

# Implementation of enterprise software

Comparison of the methods  
"Classic" and "Ramp Up"

An essay by Thomas Schlereth, CEO of Can Do



## Table of Contents

INTRODUCTION .....	3
1. WHY DOES A COMPANY IMPLEMENT NEW SOFTWARE?.....	3
2. THE CLASSIC PROCEDURE OF INTRODUCING AN ENTERPRISE SOFTWARE TO A COMPANY .....	4
3. INTRODUCTION OF ENTERPRISE SOFTWARE ACCORDING TO THE AGILE RAMP-UP PROCESS .....	8
4. PROCEDURE OF THE RAMP-UP METHOD USING THE EXAMPLE OF THE PROJECT AND RESOURCE MANAGEMENT SOLUTION CAN DO ....	9
5. CONSIDERATION OF THE RAMP-UP METHOD.....	11
6. SUMMARIZED COMPARISON OF THE METHODS.....	12

# Introduction

One of the main activities of IT departments in companies is the **introduction of new or additional business software**.

This can be, for example, a commercial system for order management, a new document management system, a complete ERP system or even a project and resource management system.

This article describes **two methods**, how companies proceed with such a "rollout", which advantages and disadvantages the two procedures have and why which procedure is used at all.

## 1. Why does a company implement new software?

Since the 1950s, computer programs have been used in companies to carry out processes more efficiently, quickly and transparently. The first companies were often public authorities or financial companies.

The focus was mostly on financial and form solutions. The products used were not standard programs as we often use them today. Rather, the systems were custom developed.

The process of programming a complete solution based on special requirements is fundamentally different from introducing an existing standard solution into a company.

The "classical" procedure for the introduction of a standard solution described in the next chapter is however interestingly very close to the procedure for the development of an individual solution.

It is true that the first standard software for companies came onto the German market in the 1970s with the product DATEV, and SAP's R/3® was introduced to the market in 1992. Nevertheless, little seems to have changed in the process to this day.

## 2. The classic procedure of introducing an enterprise software to a company

From a project management perspective, the classic method can also be referred to as a "waterfall" project. Here, the project is divided into phases. A phase can always only begin when the preceding phase has been completed.

Essentially, the phases of a product implementation are ...

- the definition of requirements for the new product,
- the selection of a product,
- the adaptation and connection in the company and then finally
- the activation for the users.

This is occasionally followed by an optimization phase, which sometimes lasts longer than the rest of the project.

All steps are carried out strictly one after the other, just like the waterfall model in project management. If one examines this process more closely, several disadvantages can be identified:

- The definition of the processes, data and use cases to be depicted by the new enterprise software is complex.
- Due to the complexity and the lack of experience, management consultancies are often commissioned to manage or accompany the introduction.
- Furthermore, all affected parties (stakeholders) are "brought on board" - usually too many. These are, for example, all middle and senior managers of the affected departments, the IT department, the in-house lawyers (for the contracts with the providers), the purchasing department, the data protection officer, the compliance officer, and so on. Each of the stakeholders formulates requirements or, in most cases, wishes. This catalog is the basis for sifting through various providers of standard products.

After an equally long process of presentations, specifications, negotiations, testing and prototyping, the decision is made in favor of one of the providers. Then the actual rollout begins, i.e. the new system is integrated into the existing IT, data transfers are prepared or carried out, training plans are created, the application is adapted or even changes are programmed, and so on.

Then, at the very end, the users finally get their turn. They are trained and supported and finally use the product. A great deal of effort is therefore put into ensuring that the users receive a perfect solution at the end of the very long project that leaves no questions or problems unanswered.

Unfortunately, this does not work in most cases. There are a lot of studies, researches and technical papers on why this approach is potentially a very bad practice. Therefore, I would like to highlight only two human aspects.

The basic assumption is that the company's processes, as formulated in the requirements catalog, are both optimal and state of the art.

