Preserving Software

Challenges and Opportunities for Reproducibility of Science and Technology

Roberto Di Cosmo

Scilabtec 2015 - Paris

Irill - INRIA - University Paris Diderot
roberto@dicosmo.org

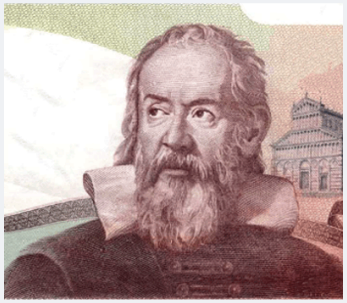
21st of May 2015

Outline

How we built our scientific knowledge

The experimental method

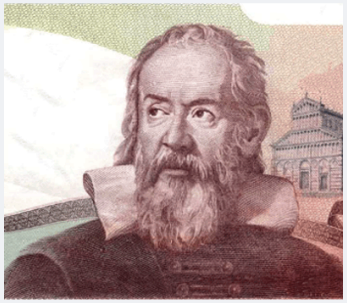
- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- formulate a *theory*



How we built our scientific knowledge

The experimental method

- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- formulate a *theory*

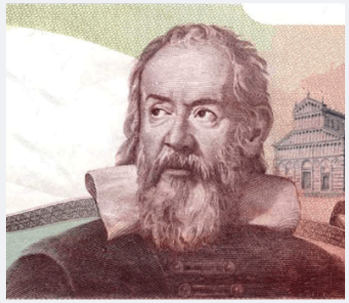


And then we reproduce and verify.

How we built our scientific knowledge

The experimental method

- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- formulate a *theory*

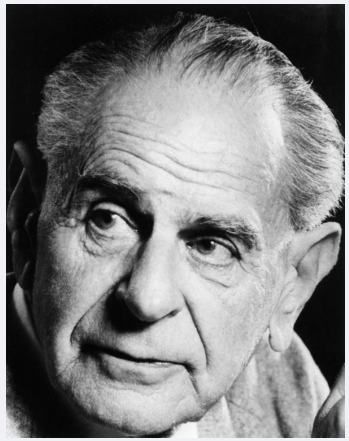


And then we reproduce and verify.

Reproducibility is the key

non-reproducible single occurrences are of no significance to science

Karl Popper, The Logic of Scientific Discovery, 1934



Reproducibility, today

Reproducibility (Wikipedia)

the ability of an entire experiment or study to be *reproduced*, either by the researcher or *by someone else working independently*. It is one of the main principles of the scientific method.

Reproducibility, today

Reproducibility (Wikipedia)

the ability of an entire experiment or study to be *reproduced*, either by the researcher or *by someone else working independently*. It is one of the main principles of the scientific method.

Why we want it

Reproducibility, today

Reproducibility (Wikipedia)

the ability of an entire experiment or study to be *reproduced*, either by the researcher or *by someone else working independently*. It is one of the main principles of the scientific method.

Why we want it

foundation of the scientific method

Reproducibility, today

Reproducibility (Wikipedia)

the ability of an entire experiment or study to be *reproduced*, either by the researcher or *by someone else working independently*. It is one of the main principles of the scientific method.

Why we want it

foundation of the scientific method

accelerator of research: allows to build upon previous work

Reproducibility, today

Reproducibility (Wikipedia)

the ability of an entire experiment or study to be *reproduced*, either by the researcher or *by someone else working independently*. It is one of the main principles of the scientific method.

Why we want it

foundation of the scientific method

accelerator of research: allows to build upon previous work

visibility reproducible results are cited more often

Piwowar et al. *Sharing Detailed Research Data Is Associated with Increased Citation Rate*. PLoS ONE 2(3): e308, 2007

Reproducibility, today

Reproducibility (Wikipedia)

the ability of an entire experiment or study to be *reproduced*, either by the researcher or *by someone else working independently*. It is one of the main principles of the scientific method.

Why we want it

foundation of the scientific method

accelerator of research: allows to build upon previous work

visibility reproducible results are cited more often

Piwowar et al. *Sharing Detailed Research Data Is Associated with Increased Citation Rate*. PLoS ONE 2(3): e308, 2007

transparency of results eases acceptance

Reproducibility, today

Reproducibility (Wikipedia)

the ability of an entire experiment or study to be *reproduced*, either by the researcher or *by someone else working independently*. It is one of the main principles of the scientific method.

