CS 445 Final Report

Understand Bitcoin

Harry Hughes

Milan Patel

Cody Williams

12-9-2015

Introduction

The purpose of this project was to gain insight into the Bitcoin system. Our specific focus was on the development history of the bitcoin project, although other aspects of Bitcoin were also investigated for context. A wealth of data was available for analysis because the entire Bitcoin project is open-source and publicly accessible on GitHub. Initially, the goal of the project was to determine the locations of participants in Bitcoin transactions worldwide as well as the nature of goods and services purchased. This initial goal was eventually abandoned due to difficulties in obtaining data. The data we gathered proved to be a gold mine of useful information, and many interesting conclusions were drawn from our results.

Project Narrative

Initially, the goal of the project was to determine where the participants in bitcoin transactions were located and what they might be purchasing. However, it quickly became clear that this was impossible for several reasons. When a new transaction is posted to the bitcoin network, only the IP address of the node that broadcast the transaction is available. Masking an IP address is quite easy, and most bitcoin users do that to preserve anonymity. Determining what goods or services were purchased is an immensely complicated problem as well and we did not have enough time to do it. At this point, the focus of the project shifted.

We decided to try and examine the changes made to the project itself to see what information we could find. We began by downloading the entire repository tree to a local machine. Using a Python module called “git”, we were able to view every change made to every file in the project and who made it. We decided to measure two main things about the Bitcoin project. First, we wanted to see what types of files had the most development going into them. This was accomplished by simply separating the documented changes by file extension. The second analysis was a bit more involved. We wanted to observe the general trends that the project’s development followed. This was accomplished by making lists of the top files changed for each year along with the total number of changes made to them.

Technical Description

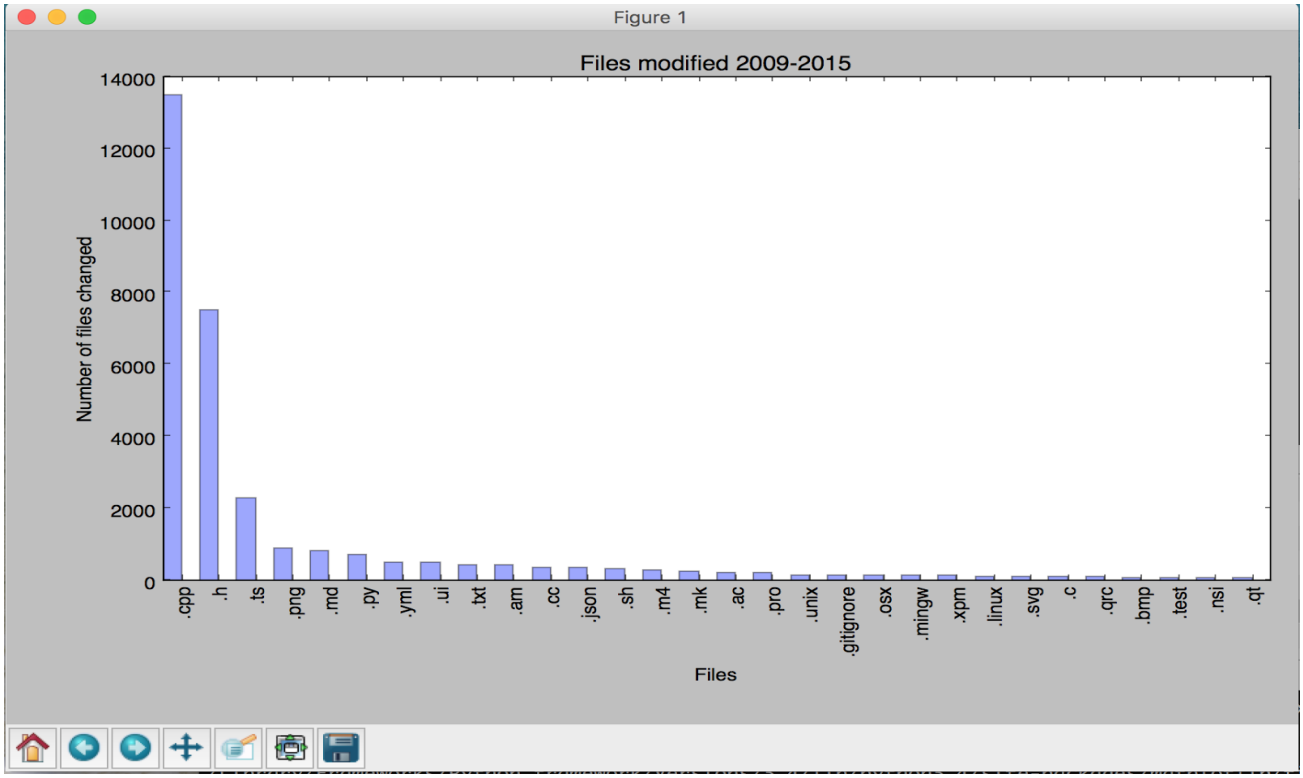
Our project was roughly divided into two phases: data acquisition and data analysis. All source code was written in Python, and all graphs were generated with Python as well. The data acquisition phase entailed gathering detailed information about the Bitcoin project’s development history and current content. Because the Bitcoin project is hosted on GitHub, all changes since the project’s inception are recorded in detail in the project’s git repository tree. This data was obtained by downloading the entire repo tree to a local machine and then utilizing Python’s “git” module to generate a text file of raw data viewable in our “understanding-bitcoin” repository called “output3.txt”. The script that generated this file is called “final_project.ipynb”. This script compiled a list of contributor’s names, the days on which they contributed, and the files they changed along

