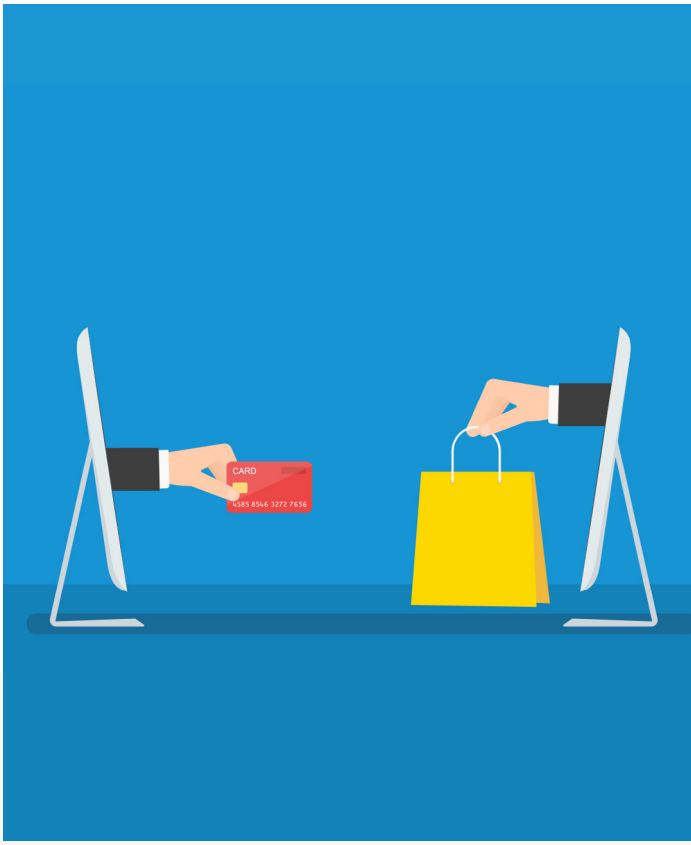
Continuous Delivery in a highly regulated environment.

Achieving PCI DSS compliance quickly, using modern software engineering practices.

EQUAL EXPERTS



Equal Experts helped a large, Middle East-based payment solutions provider build a new payment gateway.

The solution needed to improve their ability to quickly launch new features to the market, as well as expand into new opportunities and territories more easily. It had to be structured to keep them ahead of the competition and offer a SaaS white-label payment gateway to other financial institutions.

The challenge was to deliver a scalable Payment Card Industry (PCI DSS) compliant platform within the client's ambitious timescale. This involved a very different approach to Risk, using new ways of working. In addition, the security auditors were unfamiliar with the approach being pursued to achieve PCI compliance and needed to be convinced of its value.

1.5 hours

from code commit to production.

800+

Now releasing over 800 times a year.

About the client.

The client is a major provider of online payment solutions to merchants and financial institutions in the Middle East - with an annual turnover of c.$350m.

INDUSTRY

Financial Services

ORGANISATION SIZE

1,300+ employees

LOCATION

Middle East

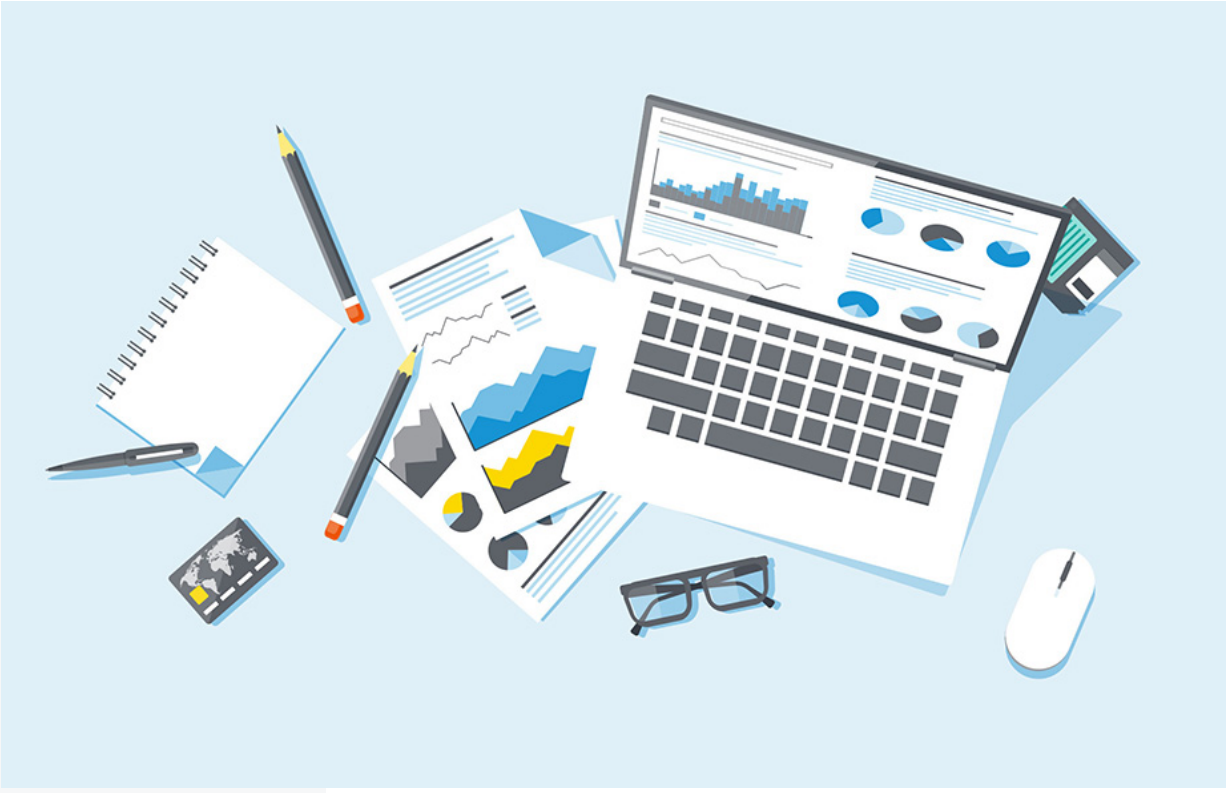
SERVICES

Platform build; knowledge transfer

Continuous Delivery, and how to keep the highest standards of compliance.

