How to exploit CI as a means of deployment?

...

Breakout session
26-27th Feb 2018
58th CREST Open Workshop

About me

- Software Engineer
- Interests: code quality, testing, performance, AI/ML, NN, etc...
- Strengthening teams and helping them go faster
- Data processing and source code analysis at Prodo.AI

Mani Sarkar
@theNeomatrix369

Thank you

- Mark Harman
- Team behind CoW
- UCL
- Facebook and other sponsors
- Guests and attendees
- Prodo.AI
- Anyone else not name...

Agenda

No agenda really!

Discussions in chronological order

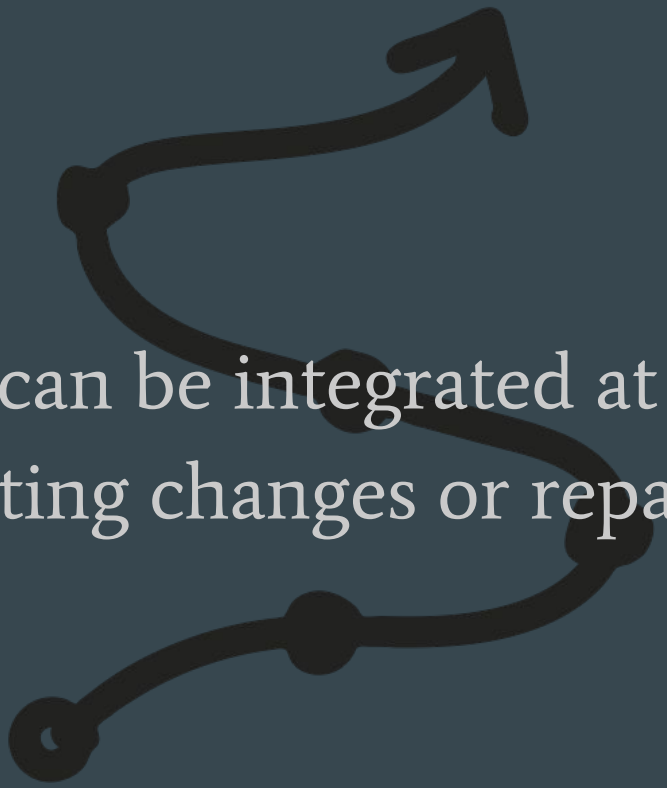
Why CI/CD?

Because....

Question?

About the locality of improvement?
Where does GI sit in the CI/CD pipeline?

Answer!!!

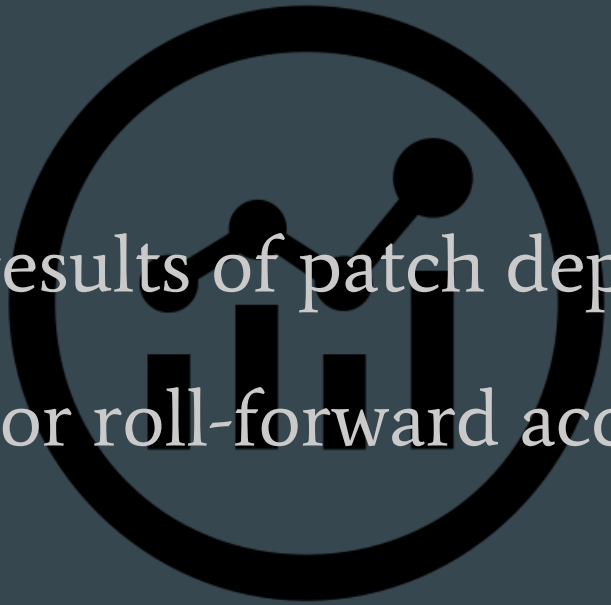


CI/CD pipeline can be integrated at various points (suggesting changes or repairing)

Answer!!!

