



# Git Flow

A simple, automated,  
Continuous Delivery compliant  
strategy for using Git



JENKINS CI  
USER EVENT  
CPH 2014



# Agenda

- Intro
- The Atmel Situation
- The branchy approach to pretested integration
-



# A transition to modern development

- 4 years ago
  - AVR Studio
    - a debugger and a target programmer for AVR 8-bit MCU's
  - AVR32-Studio
    - an IDE for AVR 32-bit MCU's development and debugging based on Eclipse
  - 10-12 SW-developers in one location



# Problems



- 1 major release a year
- 1-2 month release cycle
- Several bugfixes after release
- Upfront planning and estimates for each development cycle
- Several feature changes during development cycle
- Lots of manual tasks for building and testing
- Person dependent
- Not scalable

# Challenges

- Management wanted one solution for all MCU architectures
- Management want more features faster
- Changing priorities
- Not modular enough design
- More resources spread on several locations





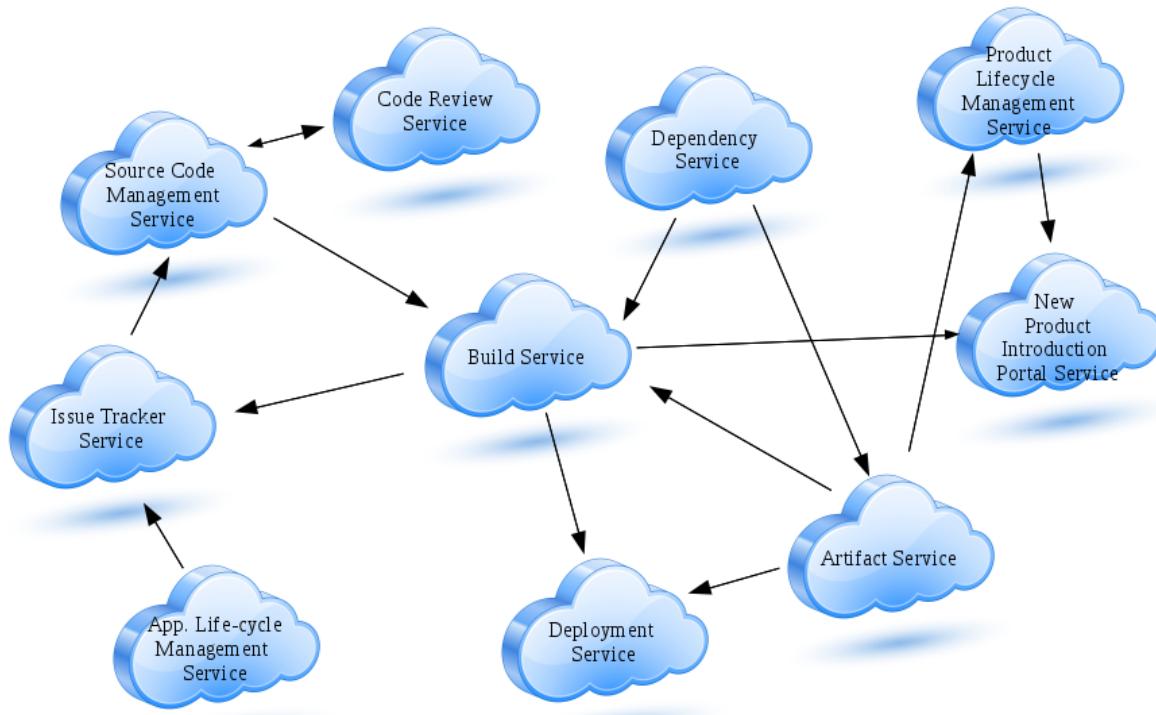
# Solution?

