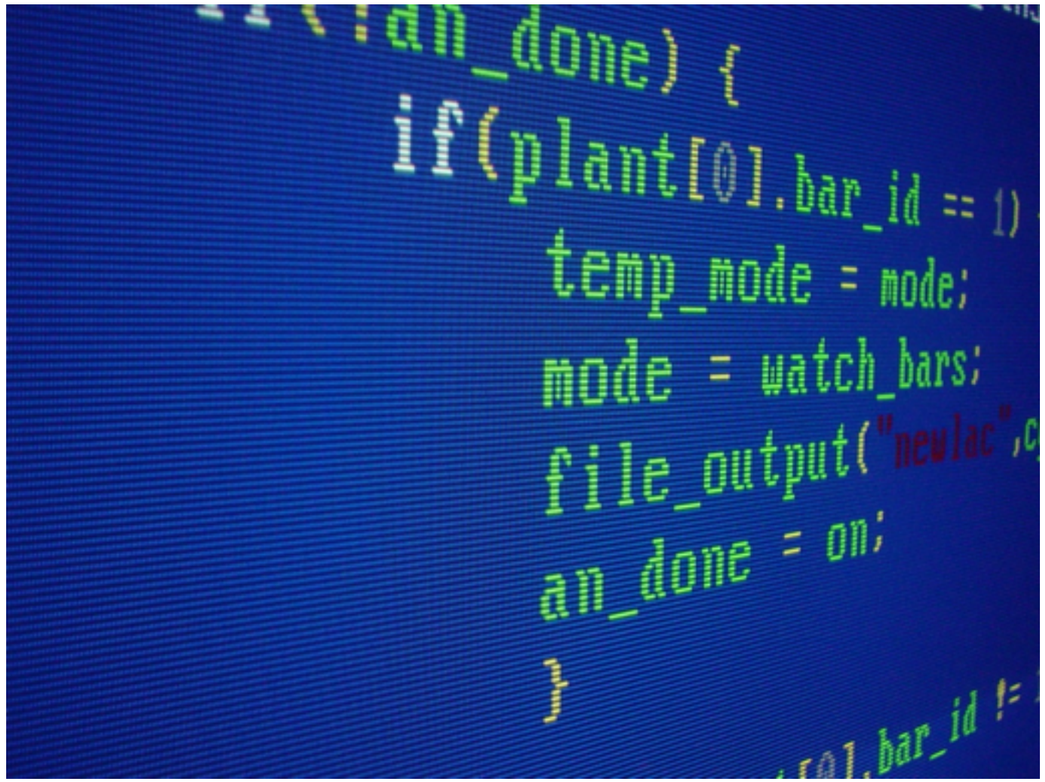
CERN Spring Campus 2014

Software Quality: A Python Case Study

Patrick GLAUNER
CERN
patrick.oliver.glauner@cern.ch
April 16, 2014

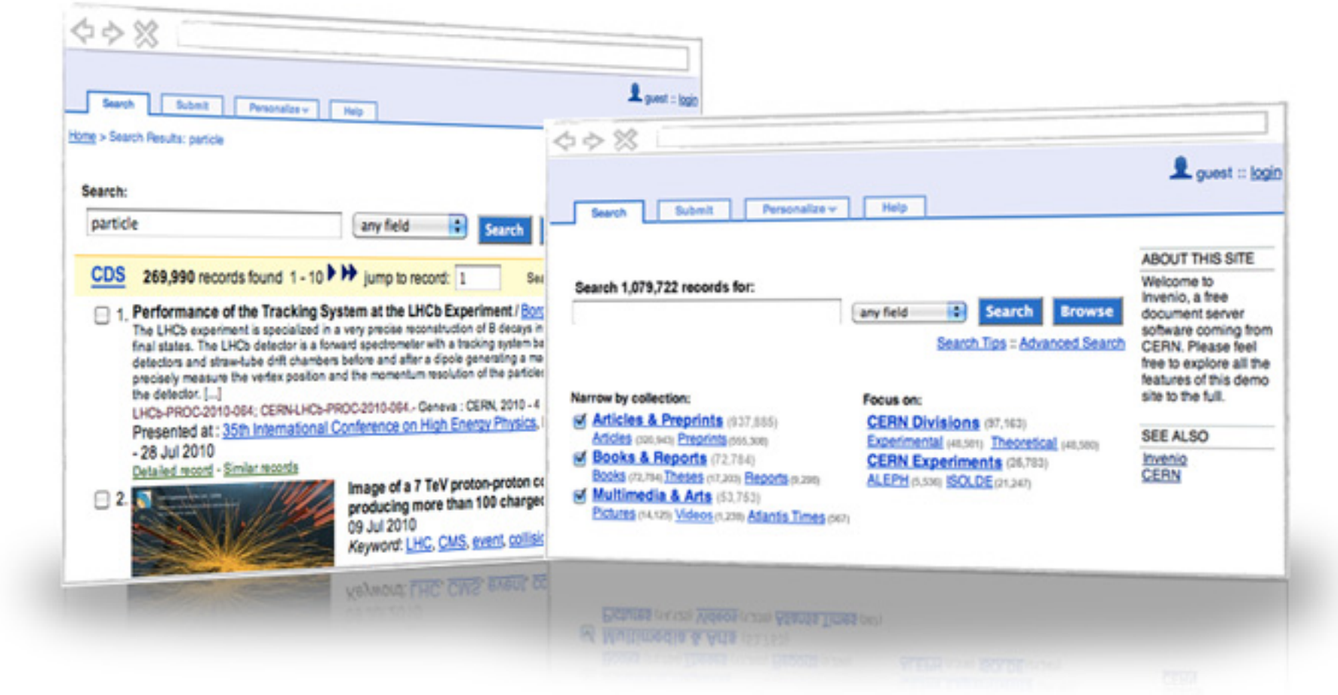
About me

- Graduated from Karlsruhe University of Applied Sciences, Germany in 2012
- Software Engineer at CERN since 2011
- R&D focus on document management and search engines



Invenio Development

- Invenio is a free document management system
- Originally developed at CERN, nowadays by an international collaboration
- More than 30 instances around the globe
- http://invenio-software.org/



Agenda

- Motivation
- Invenio Development:
  - Python
  - Git
  - Testing
  - Development Scripts
  - Continuous Integration
- Conclusions and Prospects
- Q&A

Motivation

- Software quality is important for a software product's success
- Avoiding/reducing bugs
- Readable and maintainable code
- Case study on selected QA topics of a large software project

Motivation: What is Software Quality?

- "What is a good module? That's hard to say. What is good code? That's also hard to say. "**Quality**" is not a well-defined term in computing... and especially not Perl. One man's Thing of Beauty is another's man's Evil Hack. Since we can't define quality, how do we write a program to assure it?

  Kwalitee: It looks like quality, it sounds like quality, but it's not quite quality."

  --CPAN Testing Service (quoting M. Schwern)

- Kwalitee is measurable but inexact quality

Motivation: KISS

- Keep it simple, stupid

- "Problem 5 describes a class exercise that I graded on programming style. Most students turned in **one-page solutions** and received mediocre grades. Two students who had spent the previous summer on a large software development project turned in beautifully documented **five-page programs**, broken into a dozen procedures, each with an elaborate heading. They received **failing grades**. My program worked in **five lines of code**, and their **inflation factor of sixty** was too much for a passing grade. When they complained that they were employing standard software engineering tools, I should have quoted Pamela Zave: **'The purpose of software engineering is to control complexity, not to create it.'**"

  -- Jon Bentley: "Programming Pearls" (p.123)

Invenio Development: Developer Community

- Invenio source code base:
  - 35+ modules
  - 330,000+ lines of code
- Invenio developer community:
  - ~100 authors and contributors since 2002
  - Many short-term students
- In 2012:
  - 1072 commits from 51 developers and contributors

Python: Why Python?