- Local dev environment: IDE, git hooks
- SCM integration
- Compile & build step
- Test execution step
- Deployment step

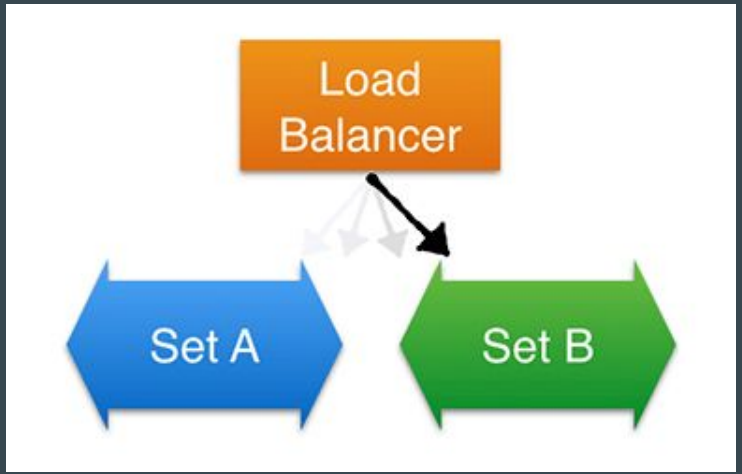
Deploy patch and analyse



Analyse results of patch deployment!

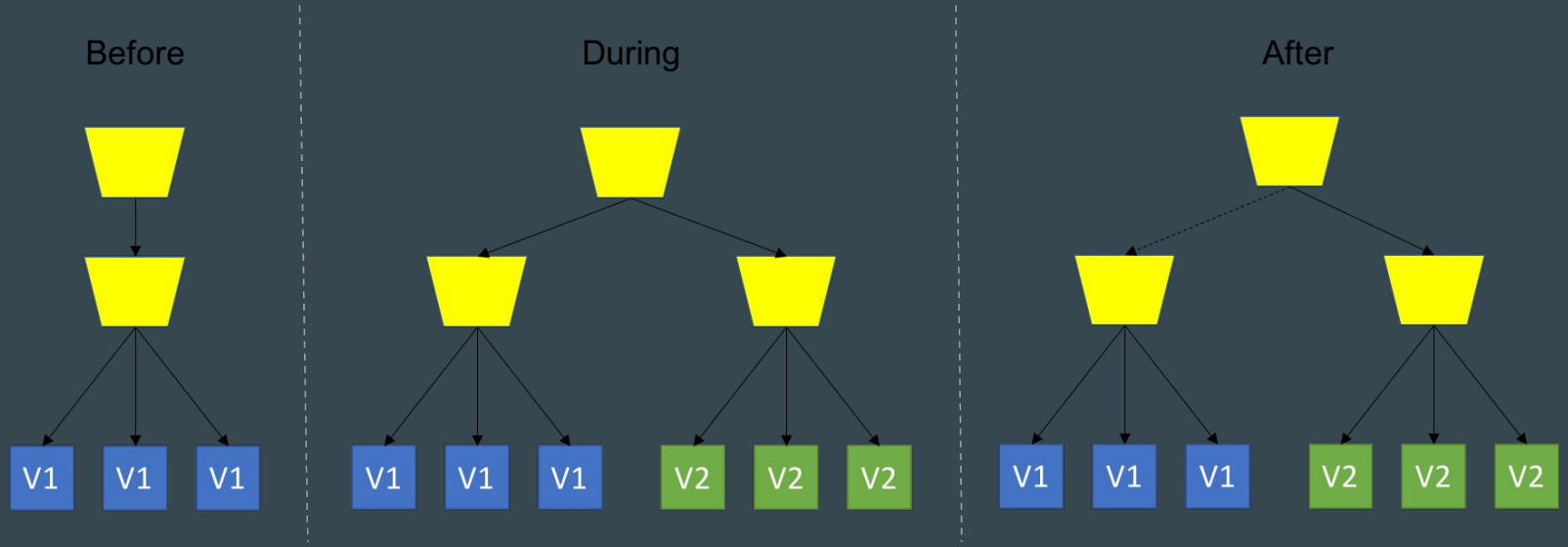
Rollback or roll-forward accordingly!