- Choose to start working Agile and Scrum
- Started using Jenkins (Hudson) to automate building
- Started automate testing
- Switch from Bugzilla to Atlassian Jira
- Added Atlassian Agile after a period with manual scrum planning
- Added several new people in new locations



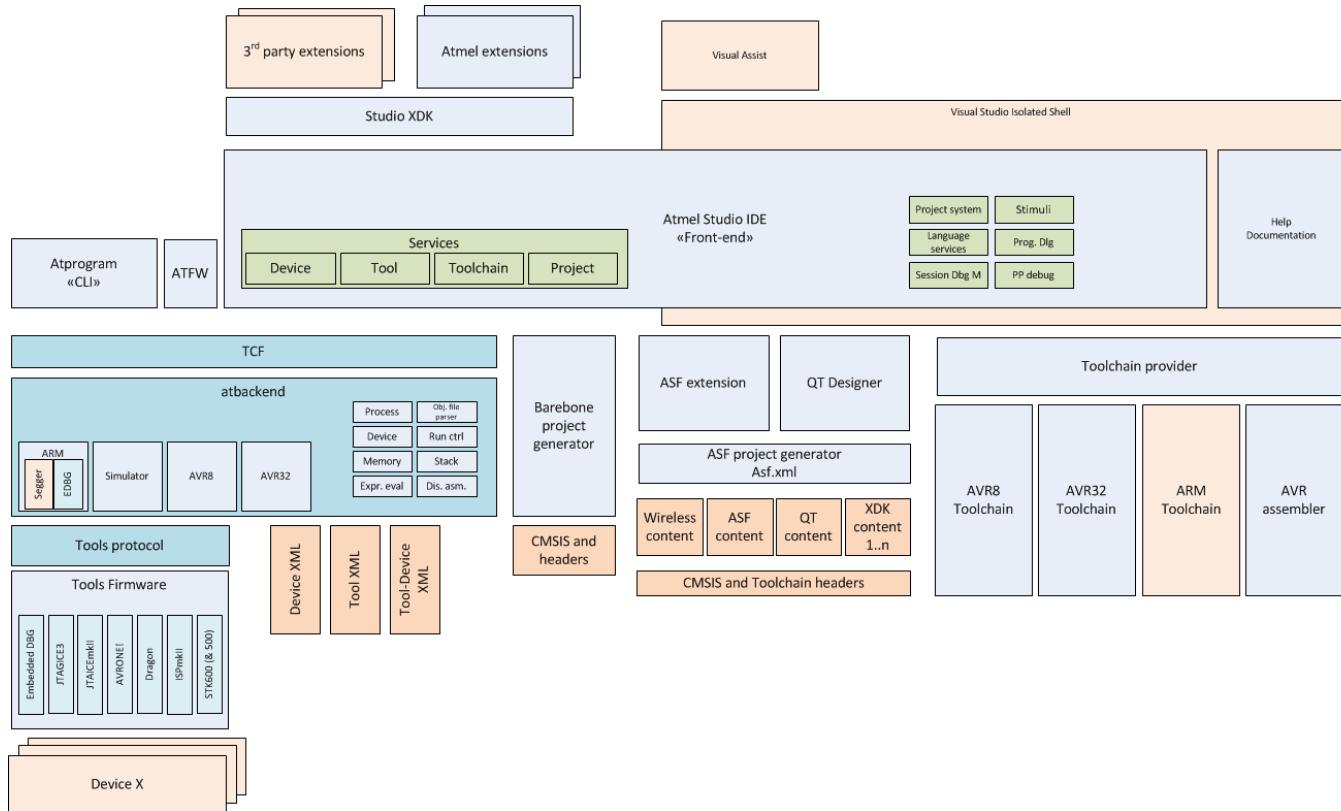
JENKINS CI  
USER EVENT  
CPH 2014

# Make services available for development





# Atmel Studio



JENKINS CI  
USER EVENT  
CPH 2014

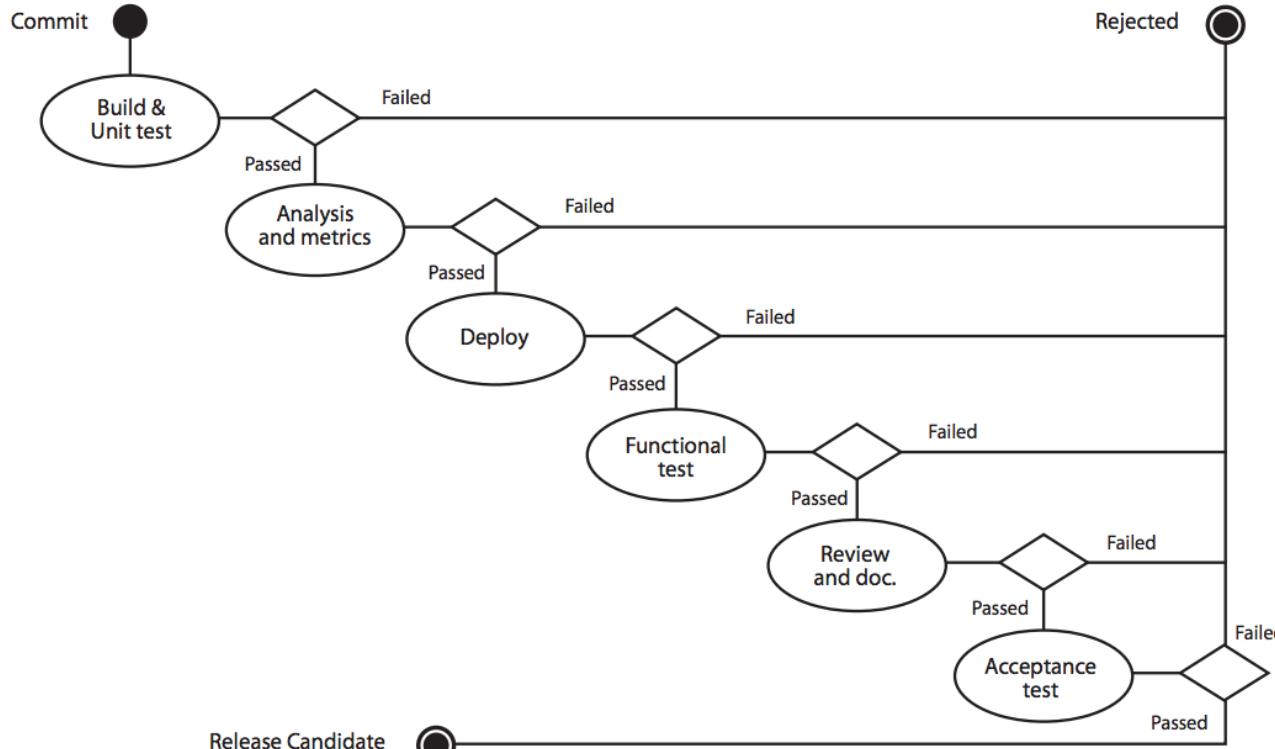
# New challenges

- Popularity of Jenkins grew very fast
- More people demanded more uniform processes
- Need for sharing info, knowledge and learning
- Management and other department need to understand how we now work scrum-wise
- Needed an outside view and input to take the next step





# Continuous Delivery Pipeline

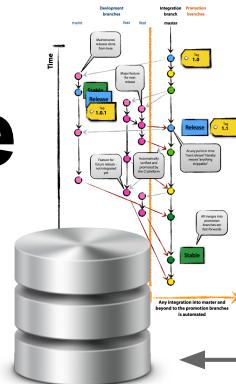


JENKINS CI  
USER EVENT  
CPH 2014

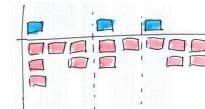
# CD Story-line



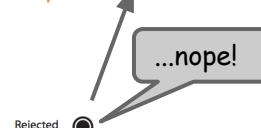
Automation Platform



Developer



Work item



Rejected

...hope!