with all relevant timestamps. It also contained a list of all the files that were changed and how many changes were made for each. All data from the project's inception is included. It is also viewable on our "understanding-bitcoin" repository.

In the data analysis phase of the project, we conducted a meaningful and thorough analysis of the data. Our analysis data was formatted to easily be represented graphically, as shown below. The script responsible for generating this data is called "bitcoin_data.ipynb". It generated several text files of data that were converted into the graphs viewable in our "Results and Conclusions" section. The first file, "extensions.txt", contains the number of file changes made for each file extension. This provides insight into what languages are most represented in the project. The remainder of the files begin with a four digit year. These files contain a "leaderboard" for files for each year. Stated another way, for each year, each file is arranged in descending order of changes made to it. This provides excellent insight as to which files development was focused on for each year that Bitcoin has been an active GitHub project. Differences in the order of files from year to year, as well as differences in the total number of changes for individual ones points to shifts in development focus in the past several years.

Results and Conclusions

This section contains all of the graphs generated by our scripts. All source code for this project is viewable on our project page on GitHub. The first graph we generated presents a high level view of which types of files in the Bitcoin project have been worked on the most. The data for this graph was accumulated from July 2009 until November 2015.



The majority of development was done in C++. This makes a lot of sense considering the nature of the Bitcoin project. It must be fast as well as efficient, and needs little in the way of visualization or user interface. A few of the file extensions, however, are very unexpected and do not have any obvious explanation. For example, the third most common file types modified are ".ts" files. This is a file format normally used for storing video data. The next most common modified file types are ".png" picture files. We did not dig any deeper to see what these files might be used for in the project, but the next step of this project would be to see exactly what these files are used for.

The next set of graphs show more detailed information about Bitcoin development trends. Instead of grouping changes based on file extensions, changes are grouped based on the particular file. Files are arranged by frequency changed in descending order from 2011 to 2015. These graphs provide a very good picture of how the Bitcoin project has evolved over time. After 2011, project file "src/net.cpp" fell from the most modified file to be replaced by "main.cpp" for all subsequent years. Development on project file "init.cpp" increased dramatically from 2011 to 2013, as is evidenced by its increasing rank throughout this timespan. After 2013, development drops off to a degree, but it remains one of the most developed project files. This analysis can be applied to any of the files in the project.

The observed development trends from year to year can likely be explained by examining what was happening in the world and with Bitcoin during each year. The first step to determining the motivation behind the development trends is to understand what the purpose of each project file was. Due to the size and complexity of the code contained in these project files, we were unable to understand what they were doing exactly. However, given enough time to familiarize ourselves with the Bitcoin codebase, we are confident that a historical context for the observed development trends could be discovered.

