

# AGILE RFP FOR SOFTWARE PROJECTS

Building value and trust since day 1

JAROSLAV PROCHÁZKA

<b>TRADITIONAL RFP</b>	<b>3</b>
<b>WHAT IS AN AGILE RFP?</b>	<b>4</b>
<b>BUILT-IN BENEFITS OF THIS APPROACH</b>	<b>5</b>
<b>PROJECT VELOCITY AS THE FAIR-ASSURANCE</b>	<b>7</b>
<b>SIMPLE CHECKLIST FOR AGILE RFP</b>	<b>8</b>
<b>BEFORE YOU EVEN START YOUR AGILE RFP THOUGHTS</b>	<b>8</b>
<b>OK, LET'S GO FOR IT!</b>	<b>10</b>
<b>3, 2, 1 ... RFP KICKOFF</b>	<b>12</b>
<b>THE RFP SPRINTS</b>	<b>13</b>
<b>THERE IS ALWAYS ONE WINNER: YOU!</b>	<b>14</b>
<b>ABOUT AUTHOR</b>	<b>16</b>

## TRADITIONAL RFP

Traditional RFP is sort of legal preparation, unintentional **declaration of mistrust** and delivers very **low business value** regarding the project itself.

The “valuable part” of the project only starts after contract is signed and the scope is fixed.

In traditional RFP, you do not have many ways how to practically verify provided information, apart from references.

Traditional procurement procedures including steps like RFI (Request for Information) and RFP (Request for Proposal) steps are based on extensive paper work consuming a lot of time and resources on customer and vendor side. Such a procedure is sort of legal preparation, unintentional declaration of mistrust and delivers very low business value regarding the project itself. We “only” select the vendor but there is almost nothing to be used on business side when this procedure is done. The “valuable part” of the project only starts after contract is signed and the scope is fixed.

**Such procedure and behavior is quite understandable from the customer point of view.** Customer would like to select the best vendor according to proposed price, promised quality, declared business knowledge, experience and speed capability. But the point is the word “proposal”, “promised” or “declared”.

In traditional RFP, you do not have many ways how to practically verify this information, apart from references. Is the proposed triplet price-quality-speed real? Is it even physically feasible? How can I – the customers that is not an IT expert – even assess it? The experience says the reality quite often varies from the proposal. And one of the reasons is the procedure itself. Big pressure on price, fixed scope with very low flexibility for change, late delivery, low value, lack of trust... Sounds familiar to you?

Such a distorted conditions are even more affected by so-called fix-price-contracts that are in reality fixing the scope, budget as well as delivery date. Still I don't get the point why we call it fix price, if we fix all project aspects, not just the price. IT vendors are forced to lie in their proposals under such conditions. Fortunately, there are also other options. One of them is an Agile RFP.

## What is an Agile RFP?

Agile RFP is a practical way of value-based competition between invited or pre-selected software vendors. Rather than asking and replying to questions and then determining which answer and proposal is better, you really try it, experience the vendors and see the **real** results.

Agile RFP is a **practical way of value-based competition between** invited or pre-selected **software vendors**.

Rather than asking and replying to questions and then determining which answer and proposal is better, **you really try the project, experience the vendors and see the real results**.

What do you need to run such a RFP? You only need Agile experience and have a **relatively small part of the future project** that can be in the scope of RFP competition. It doesn't matter whether it is a new service or a product to be built, integration project or implementation of CRM or ERP. Always, there can be found a small horizontal cross-layer-step that can be delivered to see (and immediately consume) the value of the project.

Examples can be following:

- **For the new project:** it can be one core user story like the calculation of salary, first version of management dashboard with just a few measures or the invoice issuing, not the register to insert or edit the data although vendors can tell you they need to start with this. But what is the value of simple register without any extra function on top of it?
- **For integration:** it can be just one typical scenario integrating subpart of planned systems but delivering new functionality. Typically the scenario used daily (not only sometimes) or processing 30% of all requests or low frequent one but with highest value.
- **CRM implementation:** it can be simple sales pipeline feature or new valuable report based on existing company data.

