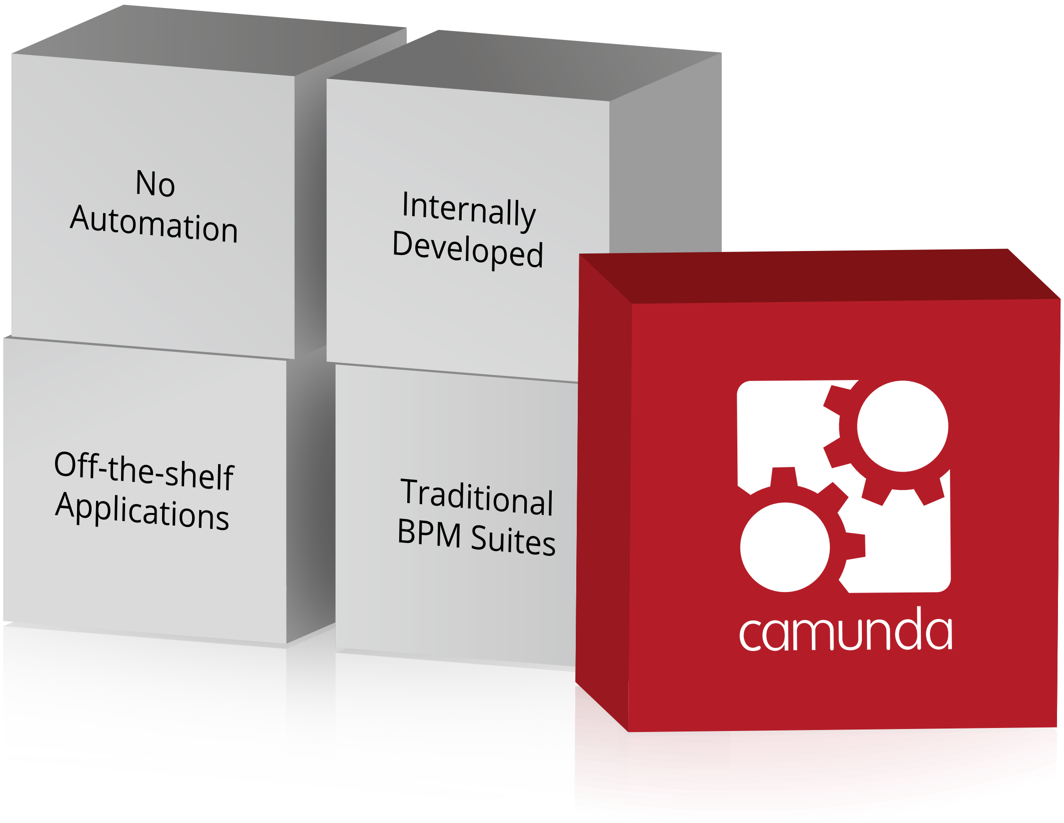
camunda
the business process company

Camunda BPM
Compared to Alternatives



May 2015 – camunda.com

Contents

Alternatives Compared

	No Automation	Internally Developed	Off-the-shelf Applications	Traditional BPM Suites	Camunda BPM
lower process costs		✓	✓	✓	✓
faster process runs		✓	✓	✓	✓
fewer process errors		✓	✓	✓	✓
real process transparency			possibly	possibly	✓
increased process agility				possibly	✓
less programming effort	✓		✓		✓
tailored processes	✓	✓			✓
integrated into existing IT		✓			✓
no vendor-specific developers needed		✓	✓		✓
no vendor lock-in	✓	✓			✓
end-to-end processes		possibly		✓	✓

Introduction: Advantages of Automation

Process automation is an intelligent way to organize business tasks throughout a process. There are obvious reasons for businesses to adopt this approach:

- Lower Costs – with automation fewer tasks need to be handled manually.
- Faster Processes – by eliminating the idle time which inevitably arises from manual processing.
- Fewer Process Errors – another advantage of reducing manual tasks in a process is that you'll experience fewer errors.

For an expanding business, having unorganized or undocumented business processes can often be a barrier to scalability. As a result a company's ability to grow will be adversely affected.

Using our first-hand experience with customer projects as well as working with issues raised by the customers themselves, we've compiled a detailed critique of the available options when it comes to process automation.

To give the document some context, it’s good to keep in mind that we’re making our evaluations based on companies that perform software development and that have appropriate developers available.

