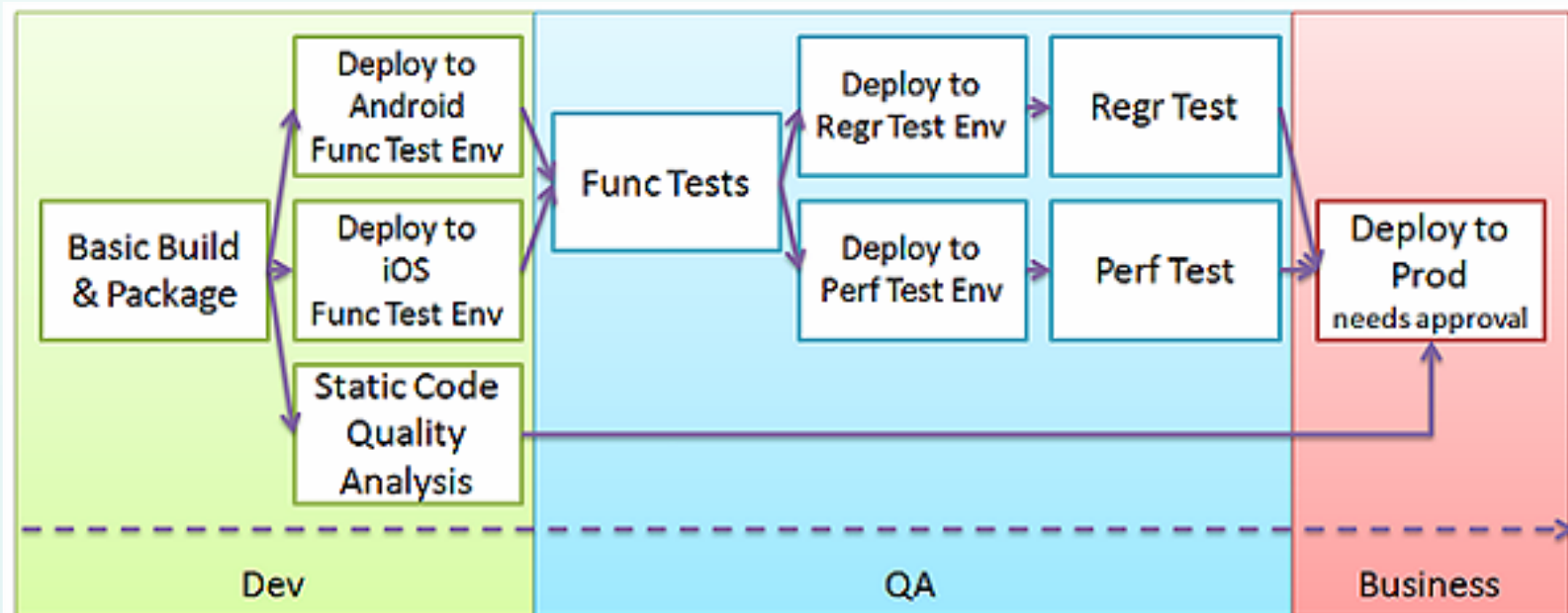




Workflow in Jenkins

Kohsuke Kawaguchi / CTO, CloudBees, Inc.
kk@kohsuke.org / @kohsukekawa

Jesse Glick / Developer, CloudBees, Inc.
@tyvole



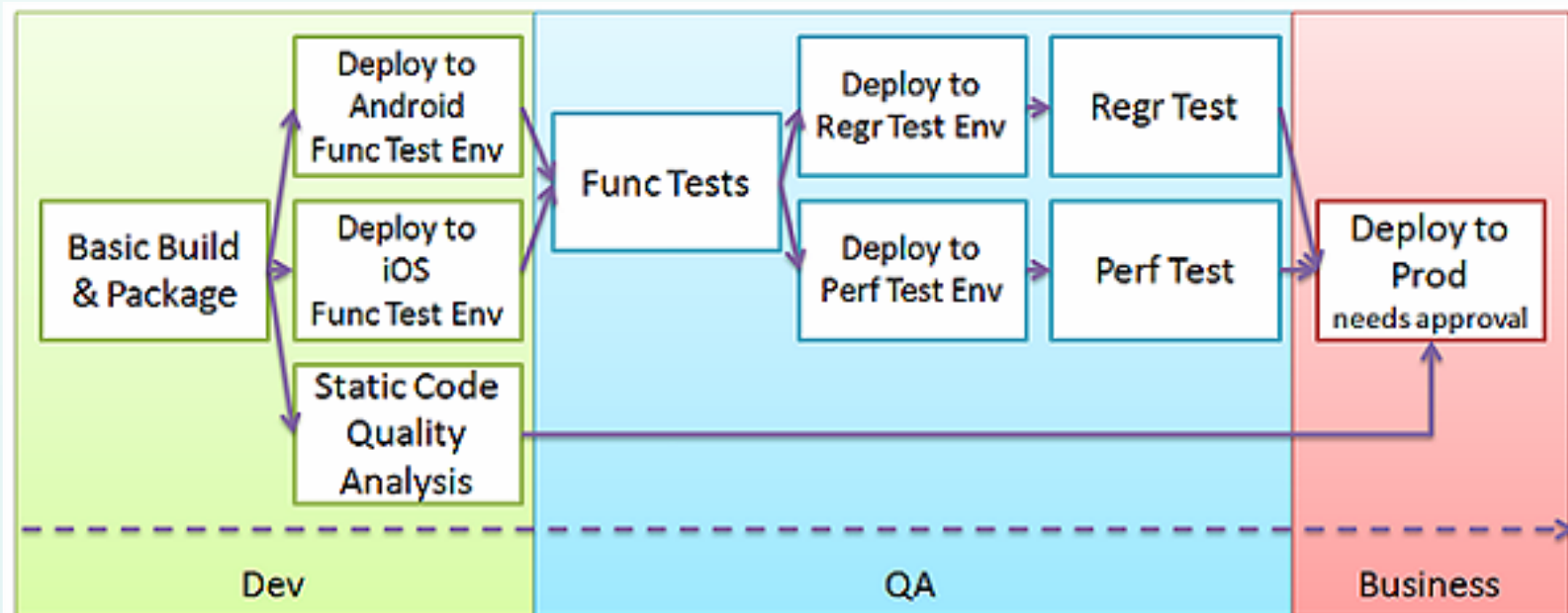
Use Cases: orchestrated activities





















- Multi-stage continuous deployment pipeline
- Run part of build with a temporary server
- Blue/green deployment + auto commit/abort
- Parallel tests with automatic sharding
- Automatic per-branch jobs (à la Literate)






Characteristics

- **Complex pipelines** involving multiple stages
- **Non-sequential logic** such as loops and forks
- **Long-running builds** must survive outages
- **Interaction with humans** pauses, input
- **Restartable builds** in case of a transient error
- **Reusable definitions** to avoid duplication
- **Comprehensible scripts** one clear definition



All	MyProject Build Pipeline	MyProject Delivery Pipeline	MyProject Jobs	Sample	+
S	W	Name ↓	Last Success		
		MyProject » 1 - Developer Jobs » Basic Build and Package	21 sec - #16		
		MyProject » 1 - Developer Jobs » Deploy to Android Func Test Env	52 sec - #17		
		MyProject » 1 - Developer Jobs » Deploy to iOS Func Test Env	41 sec - #12		
		MyProject » 1 - Developer Jobs » Static Code Quality Analysis	1 min 2 sec		
		MyProject » 1 - Developer Jobs » Trigger Deploy to Func Test Envs	1.4 sec - #1		
		MyProject » 2 - QA Jobs » Deploy to Perf Test Env	32 sec - #11		
		MyProject » 2 - QA Jobs » Deploy to Regr Test Env	31 sec - #11		
		MyProject » 2 - QA Jobs » Func Tests	25 sec - #12		
		MyProject » 2 - QA Jobs » Perf Tests	11 sec - #11		
		MyProject » 2 - QA Jobs » Regr Tests	6 hr 18 min		