When it comes to regulation, partial compliance is the same as no compliance. This created interesting Security challenges when it came to Continuous Delivery. How could we achieve compliance whilst continuously delivering outcomes within tight timescales?

The Qualified Security Assessor (or QSA) audit required evidence of compliance to gain certification. Often, this can be as simple as providing a screenshot of a one-off manual process. **At all times, the team considered the value of quick and inexpensive compliance compared to the longer-term value of automation.** In the near term, they took a hybrid approach – automating where they could and working manually to demonstrate compliance in other areas. This highly pragmatic approach was taken to help launch against very aggressive timescales. Post-launch, everything was automated, thereby enabling rapid, unlimited scalability downstream.



Engaging early, often, and throughout, to work within the regulations.

Constraints, such as regulatory requirements, can be frustrating. But viewed from another perspective, they can end up being immensely valuable to delivery teams. Unambiguously specified detail about compliance is not up for debate or interpretation.

Adherence to the regulations was measured and ultimately certified, by a QSA. As one of the principles of Continuous Delivery is to **engage early, often, and throughout the process** we worked closely with the QSA from the very beginning of the engagement. This helped the QSA understand how the requirements were being met in an unfamiliar way – in this case, agile delivery in the cloud.

At that time, our client approached compliance by tracking releases through tickets. The team challenged the process. They recognised that instead of raising a ticket for every piece of work, the regulatory mandates could be met by tracking progress in Git. The QSA confirmed that the new approach would work.

Effectively convincing stakeholders about new ways of working.

Dry run compliance exercises were run throughout the project. This ensured the team was on the right trajectory towards complete PCI compliance. The QSA's experience was of more traditional ways of working and needed this time to build up sufficient context and confidence. This **iterative, joined-up approach offered everyone a chance to challenge and educate the QSA** and each other on the principles of modern software delivery.

One example is: when it came to intrusion detection/prevention, the client had a variety of options, such as buying a product off the shelf. But there are many ways to detect intrusions. In the cloud, there were various native components that could be used, with which the auditor was unfamiliar. The team explored common scenarios to address the concerns of the QSA and other unconvinced stakeholders, by showing how security objectives could be met without spending the client's money on unnecessary products. The alternatives showed how they could build solutions themselves by leveraging cloud-native applications that were as good, if not better, than the traditional on-premise solutions.

At the same time, the team was comparing the value of the client's traditional approach, which was to buy commercial off-the-shelf (COTS) products, to building solutions using cloud-native applications. The advantage of COTS is that they guarantee compliance; the disadvantages include:

- The products seldom being designed for cloud-native use cases
- Non-ownership of the solution
- The risk of being tied into potentially expensive, long-term contracts
- The fact that it is easy to lose sight of the security threat you are trying to defend against

To proceed in a new direction required a mental shift, so to make the best decision with the greatest buy-in the team ran a series of quick proofs of concept, to evaluate the difference between the current decision-making mindset and the proposed new approach.

How to ensure Continuous Delivery = Continuous Compliance.

It was vital to continuously comply with the regulatory checkpoints. With Continuous Delivery minimising the time to Production, to remain efficient and effective security and testing had to be embedded into the pipeline. Specific tests were run to ensure compliance with specific regulatory requirements, and what's more, a full and transparent audit trail had to be maintained. This was achieved through documenting what they were releasing and why they were doing it, and, encapsulated in JIRA tickets then linked and stored in Git to provide a single, and fully traceable, source of truth.

For Continuous Delivery a key metric is time to production. To avoid an emphasis on velocity causing a proliferation of bugs, the team relied on feature flags. **The aim was to ensure new features didn't break the old ones,** and this was achieved by integrating testing and spinning up parallel testing threads to radically reduce the total test time to mere minutes.

As the team delivered more features, more tests were added, until they numbered in the many thousands. That introduced a lag between the test being run and the results. By revisiting the architecture of the tests, the team worked hard to minimise the testing time, whilst continuing to provide confidence and reassurance to the business.

When continuously releasing code, it is a reliable measure to say that if the team estimates the work will take more than a week, then the change is so large the work needs to be broken down into smaller deliverable units of functionality.

The step-by-step approach to creating more productive Release Governance.

In the past, decisions on releases were complicated, involving multiple sign-offs, with decisions typically taking weeks. When the issue reached the Change Advisory Board (CAB), it was discussed by a large group, and only if it was unanimously agreed could a ticket be raised. After that, further delays were expected as the ticket was in a queue to be approved and the code released. Many unnecessary hours were consumed at the expense of more productive work.

Creating the optimal Governance process was both important and a challenge. At the beginning of the engagement, there was a call with the CAB every week, but over time the frequency dropped to monthly calls. Together with the client we worked on transitioning from a largely uninvolved committee of senior stakeholders making decisions, to devolving the authority to approve work to those with the most informed perspective. This ensured the team operated within PCI whilst maintaining the ability to release hundreds of times per year.

The aim became how to best communicate the progress of the work, thereby giving confidence to the stakeholders using evidence – e.g. this feature was released on time, within budget, and fully compliant without creating bugs or causing production downtime.

Want to know more?

Are you interested in this project?
Or do you have one just like it?
Get in touch. We'd love to tell you more about it.