Camunda Best Practices

Doing a Proper Proof Of Concept

When evaluating your BPM approach, a "Proof of Concept" (PoC) is often a good step to check if the BPM methodology, the standards BPMN, CMMN and DMN as well as the Camunda platform suits your needs. It is vital for a proof of concept to make up your mind about your goals, to select a suitable process and to prepare it and carry it out properly.

Doing a Proper Proof Of Concept

Feedback ? best-practices@camunda.com

Understanding **Proof of Concepts**

With a **Proof of Concept** (**PoC**) you create a prototype application within no more than **three to five days**. The result of a PoC is intended to be thrown away after having served its purpose: to try and show that your project will "fly" - including all aspects relevant for your specific situation. Such aspects might be:

- Is it possible to **embed** the process engine into your own architecture?
- Does the **development approach** fit into your own organization's approaches?
- How can I **model** a specific business domain problem?
- Which kind of **know how** is needed for the business and development teams?
- Which **effort** will typically be needed for these kind of projects?
- What are the impacts of process applications for **operations**?

Often it does make sense to implement such a PoC together with the software vendor and/or specialized consultants, in order to get quick results and focused feedback with respect to your specific challenges. However, you should always at least **co-develop** the proof of concept, in order to really understand what is going on. A team size of two to four people has proven to be quite optimal.

Defining and Focusing on Specific **Goals**

Before planning and carrying out a proof of concept, you should consciously clarify the specific goals you want the PoC to achieve. Typical goals might be:

- to **verify** that the approach or the tool works under specific circumstances
- to **show** a case that **convinces** internal stakeholders that the approach makes sense

- to work through a complete **example** and get specific **questions** sorted out
- to **learn** about Camunda and **understand** how it works

When selecting your goal keep in mind the needs of all relevant stakeholders.

> ⚠ Do not just "collect" goals here, but try to make up your mind as to what really matters. Often it is better to make a clear choice, whether for example to show off a nice user interface at the end of the week or to have time to clarify all questions and to understand Camunda in depth, maybe even only using unit tests.

Defining a **Scope** Relevant to Your Business

Select a **useful** and **suitable** process, case or decision in the light of your goals.

Typically, it should …

1. be **relevant** to your **core business** stakeholders
2. make your organization’s **return on investment** on BPM more transparent
3. be **feasible** within the proof of concepts time box

> ⚠ Avoid "political" mine fields when selecting the process for your proof of concept.

Planning the Proof of Concept

Involving the Right **People**

It does make sense to implement a PoC together with the software vendor and/or specialized consultants, in order to get **quick results** and **focused feedback** with respect to your specific challenges. However, you should always at least **co-develop** the proof of concept, in order to really understand what is going on.

When planning for your team, consider that successful process modeling requires not just knowledge about the business and the targeted technical solution, but experience with BPMN modeling and methodology as well as analytical and moderation skills. We therefore typically bring together **business people** with **IT staff** and internal **business analysts**, **train them properly** and let them continue to **learn on the job** by carrying out the proof of concept together with an **experienced consultant**. A team size of up to a maximum of **four people** has proven to be quite optimal.

In case you want to access **systems interfaces** during your PoC, also make up your mind who will be a technically knowledgeable and available **contact person** for that system. In order to integrate into existing **user interfaces** you might need help from colleagues within your organization.

Define a **moderator** to avoid too many detours and keep your PoC on track.

Planning the Technical **Environment**

Make the necessary technological choices by Deciding About Your Stack and Deciding About Your Tasklist.

> ℹ Both documents clearly mark our subjective "default" choices by means of the GREENFIELD badge. Use those choices when having to select something for a "greenfield project" without facing contradictory requirements. Consult our Getting Started Guide to setup the greenfield platform components.

Perform a PoC on a Camunda **Enterprise Edition** to avoid loosing time on already patched issues and to have all monitoring features available, which is especially important for PoC phases. Request a license key with your Account Manager or online via camunda.com/bpm/enterprise/trial/[1].

Consider a simple **local developer's environment** aside the test/integration environment which runs the production configuration. So you could e.g. develop using WildFly as container and H2 as database to allow for quick development but deploy the result on on a WebSphere and DB2 to validate everything.

In case you want to access **systems interfaces** during your PoC, set up a **test** system and verify that it is usable.

Prepare a location in a **version control system** where you can develop your PoC. Having a shared repository with history does make sense also (or especially) in a two days PoC! Collaboration is simplified very much if the Camunda consultant can also access the repository. it may worth just creating a repository with weaker access limitations for the PoC.

> ℹ If your organization cannot easily setup a repository for the PoC, or access for the consultant is impossible, you can easily create a cloud repository yourself. We typically recommend Bitbucket [2], a free account is sufficient. It gives you a Git repository and you can invite all necessary people for the PoC. Afterwards you can simply delete that repository.

