



Realworld CD Pipelines

It's not always a linear stream

Nigel Harniman

About me



@harniman

Nigel Harniman

Build Engineer,
Devops and
Architect



CloudBees

DevOps, Infra as Code,
Continuous Delivery



About CloudBees

WE ARE

The Enterprise
Jenkins Company



OUR GOAL

To be the automation leader helping organizations develop and deliver better software faster.

WE OFFER

Continuous delivery solutions powered by Jenkins at enterprise scale with enterprise support.

OUR BUSINESS VALUE

We enable IT to deliver better software faster by optimizing their use of Jenkins on premise or in the cloud.

About you



Jenkins? CI? CD? Code Jenkins plugin?

About you

Developer?

About you

Operations?

About you

Tester?

About you

Build Engineer?

About you

Project Manager?

About you

Doing CI?

About you

Automated Deployment?

About you

Wants to automate
deployment?

About you

Who has an E2E Pipeline?

About you

Single component?

About you

Complex suite of components?

About you

Whose is perfect?

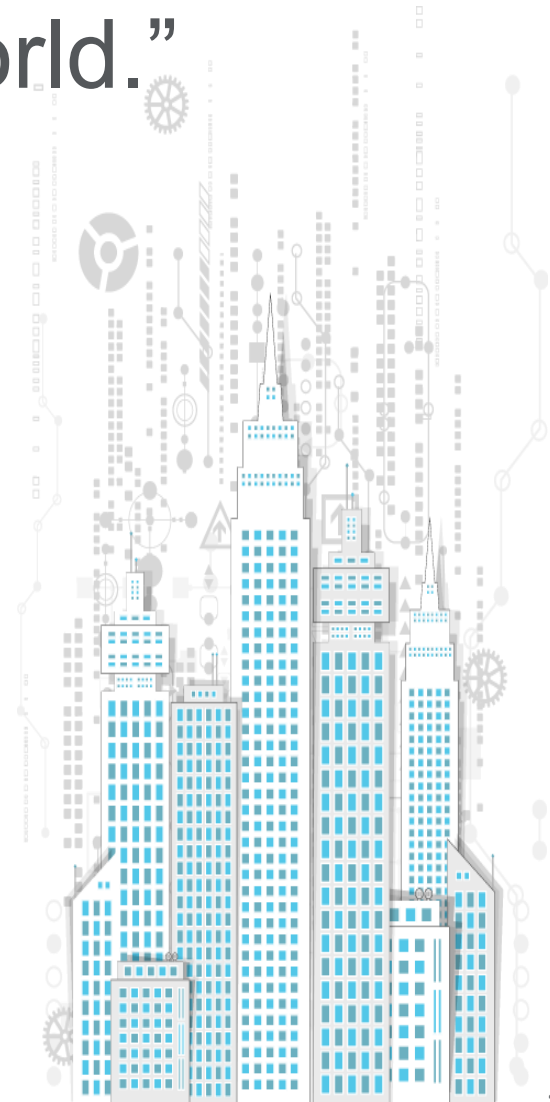
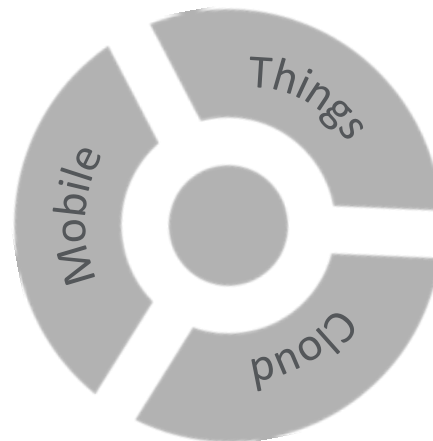
About you

Who uses Jenkins?

“Software is eating the world.”

“Every business is in the software business.”

“It’s an application economy.”



Software is Everywhere





GIZMODO

A Software Upgrade Grounded Hundreds of Flights This Weekend



Adam Clark Estes

Filed to: AIRPLANES 8/17/15 10:42am

13,375 🔥 1 ☆ ▾



Were you flying to or from the eastern seaboard over the weekend? I'm so sorry. A botched software update [caused hundreds of flights to be delayed or cancelled](#) on Saturday and Sunday, so there's a decent chance you had a bad trip. What happened? A software upgrade, of course.

This wasn't a one off....

\$57 Million Slot Machine Win Blamed on Software Error, Casino Won't Pay

ANDY CHALK | 13 DECEMBER 2011 10:18 PM

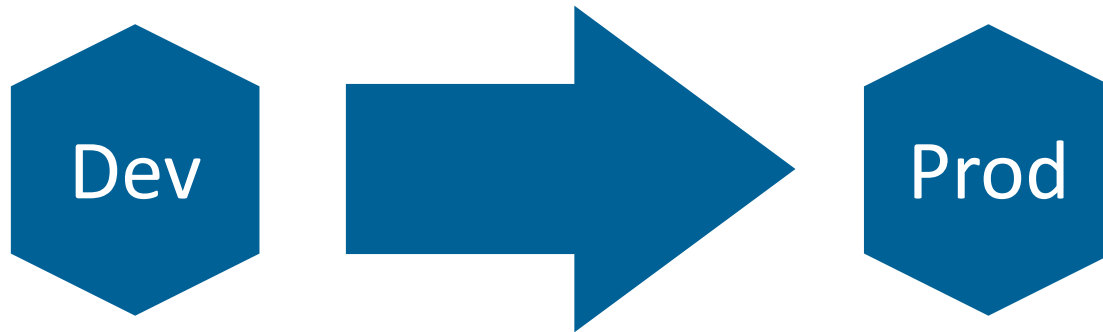
64

A man who won \$57 million from a slot machine is taking the casino to court after it told him it was a software error and refused to pay up.

It was a very good day for Behar Merlaku of Switzerland when a slot machine at an Austrian casino went bells-and-whistles bananas, telling him he'd won the grand prize of \$57 million. But it quickly turned into a very bad day when he attempted to claim his prize and was told by the casino owners that a software error had triggered the machine and that they weren't going to pay up. They offered him \$100 and a free meal instead, and he, quite naturally, told them to get bent.



How Do You Deliver Better Software Faster?



