GNU Radio in 2019: Facts and Plans
An overview of where GNU Radio is going this fine year

Marcus Müller

2018-09-18
What’ll happen in the next 40 minutes

Looking back at 5 years of 3.7

What has happened to 3.7?

Releasing GNU Radio 3.8

3.8.0.0+ and beyond

Conclusion
Marcus Müller
Bearer of a couple of roles

- Support Grumpiness supplier
- Research assistant at KIT
  - I hold the exercise classes for KIT EEs’ *Probability Theory* and *Communications Theory* courses (> 300 students) and *Applied Information Theory* (ca 13 dB fewer students)
- Freelancing Engineer
  - Technical Consulting
  - Contract Development
  - Seminars
- **Chief Architect of the GNU Radio project**
Depending on what you want to talk to me about, contact me using

- University Research & Teaching: mueller@kit.edu
- GNU Radio aspects: Preferably, discuss-gnuradio@gnu.org, for confident matters mmueller@gnuradio.org
- Freelancing & Private: mueller@hostalia.de
State of GNU Radio 2019

GNU Radio 3.7 released June 2013

next branch forked off at that point
State of GNU Radio 2019

GNU Radio 3.7 released June 2013

next branch forked off at that point
What has happened to 3.7 since FOSDEM’18?

New versioning scheme: Semantic Versioning (https://semver.org)
Formalized CHANGELOG format

New development model

- Development happens on master, bugfixes backported to maint-3.7
- next has been merged into master
- 3.8.0.0 will be tagged off master
- 3.7.x.x is tagged off maint-3.7

Retirement of maint, mergeback model
What will happen to the 3.7 series?

Stability & Maintenance

3.7 has been around for 5 years, with long stagnant periods

Lot of undocumented behaviour becoming implicit API

3.8 simply is *different* (in good ways)

- long-term commitment to support 3.7 on longer-term platforms (Debian stable, RHEL/CentOS, Ubuntu 16.04LTS)

But:

- No C++11, Python3, ... for 3.7.x.x: You probably won’t want to compile GNU Radio 3.7.13.4 on your bleeding edge Linux distro in a couple of years
What’s to come in GNU Radio 3.8?
Innovation and Future-Proofing

- Dependency cleanup: No choice, lots of benefits
  - Qt4 → Qt5
  - Cheetah/XML → YAML
  - log4cpp → own logging system

- Language progression
  - Py2 → Py2.7 OR Py3
  - C++98 → C++11

- Removed components

- New functionality
  - Better gr_modtool (shoutout to Swapnil!)
  - C++ code generation (shoutout to Håkon)
  - Overall cooler GRC
Releasing 3.8

When will we release 3.8?

When it’s ready.

I promise it no to be a minute later, but also not a (very expensive) hour early. We’re currently in a feature freeze to really focus on releasing 3.8.

Blockers:

- Modern CMake
- new OOT template (depends on modern CMake)
- logging system
- open bugs
- at least rudimentary documentation

→ https://github.com/gnuradio/gnuradio/projects/1
3.8 Release Procedure

1. Get the blockers sorted out
2. Tag RC1
3. Create Binary packages, spread
4. Fix blocking bugs that come up
5. Automated whole-tree code formatting
6. Release RC2, update binary packages
7. Give it 10 workdays to find remaining breakage
8. **Tag, sign, and release 3.8.0.0**
What’ll do once we’ve released 3.8?