In the case of project and resource management, this means that the company knows this very well, often even better than the suppliers of such products - at least that is what the company's employees believe.

However, I can say from several years of experience that in virtually no case did the customer actually have more know-how about this topic than we do. I also cannot imagine that a company, for example, has more know-how on the subject of cost/performance accounting than a provider of standard software for precisely this specialized area.

As a result, the provider actually dictates the optimal process through its system and the company adapts. In fact, this is often formulated at the beginning of the process, in which the company emphasizes "to adhere closely to the standard". Over the course of the project, however, the pressure to change often becomes too high for the customer, and the company's own familiar processes are enforced.

Another human aspect is the enormous number of participants. At first glance, the assumption might be that the target company wants to bring in the cumulative experience of all participants to find an optimal solution.

However, I believe in another reason, and this reason lies in the "waterfall" process; more precisely in a very special transition between two phases:

The most important point in the entire project for all parties involved - vendor and customer - is the signing of the sales contract. The vendor's sales staff and the customer's representatives all work towards this point. From this point on, there is no "going back". A significant investment in licenses is made, and the companies are now tied to each other for the period of the rollout - which can take years. This is why such an effort is made up front.

The customer wants to avoid any risk of failure. Extensive assurances from the vendor, a seemingly endless number of documents, commitments and concepts are intended to bind the vendor closely to the project and oblige it to keep all promises.

In order to distribute the responsibility in case of failure among as many heads as possible, as many people as possible are involved within the customer company. This can also be referred to as responsibility diffusion: Everyone assumes that another person will take responsibility and intervene in an emergency, and no one wants to take the fall in the end.

A business solution is always introduced with the aim of generating a benefit. It follows that the current solution or process is too slow, cumbersome, and ultimately too expensive. Every month that the old procedure is used costs the company money unnecessarily.

From a business point of view, it is imperative to implement the new and better solution as soon as possible in order to keep the losses caused by the old solution as low as possible.

It is astonishing, however, that this essential aspect is often ignored.

**An Example:**

The savings potential of the new solution is 500 TEU per month, but the purchasing process delays the implementation by one month due to costly negotiations for a price reduction of 25 TEU.

From a commercial point of view, this does not make sense.

The decision-making paralysis that prevails among individuals thus delays a decision and harms the company through lost value. Unfortunately, the personal sensitivities of those involved sometimes outweigh the business benefits. This is often due to the fact that top management is often inexperienced in the areas of IT and enterprise software.

It would therefore make more sense to realize the benefit, or at least an initial partial benefit, in the company as quickly as possible. This approach is diametrically opposed to the personal interests of those involved, as it places the benefit for the company above the perceived benefit of those involved.

In summary, it can be assumed that the reason for this approach lies in the decisive moment when the contract is signed. This is the turning point at which the vendor's sales representative has achieved his/her goal and hands over responsibility to the consultants and technicians. From now on, everyone is tied together and should make the rollout a success at all costs.

It is not uncommon for the preparation phase, i.e., the period until the contract is signed, to be much longer than the actual rollout, i.e., the period until the first users start generating "value" for the company.

An alternative approach would have to take these two essential human-motivated points much more into account. This alternative should enable the earliest possible generation of benefits in the company through the new solution and significantly reduce the fear of responsibility of the people in charge in the company.

The "ramp-up" rollout method represents an alternative here and is now described on the following pages.

### 3. Introduction of enterprise software according to the agile ramp-up process

The **ramp-up process** aims to ensure that users of an enterprise software take advantage of an optimized process built into its software by the product vendor, thereby getting them up to date on the latest technology and know-how in the enterprise organization.

The thinking is that the **company has a problem** and the **vendor the solution**. The solution is expressed in terms of the **software** itself and the methods and processes implemented in it. Furthermore, the software should be used in the real world as quickly as possible. Only then can it unfold the hoped-for benefits at a very early stage.