S	W	Name ↓	Last Success	Last Failure
		MyProject	22 sec - #4	N/A 

```
node('linux') {  
  git url: 'https://github.com/foo/bar.git'  
  sh 'mvn install'  
  parallel([  
    regr: {sh './regression-test.sh'},  
    perf: {sh './performance-test.sh'}  
  ])  
  ...  
}
```


Key features (what I already covered)

- Entire flow is one concise Groovy script
 - for-loops, try-finally, fork-join, etc.
- Can restart Jenkins while flow is running
- Human input/approval integrated into flow



More key features

- Allocate slave nodes and workspaces
 - as many as you like, when you like
- Standard project concepts: SCM, artifacts, ...



Design: workflow in VCS (1)

- Load the script from another file

```
node {  
    git url: 'https://github.com/acme/foo.git'  
    load 'jenkins.groovy'  
}
```

- In repository

```
pom.xml  
src  
jenkins.groovy  
...
```

Design: workflow in VCS (2)

- Shared workflow scripts/utilities in Git



- Your job just needs to have this

```
org.acme.MyWorkflow.run(this)
```



Build #153

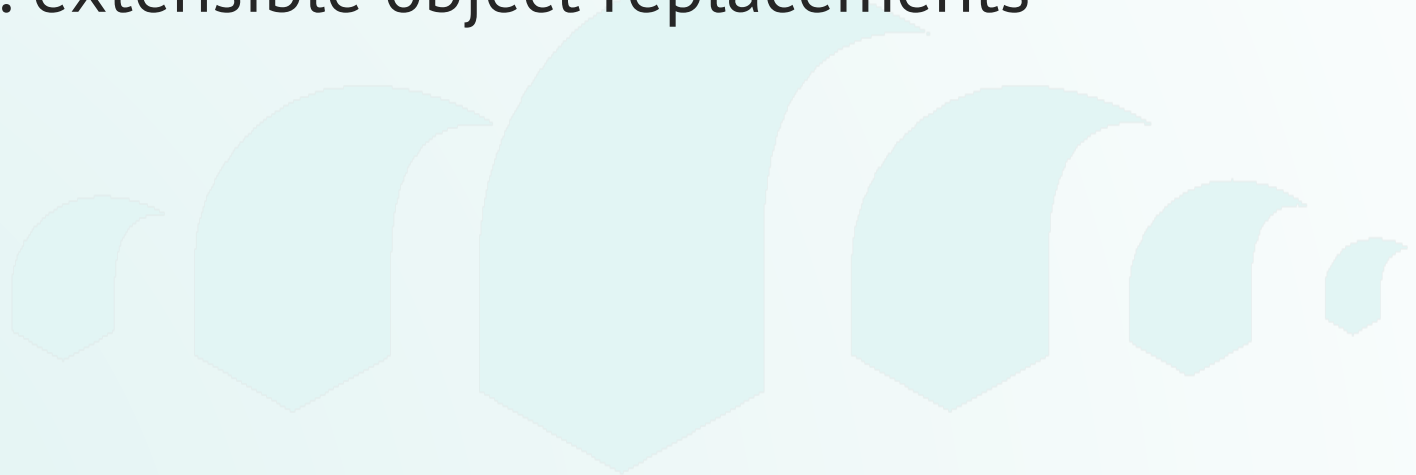


foo()

```
void foo() {  
    try {  
        sh 'sleep 1h'  
    } finally {  
        sh 'rm -rf *.tmp'  
    }  
}
```