Why we want it

foundation of the scientific method

accelerator of research: allows to build upon previous work

visibility reproducible results are cited more often

Piwowar et al. *Sharing Detailed Research Data Is Associated with Increased Citation Rate*. PLoS ONE 2(3): e308, 2007

transparency of results eases acceptance

necessary for industrial transfer

reproducibility is *the essence* of *industry*!

Outline

Today: Software is Science's cornerstone

Software is *an essential component* of modern scientific research

Deep knowledge from *all fields* is embodied in complex software systems

Today: Software is Science's cornerstone

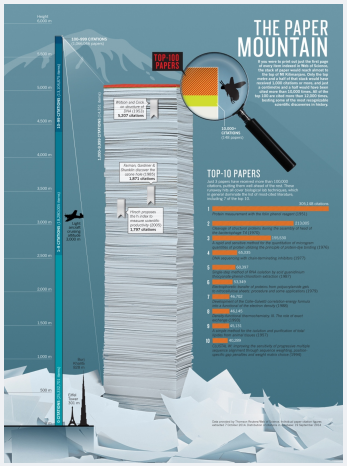
Software is *an essential component* of modern scientific research

Deep knowledge from *all fields* is embodied in complex software systems

Top 100 papers (Nature, October 2014)

[...] the vast majority describe experimental methods or software that have become essential in their fields.

http://www.nature.com/news/the-top-100-papers-1.16224



Today: Software is Science's cornerstone

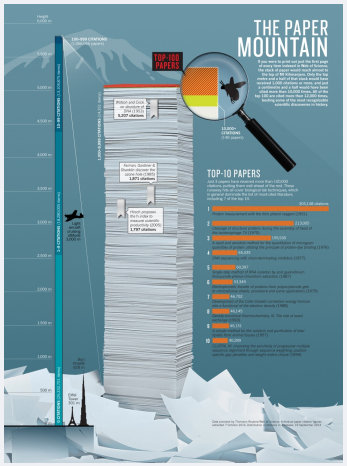
Software is *an essential component* of modern scientific research

Deep knowledge from *all fields* is embodied in complex software systems

Top 100 papers (Nature, October 2014)

[...] the vast majority describe experimental methods or software that have become essential in their fields.

http://www.nature.com/news/the-top-100-papers-1.16224



Software is *the pillar of science*

Software is Pervasive

At the heart of *technology*

televisions/fridges $\approx$ 10M SLOC

phones $\approx$ 20M SLOC

cars $\approx$ 100M SLOC

IoT ...

Software is Pervasive

At the heart of *technology*

televisions/fridges $\approx$ 10M SLOC

phones $\approx$ 20M SLOC

cars $\approx$ 100M SLOC

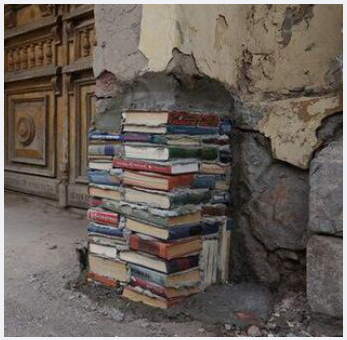
IoT ...

Key mediator for accessing *all* information

(c) Banski

Information is a main pillar of our modern societies.

> *Absent an ability to correctly interpret digital information, we are left with [...] "rotting bits" [...] of no value.*
>
> *Vinton G. Cerf IEEE 2011*



Software is Pervasive

At the heart of *technology*

televisions/fridges $\approx$ 10M SLOC

phones $\approx$ 20M SLOC

cars $\approx$ 100M SLOC

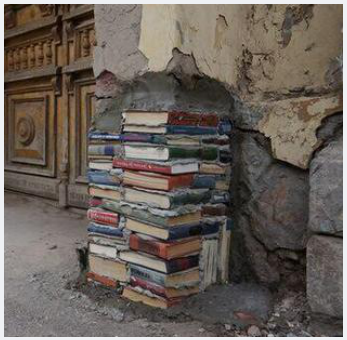
IoT ...

Key mediator for accessing *all* information

(c) Banski

Information is a main pillar of our modern societies.