We need

a process that ensures quality
executes reliably and quickly
delivers value early

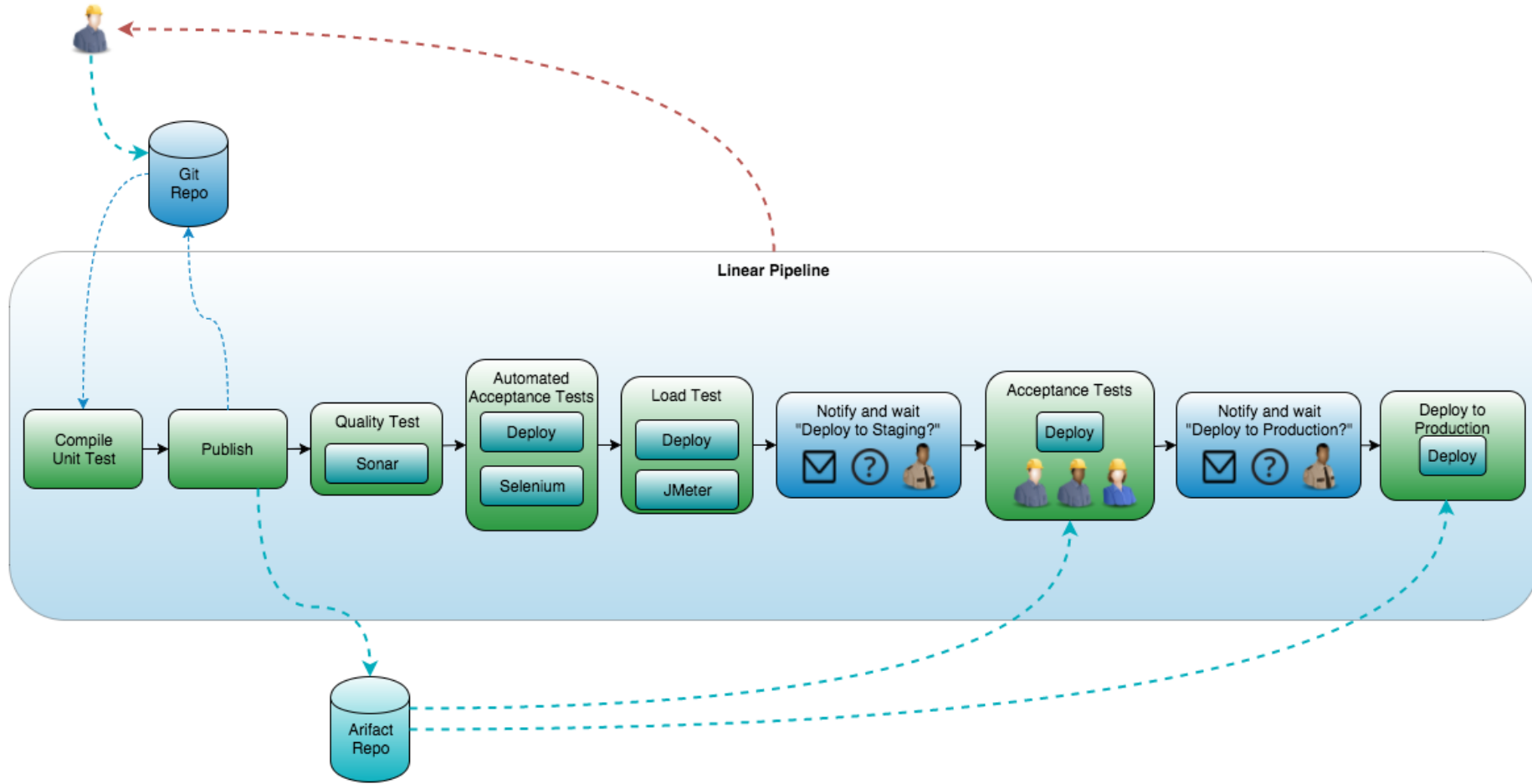
We need

to automate

We need

an automated pipeline

Typical Linear Pipeline



Jenkins Pipeline Job

Pipeline Job Type

Item name

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline

Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

```
node ("linux") {
  ensureMaven()
  stage 'Checkout'
  git url: 'https://github.com/harniman/mobile-deposit-ui'

  stage 'Compile and Unit Test'
  sh "mvn clean package -DBUILD_NUMBER=${env.BUILD_NUMBER}"

  stage 'Publish'
  sh 'mvn install'

  stage 'Quality Test'
  sh 'mvn sonar:sonar'

  stage 'Func Test'
  deploy('func-test')
  runSelenium('func-test')

  stage 'Load Test'
  deploy('perf-test')
  runJmeter('perf-test')

  stage 'Acceptance Test'
  input id: 'Deploy-stage', message: 'Deploy to staging?',
        ok: 'Deploy', submitter: 'nharniman'
  deploy('acc-test')

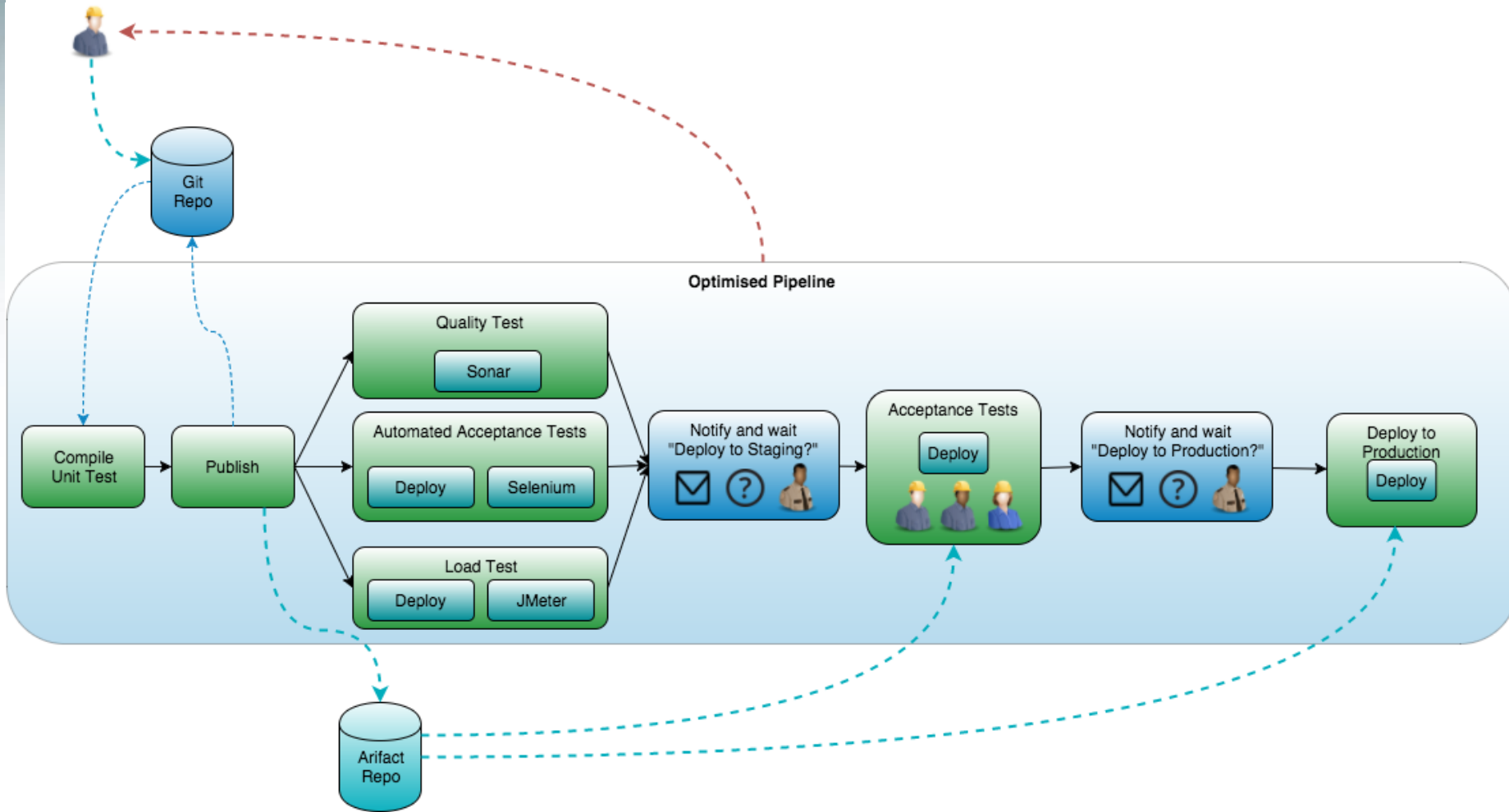
  stage 'Prod'
  input id: 'Deploy-prod', message: 'Deploy to production?',
        ok: 'Deploy', submitter: 'nharniman'
  deploy('prod')
}
```

Showing progress and history

			Checkout	Compile and Unit Test	Publish	Quality Test	Func Test	Load Test	Acceptance Test	Prod
Average stage times: (Average full run time: ~28s)			1s	10s	13ms	18s	6ms	47ms	49ms	78ms
