

## Compiere as an example of ERP open source solution

### Open source can be a compromise between purchase and application development

When the management is expecting implementation of a new system for support of business processes, it has to be decided: either to buy a new system or to develop its own one. Both decisions surely have their advantages and drawbacks.

Purchase of a new system appears to be a more comfortable and sure variant. For development of a new system it is necessary to have own IT infrastructure or to substitute it by quality outsourcing. In addition it is well known that too many projects of software development finish without success.

Otherwise it is often criticised that accomplished products do not fill implementation goals. These useless software investments hardly ever cover original requirements. The independent software development is a more difficult task. But if successfully accomplished, the user mostly gets software applications which he had chosen and defined.

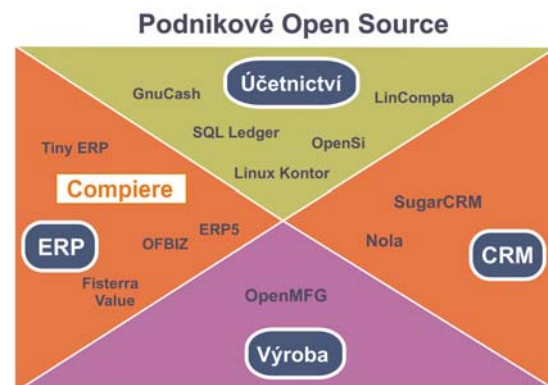
Open source could be in the role of a compromise between the both variants. Project development does not have to be done from the very beginning, because in case of quality documentation the development team can very quickly adopt the software architecture and methods of concrete open source development and fulfil needs of its user. Furthermore appropriate license politics reduces the project cost. Development is also cheaper because the development team does not have to create lower technological levels and can be concentrated on user and managements needs.

All above mentioned means just one point of view concerning open source software. This fact is so important, that it is worth understanding open source software in this context and evaluating it according to the possibility to fulfil the space between commercial products and own

development. It is applicable for example for small and medium size enterprises, which do not want to invest to big solutions and which have good specialists for own realization of their goals.

### ERP open source solutions represent nowadays reality

In the field of ERP applications many open source projects have grown up since 1990's. Some of them are so advanced that they can create real competition for standard commercial products. There are available either solutions for small consumer or medium size companies. In difference from technological projects the competition for so called big solution. It means products of kind of SAP, Siebel, etc. has not been still created, mainly due to various commercial reasons.



One of the most known ERP applications is certainly the system Compiere. Besides typical ERP requirements as accounting, warehousing, purchasing Compiere covers also requirements for CRM and E-business.



### **...and the Compiere solution**

*Compiere can demonstrate advantages and drawbacks of open source solutions from the point of view of a medium size company.*

### **Maturity, references and development manner are traditional arguments against open source**

Instability of product during a longer term, not fully defined development conception, incompleteness of projects is often criticised. It is probably caused by the fact that the many open source projects are developed as a marginal product of an enthusiastic community more than sophisticated calculated business model.

### **...and the Compiere solution**

*Compiere appeared at the end of 1990's, created by the only author Jorg Janke who had rich experience with similar ERP projects. He was in the position of technical director of Oracle in Europe and he also wrote modules of ADV/ORG which influenced the beginning of R3/ SAP product.*



*In 2000 the company Good Year sponsored the first live version of product and since then company Compiere, Inc., controls next software development. The new business and development model caused that the software was one million times downloaded and it is supported by dozens worldwide partners and different industrial solutions are running all over the world. Application is localized in different countries with different legislation. Application supports GAAP, Europe standards, and various legislation and custom rules in more exotic countries as Japan, China, South Africa. The system is being supported also in the Czech Republic and Slovakia.*

*Compiere, Inc., managed by Jorg Janke, strongly defines concept of further development. Features are also influenced by VIP users and contributors, but anyway Compiere is the main coordinator of the whole process. Several branches of software appeared; they were motivated either by advanced functionality or by technological innovation, but the question of future compatibility with the main stream is still open.*

### **Commercial products could be inspired by architecture and technology of open source solutions.**

The architecture and technology of open source systems are traditionally the strong point of open source solutions. Due to declared and also maintained principals of open solution the open source community uses such technologies, which are open, widely respected, known and standardized. Open source may influence commercial products in such way and show them what the next releases of commercial products

should involve. Due to public discussion of community non perspective ways of development are rapidly removed and technologically pure and faithful solution remains.

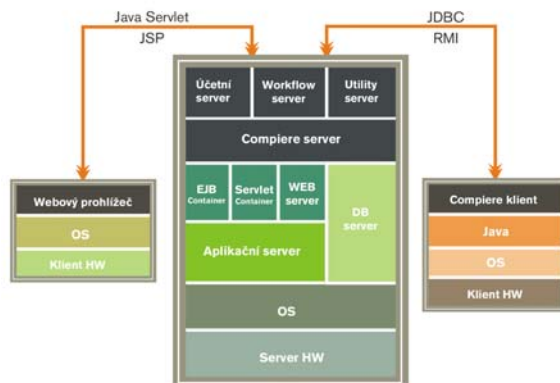
### **...and the Compiere solution**

Compiere is 100 percent Java solution based on Oracle technologies and J2EE standards. The architecture is as for platform independent and in the last release of software the independent database layer was modified to support many databases: Sybase, Microsoft SQL, IBM DB2, Postgres.

Compiere can run on many platforms: Linux, Windows, OS X, Solaris. It means on the server and client side, too.

Client is completely created in Java and it communicates with the database by JDBC protocol, with the application server by RMI methods. The alternative HTML client could be use for smaller volume of processing and it has almost full functionality like a Java client. Advantageous use of this client is the case where the installation of fully featured client is not possible or suitable. For example customers could use part of the information system of client or is possible that remote connected employee can use this HTML option.