The goal is to find few simple scenarios delivering something new that can be ideally used in real life customer environment after the RFP is finished. Of course, not always need to be released in the live environment, but it is and should be shippable. It is not

buggy-spaghetti-code prototype. This is the point. And by such mini-project you verify the promises of the vendors. You experience their way of working, communication style, the team, flexibility, skills and domain experience. At the same time, it serves also your learning. By the regular demonstrations of the initial solution, you clarify your needs and expectations. Based on this, you can still adjust the scope of the whole project.

---

*In other words, there is no better way to clarify your needs and verify vendors' promised ability of frequent releases, high quality code, velocity, ability to work with open and changing scope, and cooperation style, than to experience it in real life under future planned conditions.*

---

Already after the first sprint demonstration, you'll see the profile, abilities, communication style and attitude of the vendor. You also see the week-by-week progress. Some vendor will set the speed and run like hell all the time. Others will significantly improve as they learn. One can be really business-oriented or proactive. Other can be more technical or passive. All this will help you to clarify your needs.

### **Built-in benefits of this approach**

Why is it good to conduct such a competitive selection mini-project rather than a traditional RFP procedure?

- You build trust and cooperation routines since day 1.
- All competitors deliver real and shippable value.
- Much shorter time is spent on paper work, people work on real value product, not on paper piles used later on for decision.
- Practical verification of vendors' skills, experience, working routines and approach in the future project context.
- No (or at least less) room for politics. There is usually collective agreement on the winner. The skills and results are clearly visible to all people participating the demonstrations. It is based on facts, not on theoretical answers and personal/political preferences.

What people say after the Agile RFP? I will mention only one statement:

*“Now we have 3 usable MVPs and clarified project roadmap just after 1 month of work. In the traditional RFP, we would be still in the phase of internal discussions about what we really need with just part of initial requirement description.”*

Such Agile RFP is not so easy to run. There are many small aspects that can go wrong and a lot need to be prepared upfront (but still less than for traditional RFP). Therefore there is **one key precondition** before starting Agile RFP. **You should be practically experienced with agile way of working, not just from software development or IT operations teams, but also from business teams.**

If you answer yes to few of the following questions, you should be ready to try Agile RFP:

- Have you already tried to run project/product cross-functional business teams sitting and working together instead of “teams” in functional silos?
- Does your team have a clear prioritized backlog of tasks/projects (not just all is priority 1)?
- Do you limit Work-In-Progress (at least on project level)?
- Do you show and release outcomes of your work frequently?
- Do you ask your real users for the feedback on your work?
- Do you run regular retrospectives to improve?
- Was some of the mentioned done in the context of your business units, HR, sales, marketing or even in the management team?

If you don't understand these questions or you haven't heard yet the terms like retrospective, WIP or prioritized backlog, than I recommend to try it internally first. You can also [read more about the required Agile experience and phases of Agile evolution](#) in organization as I see it from the Agile adoption cases.

Your experience is important, because you need to prepare the business side of the project for RFP (scope, stories, benchmark estimates). You need to provide experienced Product Manager (team) that will take care of the competing vendors, consult priorities, and discuss acceptance criteria and all the proposed changes. You are also supposed to conduct regular demo sessions with vendors and all this should not be new for you.

## Project velocity as the fair-assurance

You can wonder how to “compare” the vendors’ capability, velocity, value delivered and **how to ensure the sustainability in the subsequent project**. Vendors can run exhausting sprints to deliver the highest value and story points to win the RFP or they can use “hidden” internal people. But this is not much sustainable. This is quite crucial and can be done in several ways. We had following thoughts:

- Have the same/similar environment (internal or external)
- Have the same (claimed) team size and wanted to know the changes for the sprints during the sprint planning
- We have prepared benchmark stories (with expert estimate from our side) mandatory to all vendors
- Velocity achieved with claimed team in RFP was expected as the initial one in the subsequent project.

Although we trust our pre-selected vendors, we wanted to have these checks included in the RFP procedure. Such **commitment to proven velocity removes all tempting techniques** and forces vendors to be sustainable in the speed and be open to us. It worked perfectly in our case.