#5	Mar 03 10:13	No Changes	1s <small>linux-slave-3</small>	10s <small>linux-slave-3</small>	6ms <small>linux-slave-3</small>	17s <small>linux-slave-3</small>	6ms <small>linux-slave-3</small>	6ms <small>(paused for 22s)</small> <small>linux-slave-3</small>	1ms <small>linux-slave-2</small>	78ms <small>(paused for 3s)</small> <small>linux-slave-2</small>
#4	Mar 03 10:10	No Changes	1s <small>linux-slave-3</small>	11s <small>linux-slave-3</small>	20ms <small>linux-slave-3</small>	14s <small>linux-slave-3</small>	8ms <small>linux-slave-3</small>	22ms <small>(paused for 1min 49s)</small> <small>linux-slave-3</small>	98ms <small>(paused for 3s)</small> <small>linux-slave-2</small>	
#3	Mar 03 10:04	No Changes	2s <small>linux-slave-3</small>	12s <small>linux-slave-3</small>	17ms <small>linux-slave-3</small>	17s <small>linux-slave-3</small>	6ms <small>linux-slave-3</small>	147ms <small>(paused for 5min 54s)</small> <small>linux-slave-3</small> aborted		
#2	Mar 03 10:02	No Changes	1s <small>linux-slave-3</small>	6s <small>linux-slave-3</small>	11ms <small>linux-slave-3</small>	23s <small>linux-slave-3</small>	6ms <small>linux-slave-3</small>	15ms <small>linux-slave-3</small> failed		

We can optimise this

Optimised Pipeline



```
stage 'Publish'  
sh 'mvn install'
```

```
sh "tar -c -f src.tar src/ pom.xml"  
stash includes: 'target/mobile-deposit-api*.jar, src.tar', name:
```

```
}
```

