

Lunch meeting 07-03-2013

Software Engineering @ NLeSC

Rob van Nieuwpoort

netherlands

eScience center

by SURF & NWO

A change of perspective

```
if (someCondition) foo();
```

```
if (someCondition)  
    foo();
```

netherlands

eScience center

by SURF & NWO



A change of perspective

```
if(someCondition)
    for (int i=0; i < 10; i++) {
        foo();
        bar();
    }
```



A change of perspective

```
if(someCondition)
    for (int i=0; i < 10; i++) {
        foo();
        bar();
    }
    integrate();
```



A change of perspective

- We are, of course, all über programmers. 😊
- The person extending / maintaining your code 10 years from now may not be!
- It is not a competition, you don't have to show off your skills all the time.
- Don't write code that *you* can read, write code that is easy to understand for “bad programmers”.
- Software engineering is equally important for über programmers and domain experts.

netherlands

eScience center

by SURF & NWO



A change of perspective

```
if (someCondition) {  
    foo () ;  
}
```



Software Engineering

- **eScience software quality is not up to standards**
 - PhD-ware
 - Domain experts without CS education
 - Lack of time / interest
 - Über programmers
- **Open discussion**
- **Guidelines will be enforced after discussion**

netherlands

eScience center

by SURF & NWO



A Pragmatic Approach

- **Internal / eSTeP software**
 - Follow guidelines
- **Software that is integrated into larger project code base**
 - Follow what the project does
 - Language, coding style, etc.
 - Follow our documentation guidelines
- **In project, but separate tool / library**
 - Make sure it integrates in project
 - Opportunity to educate colleagues in best practices
 - Lead by example



eSTeP Sprint

- **Second week of April**
 - Jason, Niels, Piter, Stefan, Rob
 - Integrate existing software stacks
 - JavaGAT, Ibis deploy, pilot job framework, vBrowser, web portals
 - New unified and concise interface
 - Plug-ins for local, SSH, SARA, SSH, ?
- **Apply software engineering guidelines**
 - Pilot project and example for the projects



Licensing

- **Everything we do is Open Access / Open Source unless...**
- **Proposal: Apache 2.0 License (<http://www.apache.org/licenses/LICENSE-2.0.html>)**
 - Open Source, OSI and FSF approved (Free Software)
 - Interoperable, non-viral, linking allowed
 - GPL 3-compatible, not with 1 and 2
 - Track modifications in source code
 - Permits code that it covers to be subsumed into closed source projects
 - Explicitly grants patent rights where necessary to operate, modify and distribute the software
- **Unless**
 - Project requires otherwise
 - Viral licenses are used (e.g. LOFAR is GNU GPL)

netherlands

eScience center

by SURF & NWO



How to apply the Apache License to your work

- **LICENSE** file with the license (<http://www.apache.org/licenses/LICENSE-2.0.html>)
- **NOTICE** file describing that we require acknowledgement, give paper reference if applicable
- **File header:**

```
Copyright [yyyy] [name of copyright owner]
```

```
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.
```

