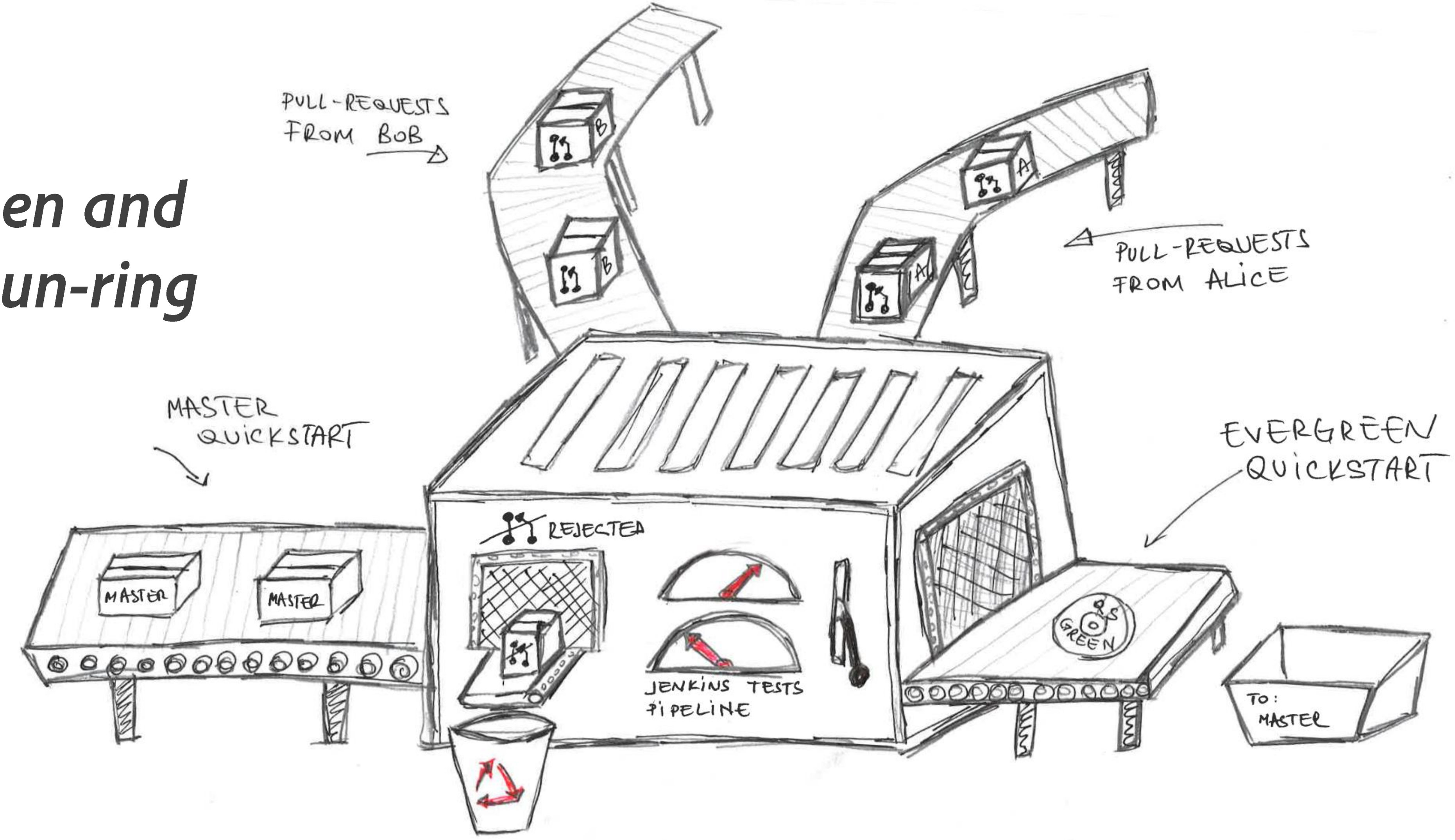
Project evergreen and how to un-ring a bell