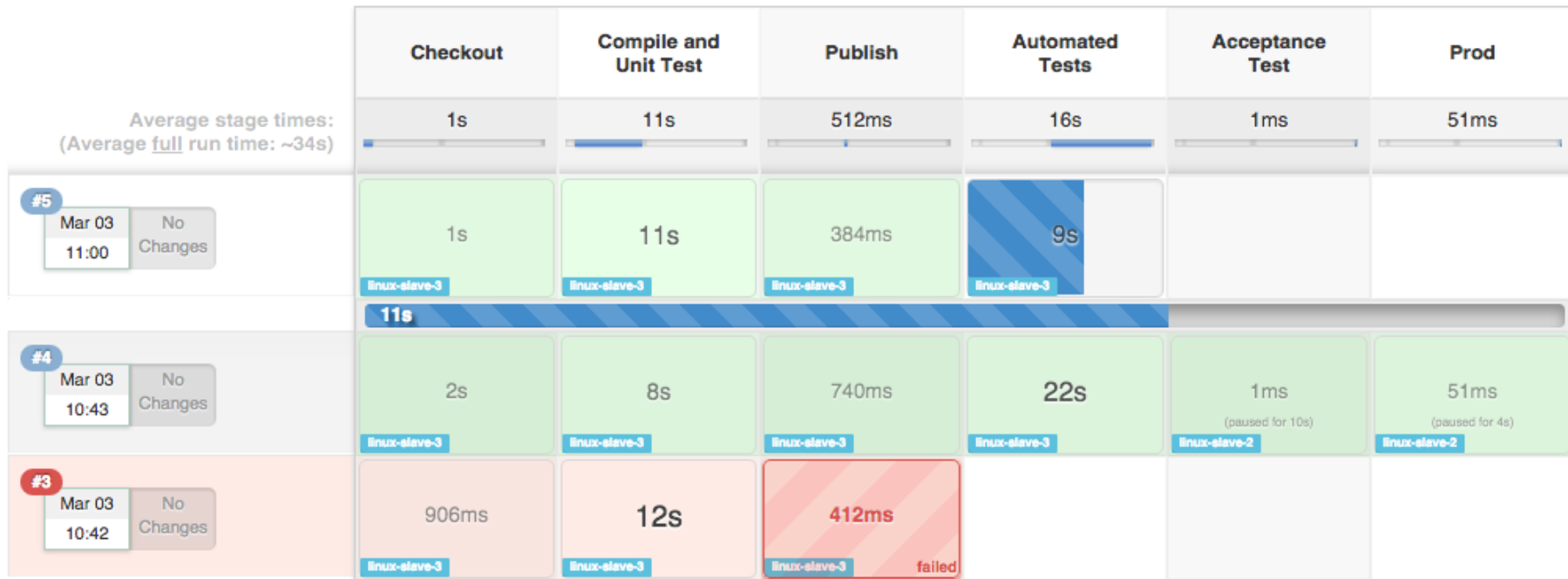
```
stage "Automated Tests"
```

```
parallel(Sonar: {  
  node ('linux'){  
    ensureMaven()  
    unstash 'build-artifacts'  
    sh 'tar -x -f src.tar'  
    sh 'mvn sonar:sonar'  
  }  
}, Functional: {  
  node ('linux'){  
    ensureMaven()  
    unstash 'build-artifacts'  
    sh 'tar -x -f src.tar'  
    deploy('func-test')  
    runSelenium('func-test')  
  }  
}, Performance: {  
  node ('linux'){  
    ensureMaven()  
    unstash 'build-artifacts'  
    sh 'tar -x -f src.tar'  
    deploy('perf-test')  
    runJmeter('perf-test')  
  }  
}  
)
```

```
}
```

```
)
```

Pipeline Stage View



We're done ?

commit change

trigger the process

verify quality

and release!!

But wait....

single component

or microservice

Only true if

tests are thorough
independently releasable
no breaking changes for
consumers

a system is comprised of
multiple apps /components

have dependencies

new feature
provided by new/updated
interface
require min version

system level concerns:
performance test
functional test

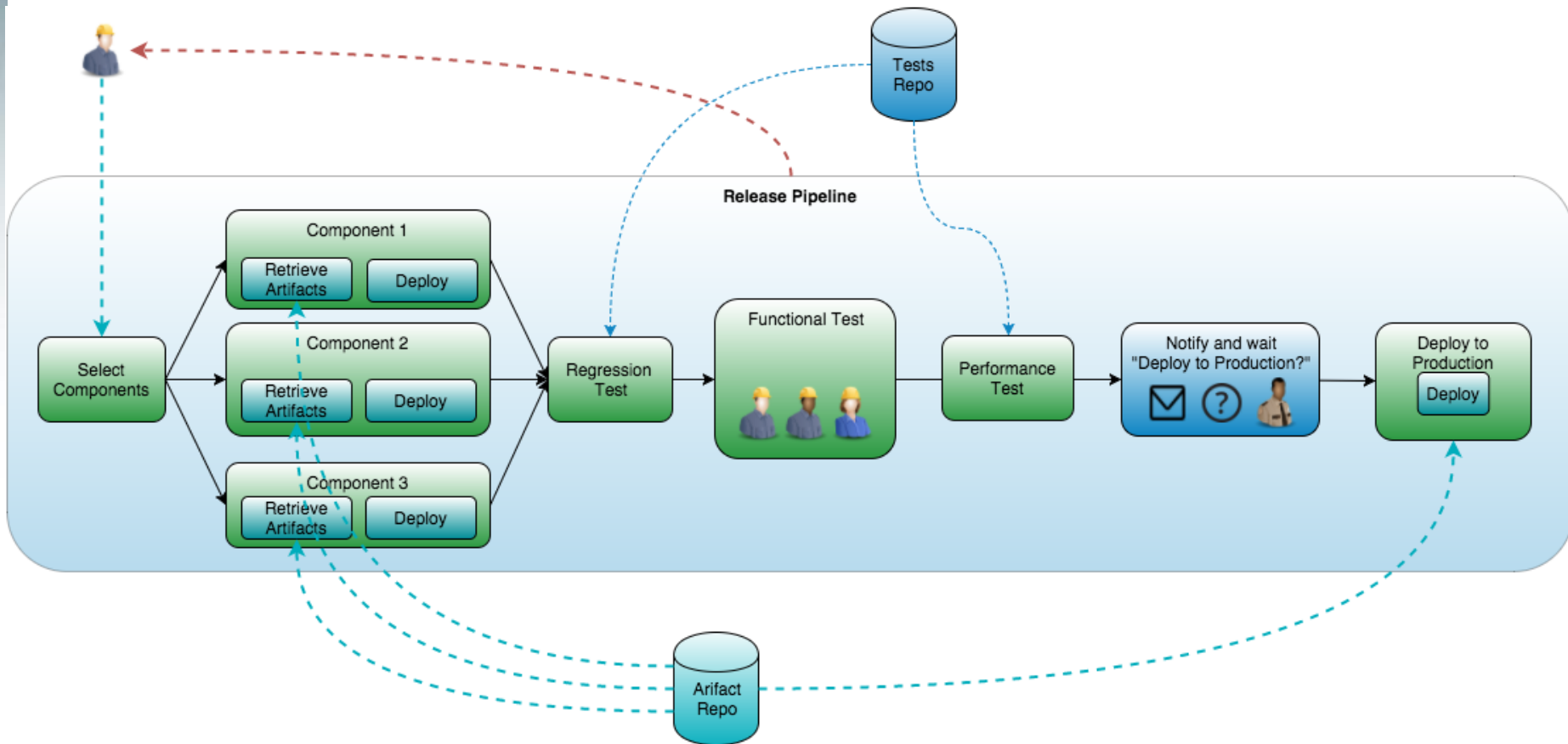
embedded systems : firmware image

???