Resumption of Groovy flows

- Transformed to “continuation-passing style”
- Run on custom interpreter of Groovy
- State of program saved at each pause point
- Variables serialized and restored after restart
 - pickles: extensible object replacements



Resumed builds to the user

- It “just works”
- (Serializable) local variables restored too
- Shell-like steps survive restart
 - Reconnection of slave, too





Jenkins Enterprise by CloudBees

- CloudBees' value-added Jenkins distro
 - Support & additional plugins
 - Upcoming: workflow enhancement features
- Resume from checkpoint
 - Can pick up artifacts from original build
 - No need to rerun earlier expensive steps
- Visualization
 - Start off with workflow stage view
 - More planned later

Demo

Still to come

- workspace management
- robustness
- polished UI
- much more, tracked in Trello



Status

- Open for contribution
- github.com/jenkinsci/workflow-plugin
- 0.1-beta-5 binaries on experimental UC
- Requires Jenkins 1.580+ today
- Fundamentals all work now
- Aiming for 1.0 shortly
- Considered strategic by CloudBees

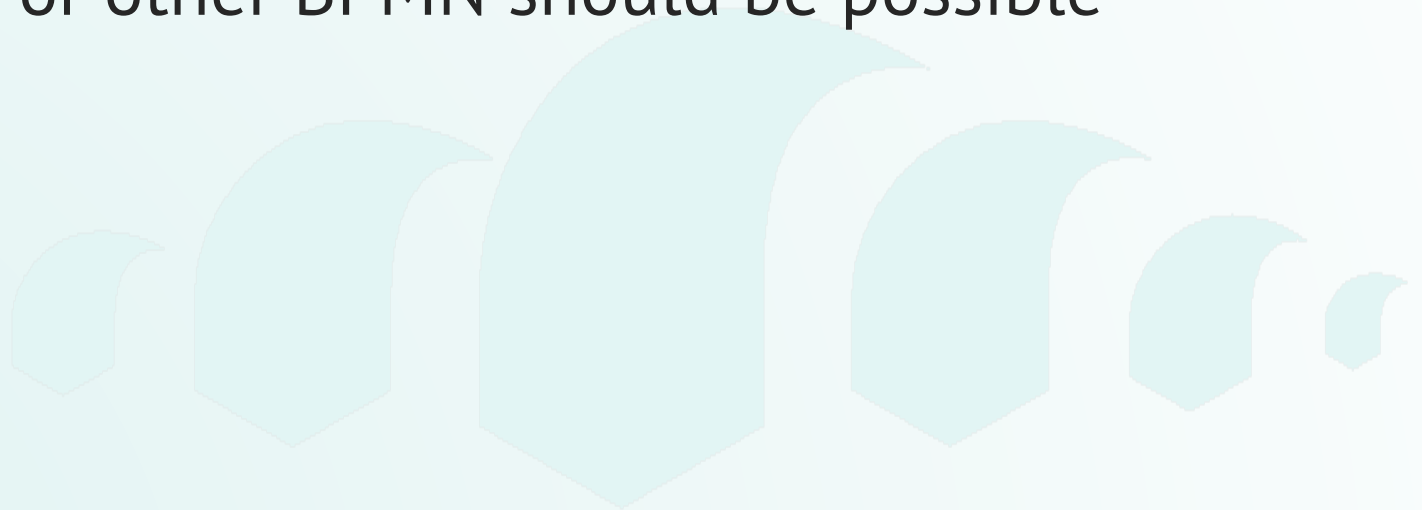
Summary

- Simplifies complex orchestration
- Define everything in single job
- Workflow as Groovy code
- Survives JVM loss, and even restartable
- Extensible
- github.com/jenkinsci/workflow-plugin



Design: overall

- suite of Jenkins plugins
 - Jenkins Enterprise may add checkpoints, etc.
- pluggable flow definition & execution engine
 - Groovy CPS is recommended choice
 - Activiti or other BPMN should be possible