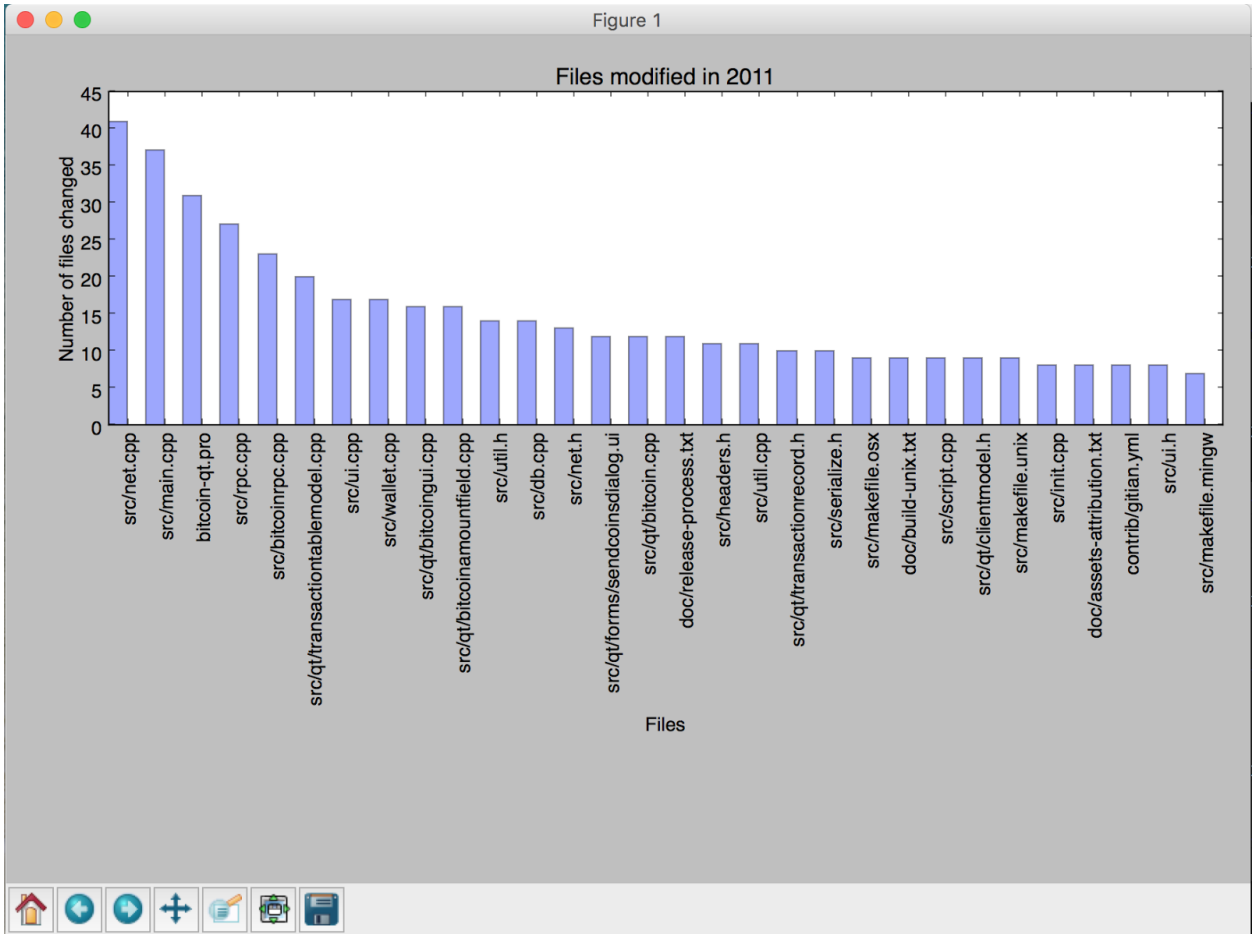


Figure 1
Files modified in 2011
Number of files changed
0
5
10
15
20
25
30
35
40
45
src/net.cpp
src/main.cpp
bitcoin-qt.pro
src/rpc.cpp
src/bitcoinrpc.cpp
src/qt/transactiontablemodel.cpp
src/ui.cpp
src/wallet.cpp
src/qt/bitcoingui.cpp
src/qt/bitcoinamountfield.cpp
src/util.h
src/db.cpp
src/net.h
src/qt/forms/sendcoinsdialog.ui
src/qt/bitcoin.cpp
doc/release-process.txt
src/headers.h
src/util.cpp
src/qt/transactionrecord.h
src/serialize.h
src/makefile.osx
doc/build-unix.txt
src/script.cpp
src/qt/clientmodel.h
src/makefile.unix
src/init.cpp
doc/assets-attribution.txt
contrib/gitian.yml
src/ui.h
src/makefile.mingw
Files