- Easy to read and understand, pseudo code like
- Its abstraction results directly in higher software quality
- Large community

Python

- Example of character replacement:

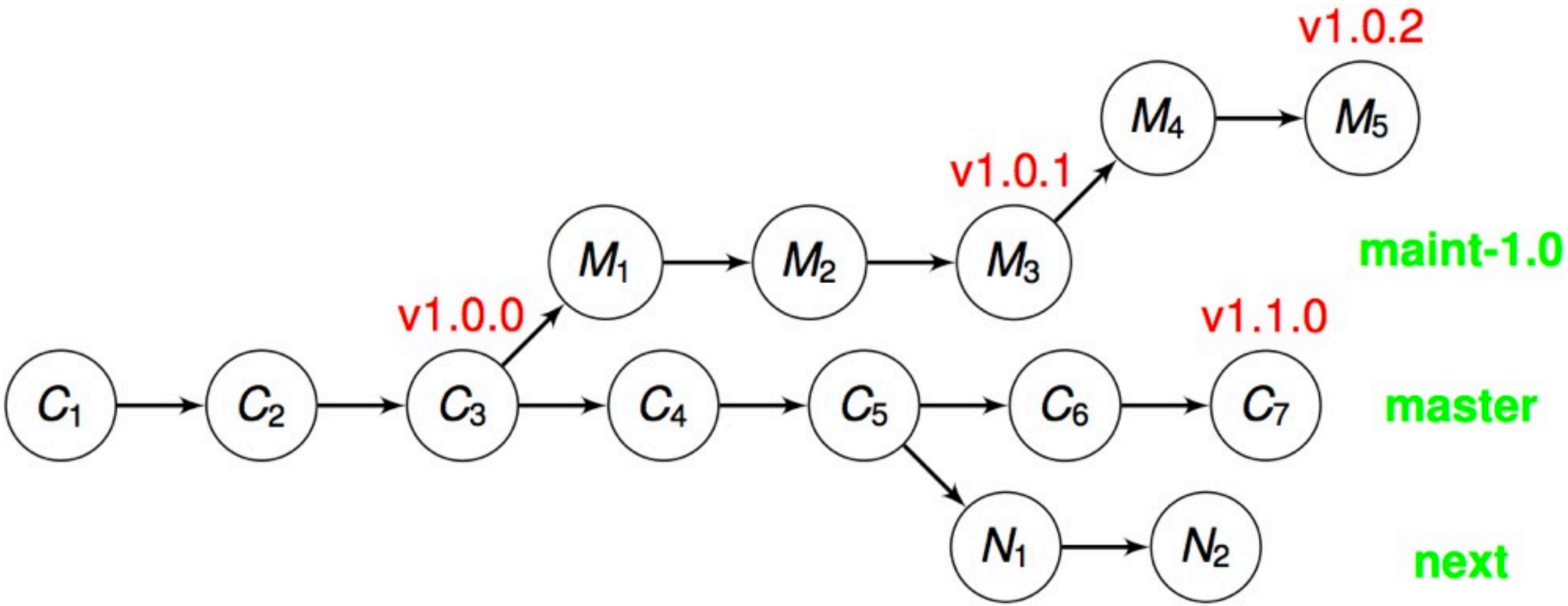
```
CFG_SOLR_INVALID_CHAR_RANGES  =  ((0,8,'  '),(11,12,'  '),(14,31,'  '),
(55296,57343,''),(65534, 65535,''))

def replace_invalid_solr_characters(utext)
    def replace(c):
        o = ord(c)
        for lhs, rhs, repl in CFG_SOLR_INVALID_CHAR_RANGES:
            if lhs <= o <= rhs:
                return repl
        return c

    utext_elements = map(replace, utext)
    return ''.join(utext_elements)
```

- In Java?

