How Open Source Geospatial Development Works

GRASS GIS, free and open source software and academia

Václav Petráš (Vashek)

Center for Geospatial Analytics
Department of Marine, Earth and Atmospheric Sciences
Open Source Geospatial Research and Education Laboratory
North Carolina State University

October 30th, 2014

Table of Contents

Motivation for open source

- *80% of developers used open source in past 12 months*[1]
  - 99% in India and China
  - numbers greater for students in general
- quality, customizability, no vendor lock-in, flexibility, interoperability[2]
- no license fees

[1] Recent (2014) survey by Forester Research (presented at All Things Open)
[2] PCWorld: 10 Reasons Open Source Is Good for Business

Karen Sandler

- executive director of the Software Freedom Conservancy
- a cyborg lawyer [3, 4]
- pacemaker/defibrillator implanted
- no review of the software or hardware available

[3] Software Freedom Conservancy: . . . Karen in Raleigh and Seattle. . .
[4] All Things Open, Raleigh

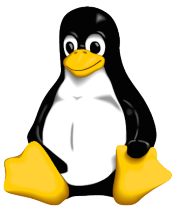
Free, Libre and Open Source Software

Very similar to each other:

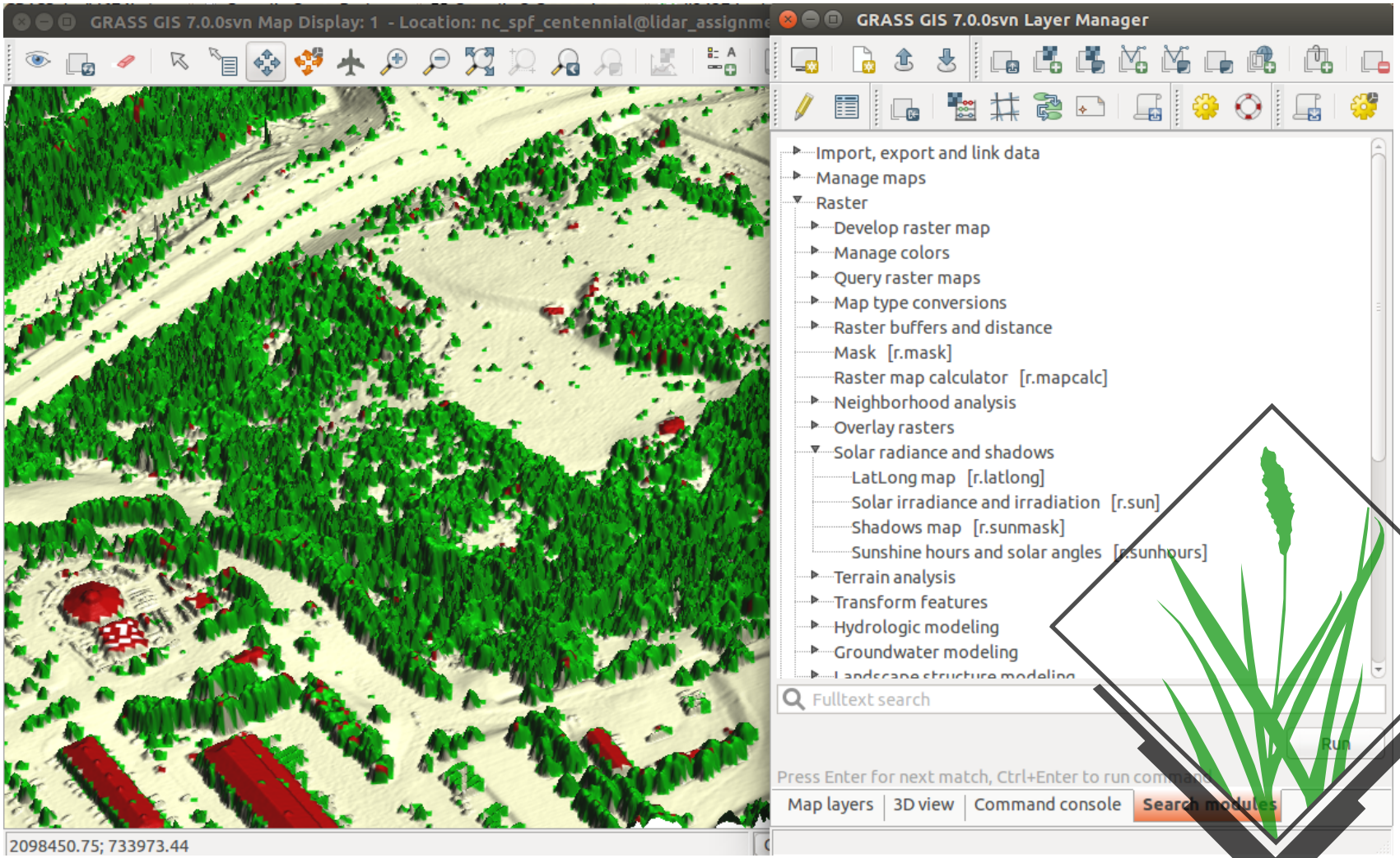
- free software
- open source software
- free/libre software
- FS, OSS, FOSS, FLOSS