And to really gain the value out of the RFP, **you should own the delivered code at the end**. Therefore prepare the license and IPRs agreement and pay all vendors for their effort if they pass agreed RFP’s Definition of Done (e.g.: conducted all demo sessions, showing the sprint plans, having code in GIT for demo, ...).

Remember, the goal of such RFP is not to act from the position of power. The goal is neither to catch vendors on something they don’t know. The goal is to build trust, uncover risks early enough, see how they progress and solve these situations, learn together and set the basis for future Agile project you are hiring for.

## SIMPLE CHECKLIST FOR AGILE RFP

In the previous chapter, I introduced the nature of Agile RFP, its benefits, important preconditions and the differences between Agile and traditional one. In this chapter, I would like to share some points to focus on before, during and after the Agile RFP.

### Before you even start your Agile RFP thoughts

As I already emphasized in the previous part, the most important precondition for smooth and valuable Agile RFP is **your own Agile experience**. The rest is more about accurate preparation, but your own Agile experience cannot be substituted by it. It has to be witnessed. Why your own experience from Agile projects is so important? Your PM and business people need to know the way planning, demo, scope changes or acceptance is done to run Agile RFP successfully. Although Agile mentor can help you to prepare it, she will not inject your business people the experience.

So, before you even start, think about following:

- Have you **run your own IT and business Agile projects** (at least several releases, in several domains/units, not just IT) to become familiar with Agile way of working and to change the mindset of business people involved in the future RFP?

*If not, first try it for several months at minimum to gain the knowledge and experience. There will be too many new things in Agile RFP, therefore no time to learn the new way of working.*

- **Involve also procurement and PMO people** in early discussions, give them some intro training and invite them to existing planning, demo, and retrospectives of existing Agile projects to see how it works.