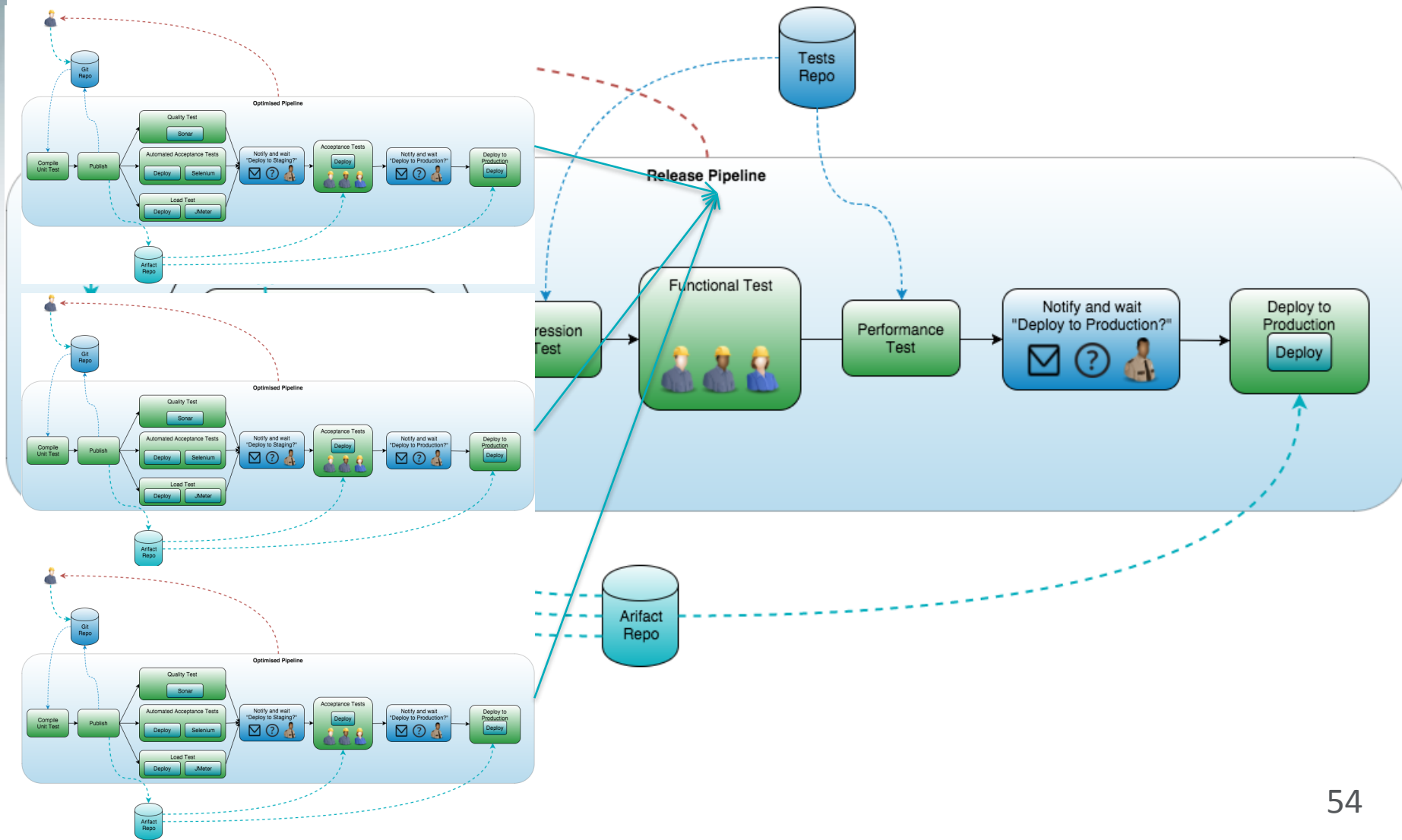
assembly build

release pipeline

Release Pipeline



Release Pipeline



Pipeline release-build

This build requires parameters:

component1

component2

component3

Build

Pipeline release-build

This build requires parameters:

component1

component2

component3

component4

component5

Build

another manual step?

which versions?

wrong build = regression

automated trigger from
upstream?

use **build** step

```
build job: 'release-build', parameters: [  
  [$class: 'StringParameterValue', name:  
  'component1', value: comp1ver],  
  [$class: 'StringParameterValue', name:  
  'component2', value: comp2ver],  
  [$class: 'StringParameterValue', name:  
  'component3', value: comp3ver]  
], wait: false
```

need build numbers for each
component

`null = lastSuccessfulBuild?`

inspect deployed system?