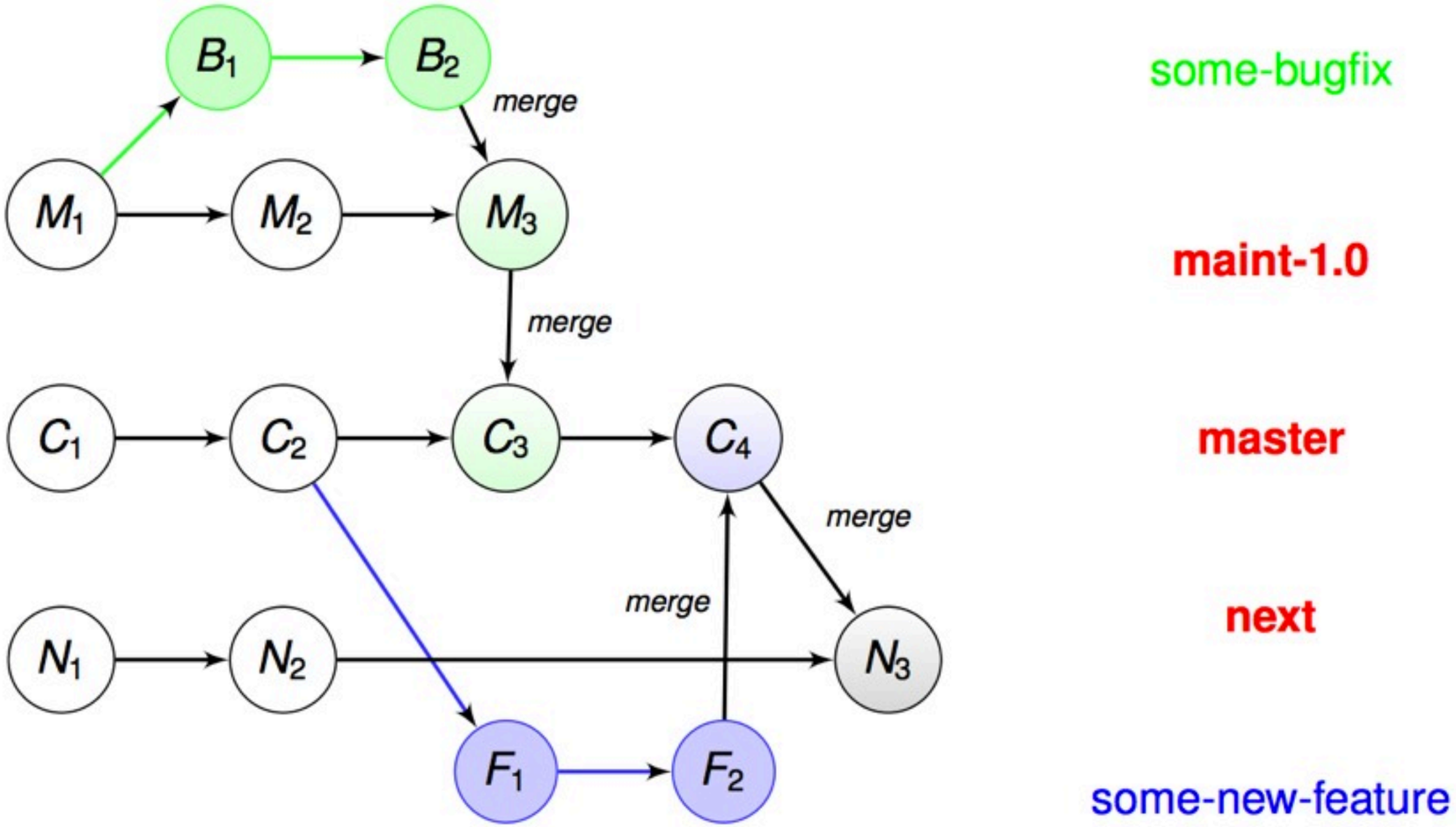
Git

- Software quality depends on effective source code management
- Git is a distributed version control system
- Good for offline development
- Powerful branching/merging
- Shared push vs. pull on demand