Very different from FLOSS:

- proprietary software
- freeware
- shareware



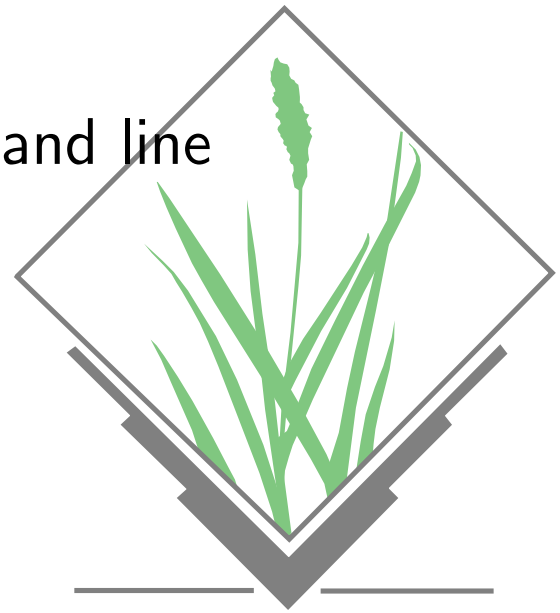
GRASS GIS



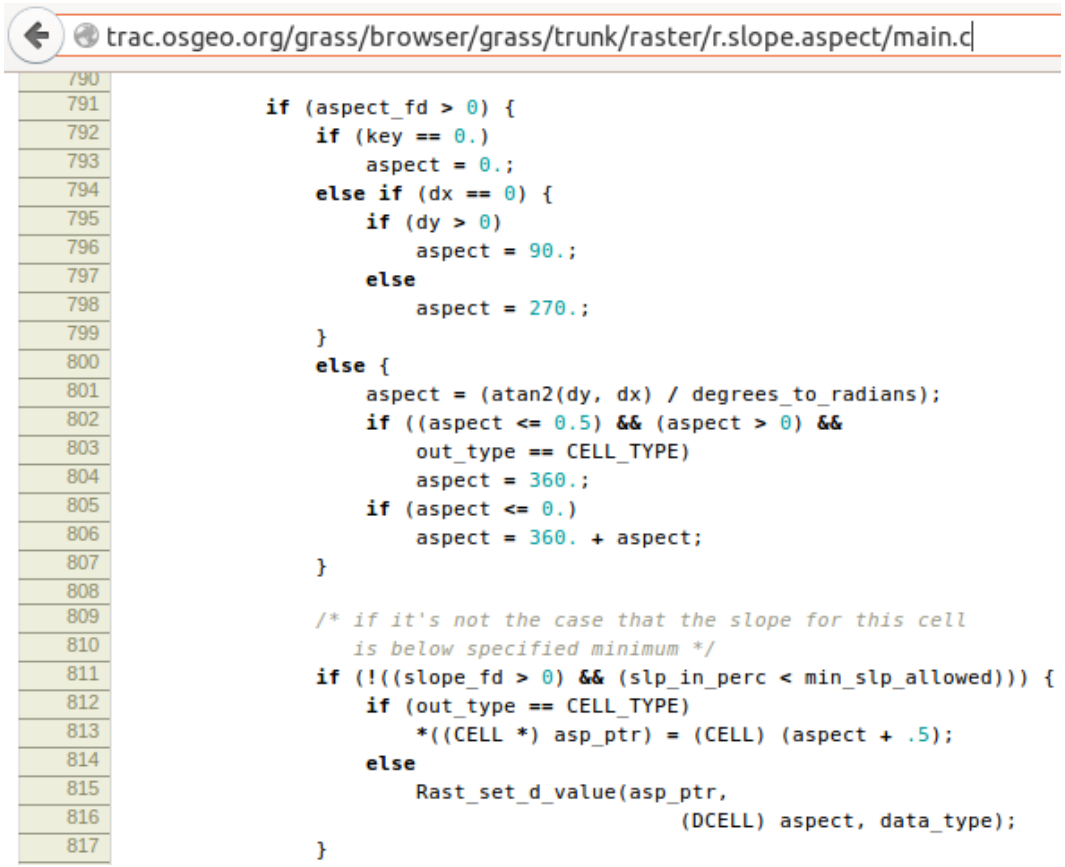
http://grass.osgeo.org/

GRASS GIS

- MS Windows, Mac OS X, GNU/Linux, clusters, ...
- Uses SQLite, GDAL/OGR, ...
- Used by QGIS, gvSIG, ...
- Works with R, PostGIS, ...
- graphical user interface, C, Python, command line
- celebrated 30 years in 2013