Blue/green deployment



Seamlessly apply patch & switch, without users noticing

Canary deployment

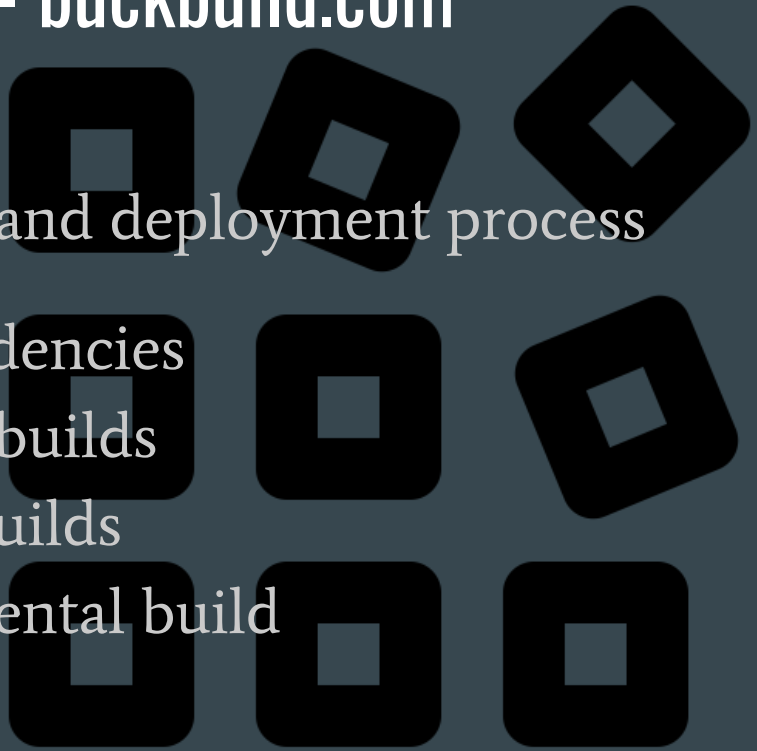


Gradually apply patch without users realising

Post patch deployment analysis

Study the changes and its impact after patch is applied, and feedback to the System

Facebook's Buck - buckbuild.com



optimising build and deployment process

- caching dependencies
- speed up your builds
- reproducible builds
- correct incremental build

Solution similar to snyk.io

- scan / investigate repo(s)
- detect vulnerabilities
- produces daily/weekly reports
- alerts on new / urgent vulnerabilities
- eventually raise PR against the repo(s)
    - contains changes version of one or more affected libraries

Using ML/AI to improve CI/CD process

- Using ML to learn and fix the build process:
  - reads build logs to understand the issue(s) to hand
  - https://harness.io/2017/11/can-apply-machine-learning-continuous-delivery/
- Gathering feedback from CI/CD and feeding it back into the system
  - https://www.youtube.com/watch?v=iGQpe5FxjOQ

Usage history: benefits

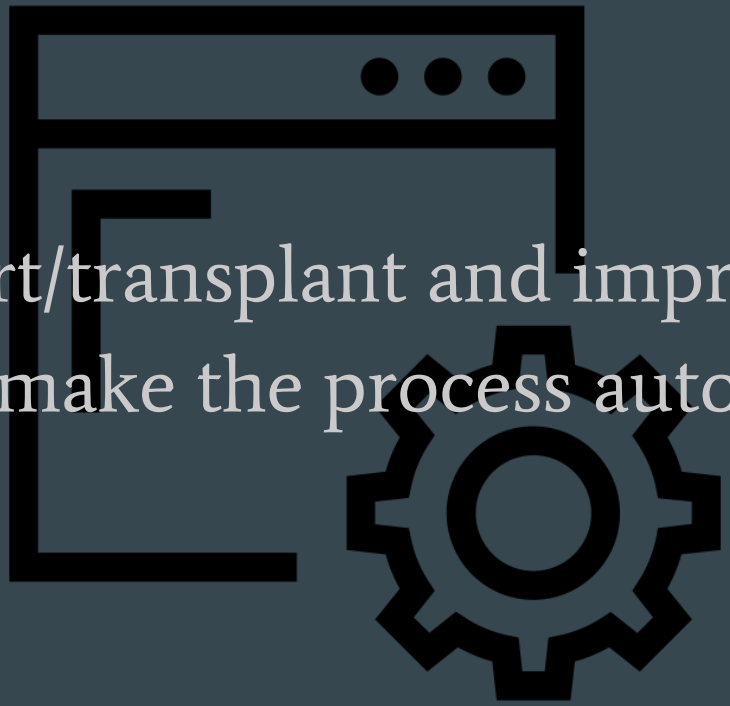
Learning from code history and CI usage history from multiple sources, how do we gather such proprietary data?

