CI/CD

Tejas Parikh (t.parikh@northeastern.edu)

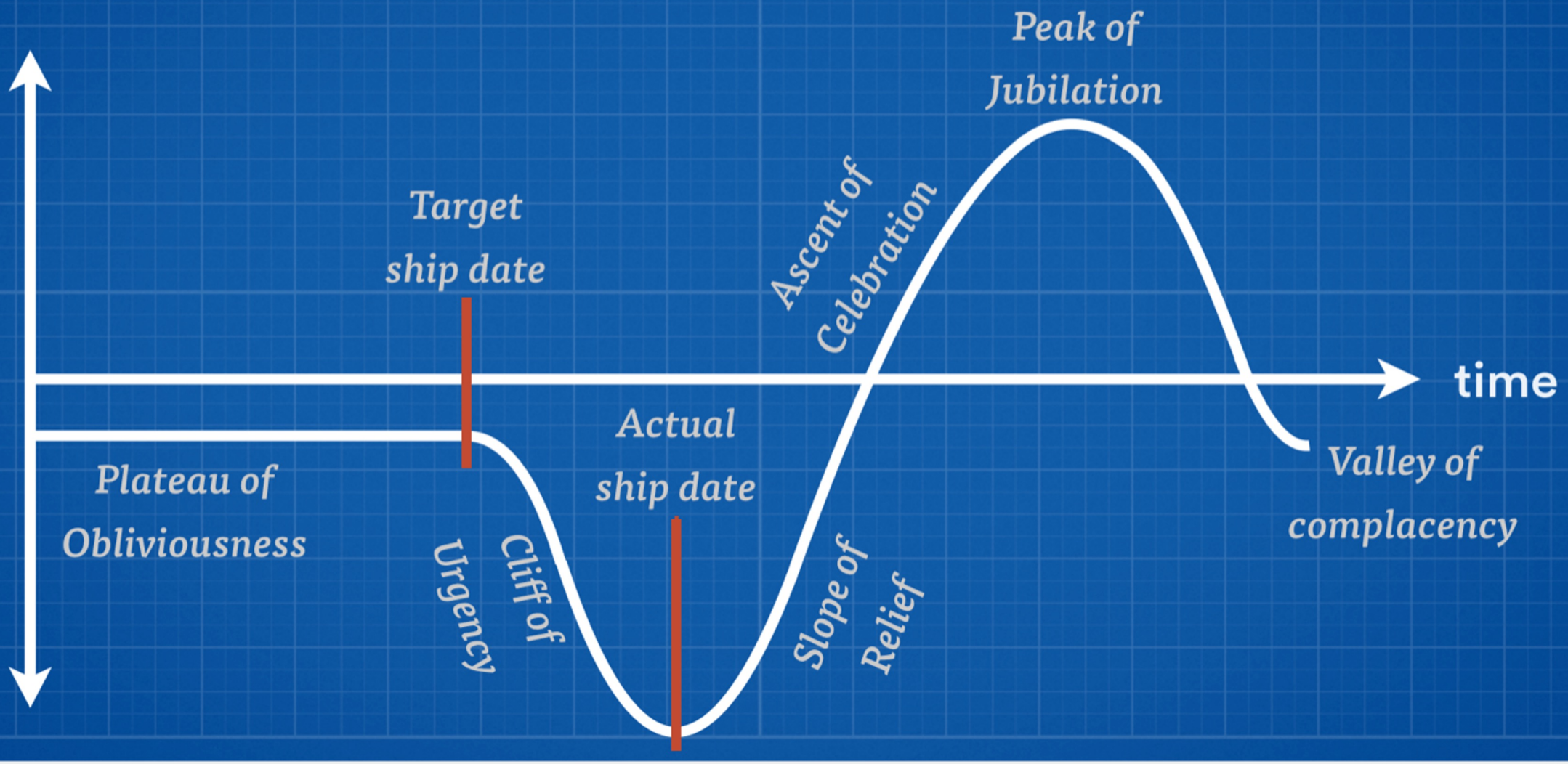
CSYE 6225

Northeastern University

Challenges with Traditional Release Method

- Servers must be set up by IT (sometimes manually)
- Third-party software such as application server, etc. must be installed.
- The software artifacts such as EAR or WAR must be copied to the production host.
- Application configuration must be copied or created.
- Finally any reference data needed must be copied over.
- As you can see there are lot of places where things can go wrong.
- With this process, the day of a software release tends to be a tense one.

Emotional cycle of manual delivery



What is Continuous Deployment?

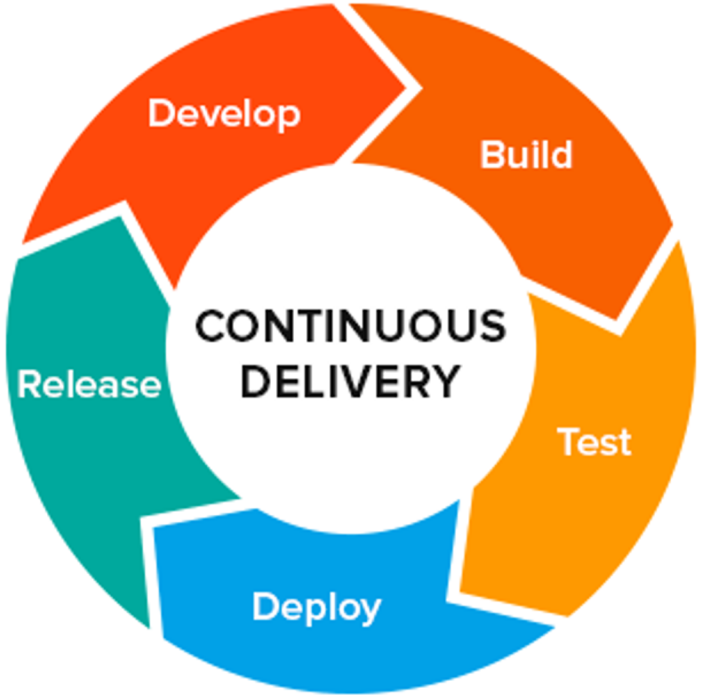
- Continuous Deployment is a software development practice in which every code change goes through the entire pipeline and is put into production, automatically, resulting in many production deployments every day.
- With Continuous Delivery your software is always release-ready, yet the timing of when to push it into production is a business decision, and so the final deployment is a manual step.
- With Continuous Deployment, any updated working version of the application is automatically pushed to production.
- Continuous Deployment mandates Continuous Delivery, but the opposite is not required.

CONTINUOUS DELIVERY

CONTINUOUS DEPLOYMENT

Continuous Delivery

- Continuous delivery is a DevOps software development practice where code changes are automatically built, tested, and prepared for a release to production.
- It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.
- When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized test process.



Continuous Delivery (contd.)

- With continuous delivery, every code change is built, tested, and then pushed to a non-production testing or staging environment.
- There can be multiple, parallel test stages before a production deployment.
- In the last step, the developer approves the update to production when they are ready.

Enabling Continuous Delivery

- Automate
  - The build, deploy, test, and release process must be automated so that it is repeatable.
- Frequent
  - Releases must be frequent.
  - The delta between releases will be small.
  - This significantly reduces the risk associated with releasing and makes it much easier to roll back.

Continuous Deployment

In continuous deployment push to production happens automatically without explicit approval.

Additional Resources

See Lecture Page