA, LDAP
9	OS independence	server and forms client run on nearly any OS

GNU Enterprise (GNUe)		
#	Sub-Criteria	Description
10	DB independence	open source: PostgreSQL, MySQL, MaxDB, xBase, Interbase, Ingres, Firebird commercial: DB2, Oracle, Sybase ODBC and CSV files
11	Programming Language	Python, server also supports other languages
SUPPORT		
1	Support infrastructure	mailing lists and IRC (Internet Relay Chat)
2	Training	n/a
3	Documentation	community wiki; developer documentation ⁴⁰
CONTINUITY		
1	Project structure	community driven
2	Community activity	IRC shows daily activity and immediate response
3	Transparency	roadmap ⁴¹ , change log, bug tracking, subversion access
4	Update frequency	regular releases of developer tools
5	Other lock-in effects	none
MATURITY		
1	Development status	developer tools are in alpha or beta
2	Reference site	no ERP users, developer tools users are not documented

F1 Customization

Forms, reports and workflow are XML-defined in the Designer Integrated Development Environment (IDE) which allows Rapid Application Development. The knowledge needed for doing custom development is database design, Python Scripting and XML.

F3 Internationalization

Translation of forms can be provided with XML files. The GUI of the developer tools is translated into English, Dutch, Hungarian, German and partly French, Spanish, Russian, Italian, Romanian, Lithuanian.

⁴⁰ Developer documentation is sometimes not complete or out of date. A summary is available in the appendix chapter [8.5 GNU Enterprise Developer Tools](#).

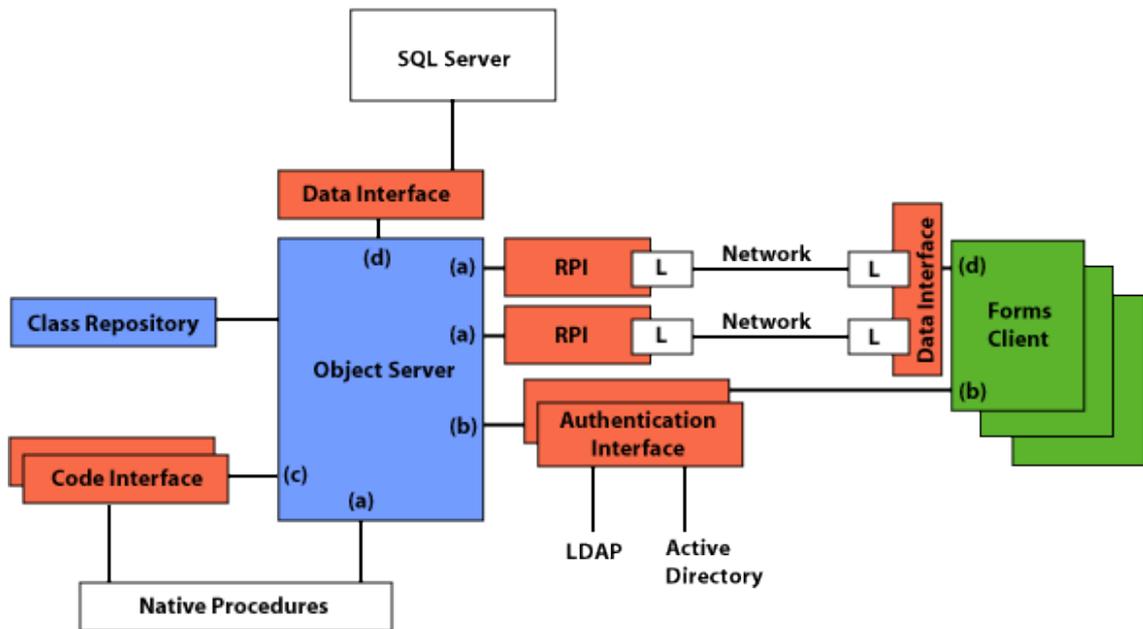
⁴¹ Roadmaps and change logs are provided for all developer tools sub projects.

F4 User friendliness

A subset of a rich interface is possible and can be deployed as fat client GUI with native look and feel for Windows, Linux/Unix, MacOS and OS/2 based on wxPython⁴², HTML Web interface and console/terminal interface.

F5 Architecture

New Architecture Diagram
GNU Enterprise Application Server v2.01



Blue represents the Application Server with some components pulled from "gnue common" functionality
Red represents GNUe Common
No arrow implies 2-way communications
RPI means Remote Protocol Interface
L means standard Library (e.g. XML-RPC, Corba etc.)
(a), (b), (c), (d) represent common API definitions

Drawing by Neil Tiffin, neilt@gnue.org
© 2002 by Free Software Foundation
24 November 2002

Figure 10: GNU Enterprise architecture

GNUe can be used in the 2-tier client/server and 3-tier mode. The application server provides various interfaces. The used components (see "[8.5 GNU Enterprise Developer Tools](#)") are divided into several sub projects :

⁴² www.wxPython.org, wxPython is a wrapper to the portable C++ GUI class library wxWindows, which provides an interface to platform specific GUI libraries.
<http://www.python.org/doc/faq/gui.html#wxwindows>

Developer Tool	Version ⁴³	Diagram name
Common Library	0.6.1	Interfaces
Forms	0.5.13	Forms Client
Reports	0.1.8	
Application Server	0.4.3	Object Server
Navigator	0.0.9	
Designer	0.1.7	
Bayonne Telephony 2	1.0.1	

F8 Interfaces

The communication between client and application server uses XML-RPC and CORBA. An LDAP interface is provided. The application server provides interfaces for native procedures.

C1 Project structure

This is a community driven project. The development team consists of 6 core developers and more than 20 active contributors. [\[Elli04\]](#) analyzes the community organization and culture of GNUe in detail.

⁴³ as published on 2005-10-25, Version numbers above 1.0 are production versions

4.6 ERP5

www.erp5.org

License: GNU General Public License (GPL)

Status: beta or stable depending on module

This chapter is based on [[Smets02](#), [Smets04](#), [Deld05](#)]. 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.
- Groupware

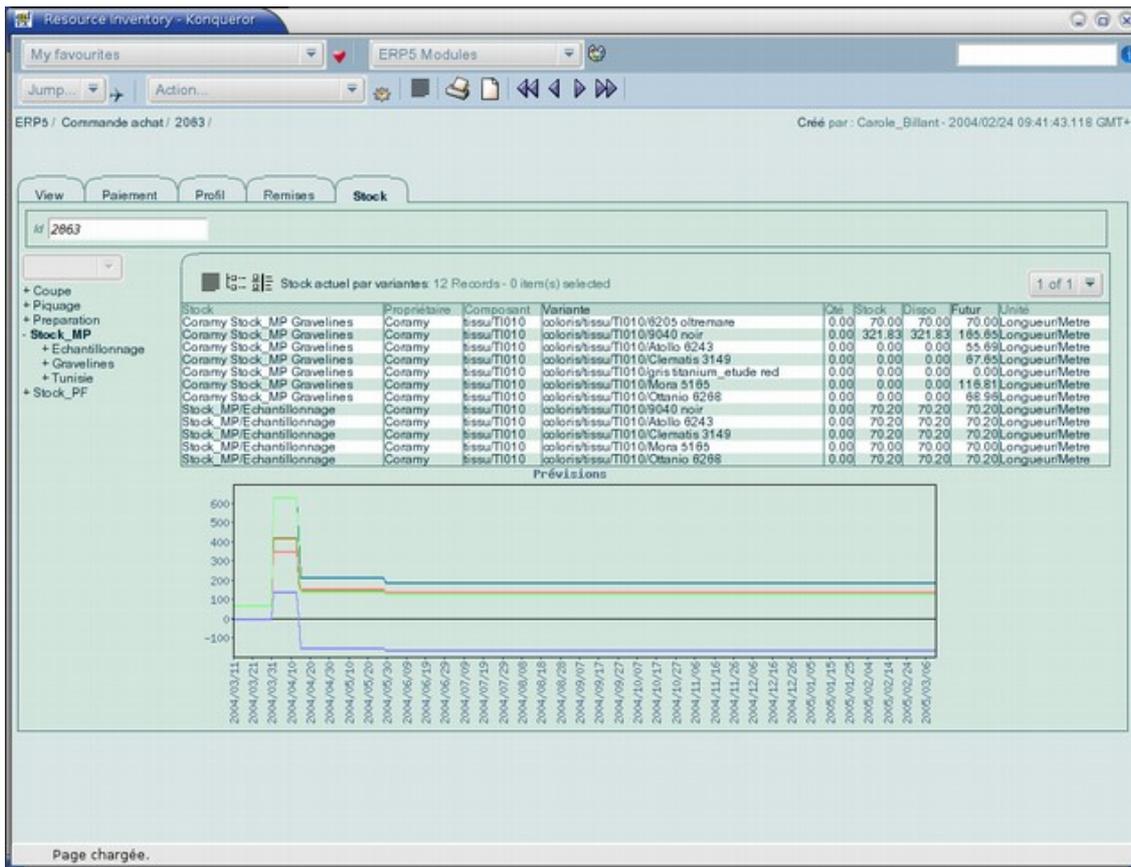


Figure 11: Trade module screen shot: multi-organization, multi-variation analytical stock management

ERP5		
#	Sub-Criteria	Description
FLEXIBILITY		
1	Customization	business templates, Zope ⁴⁴ 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 ⁴⁵ , 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 ⁴⁶
3	Transparency	subversion access, no roadmap
4	Update frequency	regular
5	Other lock-in effects	none
MATURITY		
1	Development status	stable, some modules are beta ⁴⁷
2	Reference site	solutions for apparel industry, aerospace, banking, hospital and government ⁴⁸

⁴⁴ Zope (Z Object Persistence Engine) is an open source application server for building content management systems, intranets, portals and custom applications. www.zope.org

⁴⁵ The system is tested on Linux, which is also the development platform.