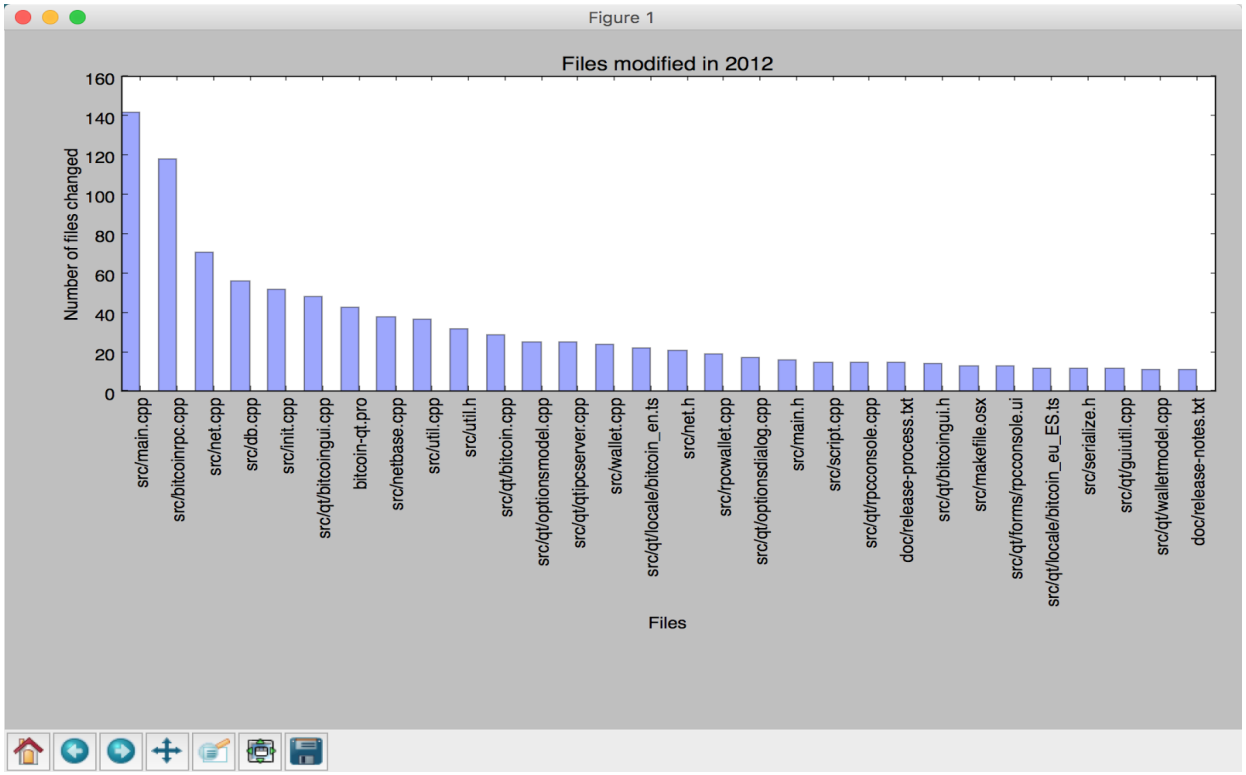


Figure 1
Files modified in 2012
Number of files changed
0
20
40
60
80
100
120
140
160
src/main.cpp
src/bitcoinrpc.cpp
src/net.cpp
src/db.cpp
src/init.cpp
src/qt/bitcoingui.cpp
bitcoin-qt.pro
src/netbase.cpp
src/util.cpp
src/util.h
src/qt/bitcoin.cpp
src/qt/optionsmodel.cpp
src/qt/qtipcserver.cpp
src/wallet.cpp
src/qt/locale/bitcoin_en.ts
src/net.h
src/rpcwallet.cpp
src/qt/optionsdialog.cpp
src/main.h
src/script.cpp
src/qt/rpcconsole.cpp
doc/release-process.txt
src/qt/bitcoingui.h
src/makefile.osx
src/qt/forms/rpcconsole.ui
src/qt/locale/bitcoin_eu_ES.ts
src/serialize.h
src/qt/guiutil.cpp
src/qt/walletmodel.cpp
doc/release-notes.txt
Files