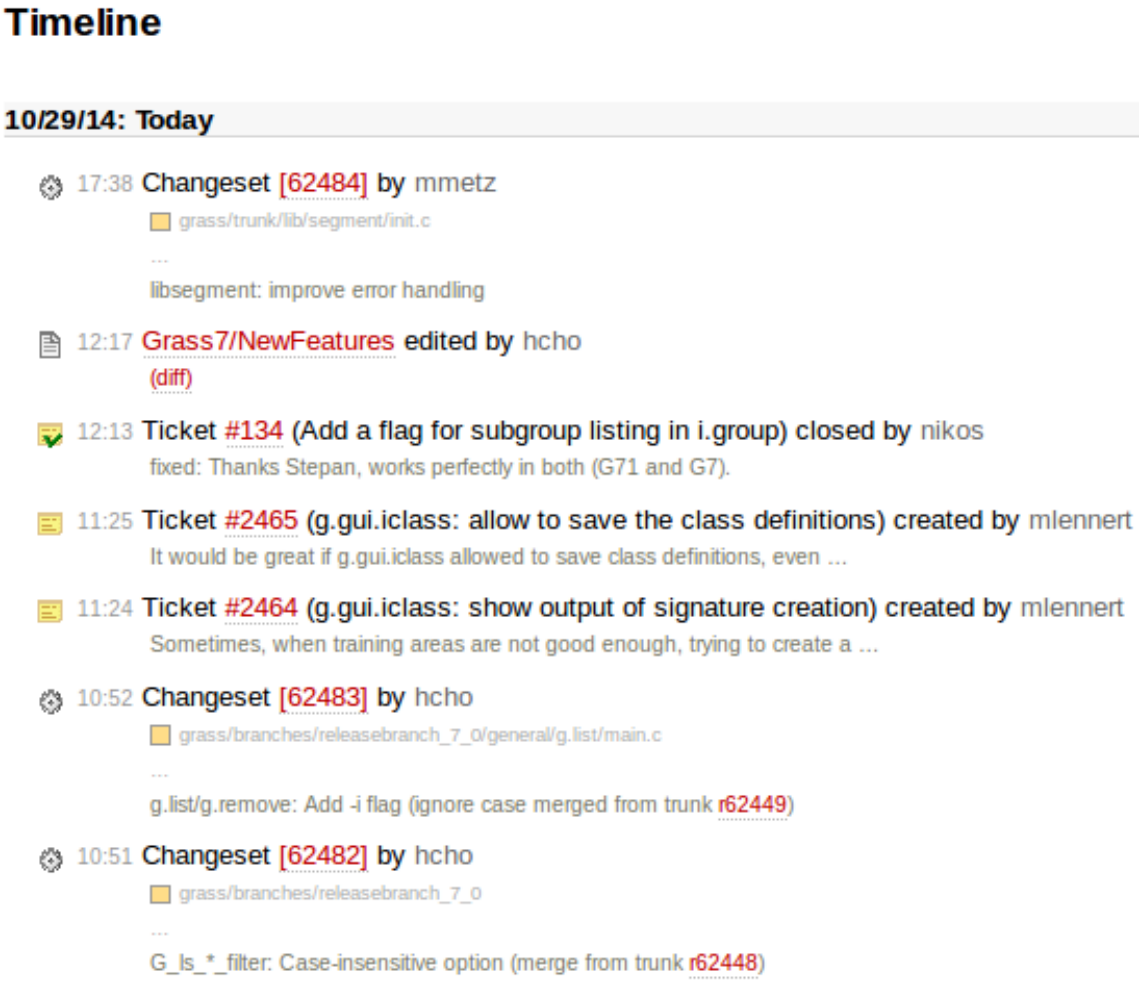


Source code



http://trac.osgeo.org/grass/
http://trac.osgeo.org/grass...r.slope.aspect...

Changes



Timeline

10/29/14: Today

17:38 Changeset [62484] by mmetz
grass/trunk/lib/segment/init.c
...
libsegment: improve error handling

12:17 Grass7/NewFeatures edited by hcho
(diff)

12:13 Ticket #134 (Add a flag for subgroup listing in i.group) closed by nikos
fixed: Thanks Stepan, works perfectly in both (G71 and G7).

11:25 Ticket #2465 (g.gui.iclass: allow to save the class definitions) created by mlennert
It would be great if g.gui.iclass allowed to save class definitions, even ...

