

# Software factory

## What is the purpose of a software factory?

Companies want to automate their application development further, in order to improve software quality and reduce development time. New software has to meet high requirements and be capable of being deployed rapidly. With a software factory, you structure your development process and make it manageable. Project managers can monitor the evolutions meticulously. Developers get better support. So, a software factory can provide high quality, rapid and manageable application development.

## What lies behind the name *factory*?

Just compare it to a real factory. Machines and robots are replacing manual activities. Measuring points monitor the status of production and test the quality. Precisely the same happens in a software factory. For example, by automatically generating code or working with templates you avoid manual coding work. It works faster and more correctly. A good development model allows you to check easily whether the code satisfies the objectives. Is it delivering the right results? By integrating tests regularly and right from the outset of the development process, you know quickly whether the code is in order.

## What does a software factory consist of?

A software factory is a total approach, which perfectly harmonises development tools, guidelines, templates and systems. There are no universal turnkey solutions that you can just put in with any customer. One team will place the emphasis on tools for code generation, while another will place it on quality control or project progressing. Implementing a factory is a quest for the right components for your company's challenges.



## What is the difference between a software factory and an application framework?

A *framework* is usually located in the developer's field. It is a library of code that helps the developer to work within an established structure. A framework screens him off from the complexity and the many possibilities of .NET or J2EE. A *factory* goes further. It comprises the entire development process, from needs analysis through to implementation. The aims of a factory are also broader: improving production speed and quality.

## How does a software factory support an application's life cycle?

- The first step is determining **requirements and working methods** based on templates. The information is saved centrally, with access by every team member, for example via intranet portal. Any subsequent documents such as analyses and bug reports will be saved here too. Guidelines determine how documents are controlled and how team members use them. A project manager for example defines on the basis of the analysis *iterations* and *work items* or tasks. These are integrated in a source control system, so that the developer sees them in his own development environment. He checks off finished tasks. This keeps progress perfectly controlled. A factory is therefore a win-win for the project *and* for the developers.
- A subsequent step in the factory can be **generating software code**. Metadata are needed for this. You set these out in a graphical model that you draw with tools, which you integrate in the development environment. Popular ones are for example Microsoft's *domain specific languages*. The code is generated from the model in accordance with the customer's chosen architecture and the framework used. Much of the work is automated, with a minimum chance of errors. The model has yet another important function: it simplifies communication between the developers and the people managing the requirements.

### Ordina Belgium offices

B-3001 Heverlee, Interleuvenlaan 15H • tel. +32 (0)16 27 00 80  
B-3560 Lummen, Bosstraat 52/2 • tel. +32 (0)13 38 00 00  
B-9820 Merelbeke, Guldensporenlaan 84/I • tel. +32 (0)9 280 23 90  
B-2627 Schelle, Boomssesteenweg 28 • tel. +32 (0)3 866 00 22

[www.ordina.be](http://www.ordina.be)



CONSULTING | ICT | OUTSOURCING

- In a factory, you can also set up a **continuous integration system**. The code is then in a central source code control system, such as Microsoft's *Team Foundation Server* or IBM's *Rational ClearCase*. In addition, a system runs that at regular intervals downloads the source code and compiles it into *executable code*. The test team is able to set to work on the application quickly and gives feedback via the project portal. The project manager closely monitors the quality and prepares new work items where necessary. Automation and quality go hand in hand.

### How does a software factory enhance the quality of the code?

The quality improves thanks to among others the faster progress of the development process. This therefore creates extra time for quality control. The main technique for controlling the quality is working with *unit testing*. The developer decides beforehand exactly what he is expecting from a piece of code. When testing he always works with the same data. So the expectations are established, the data too. The code is the sole variable. Should the test produce an incorrect result, then that automatically means that the code is not working as intended. Tracing the cause becomes a good deal easier. The combination of *unit testing* and *continuous integration* allows you to discover quality problems faster, and you can resolve them faster. Once again, the project portal plays an important role.

### Are there different forms of software factory?

Customers' aims and starting points differ and so the software factory models do too.

- Teams working with development in offshoring want to follow the new source code from close at hand. A central source code system is therefore very important for them.
- Customers' methodologies also determine the appearance of the factory. With an iterative Scrum methodology, the structure of the templates for documents and work items looks different to a traditional waterfall methodology.
- The choice of development platform plays a lesser role. A factory must be as independent of it as

possible. The factory approaches development from a higher abstraction level.

- The main differentiator for software factories is the service level of the applications for which you set up your factory. For big, business-critical applications, the sights in terms of quality and manageability are set a great deal higher than for small, short-life applications. You cannot burden them with a heavy factory. Just as you cannot approach an application with a high service level with a factory that is too light.

### How do you build a software factory?

It is a good idea to start from an application that has been developed without a factory. Vetting the manual development process allows you to determine how it can be more efficient. The choice of tools and methodology is very important, just as the guidance of the project teams in using them. The team members must after all apply the components of the factory correctly. Make sure that the technology is serving the people and not the other way round. Carefully structure the documents and templates, paying attention to the interaction with the underlying methodology. You can perfect test techniques, modelling and code generation at a later stage. In a parallel route, you build a source code control system and a continuous integration system.

### How can Ordina help you?

*A software factory is no out-of-the-box product. The Ordina Competence Centre Software Factories supports numerous tools, and guides customers in deciding on the factory and its implementation. Ordina also trains you to work with your software factory environment. The main tasks of the Competence Centre are R&D, product evaluation, accumulating knowledge about tools and methodologies, knowledge sharing and project support.*

ORDFAQsoffacENG1558



#### Ordina Belgium offices

B-3001 Heverlee, Interleuvenlaan 15H • tel. +32 (0)16 27 00 80  
 B-3560 Lummen, Bosstraat 52/2 • tel. +32 (0)13 38 00 00  
 B-9820 Merelbeke, Guldensporenlaan 84/1 • tel. +32 (0)9 280 23 90  
 B-2627 Schelle, Boomssteenweg 28 • tel. +32 (0)3 866 00 22

[www.ordina.be](http://www.ordina.be)



CONSULTING | ICT | OUTSOURCING