🕘 Selecting the **Time Frame**

As already mentioned above, we typically plan no more than **a focused week** for the PoC workshop itself. Sometimes it also works well to split up the PoC into two weeks of 2-3 days each, which allows everybody to reflect on the PoC over the weekend.

> ℹ To our mind you might have chosen the wrong tool if a PoC is not doable within a week! ☺

- Plan **1-3 days** for **modeling** the process with Camunda Modeler.
- Plan **2-3 days** for **implementing** the process with Camunda.

When selecting the exact time frame, consider all the people involved as well as any technical preparation you need to do up front. You also might want to plan for further steps, like a few more things you implement yourself internally in a second follow up week.

Presenting the Results

Before presenting the results of your proof of concept to a wider audience of stakeholders, select a **speaker** who is comfortable with presenting, prepare a set of focused **slides** illustrating your progress and the lessons learned, and **test** your solution and presentation at least once up front.

> ℹ The speaker might also be your Camunda Consultant - s/he is pretty used to present BPM to a wide audience! ☺

Bonus 🎁 Preparing a PoC with your Camunda Consultant

Technical Checklist

- **Installations**: Make sure your **developer systems**, as well as any **target systems** for the PoC test and production you wish to use are set up. Consider our recommendations for the technical environment and in particular install:
  - Camunda **BPM** Enterprise Edition, e.g. Wildfly Distribution (docs.camunda.org/enterprise/download/[3])
  - Camunda **Modeler** (camunda.org/download/modeler/[4])
  - Java **IDE** and build tools you trust (e.g. **Eclipse** [5] for Java) and **Maven** [6]

    Make sure **Maven** runs and builds and is wired to access the public Camunda [7] and JBoss [8] repositories. Prepare our Get Started Tutorial [9] and make sure it builds and runs.

- **Developer Computers**: For maximum productivity, all participating developers should use the computer with which they work every day. Avoid using computers from a training room or shared laptops unless they allow a remote connection to the developer's personal computer. If the developer's computers are neither portable nor remotely accessible consider conducting the PoC in the regular office space of the developers. If your company network is restricting access to Maven and Git repositories on the Internet, consider using laptops that are not connected to the company network. Similarly, you should not force the external consultants to work on one of your computers. They will be twice as productive on their laptops and not lose time with software setup, configuration, and access restrictions. Obviously, you do not have to connect the consultant's laptop to your company network. Internet access and a shared code repository are enough to collaborate.
- **Files** or **Version Control System**: Make sure we can easily exchange files and code during the PoC, preferably via your own version control system (e.g. Git or SVN) or at least via shared folders, USB sticks or email attachments.

> If your organization cannot easily set up a repository for the PoC, or access for the consultant is impossible, you can easily create a cloud repository yourself. We typically recommend Bitbucket [2], a free account is sufficient. It gives you a Git repository and you can invite all necessary people for the PoC. Afterwards, you can simply delete that repository.

- **Interfaces**: Clarify which technical systems interfaces you want to access during your PoC, make any **documentation** for those available to the whole PoC team and make sure there is a technically knowledgeable **contact person** for the interface available to the team during the PoC. Set up a **test system** and verify that it is usable. Verify with Camunda that everything is clear to the team, in particular from a technological perspective.

Organizational Briefing

Inform all PoC team members and other relevant stakeholders about

- specific **goals** and the selected **scope** for the PoC
- exact **location/address** at which the PoC is taking place as well as instructions about how to find together when arriving
- **start** and **end times**, as well as any additional preparation/meet-up times
- **names and roles** of all involved **people**
- **beamer/projector**, white-board and flip-chart availability
- **Internet** availability for team members and external consultants

Ideally, prepare a few **organizational** and/or **project** info slides to get everybody up to speed on day 1 of the workshop.

Links

[1] http://camunda.com/bpm/enterprise/trial/
[2] https://bitbucket.org/
[3] https://docs.camunda.org/enterprise/download/
[4] https://camunda.org/download/modeler/
[5] https://eclipse.org/downloads/
[6] https://maven.apache.org/download.cgi
[7] https://app.camunda.com/nexus/content/groups/public
[8] http://repository.jboss.org/nexus/content/groups/public
[9] http://camunda.org/get-started/developing-process-applications.html

Disclaimer & Copyright

No guarantee - The statements made in this publication are recommendations based on the practical experience of the authors. They are not part of Camunda's official product documentation. Camunda cannot accept any responsibility for the accuracy or timeliness of the statements made. If examples of source code are shown, a total absence of errors in the provided source code cannot be guaranteed. Liability for any damage resulting from the application of the recommendations presented here, is excluded.

Printed November 1, 2018. Applies to Camunda **7.9**. Any feedback? **best-practices@camunda.com**!