Of course, the selection process of the provider cannot be omitted. However, a solution from the point of view of the company is not imposed on the supplier (in the form of a tender or a specification sheet), but the people who are ultimately to work with the solution describe their problems, wishes and tasks to the supplier.

The supplier must now **show in reality** how he would solve the problem with his product.

The approach of this ramp-up method is to reverse the waterfall process described at the beginning. The benefit arises in the waterfall process at the end, namely when the users use the software. This is the actual **proof of concept**, the point of truth. Does the solution meet expectations, i.e. does it deliver the benefits you want?

In the ramp-up process, this section is placed at the beginning.

What happens now with the other participants, such as lawyers, buyers or consultants? These are not important for the generation of the benefit, they can be left out at the beginning. First, a professional evaluation from the user's point of view is carried out, then the further necessary steps are taken.



Another aspect is the predictability of a functioning (in terms of business benefit) of a complex business solution.

In the classic **waterfall model**, a sample solution is theoretically designed as a presentation - usually by a group of people who are not necessarily the end users.

This is like betting on whether everything will then work out in practice and whether there will be a benefit for the company. In a complex networked world whose technologies are being developed at breakneck speed, it is important to be fast.

## **4. Procedure of the Ramp-Up Method using the example of the project and resource management solution Can Do**

The actual procedure is very simple and quickly explained - and perhaps that makes it so **attractive**.

A first department or division in a company has clearly recognized that in a topic - for example, project and resource management - there are considerable weaknesses in the approach and the software used, often Excel®, is not target-oriented.

Various providers are confronted with the problem via video conferences, each of which shows a concrete proposal with their own solution. No sales presentations, no slides, just the product itself.

The contact persons at the vendors are therefore the consultants (who would carry out the rollout according to the classic procedure) and not the sales staff. The contact persons at the customer are the **users** (in our example project managers) who are to work with the product, not people from outside the field such as purchasers, middle management or consultants.

The next step is to start on a small basis, minimally invasive, so to speak. The provider makes the software available to the customer immediately. Ideally, this is done in the cloud to keep the test at a low threshold and avoid costly discussions with the internal IT.

In return, he receives a rent (**Software as a Service**). Furthermore, the provider's consultants are paid for their support.

Although no large amounts are involved here, it is nevertheless important and this underlines the fact that there is a certain seriousness in the approach. It must be clear to the end customer that every month and every hour of conversation costs money.

**Speed** and **efficiency** are therefore required.

An initial simple use case is formulated together that can be implemented quickly. In the case of Can Do, this means **recording** the **resources of one department** (with working time model, basic load and vacation entitlement) and the **projects** (with start, end and rough imprecise indication of who is working on them and how much) **in the system**.

This creates an overview of all ongoing projects (of the department), the workload and possible overload, etc. The employees could already report their progress or working time directly on the projects, and overloads are avoided. The company - a department - has an overview of the current projects, workloads, risks etc.:

There is an immediate **benefit!**

Now, the next step is to define another use case for implementation, for example, the simulation of new projects.

This agile **iterative approach** is **continued** repeatedly. At a later point in time, another department is added, in which it is recognized that the offered solution works for the first department.

Ultimately, the solution will prevail in practice within the company. This does not necessarily have to be the case: It is also possible that after a few weeks it is determined that the offer is not suitable after all. There are many reasons for this, but ultimately the ramp-up is terminated. No more rent is paid for the software, and no more consulting hours are incurred. A little money and a little time have been invested, but it has also been learned what cannot be implemented. With this expanded knowledge, the next solution approach is now being attempted.

## 5. Consideration of the Ramp-Up Method

In retrospective of the method (based on approximately 30 rollouts of the Can Do solution in this format over the past two years), the mundane reason for success is simply licensing the software as a "rental" for one department instead of a large investment for the entire company.

This part of the ramp-up process is the "game changer".

