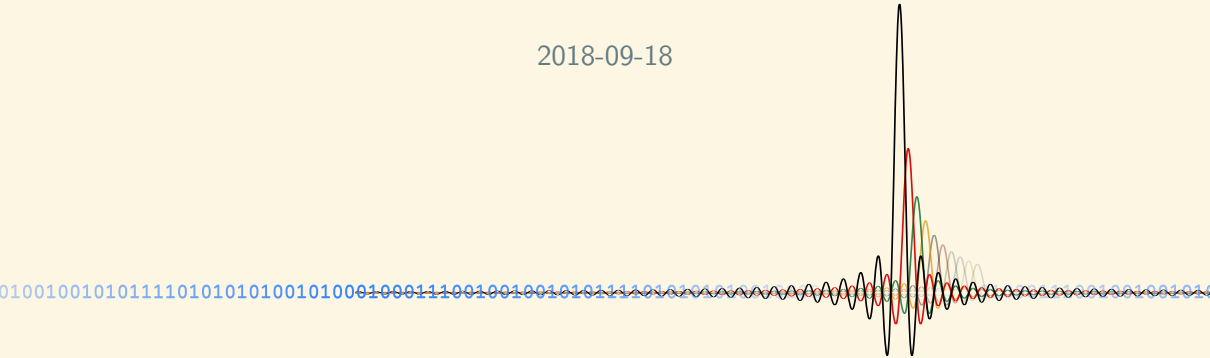


# GNU Radio Technical Update

Moving at  $c_0$ : Advancing GNU Radio

Marcus Müller

2018-09-18

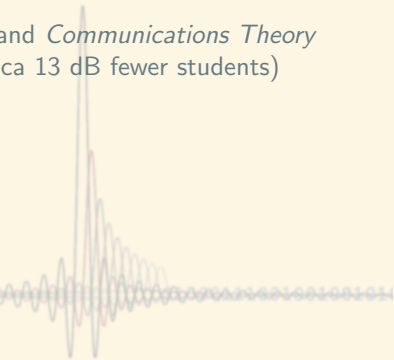




# Marcus Müller

Bearer of a couple of roles

- ▶  Support Grumpiness supplier
- ▶ Research assistant at  / 
  - ▶ 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
- ▶ Maintainer of the GNU Radio project