⁴⁶ Mailing list archive and statistics available at:

<http://blog.gmane.org/index.php?prefix=gmane.comp.python.zope.erp5>

⁴⁷ <http://www.erp5.org/sections/download/status/view>

⁴⁸ <http://www.erp5.com/industry>

F1 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.

F3 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 to group 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.

F4 User friendliness

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

F5 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. ERP5 uses, extends or provides 5 important Zope components:

Page Templates implement the web user interface and presentation logic. Workflows are used for the development of various types of decision workflows. ERP5Form⁴⁹ is used for web form development. XMLObjects add synchronization capabilities and autonomous activity (timed, pre- and post-condition triggers) to ERP5- Zope objects. The synchronization of distributed sites is carried out through XML export and import and also supports unreliable communication channels, which are allowed to fail. The duplication of reporting data to a relational database allows simple SQL querying.

⁴⁹ ERP5form is based on ZOPE Formulator: <http://www.zope.org/Members/infrae/Formulator>

F8 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.

F10 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 Online Analytical Processing (ROLAP) for object databases. Those attributes used in reports are stored in the relational database for faster querying together with its references to the object database. The advantages of quick access and flexible data structures of object databases on the one hand, and easy querying and grouping of relational and multidimensional databases on the other hand are combined. If reporting data is not duplicated simple scripting is required for custom reports. Beside its built in object database, Zope supports various relational databases. No information was found, why the object oriented database approach was chosen.

S1 Support infrastructure

Due to the small amount of available documentation, support from Nexedi is needed to implement ERP5. Usually at least one person from Nexedi is engaged in the implementation team and responsible for knowledge transfer. Online support is provided by users' and developers' mailing lists.

S2 Training

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

C1 Project structure

ERP5 is run by the French consulting company Nexedi⁵⁰ with partners in France, Luxembourg, Germany, Poland, Brazil, Senegal and India⁵¹. The Brazilian research group is also contributing. Their work is documented in Portuguese on the project homepage. A project also supported by Nexedi, is Z3ERP⁵² 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.

C3 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.

C4 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 document⁵³ which shows the completion percentages of components and modules, but you can not see which functionality was introduced and when.

M2 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 authorities⁵⁴.

⁵⁰ <http://nexedi.com/>

⁵¹ <http://www.erp5.com/consulting/erp5-consulting-partners>

⁵² <http://www.z3erp.org>

⁵³ <http://www.erp5.org/sections/download/status/view>

⁵⁴ <http://www.erp5.com/industry>

4.7 Opentaps - OfBiz

www.opentaps.org

www.ofbiz.org

www.opensourcestrategies.com

www.opensourcestrategies.org

License: MIT Public License, accounting and CRM module are dual licensed GPL/commercial

Status: stable

Opentaps is a production-ready version of the Open for Business (OfBiz) framework that offers stable releases and upgrade patches. Thus Opentaps is used as a synonym for OfBiz in this paper. The framework is intended for developing enterprise-level custom business applications. Its initial focus was on e-Commerce solutions, but it evolved into a general framework for ERP development.

The target groups are retailers, distributors and manufacturers. There are modules for e-Commerce, product catalog, point of sales, order entry, inventory and warehouse management, manufacturing, customer service, CRM, marketing and content management. The accounting/general ledger module⁵⁵ is published under a GPL license with the option of buying a commercial license. No national accounting schemes are available. It supports multiple sites, multiple costing methods and file based interfaces to external general ledger applications (XML, ASCII and CSV export). The data model allows features like planning, forecasting and budgeting. Multiple web stores and distributed brick and mortar stores with separate inventories and data synchronization are supported, too.

Opentaps' module structure is an attempt to group OfBiz application functionality like traditional ERP modules. From the documentation available it is sometimes not clear to which module a certain functionality belongs to.

⁵⁵ www.opensourcestrategies.com/ofbiz/accounting.php, viewed on 2006-01-17

Opentaps - OfBiz		
#	Sub-Criteria	Description
FLEXIBILITY		
1	Customization	XML defined data, business and presentation layer. Business layer services are called on certain events and conditions (XML specified).
2	Flexible upgrades	for high level customization
3	Internationalization	39 languages, multiple int. distr. sites support
4	User friendliness	intuitive simple Web GUI
5	Architecture	3-tier web architecture
6	Scalability	good (many deployment options)
7	Security	fine grained access control (user, group, role and other categories ⁵⁶)
8	Interfaces	SOAP, XML, CSV, PDF
9	OS independence	Windows, Linux/Unix, Mac OS X, web client
10	DB independence	open source: MySQL, MaxDB, PostgreSQL commercial: Oracle, DB2, Microsoft SQL,
11	Programming Language	Java, Scripting (Beanshell, XML-Minilang)
SUPPORT		
1	Support infrastructure	internationally distributed service providers; mailing lists, forums
2	Training	recorded training sessions, user conference
3	Documentation	Wiki, many other documentation, unstructured
CONTINUITY		
1	Project structure	community driven
2	Community activity	~400 messages per month
3	Transparency	change log, bug and issue tracking, subversion access
4	Update frequency	regular
5	Other lock-in effects	other open source projects
MATURITY		
1	Development status	stable
2	Reference site	many references, just a part uses it as ERP system.

⁵⁶ www.ofbiz.org/general-overview.html, www.opensourcestrategies.com/ofbiz/security.php, as viewed on 2006-01-16

F1 Customization

As shown on Figure 15 in the architecture section Opentaps has a n-tier web architecture with three layers on the application server tier: data (entity engine), business logic (service engine) and presentation.

Each layer is defined with meta data or an easy-to-use scripting language. The advantages of high-level languages and meta data based code generation are combined with the structural modeling capabilities of Java⁵⁷. This is achieved through publishing APIs⁵⁸ (application programming interfaces) of Java components and their use on a higher level⁵⁹. Because of the meta data approach used on all layers changes on the data model can automatically change the services on the business logic layer and the forms on the presentation layer working with this data. All layers are mutually aware of changes on other layers. For customization and implementation basic Java⁶⁰, web engineering, XML and database design knowledge is necessary.

F2 Flexible upgrades

For the purpose of low level customization flexible upgrades are possible, but detailed documentation is missing on this topic. An Implementation Certification Service is available proving future compatibility of customizations and the use of best practices.

F3 Internationalization

The user interface language can be switched any time and translations into 39 languages are at least partially provided. There is no translation mechanism for dynamic content like workflow states. Multiple sites as well as multiple accounting and costing methods are supported.

F4 User friendliness

All Web GUI elements are supported. The presentation is separated into pieces like header, footer, left bar, right bar which are composed of several templates.

F5 Architecture

[\[based on Chen05, Basil05\]](#) Opentaps is built upon a standards based service-oriented web architecture. The application server consists of presentation, business logic and data layer. All layers are meta data or scripting language

⁵⁷ Based on www.opensourcestrategies.com/ofbiz/developing_overview.php, viewed on 2006-01-16

⁵⁸ www.ofbiz.org/documents.html, viewed on 2006-01-16

⁵⁹ The productivity gains stemming from this approach and a comparison with other general frameworks are discussed in the FAQ: <http://ofbizwiki1.go-integral.com/Wiki.jsp?page=FAQ1>, viewed on 2006-01-17

⁶⁰ www.opensourcestrategies.com/ofbiz/java_knowledge.php, viewed on 2006-01-17

defined.

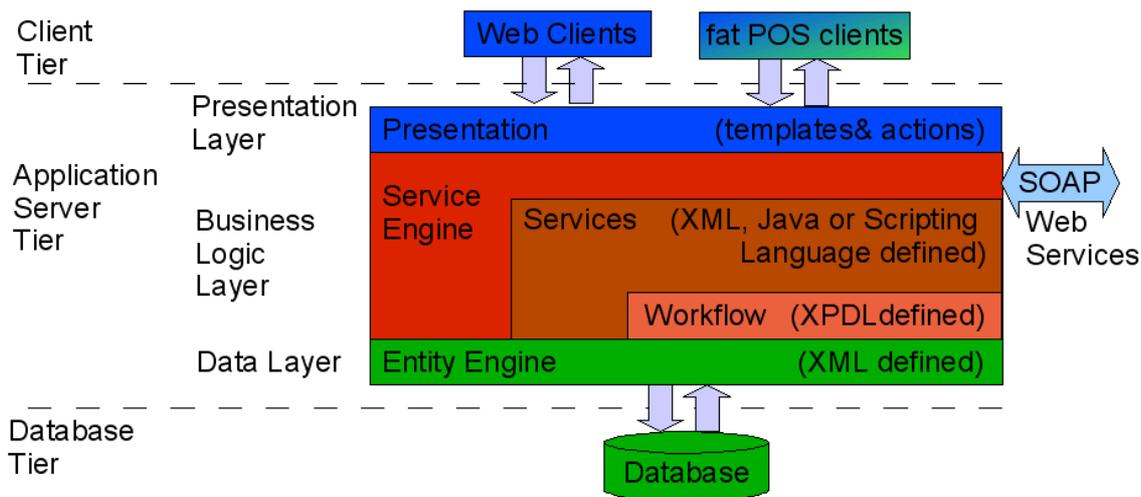


figure 12: Opentaps/OfBiz service oriented architecture

The entity engine handles data access and persistence via data models defined in XML files. There is no need to work with Java or SQL code.

