

# Deploying Continuous Integration with XIFI

Gerrit, “Future Internet”, PPP, and KDE

Jan Kundrát

[jan.kundrat@cesnet.cz](mailto:jan.kundrat@cesnet.cz)



September 7<sup>th</sup>, 2014

Who am I?

# KPhotoAlbum







Trojita, a fast Qt IMAP e-mail client

## In a Nutshell, Trojitá...

... has had 5,618 commits made by 59 contributors  
representing 72,418 lines of code

... is mostly written in C++  
with an average number of source code comments

... has a well established, mature codebase  
maintained by a large development team  
with decreasing Y-O-Y commits

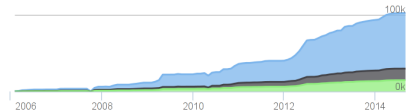
... took an estimated 18 years of effort (COCOMO model)  
starting with its first commit in February, 2006  
ending with its most recent commit 23 days ago

## Languages



C++	75%	C	8%
TeX/LaTeX	5%	11 Other	12%

## Lines of Code



## Managing Contributions

- Number of contributors grows
- Some patches are crappy
- Shuffling them around manually takes time
- All people make mistakes
  - Let computers take care of as much stuff as possible

# Code Review

- ReviewBoard
  - Has been used by KDE for years
  - Not integrated with CI
  - Patches not accessible through git
- Gerrit
  - Used by Qt, Chromium, Android, OpenStack, WikiMedia,...
  - Pushes go to `refs/for/master` instead of `master`
  - Stuff available at `refs/changes/03/3/1`





### Change 3 - Abandoned

Move PartWidgetFactory to UiUtils and make it not QWidget specific

Change-Id: I3e407b3b07f6a91bae5af0067a5b0195dc77071f  
REVIEW: 119818

#### Reply...

[Patch Sets \(1/1\)](#) [Download](#)

Owner [Jan Kůdrát](#)  
Reviewers [Caspar Schütjser](#)  
[Karan Luthra](#)  
Project [trojita](#)   
Branch [master](#)  
Topic  
Updated 7 hours ago

#### Related Changes (1)

► Move PartWidgetFactory to UiUtils and make it not QW...

Code-Review +2  
[Caspar Schütjser](#)  
[Karan Luthra](#)  
-2 [Jan Kůdrát](#)

Author [Boren Zhang <bobo1993324@gmail.com>](#) Aug 19, 2014 1:34 PM  
Committer [Jan Kůdrát <jkt@flaska.net>](#) Aug 21, 2014 11:32 PM  
Commit 08501065cfa97b79f3701008e151e79133b3460  
Change-Id I3e407b3b07f6a91bae5af0067a5b0195dc77071f

#### Files

[Open All](#) Diff against: [Base](#)

	File Path	Comments	Size
►	Commit Message		
	CMakeLists.txt	2	
	src/Gui/AttachmentView.h	3	
	src/Gui/LoadablePartWidget.cpp	14	
	src/Gui/LoadablePartWidget.h	11	
	src/Gui/MessageView.cpp	11	
	src/Gui/MessageView.h	6	
	src/Gui/PartWidget.cpp	39	
	src/Gui/PartWidget.h	19	
D	src/Gui/PartWidgetFactory.cpp	230	
D	src/Gui/PartWidgetFactory.h	78	
A	src/Gui/PartWidgetFactoryVisitor.cpp	89	
A	src/Gui/PartWidgetFactoryVisitor.h	56	
A	src/UiUtils/PartLoadingOptions.h	47	
A	src/UiUtils/PartVisitor.h	61	
A	src/UiUtils/PartWalker.h	247	
		+556, -357	

#### History

[Expand All](#)

Jan Kůdrát	Uploaded patch set 1.	Aug 21 11:32 PM
Karan Luthra	Patch Set 1: Code-Review+2	Aug 22 11:01 AM
Caspar Schütjser	Patch Set 1: Code-Review+2 I approve.	Aug 22 12:02 PM
Jan Kůdrát	Patch Set 1: Code-Review-1 Testing, let's see if this mail makes it through.	Aug 22 12:27 PM
Sysadmin Testing...	Patch Set 1: Code-Review+1 This is a regular KDE user, so I can add just +1/-1. Let's see what happens when I untick that "send mail" che...	Aug 22 12:33 PM
Jan Kůdrát	Removed the following approvals: * Code-Review+1 by Sysadmin Testing Account <null@kde.org>	Aug 22 12:36 PM
Jan Kůdrát	Patch Set 1: Code-Review-2 As we're now in the phase where approvals on this Gerrit instance will get propagated to git.kde.org, let's kill t...	11:16 AM
Jan Kůdrát	Abandoned	11:16 AM