Git: Branches

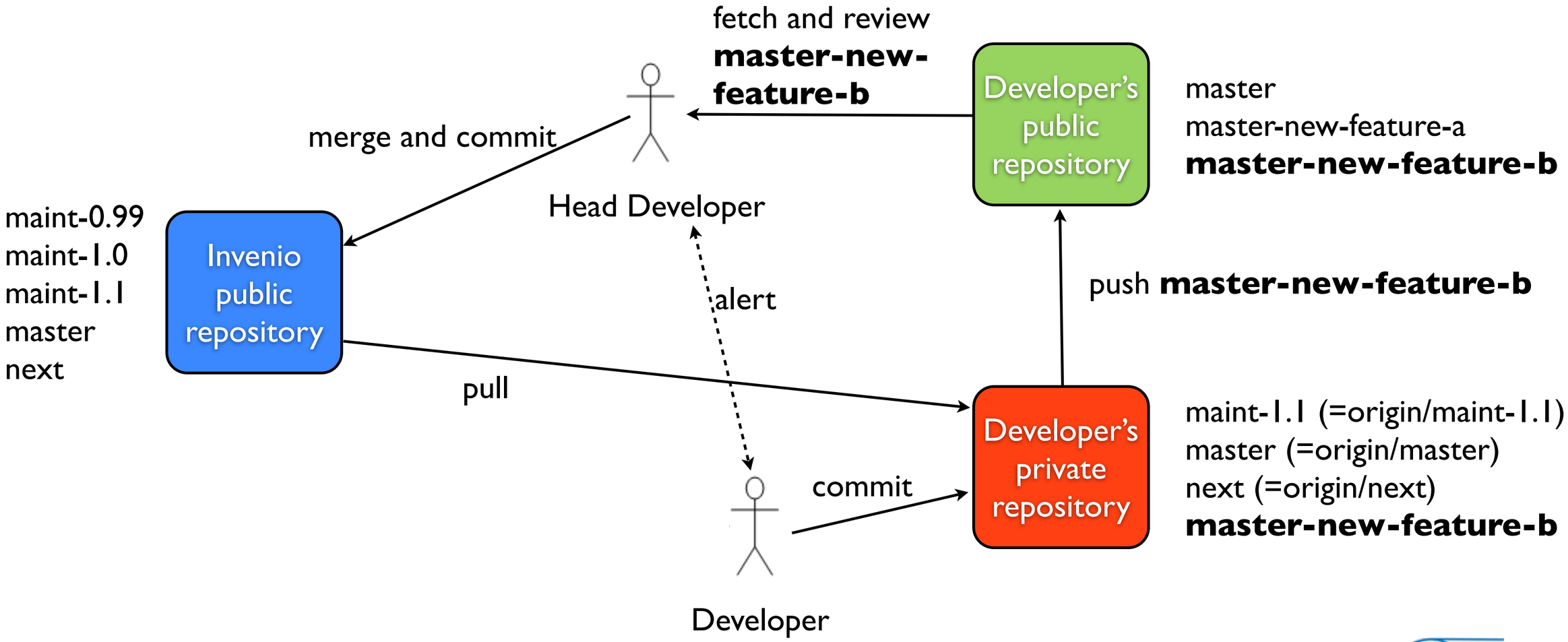


Git: Branches

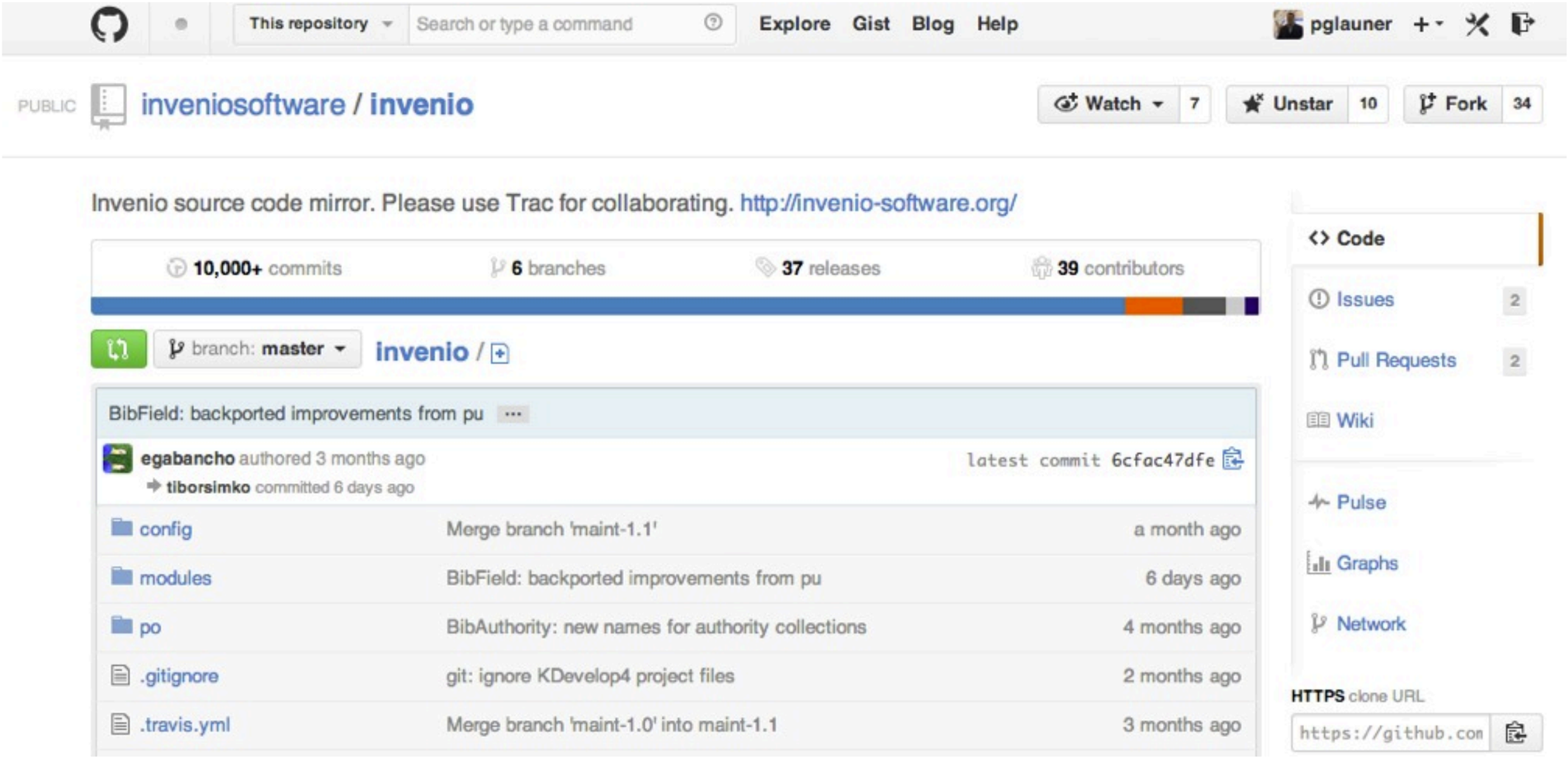


P. Glauner: Software Quality: A Python Case Study

Git: Workflow



Git: Migrating to Github



Git: Migrating to Github

This repository | Search or type a command | Explore | Gist | Blog | Help | pglauner

PUBLIC inveniosoftware / invenio

Watch 7 | Unstar 10 | Fork 34

All requests 2 | Open | Closed | Sort: Newest | New pull request

Yours

Find a user...

arcolife 1

GiorgosPa 1

fixes webaccess-api.webdoc grammatical mistake #2

Came across this while reading docs at http://cds.cern.ch/help/hacking/webaccess-api : The line ...

by arcolife a month ago | arcolife:master | 1 comment

errorlib.py: Modified register_exception function #1

Do not report KeyboardInterrupt exception fixes #1701

by GiorgosPa 2 months ago | GiorgosPa:KeyboardInterupt

Keyboard shortcuts available

Testing

- We *want* to avoid bugs
- "Program testing can be used to show the presence of bugs, but never to show their absence" (E. Dijkstra)
- We *can* reduce the amount of bugs
- Automated tests that can be run continuously
- **Test-driven development** when appropriate

Testing: Unit Tests

- Test of functionality
- Use of the `unittest` Python module
- ~600 unit tests
- Example `solrutils_unit_tests.py`:

```
class TestReplaceInvalidCharacters(unittest.TestCase):
    """Test for removal of invalid Solr characters and control characters."""

    def test_replace_control_characters(self):
        """solrutils - replacement of control characters"""
        self.assertEqual(u'abc \nde',replace_invalid_solr_characters(u'abc\u0000\nde'))
        self.assertEqual(u'abc \nde',replace_invalid_solr_characters(u'abc\u0003\nde'))
        self.assertEqual(u'abc \nde',replace_invalid_solr_characters(u'abc\u0008\nde'))
```