No investment decision is made, but the software is rented on a monthly basis and can also be cancelled on a monthly basis. The effects are dramatic. Because only a small amount needs to be spent by the customer at this stage of the project, the costly process of a large investment is not necessary.

Further, the solution is deployed in the cloud (albeit with a data processing mandate and appropriate security features of the solution). Thus, IT only needs to approve this "pilot" operation but does not need to contribute anything itself. Since the IT departments of the companies are chronically overloaded (because they often have no functioning resource management), this approach is usually received positively in IT accordingly.

The verification of the solution's suitability (i.e., the decisive benefit) is carried out immediately by the end users. As a result, many stakeholders are not initially involved in the first phase.

There is no need for an elaborate purchasing process, since only small monthly amounts are involved. Often, the head of the "pilot" department can decide on such a budget himself.

No other departments and their managers need to be involved, only one department is involved. External consultants are not necessary, and once the entire project takes place as an isolated pilot or test project, approvals regarding data protection or compliance can also be granted much more quickly.

Because the project involves only one department, only a few people are involved. This leads to faster and more agile implementation.

This approach also changes a lot for the vendor's representatives. First and foremost for the sales representatives who are used to selling the solution "once". In the ramp-up method, the solution must be sold anew every month. The customer company - the user - must be convinced each month that the solution will generate value. Otherwise, the system could be turned off by the customer company and there would be no further cost or obligation.

However, another problem arises for the vendor's sales personnel: the decision makers (economic buyers) are now the users, i.e., professionals. This group of people is immune to sales talks and is only interested in technically sound statements. These people don't want presentations, they want to see the solution live in action. This is a problem for some sales professionals, as they often do not have the product know-how at all.

At the beginning of these remarks, it was emphasized that the solution consists not only of the software itself, but also of the increase in the customer's know-how.

This knowledge and the resulting optimization of the procedure in the company arise from the step-by-step use of the software. Step by step use cases are solved by the standard, each step is checked for benefit and practicability. If a use case brings no benefit, it is simply not executed by the users and can be ignored.

The application is now placed wider and deeper in the company in two directions. Provided that there is always a benefit visible to the users. Other departments also realize the benefits of using the offered software. They are not forced to use the system as well, since there is no compulsion from "above". However, when they see the benefits, they may join the first department.

Furthermore, the system will be integrated deeper into the existing IT. Sooner or later, other systems will have to be connected. But not all theoretically conceivable products, but only integrations where the users really see an advantage.

## 6. Summarized comparison of the methods

"Waterfall Method" or "Ramp-Up"? Both methods have their justification. However, the trend is clearly in the direction of the ramp-up method, because its advantages outweigh the disadvantages.

Resistance to the ramp-up process often comes from stakeholders who feel ignored or restricted in their sphere of influence. These people can be involved as observers in parallel to the Ramp-Up.

The management of a company is well advised to strengthen and promote the competence and possibilities for this type of product selection at the departmental level. In this way, responsibility for the success of the company is assumed at the operational level, and acceptance of the solution is incomparably greater because it is proposed from within the company's own ranks. Finally, selection costs are also significantly reduced.

Perhaps not all challenges in a company are suitable to be met with this procedure. However, for every issue that requires new or additional software, the ramp-up method should be considered for finding a solution.

Sparked your interest?

Simply give it a try:

[BOOK A FREE CONSULTATION NOW](#)



**Thomas Schlereth**  
Managing Director

**Can Do GmbH**  
Implerstr. 26, 81371 Munich

Mobile +49 (0) 171 33 94 085  
E-Mail [thomas.schlereth@can-do.de](mailto:thomas.schlereth@can-do.de)  
LinkedIn [Thomas Schlereth](#)  
Web [can-do.de](http://can-do.de)

→ [Schedule an appointment with me now](#)

cando

With the help of artificial intelligence, we make sure that planning mistakes, incalculable risks and projects that run out of time are a thing of the past

Made in Germany



DSGVO



REVISIONSSICHER  
nach HGB §§239,257