## trojita / src/Gui/LoadablePartWidget.cpp

## Patch Set Base 1

```
... skipped 22 common lines ...
23 #include "Gui/MessageView.h" // so that the compiler knows that it's a QObject
24 #include "Inap/Model/ItemRoles.h"
25 #include "Utils/Formatting.h"
26
27 #include <QPushButton>
28
29 namespace Gui
30 {
31
32 LoadablePartWidget::LoadablePartWidget(QWidget *parent, Inap::Network::MsgPartNetAccessManager *manager, const QModelIndex &part,
33                                     PartWidgetFactory *factory, int recursionDepth,
34                                     const QUrl::PartWalkOptions loadingMode) :
35     QObject(parent), manager(manager), partIndex(part), m_factory(factory),
36     m_recursionDepth(recursionDepth), m_loadingMode(loadingMode), realPart(0), loadButton(0), m_loadOnShow(false)
37 {
38     Q_ASSERT(partIndex.isValid());
39
40     QString mimeType = partIndex.data(Inap::Mailbox::RolePartMimeType).toString().toLower();
41
42     if ((loadingMode & PartWidgetFactory::PART_IS_HIDDEN) ||
43         (loadingMode & PartWidgetFactory::PART_IGNORE_CLICKTHROUGH) ||
44         partIndex.data(Inap::Mailbox::RoleIsPatched).toBool()) {
45         m_loadOnShow = true;
46     }
47     else {
48         loadButton = new QPushButton(tr("Load %1 (%2)").arg(partIndex.data(Inap::Mailbox::RolePartMimeType).toString()),
49                                     this);
50         connect(loadButton, SIGNAL(clicked()), this, SLOT(loadClicked()));
51         addWidget(loadButton);
52     }
53
54     ... skipped 5 common lines ...
55
56     loadButton->setEnabled(false);
57     return;
58 }
59
60 if (loadButton) {
61     loadButton->deleteLater();
62     loadButton = 0;
63 }
64
65 // We have to disable any flags which might cause recursion here
66 realPart = m_factory->create(partIndex, m_recursionDepth + 1,
67                             (loadingMode & PartWidgetFactory::PART_IGNORE_CLICKTHROUGH
68                              ? PartWidgetFactory::PART_IGNORE_LOAD_OR_SHOW) : PartWidgetFactory::PART_IS_HIDDEN);
69
70 addWidget(realPart);
71 setCurrentIndex(i);
72 }
73
74 QString LoadablePartWidget::quoteMe() const
75 {
76     AbstractPartWidget *part = dynamic_cast<AbstractPartWidget*>(realPart);
77     return part ? part->quoteMe() : QString();
78 }
79
80 ... skipped 16 common lines ...
81
```