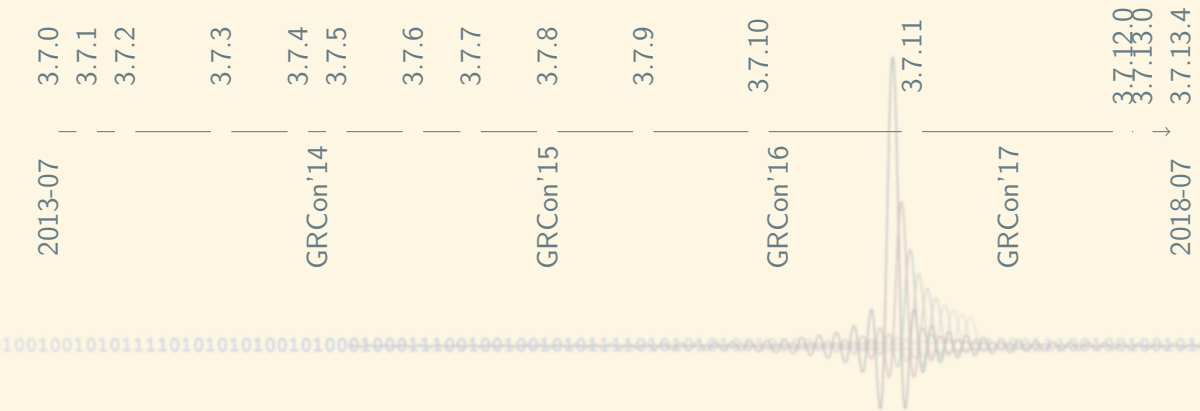




## State of GNU Radio 2017

GNU Radio 3.7 released June 2013

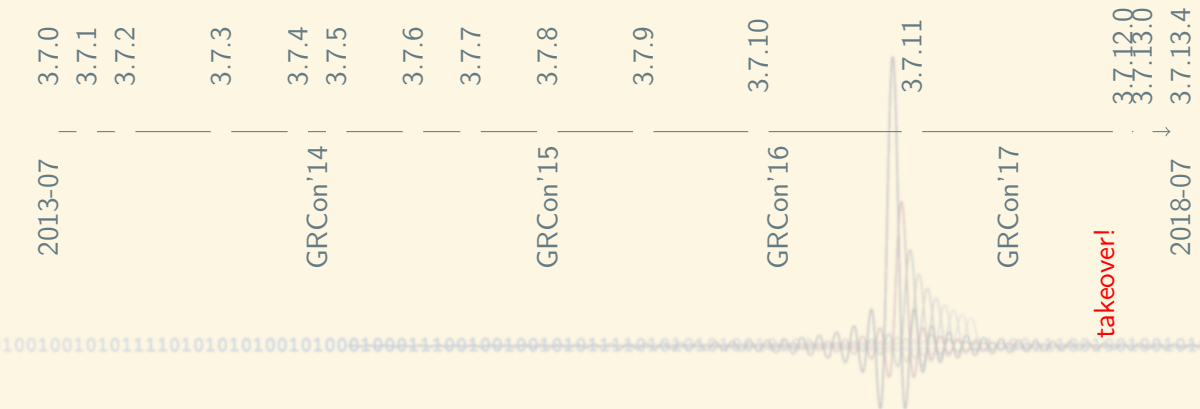
next branch forked off at that point



# State of GNU Radio 2017

GNU Radio 3.7 released June 2013

next branch forked off at that point







# Old Development Model

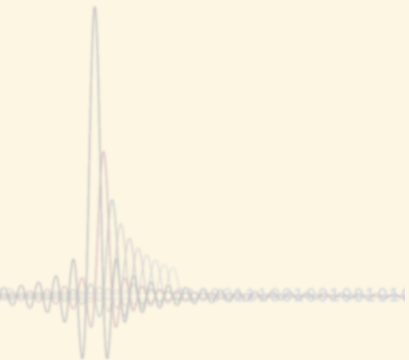
## The Mergeback Model

GNU Radio 3.7 released June 2013

- ▶ `maint`: Bugfixes
- ▶ `master`: Short-term / small-scope feature development
- ▶ `next`: Long-term / coming release (3.8) development
- ▶ `maint` regularly gets merged into `master`
- ▶ 3.7.X releases practically defined by those merges

## Changing dependencies

- ▶ develop for `master`, `next` XOR `maint`
  - ▶ forward development happening on `next`, starving `master`
- ▶ But: `next` relatively unstable
  - ▶ little chance of soon 3.8 releases



# Old Development Model

## The Mergeback Model


GNU Radio 3.7 released June 2013

- ▶ `maint`: Bugfixes
- ▶ `master`: Short-term / small-scope feature development
- ▶ `next`: Long-term / coming release (3.8) development
- ▶ `maint` regularly gets merged into `master`
- ▶ 3.7.X releases practically defined by those merges

## Changing dependencies

- ▶ develop for `master`, `next` XOR `maint`
  - ▶ forward development happening on `next`, starving `master`
- ▶ But: `next` relatively unstable
  - ▶ little chance of soon 3.8 releases

**Development slowing down**



## What has happened to 3.7 since then?

New versioning scheme: Semantic Versioning (<https://semver.org>)

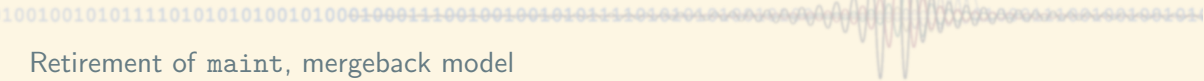
- ▶ MAJOR.MINOR.ABI.PATCH
- ▶ Supposed to make releasing quick and understandable

Formalized CHANGELOG format

New development model

- ▶ Development happens on `master`, bugfixes backported to `maint-3.7`
- ▶ `next` is being merged into `master`
- ▶ 3.8 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 has *exciting* new features and different dependencies

- ▶ 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















## Removed Components

- ▶ `gr-wxgui`
  - ▶ nobody knows how to maintain
  - ▶ bad computational performance
  - ▶ `gr-qtgui`: we have an option
  - ▶ But: `gr-qtgui` doesn't (yet) have full feature set
- ▶ `gr-comedi`
  - ▶ Does someone remember what that was used for?
- ▶ `gr-fcd`
  - ▶ needs hardware interface libraries
  - ▶ Should spin off nicely into an Out-Of-Tree (OOT) module
- ▶ `gr-atsc`, `gr-noaa`, `gr-pager`
  - ▶ Application-specific modules
  - ▶ Insufficient test coverage → Want us to get it back into the tree? Submit tests!
  - ▶ Should spin off nicely into an Out-Of-Tree (OOT) module



## Challenges To Face

- ▶ Fight the bitrot!
  - ▶ Unit testing
  - ▶ Integration testing
  - ▶ *Understanding* the scheduler
- ▶ Change of compute platforms
  - ▶ Thread-Per-Block Scheduler just lets the OS decide where and when to schedule (with no knowledge or understanding of the data flow whatsoever)
  - ▶ We've seen (simpleXecutive at GRCon'17) tremendous performance increases by *not using* arbitrary multi-threading for block scheduling
    - ▶ Limit number of threads to something feasible on a given platform
    - ▶ Something like single-threaded scheduling domains
    - ▶
  - ▶ Come and argue with me at the heterogeneous computing workgroup (Fri 08:45)
- ▶ Examples, Reference Designs
  - ▶ There's more than one way to do things, but some ways are better than others
  - ▶ DSP is something people know are taught how to deal with, software architecture . . . not so much
- ▶ De-screw-ify a lot of aspects
  - ▶ PMT is plain bad (no actual cross-language bindings, not actually portable, slow, unsafe, hard to use)



# Questions?