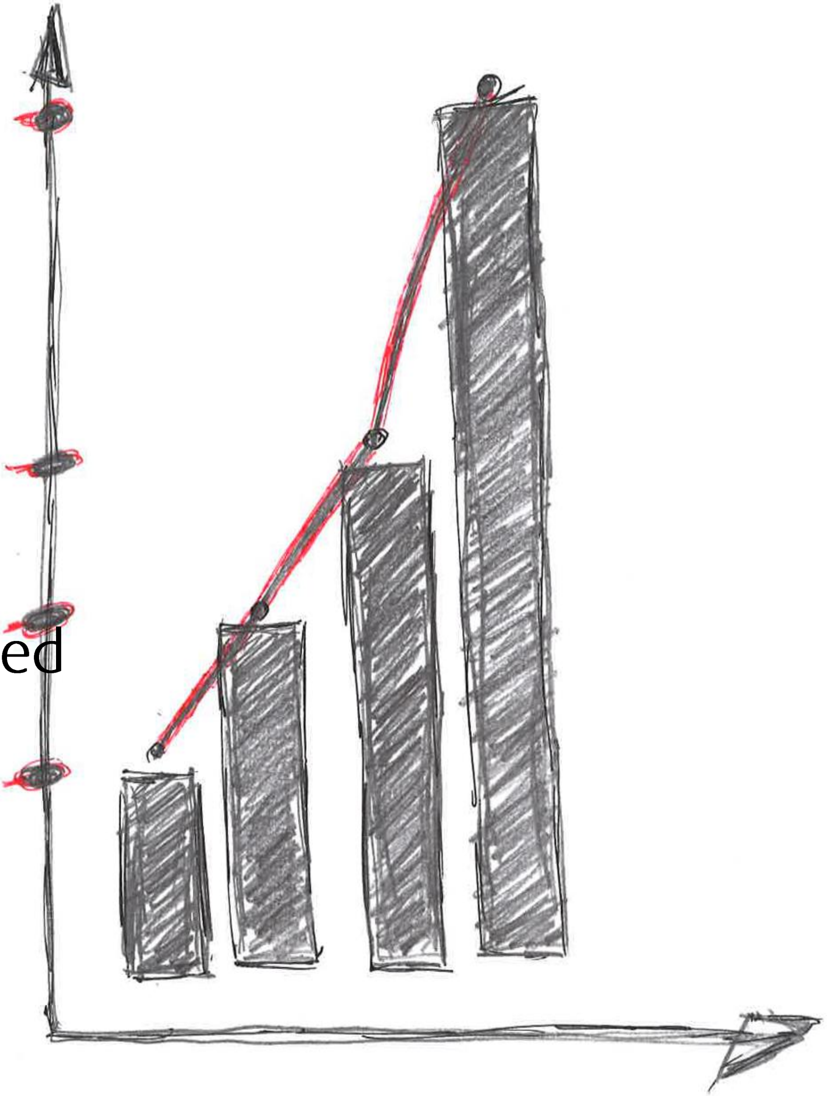
The Background

The AEM team has grown

- ***>500 AEM engineers*** – all pushing to CQ/quickstart
- More specialized teams
- Teams have different ways of working
- Pushing to the same place without validation potentially ***blocks*** everybody

We have a lot of functional tests and they're almost always red

- ***>2000 functional tests*** (ITs) + UI tests, great testing infra
- Failing IT tests are ***load blockers*** and those jobs are almost always RED
- We test after pushing a commit – ***reactive vs proactive***



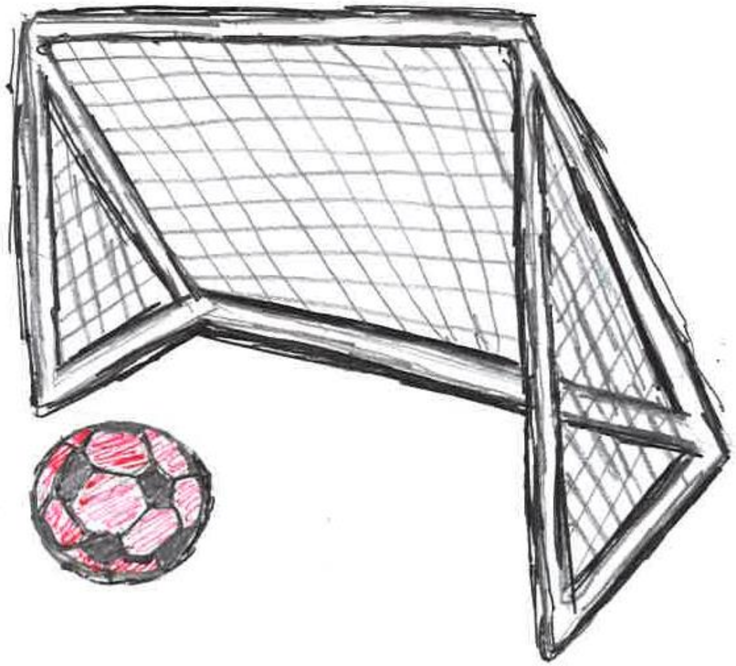
The Goal

Need to make the ***quickstart green*** from an IT point of view

- No more functional ***regressions*** – no *IT_failing* load blockers
- Less or no delay for ***load releases***
- Immediate ***accountability***

Desired solution

- ***Prevent*** a breaking change to CQ/quickstart (I lied, you can't really un-ring a bell)
- Minimal or no added ***complexity*** for development
- Minimal or no added ***delay*** for resolving issues



The Evergreen solution

Simple to use and maintain

- Technologies we already use – ***github*** and ***jenkins***
- Github ***pull-requests*** automatically run the ***evergreen pipeline*** and post back the status
- Automatic code ***merge*** or PR ***rejection***
- E-mails and notifications are sent; mention to the @creator of the pull-request
- Easy to fix issues (push more to the branch) and to ***retest*** by asking ***greenbot***
- Green-who? - a ***github user*** who triggers the pipeline automatically or if asked nicely