The business logic is defined as small reusable pieces of code that can be written in various high-level scripting languages or XML. Services can be called directly from web applications, workflows and remotely via SOAP and RMI (Remote Method Invocation). The workflow can be defined in XML Process Definition Language (XPDL), a Workflow Management Coalition⁶¹ standard supported by many modeling tools⁶² and workflow engines. The presentation layer uses a Model View Controller (MVC) pattern to allow reuse of design elements, view logic and form action. Design and programming tasks are separated. Due to the highly component-based architecture customization tasks can be distributed to different developer roles.

F6 Scalability

Many deployment options are provided. The server can be run in parallel with fail over synchronization. The service-oriented architecture allows the distribution of application server parts in many ways. Load balancing by using multiple web servers and databases can be used for an application.

F8 Interfaces

As the service-oriented architecture is used internally, it can also be used to communicate with other systems. Each service can also be published as web

⁶¹ www.wfmc.org

⁶² A open source graphical workflow process editor used by OfBiz project members is JaWE (<http://jawe.enhydra.org>).

service using the SOAP protocol. Jasper Reports⁶³ and FreeMarker⁶⁴ PDF creation are used for reporting. For shipping and tracking of parcels there is an interface with UPS (United Parcel Services) online tools⁶⁵.

S1 Support infrastructure

Besides professional support services from the Opentaps maintainer Open Source Professional Inc. and the OfBiz founders Undersun Consulting, local support can be given by the OfBiz community. Customers and service providers for OfBiz are distributed worldwide. Some countries have a local community site, listing consulting companies offering support and services.

S2 Training

Recorded training sessions, recorded user conference sessions, remote training, additional online documentation and a quick reference book are sold by Undersun Consulting⁶⁶. A user conference with training sessions is organized annually.

S3 Documentation

Free developer, technical and some user documentation is available online. A Wiki system serves for collaborative documentation authoring. Additional documentation (800 pages) with detailed information for end users and additional information for developers is sold. The data model is designed for flexibility and well documented in [Silver01]. What is missing to gain more awareness and to reduce the learning curve for the project, is a book with well structured information for ERP implementors and developers.

C1 Project structure

Opentaps is driven by Open Source Strategies Inc., a Los Angeles based consulting company specializing on open source enterprise solutions. It publishes stable ERP oriented releases of OfBiz and back-patched bug fixes/updates under the Opentaps brand [Jones05]. Ofbiz is a developer oriented and community driven project. It is built weekly and supported as well as used by many consulting companies worldwide. Due to its developer orientation the community is very active, and the code is contributed by many individuals and groups which use this project as base for various custom software or provide related consulting services. [Chalif04a] features an interview

⁶³ <http://jasperreports.sf.net>

⁶⁴ <http://freemarker.org>, used as described in www.ofbiz.org/best-practices.html, viewed on 2006-01-17

⁶⁵ <http://ofbizwiki1.go-integral.com/Wiki.jsp?page=UpsOnlineTools>, viewed on 2006-01-18

⁶⁶ www.undersunconsulting.com/e-commerce/control/main, viewed on 2006-01-06

about OfBiz project structure, community, target markets and open source business models with David Jones, a founder of OfBiz. At the time of writing OfBiz was in the incubation⁶⁷ phase of becoming an Apache Software Foundation⁶⁸ project. The community structure and multi branding of Opentaps/OfBiz also results in various online information sources and unstructured documentation.

The OfBiz team consists of about 6 core developers. A second open source ERP project based of the OfBiz framework is Neogia⁶⁹ which delivers add-on components to OfBiz.

C2 Community activity

The community is very active in contributing source code. Relevant online channels used for discussion are:

Communication channel	Messages per month
dev@lists.ofbiz.org	~100-150
users@lists.ofbiz.or	~200-300
sourceforge forum	~75
opentaps-users@lists.sourceforge.net	~50

C3 Transparency

Opentaps has a documented project roadmap with time schedules for features to implement. A blog⁷⁰ is summing up weekly changes of the OfBiz project and its contributors. Everyone is invited to contribute and the subversion source versioning system is public. A professional bug and issue tracking system is used⁷¹.

C5 Other lock-in effects

The OfBiz project uses and integrates many open source projects⁷². If a used project loses continuity, it is either maintained by the OfBiz community or replaced.

M2 Reference site

Most reference sites of them use only a part of the functionality provided. It is also proofed as a general purpose framework for database-centric applications.

⁶⁷ <http://wiki.apache.org/incubator/OFBizProposal>, as viewed on 2006-01-19

⁶⁸ <http://www.apache.org>

⁶⁹ <http://sourceforge.net/projects/neogia/>, viewed on 2005-11-12

⁷⁰ <http://ofbiz-new.blogspot.com>, viewed on 2006-01-17

⁷¹ <http://jira.ofbiz.org>, viewed on 2006-01-17

⁷² www.smedia.info/ofbiz-components.asp, viewed on 2006-01-15

4.8 Compiere

www.compiere.org

sourceforge.net/projects/compiere

www.red1.org (forum and developer information of a Compiere partner)

www.knowledgesuccess.com/compiere

License: Mozilla Public License (MPL 1.1)

Status: stable

This is the most popular open source ERP system nowadays. The community is very active in Sourceforge user forums and there are more than 900000 downloads, an international partner network and several reference installations. Compiere is targeted to small and medium companies operating in distribution/retail, and service areas. Its functionality is organized along business processes:

- Quote-to-Cash covers the whole customer process from first customer contact, pre-sales, sales and order activities to cash receive. This process also has strong CRM functionality.
- Requisition-to-Pay covers the whole supplier process from request, purchase order, receipt to payment.
- Customer Relationship Management (CRM) is no separate module. It is integrated into the other modules.
- Performance Analysis covers accounting and controlling.
- Production is currently in beta state and reference customers are missing.
- Web Store delivers a basic shop system and is integrated into the core ERP system.
- Service management and basic project management functionality is also available.

Compiere		
#	Sub-Criteria	Description
FLEXIBILITY		
1	Customization	meta data (application dictionary) editing and callout functions
2	Flexible upgrades	for high level customizations (as above)
3	Internationalization	26 languages, 4 accounting schemes, multiple internationally distributed sites support
4	User friendliness	fat client, no flexible layout
5	Architecture	mixture of fat (Java) client 2-tier and 3-tier architecture
6	Scalability	good, depends mostly on database
7	Security	fine grained access control (roles)
8	Interfaces	CSV, PDF
9	OS independence	Windows, Linux/Unix, Mac OS X
10	DB independence	Oracle, Sybase (beta)
11	Programming Language	Java
SUPPORT		
1	Support infrastructure	partner network, forums, contracts
2	Training	user and developer training
3	Documentation	detailed user and minimal developer documentation
CONTINUITY		
1	Project structure	company driven, partner network
2	Community activity	~500 messages per month
3	Transparency	roadmap, change log, bug tracking, CVS access is public
4	Update frequency	regular
5	Other lock-in effects	PDF generation, migration tool and Oracle database need a license
MATURITY		
1	Development status	stable, manufacturing module and full web client are beta
2	Reference site	many reference sites

F1 Customization

[Lomb05] Compiere allows meta data based customizations on the fly without shutting down the system. Meta data is stored on the database in 114 tables called Application Dictionary. It can be modified with the Compiere client having administrator rights.

To create a new window in Compiere the database structure must be defined first. A Compiere GUI consists of a window with several tabs and fields and the corresponding database elements:

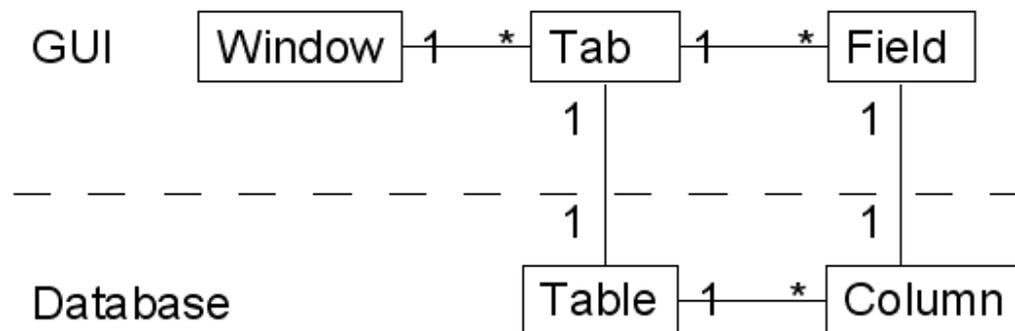


Figure 13: Compiere Application Dictionary GUI elements, taken from [Lomb05]

Windows GUI structure input handling and validation, print formats, default accounts, process buttons, menu structure and callouts are defined in the Application Dictionary. Application Dictionary modifications generate model code. For lower level customization this model can be extended with callouts, document process workflows (process button actions) and SQL-stored procedures.

A Callout is a custom Java code attached to a field and called after the input activity. It can change the values of all fields in a window and access all tables. Experts knowing the code structure can do further customizations by following the patterns used by Compiere.

F2 Flexible upgrades

The commercial migration tool supports high-level customizations done via application dictionary and callout functions. For changes on the core code upgrade flexibility depends on the patterns used.

F3 Internationalization

The GUI and reports are translated into English, Chinese, German, Spanish, French, Italian and Dutch. These translations are maintained by Compiere Inc. The Status of other translation efforts (26 languages) is available online⁷³. Accounting schemes are available for US, Spain, Italy and Germany. Compiere supports multiple sites⁷⁴ by providing three entity levels: System level is accessible by all organizations. Client level defines information and accounting structure. Multiple accounting schemes are allowed for a single client⁷⁵. Compiere supports multiple costing methods⁷⁶. One of them is assigned to an accounting schema. Organization is the lowest level, where transactions are performed. They may be hierarchically structured and have additional information structures. Virtual organizations (service centers) are supported.