Testing: Regression Tests

- Regression/functional tests use a demo site and demo data ("Atlantis Institute of Fictive Science")

- Example `solrutils_regression_tests.py`:

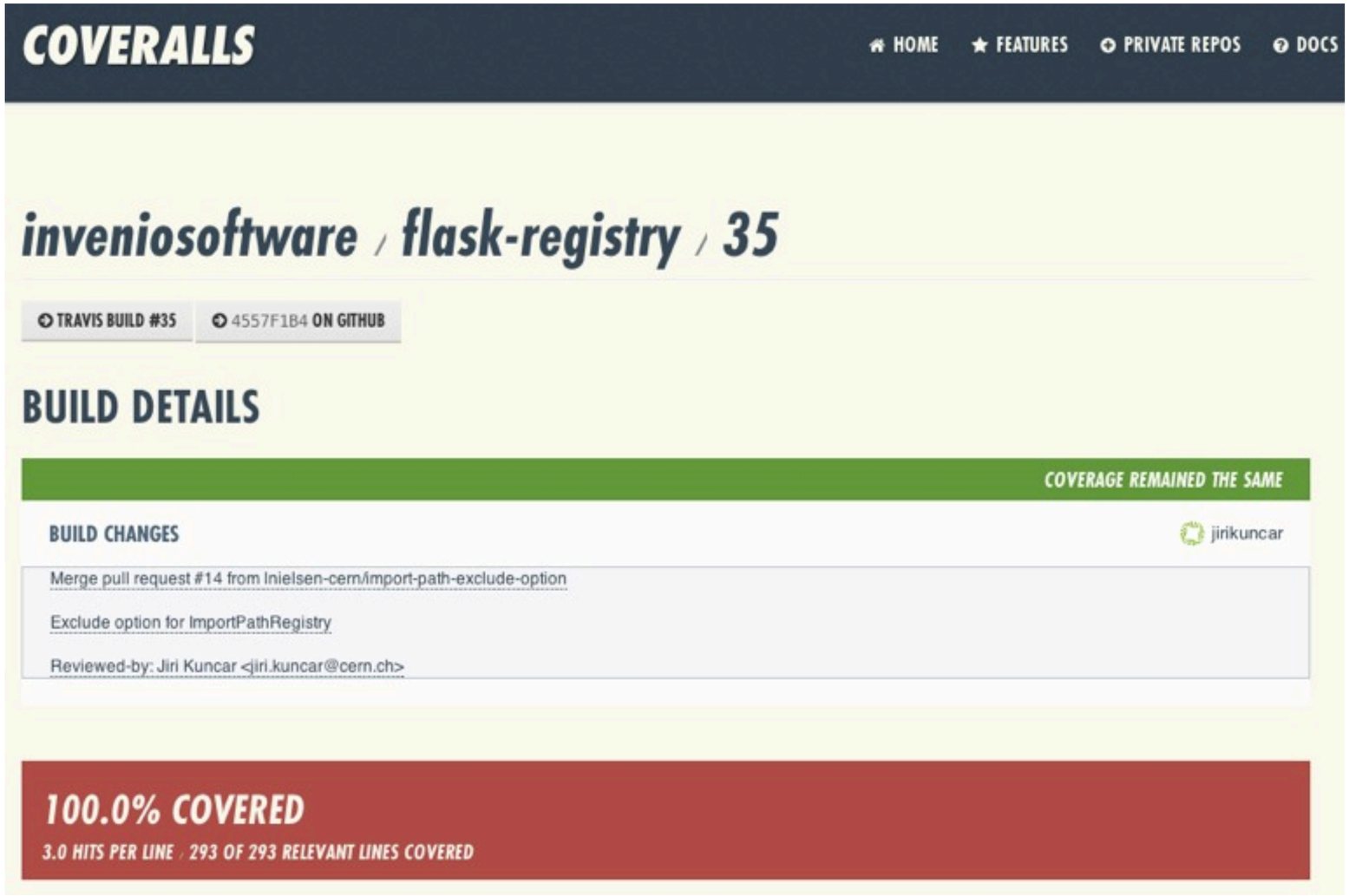
```
HITSETS = {
    'Willnotfind': intbitset.intbitset([]),
    'higgs': intbitset.intbitset([47,48,51,52,55,56,58,68,79,85,89,96]),
    '"higgs boson"': intbitset.intbitset([55, 56]),
}

class TestSolrSearch(unittest.TestCase):

    def _get_res(self, query, index='fulltext'):
        return solr_get_bitset(index, query)

    def test_get_bitset(self):
        self.assertEqual(HITSETS['Willnotfind'],self._get_res('Willnotfind'))
        self.assertEqual(HITSETS['higgs'],self._get_res('higgs'))
        self.assertEqual(HITSETS['"higgs boson"'],self._get_res('"higgs boson"'))
```