Internally Developed Process Automation

Example

Often a business will recognize that some of its manual processes would be better automated. Therefore, someone in the business may decide to spend some time writing a simple program which automates a common process. The benefits of automating manual processes are likely to become apparent quite quickly. The simple program may be expanded to include more processing. The logical conclusion of this is a program that operates like a black box, in which it starts then something happens and at some point the process either completes or fails.

There are unfortunately some big disadvantages with this approach.

Disadvantages

No Process Transparency

The process logic is expressed directly in source code which is only comprehensible to, or known by, the respective programmer. It’s also not at all unusual for new code to be added without any update to documentation. Processes are often modeled in a requirements document or written down at a later stage. However, you can never know for sure whether these manually created representations actually correspond to reality.

This applies to both the fundamental question of how the processes are executed, as well as to the question of how a particular process (process instance) was actually executed in individual cases. This can be a big problem for industries that are required to adhere to strict regulation, compliance with these regulations cannot often be guaranteed.

No Process Agility

Both the initial technical implementation of the processes as well as the adaptation of the already implemented processes are communication-intensive. Any technical implementation requires a complex coordination between subject matter experts as well as IT professionals, which is error-prone and time-consuming. Often, the desired processes are described by the business side in a relatively non-binding notation which the IT side will have to first understand before being able to produce any meaningful results. These results are very often hampered by delays and errors at various stages due to this inaccurate communication process.

High Programming Effort

The challenge of automating business processes is often underestimated. At first glance it appears to be technically simple to program a sequence of activities whilst also taking rudimentary control structures (if / then, etc.) into account. However, challenges arise frequently if wait-states are reached during execution.

For example: We receive an order that must be examined by a clerk. After examination, the item gets shipped. If the examination exceeds a certain time limit, a reminder must be initiated, and if this is unsuccessful, a reassignment to another clerk needs to take place. The wait-state described in this example (the process waits for the completion of the task), in combination with the tight timeframe required for a two-tiered escalation is relatively challenging to program. The effort increases exponentially with the complexity of the process: in real end-to-end processes there are often more than 50 wait-states and as many temporally dependent activities.

The above example is merely the tip of the iceberg. There are numerous other technical challenges in process automation (e.g. in the asynchronous processing of transactions, in complex service orchestrations, in high volume processes with respective scaling etc.). They are often not known at the start of a selective process automation and will only be understood as such when it is already too late.

In summary, it's quite clear that the internal development of a process engine makes as little sense as the internal development of a database or an operating system.

Conclusion

Generally speaking, process automation with internal software developers may appear like a logical approach at first. Especially where aspects of "tailoring" and "vendor independence" are a concern, and certainly when automating core business processes. Camunda manages to embrace the advantages of internal development without putting developers in a position where they're reinventing the wheel.

Implemented Within an Off-the-shelf Application

Example

There are lots of processes that are common to many businesses and there are off-the-shelf applications that takes advantage of that fact.(e.g. CRM, ERP etc.). This software tries to make it broadly possible to manage a business's process without the need for developers. This approach of course comes with the advantage of not requiring any programming to work but there are also serious disadvantages to consider.

Disadvantages

No Tailored Processes

As this kind of off-the-shelf application aims to be an "off-the-shelf" product, its main goal is to be of broad appeal. These processes often do not fit the individual reality of a company. While automating common support processes like holiday requests or invoicing can often fit in quite well, the situation isn't at all the same when a business' core processes are concerned. More individualized processes like insurance claims or telecommunication ordering processes have components unique to an individual business. In fact, the core of every business is by design unique in order to set itself apart from the competition. It is here where a broad business processing tool loses all advantage.

Some off-the-shelf application vendors promise to solve this problem by "customizing", which is offering client-specific adjustment of processes. This raises the question of whether this can be sufficient to reflect the degree of individuality as shown in the business model. Our experience has shown that this is often not the case. But even if the software vendor is able to achieve this customization there are still two big disadvantages for the end user:

No Process Agility

In very rare cases, process adjustments can be made without the involvement of the vendor. But for the vast majority of cases, customization is a service chargeable by the vendor. Apart from the costs associated with the adjustments, the time frame allocated to implement a change is dictated by the vendor themselves. Any time constraints the business might have would not be taken into account.

Vendor Lock-in

Using off-the-shelf applications which rely on the vendor for customization always results in a vendor lock-in with that company. Should the supplier relationship turn sour, the vendor be acquired or becomes insolvent, this would result in serious risks for the company. This fact becomes more critical the deeper the automation processes are involved in the core business.