11:24 Ticket #2464 (g.gui.iclass: show output of signature creation) created by mlennert
Sometimes, when training areas are not good enough, trying to create a ...

10:52 Changeset [62483] by hcho
grass/branches/releasebranch_7_0/general/g.list/main.c
...
g.list/g.remove: Add -i flag (ignore case merged from trunk r62449)

10:51 Changeset [62482] by hcho
grass/branches/releasebranch_7_0
...
G_ls_*_filter: Case-insensitive option (merge from trunk r62448)

`http://trac.osgeo.org/grass/timeline`

Ticket: Feature request

Ticket #2368 (new enhancement)

Python version of r.grow does not support shrinking

Opened 3 months ago
Last modified 3 weeks ago

Reported by:	wenzeslaus	Owned by:	grass-dev@...
Priority:	normal	Milestone:	7.1.0
Component:	Raster	Version:	svn-trunk
Keywords:	r.grow, r.grow.distance, r.buffer, r.buffer.lowmem	Cc:	
Platform:	Unspecified	CPU:	Unspecified

Description

Reply

While C version of `r.grow` supports shrinking (negative distance/radius) since r59735, the Python version of `r.grow` based on C module `r.grow.distance` and `r.mapcalc` expression supports only growing (positive distance/radius). Shrinking (negative buffer) is a useful feature, so I think we should add it.

I'm not sure if negative distances can be added to `r.grow.distance`.

If not, I would say that we need to use C implementation of `r.grow` (which might be even faster then the Python script version which calls several modules and creates temporary maps).

If duplication of code would be an issue in case of C `r.grow` and `r.grow.distance`, we may consider adding some functions to the library if they are useful for more modules.

`r.buffer` and `r.buffer.lowmem` does not seem to support it neither, as far as I know.

`v.buffer` supports negative distances ("inward buffer" / "negative buffer").

`http://trac.osgeo.org/grass/ticket/2368`

Ticket: Bug report

Ticket #2427 (closed defect: fixed)

r.cost doesn't finish on Windows

Opened 6 weeks ago
Last modified 5 weeks ago

Reported by:	annakrat	Owned by:	grass-dev@...
Priority:	major	Milestone:	7.0.0
Component:	Raster	Version:	svn-releasebranch70
Keywords:	r.cost	Cc:	
Platform:	MSWindows 8	CPU:	Unspecified

Description

On Windows, r.cost doesn't finish, it keeps running and outputting progress over 100%. It seems that some loop is not finishing.

How to test in nc_spm:

```
g.region swwake_30m -p
v.to.rast roadsmajor out=roadsmajor use=val type=line
r.mapcalc "area_one = 1"
r.cost -k in=area_one output=dist_toroad start_rast=roadsmajor
```

On Linux, this seems to be running fine. When I try not to use -k on Windows, it finishes correctly. But the user who actually found this was trying to run it without -k and still it didn't work (but I can't confirm it).

http://trac.osgeo.org/grass/ticket/2427

Ticket: Bug report

Change History

Changed 5 weeks ago by mmetz — in reply to: ↑ description ; follow-up: ↓ 2

Replying to ~~annakrat~~:

> On Windows, r.cost doesn't finish, it keeps running and outputting progress over 100%. It seems that some loop is not finishing.

Fixed in r62043,4 (trunk, 7.0).

Changed 5 weeks ago by annakrat — in reply to: ↑ 1

- **status** changed from *new* to *closed*
- **resolution** set to *fixed*

Replying to mmetz:

> Replying to ~~annakrat~~:
>
> > On Windows, r.cost doesn't finish, it keeps running and outputting progress over 100%. It seems that some loop is not finishing.
>
> Fixed in r62043,4 (trunk, 7.0).

Thanks for such a quick fix. It's working now.

`http://trac.osgeo.org/grass/ticket/2427`

Ticket: Commit

Changeset 62446

Timestamp: 10/28/14 12:26:15 (11 hours ago)
Author: martinl
Message: v.kriging: keyword / label cosmetics
Files: 1 modified

grass-addons/grass7/vector/v.kriging/main.cpp (1 diff)

Unmodified Added Removed

grass-addons/grass7/vector/v.kriging/main.cpp

r62445	r62446	
48	48	`/* ------- Module creation ------- */`
49	49	`module = G_define_module();`
50		`G_add_keyword(_("Raster"));`
	50	`G_add_keyword(_("raster"));`
51	51	`G_add_keyword(_("3D raster"));`
52		`G_add_keyword(_("Ordinary kriging - for 2D and 3D data"));`
53		`module->description =`
54		`  _("Interpolates 2D or 3D raster based on input values located on 2D or 3D point vector layer (meth`
55		
	52	`G_add_keyword(_("ordinary kriging"));`
	53	`module->label =`
	54	`  _("Interpolates 2D or 3D raster map based on input values located on 2D or 3D vector point map.");`
	55	`module->description = _("Method ordinary kriging extended to 3D.");`
	56	
56	57	`// Setting options`
57	58	`opt.input = G_define_standard_option(G_OPT_V_INPUT); // Vector input layer`

http://trac.osgeo.org/grass/changeset/62446

Who can make changes?

