From 3 failed apps to a big success in a year – and how Forge made it possible.

From the PATROL FOR JIRA project

A year ago, I typed '*forge create*' and it changed my life.

I was looking for a platform for building my next startup and decided to try my luck with Atlassian addons. What got the attention was one of the most competitive and vast ecosystems with many companies dedicated to Atlassian-specific development. Also, my current employer had Jira and Confluence, all my previous employers also had it – and I got the first-hand experience of what could make companies more productive, innovative and successful, if I manage to turn it into an Atlassian app.

Discovering Forge

Now, the problem was to learn how to do it – when I first checked out the Atlassian development framework, most of the resources pointed to Atlassian Connect. It was powerful, complex… but did not feel much like 2020, to be honest. Everything took 10x time in Connect – from checking out the code to running a hello-world app, it was like participating in heavy lifting session. Sure, you can train yourself to lift but it takes time, sweat and blood. Soon after trying and abandoning Connect, mainly due to the lack of time and uncertainty if it is a good idea to commit to it, I read about Forge, a new kid on the block, the framework that was supposed to make development of Atlassian apps simple and easy like walk in a park.

So, I typed '*forge create*', tunneled into the container running the app, opened Jira – and got my new app up and running. It took me about a minute – that's when I realized that **this** is the tool I was after, the perfect delivery engine to bring my ideas for productivity apps to the world.

How I won a battle and lost the war

At the same time, Atlassian announced a new round of hackathon Codegeist, where everyone was encouraged to build and submit their apps – and I had an idea that could be just great for it. AI helper that helps teams be productive, AI Scrum Master – Scrum mAIster. I quickly bootstrapped the concept and got to work – two months of long nights, and by the deadline I had a Scrum Maister in "production" environment. It used all possible Forge features, employed machine learning models to help people write better Jira issues, run more efficient sprints and generated inputs for retrospectives. Everyone I talked to, loved the idea. Including Atlassian judges. I was blown away when Scrum Maister won Codegeist 2020 and snatched the grand prize for Forge apps.

I was more than happy that day – finally, my dream came true. One of my ideas turned into a real MVP, and was off to a great start. I got to the hard work. Six months later, I published Scrum Maister on Atlassian marketplace, and it was one of the first Forge apps. It got a bunch of new features, new designs and capabilities. I thought, it will save people tons of time and make teams around the world so much more productive. I could not be more wrong.

The first week of Scrum Maister on the Marketplace passed. I was checking stats many times a day. Grand total - zero new installs. I order some ads online, I see people visiting the page, but nothing happens. The first month of Scrum Maister, no production use of the app. I was disappointed and curious – where did it go wrong? Why does an award-winning idea not find the same response in the market?
I get to mind mapping, retrospecting and self-reflection - and soon figure out the reason for my first failure. Development of the new features was so quick and simple, I took it too far and never found a real customer or a team to iterate with. The usage and use cases were based on my own experience and were far away from the real world where people tend to think and do things differently. Also, Scrum Maister was like a swiss knife, it had everything in it, while Atlassian Marketplace was much more specialized, and people turned to it mainly when looking for powerups and improvements to their Atlassian products.

Now it was as clear as a day what to do next.

How I tried and failed again - twice

While keeping Scrum Maister alive and slowly working on expanding its footprint and brand recognition, I type '*forge create*' and start my next app – Whistle Willow. Thankfully, with Forge, it takes minutes to ship your hello-world app, and hours to get the first prototype of a proper product. This time, I turn to real world for feedback and find something that looks like a real problem – whistleblowing. EU directive on Whistleblowing will require all companies larger than 250 people to provide their employees trusted and secure channels to blow the whistle, which means, report nonconformities, raise alerts, spot bad behaviors or wrongdoing – all without a risk of being prosecuted. In my mind, having people reporting these issues through Jira is a good idea - as they already use and trust the product.

Two weeks later, I have my MVP ready. With Scrum Maister, it took 8 months. Learning Forge, Forge UI module, building up my own codebase of helpers, primitives and tools allowed to dramatically cut time from idea to the market. Furthermore, only one week later I have the third product MVP in production – Whistle Willow for Confluence.

When I release them to the Marketplace, I get into the Scrum Maister's pitfall once again. No installs after first months. No interest in the topic.

In the retrospect, I understand that the whole whistleblowing concept and EU Directive still require education and promotion. As whistleblowing is often driven by HR departments, they turn to existing vendors that specialize in this type of systems, and rarely think of Confluence or Jira as a place for people to report wrongdoings. I still believe that when the topic becomes more urgent in December, Whistle Willow will get its share of customers. But the lesson is: the right timing, marketing and positioning are as important as a good idea.

Why I won't fail again

After falling and failing for three times in the row, everyone would ask themselves – is it worth it? Spending time and putting effort into something that does not translate into money, fame or helping people and companies to be successful. Maybe it is time to close the shop and try some other field? I had these thoughts myself – but every time Forge released a weekly update with new features, I was thinking what else I can build. The newly released capabilities, elements and APIs were exciting and tempting.
And I knew I had everything lined up – tools, experience, Forge know-how and valuable lessons from three failed launches.
So, this time, I decide to go for a certain hit. My daily job is in cybersecurity – for living, I help protect information and resources against hackers. On the list of targets, Jira is among the top. While Jira Cloud is secure in terms of infrastructure and operations, I found out that just on compromised account, vulnerable third-party app or leaked API token can do as much harm as full-blown hacking attack with the most skilled hackers.

So, I typed '*forge create*' and got to work, with no less enthusiasm than the first time. Three weeks after, it was ready.

My fourth app, Patrol for Jira, is intended to protect what's inside of Jira against internal and external threats. In the essence, it is a security scanner for Jira – it finds vulnerable apps, hard-coded passwords in items, old or leaked API tokens, detects suspicious admin actions and much more. It is simple, useful and has a clear purpose. Also, there are no direct competitors. Its audience - Jira admins and security teams, folks who are frequent guests of Atlassian marketplace. And this time, I believe, everything is set for a successful launch. The app was submitted to Codegeist 2021 hackathon – and if it does not score any prizes, I'm sure it has a solid future in the real world.

Fourth time is the charm

It took me three failed attempts to come up with this product and implement it to fit the market. With other frameworks and ecosystems, it would have costed me at least three to five years of work, good amount of money spent and definitely much less courage and inspiration left after learning from own failures. Forge allowed me to prototype and go from zero to MVP incredibly quickly, in a matter of 30-40 work hours per app. And the results can be published to Marketplace and start making money and helping people almost right away. With the built-in Storage API, FaaS and triggers, one can operate apps with virtually no owned infrastructure – which also makes the entire operation model not only incredibly cheap but also much more secure.

At the end, I do not give up on my previous projects. Scrum Maister starts to get more interest from teams who search for simple but efficient Scrum helpers. To fit in, I cut off some of Maister's beloved features just to make it less bulky and more fit for the job it is supposed to do – help teams all the way from grooming to retros.

With Whistle Willow, I'm waiting for a right moment and working on a better messaging. Also, as I learn new tricks and develop more utils and primitives, I port them into all of my products. This way, they all get slightly better with time, like a good wine, and just wait for their moment to shine.

For Patrol, it will be surely an exciting journey, with the ambitious goal of making the Jiras around the world more secure, from internal and external threats. And believe it or not, I already got the idea for my fifth app – Patrol for Confluence, and hoping to get MVP ready... in a week or two. Would it be possible elsewhere than in Forge? I do not think so.

Igor Andriushchenko
Codegeist 2020 winner, Forge enthusiast, Optimist and Life-long learner