*This is quite important to absorb the mindset and routines.*

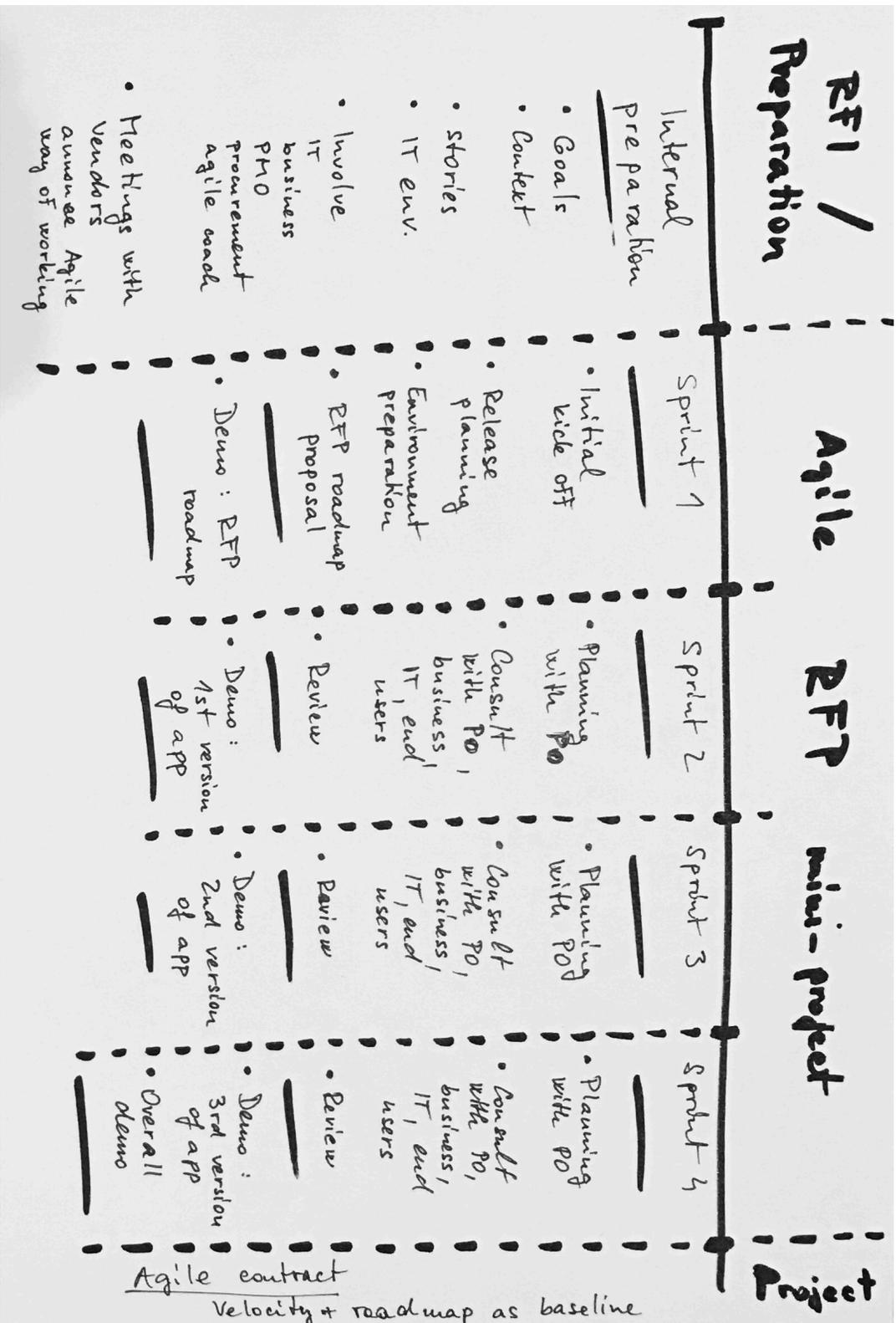


Figure 1: Map of Agile RFP

## Ok, let's go for it!

Preparation itself (several weeks) can be also run in several one-week iterations and should cover following activities:

- **Nominate Product Manager/Product team**, who will be accessible for the vendors during the project (someone on business side from existing Agile project being purchased). It can be also small group of, e.g. 3 people to cover all business aspects, knowledge and needs, but also for the case of substitutability.

*We had just one Product Manager and it was quite exhausting for him, since he was still involved in the business part of the project being contracted for IT implementation. But he was very capable and efficient and he managed it quite well.*

- **Prepare the high-level business scope of the RFP** (goal, context, selected business area and first few stories to be implemented, those can be more detailed).
- **Prepare benchmark stories/use cases** with business and IT people (description, acceptance criteria, estimates) that will be mandatory to implement and can serve as a basis for the “vendors’ comparison”.

*Although the velocity among the teams is not comparable, we've chosen to set story etalons to compare somehow the speed as well as thinking of vendors. We described several basic and integration stories, set their effort estimation (with IT guys), set them as mandatory and kept the rest of the scope open for modifications to see vendors' approach, communication style, proposal ideas and business thinking. It was risky, but one of the best decisions at the end!*

- **Set the size of allowed vendors' development team** (keep it small, something like 3 to 5 people including analysts and testers) to have comparable results and deliverables.

*Ask for openness if vendors use other people to help, e.g. UX people, other analysts, architect helping with design.*

- **Communicate clearly the way RFP will be handled** to all vendors invited. Show them the model, calendar of planned sessions, explain them the conditions and Definition of Done.

*Not all may potentially accept such form of RFP, mostly if they are not experienced with Agile. By the way, you set the expectations and demonstrate one key Agile principle, that is open communication.*

- **Prepare IT environment** for all vendors invited to RFP (e.g. to verify the integration with selected internal system/interfaces).

*You can transfer this responsibility to vendors by proposing external IT environment, but you will not verify their ability to integrate with your systems if you plan to use your internal environment. If you plan to use some cloud platforms, use it from start. This setup depends a lot on the nature of the project and your intention with it.*

- **State technology preferences and limitations if any.** Many customers have their IT strategy, preferred architecture, communication systems (e.g. SOA or MQ), databases and people skills (which is important for daily support).