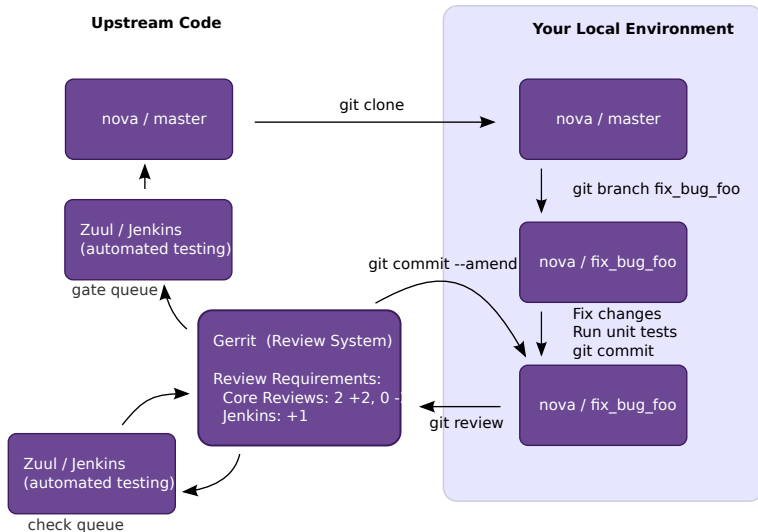
## Patch Set 1

```
... skipped 22 common lines ...
23 #include "Gui/MessageView.h" // so that the compiler knows that it's a QObject
24 #include "Inap/Model/ItemRoles.h"
25 #include "Utils/Formatting.h"
26
27 #include <QPushButton>
28
29 namespace Gui
30 {
31
32 LoadablePartWidget::LoadablePartWidget(QWidget *parent, Inap::Network::MsgPartNetAccessManager *manager, const QModelIndex &part,
33                                     QUrl::PartWalkOptions loadingMode,
34                                     const QUrl::PartWalkOptions loadingMode) :
35     QObject(parent), manager(manager), partIndex(part), m_factory(factory),
36     m_recursionDepth(recursionDepth), m_loadingMode(loadingMode), realPart(0), loadButton(0), m_loadOnShow(false)
37 {
38     Q_ASSERT(partIndex.isValid());
39
40     QString mimeType = partIndex.data(Inap::Mailbox::RolePartMimeType).toString().toLower();
41
42     if ((loadingMode & QUrl::PART_IS_HIDDEN) ||
43         (loadingMode & QUrl::PART_IGNORE_CLICKTHROUGH) ||
44         partIndex.data(Inap::Mailbox::RoleIsPatched).toBool()) {
45         m_loadOnShow = true;
46     }
47     else {
48         loadButton = new QPushButton(tr("Load %1 (%2)").arg(partIndex.data(Inap::Mailbox::RolePartMimeType).toString()),
49                                     this);
50         connect(loadButton, SIGNAL(clicked()), this, SLOT(loadClicked()));
51         addWidget(loadButton);
52     }
53
54     ... skipped 5 common lines ...
55
56     loadButton->setEnabled(false);
57     return;
58 }
59
60 if (loadButton) {
61     loadButton->deleteLater();
62     loadButton = 0;
63 }
64
65 // We have to disable any flags which might cause recursion here
66 realPart = m_factory->walk(partIndex, m_recursionDepth + 1,
67                             (loadingMode & QUrl::PART_IGNORE_CLICKTHROUGH
68                              ? QUrl::PART_IGNORE_LOAD_OR_SHOW) : QUrl::PART_IS_HIDDEN);
69
70 addWidget(realPart);
71 setCurrentIndex(i);
72 }
73
74 QString LoadablePartWidget::quoteMe() const
75 {
76     AbstractPartWidget *part = dynamic_cast<AbstractPartWidget*>(realPart);
77     return part ? part->quoteMe() : QString();
78 }
79
80 ... skipped 16 common lines ...
81
```

```

60 Patch Set (-) Base 1
61 Patch Set (+) 1
diff --git a/src/Gui/LoadablePartWidget.cpp b/src/Gui/LoadablePartWidget.cpp
index f5a25ba..a623996 100644
--- a/src/Gui/LoadablePartWidget.cpp
+++ b/src/Gui/LoadablePartWidget.cpp
@@ -23,21 +23,21 @@
23 #include "Gui/MessageView.h" // so that the compiler knows that it's a QObject
24 #include "Imap/Model/ItemRoles.h"
25 #include "UUtils/Formatting.h"
26
27 #include <QPushButton>
28
29 namespace Gui
30 {
31
32 LoadablePartWidget::LoadablePartWidget(QWidget *parent, Imap::Network::MsgPartNetAccessManager *mana
33 ger, const QModelIndex &part,
34     PartWidgetFactory *factory, int recursionDepth,
35     const PartWidgetFactory::PartLoadingOptions loadingMode);
36     UUtils::PartWalker *factory, int recursionDepth,
37     const UUtils::PartLoadingOptions loadingMode);
38     QStackedWidget(parent), manager(manager), partIndex(part), m_factory(factory),
39     m_recursionDepth(recursionDepth), m_loadingMode(loadingMode), realPart(0), loadButton(0), m_load
40     OnShow(false)
41 {
42     Q_ASSERT(partIndex.isValid());
43
44     QString mimeType = partIndex.data(Imap::Mailbox::RolePartMimeType).toString().toLower();
45
46     if ((loadingMode & PartWidgetFactory::PART_IS_HIDDEN) ||
47         (loadingMode & PartWidgetFactory::PART_IGNORE_CLICKTHROUGH) ||
48         (loadingMode & UUtils::PART_IS_HIDDEN) ||
49         (loadingMode & UUtils::PART_IGNORE_CLICKTHROUGH) ||
50         partIndex.data(Imap::Mailbox::RoleIsFetched).toBool()) {
51         m_loadOnShow = true;
52     } else {
53         loadButton = new QPushButton(tr["Load %1 (%2)"].arg(partIndex.data(Imap::Mailbox::RolePartM
54 imeType).toString(),
55     UUtils::Formatting::prettySize(partIndex.data(Imap::Mailbox::R
56 olePartOctets).toInt()),
57     this);
58         connect(loadButton, SIGNAL(clicked()), this, SLOT(loadClicked()));
59         addWidget(loadButton);
60     }
61 }
62
63 @ -59,23 +59,23 @@
64 loadButton->setEnabled(false);
65
66 }
67 return;
68
69 if (loadButton) {
70     loadButton->deleteLater();
71     loadButton = 0;
72 }
73
74 // We have to disable any flags which might cause recursion here
75 realPart = m_factory->create(partIndex, m_recursionDepth + 1,
76     (m_loadingMode | PartWidgetFactory::PART_IGNORE_CLICKTHROUGH
77     | PartWidgetFactory::PART_IGNORE_LOAD_ON_SHOW) ^ PartWidgetFactory
78     :PART_IS_HIDDEN);
79 realPart = m_factory->walk(partIndex, m_recursionDepth + 1,
80     (m_loadingMode | UUtils::PART_IGNORE_CLICKTHROUGH
81     | UUtils::PART_IGNORE_LOAD_ON_SHOW) ^ UUtils::PART_IS_HIDDEN);
82
83 addWidget(realPart);
84 setCurrentIndex(1);
85
86
87 QString LoadablePartWidget::quoteMe() const
88 {
89     AbstractPartWidget *part = dynamic_cast<AbstractPartWidget*>(realPart);
90     return part ? part->quoteMe() : QString();
91 }
92
93
94