F4 User friendliness

The user interface has a “tabbed” layout and supports a subset of a rich GUI (also see figure 13). For example master/detail views are not possible on one screen. Switching to detail rows requires the use of page tabs (left side of figure 14). Some users complain about bad look and feel.



Figure 14: Compiere GUI, taken from [Pink05]

⁷³ <http://www.compiere.org/product/status.html>, as viewed on 2005-02-08

⁷⁴ <http://www.compiere.org/technology/multi/org.html>, as viewed on 2005-02-08

⁷⁵ <http://www.compiere.org/technology/multi/accounting.html>, as viewed on 2005-02-08

⁷⁶ <http://www.compiere.org/technology/multi/costing.html>, as viewed on 2005-02-08

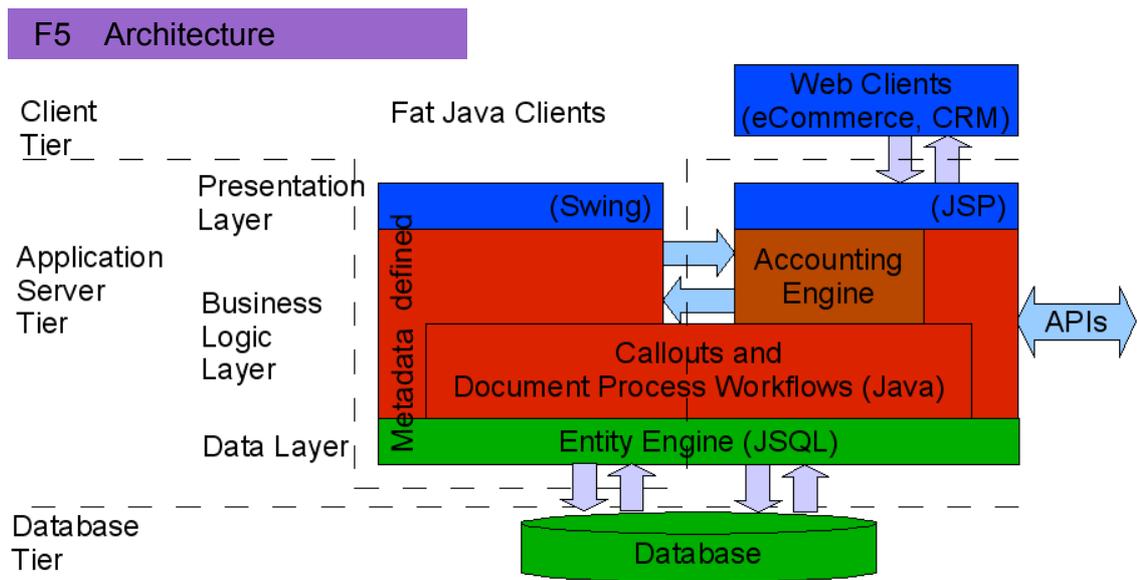


Figure 15: Compiere architecture

Compiere's architecture is a mixture of fat client 2-tier and 3-tier architecture. It uses a fat Java Client (Java Webstart). The accounting engine is on the application server (JBOSS J2EE container). Thin Web clients are used for web store and some CRM functionality. There is also a beta thin client serving all forms, but it needs further work to be equivalent to the fat client.

F6 Scalability

Scalability is no problem on the database side. The application is mostly 2-tier based and thus heavily depends on database scalability. Additionally the used application server supports clustering. No cluster installation is documented. Scalability is not proofed, but the used components are known to scale well.

F8 Interfaces

Emails can be sent and files can be attached to documents. Compiere has a built-in reporting engine. A Sourceforge side project provides Jasper Report⁷⁷ integration. CSV export and import of data is provided by the commercially available migration tool. Other interfaces can be developed in Java using the briefly (Javadoc) documented APIs (application programming interfaces).

⁷⁷ <http://jasperreports.sf.net>

F10 DB independence

The database abstraction layer is very SQL-centric and was initially developed for Oracle. There is a concept for database independence by translating Oracle SQL statements to other SQL dialects. Sybase support is currently in beta. Some open source databases are available as Sourceforge side projects, but they are neither supported nor tested. PostgreSQL related problems are discussed in developer forums. The SQL code for PostgreSQL was translated manually. Another company has ported Compiere to Fyrcle, a Firebird database with partly Oracle PL/SQL compatibility. Compiere Inc. states that open source databases are not supported because it is unclear if problems stem from the database or the application when supporting customer incidents. Requirements for open source database support are ANSI SQL99 compliance and professional commercial support.

On the project roadmap support for MS SQL, DB2 and Cloudscape is announced. Cloudscape is an open source Java database with DB2 compatible SQL dialect, sufficient for demo applications.

S3 Documentation

A very detailed user manual consisting of 850 pages is commercially available online. Apart from the developer training, little developer documentation is available. The auto-generated API documentation in Javadoc format is minimal. A community maintained Compiere Developer Documentation [[Lomb05](#)] consisting of 36 pages, as well as the Compiere web page and developer forums are a good starting point.

C1 Project structure

[[Chalif04b](#)] Compiere is a company driven project. ComPiere Inc., led by the main developer Jorg Janke, is based in Portland, US. It concentrates on development, second level (partner) support and training. ComPiere Inc. does not provide implementation services and thus does not compete with partners. More than 70 certified partners sell implementation and consulting services in at least 25 countries⁷⁸. Most development is done by Jorg Janke and another ComPiere Inc. employee. Some partners and users assist the project in requirements gathering, quality assurance, testing and patching. The Partners determine the development priorities. Some Sourceforge side projects are community driven.

⁷⁸ <http://www.compiere.org/support/index.html#list>, as viewed on 2006-02-08

C2 Community activity

Compiere has a very active user community. The forums are very responsive. It uses primarily Sourceforge forums and tracking services for communication with about 500 forum messages a month⁷⁹. Support requests, bugs and feature requests are tracked separately and have about 150 messages a month.

C3 Transparency

Project roadmap and detailed change log are available⁸⁰. Public Sourceforge services are used for bug tracking and change requests. CVS access is also public.

C5 Other lock-in effects

The PDF generation tool used as well as the migration tool are not open source and need a license. Customers with support contract also have access to installable fixes and patches, get priority support and optionally get an embedded Oracle license.

M2 Reference site

The initial application was built for a tire distribution company. It is documented together with a few other reference customers⁸¹. According to Compiere Inc. at least 100 companies use the software. Some partners have their customers cases described on their home page, and more details are provided during the contracting phase. The use of Compiere in two German companies was presented on an open source conference⁸²:

The first company used Compiere for make-to-order real-time price calculation of individually made front plates for electronic devices. A customer can order a front plate by providing the front plate design with a free software. The complete process of ordering, price calculation with the help of the design file, technical feasibility control, program generation for machine controlling, production of the front plate, quality control, packaging and shipping is covered by Compiere. An internal programmer partly supported by a Compiere partner did the customization. The core source code was customized, easy updatability was maintained.

[\[Dethle05\]](#) The second company is a small, fast growing IT system and service provider with several affiliates which used office tools for its processes before.

⁷⁹ http://sourceforge.net/project/stats/?group_id=29057&ugn=compiere&type=&mode=year, as viewed on 2006-02-08

⁸⁰ <http://www.compiere.org/product/status.html>, as viewed on 2005-02-08

⁸¹ <http://www.compiere.org/about/references.html>, as viewed on 2005-02-08

⁸² www.open-source-meets-business.de, as viewed on 2005-01-31

Compiere is used for the management of products, for providing virtual warehouse functionality, and purchasing and sales processes. The complete process from initial customer contact to accounting (quote to cash) afforded a highly flexible system for the generation of quotes and customer tracking. Interfaces to external accounting and project management were required. Furthermore a high degree of automation and the adaption to fast changing requirements were needed. With Compiere the company could cover all processes, do them faster and make them more reliable and transparent with automated reports.

Both systems were highly customized. There is a discussion about the compatibility of the US-oriented accounting module with German national requirements. National interfaces were not delivered by partners. Both companies were satisfied with the system and would make the same choice again. Cost, flexibility (customization possibilities), investment safety (vendor independence) and company-internal knowledge generation were the main reasons for the selection of Compiere.

5 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. GNU Enterprise is not production ready and missing any reference customers. TinyERP is evolving fast, provides more ERP functionality than SQL Ledger and can be customized with add-on modules. ERP5 and Opentaps (OfBiz) have very flexible architectures. ERP5 is following a purely object oriented approach, whereas Opentaps/Ofbiz is more relational database centric. Compiere, the current market leader, is a mature system, providing many customization possibilities. For a more detailed summary please look at the comparison table starting on page 33.

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.

Appendix A: Glossary

All Terms can also be looked up at <http://en.wikipedia.org/>. This online encyclopedia is especially suitable for technical and computer related terms. Many entries are Wikipedia based.

Alpha status	Alpha version or alpha release: the first release of a computer program, likely to be instable and incomplete but useful for demonstration, proof of concept
API	Application Programming Interface
Beta status	Beta version or beta release: release of a computer program that is still under development but published for testing purposes.
C	Programming language, Unix is developed in C
CGI	Common Gateway Interface, a standard web server interface
CORBA	Common Object Request Broker Architecture is a communication protocol, and object/service information model. It allows heterogeneous applications written in different languages and running on different platforms to interoperate. http://www.corba.org/
CRM	Customer Relationship Management
CSV	Comma Separated Values: ASCII file format where the attribute values of a table row are separated by commas.
CVS	Concurrent Versioning System: A client/server system for the central management of concurrent changes of source files. In most open source projects the source code is CVS managed. Also see www.cvs.org . A newer approach for versioning is subversion , used by open source projects like GNUenterprise, LX-Office and OfBiz. www.subversion.tigris.org
DATEV	An interface to DATEV Software, used by many accountants, especially in Germany. DATEV itself is an accountant cooperative providing software for accountants in Germany and Central Europe.