*If you request specific technologies, state it in RFP conditions. But also keep your RFP open for new possibilities and technologies that can improve your IT environment as well as teams productivity.*

- **Prepare the calendar of all RFP ceremonies:** kick-off workshop, separate planning sessions with vendors, consultancy with PO, IT, business users (and others if needed) and demonstration sessions.

*We discussed common or separate planning sessions also with vendors and decided to have separate session with each of them. The main reason was the voice of vendors. They wanted to use their domain expertise and knowledge during the planning sessions to propose feature updates, look and flow and we liked it.*

- **Prepare your Definition-of-Done (DoD)** for the RFP. Based on their achievement you will pay vendors for their effort (for source code ownership). Use standard DoD items like demo happened, code stored in and built from agreed repository, code reviewed by your IT people, etc.

*The amount of money you pay is up to you (remember, in traditional RFP you pay vendors nothing for their time spent on RFP answers), but it should be fair price for both sides. Depending on the length of the RFP project it can be 5.000-50.000 Euro per vendor.*

- **Prepare further material for vendors:** project details (context, needs, goals), epics and stories for the RFP sprints (2 to 4), templates for planning, assessment and demo.

*We had 3 one-week working sprints and initial release planning sprint to see their progress, ability to work with changes, and ability to understand and propose business value.*

- **Prepare RFP evaluation criteria and templates** to be used by your internal people during the sessions (mostly demo) to evaluate the vendors. This is usually based on your RFP selection criteria, therefore items like business knowledge, added value, usability, Agile skills demonstrated are used. This would be later on completed by proposed price for the future project development, delivered value in RFP mini-project and support conditions.

As you can see, there are many steps that need to be prepared. These steps consume mostly the time of business people. You may be confused now, where is the promised benefit of less time spent and more value delivered ;) Although mentioned activities can look time-consuming, it is and should be less time than would be spent on preparation of full detailed scope of traditional RFP.

Moreover, you get a working code for it, not just paper answers! And last, it is also preparation for the future project development continuing after best vendor is selected. Now you also see why there is no time to teach people Agile, they need to be already aware, skilled and self-sufficient.

### 3, 2, 1 ... RFP kickoff

Kick the RFP off by the initial workshop with all the vendors to ensure all hear the same project information at the same time. Explain the context and the goals of the business project that will be in the scope of this RFP. It is great, if business owner can present it herself. Of course, the details are provided by Product Manager or by other team members.

Explain and have discussion about the initial mandatory stories required in the RFP and ask vendors to prepare RFP roadmap (it was our sprint 0, so called release planning). We had four one-week-sprints. The first one was release planning and preparation and the 3 following sprints were already delivering the value in form of functioning software.

## The RFP sprints

What to focus on during the Agile RFP:

- **Conduct planning and consulting sessions with PO** (or Product team)
  - Separately for each vendor so they can be safe to ask the questions based on their domain and engineering expertise.
- **Conduct regular demo** (separately for each vendor), evaluation criteria could contain following:
  - Demo itself (professional, clear, on time, achieved scope?)
  - Business value delivered (acceptance criteria and value thinking)
  - Agility (cooperation, learning, openness, visibility)
  - DoD fulfilled (roadmap presented, demo happened, code in repository, agreed documentation e.g. as commented code)
- **Provide whatever support** if needed
- **Assess the vendor continually and synchronize quickly** after each demo. For this reason you prepared the internal evaluation template.
- **Run regular retrospective** with internal team and adjust what is needed.
- **Explain to vendors future Agile Contract** model and send it for review if planned (current mini-project velocity will be set as a basis to avoid any tricks and cheating in the RFP mini-project).
- **Ask vendors to propose release plan** for the first release of the subsequent project to see how they understood the project and think about the value for customer.

Be also prepared to change and tune up the sessions, templates and way of working based on the real findings. In other words, **you will be also iterative and change according the vendors' feedback and the real needs.**

We used verified vendor's velocity from RFP mini-project as the baseline for the future Agile contract, to be realistic in the competition and not to run overtimes or have it as door opener project. Of course, it was taken as baseline with +/-% interval, not as hard parameter.

