Situation

1. Clients experience bugs that developers failed to catch during testing.
2. Developers spend a lot of their time to deploy their code on the production servers.
3. Clients experience downtime from issues during this process.

Customer satisfaction is low because they experience bugs and we fail to deliver on requests on time. These are requests of features that generate value for UdaPeople.

Clients experience bugs that we failed to catch during testing

1. Developers spend a lot of time to test their changes manually. It's hard to test the complete set of changes, and even then it's possible the changes affect a seemingly unrelated part of the application.

We can invest some time in setting up a suite of automated tests that check the whole application. This results in less bugs in production and less developer time spent testing.

Developers not working on value generating features

1. Developers spend around 1 hour per change to get their code on the production servers.
2. Depending on the size of the change, an additional 15 to 30 minutes is needed to test their changes.
3. Any failure in this manual process leads to a 30 minutes rollback

A pipeline that automatically builds, tests and deploys the features would save developers approximately 2 hours per deployment. Most features require multiple deployments. We would reduce our time to market as well as release new value-generating features more quickly.

Clients experience downtime during code deployment

1. We manually update servers with the most recent changes. There is no easy switch from the old version to the new version. Clients experience moments of downtime during this process.

We can setup the pipeline in such a way that a new version of the application is launched. After it is verified by automated checks that the new version works, the client is routed to the new version, after which the old version is terminated. We will reduce downtime from deployment related crashes and bugs.

Conclusion

With an automated pipeline we can achieve:

1. Less bugs in production and less developer time spent testing
2. Reduced time to market for revenue generating features.
3. Reduced downtime from deployment related crashes and bugs.