DCOM	Predecessor technology of .net for distributed computing on Microsoft Windows platforms
Delphi	Pascal Style programming language and IDE from Borland. http://www.borland.com/us/products/delphi/index.html
ERP	Enterprise Resource Planning
Gnome	GNU Network Object Model Environment: open source GUI (Desktop) for X-Server on Linux/Unix.
GNU	“GNU’s not UNIX” is a recursive acronym for the www.gnu.org project which aims to develop a free Unix – GNU/Linux
GNUe	GNU Enterprise www.gnuenterprise.org
GPL	GNU Public License, www.gnu.org/copyleft/gpl.html
GUI	Graphical User Interface
HBCI	Home Banking Computer Interface, is a bank independent online banking protocol used especially by German banks
HRMS	Human Resources Management System, an ERP module for human resources
IDE	Integrated Development Environment
Java	A platform neutral programming language widely used for enterprise software development. http://java.sun.com
JBOSS	A open source J2EE application server. http://www.jboss.org
J2EE	Java 2 Enterprise Edition, the Java distribution for developing enterprise applications. (http://java.sun.com/j2ee/index.jsp)
Kylix	integrated C++ and Delphi language solution for developing database, GUI, and Web applications for Linux, which are cross-platform ready for Windows. http://www.borland.com/us/products/classic_products/index.html
LINUX	Open source Unix operating system, also called GNU/Linux. It is the Linux kernel developed by Linus Thorvalds, combined with GNU libraries and other software.
LISP	Programming language
MDA	Model Driven Architecture, is a software design and development methodology proposed by the Object Management Group (www.omg.org).
MPL	Mozilla Public License (http://www.mozilla.org/MPL/)

MySQL	A popular open source database. www.mysql.org
.net	Programming framework from Microsoft
OSI	Open Source Initiative www.opensource.org
PDF	Portable Document Format is a file format developed by Adobe Systems (http://www.adobe.com/) for representing documents independent of platform and application. It can contain any combination of text, graphics and images.
Perl	Practical Extraction and Report Language; scripting language
PostgreSQL	Free object-relational database with enterprise features.
Python	A high level, productive, object-oriented scripting language. www.python.org
Scheme	Programming language, a dialect of LISP
SME	Small or Medium Enterprise
SOAP	SOAP is a protocol for exchanging XML-based messages over a computer network.
SQL	Structured Query Language, standardized language for querying relational databases
subversion	see CVS
Tcl	pronounced "tickle", a scripting language often used for rapid prototyping
UML	Unified Modeling Language, a widely accepted software modeling approach with several diagram types defined by OMG. http://www.uml.org/
UNIX	UNIX means all UNIX-like operating systems (AIX, BSD, HP-UX, Linux, Solaris,...).
Wiki	(interpreted as "What I Know Is") A website that allows users to add, edit and link content especially suitable for collaborative authoring.
Workflow	Workflow is the automation of business processes, during which information is passed along the system according to a set of rules. www.wfmc.org
XML	eXtensible Markup Language is a widely used, standardized data exchange format (http://www.w3.org/XML)

- XML-RPC A simple remote procedure call protocol which uses XML to encode its calls and HTTP as a transport mechanism (<http://www.xmlrpc.com/>).
- XPDL XML Process Definition Language is a standard of the Workflow Management Coalition (WfMC, www.wfmc.org) for the exchange of workflow definitions (www.wfmc.org/standards/XPDL.htm).
- ZOPE (Z Object Persistence Engine) Zope is an open source application server for building content management systems, intranets, portals and custom applications. The Zope community consists of hundreds of companies and thousands of developers. Zope is written in Python, a highly productive, object oriented scripting language. www.zope.org

Appendix B: Bibliography

- [Alsha04] Alshawi, Sarmand; Themistocleous, Marinos; Almadani, Rashid: Integrating diverse ERP systems: a case study. In: The Journal of Enterprise Information Management Volume 17, Number 6, 2004. Emerald Group Publishing Limited, pp.454-462.
<http://www.emerald-library.com/Insight/viewContentItem.do?contentType=Article&contentId=1529220>, viewed on 2005-11-23
- [Basil05] Basil Argasosy: OfBiz An Insider View, 2005-09-01
<http://ofbizwiki1.go-integral.com/Wiki.jsp?page=OFBizInsiderTutorial>, viewed on 2006-01-16
- [Bern01] Bernroider, Edward; Koch Stefan: ERP selection process in midsize and large organizations. In: Business Process Management Journal, Vol. 7 No. 3, 2001. MCB University Press, pp. 251-257.
<http://www.emerald-library.com/Insight/viewContentItem.do?contentType=Article&contentId=843480>, viewed on 2005-12-01
- [Cater05] Carter, Jason: GNUe Forms: A Developer's Introduction 0.5.4
<http://www.gnuenterpise.org/tools/forms/docs/Developers-Guide.pdf>
2005-02-14, viewed on 2005-10-25
- [Chalif04a] Chalifour, Josh: TEC Talks to the Open For Business Project- Free and Open Source Software Business Models- Part One: OFBiz. in: Technology Evaluation Centers, 2004-09-07
<http://technology-evaluation.com>, viewed on 2006-01-19, registration required (and recommended)

- [Chalif04b] Chalifour, Josh: TEC Talks to the Open For Business Project-Free and Open Source Software Business Models- Part Three: Compire. in: Technology Evaluation Centers, 2004-09-07 <http://technology-evaluation.com>, viewed on 2006-01-19, registration required (and recommended)
- [Chan04] Chan, Tzu-Ying; Lee, Jen-Fang : A Comparative Study of Online User Communities Involvement In Product Innovation and Development. National Cheng Chi University of Technology & Innovation Management, Taiwan http://opensource.mit.edu/online_papers.php, as viewed on 2006-01-10
- [Chen05] Chen, Si: Open For Business in a Nutshell, 2005 http://www.opensourcestrategies.com/downloads/ofbiz_nutshell.pdf, as viewed on 2005-12-22
- [Chun04] Chun-Chin Wei, Chen-Fu Chien and Mao-Jiun J. Wang: An AHP-based approach to ERP system selection. In: International Journal of Production Economics, Volume 96, Issue 1 , 18 April 2005, Pages 47-62
- [Dave98] Davenport, T.: Putting the enterprise into the enterprise system. In: Harvard Business Review, July-August 1998, pp. 121-131.
- [Deld05] Deldycke, Kevin: ERP5 Tutorial: Develop your own ERP with ERP5 Business Templates 0.9 http://www.erp5.org/sections/documentation/articles/erp5_developer_tutor3829/downloadFile/file/Tutorial-Kevin-en.pdf 2005-02-02, viewed on 2005-12-16

- [Dethle05] Dethlefs, Rolf: ERP, CRM und Projektmanagement mit Open Source Software: Unternehmenskritische Open-Source-Software bei der Probusiness AG.
<http://www.heise.de/open/artikel/68259/>, 2006-01-13
- [Elli04] Elliott, Margret, Scacchi, Walter: Mobilization of Software Developers: The Free Software Movement. University of California, Irvine, 2005-04
http://opensource.mit.edu/online_papers.php, as viewed on 2006-01-10
- [Everd00] Everdingen, Y.V.; Hillergersberg, J.V.; Waarts, E.: ERP adoption by European midsize companies. In: Communications of the ACM, Vol. 43 No. 4, 2000, pp. 27-31
- [Ferg04] Ferg, Stephen: Python & Java: a Side-by-Side Comparison. 2004-02-07
http://www.ferg.org/projects/python_java_side-by-side.html, as viewed on 2005-11-11
- [Fisch04] Fischer, G.; Giaccardi, E.; Ye, Y., Sutcliffe, A.G.; Mehandjiev, N.: Meta-Design: A Manifesto for End-User Development. In: Communications of the ACM, September 2004, Vol. 47. No. 9, pp. 33-37
- [Gamma95] Gamma, Erich; Helm Richard; Johnson Ralph; Vlissides John: Design Patterns: Elements of Reusable Object-Oriented Software Addison-Wesley Professional, Boston... 1995

- [Gold04] Golden, Bernard: Succeeding with Open Source Addison-Wesley Professional, Boston... 2004
the Open Source Maturity Model presented in this book is also available online:
<http://www.navicasoft.com/pages/osmmoverview.htm>, viewed on 2005-12-22
- [Gold05] Golden, Bernard: Community: The Difference between Friend and Faux Open Source.
<http://www.navicasoft.com/Newsletters/August%202005%20Newsletter.htm>, as viewed on 2006-01-10
- [GNUe02] GNU Enterprise Brochure
<http://www.gnu.org/software/gnue/brochures/Brochure.pdf>
2002-08-13, viewed on 2005-10-24
- [Harr00] Harris, Russ: Customization versus Standardization: Striking a Balance in ERP Software. In: Machine Design, Jul. 20 2000, Vol. 72, Iss. 14, pp. 64- 67
- [Howi05] Howison, James; Crowston, Kevin: The perils and pitfalls of mining SourceForge. 2004 - 05
<http://opensource.mit.edu/papers/howison04msr.pdf>; as viewed on 2006-01-10
- [Hyos05] Hyoseob, Kim; Boldyreff, Cornelia: Open Source ERP for SMEs.
<http://eprints.lincoln.ac.uk/67/>, as viewed on 2005-12-12
- [Kay98] Kay, Emily: Going global with ERP, July 1 1998
http://www.itmanagement.earthweb.com/erp/article.php/11072_603341_2, as viewed on 2006-01-3