Release Candidate

Continuous delivery  
Pipeline



Done done



JENKINS CI  
USER EVENT  
CPH 2014



# The “branchy” approach

...to pretested integration



# ClearCase UCM Plugin

ClearCase UCM

Stream

Component

Promotion level

Load modules

Polling

Create baseline

Name template   
The template seems ok

Recommend baseline

Make tag

Set description



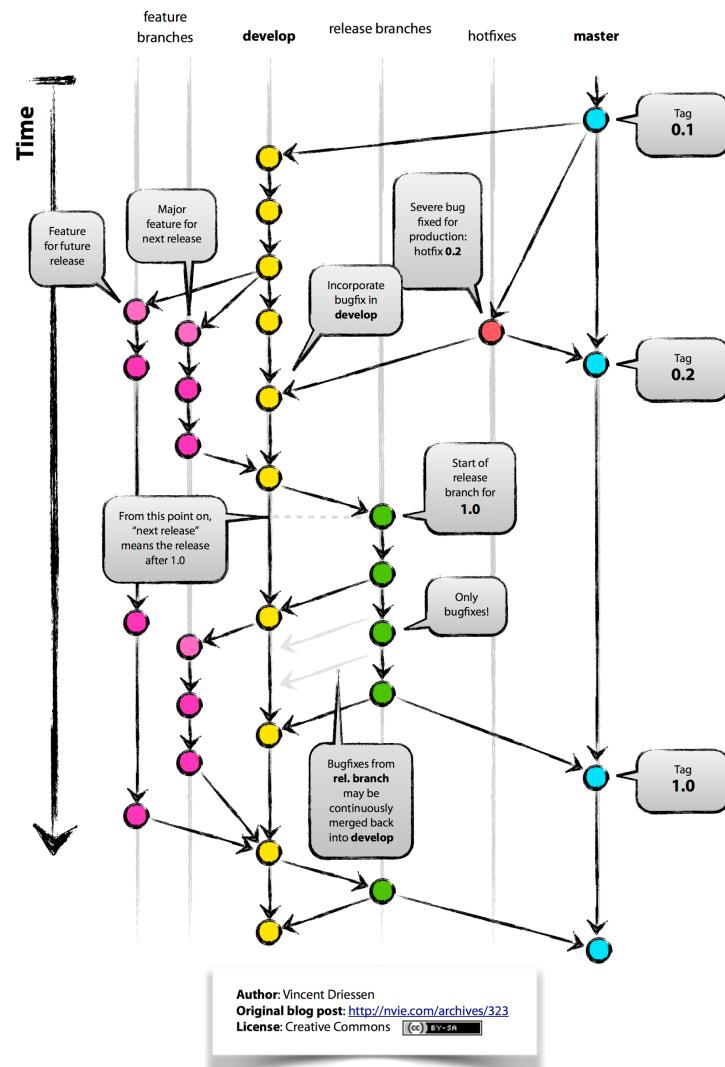
**pelagicore**

**syntigo**

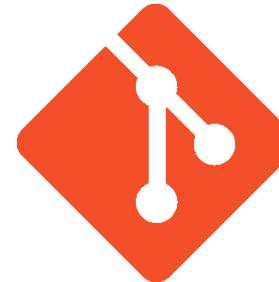


[www.josra.org](http://www.josra.org)





# @nvie Git-flow



JENKINS CI  
 USER EVENT  
 CPH 2014





# Vocabulary

## **Project integration branch**

The shared work context of the developers where integrations are done frequently.  
Typically master but occasionally a maintenance branch.

## **Promotion branches** (a.k.a big picture branches)

All the branches which are above and beyond the project integration branch.

## **Development branches** (a.k.a small picture branches)

All the branches that produce contributions to the integration branches. These are the branches where developers actually commit code.



JENKINS CI  
USER EVENT  
CPH 2014



# Vocabulary

## Maintenance branches

Seen from the perspective of the project integration branch it's merely a development branch - which someone just happened to allow a release from. Maintenance branches are guarded by the same level of verifications as the project integration branch.

## ~~Hot-fix branches~~

In Continuous Delivery a hot-fix in the meaning "*a shorter and faster track to a release than the regular path*" is a contradiction in terms, they simply do not exist. There is only one track to a release and that is a successful ride through the pipeline. We will erase this term from our vocabulary.



