It can be extremely difficult to change release dates from ten weeks to a single week. And that just what Ernest Mueller assisted with for the company Bazaarvoice. Bazaarvoice had $120,000,000 in revenue and was seeking an IPO at the time. This was a Java application that started in 2006. The goal was to reduce their ten-week release schedule into just two weeks and eventually one week.

The software engineer teams switched to Agile for their two-week deployments. They also began to break up sections of their code into microservices. (Kim et al., 2016) At there first attempt at the two-week schedule it Mueller states “It didn’t go well.” (Kim et al., 2016) and they had 44 customer incident reports to fix. Management did not seem to want to progress with the two-week model.

There we three main problems according to Ernest Mueller that made the attempt a challenge. First there was not a lot of testing during any of the stages. (Kim et al., 2016) Second, the way they did their version control allowed the developers to release the code immediately. (Kim et al., 2016) Lastly, the teams of developers were also working on other releases which would cause issues with the larger application. (Kim et al., 2016)

Ernest Mueller realized that their “application deployment process needed to be stabilized” (Kim et al., 2016) which required integration continuously. To cut down on the issues developers were able to stop working on features to focus on creating automatic tests. (Kim et al., 2016) The developers also reworked their versioning control into a trunk/branch model. Which allowed them to reduce their 44 incidents in January to March 6 being “five days late, five customer incidents” (Kim et al., 2016) to further reducing later in the month on March 22 to having delivered deployments on schedule with only one customer incident and eventually to have no customer incidents. And shortly after reducing their releases from two weeks to a single week. (Kim et al., 2016)

Something to note that is common to a lot of these case studies seems to be the integration of automatic testing. If it can be automated, create the test for it and test frequently. Especially test during the unit tests because they are of the most secluded portions of code and often time it takes a few minutes.

Refence:

Kim, G., Debois, P., Willis, J. O., & Humble, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. https://dl.acm.org/citation.cfm?id=3044729