

CASE STUDY

Dixons Retail—Blue-Green Deployment for Point-Of-Sale System (2008)

Dan Terhorst-North, technology and organizational change consultant, and Dave Farley, co-author of *Continuous Delivery*, were working on a project for Dixons Retail, a large British retailer, involving thousands of point-of-sale (POS) systems that resided in hundreds of retail stores and operated under a number of different customer brands. Although blue-green deployments are mostly associated with online web services, North and Farley used this pattern to significantly reduce the risk and changeover times for POS upgrades.²⁹

Traditionally, upgrading POS systems is a big-bang, waterfall project: the POS clients and the centralized server are upgraded at the same time, which requires extensive downtime (often an entire weekend), as well as significant network bandwidth to push out the new client software to all the retail stores. When things don't go entirely according to plan, it can be incredibly disruptive to store operations.

For this upgrade, there was not enough network bandwidth to upgrade all the POS systems simultaneously, which made the traditional strategy impossible. To solve this problem, they used the blue-green strategy and created two production versions of the centralized server software, enabling them to simultaneously support the old and new versions of the POS clients.

After they did this, weeks before the planned POS upgrade, they started sending out new versions of client POS software installers to the retail stores

over the slow network links, deploying the new software onto the POS systems in an inactive state. Meanwhile, the old version kept running as normal.

When all the POS clients had everything staged for the upgrade (the upgraded client and server had tested together successfully, and new client software had been deployed to all clients), the store managers were empowered to decide when to release the new version.

Depending on their business needs, some managers wanted to use the new features immediately and released right away, while others wanted to wait. In either case, whether releasing features immediately or waiting, it was significantly better for the managers than having the centralized IT department choose for them when the release would occur.

This case study demonstrates how DevOps patterns can be universally applied to different technologies, often in very surprising ways but with the same fantastic outcomes.