

# Future of KDE Development

Dirk Müller  
Sebastian Kügler



<http://www.flickr.com/photos/cdm/54246114/>

# Current Limitations

- It's hard to share work in a team
- Patch review on mailinglists makes it hard to track changes and status (Allen Winter: "Has this patch actually been committed?")
- No offline operation
- Meaning of trunk/ changes all the time
- Patches are reviewed elsewhere (mailing list, review board, private email)

# Why not just switch to Git?

Git seems to be the right tool for us to move forward.

- CVS -> SVN is straightforward, SVN -> git is not
- Which "tree" is the official KDE flavor?
- How to keep track of what's going on?
- Estimated cost: ~1 year of reduced productivity
  - We want to be more productive !

# What if we switch to \$DVCS?

- What ensures that the code that is released is feature complete?
- Who ensures that code that is in the stabilization branch is getting stabilized?
- How can we provide playgrounds for developers to share their branches with others?

# KDE 4.0 development

KDE 0.0 - KDE 3.5: ~ 420.000 SVN revisions

KDE 3.5 - KDE 4.0: ~ 300.000 SVN revisions



09/2005: KDE 4.0 development starts

09/2006: Akademy talk: Most (?) attendees haven't seen the KDE 4 splash screen yet

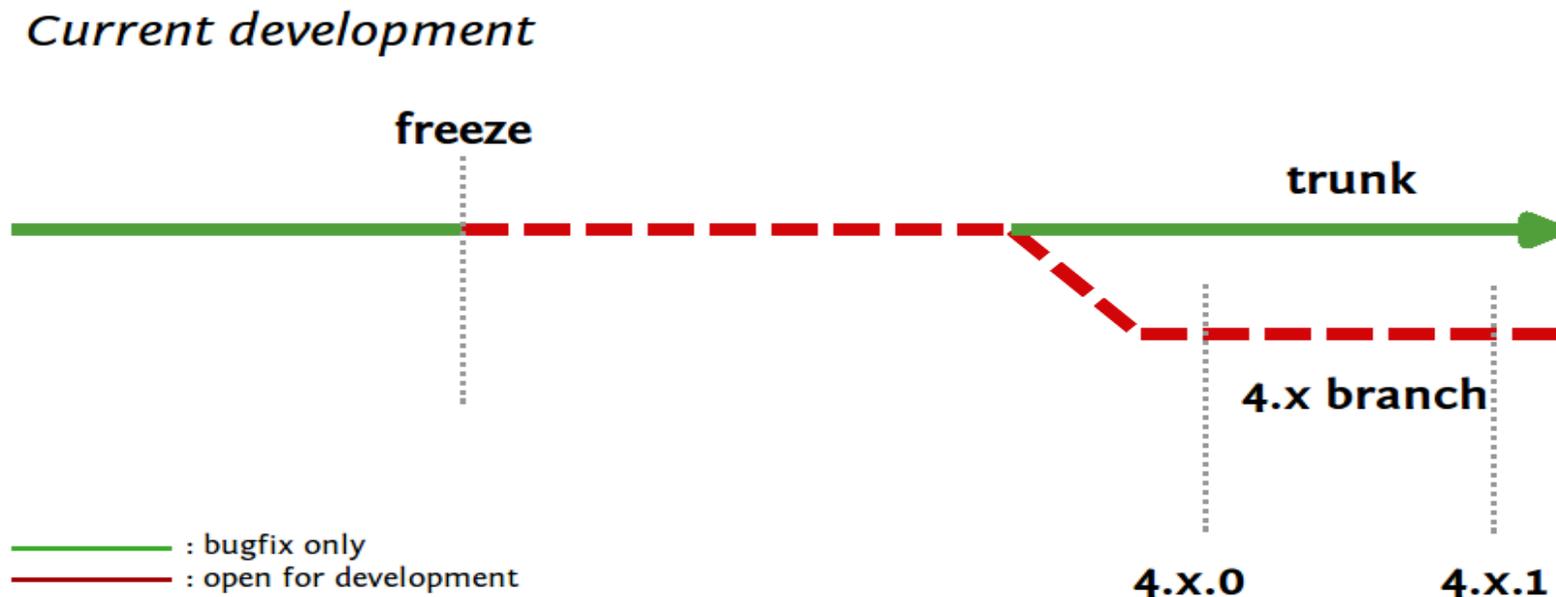
09/2007: Beta phase starts

01/2008: KDE 4.0

07/2008: KDE 4.1

# KDE's Current Development Model

- Centralized and Flat
- kde-core-devel and release-team
- Release cycle: 6 months, time-based: 4 months coding, two months freeze
- Distros "fork" KDE
- KDE development is easing into calmer waters



# A changing community ...

- Not every subproject is happy with a 6 month release cycle
- A growing, more complex community is more likely to feel those mismatches
  - individuals who having time in bursts ("Michael Pyne")
  - companies with contractual obligations (enterprise support, development contracts)
- "usable for some isn't usable for others" (enterprise, user, poweruser, developer)
- More disruptive changes in some parts, others are really stable

# ... in a changing environment.

- Growing, more diverse community
- New development models and tools supporting have appeared
- Developers want more powerful collaboration tools
- More integration of third party technology
- Software dependencies: kdesupport, Phonon
- It's not only OSVs that want to use KDE: KDE on Mac and Windows, mobile devices, ...

# KDE 4.0 Release Team



[http://www.flickr.com/photos/reg\\_00/763231398/](http://www.flickr.com/photos/reg_00/763231398/)

**Solution?**



# Agile Development

- **Power to the People** (and not to the Policies)
- Policies avoid chaos, but do not Guide

Individuals define KDE,  
Policies do not motivate



# Agile Dev. (II)

Working Software is important

- Deliver it as soon as possible, as often as possible
- Ensure that whatever has survived testing will be released
- Motivate users to work with the developer
- Motivate developers to work with users

# Agile Dev. (III)

## Rapid Development

- No process inducted slowdown
- Simplicity - The art of avoiding unnecessary work
- Reflect upon your workflow and share your experiences with others

# Agile Dev. (IV)

It's more important to adopt than to prepare.

- Any deviation from prepared plans is an emergency situation
  - Planning is impossible without fixed resources
- Adaptive development
  - Optimized for unplanned change

# Freezes?



and how about?