Transport/transplant vulnerability patches

Publish patches to implement and transport/transplant them to F/OSS projects lacking them:

- CVE Id
- reference to the buggy code
- patch to apply to remedy the vulnerabilities

Research question?



Patch transport/transplant and improvement: how do we make the process automatic?

Research question?

GI: good for first level or last mile improvement?

Research question?

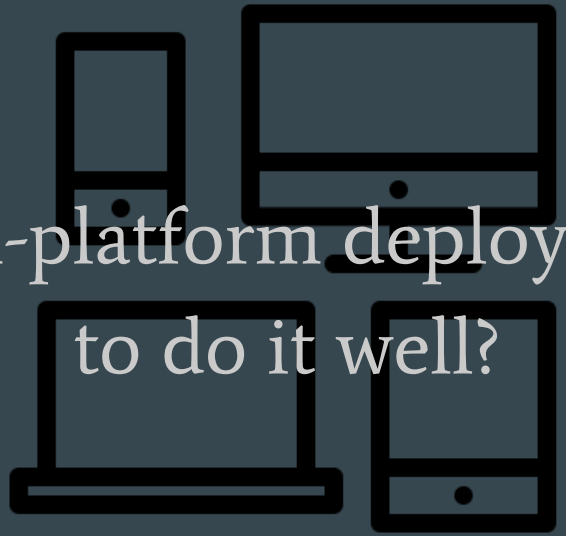
Apply GI on itself: automating its own repair/healing process

System learning from its environment and feeding back to itself (remembering / memory)

Research question?

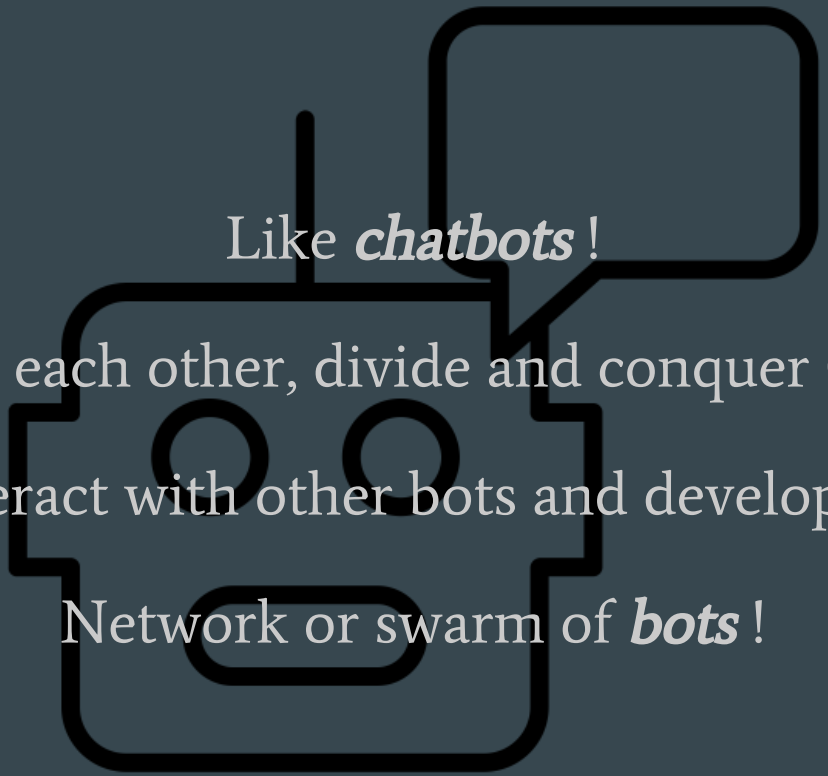
How to fix flaky tests with noisy test results?

Research question?



How to do multi-platform deployments? And how to do it well?

GI Bots



Like ***chatbots*** !

GI Bots help each other, divide and conquer CI/CD tasks!

Interact with other bots and developers

Network or swarm of ***bots*** !

Closure: Java & JS optimisation project at Google

https://developers.google.com/closure/

took over the task of compiling and optimising submitted code

applying best practices and optimisation to the code by GI

project abandoned after sometime

Citations

All images used in this presentation are owned by the respective authors, and most of them come from the https://thenounproject.com

Thank you

For your time and attention! We hope you have enjoyed it and found it useful!