> *Absent an ability to correctly interpret digital information, we are left with [...] "rotting bits" [...] of no value.*
>
> *Vinton G. Cerf IEEE 2011*



Software is an *essential* enabler for all technology

Reproducibility in the digital age

For an experiment involving software, we need

- **open access** to the scientific article describing it
- **open data sets** used in the experiment
- **source code** of all the components
- **environment** of execution
- **stable references** between all this

Reproducibility in the digital age

For an experiment involving software, we need

- **open access** to the scientific article describing it
- **open data sets** used in the experiment
- **source code** of all the components
- **environment** of execution
- **stable references** between all this

The first two items are already widely acknowledged

- **open access** Berlin declaration, 2003 `http://openaccess.mpg.de`
- **open data sets** Declaration on open access to scientific data, OECD, 2004 `http://bit.ly/1lGubJQ`

Reproducibility in the digital age

For an experiment involving software, we need

- **open access** to the scientific article describing it
- **open data sets** used in the experiment
- **source code** of all the components
- **environment** of execution
- **stable references** between all this

The first two items are already widely acknowledged

- **open access** Berlin declaration, 2003 `http://openaccess.mpg.de`
- **open data sets** Declaration on open access to scientific data, OECD, 2004 `http://bit.ly/1lGubJQ`

What about the code?

Why Open Source?

Having access to (all) the source of (all) the code used in an experiment is important:

- even the simplest experiments depend on a wealth of components and configuration options

Why Open Source?

Having access to (all) the source of (all) the code used in an experiment is important:

- even the simplest experiments depend on a wealth of components and configuration options
- source code is like the proof used in a theorem: no more *Fermat statements* like "*the details are omitted due to lack of space*"

Why Open Source?

Having access to (all) the source of (all) the code used in an experiment is important:

- even the simplest experiments depend on a wealth of components and configuration options
- source code is like the proof used in a theorem: no more *Fermat statements* like "*the details are omitted due to lack of space*"
- access to all the source code is not just necessary to *reproduce*, it is also useful to *evolve and modify*, to *build new experiments* from the old ones

Why Open Source?

Having access to (all) the source of (all) the code used in an experiment is important:

- even the simplest experiments depend on a wealth of components and configuration options
- source code is like the proof used in a theorem: no more *Fermat statements* like "*the details are omitted due to lack of space*"
- access to all the source code is not just necessary to *reproduce*, it is also useful to *evolve and modify*, to *build new experiments* from the old ones

For science, *and industry* ...
Free and Open Source Software is the best choice!

Outline

Software and reproducibility

A fundamental question

How are we doing, regarding reproducibility, in *Software*?

Software and reproducibility

A fundamental question

How are we doing, regarding reproducibility, in *Software*?

The case of Computer Systems Research

A field with Computer experts ... we have high expectations!
Christian Collberg set out to check them.

Software and reproducibility

A fundamental question

How are we doing, regarding reproducibility, in *Software*?

The case of Computer Systems Research

A field with Computer experts ... we have high expectations!
Christian Collberg set out to check them.

Measuring Reproducibility in Computer Systems Research

Long and detailed technical report, March 2014
`http://reproducibility.cs.arizona.edu/v1/tr.pdf`

Collberg's report from the trenches

Analysis of 613 papers

- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

all very practical oriented

The basic question

- can we get the code to build and run?

Collberg's report from the trenches

Analysis of 613 papers

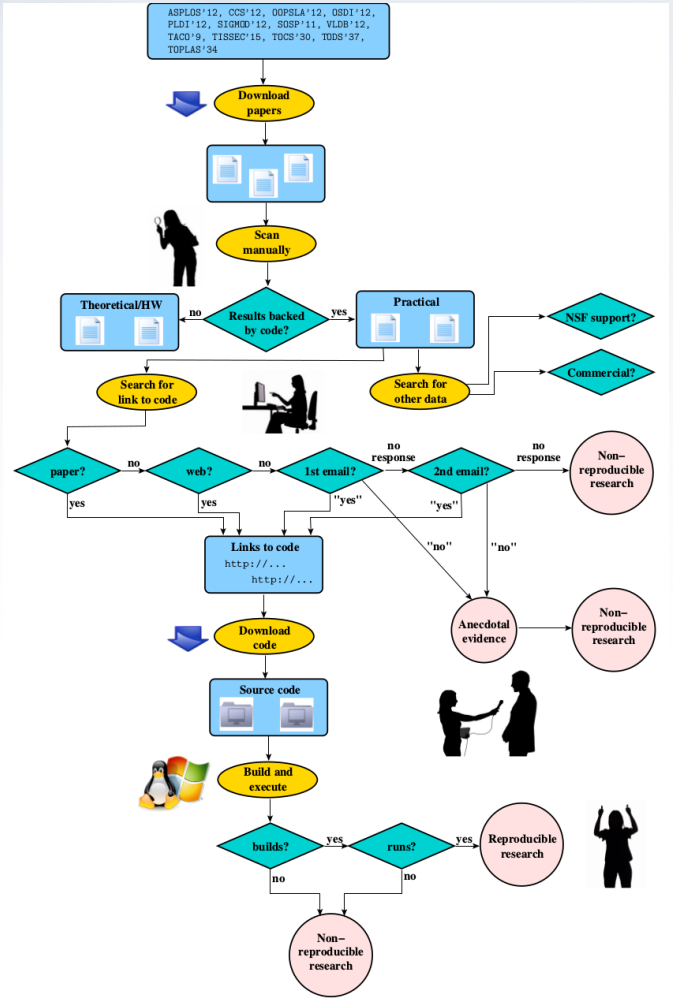
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