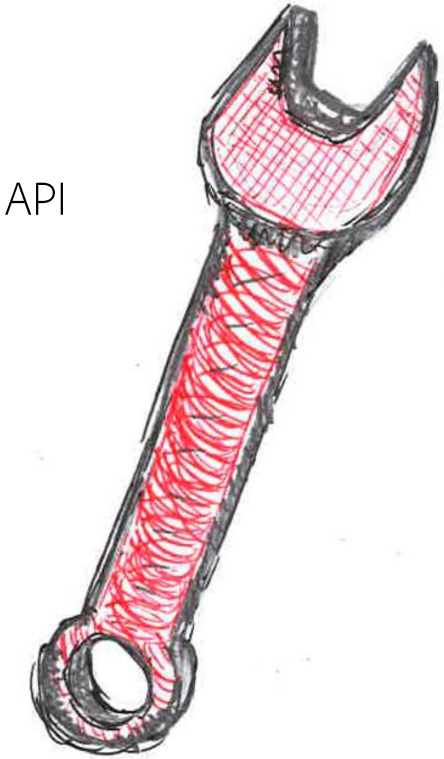
Development workflow

- Create a pull-request and wait ***~45 minutes*** for the results
- If nothing is wrong - code is ***automatically merged***
- If there are tests failing,
  - the problem is fixed, code is pushed to the PR branch
  - pipeline is automatically triggered and code is merged

Let's get technical

The evergreen pipeline jenkins job

- Uses the GHPRB plugin (https://github.com/jenkinsci/ghprb-plugin) - uses the github HTTP API
- Take PR, ***merge*** with current master locally
- Build a special AEM quickstart with a ***classifier*** and deploy to our QE nexus
- Trigger a multi-job IT job with ***1600+ very stable tests***
- If successful, ***merge*** the PR **in master**, comment and close the PR
- If failed, ***don't merge*** post status and link to results

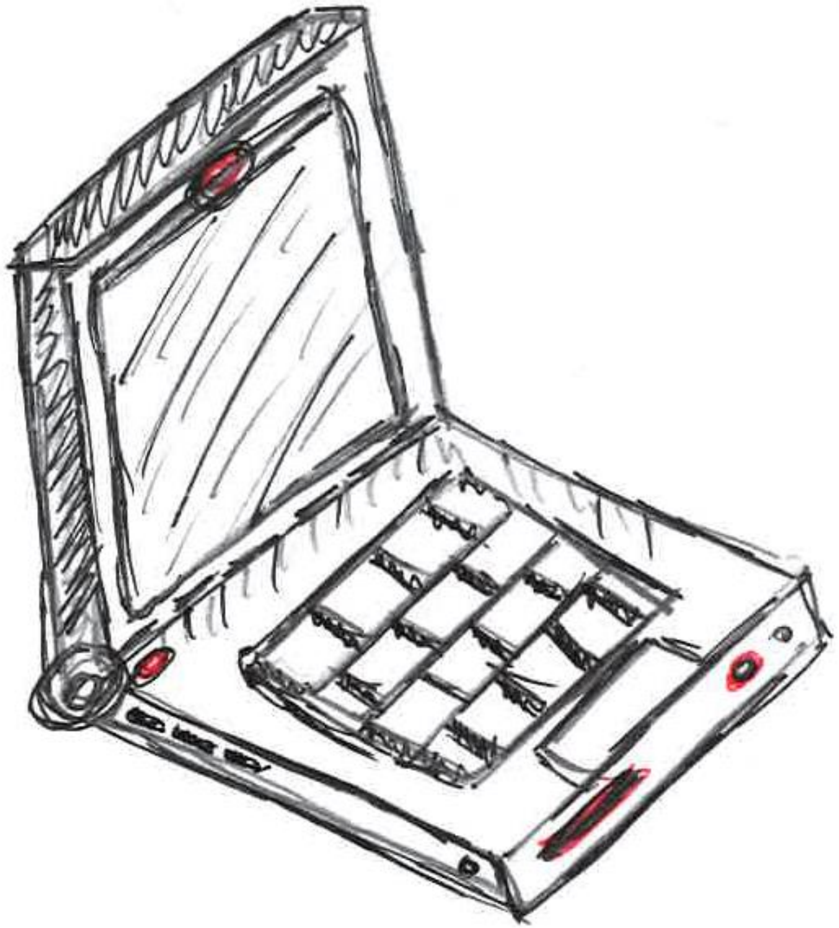


The greenbot

- NOT an external service, just a github user (YAGNI) – for now
- Used from the jenkins job and GHPRB to configure hooks, merge code and make PR comment
- Extensions for GHPRB to post customized comments

Demo time

- Trigger a PR in github
- Show jenkins job
- Questions



Adobe