Not integrated with existing IT

Another very rare case is that the procured off-the-shelf application is the only element in the existing IT landscape. But in most cases the situation requires adding the software to the existing infrastructure. The problem occurs that off-the-shelf applications cannot easily be embedded into existing technical structures and therefore can only be operated with additional work.

No End-to-End Processes

From a business perspective there is a more critical problem that arises from the use of off-the-shelf applications. That is that a off-the-shelf application usually only represents a fraction of the process landscape (e.g., ERP, CRM, etc.). A true end-to-end process that naturally reaches across multiple application systems can never be fully implemented in the off-the-shelf applications and this can only be holistically managed, measured or improved.

This results in many subsequent problems. For example the employee concerned must often work with multiple task lists as every off-the-shelf application comes with its own task list that is – of course – product-specific. Each task list would operate differently and vary in the functionality that it provides.

Conclusion

The argument for delivery “out-of-the-box” initially appears strong. However, it should be kept in mind that the benefits are predominantly short-term, whilst the drawbacks have a long-term impact. A company whose business model is implemented in IT should be aware that IT is the heart of the company and a critical success factor how it differentiates itself from the competition. This realization subsequently demonstrates that such a key component cannot be bought hastily but must be developed on an individual basis.

This does not mean that you have to reinvent the wheel. It means assembling one’s own individual process automation from existing components. With those components you have significantly more control than you would with off-the-shelf applications. Camunda BPM gives you those components which in turn gives you complete control over how your processes are designed and maintained.

Traditional BPM Suites

Example

Traditional BPM suites are quite similar to off-the-shelf applications in their attempt to appeal to a wide audience – their goal is to try and add value by introducing an element of programmability to the software. They allow individual modeling and technical design of automatic and semi-automatic processes, and thus can offer measurement capabilities and systematic improvement.

However, to a very large extent they miss out on all the benefits associated with model-driven design. By being forced to model aspects of the process application like masks, interfaces and the like, the ease of mapping a process without needing to know about the infrastructure is lost.

As a consequence of wanting to provide programmability, very proprietary software development is of course required. This in turn pushes the responsibility of learning how to develop the vendor's specific software implementation on the business.

In our vast BPM project experience, we have found that it is in fact this perceived "benefit" that causes the most issues with transitional BPM suites. The problem is especially pronounced when in-house software development is already taking place in a structured way, for example in Java.

Disadvantages

High programming effort

As software development is vendor specific, the in-house developers need to learn and practice the vendor's specific platform. The related expense is not one-off but continuous, retraining is required to ensure that the knowledge is maintained. Any existing knowledge of software development in Java (for example) cannot be applied. In addition, existing tools, techniques, and best practices of software development (e.g., unit testing) cannot or can only be partially applied. This severely limits the developers' productivity. As a result, the technical implementation is much more complex than it appears at first.

Inability to model distinctive parts of a process

Due to the predominantly model-driven development approach, the possibilities of technical implementation are limited. The following comparison illustrates this problem: On a blank canvas, an artist can paint a picture in exactly the way he imagines. Alternatively, there is the principle of "painting by numbers", where even the artistic layman can create stunning images by coloring in predetermined areas. However, they can only create what was already pre-designed.

Similar to off-the-shelf applications, the principle of "painting by numbers" in BPM suites is often sufficiently flexible for standard support processes (e.g., holiday requests, invoicing). The limited possibilities of technical implementation are, however, insufficient to capture and implement the core business processes.

Not integrated into existing IT

On an operational level, the drawbacks of off-the-shelf applications also apply to traditional BPM suites as well: they cannot easily be embedded into existing IT structures.

Specialized Developers Needed

As mentioned already, a model-driven development approach is inevitably vendor specific. So there's no getting away from the fact that you will need developers that are specifically trained to use a specific BPM suite. If they are not available, they are much more difficult to find than developers for popular programming languages such as Java.

Vendor Lock-in

Consequently, there is a strong vendor dependency as they and their partners are usually the only ones who have developers with the required level of expertise. This is acceptable in the context of support processes (e.g., holiday requests, invoicing), however, it provides an unacceptable risk when capturing and implementing core business processes.

Conclusion

Traditional BPM suites suffer from a "stuck in the middle" problem, usability is stretched trying to accommodate features from two different approaches.