Wikipedia:

- everybody can make changes

Open source projects:

- only people with granted access can make changes
  - everybody can make changes and submit them for approval
  - everybody can discuss changes
  - everybody can view changes

Peer review: Revert

Vaclav Petras wrote:

> Anna and I have fixed the script handing on MS Windows. Modules which are
> Python scripts and/or are from GRASS addons and/or calls other modules
> which are Python scripts didn't worked for various reasons. The changeset
> r57910 should fix them all (and thus should fix #2039, particularly the
> MAXREPEAT error).

Note that I've removed the hacks from core.py. So whatever bugs they were trying to work around will have to be fixed where they originate.

[GRASS-dev] Handling of Python scripts on MS Windows (October, 2013)

Mailing lists

October 2014 Archives by thread

- **Messages sorted by:** [subject] [author] [date]
- **More info on this list...**

Starting: *Wed Oct 1 09:04:55 PDT 2014*
Ending: *Wed Oct 29 13:50:47 PDT 2014*
Messages: 161

- [GRASS-user] r.stream.order *César Augusto Ramírez Franco*
  - [GRASS-user] r.stream.order *Svein Harald Sønderland*
- [GRASS-user] Grass70 win7 r.basin *César Augusto Ramírez Franco*
- [GRASS-user] Grass70 win7 r.basin *César Augusto Ramírez Franco*
  - [GRASS-user] Grass70 win7 r.basin *José Anderson*
    - [GRASS-user] Grass70 win7 r.basin *Margherita Di Leo*
- [GRASS-user] Gauss filter in r.neighbors *Martin Album Ytre-Eide*
  - [GRASS-user] Gauss filter in r.neighbors *Glynn Clements*
    - [GRASS-user] Gauss filter in r.neighbors *Martin Album Ytre-Eide*
      - [GRASS-user] Gauss filter in r.neighbors *Glynn Clements*
- [GRASS-user] r.neighbors selecting a random value from the window *Milton Ribeiro*
  - [GRASS-user] r.neighbors selecting a random value from the window *Glynn Clements*

`http://lists.osgeo.org/mailman/listinfo/grass-user`

`http://lists.osgeo.org/mailman/listinfo/grass-dev`

Other means of communication

- Geographical Information Systems Stack Exchange
  - question and answers are ranked to increase quality
- discussions for bug reports and feature requests
- code/community sprints

Development statistics

In a Nutshell, GRASS GIS...

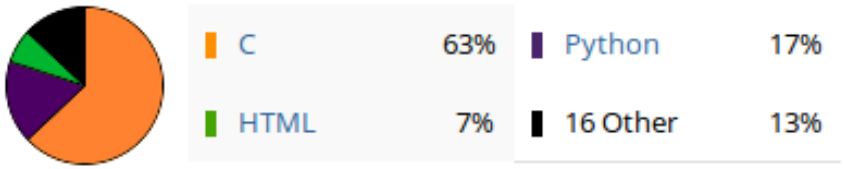
... has had 82,622 commits made by 71 contributors
representing 2,039,215 lines of code

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

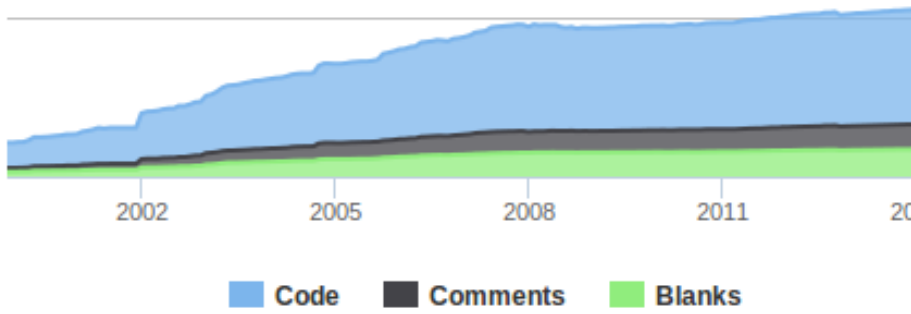
... has a well established, mature codebase
maintained by a large development team
with stable Y-O-Y commits

... took an estimated 585 years of effort (COCOMO model)
starting with its first commit in December, 1999
ending with its most recent commit 2 days ago