JENKINS CI  
USER EVENT  
CPH 2014

# Principles



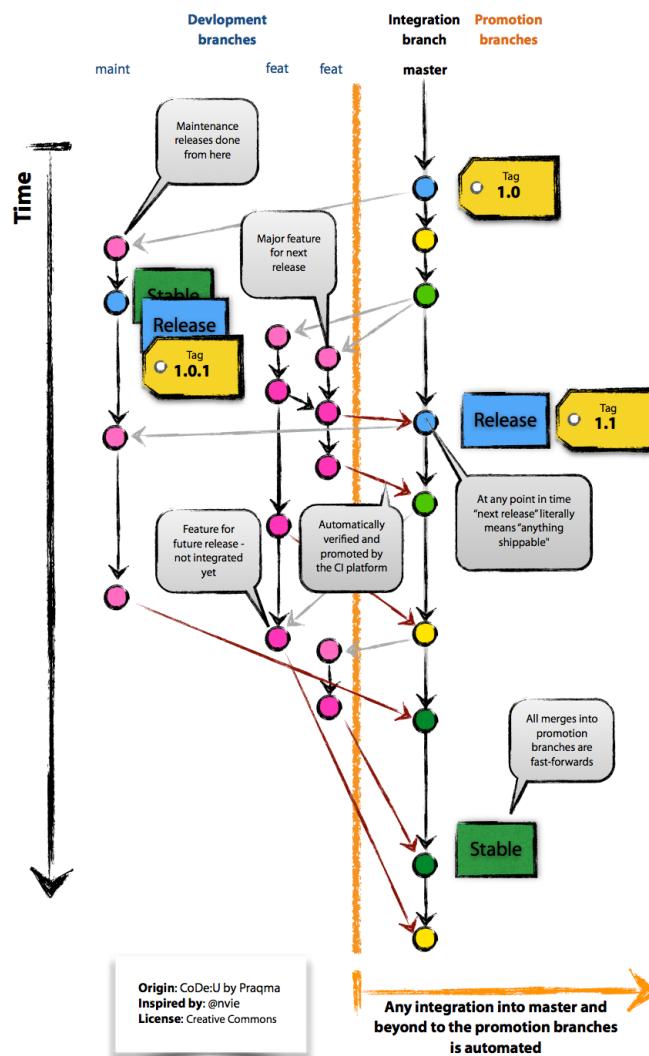
## Git-flow



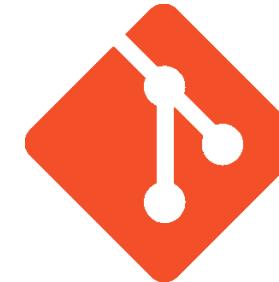
1. The project integration branch is where developers continuously integrate (CI).
2. The project integration branch in Git is master.
3. All integrations onto an integration branch must pass an automated toll-gate.
4. Only trivial merges are allowed onto the integration branch.
5. Every successful integration onto master kicks off the pipeline.
6. The integration branch is always aiming for the next release, anything that shouldn't go into the next release are kept away from master on development branches until it's due.
7. Any push to a centralized repository, that contains a commit on a development branch matching the naming convention triggers an automated integration..
8. Some development branches may be tied to maintenance branches rather than master.
9. All integrations onto promotion branches are automated (*"the definition of done"*).
10. Any given promotion branch can only have one contributor - merges are guaranteed to be *fast-forward*.



JENKINS CI  
USER EVENT  
CPH 2014



# Git-flow



JENKINS CI  
 USER EVENT  
 CPH 2014

## Branching model

|               |                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Development * | <input style="border: 1px solid #ccc; padding: 2px 10px; width: 150px; height: 20px; border-radius: 5px; font-size: 10px; font-weight: bold; color: inherit; background-color: inherit;" type="button" value="master"/><br>The integration branch used for development. Feature branches are merged back into this branch                                                              |
| Production    | <input style="border: 1px solid #ccc; padding: 2px 10px; width: 150px; height: 20px; border-radius: 5px; font-size: 10px; font-weight: bold; color: inherit; background-color: inherit;" type="button" value="Select branch"/><br>The branch used for deploying releases. Typically, this branches from the development branch and changes are merged back into the development branch |

"Project Integration Branch" choose master



No Entry

## Branch prefixes

Use the prefixes below as part of your branch names to categorize them and take advantage of automatic branching workflows. [Learn more](#)

|                                                                                                                                                                |                                                                                                                                                                                                                                        |                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bugfix                                                                                                                                                         | <input style="width: 150px; height: 20px; border: 1px solid #ccc; border-radius: 5px; font-size: 10px; font-weight: bold; color: inherit; background-color: inherit;" type="text" value="fix/"/> <input checked="" type="checkbox"/>   |  "Development branch" aiming for "maintenance"                         |
| Typically used for fixing bugs against a release branch                                                                                                        |                                                                                                                                                                                                                                        |                                                                                                                                                         |
| Feature                                                                                                                                                        | <input style="width: 150px; height: 20px; border: 1px solid #ccc; border-radius: 5px; font-size: 10px; font-weight: bold; color: inherit; background-color: inherit;" type="text" value="dev/"/> <input checked="" type="checkbox"/>   |  "Development branch" aiming for master - This is your <u>default!</u> |
| Used for specific feature work. Typically, this branches from and merges back into the development branch                                                      |                                                                                                                                                                                                                                        |                                                                                                                                                         |
| Hotfix                                                                                                                                                         | <input style="width: 150px; height: 20px; border: 1px solid #ccc; border-radius: 5px; font-size: 10px; font-weight: bold; color: inherit; background-color: inherit;" type="text" value="hotfix/"/> <input type="checkbox"/>           |  No such thing as a "Hotfix"                                           |
| Typically used to quickly fix the production branch                                                                                                            |                                                                                                                                                                                                                                        |                                                                                                                                                         |
| Release                                                                                                                                                        | <input style="width: 150px; height: 20px; border: 1px solid #ccc; border-radius: 5px; font-size: 10px; font-weight: bold; color: inherit; background-color: inherit;" type="text" value="maint/"/> <input checked="" type="checkbox"/> |  "Maintenance branch"                                                  |
| Used for release tasks and long-term maintenance. Typically, this branches from the development branch and changes are merged back into the development branch |                                                                                                                                                                                                                                        |                                                                                                                                                         |