all very practical oriented

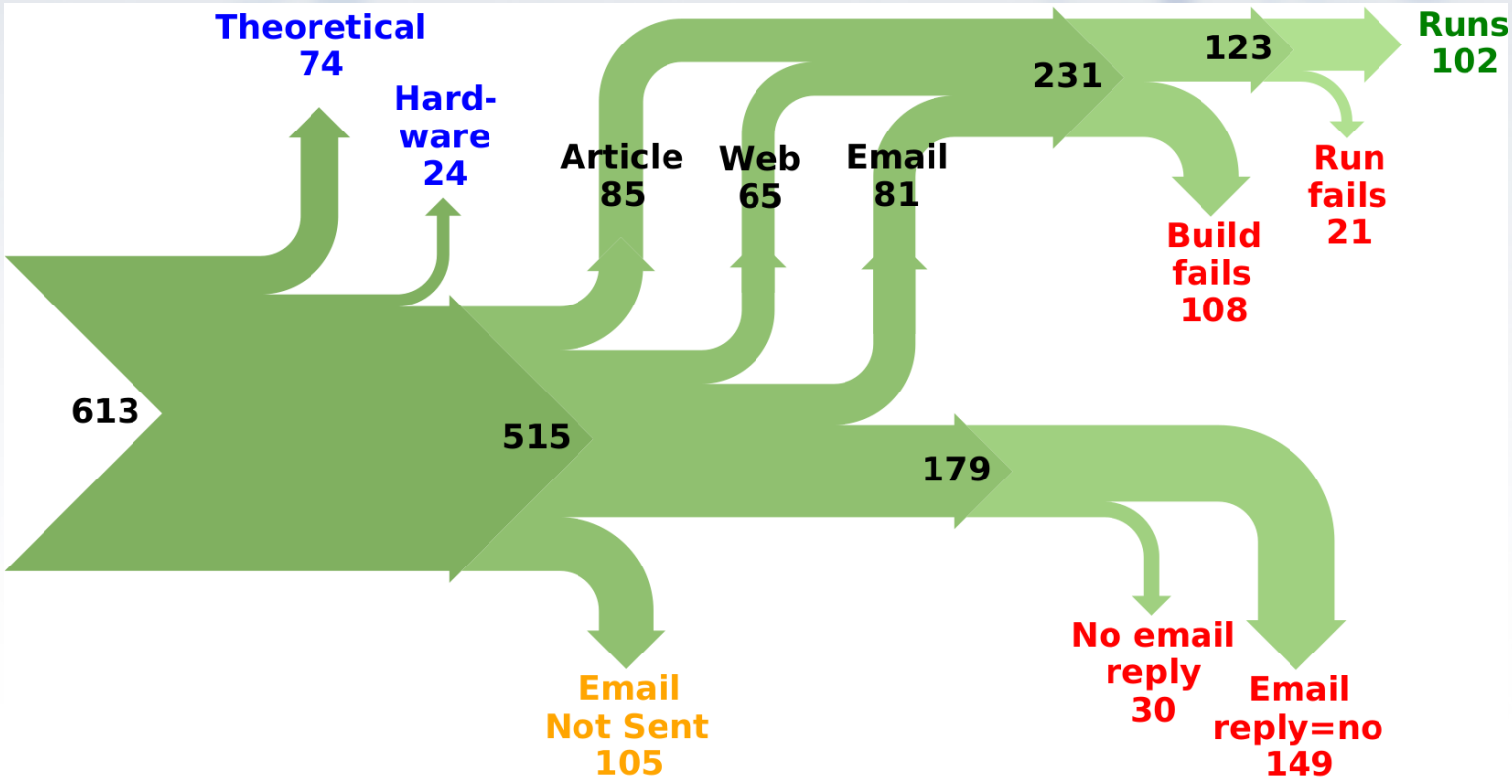
The basic question

- can we get the code to build and run?