3.8.x.x development continues on the maint-3.8 branch
  ▶ Bugs happen, let’s fix them

Feature development mainly on master → 3.9

3.7.13.4++ needs to be maintained, too.
Kicking off 3.9 development

Whole-tree code formatting officially terminates feature freeze

Use 3.8 as stepping stone towards a more modern future
... without breaking everything at once

3.9 will not take another 5 years to be released \(^1\)

---

\(^1\)paramount to everyone’s sanity
Where we’ll take GNU Radio

Be more inclusive

- GNU Radio should come with batteries included
- Selling point is the ecosystem, so let’s notice when that breaks
  → upstream central infrastructure
Where we’ll take GNU Radio

Be more inclusive

- GNU Radio should come with batteries included
- Selling point is the ecosystem, so let’s notice when that breaks → upstream central infrastructure

Be future-proof

- Track dependencies → deprecate earlier, clearer
- Remove old architectural burdens, replace with new ones
Where we’ll take GNU Radio

Be more inclusive

- GNU Radio should come with batteries included
- Selling point is the ecosystem, so let’s notice when that breaks
  → upstream central infrastructure

Be future-proof

- Track dependencies → deprecate earlier, clearer
- Remove old architectural burdens, replace with modern alternatives
Where we’ll take GNU Radio

Be more inclusive

- GNU Radio should come with batteries included
- Selling point is the ecosystem, so let’s notice when that breaks
  → upstream central infrastructure

Be future-proof

- Track dependencies → deprecate earlier, clearlier
- Remove old architectural burdens, replace with modern alternatives

Be easier to use

- Regular builds of binary packages for major distros
  - My dream: a package feed “zoo” for OOTs
- Better docs (shoutout to Marc Lichtman)
Where we’ll take GNU Radio

Be more inclusive

- GNU Radio should come with batteries included
- Selling point is the ecosystem, so let’s notice when that breaks → upstream central infrastructure

Be future-proof

- Track dependencies → deprecate earlier, clearlier
- Remove old architectural burdens, replace with modern alternatives

Be easier to use

- Regular builds of binary packages for major distros
  - My dream: a package feed “zoo” for OOTs
- Better docs (shoutout to Marc Lichtman)

Be better

- Make Scheduler exchangeable
- Remove shared state between blocks, use RPC
- Actually write a new scheduler
Ha! What of that will happen in 2019?

For 3.9, we’ll

- **Upstream:**
  - gr-soapy will become part of GNU Radio → we’ll “natively” support your favourite SDR
  - gr-iio will support ADI, esp. on SoCs
  - Networking
Ha! What of that will happen in 2019?

For 3.9, we’ll

- **Upstream:**
  - gr-soapy will become part of GNU Radio → we’ll “natively” support your favourite SDR
  - gr-iio will support ADI, esp. on SoCs
  - Networking

- **Future-proof:**
  - Remove Py2 compatibility
  - Depend on modern CMake, Boost, …
  - Replace PMT internally as well as our external serializer
Ha! What of that will happen in 2019?

For 3.9, we’ll

- **Upstream:**
  - gr-soapy will become part of GNU Radio → we’ll “natively” support your favourite SDR
  - gr-iio will support ADI, esp. on SoCs
  - Networking

- **Future-proof:**
  - Remove Py2 compatibility
  - Depend on modern CMake, Boost, ...
  - Replace PMT internally as well as our external serializer

- **Simplify:**
  - Have nightly release binaries
  - Include distribution package infrastructure for OOTs
Ha! What of that will happen in 2019?

For 3.9, we’ll

- **Upstream:**
  - gr-soapy will become part of GNU Radio → we’ll “natively” support your favourite SDR
  - gr-iio will support ADI, esp. on SoCs
  - Networking

- **Future-proof:**
  - Remove Py2 compatibility
  - Depend on modern CMake, Boost, ...
  - Replace PMT internally as well as our external serializer

- **Simplify:**
  - Have nightly release binaries
  - Include distribution package infrastructure for OOTs

- **Better ourselves:**
  - Have reliable RPC based on PMT successor + transparent transports
  - Finally remove VOLK from the main tree and treat it as dependency
Conclusion

- 3.7 development has reached a matured state
- 3.7 will still be supported as long as the typical platforms it runs on
- 3.8 is coming very soon™
  - C++11, Python3, Qt5, YAML
  - Cooler GRC, Code Generation, CMake
  - Want it faster? Fix some bugs!
- 3.9 is the next big thing
  - New upstreaming strategy
  - Getting rid of architectural debt
Questions?