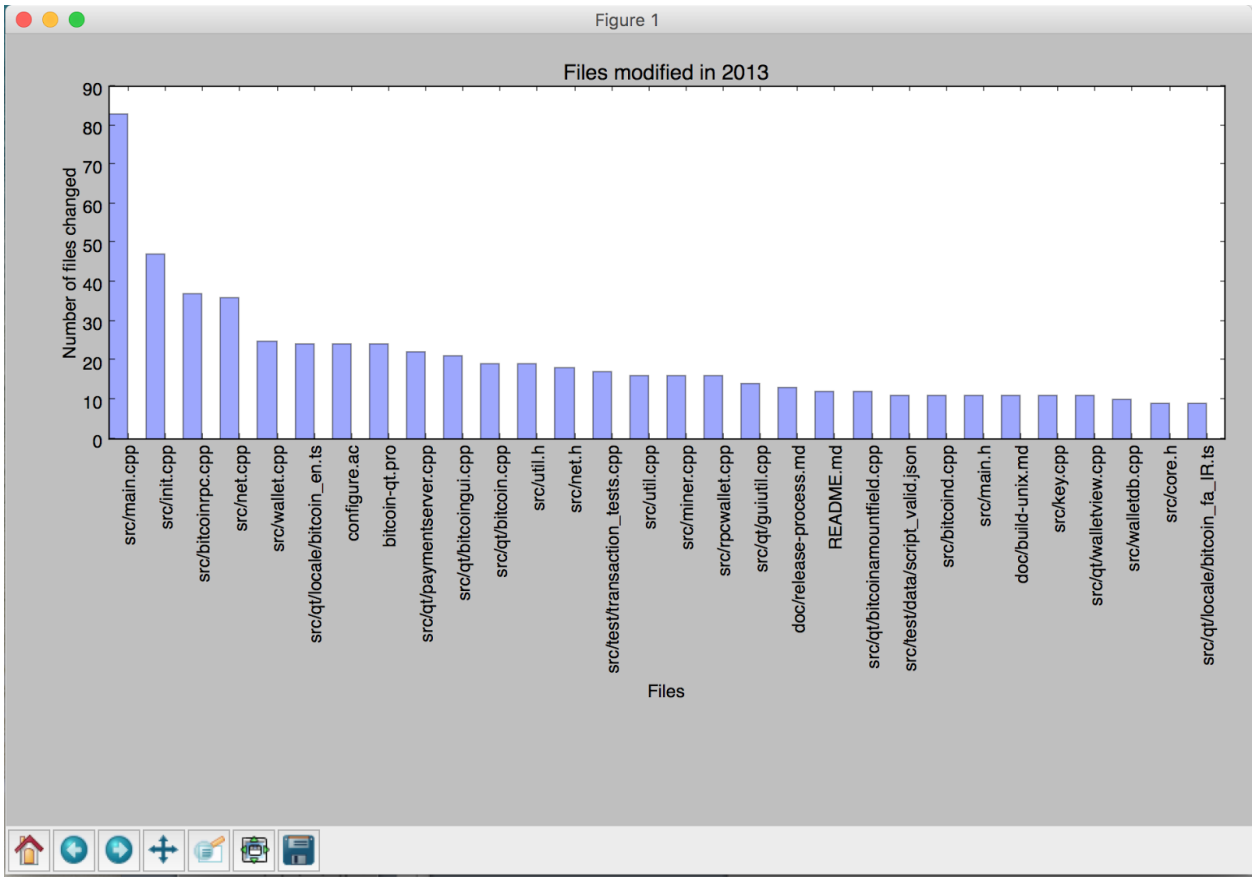


Figure 1
Files modified in 2013
Number of files changed
0
10
20
30
40
50
60
70
80
90
src/main.cpp
src/init.cpp
src/bitcoinrpc.cpp
src/net.cpp
src/wallet.cpp
src/qt/locale/bitcoin_en.ts
configure.ac
bitcoin-qt.pro
src/qt/paymentserver.cpp
src/qt/bitcoingui.cpp
src/qt/bitcoin.cpp
src/util.h
src/net.h
src/test/transaction_tests.cpp
src/util.cpp
src/miner.cpp
src/rpcwallet.cpp
src/qt/guiutil.cpp
doc/release-process.md
README.md
src/qt/bitcoinamountfield.cpp
src/test/data/script_valid.json
src/bitcoind.cpp
src/main.h
doc/build-unix.md
src/key.cpp
src/qt/walletview.cpp
src/walletdb.cpp
src/core.h
src/qt/locale/bitcoin_fa_IR.ts
Files