tex

tex

tex

tex

CMAI

ECON

ECON

ECON

ECON

TAL

TEL

CAPITAL

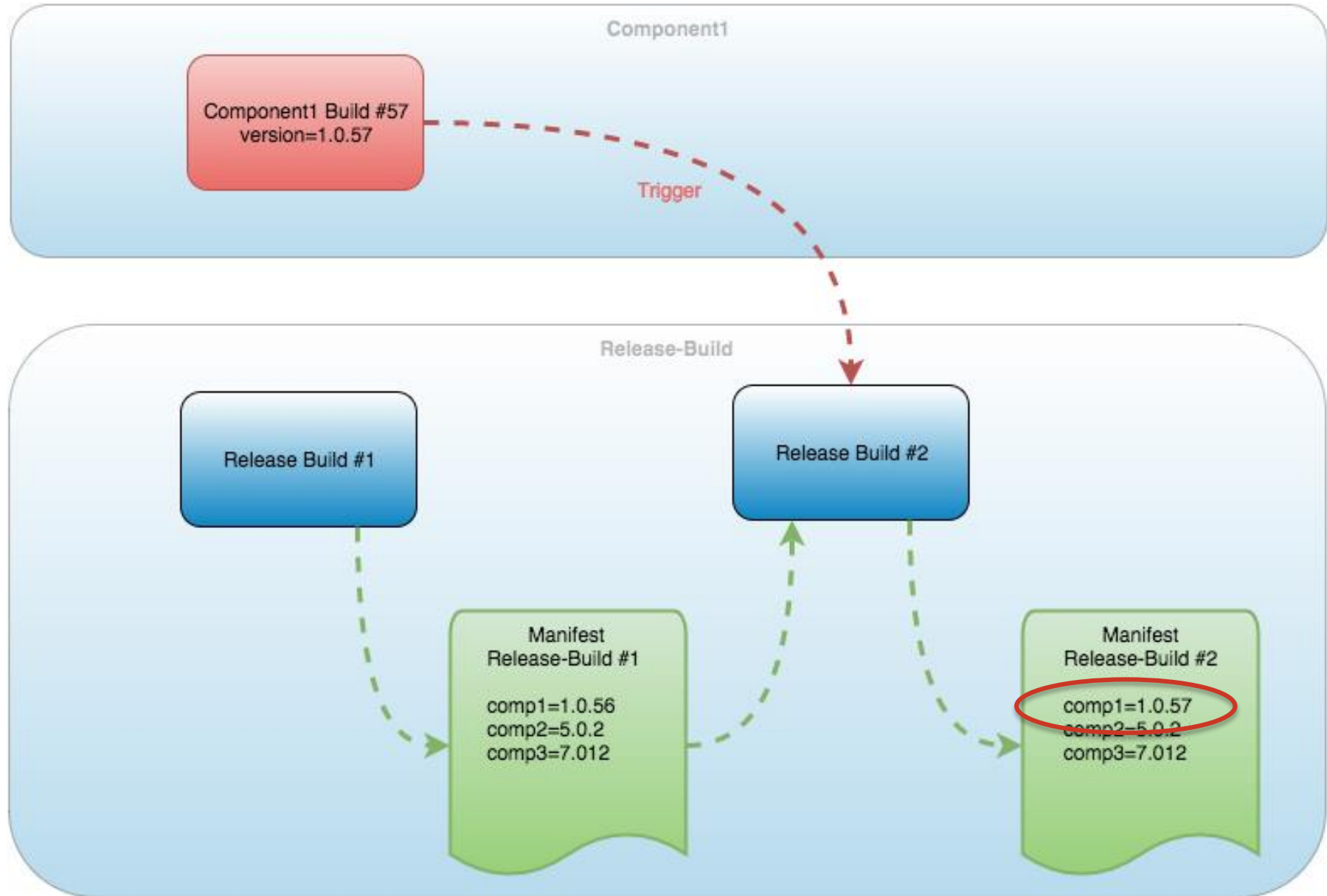
tex

Glyn Lowe.com

Manifest

n. cargo-list for use of Customs officers; list of passengers in aircraft or of trucks etc. in goods train

Oxford English Dictionary



what's in the manifest?
list of components and versions

```
#Tue Mar 01 23:11:41 GMT 2016  
mobile-deposit-api=1.0.0-21  
mobile-deposit-ui=1.0.0-4
```

read/write manifest

```
def writePropertiesFile(props, file) {  
    log "WriteProperties", "File = $file"  
    writeFile file: file, text: writeProperties(props)  
}
```

```
@NonCPS def writeProperties (props) {  
    def sw = new StringWriter()  
    props.store(sw, null)  
    return sw.toString()  
}
```

where to store the manifest?

jenkins build workspace
scm
Jenkins build artifact
external file system
artifact repository



Last Successful Artifacts



manifest

129 B



view

```
stage name: 'Writing Manifest'
writePropertiesFile(versions, "manifest")
archive 'manifest'
```

are we done?

- Execute concurrent builds if necessary

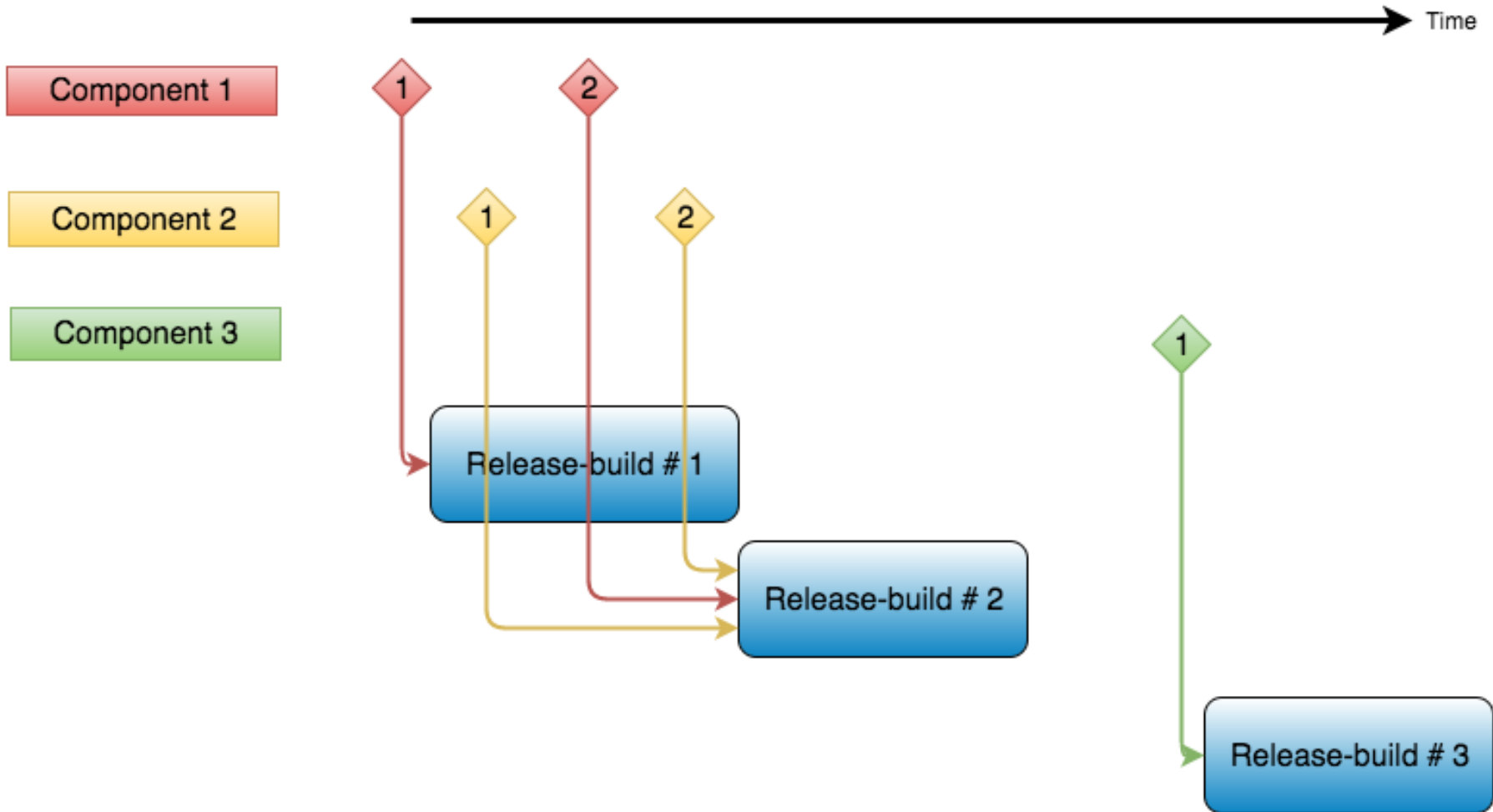
cost of environments
parallel or single build?
sequencing