The application server contents business logics and it is still implemented in open source J2EE application server JBoss. Other J2EE servers are planned.



The Structure of the whole system is implemented with huge independence of particular objects. Application logic is

strictly separated from the presentation layer and is described in so called "Rule engine". The originally closely connected processes are separated from one another and processed by asynchronous way and the system checks the finishing of all processed branches. Only the main task influences the work of user. Subsidiary faulty processes do not block of user. The user and the administrator are alerted by system as for faults, but the system can continue processing. This concept is called "Fail-safe transaction".

### **Implementation and maintenance cost are highly important**

More or less successful discussion on economic advantage of open source is being held from the very beginning. Type of open source licence of course is very important. But generally we can say that future cost savings are the most attractive. So Open Source offers real ownership of software that will not be charged with any fees in future. Te user also won't be restricted by any future use limitation. Real open source gives first of all the liberty of future.

For a strong company offering a commercial product can't be a problem to offer such business conditions which can be compared with cheaper Open Source variants in the time of purchase, but there are also disadvantages of commercial solution. The cost of further implemented service and software fees will never be as advantageous as in the time of purchase. The same is valid for other conditions of software use. For example the user will never have the freedom of using the software code, the possibility of replacement of software use according his future needs, the possibility to change the service supplier and so on. It would be of course against any logics of software houses.

### **...and the Compiere solution**

Compiere is distributed under Mozilla license (MPL 1.1). It allows the user to use it for himself, his partners, and clients. The application can be modified and these changes are allowed to be sold without any duty towards the open source community. Some license conditions of distribution must be observed (for example using of

*project name, adding license text to sources, etc).*

*But Compiere also includes next hidden costs. One of them is still big dependence on commercial database Oracle. Although Compiere supports Oracle Express, this variant is not sufficient for real installation. Compiere offers migrating tools included in annual fee. Without these tools client can hardly follow the actual version. Due to used technologies the application of Compiere requires a system specialists for Oracle and partly also for application server.*

*In spite of the fact that Compiere specialists say, that the system implementation takes just a few days, we can hardly agree, because the system is so complex that its implementation is not possible without deep knowledge of logics and functionality. In any case the client must become aware that it is not just the simple system installation but the company own processes must be compared with preset once in application very thoroughly and their modification, parameterization, adaptation for concrete user must be taken in account.*

### **Business model and application functionality are surprisingly in close link**

The company that decides to develop open source applications expects returns of investments in different way than for license sale. There are many business models that deny this statement. For example the user must pay for advanced functionality that is not included in the basic edition of open source software. Or the user must pay for upgrades of application. In other case the user pay for wider use of application, based for example on users connections, and so on.

But we would like to notice those cases, when company ignores any fees and concentrates on service providing. Of course it uses its deep author's software knowledge and by providing software services it reaches very good ratio performance / price. It contents such services as software localization, software tailoring, data exports, and software integration. The author of ERP system is

likely the expert for software development not specialist in software implementation by different clients. This weakness is better eliminated by commercial products. There is clear differentiation between implementation part of the whole process and the development core of company.

Commercial product had mostly time and especially motivation to understand better specialised needs of their customer. Good understanding of nowadays trends in requirements for application is survival factor for companies producing commercial software. The end user stands closer to these companies; responses for users are quick and decisive for further software development.

Open software projects miss this strong and close relationship influencing software development. The result is that open source projects user's functionality is not so perfect like functionality of commercial projects forced by daily users needs.

Open source authors respond to different fact by adding such tools to their software, that users themselves can easily or with help of experts change required features. Of course the system implementation becomes more complicated and expensive this way.

### ***...and the Compiere solution***

*That is the case of Compiere, when the application framework is the integral part of system and the whole application is developed by this framework. The user can add whole modules with minimal risk of application destabilisation in standard way and user profits of standard user's environment and standard security model and standard databases.*

*The concept of so called application dictionary assures quick changes and fluent run of application.*

*Accounting server and workflow server are at disposal for the user and they provide planning and processing, as for accounting record and warehousing transaction or processing of customer's requirement, mail reaction, alerts or document printing. Client works with the system by means of personalised environment. User's environment is generated on the basis of rules and therefore the application*

is stable and consistent. Time-demanding user's inputs are replaced by the defaults and by selections from once entered data.

User can reorganise input form, change fonts, field names or hide unimportant data or add specific helps. He can also add new fields validation criteria. User's access to data and processes are controlled by elaborated security model.

Compiere product includes processes in unified application more than the set of traditional independent ERP modules. Thanks to supported CRM features Compiere is flexible enough to be used for various businesses.

**Klasické rozdělení ekonomických modulů versus procesní pojetí Compiere**

MODULY	OD NABÍDKY K PLATBĚ	OD POPTÁVKY K PLATBĚ	SPRÁVA ZÁKAZNÍKŮ	SPRÁVA PARTNERŮ	ZÁSOBOVÁNÍ	ANALÝZA VÝKONNOSTI
Hlavní kniha						X
Kniha závazků		X			X	X
Kniha pohledávek	X		X			X
Nákupní objednávky		X	X		X	X
Prodejní objednávky	X		X		X	X
Skladové hospodářství	X	X	X		X	X
Základní prostředky						X
CRM	X		X	X	X	

Today open source covers already different software markets. ERP/CRM market is not an exception and open source solutions are real competitors for advanced commercial products. If they are still understood just as alternative business possibilities it is not valid in technological sphere any more. The open source projects are worth close watching. Even if you are not decided to implement the concrete open source solution you will find inspiration for your own software development or for choice of commercial product in these projects.