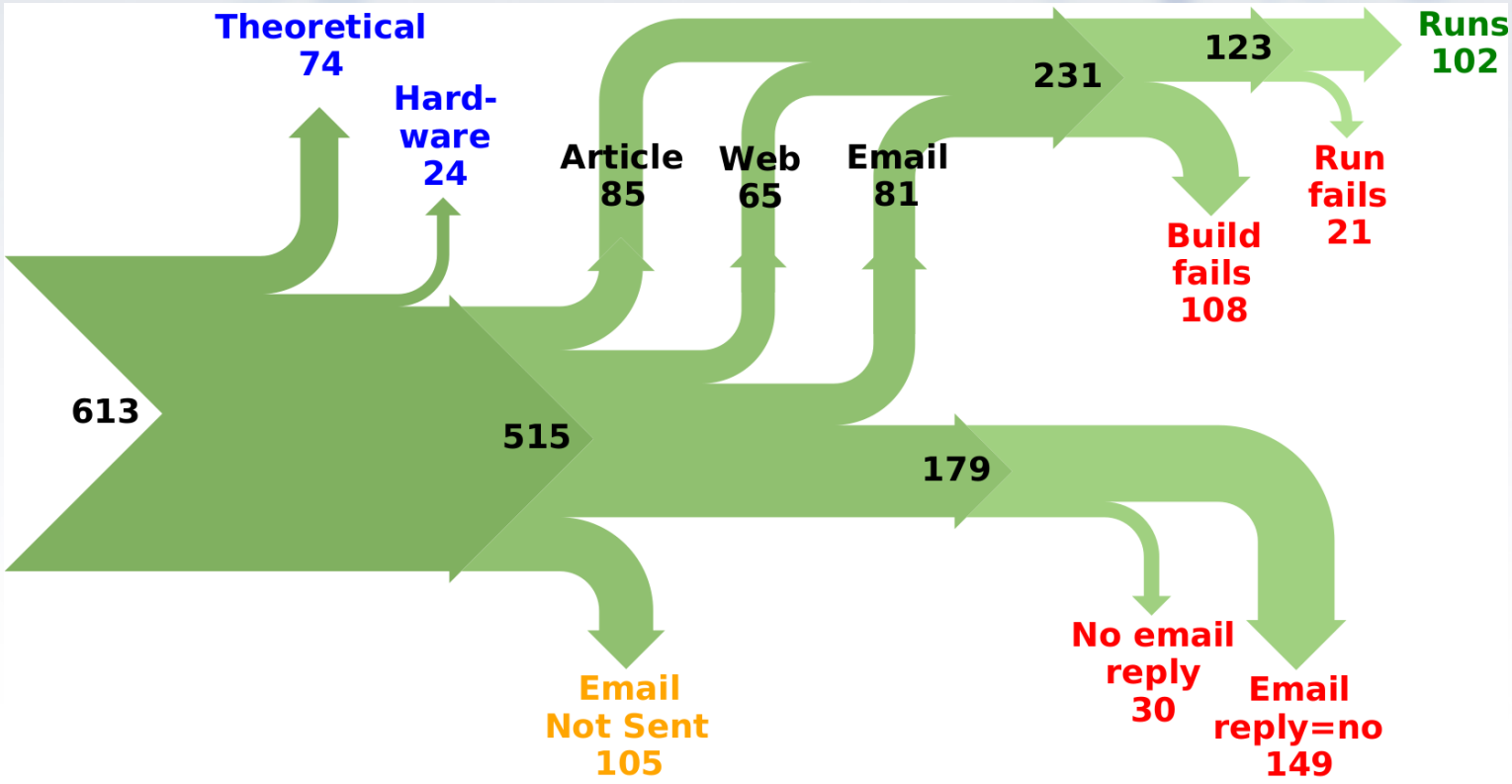
The workflow



The result


Theoretical
74
Hard-
ware
24
Article
85
Web
65
Email
81
231
123
Runs
102
Run
fails
21
Build
fails
108
613
515
179
Email
Not Sent
105
No email
reply
30
Email
reply=no
149

The result



That's a whopping 81% of non reproducible works!