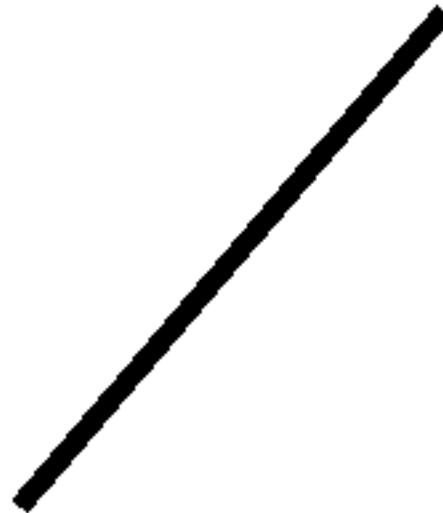
**Release cycle**

**!=**

**Development Cycle**

Urma User

David Developer



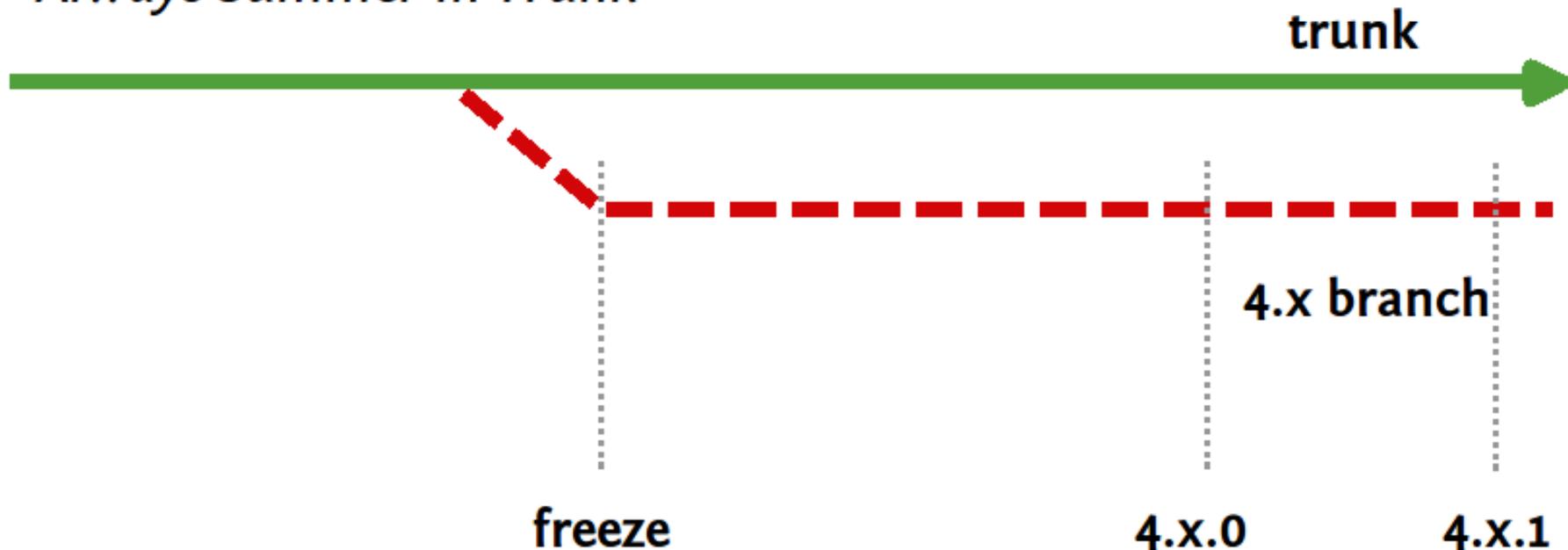
Tim Tester

# "Always Summer in Trunk?"

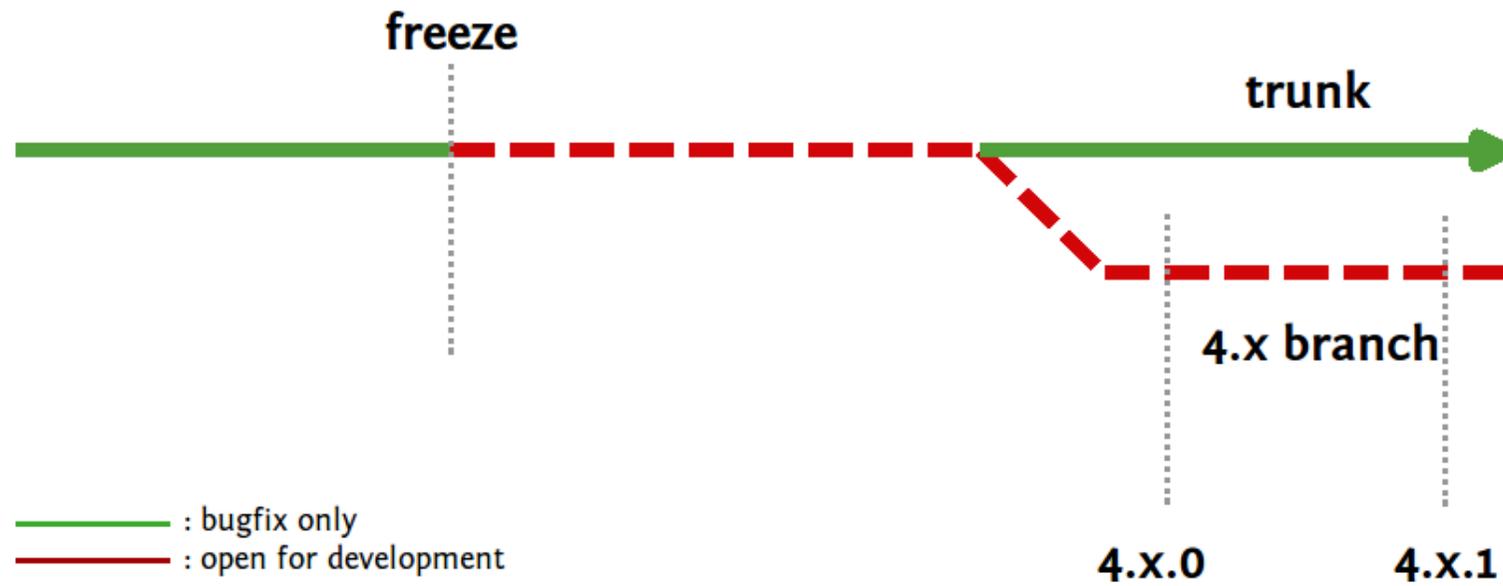
Possible solutions:

- trunk/ is always open for feature commits
- those commits should be 'reasonable stable'
- But: Translations for /trunk are never complete

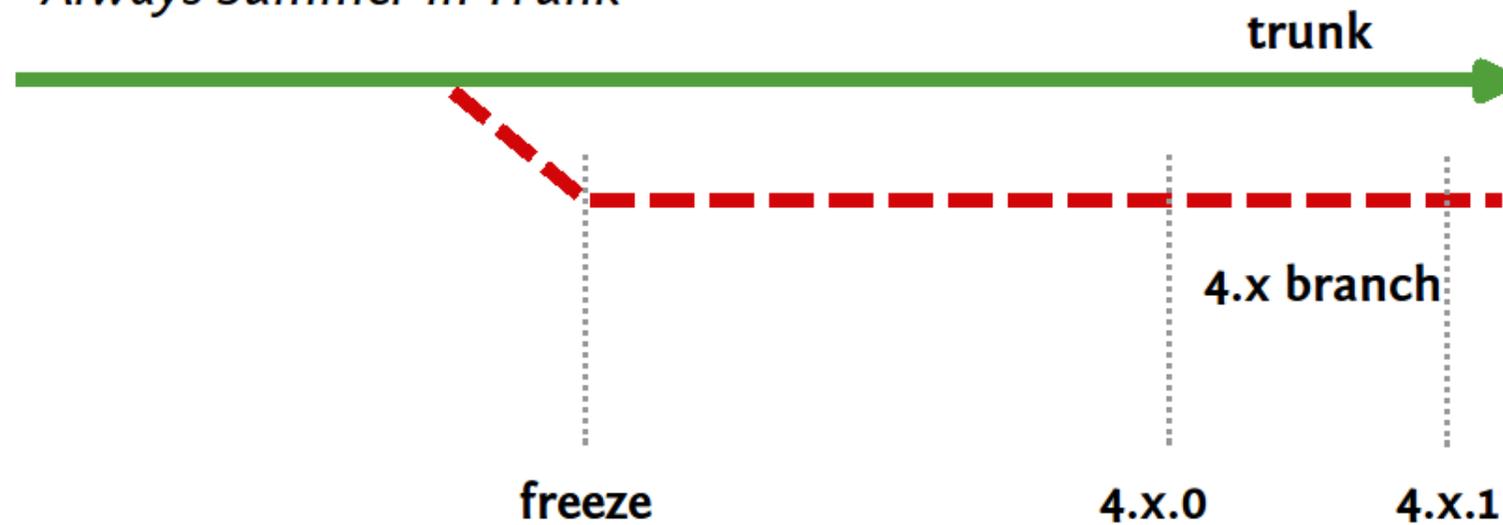
*"Always Summer in Trunk"*



## Current development



## "Always Summer in Trunk"



# Only one stone ...

- Publish milestone:
  - Exemplifies new features to the non-developers and developers
- Branch milestone:
  - Testers wanted, feedback cycle, stabilisation
  - Test suite? Disable incomplete features?
- Tested milestone:
  - stable, feature-complete, ready for user
  - ??delivered on time??

# Opportunities

- Create a more agile and scalable development process
- More possible workflows, support for 'team-based development'
- Open up the development KDE community
- **Getting ready for our Future**

# Questions / Discussion

We knew we wouldn't have any time left.

Let's schedule a BoF session.