queuing theory

number of upstreams
upstream build freq
release-build trigger rate
release-build duration
wait time

batching

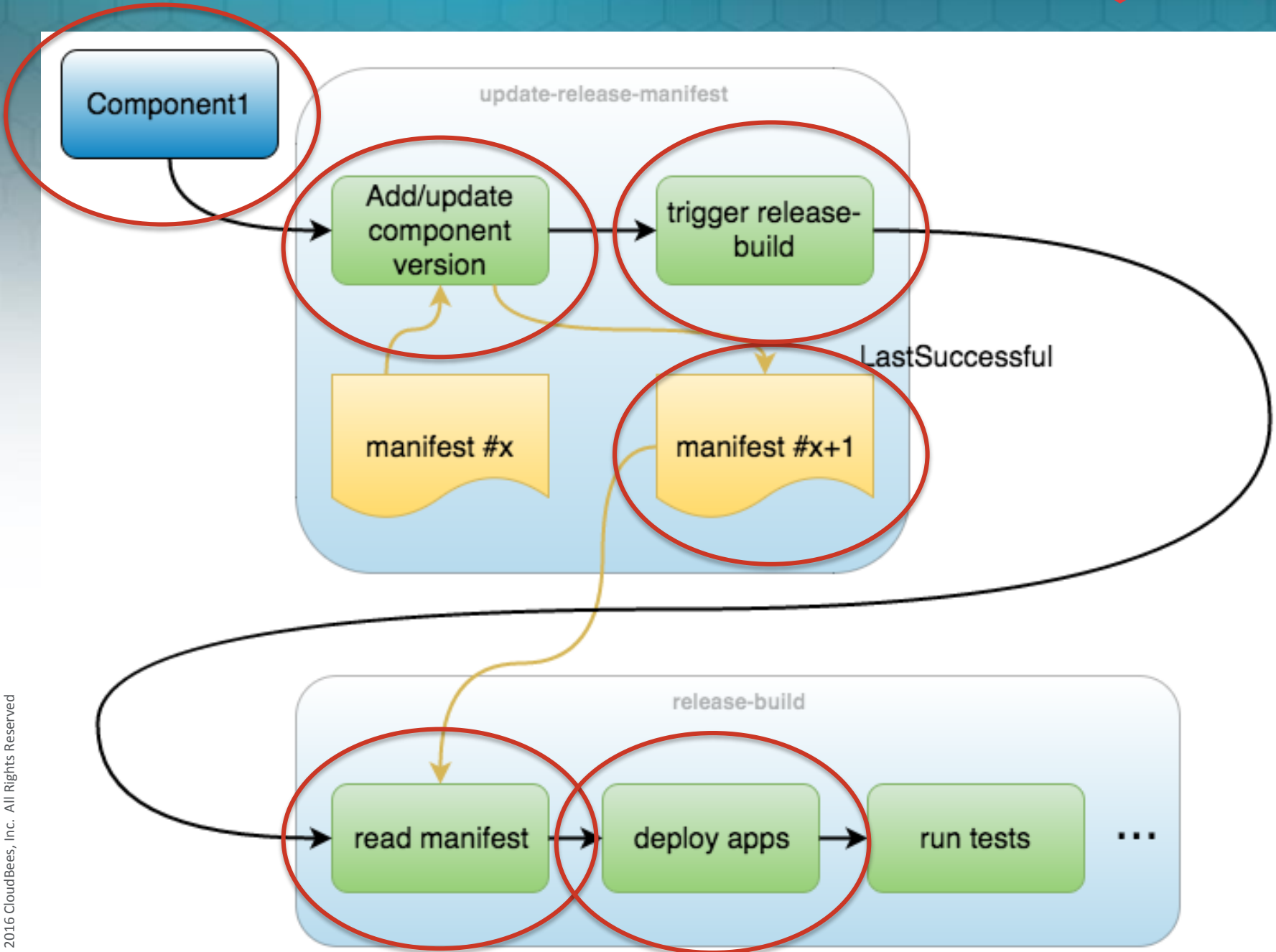
Batching



jenkins queues builds
pipeline stage concurrency
concurrency rule skips builds

works with scm
using Jenkins artifacts needs
additional step

manifest update async to
release-build



manifest-update

```
node("linux") {
  stage name: 'Reading Manifest'
  try {
    step([$class: 'CopyArtifact', filter: 'manifest',
          projectName:env.JOB_NAME,
          selector: [$class: 'StatusBuildSelector',
                    stable: false]])
    versions = readPropertiesFromFile ("manifest")
  } catch (Exception e) {
    echo e.toString()
    versions = new Properties()
  }

  stage name: 'Merging Manifest'
  versions[app]=revision


  stage name: 'Writing Manifest'
  writePropertiesFile(versions, "manifest")
  archive 'manifest'
}
```



release-build

handling the deployment


```
def branches = [:]
for (i=0; i < appsToUpdate.size(); i++) {
  def app=appsToUpdate[i]
  def revision = requiredVersions.getProperty(app)
  branches[app] = {
    decom(app, revision)
    deploy (app, revision)
  }
}
parallel branches
```




re-deploy all
compare with running

GET ▼ localhost:9010/spring/info Params Send ▼  ▼

Authorization Headers (1) Body Pre-request script Tests  

No Auth ▼

Body Cookies Headers (5) Tests (0/0) Status **200 OK** Time **17 ms**

Pretty Raw Preview JSON ▼   

```

1 {
2   "build": {
3     "url": "http://localhost:8080/job/qconn/job/mobile-deposit-api/branch/qcon/3/",
4     "number": "3",
5     "commit": "e49b3e3bf99dbe66d4acb0047af3b42fcf0809fd",
6     "version": "1.0.0-3"
7   }
8 }