netherlands

eScience center

by SURF & NWO



Tracking Changes

/**

@author John Doe

09/11/2011: Initial creation

01/01/2012: Jane Doe: Fixed Bug X

01/05/2012: Joe The Plumber: Added feature Y

03/10/2012: Mr. X: Rewrote function Z

****/**

netherlands

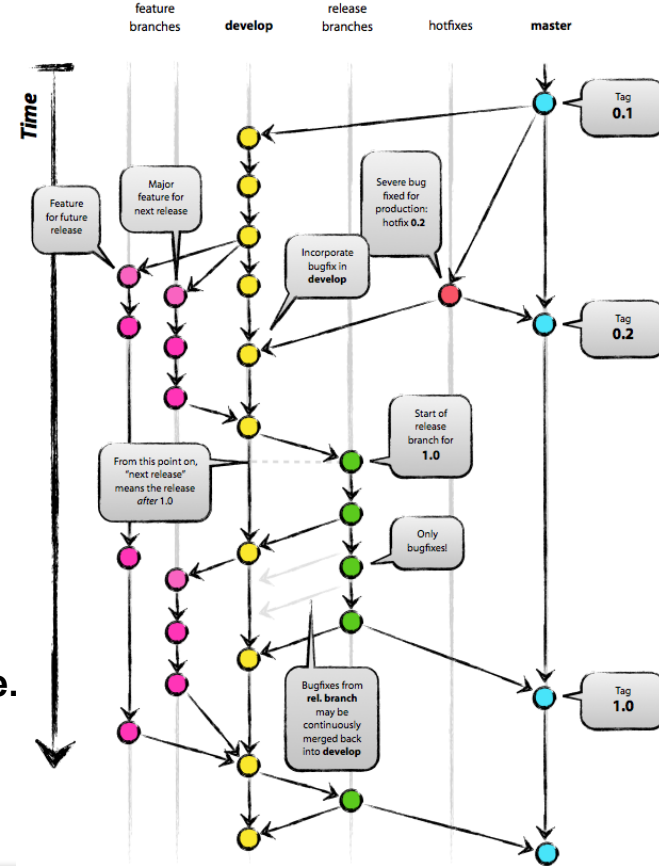
eScience center

by SURF & NWO



Version control

- [Git](#)
 - Free and open source distributed version control system.
 - Branching model:
<http://nvie.com/posts/a-successful-git-branching-model/>
 - Mateusz will give tutorial
- Github: <https://github.com/NLeSC>
 - We have a Github organizational account, stores all code developed here.
 - Also need personal account, which is coupled with eScience.
 - Organizational account supports closed-source code.
 - Also hosts Software Engineering wiki.
- Unless project uses different system



netherlands

eScience center

by SURF & NWO

Also in GitHub

- **Code snippets library**

- Small snippets of code that can be directly reused in other projects, but that are too small to put in a library.
- E.g., Maarten's colorspace code.

- **Ready-to-go demos**

- Attractive demos to be shown at any time by anyone
- Special branches in git that contain fully stand-alone compiled demos, including a slide deck, that can just be checked out and used directly.



Preferred Languages

- **Java**
 - Preferably not C, C++, C#, Fortran, ...
- **OpenCL**
 - Preferably not Cuda, ...
- **Python**
 - Preferably not Perl, ...
- **Coupling languages (see wiki)**
- **Again, unless ...**



Preferred Frameworks / Libraries

- **Extjs javascript webGUI framework**
 - <http://www.sencha.com/products/extjs/>



Good practices

- **Documentation**
- **Formatting / coding style**
- **Testing**
- **Bug tracking**

netherlands

eScience center

by SURF & NWO



Documentation

- **Source code:**
 - Javadoc, doxygen, etc, including private fields
 - Documentation is for users and maintainers!
- **Manual**
 - PDF, include installation instructions, usage, examples, screenshots, NLeSC logo
 - Send to coordinator to proof read



Coding style

- Here a very nice article about why coding styles matter, and increases software quality:
[Improving software quality, why Coding Style Matters](#)
- *“Programs are meant to be read by humans, and only incidentally for computers to execute.”*
— H. Abelson and G. Sussman (in “Structure and Interpretation of Computer Programs”)
- For eclipse, we will provide good default settings for formatting and cleaning your code, for different languages.
 - Standard Java Coding guidelines, with wider columns.
 - Default eclipse settings are broken.
- Name spaces: org.nlesc ?
 - Package name in Java
 - Namespace in C++



Testing

- **Unit tests**
- **Continuous integration**
 - Jenkins, on many different platforms
- **Let coordinator test before release**
 - (She/he can delegate)



Bug tracking

- **Github built in**
- **Redmine**



Releases

- **Binary**
 - Tag version in git
 - Put in GitHub, also documentation
 - Link to GitHub from project web site
- **Source**
 - Tag version in git, provide src.zip (can be automated in GitHub)
 - Also provide javadoc, manuals in repository
- **Let coordinator test and proof read**
- **Send description to Scott for dissemination**