- [Jones05] Jones, David, E.: Getting and using OfBiz, 2005-11-11
www.ofbiz.org/docs/GettingAndUsingOFBiz.pdf, viewed on 2006-01-17
- [Krumb05] Krumbein, Thomas: Open Source Software einsetzen und integrieren (German). Galileo Press, Bonn 2005.
- [Laur04] Lauren, Andrew M. St.: Understanding Open Source and Free Software Licensing
O'Reilly Media, Inc.; August 2004
- [Lomb05] Lombardo, Marco; Oon, Redhuan D., Shen, Peter: Compilo Compiere Developer Documentation, 2005
<http://sourceforge.net/projects/compilo>, as viewed on 2005-10-10
- [Mata95] Mata, Francisco J.: Information Technology and Sustained Competitive Advantage: A Resource-Based Analysis. In: MIS Quarterly; Dec. 1995; pp.294-505; ABI/INFORM Global
- [Matus03] Kategorisierung von Open Source Projekten: Aufbau und Ablauforganisation. (German), Master Thesis, University of Economics and Business Administration, Vienna, 2003
http://opensource.mit.edu/online_papers.php, as viewed on 2006-01-10
- [Mend04] Mendham, Tim: Open For Business. in: CIO Australia, 2004-11-09
<http://www.cio.com.au/index.php?id=318248421>, viewed on 2006-01-18
- [Mill03] Miller, Joaquin; Mukerji, Jishnu et. al.: MDA Guide Version 1.0.1
<http://www.omg.org/docs/omg/03-06-01.pdf>
2003-06-12, viewed on 2005-12-14

- [Müll05] Müller, Reinhard: GNU Enterprise Application Developer's Guide
Ed. 0.4.3.99
<http://www.gnuenterprise.org/tools/appserver/docs/manual/devguide.pdf>
2005-10-11, viewed on 2005-10-26
- [Pink05] Pink, Kathy; Janke, Jorg; Wassmer, Anne: Compiere User
Manual Version 2.52e, www.compiere.org, 2005
- [Rama00] Ramanathan, J.: Process-based architecture for back office:
successful supply chain requires EAI, ASCET, Vol. 2, No. 253,
2000
- [Raym00] Raymond, Eric: Why Python? In: Linux Journal, 2000-05-01
<http://www.linuxjournal.com/article/3882>, as viewed on 2005-11-11
- [Sarang02] P.G. Sarang, Christopher Browne, Dietrich Ayala, Vivek Chopra:
Professional Open Source Web Services. John Wiley & Sons Inc.,
2002
- [Silver01] Silverstone, Len: The Data Model Resource Book, Vol. 1: A
Library of Universal Data Models for All Enterprises
John Wiley & Sons Inc., 2001
- [Smets02] Smets-Solanes, Jean-Paul: ERP5: a technical Introduction
[http://www.erp5.org/sections/documentation/articles/linuxtag.html/
view](http://www.erp5.org/sections/documentation/articles/linuxtag.html/view)
2002-04-26, viewed on 2005-10-21
- [Smets04] Smets-Solanes, Jean-Paul: ERP5: Mission-critical ERP/CRM with
Python and Zope
<http://pythonology.org/success&story=nexedi>
2004-06-02, viewed on 2005-12-16

- [Song01] Songini, Marc L.: To Customize or Not?
In: Computerworld, September 3, 2001, pp. 42-43
- [Stoy05] Stoy, Guido: Pro und contra Open-Source-ERP (German)
http://www2.computerwoche.de/index.cfm?pageid=255&artid=73079&main_id=73079&category=160&currpage=1&type=detail&kw=
2005-03-29, viewed on 2005-10-21
- [Sum05] Sumner, Mary: Enterprise Resource Planning, Pearson Prentice Hall, 2005
- [Telt00] Teltumbde, Anand: A framework for evaluating ERP projects
In: International Journal of Production Research, 2000, VOL. 38, NO. 17, pp. 4507-4520
- [Tomps05] Tompson, James: GNUe Common: A Developer's Introduction 0.5.15
<http://www.gnenterprise.org/tools/common/docs/Developers-Guide.pdf>
2005-06-05, viewed on 2005-10-25
- [Vittie01] MacVittie, Lori: Buckle Up: Implementing an ERP Takes Time and Patience. In: Network Computing
<http://www.networkcomputing.com/1206/1206ws2.html>, as viewed on 2005-11-25

Appendix C: Surveys

Two surveys concerning decision making criteria and advantages of open source ERP systems are introduced to further strengthen the argument for flexibility and other evaluation criteria.

C.1 Decision Making Criteria: A Survey

Bernroider and Koch [Bern01] analyzed the decision-making criteria in the ERP selection process of 21 Austrian small and medium enterprises. The high importance attributed to fit with business procedures, flexibility and costs has also been found in another study of European mid size companies [Everd00, as cited in Bern01].

This is the result of their survey in ascending order of the last column. “Sum important” which is the sum of the columns “very important” and “important”.

#criteria	very		rather		Sum
	important	important	unimportant	irrelevant	
12 Customer and supplier needs	5,3	10,5	26,3	57,9	15,8
11 Guidelines from a controlling company	10,5	21,1	10,5	57,9	31,6
10 Improved innovation capabilities	11,8	29,4	52,9	5,9	41,2
9 Operating system independency	15,8	26,3	52,6	5,3	42,1
8 Internationality of software	31,6	10,5	26,3	31,6	42,1
7 Increased customer satisfaction	26,3	42,1	26,3	5,3	68,4
6 Market position of vendor	16,7	55,6	11,1	16,7	72,3
5 Increased organizational flexibility	31,6	42,1	21,1	5,3	73,7
4 Process improvement	31,6	47,4	21,1	0	79
3 Short implementation time	36,8	57,9	5,3	0	94,7
2 Adaptability and flexibility	68,4	26,3	5,3	0	94,7
1 Good support	42,1	57,9	0	0	100

Table 2: ERP decision making criteria of SMEs

Criteria 1 “Good support” is treated in Chapter “[3.3 Support](#)”.

Criteria 2, 3 “Adaptability and flexibility” and “Short implementation time” are addressed in chapter “[2.4 Advantages of Using a Flexible ERP package](#)” and chapter “[3.2 Flexibility](#)”. Also criteria 4, 5, 7, 8, 9 are discussed there, some are

simply the consequences of an ERP implementation that fits to the processes. Possible reasons for the importance of Criterion 6 “Market position of a vendor” are the continuous development of the system and the support provided (see [“3.3 Support”](#) and [“3.4 Continuity”](#)).

C.2 Key Advantages of Open Source ERP: An Online Poll

The following survey was conducted on the (root) homepage of www.Opentaps.org:

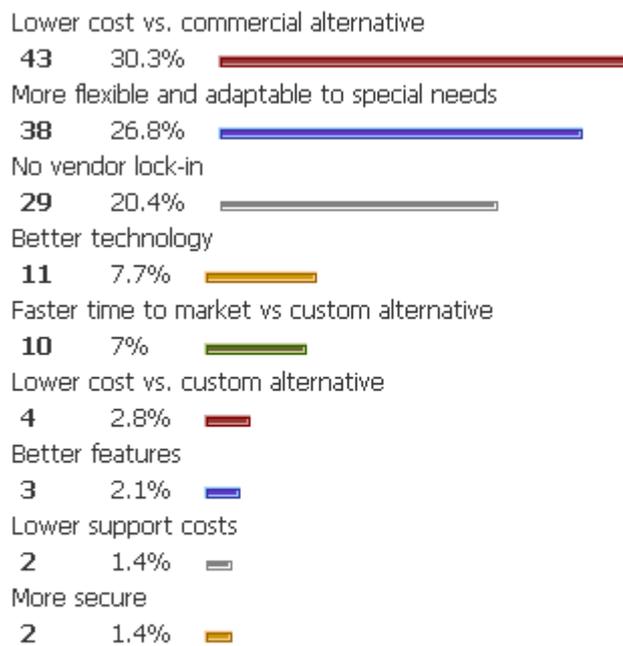
Polls

The key advantages of open source ERP to me are:

- Lower cost vs. commercial alternative
- Lower cost vs. custom alternative
- Faster time to market vs custom alternative
- Lower support costs
- More flexible and adaptable to special needs
- No vendor lock-in
- More secure
- Better features
- Better technology

Figure 16: Survey form

📊 The key advantages of open source ERP to me are:



Number of Voters : 142

First Vote : Tuesday, 08 November 2005 06:56

Last Vote : Thursday, 22 December 2005 02:36

figure 17: Survey result

As all ERP system evaluation criteria have cost influence, “*Lower cost vs. commercial alternative*” is not a general open source specific advantage.

“*More flexible and adaptable to special needs*” is covered in chapter “[3.2 Flexibility](#)”, which is the core criterion of this work. The same applies to “*Better technology*”.

“*No vendor lock-in*” is addressed in chapter “[3.4 Continuity](#)”.

Appendix D: Challenges for Open Source ERP Systems

Issues arising for open source systems are described here. Most arguments introduced in this chapter are widely dependent on the specific ERP System, whether it is open or closed source. This chapter is based on [\[Stoy05\]](#).