Testing: Web Tests

- Run tests in a browser
- Use of **Selenium** with Firefox/Chrome
- Example `websearch_web_tests.py`:

```
class InvenioWebSearchWebTests(InvenioWebTestCase):
    """WebSearch web tests."""

    def test_search_ellis(self):
        """websearch - web test search for ellis"""
        self.browser.get(CFG_SITE_URL)
        p = self.browser.find_element_by_name("p")
        p.send_keys("ellis")
        p.submit()
        self.page_source_test(expected_text=[
            'Thermal conductivity of dense quark matter ' + \
            'and cooling of stars'])
```

Testing: Coverage



P. Glauner: Software Quality: A Python Case Study

Development Scripts

- The Invenio development scripts repository contains a collection of scripts which are useful when hacking on Invenio
- https://github.com/tiborsimko/invenio-devscripts
- Allow to unite and share development practices
- Contain also QA related scripts

Development Scripts: Check Kwalitee

- Detects the most common code kwalitee problems:
  `$ invenio-check-kwalitee <OPTION> source_file.py`

- `<OPTION>` includes among others:

  - `--stats` generate kwalitee summary stats (**metrics**)
  - `--check-errors` check **Python errors**
  - `--check-indentation` check **Python code indentation**
  - `--check-pep8` check PEP8 compliance (**code style**)
  - `--check-sql` check **SQL queries**

Development Scripts: Check Branch and Retest Demo Site

- Checks the overall code kwalitee changed in a branch with respect to a reference branch:
  `$ invenio-check-branch master new-feature`

- Should be run on feature branches before every merge request

- Launches all unit/regression/web test suite cases on an installation:
  `$ invenio-retest-demo-site`

- Is useful to see whether a branch did not accidentally break some tests

Continuous Integration

- Use of build servers, e.g. Jenkins
- Daily build of code vs. on-demand by developer (for every pull request)
- Automatic execution of development scripts and tests
- Reporting of errors

All

S	W	Name ↓	Last Success	Last Failure	Last Duration
		inspire-master	28 days (#32)	13 hr (#60)	57 min
		invenio-maint-1.0	4 days 16 hr (#52)	16 hr (#56)	1 hr 6 min
		invenio-maint-1.1	5 days 16 hr (#47)	16 hr (#52)	31 min
		invenio-master	3 days 14 hr (#81)	14 hr (#84)	32 min
		invenio-next	N/A	16 hr (#65)	21 min
		invenio-ondemand	5 days 11 hr (#3)	7 hr 22 min (#5)	2 min 14 sec

Conclusions

- Software quality is important for a software product's success
- Invenio project case study
- Python's abstraction directly results in a higher software quality
- Version control to keep track of changes
- Development scripts to unite and share QA practices
- Continuous integration building software and executing automated tests

Prospects

- Security tests
- Project and test management
- Build boxes (vagrant)
- ...

Q&A

References

- http://invenio-software.org/
- https://github.com/
- T. Simko. Invenio Technology. http://tiny.cc/haoouw
- T. Simko. Invenio Development Practises. http://tiny.cc/xcoouw