

04 HSE's Advantages



Disclaimer: This article is meant to discuss the benefits and advantages we (Hampel Software Engineering) experience on a daily basis when working together with our customers on their projects. It is not about the technical aspects of software architecture and LabVIEW implementation details!

With the framework that we used to develop ourselves a few years ago, which is based on queues and strings, we were very flexible as it was very weakly typed. That flexibility came with a need for very detailed documentation when sharing code with others.

In comparison, the [DQMH® framework](#) has proven to be a much better fit for our way of working:

A Single Tool

We stand by our (controversial) motto that we like DQMH being our hammer, making all our customer projects look like nails. And it's definitely possible! Without this approach, it would be virtually impossible for us to work on the large number of projects that we do with our very small team.

Strongly-typed API

DQMH creates the public API in the form of proper VIs with proper connector patterns (= parameters) just as you go. It even offers dialogue popups for entering the documentation. As soon as your module is finished, so is all the documentation your co-workers will need for most use cases!!

Better Practices

For our customers, DQMH helps to adhere to various better-practice approaches. The familiar QMH architecture helps flatten the learning curve. Especially small teams benefit A LOT from its ease of use, its self-documentation features and all the concepts it introduces (like modularity, synchronous and asynchronous communication, ...).

Maturity

On top of all that, it is used by 1000s of people all around the world. DQMH is probably the most popular 3rd party framework. It is distributed by NI via their Tools Network. Many very, very clever heads have discussed its pros and cons - so it's quite safe to assume that most kinks and bugs have been ironed out already.

Flexibility

DQMH can be used together with other architectures, frameworks and tools. You can start a migration from another framework to DQMH by implementing a single module, testing it to work stand-alone, and start using it in the existing project. That way, engineers can get accustomed to DQMH and start appreciating the benefits before jumping into the deep end.

Availability

As DQMH is publicly available for free (commercial license available upon request), you're not dependent on one single consultancy at all. You can keep on working on your own, or employ a different supplier or consultancy. Chances are if you hire new staff, they are already familiar with DQMH (and if not, there's an abundance of documentation available).

The Hammer

We've been heard to claim that DQMH is "our hammer". This is true, but not in the sense of [Maslow's Hammer](#). What we mean by this is that DQMH is a very powerful and versatile tool, and we want to use it and apply it as often as possible, to become as proficient as possible.

You can find more information on cognitive biases in Wikipedia's article on the [Law of the instrument](#).

What's your take on this? What's your opinion? Do you have questions about [DQMH](#) or our (Hampel Software Engineering's) [way of working](#)? Please don't hesitate to [reach out](#)!

From:
<https://dokuwiki.hampel-soft.com/> - **HAMPEL SOFTWARE ENGINEERING**

Permanent link:
<https://dokuwiki.hampel-soft.com/kb/labview-frameworks/dqmh/hse-advantages>

Last update: **2026/02/18 11:32**

