

# THE IMPORTANCE OF DEVELOPER-TO-DEVELOPER COMMUNICATION WITH CUSTOMERS



**Overview: Early in 2019, Compuware developers traveled to Edinburgh, Scotland as well as other locations in the UK to spend the week with mainframe developers at our customer Lloyds Banking Group. Here, one of our developers relays his experiences.**

It's a misconception that the success of large software products rests on a vendor's product management team and its software developers knowing exactly what customers need. In truth, from system permissions to development priorities, vendors and customers have different perspectives. Without developer-to-developer communication between a vendor and its customers, the vendor cannot solve customers' most grueling issues.

Thanks to a recent visit to Lloyds Banking Group in the UK with a few of my Compuware developer colleagues, and then a follow-up visit from Lloyds' developers to Compuware's Detroit headquarters two weeks later, I understand more than ever the importance of maintaining this developer-to-developer communication with our customers.







## **Solving Problems, not Symptoms**

The core customer issues a software vendor's products should solve are not always made obvious to either party. Through inadequate channels, the main problem can be lost, miscommunicated or left undiscovered. Oftentimes, only the symptoms of the real issue are communicated, due to the complexity of enterprise software.

Sprint reviews are great opportunities to work with customers to ensure a product adapts to their vision. However, while using this meeting in such a way is helpful, it is also important for vendors to have face-to-face discussions with customers, as problems emerge when these groups are not completely in sync.

Furthermore, data analytics also are not always sufficient for pinpointing issues customers face, though they certainly reveal more than we could ever understand without them. Still, without having a way to study a customer's development process and their use of your products—such as by visiting them on-site—it can be difficult to understand their core issues. Thus, a vendor's developers are bound to fix symptoms, not problems.

# Achieving Force Multiplying Effects

When developers from both parties are able to meet face-to-face for an extended period of time, it's easier to focus on the real problems. In our case, this level of collaboration results in products that are force multipliers for customers like Lloyds, who is transitioning from waterfall to Agile and DevOps, and Compuware as we accelerate our journey of DevOps continuous improvement.

An example of a resulting force multiplier is Compuware Topaz, our suite of Agile mainframe development and testing tools that integrate into a DevOps toolchain. As we continue to evolve this product, we're focused on both the user experience for customers like Lloyds as well as our own developers' needs, as new features are based on problems both our customers and our own developers encounter.

The way Lloyds' developers utilize our products to solve problems can be completely different from how I may be using them. Understanding these differences can lead to enhancements or knowledge transfers that enable us to continuously improve our products to serve customers better.

As quality, velocity, and efficiency continue to increase at Lloyds, their demands as a customer adjust. Enabling developer-to-developer communication through our recent visit was a powerful way to learn more about how we can help them.

Being customer-obsessed shouldn't stop at the executive level; the entire development organization should be involved in analyzing customer needs, improving the creative selection process and building customer relationships.