The reasons (or, "the dog ate my program")

Many issues, nice anectotes, and it finally boils down to

- Availability
- Traceability
- Environment
- Automation
- Documentation
- Understanding (Open Source)

The reasons (or, "the dog ate my program")

Many issues, nice anectotes, and it finally boils down to

- Availability
- Traceability
- Environment
- Automation
- Documentation
- Understanding (Open Source)

The first two are actually important *software preservation* issues:
yes, code can be destroyed, and we can loose trace of it!

Outline

Like all digital information, software is *fragile*

An example is worth a thousand words...

Inconsiderate or malicious loss of code

The Year 2000 Bug ... uncovered an inconvenient truth

in 1999, an estimated 40% of companies had either *lost*, or thrown away the original source code for their systems!

Inconsiderate or malicious loss of code

The Year 2000 Bug ... uncovered an inconvenient truth

in 1999, an estimated 40% of companies had either *lost*, or thrown away the original source code for their systems!

CodeSpaces: source code hosting, 2007-2014

Murder in the Amazon cloud

The demise of Code Spaces at the hands of an attacker shows that, in the cloud, off-site backups and separation of services could be key to survival

InfoWorld | Jun 23, 2014

Inconsiderate or malicious loss of code

The Year 2000 Bug ... uncovered an inconvenient truth

in 1999, an estimated 40% of companies had either *lost*, or thrown away the original source code for their systems!

CodeSpaces: source code hosting, 2007-2014

Murder in the Amazon cloud

The demise of Code Spaces at the hands of an attacker shows that, in the cloud, off-site backups and separation of services could be key to survival

InfoWorld | Jun 23, 2014

Yes, for *seven years* all seemed ok.

Inconsiderate or malicious loss of code

The Year 2000 Bug ... uncovered an inconvenient truth

in 1999, an estimated 40% of companies had either *lost*, or thrown away the original source code for their systems!

CodeSpaces: source code hosting, 2007-2014

Murder in the Amazon cloud

The demise of Code Spaces at the hands of an attacker shows that, in the cloud, off-site backups and separation of services could be key to survival

InfoWorld | Jun 23, 2014

Yes, for *seven years* all seemed ok.
No, they did not recover the data.

Business-driven loss of code support: Google

A Change to Google Code Download Service

Posted: Monday, May 20, 2013

+1 391 Tweet 249 Like 295

Project Hosting on Google Code provides a free collaborative development environment for open source projects. Each project comes with its own member controls, Subversion/Mercurial/Git repository, issue tracker, wiki pages, and downloads service.

Downloads were implemented by Project Hosting on Google Code to enable open source projects to make their files available for public download. Unfortunately, downloads have become a source of abuse with a significant increase in incidents recently. Due to this increasing misuse of the service and a desire to keep our community safe and secure, we are deprecating downloads.

Starting today, existing projects that do not have any downloads and all new projects will not have the ability to create downloads. Existing projects with downloads will see no visible changes until January 14, 2014 and will no longer have the ability to create new downloads starting on January 15, 2014. All existing downloads in these projects will continue to be accessible for the foreseeable future.

If your project is using downloads to host and distribute files and has a need to periodically create new downloads, we recommend you move your downloads to an alternate service like Google Drive before January 15, 2014. If you choose to move your files to Google Drive, check out our help article.

By Google Project Hosting

Business-driven loss of code support: Google, cont'd.

Posted: Thursday, March 12, 2015

When we started the Google Code project hosting service in 2006, the world of project hosting was limited. We were worried about reliability and stagnation, so we took action by giving the open source community another option to choose from. Since then, we've seen a wide variety of better project hosting services such as GitHub and Bitbucket bloom. Many projects moved away from Google Code to those other systems. To meet developers where they are, we ourselves migrated nearly a thousand of our own open source projects from Google Code to GitHub.

As developers migrated away from Google Code, a growing share of the remaining projects were spam or abuse. Lately, the administrative load has consisted almost exclusively of abuse management. After profiling non-abusive activity on Google Code, it has become clear to us that the service simply isn't needed anymore.

Beginning today, we have disabled new project creation on Google Code. We will be shutting down the service about 10 months from now on January 25th, 2016. Below, we provide links to migration tools designed to help you move your projects off of Google Code. We will also make ourselves available over the next three months to those projects that need help migrating from Google Code to other hosts.

- March 12, 2015 - New project creation disabled.
- August 24, 2015 - The site goes read-only. You can still checkout/view project source, issues, and wikis.
- January 25, 2016 - The project hosting service is closed. You will be able to download a tarball of project source, issues, and wikis. These tarballs will be available throughout the rest of 2016.