Languages



Lines of Code



Black Duck Open Hub: GRASS GIS

Testing and quality assurance

Testsuite	Status	Test files	Successful	Percent
./lib/python/temporal	FAILED	4	3	75%
./raster/r.mapcalc	succeeded	3	3	100%
./lib/python/pygrass/gis	FAILED	2	1	50%
./lib/python/exceptions	succeeded	1	1	100%
./lib/gis	succeeded	1	1	100%
./lib/python/pygrass/modules/interface	succeeded	3	3	100%
./temporal/t.rast3d.univar	succeeded	1	1	100%
./general/g.remove	succeeded	1	1	100%
./temporal/t.rast.aggregate	succeeded	2	2	100%
./lib/python/pygrass/messages	succeeded	1	1	100%
./lib/python/pygrass/vector	FAILED	5	2	40%

`http://fatra.cnr.ncsu.edu/grassgistests/`

Git and GitHub: latest phenomena

- GitHub is a web-based hosting service
- GitHub hosts Git repositories
- Git is a revision control system
- manages source code and its changes

- `https://github.com/ncsu-osgeorel/grass-temporal-workshop`
- `https://github.com/qgis/QGIS`

GRASS GIS temporal framework: sharing and reuse

- analysis and management of spatio-temporal data
- published paper with description and use cases
- created for environmental modeling
- integrated into GRASS GIS
- we all can use it to work with our temporal data

Gebbert, S., Pebesma, E. (2014). TGRASS: A temporal GIS for field based environmental modeling. Environmental Modelling & Software, 53, 1-12.

GRASS GIS r.li: sustainable reproducibility and reuse

multiscale analysis of landscape structure

☐	@62462	[62462]	21 hours	lucadelu	r.li.setup: removed because obsolete
☐	@62421	[62421]	2 days	hcho	Rename g.remove's names= to name= and use name= instead of pattern= as ...
☐	@62280	[62280]	12 days	wenzeslaus	remove C++ style comments of old code (overleft from r61812)
☐	@62015	[62015]	6 weeks	hcho	Rename g.mlist to g.list and update g.remove usage
☐	@61813	[61813]	8 weeks	neteler	r.li manual: North Carolina example added
☐	@61812	[61812]	8 weeks	wenzeslaus	r.li: fix memory handling (memory leak in avl_to_array function) ...
☐	@61681	[61681]	2 months	hellik	r.li.*: better wording in manual for folder where conf file is stored in ...
☐	@21812	[21812]	8 years	markus	Serena Pallecchi/Faunalia?: various fixes; r.li lib now shared lib
☐	@21811	[21811]	8 years	markus	Serena Pallecchi/Faunalia?: new r.li index added
☐	@21810	[21810]	8 years	markus	Serena Pallecchi: new r.li index added
☐	@21809	[21809]	8 years	markus	dos2unix'ed to avoid patch evil
☐	@21723	[21723]	8 years	markus	these TODOs should be really addressed now
☐	@21597	[21597]	8 years	markus	C Porta, L D Spano, S Pallecchi; commission from www.faunalia.it, Italy

`http://trac.osgeo.org/grass/log/grass/trunk/raster/r.li...`

Reproducibility — challenge of current science

- research depends on software
  - open source is available to anyone
- format in which we share is not helpful
  - source code is suitable for reproducing
- hard to report that something does not work
  - pointing out things which are wrong works in open source

Reproducibility — challenge of current science

- research depends on software
  - open source is available to anyone
- format in which we share is not helpful
  - source code is suitable for reproducing
- hard to report that something does not work
  - pointing out things which are wrong works in open source

Reproducibility — challenge of current science

- research depends on software
  - open source is available to anyone
- format in which we share is not helpful
  - source code is suitable for reproducing
- hard to report that something does not work
  - pointing out things which are wrong works in open source

Reproducibility — challenge of current science

- research depends on software
  - open source is available to anyone
- format in which we share is not helpful
  - source code is suitable for reproducing
- hard to report that something does not work
  - pointing out things which are wrong works in open source

Other benefits

- each student has all the tools available, even those which are not used by school
- possibility to work closely with developers
- history of changes and discussions available after years for everybody to learn
- tools to collaborate and share research results

Other benefits

- each student has all the tools available, even those which are not used by school
- possibility to work closely with developers
- history of changes and discussions available after years for everybody to learn
- tools to collaborate and share research results

Other benefits

- each student has all the tools available, even those which are not used by school
- possibility to work closely with developers
- history of changes and discussions available after years for everybody to learn
- tools to collaborate and share research results