Design: flows

- persistent record of execution
- directed acyclic graph of nodes
- some nodes represent one step
- others indicate block start/stop structure
- nodes may have associated metadata
 - console log fragment contributes to main log
- pluggable visualizations for different views

Design: steps

- standalone API for asynchronous build steps
- *context* serves as an identifier & callback
 - also offers logger, build, workspace, &c.
- support for block-structured steps
 - invoke body 0+ times with revised context
- standard step for “durable” shell/batch scripts
- standard steps for SCMs (git, svn, hg)
 - >1 SCM per build possible

Design: interoperability

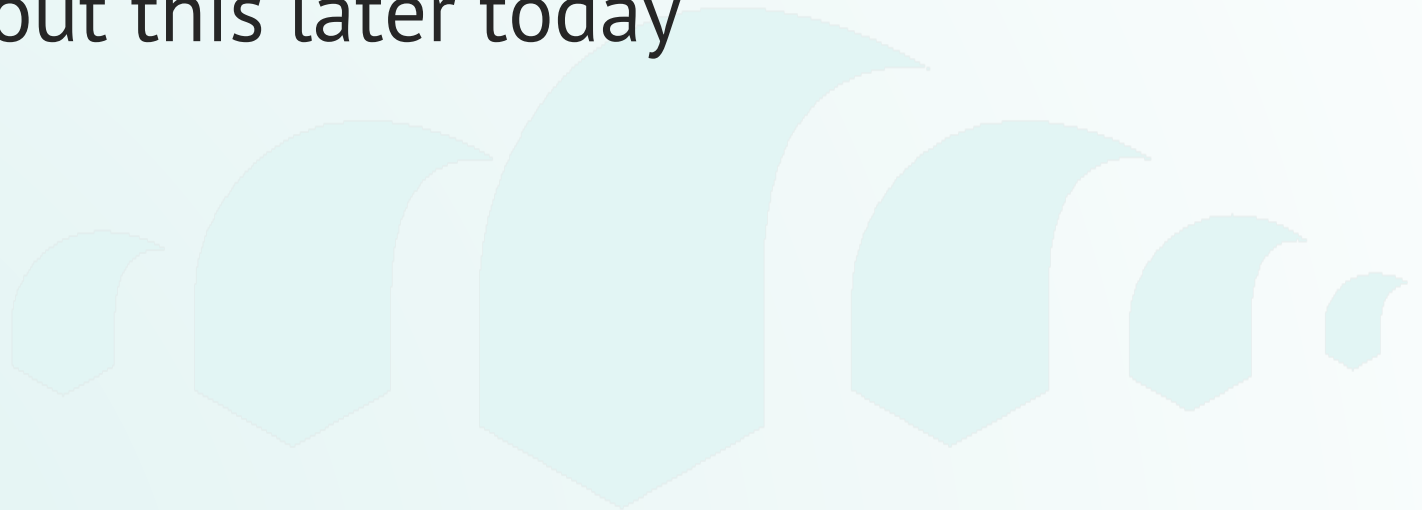
- run on existing Jenkins slaves
- SCM plugins supported with modest changes
- some existing build steps & publishers
- trigger existing jobs
- standard build history, artifacts
- needs Jenkins 1.580.x or later
 - features factored out of standard projects

<END>



Workflow

- github.com/jenkinsci/workflow-plugin
- Bunch of new extension points
- Expect some value-adds in Jenkins Enterprise by CloudBees
- More about this later today