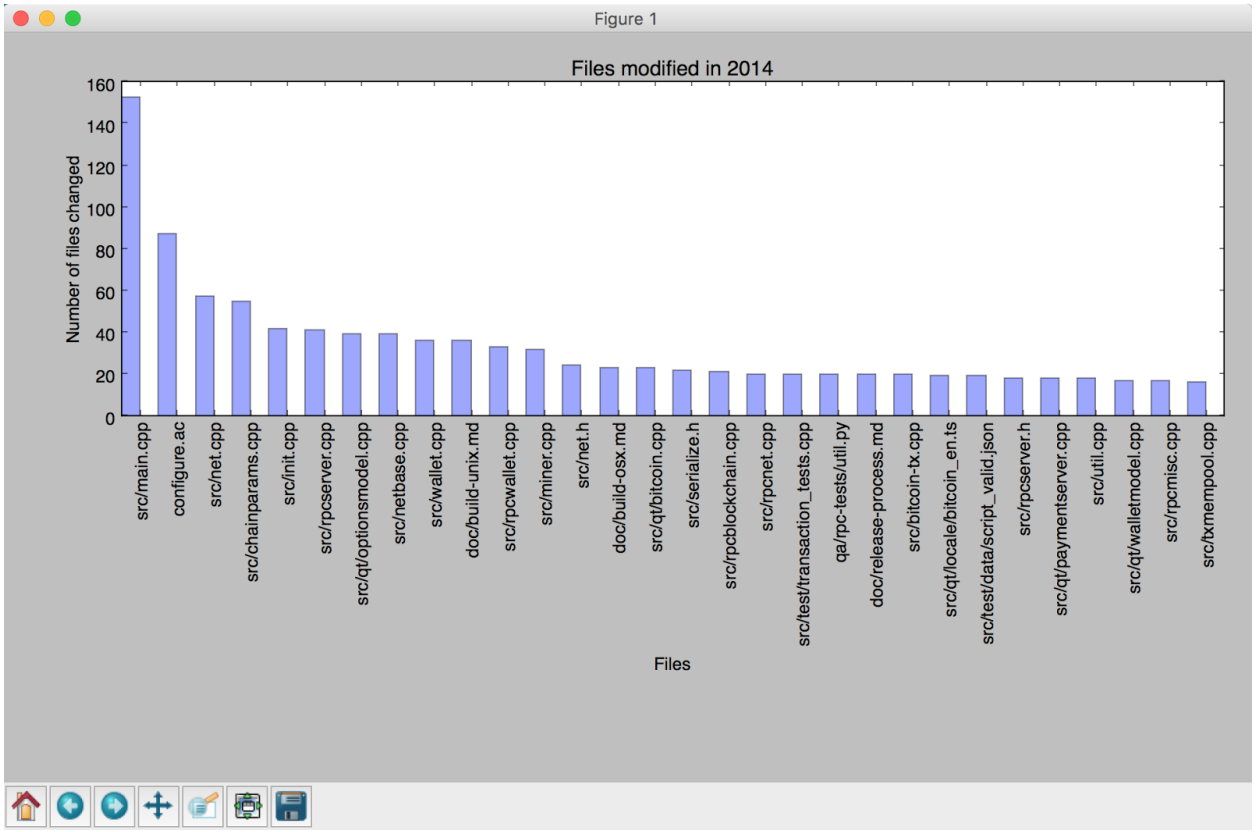


Figure 1
Files modified in 2014
Number of files changed
0
20
40
60
80
100
120
140
160
src/main.cpp
configure.ac
src/net.cpp
src/chainparams.cpp
src/init.cpp
src/rpcserver.cpp
src/qt/optionsmodel.cpp
src/netbase.cpp
src/wallet.cpp
doc/build-unix.md
src/rpcwallet.cpp
src/miner.cpp
src/net.h
doc/build-osx.md
src/qt/bitcoin.cpp
src/serialize.h
src/rpcblockchain.cpp
src/rpcnet.cpp
src/test/transaction_tests.cpp
qa/rpc-tests/util.py
doc/release-process.md
src/bitcoin-tx.cpp
src/qt/locale/bitcoin_en.ts
src/test/data/script_valid.json
src/rpcserver.h
src/qt/paymentserver.cpp
src/util.cpp
src/qt/walletmodel.cpp
src/rpcmisc.cpp
src/txmempool.cpp
Files