Other benefits

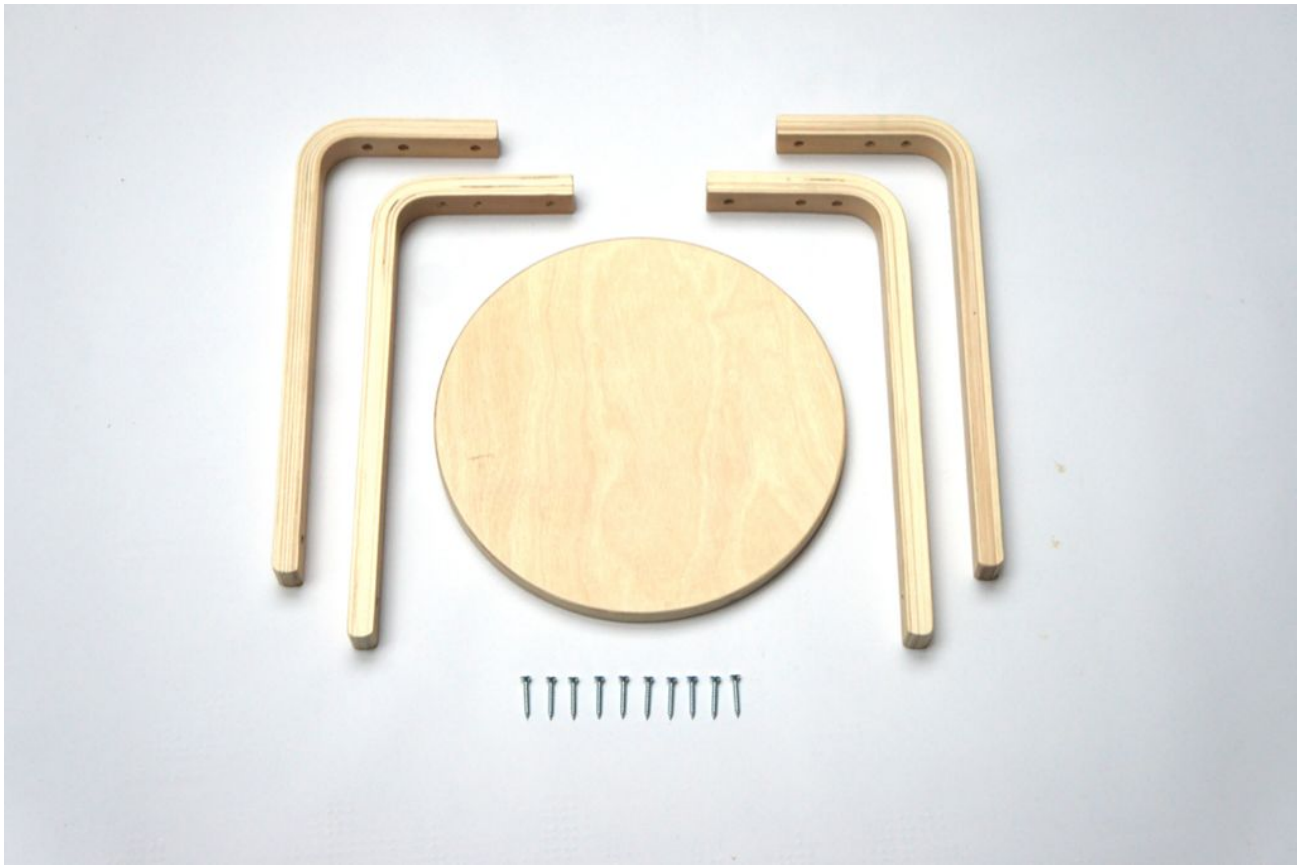
- each student has all the tools available,
  even those which are not used by school
- possibility to work closely with developers
- history of changes and discussions available after years for everybody to learn
- tools to collaborate and share research results