In the ERP area the use of open source systems is not that common. There are several reasons, why this is the case:

- They have a relatively **small user base** and just a small part of them has the capabilities and the interest to be actively engaged in development.
- The **requirements** of different industries **vary widely**. Systems that try to solve all issues are overly complex.
- There are specific **national requirements** for controlling, accounting, interfaces to public authorities, interfaces to banks, language translation. In the case of international organizations these requirements must be met for all countries they operate in.
- Organizations need **reliable development, support** and **training** for mission critical systems, and some systems are not ready for production.
- **Cost for customization and maintenance** is an important factor apart from license costs.
- Organizations have to take care about possible (legal) licensing issues.
- Big players have little strategic interest in open source ERP systems.
- Low **reputation and awareness** due to limited marketing capabilities.
- Open source systems have gained wider acceptance in a technical audience than in the business community.
- It is hard to find **reference customers**, because they do not want to be bothered by the large open source interest group.
- The **documentation** might be **incomplete** or not up to date.

The arguments for open source compared to black-box closed source ERP systems are:

- Direct access to the code makes the systems easier customizable, provided that there are procedures for applying customizations to new

updates of the core version.

- Development **process and the bug list are transparent** to the user.
- Custom functionality can be shared and integrated back to the core version. So the introduction of new **functionality is driven by real user needs** and less by marketing thoughts. There is more direct communication with the developers.
- **Continuous development** of the Software. There is a consolidation in the ERP market. Customers might be forced to follow the vendor product strategy or migrate to a new product line. These possible problems are diminished with the help of:
- Independence from ERP vendor strategy. Open source reduces vendor lock-in.

Appendix E: Model Driven Architecture MDA

This is a different paradigm for software development promoted by the Object Management Group⁸³(OMG). [\[based on Mill03\]](#) The software is modeled with UML (Unified Modeling Language) diagrams. The code is generated automatically, just additions and customizations are coded manually. In the open source area there are several MDA frameworks. All are intended for developing enterprise web applications and do not have special ERP/accounting focus. Following is a selection of a few well known frameworks:

- androMDA - www.andromda.org
- openArchitectureWare - www.openarchitectureware.org
- openMDX – www.openmdx.org

All Projects are Java J2EE-based and offer professional training, consulting and support. The proof of concept application for openMDX is the popular enterprise class open source CRM solution openCRX⁸⁴ which also provides some ERP functionality.

⁸³ <http://www.omg.org>

⁸⁴ <http://www.opencrx.org>, as viewed on 2005-11-17 it has more than 100000 downloads from www.sourceforge.org

Appendix F: Other Open Source ERP Systems

Here is a collection of links to other open source ERP systems, which will not be further addressed. More systems can be found on www.sourceforge.org.

- www.aria-erp.org – ARIA Accounting Receiving and Inventory Administration.

This project was forked from NOLA in 2002. It is LAMP (Linux/Apache/PHP/MySQL) based and licensed under GPL. It handles accounting, receiving and inventory management and payroll. Its state is beta and there has been no CVS commit in the recent 21 months.

- www.ck-erp.org – A LAMP-based ERP system built upon PHPGroupWare (www.phpgroupware.org). It is in beta state and licensed under GPL. An online demo is available. The project is reviewed in www.linuxgazette.com/node/9782. The files are hosted at <http://sourceforge.net/projects/ck-erp/>. An active forum is hosted by <http://groups.google.com/group/CK-ERP-en>.

- www.evaluate.org – A platform and database neutral modular ERP system developed in Delphi/Kylix, Java, PHP. Its state is beta and there were no changes in the recent two years. The modules available are:

- General Ledger System for multiple companies.
- Inventory Management for Manufacturing and Retailing linked to the GL System
- Purchase-Receiving-Sales systems for point-of-sales operations
- Sales order processing and
- Customer Relationship Management.

- www.fisterra.org – Fisterra is a GNOME development framework for the implementation of ERP systems.

- www.nakedobjects.org – a framework for developing business applications in Java

- nola.noguska.com – This is the basis project of ARIA, it is not actively developed, the latest CVS change is from 2002. The company sells a

professional version.

- sourceforge.net/projects/kontor/ - Linux- Kontor: ERP system with Java based n-tier architecture. No CVS commits in recent years. There are just advertisements on the original homepage. German documentation is available⁸⁵.
- <http://www.taika-informatique.com/pgi.php> – Another LAMP based ERP system. All available information is in French⁸⁶.
- WebERP

www.weberp.org – LAMP based ERP system licensed under GPL.

The following modules are available: Order Entry, Accounts Receivable, Inventory, Purchasing, Accounts Payable, Bank, General Ledger.

Flexibility:

WebERP offers role based security management.

It is localized for English, Brazilian Portuguese, Chinese, Indonesian, Polish, Russian, Spanish, Spanish (South American) and Spanish (Venezuela).

Support:

Support is available through mailing lists and a partner network in US, CN, UK, South Africa and Pakistan.

The project is well documented, a demo is available online.

Maturity and Continuity:

The community is very active.

- www.wyatt-erp.com – ERP system developed in Tcl.

- gnuCash

www.gnucash.org

sourceforge.net/projects/gnucash/

Programming Language: C, Scheme, Perl

Database: XML-file for single user, PostgreSQL for multiple users

Platforms: Linux, Unix, Mac OS

License: GNU General Public License (GPL)

⁸⁵ <http://www.linux-magazin.de/Artikel/ausgabe/1999/08/Kontor/kontor1.html>, viewed on 2005-11-23

⁸⁶ http://solutions.journaldunet.com/0407/040723_panorama_erp_open_source.shtml, viewed on 2005-11-23

Status: Beta/Stable

Translations: English, Czech, Italian, Russian, German, Portuguese, Dutch, French, Spanish, Slovak, Chinese, many other languages are partly translated

Fat client, also for multiple sites, well documented especially in English, French, Spanish and Portuguese.

[\[Krumb05, pp. 459-471, with additions\]](#) **This is not a full ERP System.** It is a standalone, out-of-the-box accounting solution. If you want to integrate it with your processes: look somewhere else! It is also part of Gnome Office, which is included in many Linux distributions. The target group is home users, but it is also suitable for accounting purposes in micro enterprises. It is an open source clone of Quicken⁸⁷, a very popular accounting software and it is easy to migrate from Quicken or Quickbooks to GnuCash. Special features for the German market are Bank interfaces like HBCI and a DATEV interface.

- AvERP

www.synerpy.de

Programming Language: Delphi, Kylix

Database: Interbase, Firebird

Platforms: Windows, Linux (beta)

License: proprietary, **not OSI compliant**

Status: Stable

Translations: German

About 450 customers in Germany.

All logic and design is meta data based and stored in the database. The product is targeted to small and medium enterprises.

⁸⁷ www.intuit.com

Appendix G: GNU Enterprise Developer Tools

The Developer Tools are divided into several sub-projects:

Developer Tools Sub-Project	Version ⁸⁸
Common Library	0.6.1
Forms	0.5.13
Reports	0.1.8
Application Server	0.4.3
Navigator	0.0.9
Designer	0.5.7
Bayonne Telephony 2	1.0.1

Table 3: GNUe Developer Tools and Versions

GNUe can be run in client-server mode using Common Library and the wanted tool together, or in n-tier mode using the Application Server as middle tier.

G.1 Common Library

This chapter is a summary of [\[Toms05\]](#). This is the basis of all development tools and provides the database abstraction layer, the XML-to-Object mapping and Remote Procedure Call (RPC) abstraction layer used by Forms and Reports.

The detailed features are:

- The database abstraction layer supports most major databases, allows mixing of data sources/ databases and is configured in XML.
- The RPC (remote procedure call) abstraction layer allows the definition of public methods once they are on the server and makes them available in CORBA, XML-RPC, SOAP, and DCOM clients.
- A trigger system allows easier customization of the system using python methods with access to all Application Server managed objects, custom functions and runtime parameters. Triggers can be fired on a certain level under a certain condition:

⁸⁸ as published on 2005-10-25, Version numbers above 1.0 are production versions

Trigger level						Trigger condition
Form level	Block/record level	Field level	Page level	Entry level	Button level	
X						On-Startup
X						On-Activate
X						On-Exit
X						Pre-Commit
X						Post-Commit
	X					Pre-Query
	X	X				Post-Query
	X	X				Pre-Change
		X			X	Post-Change
		X				Pre-Modify
	X	X				Pre-Insert
	X	X				Pre-Update
	X	X				Pre-Delete
	X	X				Pre-Commit
	X	X				Post-Commit
	X					On-NewRecord
	X	X	X	X	X	Pre-FocusIn
	X	X	X	X	X	Post-FocusIn
	X	X	X	X	X	Pre-FocusOut
	X	X	X	X	X	Post-FocusOut
					X	On-Action

Table 4: GNUe trigger levels and conditions

- Integrated debugger allows Python debugging, profiler, debug levels and log files.
- An internal event system allows the registration of and the reaction to events.
- Automatic document generation.

G.2 Forms

This chapter is a summary of [\[Cater05\]](#). This is an user interface generator based on XML definitions. With one definition the User interface can be generated for several physical platforms:

- Fat Client GUI with native look and feel for Windows, Linux/Unix,

MacOS and OS/2 based on wxPython⁸⁹

- HTML Web interface
- Console/Terminal interface
- Telephone response system

Triggers are used for creating data verification and workflow definition. The graphical elements provided by Forms are: button, label, field, image, tree, box (draws a box around grouped elements), page (tabbed or pop up), menu, tip (provides explanation when mouse moves over element), scrollbar, pull down menus and detail tables.

The form component ties database tables to graphical elements via the logical constructs data sources, blocks and fields.