One point I will make here is the visible progress and attitude of the vendor. Usually after sprint 2 or 3, patterns start to emerge and vendors stand "naked in the light". Everybody can see their strengths, weaknesses, key differentiators and communication style and there is common agreement among all internal people on it. In such environment every vendor's progress is very visible and appreciated. Quite objective and fair, right?

## There is always one winner: you!

There should be always one winner after the Agile RFP mini-project and it should be you, the customer. Now you should own several working software implementations and possibly some can be already shipped to LIVE environment with just slight code additions! All this just after one month, or so.

If you cannot ship to LIVE, there is still quite high value. Now, you should have cleared your needs and the project goals, you should get common agreement among internal business people and decision makers and have also plenty of ideas visible and demonstrable in the form of working software.

So, how to close the RFP project and how to continue with the project itself?

- **Run separate overall demo** with all vendors (check planned vs. delivered, changes, progress, learning, feedback).
- **Conduct the final vendor assessment** based on your selection criteria gathered continually after each demo.
- **Check the DoD** defined for RFP. Have all vendors fulfilled it? Can you pay them for the delivery?
- **Have common retrospective** with all vendors. Give them feedback, thank for their effort and also ask for the feedback on the routine as the whole.
- Ask vendors for the **updated** price and roadmap **proposals** for the future project being purchased.
- **Sign an Agile contract** with the winner.

Remember, the goal of Agile RFP is not to catch vendors on something they don't know. The goal is to uncover it early enough, see how they communicate, learn and progress, see how they solve these situations, learn together and set the basis for future Agile project you are purchasing for.

**Be also careful about your K.O. criteria.** I recommend not to state Agile experience and references to similar Agile projects as K.O. criteria. Why? Many flexible and skillful vendors can teach Agile way of working during the RFP and overcome the experienced one. And this is good for you as a customer. Isn't it? I experience this again and again.

If proposed price is one of your most important selection criteria, I would like to stress its relativity. **Higher price with higher vendor velocity can actually mean cheaper project and higher customer value at the end of the project.** More quality and efficient vendor can have e.g. 20% higher velocity. Such vendor will therefore deliver the

value earlier than others. You as the customer can consume these benefits earlier as well (and generate already higher revenue). Therefore, you can afford to pay such vendor more money and still have higher ROI. Don't forget about this math when making final selection!

I probably skipped many important details, so you can have even more questions than answers. Do not take this example as one-and-only approach. I rather wanted to show, how Agile principles can be applied in other areas in business.

How does it sound to you? Have you already tried? What was your experience, value, biggest risk? Please share your experience with me at [jarek@diffier.cz](mailto:jarek@diffier.cz)

## ABOUT AUTHOR

**Jaroslav Procházka**, Agile and Innovation Mentor, Czech Republic



Jaroslav is utilizing his strengths for more than 13 years in IT industry in various development, maintenance and management roles and positions. He has 10 years coaching and mentoring experience in Agile and distributed environment. He has worked with about 700 people and has trained few thousands people so far. He published a book on Agile and Lean support and maintenance in 2011 (Grada publishing) and other free e-books on Agile and Lean topics (can be downloaded from [www.differ.cz](http://www.differ.cz)).

Jaroslav focuses on longer-term initiatives conducted together with management (more systematic and sustainable) also including soft skill workshops and coaching to utilize his experience and interest in psychology. His favorite tools are Kaizen workshops and (Agile, Lean ITIL and Service Design) games, because people find their “aha” moments by themselves.

Jaroslav earned his Ph.D. at University of Ostrava in 2007 and was also researching and teaching Software development and Information Systems for 10 years there. He speaks at international conferences like Agilia Conference, IBM RSDC Conference 2009, Information Systems Development 2010 or International Conference on Global Software Engineering 2011.

Twitter: [@JarekProchazka](https://twitter.com/JarekProchazka)

LinkedIn: [Jaroslav Prochazka](https://www.linkedin.com/in/JaroslavProchazka)