They are as unsuitable as off-the-shelf application products to the already existing software development in any company, while not even being able to offer an out-of-the-box solution for process automation. This dilemma is a result of an unsuccessful search for a compromise between the two extremes, and to a large extent, the result of a more academic flow of the last decade (model-driven software development). The modest growth rates over the last decade for these products seem to confirm this assumption.

Camunda BPM

Brief example of stack

Camunda doesn't believe in the abilities of out-of-the-box solutions when it comes to process automation and it's not attempting to lock businesses in by insisting on anything proprietary in nature. Those facts mean that all a business needs is the knowledge, skills and tools of their existing developers in order to design and execute full end to end process management. The modeling standards and Java integration that Camunda employs are entirely open and widely accepted. This gives us a host of advantages against alternative BPM options, which can all be backed up by real world examples.

Advantages

Process transparency and agility

Camunda very strictly adheres to the ISO-standard for process modeling, BPMN 2.0. The processes that need to be automated can be documented graphically by the respective departments and directly executed in Camunda without further transformation. Operationally, the current processes can be viewed directly, the "source code" of the process can also be viewed and easily understood by the business departments.

This fact, for example, has led to increased agility of processes within Zalando AG, as confirmed by Marko Lehn (Team Lead Software Engineering):

zalando

Zalando AG

Marko Lehn
Team Lead Software Engineering

»Our BPMN 2.0 process models are executed directly, which improved communication between business departments and development and shortens development cycles.«

Less programming effort

The technical challenges experienced during process automation of an internally developed solution are solved in the development of Camunda BPM. The process engine at the core of the platform is a mature, stable component with powerful functions, which can be used for projects immediately.

This has already been proven in a challenging context at Lufthansa Technik, as Tobias Mohr (Team Lead IT Projects and Systems) explains:

Lufthansa Technik

Tobias Mohr
Team Lead IT Projects and Systems

»In Camunda BPM we have found a lean and stable platform for our agile BPM / SOA projects in a complex process environment. The performance and reliability of Camunda BPM was demonstrated in various projects and confirmed our decision for the product.«

Bespoke embedded applications

Unlike traditional BPM suites, Camunda is an open framework that can be seamlessly embedded into the existing technical environment. It allows for the use of the entire Java ecosystem for the development of process applications, and makes no restrictions on the use of other components and frameworks (e.g., Spring, Java EE, etc.).

The existing integration with Java EE was also a central argument for the use of Camunda BPM at Freenet AG:

Freenet AG

»Two things are important to us for the automation of our core processes: high availability in a high-load scenario and the integration into our existing Java EE6 programming model. Both are given with Camunda BPM.«

No specialist developers needed or Vendor lock-in

Camunda BPM is open source and therefore freely available. If required, Camunda, as a vendor, also offers an enterprise edition with SLA-based support. This combination means that any Java developer can work productively in a short time with Camunda BPM and the dependency on Camunda as a vendor is minimal.

This fact was decisive for the use of Camunda BPM at 1&1 Internet AG:

1&1 Internet AG

»We prefer open-source solutions which give us full control over the technology and allow for the development of specific adaptations if needed. Camunda BPM turned out to be an ideal solution for us. We also see an excellent opportunity to share our process knowledge with others and to benefit from an active community behind Camunda BPM.«

Conclusion

- Camunda vs. Internal Development: When it comes to internally developed process automation systems, Camunda BPM has all the advantages of that approach combined with a better process transparency and scalability.
- Camunda vs. Off-the-shelf Applications: Camunda BPM is vastly superior to process automation that comes embedded within off-the-shelf applications. It’s agile, can be completely tailored to the needs of the business and even be entirely integrated into the existing IT infrastructure.
- Camunda vs. Traditional BPM-Suites: Camunda BPM can also match any apparent benefit of a traditional BPM suite without having to make the developer suffer with proprietary development practices.
- Finally, if you are looking to help a business maintain cohesion and scalability while it grows, implementing Business Process Management is essential. If you want to enjoy every available benefit from that approach you’re going to need Camunda BPM.

Imprint

Europe / Asia

Camunda Services GmbH
Zossener Str. 55
10961 Berlin
Germany

Phone: +49 (0) 30 664 04 09 - 00
E-Mail: info@camunda.com
www.camunda.de

America

Camunda Inc.
44 Montgomery St, Suite 400
San Francisco, CA 94104
USA

Phone: +1.415.548.0166
E-Mail: info@camunda.com
www.camunda.com