

[How DORA capabilities are shaping software delivery](#)

Breadcrumb

1. [Home](#) /
2. [Print](#) /
3. [Pdf](#) /
4. [Node](#) /
5. [Entity Print](#)

Insight navigation

- [Latest insights](#)
- [Latest news](#)
- [Articles](#)
- [Case studies](#)

[Download PDF](#)

Mar 2024

How DORA capabilities are shaping software delivery

As a bespoke software developer, you really only have one goal: deliver high-quality software that meets the needs, and exceeds the expectations, of your clients

Categories Bespoke Software, Software Delivery & Consultancy, Software Operations & Support

Mar 2024

-
-
-
-



[Luke Kirby](#)

Software Consultant

Email

luke.kirby@waterstons.com

Consciously or not, you'll always be asking yourself what you can do better:

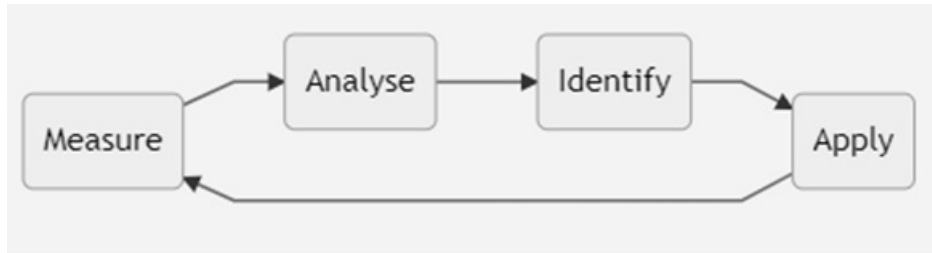
- How can I be more efficient in my workflow?
- How do I improve the developer experience to get the most out of my tech and people?
- How do I apply new teachings and learn from my mistakes?

DevOps Research and Assessment (DORA) is a research program focused on understanding and extracting the key capabilities exhibited by high-performing software teams, presenting them in [annual reports](#). Regularly measuring against these capabilities

provides both an incentive and a means to answer these questions, and monitor the effectiveness of the ideas and solutions applied. Crucially, it also shows how measurements and strategies differ between client projects, and prompts discussion as to why this is the case.

How we are using DORA

The process for a given project looks something like this:



1. Measure against DORA metrics to indicate where you are currently.
2. Analyse the results, accounting for the bespoke requirements, challenges and limitations of the project.
3. Identify the DORA capabilities you'd like to focus on, based on the analysis.
4. Apply practices and strategies in your workflows designed to satisfy the capabilities in focus.

The cycle is then repeated frequently, with each subsequent analysis trying to answer the following questions in particular:

- Do the metrics indicate that the applied strategies have been effective?
- If not, then why – is it a limitation of the strategy, or what is being measured, or how it's being measured?
- Either way, how can you improve and refine your strategies/measurements?
- Should the focus change to other DORA capabilities?

What really makes this sing is regularly sharing these findings across projects, discussing why some strategies might greatly benefit one project but hinder another. It's not all about the numbers – the secret sauce is in taking the time to reflect on what you do and how you do it.

Measuring Tools

There are many approaches to consider in not just *what* to measure, but *how* to measure it, which ultimately dictates the tools used.

The most readily accessible tool is DORA's own [Quick Check survey](#), which baselines your own results against the industry averages in four key areas: lead time, deployment frequency, time to restore, and change fail rate.

For more specific measurements, we've also built our own internal platform for aggregating data about branches, pull requests and deployments. This is hugely advantageous for the ability to extend the bespoke functionality of the app to suit the desires of one project team, then being able to share that across all other project teams.

Client Project Case Studies

Project A

The first target area to greatly improve developer experience was to shift away from large pull requests (which often included re-factoring or small fixes alongside full feature implementation) that required significant time investment and cognitive load to peer-review. By applying techniques to break down work into smaller batches, measurements have shown subtle trends towards a larger number of PRs being opened/closed and a lower mean PR lifetime.

The obvious next step was to improve the continuous delivery of changes to production environments. Strategies for increasing deployment frequency included:

- Writing up UAT test plans to provide clear steps and success/failure conditions for testing new features/fixes that any member of the team can follow.
- Making writing the test plans and pre-populating release notes a required part of the PR process, allowing changes to be deployed to UAT at the drop of a hat.
- Introducing "Deployment Trains" – a regular schedule for deployments. If a change "misses the train", it simply goes out on the next one. Each deployment is relatively small, therefore requires less time and effort to test before shipping to production. Overall deployment frequency has increased from about 1 per month to 1 per week.

Another key strategy has been the use of feature flags to distinguish 'deployments' from 'releases'. In this case, feature flags have enabled gradual rollout of a brand-new system by only releasing a subset of functionality to a subset of users initially, then expanding both over time as features are tested and fixed with minimal user impact.

Project B

One of the largest obstacles to continuous delivery in this project lay in the build times, which through optimisations have been slashed from an average of 115 minutes down to 25 minutes, increasing the throughput to UAT.

However, further refinements in this area are difficult because the client fully controls the release cycles; the developers have limited ownership of the DORA capabilities for the project. To offset this, greater emphasis has been placed on identifying bugs or issues early, employing strategies such as:

- Increased focus on testing during development through unit tests, integration tests and the addition of scripts/walkthroughs for testing specific areas of the solution.
- Improving the quality of work items by having more detailed acceptance criteria and highlighting which areas of the system should be tested in relation to the change.
- Actively tracking bugs across environments along with resolution reasons to validate the high quality of deployed code and provide a solid framework for maintaining that standard.

Summary

DORA metrics and capabilities provide an excellent framework for shaping both faster, higher-quality software delivery to your clients and an improved experience for developers. Whether the measurements are quantitative or qualitative, ultimately they drive discussion and reflection. Every project will have its own bespoke requirements and restrictions, so it's crucial to ensure that developers are empowered to make decisions they feel will reap the most reward. What matters is that these lessons are shared, analysed and built upon.

For more information about DORA, its principles and the difference it can make in your organisation, get in touch with Luke at luke.kirby@waterstones.com

<https://waterstones.com/print/pdf/node/8708>