```


re-deploy all
compare with running
compare with manifest

```
def compareVersions ( requiredVersions, currentVersions) {  
  
    currentapps = currentVersions.stringPropertyNames().toArray()  
    reqapps = requiredVersions.stringPropertyNames().toArray()  
    Properties updatedVersions = new Properties()  
  
    for (i=0; i < reqapps.size(); i++) {  
  
        def app=reqapps[i]  
  
        if (currentVersions.getProperty(app) == requiredVersions.getProperty(app) ) {  
            log "Calculating Deltas", "Correct version of $app already deployed"  
        } else {  
            log "Calculating Deltas", "Adding $app for deployment"  
            updatedVersions.setProperty(app, requiredVersions.getProperty(app))  
        }  
    }  
  
    return updatedVersions  
}
```

re-deploy all
compare with running
compare with manifest



Last Successful Artifacts

	manifest	93 B		view
	updates	62 B		view

```
#Fri Mar 04 19:45:36 GMT 2016
mobile-deposit-ui/qcon=1.0.0-1
mobile-deposit-api/qcon=1.0.0-3
```

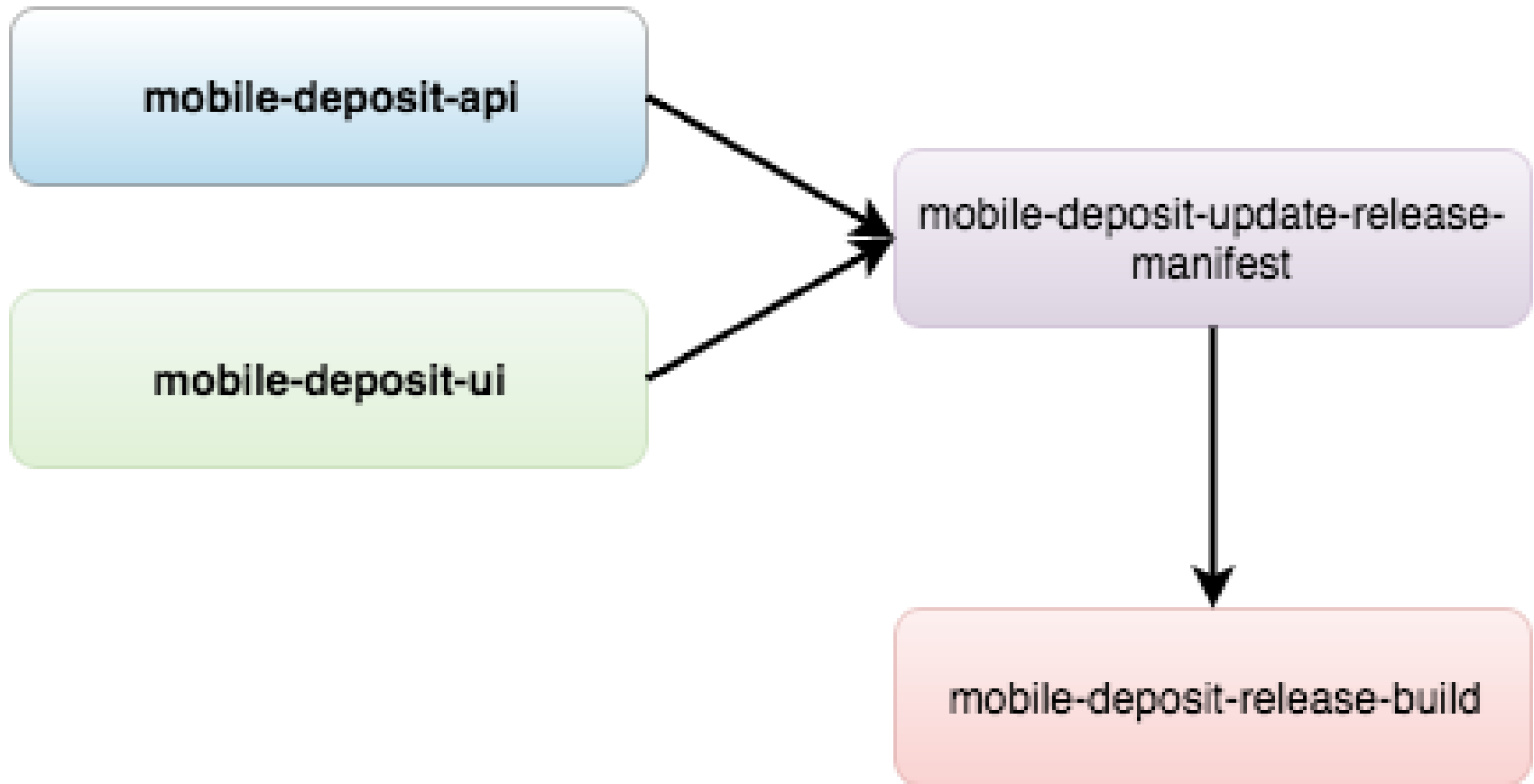
manifest

updates

```
#Fri Mar 04 19:45:36 GMT 2016
mobile-deposit-api/qcon=1.0.0-3
```

demo

Demo job flow



conclusion

pipelines \neq linear
assembly: multi-component
build fan-in
trigger downstream
track with manifests

Pipeline job type

immutable versions – single binary

How to get started

- Install pipeline plugin
 - <https://wiki.jenkins-ci.org/display/JENKINS/Pipeline+Plugin>
- Read the docs
 - https://documentation.cloudbees.com/docs/cookbook/_continuous_delivery_with_jenkins_pipeline.html
- Dzone ref card
 - <https://dzone.com/refcardz/continuous-delivery-with-jenkins-workflow>
- Download the mobile-api projects (qcon branches)
 - <https://github.com/harniman/mobile-deposit-api/tree/qcon>
 - <https://github.com/harniman/mobile-deposit-api/tree/qcon>
 - <https://github.com/harniman/mobile-deposit-release/tree/qcon>

Get Certified!



Be recognized for your Jenkins skills
Differentiate yourself from your peers
Increase your value to your employer
Hiring Managers can more easily identify qualified candidates



Learn more, register and get the Study Guides:
<https://www.cloudbees.com/jenkins-certification>



Table 12 on Level 3

Thank you and questions!!

www.cloudbees.com

nharniman@cloudbees.com
@harniman



CloudBees[®]

The Enterprise Jenkins Company



CloudBees®

CloudBees®