```



...but that needs plenty of CPU cycles



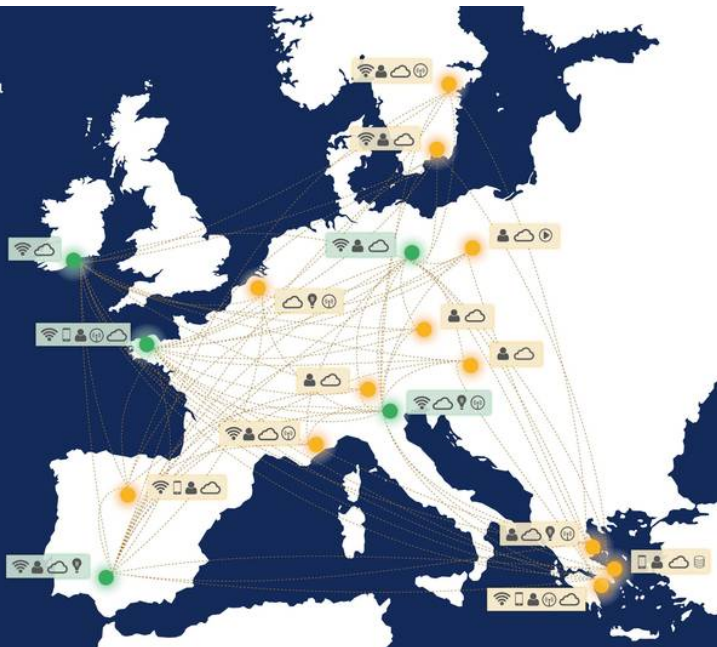
*XIFI is a project of the European Public Private Partnership on Future Internet (FI-PPP) programme.*

*XIFI is the project responsible for the capacity building part of the programme.*

– <https://www.fi-xifi.eu/about-xifi/what-is-xifi.html>

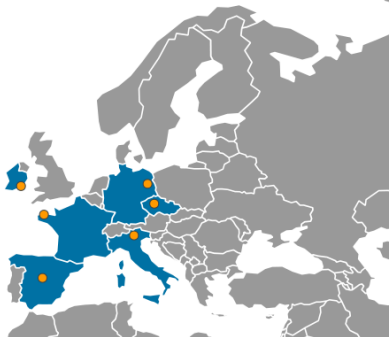
- Origin nodes
- Nodes selected by Open Call

- Wireless access
- Customer devices
- User community
- Cloud
- Energy measure
- Sensor network
- Open data
- Media services





6 regions



1584 core



4896581 Mb



40252 Gb



## 80 Million € for SMEs

*Through open calls, the third phase calls for SMEs and Web entrepreneurs to develop highly innovative services and applications.*

*The 16 selected “accelerator projects” will publish Open Calls for the distribution of grants to SMEs and Web entrepreneurs as of September 2014. SMEs and Web entrepreneurs may submit proposals to one (or perhaps more) of these Open Calls, in accordance with the requirements defined by the projects launching the Open Calls.*

– <http://www.fi-ppp.eu/fi-ppp-phase-3-projects-and-their-calls-for-smes/>

## KDE's Gerrit

- Gerrit at <https://gerrit.vesnický.cesnet.cz/>
  - Use KDE Identity for login
- Approved changes are pushed to KDE's git
  - Everyone can submit a patch
  - Any KDE developer can approve anything
  - No countermeasures against self-approvals
- Use [dummy.git](#) for testing
- **Ask me if you want your project in there**
  - In person here at Academy
  - [jkt](#) on [#kde-sysadmin](#) at Freenode
  - [jkt@flaska.net](mailto:jkt@flaska.net)

# Future Plans

- Set up pre-merge CI
- Don't bother reviewers until the patch at least builds
- Fancy stuff
  - IRC hooks, Bugzilla, . . .
- **Do you want to help?**
  - Write tutorials
  - Report bugs
  - Become a Gerrit admin

## “Social Issues”

Do we want to limit approvals to project maintainers?

Is the current situation where everyone can commit and push everywhere a goal, or a way towards some other goal?

What is the **actual goal**?

Thanks for listening

Questions?