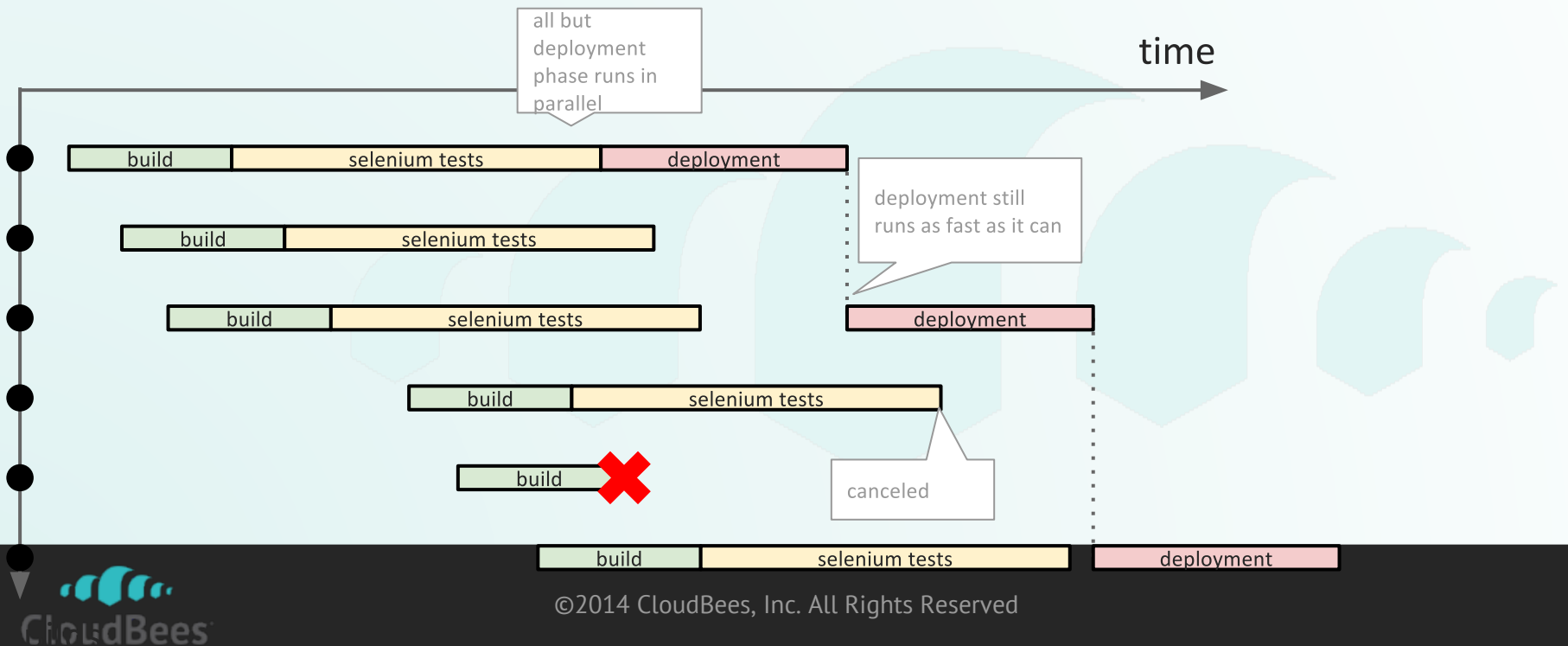
Workflow

Workflow: the one-pager

```
with.node('linux') {  
  git(url: 'git://server/myapp.git')  
  sh('mvn clean package')  
  archive('target/myapp.war')  
  stage('Test')  
  parallel({  
    sh('mvn -Psometests test')  
  }, {  
    sh('mvn -Pothers tests test')  
  })  
  input('OK to deploy?')  
  stage(value: 'Deploy', concurrency: 1)  
  sh('mvn deploy')  
}
```

Stages (aka James Nord operator)

- Pipeline throttling primitive
- Special semaphore: only newest build may wait



Lots to get excited about!

- Enterprise?
 - JOC / JE
- OSS / early adapter?
 - Workflow
 - Traceability



Project setup

- one workflow is defined as a job
- single script for all steps
- build triggers & parameters like regular projects
- SCM, publishing, ... are all part of script
- each build shown using regular Jenkins view
- Graphical visualizations of actual build possible
 - (not of job definition; could be too dynamic)