Google will continue to provide Git and Gerrit hosting for certain projects like Android and Chrome. We will also continue maintaining our mirrors of projects like Eclipse, kernel.org and others.

Disruption of the *web of reference*

Web links *are not* permanent (even *permalinks*)

there is no general guarantee that a URL which at one time points to a given object continues to do so

T. Berners-Lee et al. RFC 1738.

404

Disruption of the *web of reference*

Web links *are not* permanent (even *permalinks*)

there is no general guarantee that a URL which at one time points to a given object continues to do so

T. Berners-Lee et al. RFC 1738.

404

URLs used in articles *decay*!

Analysis of IEEE Computer (Computer), and the Communications of the ACM (CACM): 1995-1999

- *the* **half-life** *of a referenced URL* **is approximately 4 years** *from its publication date*

D. Spinellis. The Decay and Failures of URL References. Communications of the ACM, 46(1):71-77, January 2003.

Outline

Preservation of digital information is on the rise

A wealth of initiatives around us

generalist	the Web at `archive.org`; National Digital Information Infrastructure and Preservation Program (NDIIPP, USA); ...
culture	books, music, video: INA (FR); `http://www.nationalarchives.gov.uk` (UK) ...
social networks	Twitter is archived by the Library of Congress!
libraries and scholarly work	ArXiv; Digital Preservation Network `http://www.dpn.org/`; ...
scientific data	CINES (FR); Zenodo/OpenAire (CERN); ...

Preservation of digital information is on the rise

A wealth of initiatives around us

generalist	the Web at `archive.org`; National Digital Information Infrastructure and Preservation Program (NDIIPP, USA); ...
culture	books, music, video: INA (FR); `http://www.nationalarchives.gov.uk` (UK) ...
social networks	Twitter is archived by the Library of Congress!
libraries and scholarly work	ArXiv; Digital Preservation Network `http://www.dpn.org/`; ...
scientific data	CINES (FR); Zenodo/OpenAire (CERN); ...

And yet, up to now...

software is *largely ignored* as an object of preservation...
computer scientists are absent from the preservation landscape!

Preservation of digital information is on the rise

A wealth of initiatives around us

generalist the Web at `archive.org`; National Digital Information Infrastructure and Preservation Program (NDIIPP, USA); ...

culture books, music, video: INA (FR); `http://www.nationalarchives.gov.uk` (UK) ...

social networks Twitter is archived by the Library of Congress!

libraries and scholarly work ArXiv; Digital Preservation Network `http://www.dpn.org/`; ...

scientific data CINES (FR); Zenodo/OpenAire (CERN); ...

And yet, up to now...

software is *largely ignored* as an object of preservation...

computer scientists are absent from the preservation landscape!

It's time to change all this!

Software Preservation Challenges: it's *different*!

Unlike for books or movies, there is a big difference between *using* and *understanding* a piece of software.

Using software

Requires an executable, and access to the *execution environment*

Software Preservation Challenges: it's *different*!

Unlike for books or movies, there is a big difference between *using* and *understanding* a piece of software.

Using software

Requires an executable, and access to the *execution environment*

Understanding software

Requires access to the *source code*:

> *The source code for a work means the preferred form of the work for making modifications to it.*
>
> — *GNU General Public Licence, version 2*

Software Preservation Challenges: it's *different*!

Unlike for books or movies, there is a big difference between *using* and *understanding* a piece of software.

Using software

Requires an executable, and access to the *execution environment*

Understanding software

Requires access to the *source code*:

> *The source code for a work means the preferred form of the work for making modifications to it.*
>
> — *GNU General Public Licence, version 2*

For reproducibility, we need *both*!

Software Preservation Challenges: it's *different*!

Preserving *software* is more complex than archiving books or scientific articles.

interdependencies a program relies on other software as well as specific hardware components to be executed; so does our understanding of its functionality

Software Preservation Challenges: it's *different*!

Preserving *software* is more complex than archiving books or scientific articles.

interdependencies a program relies on other software as well as specific hardware components to be executed; so does our understanding of its functionality

evolution software is a *live object*: its detailed history contains key knowledge

Software Preservation Challenges: it's *different*!

