

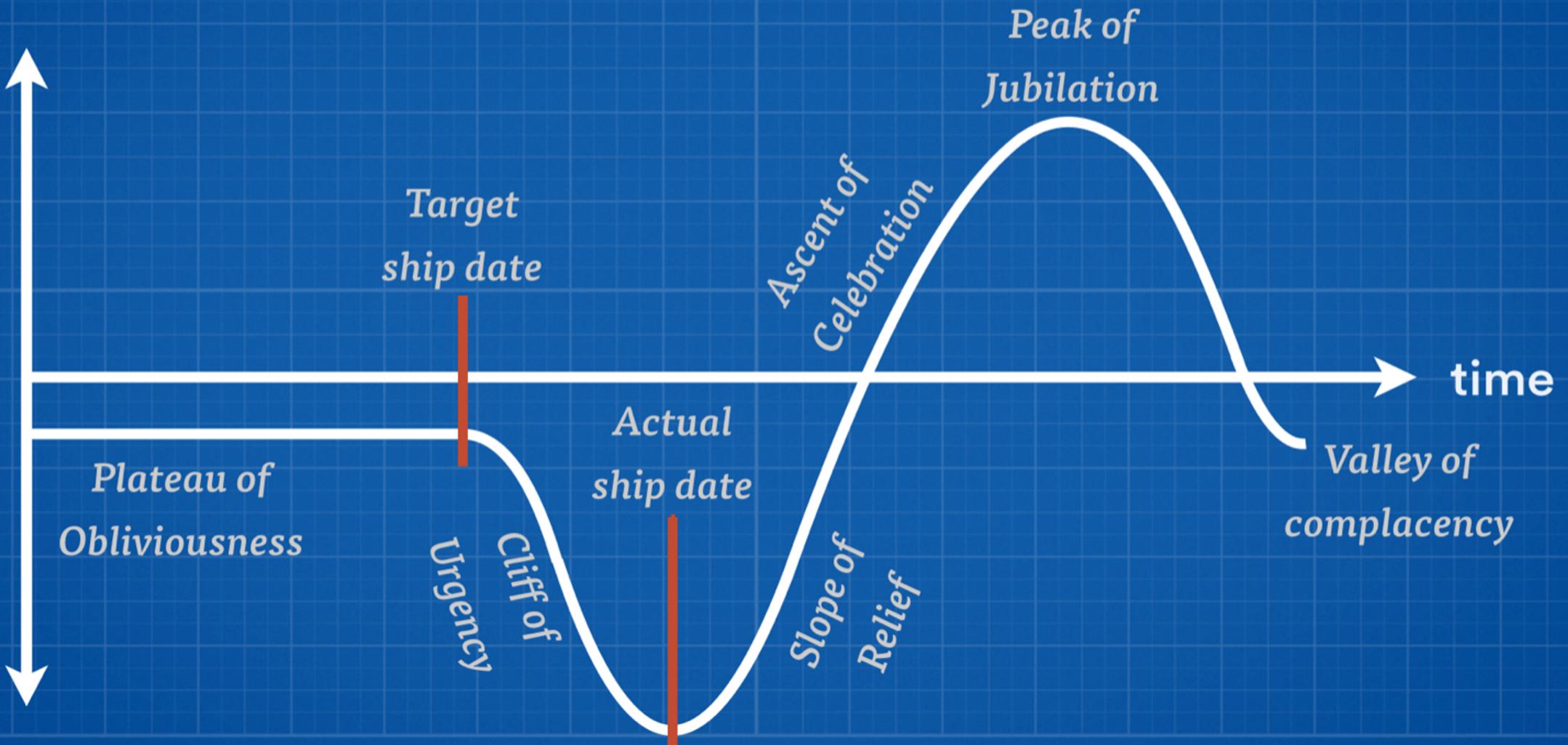
Continuous Integration with CircleCI

Tejas Parikh (t.parikh@northeastern.edu)

Spring 2020, CSYE 6225

Northeastern University

Emotional cycle of manual delivery



CONTINUOUS DELIVERY



CONTINUOUS DEPLOYMENT



CircleCI

CircleCI Projects

- A software repository on GitHub is authorized and added as a project to circleci.com.
- Every code change triggers a build and automated tests in a clean container or VM configured for your requirements.

Steps

- Steps are actions that need to be taken to perform your job.
- Steps are usually a collection of executable commands.
- For example, the checkout step checks out the source code for a job over SSH. Then, the run step executes the make test command using a non-login shell by default.

```
steps:  
  - checkout # Special step to checkout your source code  
  - run: # Run step to execute commands, see  
    # circleci.com/docs/2.0/configuration-reference/#run  
      name: Running tests  
      command: make test # executable command run in  
        # non-login shell with /bin/bash -eo pipefail option  
        # by default.
```

Image

- An image is a packaged system that has the instructions for creating a running container.
- The Primary Container is defined by the first image listed in ***.circleci/config.yml*** file. This is where commands are executed for jobs using the Docker executor.

Jobs

- Jobs are a collection of steps and each job must declare an executor that is either *docker*, *machine*, or *macos*.
- Machine includes a default image if not specified.
- For Docker and macOS you must also declare an image.

Workflows

- Workflows define a list of jobs and their run order.
- It is possible to run jobs in parallel, sequentially, on a schedule, or with a manual gate using an approval job.

Continuous Integration

Continuous Integration – Web Application

- Commit code changes to GitHub repository.
- CircleCI triggers a new build on commit notification.
- CircleCI will run the build steps from CircleCI config file *.circleci/config.yml* from your repository. Build steps does the following:
 - Checkout code from GitHub.
 - Install dependencies & build code
 - Run unit tests

GitHub PR Checks

GitHub Checks

See <https://circleci.com/docs/2.0/enable-checks/>

Additional Resources

<https://spring2020.csye6225.cloud/>