

Product Management

Thierry Sans

Outline

After automatic build (so far)

- **Continuous Development**
working with branches in version control
- **Continuous Integration**

Continuous Development

Working with branches in version control

Problems

- A developer will do several commits before having a working product with a new feature
- ➔ **We should always keep the latest working version of the product somewhere**
- Developers might work on different features and their work-in-progress commits will keep on conflicting with each other
- ➔ **Developers should work on independent versions of the product until their feature is ready to be merged**

Why using branches

- Keep the latest working version of the product on the **master branch**
- Develop each feature on its own **dev branch** (or feature branch) separated from the master
- Once a feature is ready, **merge its dev branch into the master branch** to create a new version of the working product

Branch

Create a branch

```
$ git branch branch_name
```

Switch to that branch

```
$ git checkout branch_name
```

First push on a branch

```
$ git push --set-upstream origin branch_name
```

GIT MERGE



Merge a dev branch

Always merge the master into the dev branch first

- Step 1 : merge master into branch
\$ git merge master
- Step 2 : solve conflicts (if any) and push
- Step 3 : merge branch into master (no conflict this time)
\$ git checkout master
\$ git merge development

Continuous Integration



Problem

- Developers implement, build and test each feature separately
- The product owner merges features (dev branches) into the working version of the product (master)
- ◉ Two features might interfere with each other and break the build and/or the tests when put together
- ➔ The product owner must (re)build and the (re)test after each feature merge

Continuous Integration (CI)

Automatically pull, build and test the latest commit (usually from the master) on a non-developer machine

Existing CI tools

Type	Example
Non hosted	 Jenkins
Hosted	 Travis CI

Going further

- **Continuous Delivery**

Automatically validate that your product is ready to be release to customers (a.k.a production ready)

- **Continuous Deployment**

Automatically release your product to customers (a.k.a migrate and deploy your product to production)