A data source consists of

- Name (data source name)
- Connection (connection to database)
- Table
- Condition (SQL where clause)
- If the data source is the detail source of a master-detail relationship, then
 - Master (name of master data source),
 - Master link (name of the field/fields that link to the detail data source) and
 - Detail link (name of field/fields that link to the master data source) need to be provided.

A block is the display equivalent to a data source. It consists of

- Name (block name),
- Datasource.

A field consists of

- Name (logical field name on UI)
- Field (corresponding field name on database)
- Case [mixed | upper | lower] defines if input characters of the field are converted.

⁸⁹ www.wxPython.org, wxPython is a wrapper to the portable C++ GUI class library wxWindows, which provides an interface to platform-specific GUI libraries.
<http://www.python.org/doc/faq/gui.html#wxwindows>

G.3 Reports

GNUe Reports is a platform and output-independent reporting system. It reads an XML-based report definition and generates arbitrary XML output. This output is then transformed into the required format using normal XML tools.

Text, HTML, CSV and Label Stock are currently supported, PDF, Postscript and Gnumeric⁹⁰/Excel are under development. The output can be generated as data file, e-mail attachment, printer output or fax via HylaFax⁹¹ server.

G.4 Application Server

This chapter is a summary of [\[Müll05\]](#). The application server is the core of GNUe and manages database communication, feeds the frontends generated with Forms and Reports Tools and calls necessary programs and scripts. It handles both data storage and the business logic.

The Application Server consists of two abstraction layers which are defined by XML files:

Layer	XML file
Database Abstraction Layer	GCD GNUe Class Definition
Form (User Interface) Layer	GLD GNUe Language Definition

Table 5: GNUe Application Server Abstraction Layers

G.4.1 GNUe Class Definition

GNUe classes⁹² are the application server equivalent to Database tables and related business logic. **GCD (GNUe Class Definition)** files describe the namespace (module), classname and properties in XML-Format:

⁹⁰ Gnumeric is part of Gnome Office

⁹¹ www.hylafax.org

⁹² GNUe class is not the same as class generally used in object oriented programming literature, e.g. inheritance is not supported.

```

<module name="address">
  <class name="country">
    <property name="code"    type="string(2)" comment="ISO-Code" />
    <property name="name"    type="string(35)" />
  </class>
  <class name="person">
    <property name="name"    type="string(35)" />
    <property name="street"  type="string(35)" />
    <property name="zip"     type="string(8)" />
    <property name="city"    type="string(35)" />
    <property name="country" type="address_country"
                                comment="reference to class country"
    />
  </class>
</module>

```

Listing 1: a sample "address.gcd" GNUe Class Definition file

Database term	GNUe Class Definition term
Schema	Module
Table	Class
Field	Property

Table 6: GNUe Class Definition terms

Reference properties are defined by using *module_class* as type. E.g.:
 type=address_country.

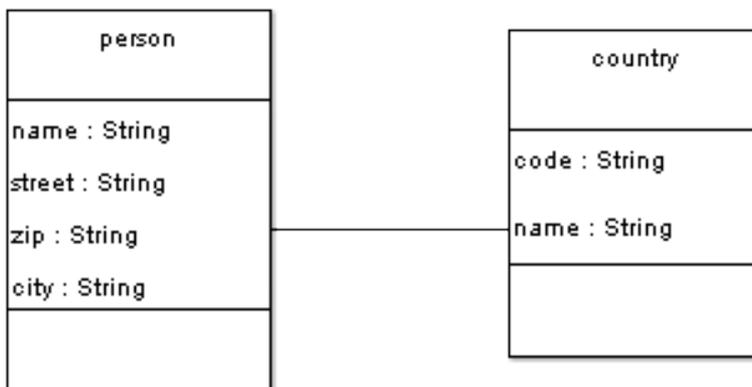


Figure 18: Class diagram of address.gcd GNUe Class Definition

Procedures are defined with the <procedures> tag.

```
<procedure name="OnChange" >
  <![CDATA[
    if propertyName == 'address_zip':
      if newValue and int (newValue) < 10000:
        abort ('ZIP codes must be less than 10000')
  ]]>
</procedure>
```

Listing 2: GNUe Trigger procedure definition

If a procedure name attribute has the name of a **trigger** condition (OnInit, OnChange, OnValidate, OnDelete), it is executed automatically when the condition occurs.

GCD Procedures can also be called from form triggers defined in the GNUe Form Definition (GFD) file.

Global Objects that can be accessed in all procedures are:

- **self** allows access to the object instance the procedure was called from. E.g.: self.zip accesses the zip code of the actual person.
- **session** represents the connection to the GNUe Application Server. It has following methods:
 - **new** (classname) creates and returns an instance of a class.
 - **get** (classname, objectid) returns an existing instance of a class
 - **find** (classname, condition, sortorder, properties) returns a list of matching instances. It supports sophisticated SQL-like queries.
- **abort** (errormessage) aborts the current procedure.

Virtual aka **calculated properties** are defined within the <properties> tag:

```
<property name="czc" type="string(47)" >
  return self.country.code + ' ' + self.zip + ' ' + self.city
</property>
```

Class definitions are published to the Application Server with “gnue-readgcd address.gcd”. The form can be viewed with “gnue-forms appserver://appserver/form/address_person”, providing following initial output:

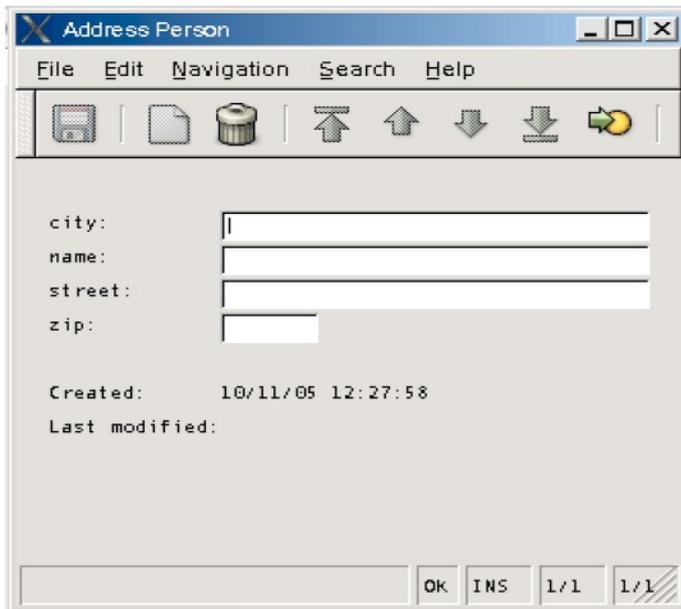


Figure 19: Initial form generated from "address.gcd "
GNUe Class Definition

G.4.2 GNUe Language Definition

The user interface can be refined with the help of **GNUe Language Definition** – **GLD** files:

```
<module name="address" language="En">
  <class name="country"    label="Country Codes">
    <property name="code"  pos="100"  info="200" label="ISO Code" />
    <property name="name"  pos="200"  search="100" label="Name" />
  </class>
  <class name="person"    label="Address management">
    <property name="name"  pos="100" label="Name" />
    <property name="street" pos="200" label="Street" />
    <property name="zip"   pos="300" label="Zip-Code" />
    <property name="city"  pos="400" label="City" />
    <property name="country" pos="500" label="Country" />
  </class>
</module>
```

Listing 3: a sample "address-En.gld" GNUe Language Definition file

The **search** attribute of the <property> tag indicates which properties are shown as drop down menu in the master form. In this case country->name is shown in the “Adress management” form as country property. The **info** attribute of the <property> tag is handled like search but not as drop down but as simple informative field, changing according to the drop down menu.

Internationalization issues are solved with GLD files: If you need translated user interfaces you need to define a separate GLD file for each language. Custom language/country specific behavior can also be defined at this level.

GNUe Lanaguage Definitions are published to the Application Server with “gnue-readgld address-En.gcd”.

The form changes to:

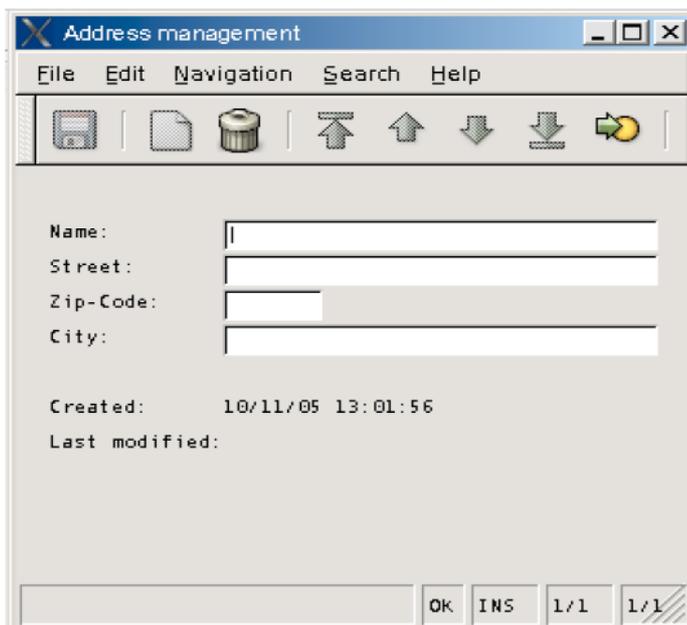


Figure 20: Refined form defined in "address-En.gld" GNUe Language Definition

G.5 Navigator

This is the user frontend for GNUe displaying tree-based menus and calling forms and reports based on a XML process definitions. Navigator uses Forms or Reports to run the requested task.

G.6 Designer

This is the IDE (Integrated Development Environment) of GNUe and it is used for testing and developing forms.

G.7 Bayonne Telephony

This module is used for telephony integration.