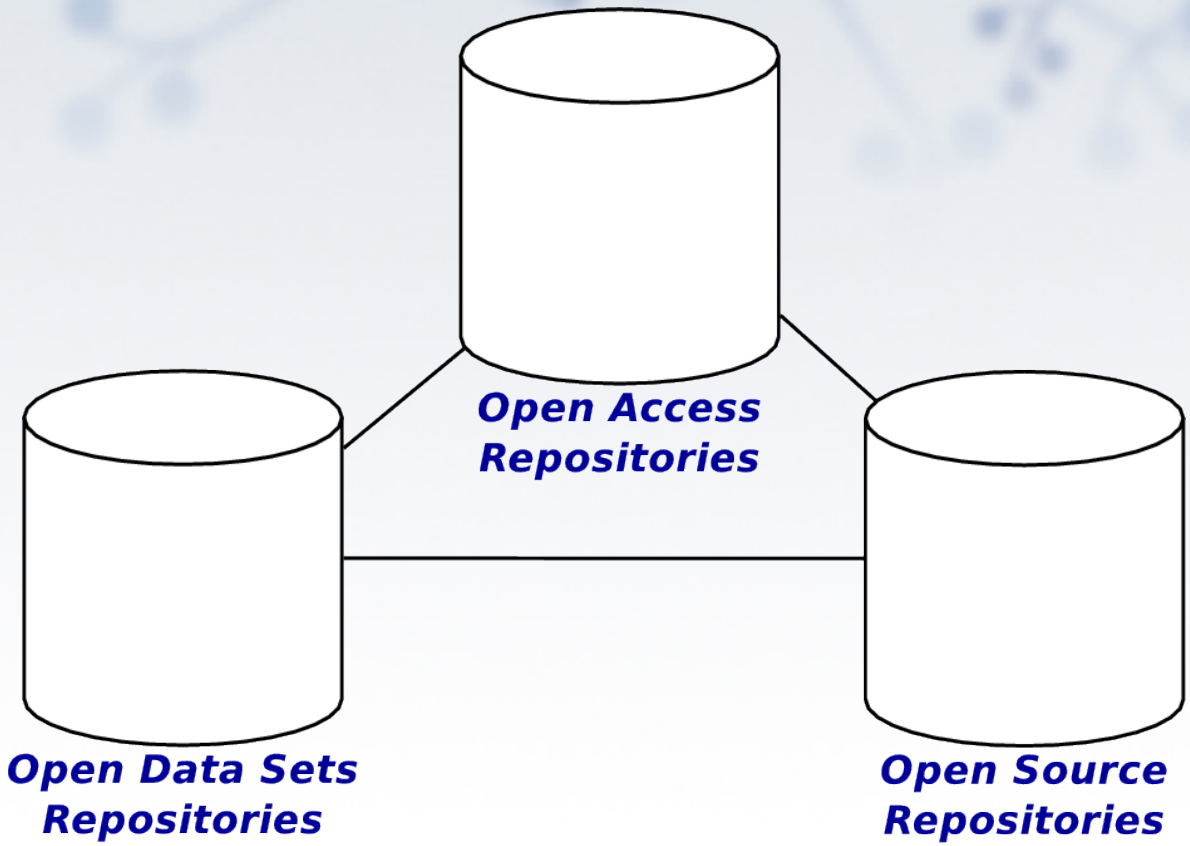
Preserving *software* is more complex than archiving books or scientific articles.

interdependencies a program relies on other software as well as specific hardware components to be executed; so does our understanding of its functionality

evolution software is a *live object*: its detailed history contains key knowledge

To preserve *software* it is *not enough* to mimic processes that were intended to archive books, scientific articles or data.

The Knowledge Conservancy Magic Triangle



articles	ArXiv, HAL, ...
data	Zenodo, OpenAire, ...
software	is the next to come, and is our mission

Conclusions

Preservation is the _foundation_ of all reproductibility efforts

We all must do our best to

- preserve and make accessible scientific articles and data
- preserve and make accessible the software

Conclusions

Preservation is the _foundation_ of all reproductibility efforts

We all must do our best to

- preserve and make accessible scientific articles and data
- preserve and make accessible the software

Because software embodies our scientific and technical knowledge

Conclusions

Preservation is the _foundation_ of all reproductibility efforts

We all must do our best to

- preserve and make accessible scientific articles and data
- preserve and make accessible the software

Because software embodies our scientific and technical knowledge

> Conoscere è saper fare.
>
> *Giambattista Vico, 1668–1744, Naples*

Conclusions

Preservation is the _foundation_ of all reproductibility efforts

We all must do our best to

- preserve and make accessible scientific articles and data
- preserve and make accessible the software

Because software embodies our scientific and technical knowledge

> Conoscere è saper fare.
>
> *Giambattista Vico, 1668–1744, Naples*

Questions ?