

SCRUM is not agile enough

You really need that change. That small extra step that brings your company the innovation everybody wants. Or the fix to the liability risk you are facing. No problem. Your team is just wrapping up a sprint today. You may have something in two weeks.

We know some things can't take two weeks. They can't even take two hours. Business can't always wait two weeks and SCRUM is just not fast enough.



Change is hard and costly

The more knowledge and regulations are involved, the more expensive change becomes. It's like changing the string of a spider web. Change one thing and risk breaking it all.



You need work from third parties

With most changes, you need work from outsiders. You are never sure how much work it will take and how much it will cost. And your change is queued on the roadmap.



IT people are scarce

IT staff is hard to come by. But you need them to make changes to your system. That means you have to prioritize work.



The business is not in charge

The subject specialist has the power of knowledge, the IT has the power of change. And neither are eager to give up that power. The process is leading, not the business.

“Small changes at the right time can have an enormous impact.”

Hyper-agility: being able to change in a second

We believe the answer to these challenges is hyper-agility. We like to refer to hyper-agility as a state of elevated agility - the capability of an organization to change direction in a second. Small changes at the right time can have an enormous impact.

There are three fundamental requirements of hyper-agility.

1. minimization of system impediments.

Any change should not require updates of other systems, break integrations or - from a business perspective - involve authorization by people not impacted by the change. Contributors to this state are API versioning, backward compatibility, and no-code configuration.

2. minimization of key man risks

The more legacy systems, the more key man risk exists. Old code is often only comprehensible to the one developer who was around when it was written. To avoid paralysis, the architecture must be modern and as clean and self-explanatory as possible.

Transparent processes, accessible configuration, and well-documented business rules and integrations lead to better understanding and faster adaptation and change. Organizations should favor configuration over development.

3. minimization of redundancy work

Nothing is more frustrating than having the tech in place but not being able to launch because of blocking preconditions. These preconditions can vary from communication, compliance, and lack of documentation to API interdependencies. Anything strictly a matter of execution within a system not requiring human contribution (anything non-emotional) can and should be automated, either by system automation or by business rules and process management tooling.

If you eliminate impediments, dependencies on specific people, and the vast amount of administrative hassle in your change process, you might be able to achieve hyper-agility. You might just win the race. Whatever industry you are in.

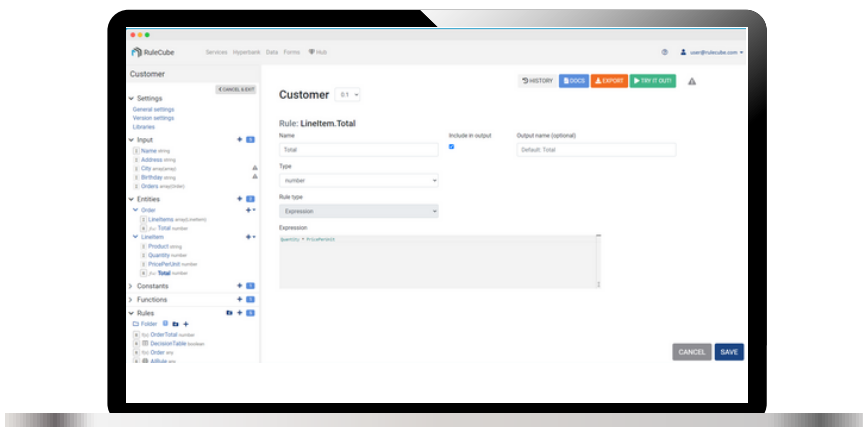
About Rulecube

If you want to see more of our low-code business rules and api platform, contact us for a personal demo. In just 15 minutes we can show you just how easy it can be.

+31 (0)6 53 67 43 19

contact@rulecube.com

rulecube.com



**Free yourself from hidden
hardcoded business rules**

**Take back control of change in
your organization**

get your demo today