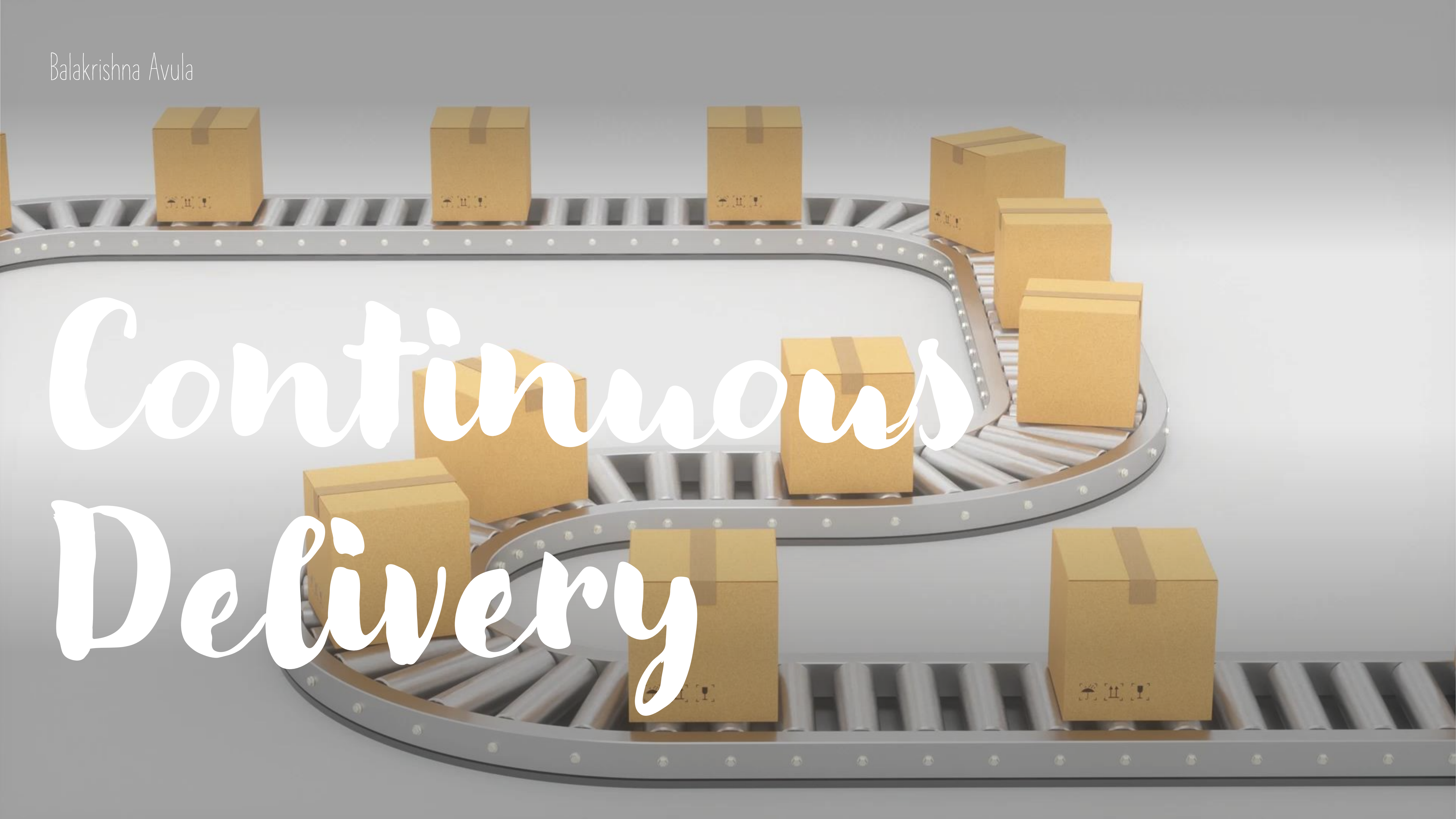
Balakrishna Avula

Continuous Delivery



What & Why

- Continuous Delivery enables agile application delivery using Continuous Integration (CI) and Continuous Deployment (CD) to
  - Improve time to market (e.g. Our flag ship UdaPeople product bug fixes can be delivered in hours instead of days for our customers)
  - Reduce costs of our product development (e.g. a feature can be added in 10s of man hours instead of 10s of may days)
  - Improve product quality to increase sales (e.g. automated quality checks and automated testing improve our product thereby improving our reputation and sales)

Current

- Manual compilation and testing followed by manual reporting on the progress of the deliverables.
- Use of a source control system and individual or shared test environments.
- Production release requires technical and business checkouts before acceptance.
- Production release requires downtimes which is a limiting factor when we compare our product to competitors.
- Our 50 developers need 50 test environments (each costing about £1000 per month) costing about £600,000 just for testing environments.
- Major product releases only happen every 2 years and maintenance releases happening every 6 months. Bug fixes taking up to a week which is one of the major complaints from our customers.

Future

- Use of automated code checks and automated unit testing to improve code reliability and reduce turnaround times
- Use an automated compile, review and test for all branches.
- Use of automated deployment for main branch to enable some testing and to enable bule-green switchover for production releases.
- Use of on demand environment to reduce costs of infrastructure for testing.
- Automated notifications for failures in compilation or testing.
- Deployment of monitoring for on demand and production environments to help with non functional testing (e.g. reporting of resource usage) and troubleshoot application issues (e.g. access to application logs).

Costs and Benefits

- We should be able to reduce our £600K annual test infrastructure substantially by eliminating paying for idle environments.
- These savings are achievable even if we extra for the automated CI/CD pipelines using TeamCity, Artifactory and OpenShift.
- We should be able release major product releases every year instead of every two years with these changes.
- We should also be able to carry out maintenance releases every three months putting us on par with out competitors.
- We can provide bug fixes with in 48 hours by use of automated unit testing and automated smoke testing.
- Above changes will improve customer satisfaction enabling us to retain our maintenance income streams and improve revenue by the reduced time to market for new product releases.