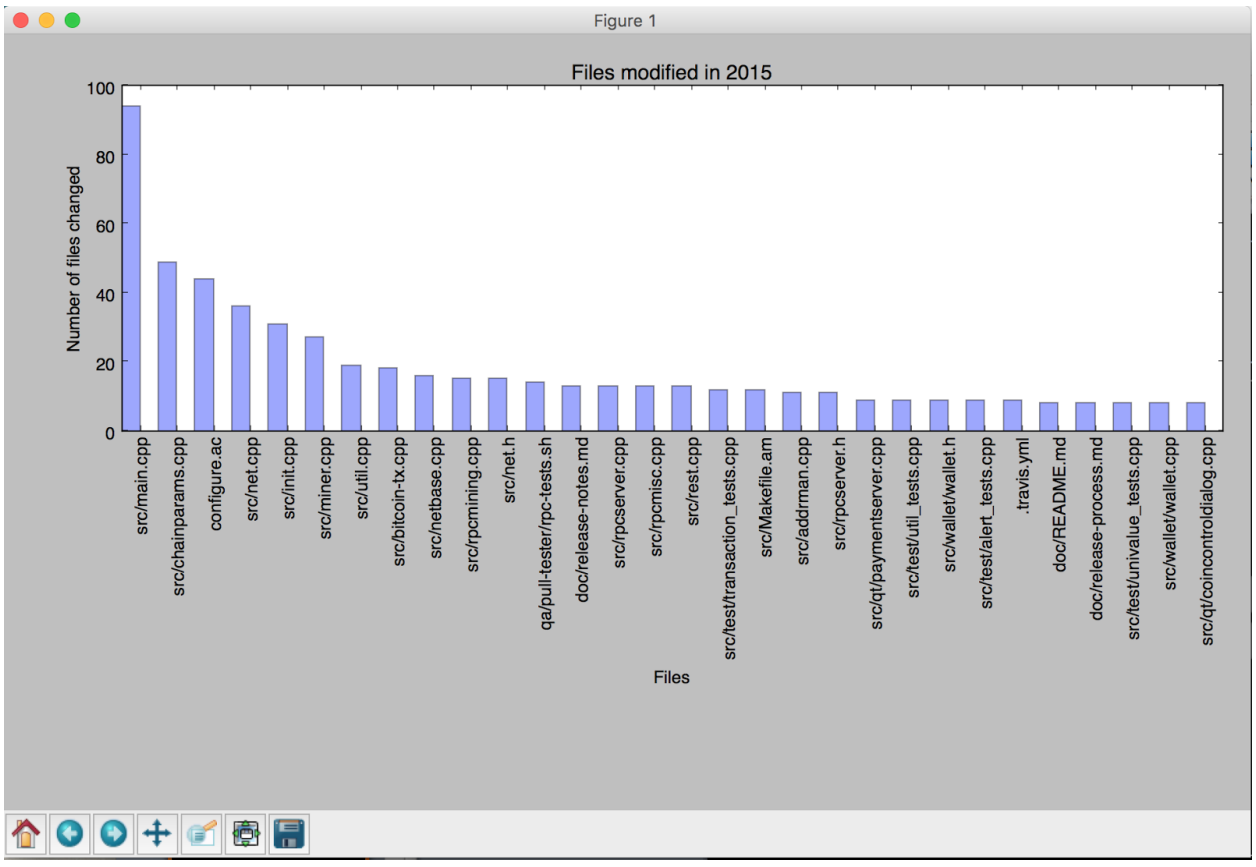


Figure 1
Files modified in 2015
Number of files changed
0
20
40
60
80
100
src/main.cpp
src/chainparams.cpp
configure.ac
src/net.cpp
src/init.cpp
src/miner.cpp
src/util.cpp
src/bitcoin-tx.cpp
src/netbase.cpp
src/rpcmining.cpp
src/net.h
qa/pull-tester/rpc-tests.sh
doc/release-notes.md
src/rpcserver.cpp
src/rpcmisc.cpp
src/rest.cpp
src/test/transaction_tests.cpp
src/Makefile.am
src/addrman.cpp
src/rpcserver.h
src/qt/paymentserver.cpp
src/test/util_tests.cpp
src/wallet/wallet.h
src/test/alert_tests.cpp
.travis.yml
doc/README.md
doc/release-process.md
src/test/univalue_tests.cpp
src/wallet/wallet.cpp
src/qt/coincontroldialog.cpp
Files