Yack! Sound like one thing that shouldn't be automated

# Stash Compliancy

## Automatic merge

Enable automatic cascading merging from release branches to their downstream branches. To use this feature, the release branch type must be enabled. [Learn more](#)

Enable automatic merging





# CoDe:U git-flow

Git  
Repositories

Repository URL  ? ?

Credentials  Advanced...

Branches to build

Branch Specifier (blank for 'any')

Git executable

Repository browser  ?

Additional Behaviours

**Clean after checkout** ?

**Prune stale remote-tracking branches** ?



JENKINS CI  
USER EVENT  
CPH 2014



# CoDe:U git-flow

Use pretested integration

Git

Integration branch

**Pre-tested integration strategy**

Accumulated commit ?

Squashed commit ?

Atmel

```
git push origin mydev:ready/case_167
```



JENKINS CI  
USER EVENT  
CPH 2014



# Know your tools - practice your knife skills



JENKINS CI  
USER EVENT  
CPH 2014



# Git branching Kung Fu

- A branch is a substitute name for a HEAD, that dynamically changes as you add new commits
- Anything *reachable* from the branch - belongs to the branch
- Using **bisect** as an example:
  - `merge ...plain vanilla` doesn't cut it
  - `--no-ff ...` doesn't quite either
  - `--squash ...` is the solution

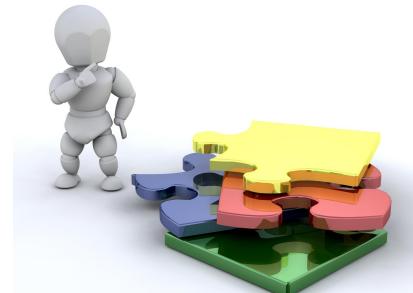
More details at [www.josra.org/blog](http://www.josra.org/blog)



JENKINS CI  
USER EVENT  
CPH 2014

# Status today

- More than 1200 users on internal Jira
- 300 developers uses Stash/Git
- Several departments looks to the Tools department for adapting the new workflow
- Linking Salesforce and Jira
- Quality much better
- Release cycle much shorter
- It's fun to develope



# Future and coop with Praqma?

- Use Praqma's expertise and network to further develop the workflow and best practice
- Participate in Josra to cooperate with others in the same situation
- Continuous review and refactor processes and tools to optimize and make sure we follow good Software Craftsmanship





# References

- **The Pretested Integration Plugin:** [wiki.jenkins-ci.org/display/JENKINS/Pretested+Integration+Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Pretested+Integration+Plugin)
- **The CoDe:U Git Flow paper:**  
[www.praqma.com/git-flow](http://www.praqma.com/git-flow)
- **Blog post on some of the finer details of the plugin:** [www.josra.org/blog/2014/06/23/Pretested+integration+plugin.html](http://www.josra.org/blog/2014/06/23/Pretested+integration+plugin.html)
- **Joint Open Source Roadmap Alliance:**  
[www.josra.org](http://www.josra.org)
- **Continuous Delivery Users on LinkedIn:**  
[www.linkedin.com/groups/Continuous-Delivery-Users-4893734](http://www.linkedin.com/groups/Continuous-Delivery-Users-4893734)
- **Praqma:**  
[www.praqma.com](http://www.praqma.com)
- **Mail to Lars:**  
<mailto:lak@praqma.com>



JENKINS CI  
USER EVENT  
CPH 2014