Other benefits: open source has strong culture of reuse

IKEA FROSTA stool

Other benefits: open source has strong culture of reuse

IKEA FROSTA stool



Other benefits: open source has strong culture of reuse

IKEA FROSTA shelf

Other benefits: open source has strong culture of reuse

IKEA FROSTA table

Other benefits: open source has strong culture of reuse

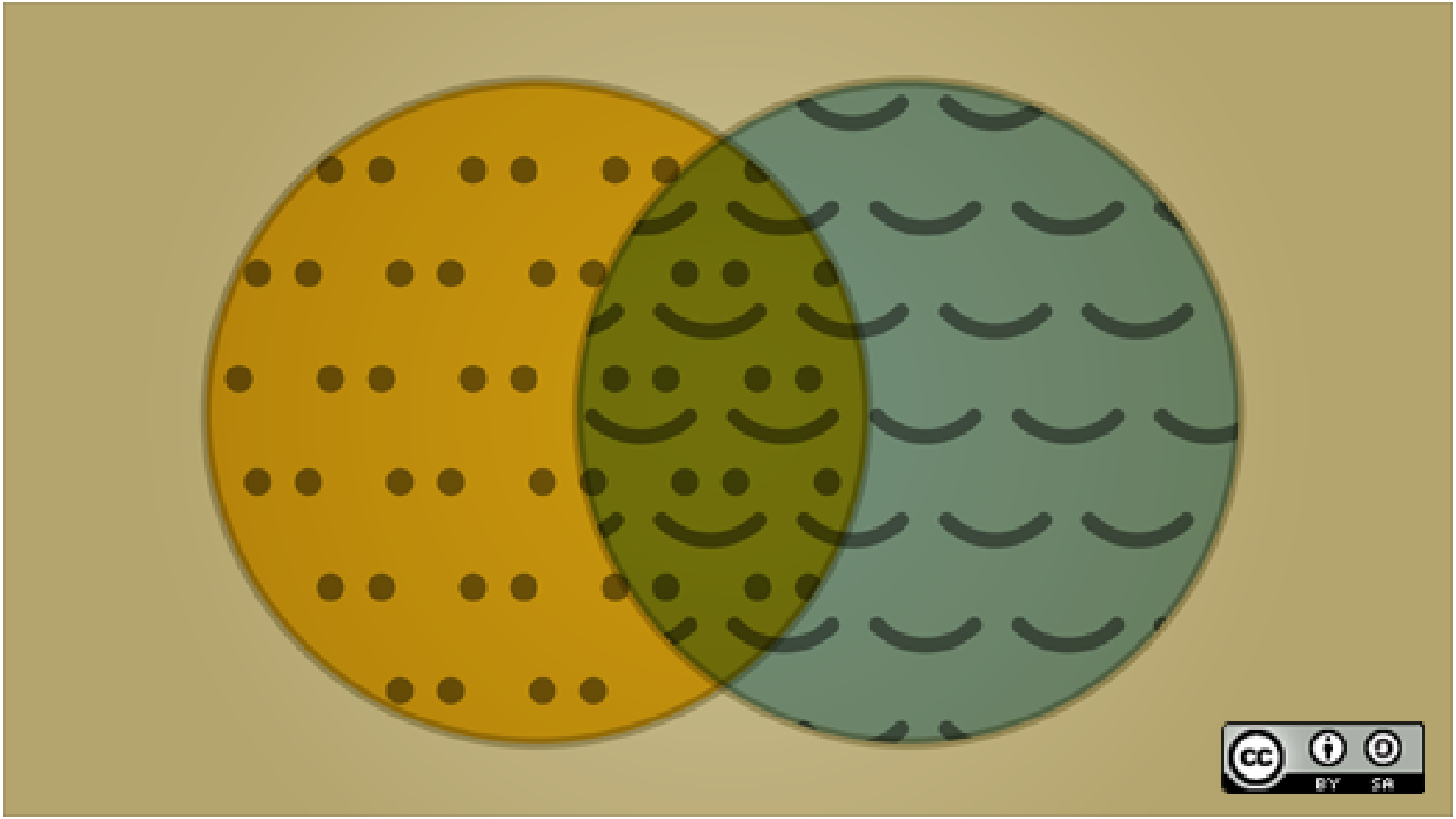
IKEA FROSTA bike

Other benefits: open source has strong culture of reuse

IKEA FROSTA sled

How to use open source

- if your search for a solution was not successful, ask
- report bugs if you encounter them
  - open source software is something you can change and influence
- if you change something, contribute back
  - you cannot maintain your own version
- share your own work to enable reuse
  - your research is worth sharing
- incorporate your work into existing projects
  - this is the way to preserve it



Users of open source are more happy when using the software.

Conclusion

What can science gain by using open source?

- there is no inside or outside (everybody can access and contribute)
- shared and reproducible by default (results and processes)

How to make science better using open source?

- start with yourself, add open source tools to your daily workflow

Acknowledgement

- presentation created in LaTeX Beamer

Pictures:

- smile and open here images from `http://opensource.com/`
- original Frosta stool image from `http://www.ikea.com/`
- furniture Frosta derivatives images from `http://www.coroflot.com/`
- bike and sled Frosta derivatives images from `http://www.3ders.org/`
- KS photo from `http://allthingsopen.org/`
- Linux (penguin), GNU, and GPL logos from `http://gnu.org/`
- CC logo from `https://creativecommons.org/`
- Open Source Initiative logo from `http://opensource.org/`
- GRASS GIS logo from `http://opensource.org/`
- Git logo from `http://git-scm.org/`
- GitHub (octocat) logo from `https://octodex.github.com/`