
Mercurial Geek Night II

Using Mercurial in software product management

Gonzalo Casas
gc@isonet.ch

About the speaker

- Gonzalo Casas
 - Software Engineer at isonet since 2007
 - Maintains the Mercurial repositories
 - Manages the development process on top of it
 - BS in Computer Science from University Siglo 21, Córdoba, Argentina

Outline

- Introduction
 - Background
 - Terminology
- Before Mercurial
 - In the beginning...
 - The migration
- Current approach
 - Mercurial deployment
 - Development process
- Wrapping up



Mercurial Geek Night II

INTRODUCTION



Outline

- Introduction
 - Background
 - Terminology
- Before Mercurial
 - In the beginning...
 - The migration
- Current approach
 - Mercurial deployment
 - Development process
- Wrapping up

Background

- isonet ag is a software house founded in 1994
- Scrum for development process
- Three development teams
 - Two in Zürich, Switzerland
 - One in Leipzig, Germany
- Two *software products*
 - TicketXPert.NET
 - WorkflowXPert.NET

Background

- Our products are deployed to over 50 customers
- Releases must have long-term support
 - At least 2 years per release
- New features are required constantly based on short-term customer project
 - Usually less than 3/6-months projects
- Hot-fixes are required for builds of the product

Outline

- Introduction
 - Background
 - Terminology
- Before Mercurial
 - In the beginning...
 - The migration
- Current approach
 - Mercurial deployment
 - Development process
- Wrapping up

Terminology

- Software product management
 - “...managing software that is built and implemented as a product, taking into account life-cycle considerations and generally with a wide audience.” [Wikipedia]

Terminology

Build

- A compiled and packaged version of the software.
- The following are examples of our builds:
 - 3.6.0.0; 3.6.1.0; 3.7.9.0; etc.

Release

- Comprises a series of builds of the same code line.
- The following are examples of our releases:
 - 3.4.x; 3.6.x; 3.7.x; etc.



Mercurial Geek Night II

BEFORE MERCURIAL

Outline

- Introduction
 - Background
 - Terminology
- Before Mercurial
 - In the beginning...
 - The migration
- Current approach
 - Mercurial deployment
 - Development process
- Wrapping up

In the beginning...

- ...there was nothing (i.e. Source Safe).
- Three years ago, we migrated to Subversion.
- Things went well...until we branched.
- Merging effort greatly increased as we started building complex new features using feature branches.

In the beginning...

- Subversion doesn't store enough information to make merging simple.
- We waited for merge tracking support in Subversion 1.5, but it only made things worse.
 - Tracking information stored in properties caused merging conflict on meta-data.
- By the end of 2009, we spent 1 full month merging a feature branch...
- ...so we decided to change.

Outline

- Introduction
 - Background
 - Terminology
- Before Mercurial
 - In the beginning...
 - The migration
- Current approach
 - Mercurial deployment
 - Development process
- Wrapping up

The migration

- Migration task force
- The requirements:
 - Simple merging in complex branched scenario.
 - Support for offsite teams.
 - Good Windows support/client.
 - Lightweight maintenance.
- Alternatives:
 - Mercurial, GIT, Bazaar, Perforce, TFS, AccuRev, etc.

The migration

- Mercurial won the evaluation:
 - Community and enterprise support were crucial in selecting it.
 - The only real contender was GIT, but Mercurial was selected mainly due to Windows support.
 - Most others didn't offer all the required features, or the cost was too high.

The migration

- Migration from Subversion:
 - Using hg convert
 - Full history migrated
 - Almost out-of-the-box process:
 - Only addition was including SVN rev numbers in the commit message.
 - Migrated client from TortoiseSVN to TortoiseHG
- Developer re-education:
 - Internal trainings and cheat sheets



Mercurial Geek Night II

CURRENT APPROACH

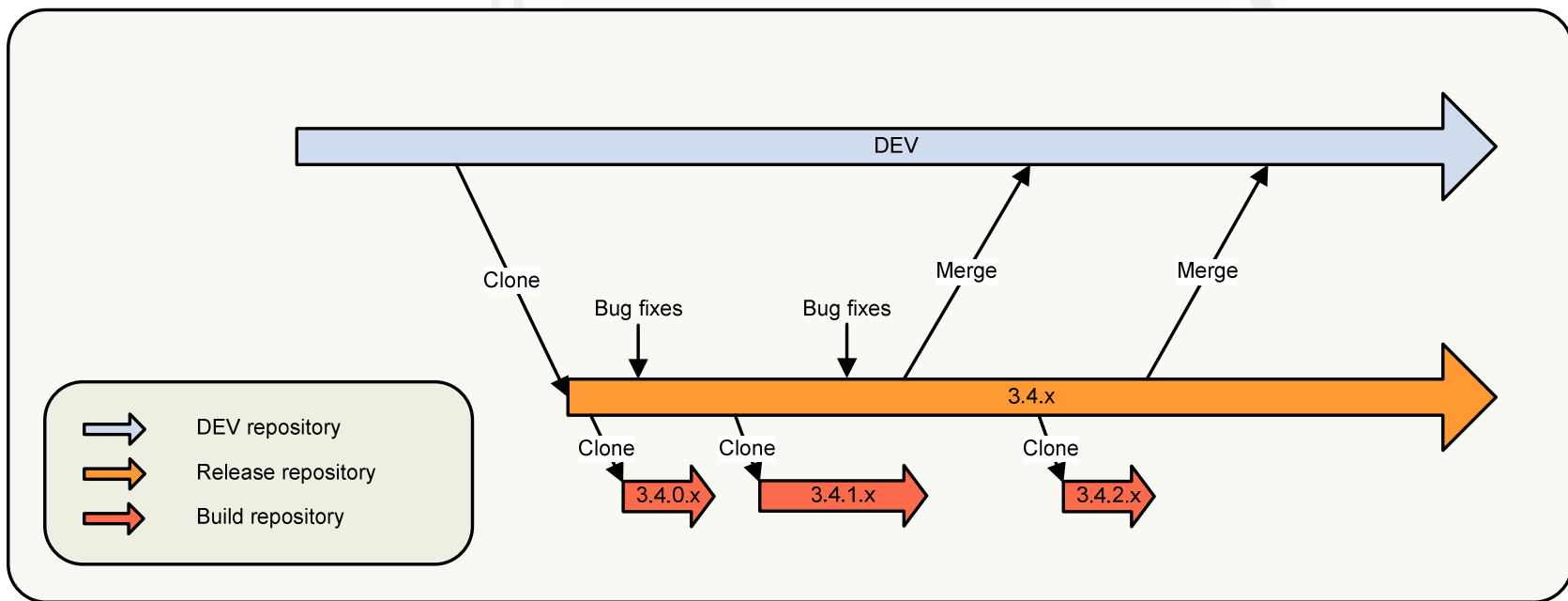
Outline

- Introduction
 - Background
 - Terminology
- Before Mercurial
 - In the beginning...
 - The migration
- Current approach
 - Mercurial deployment
 - Development process
- Wrapping up

Mercurial Deployment

- Mercurial servers:
 - Zürich: One central Mercurial server
 - Leipzig: Clones of main repos, sync'd daily
- Repositories are our main branching unit
- Basic repositories:
 - DEV ('trunk' in SVN terminology)
 - Releases repositories
 - Build repositories

Mercurial Deployment



Mercurial Deployment

- Work environment:
 - TortoiseHG on clients.
 - Perforce Merge as 3-way merge tool (free)
 - No IDE integration required.
 - Extensions in use: MQ, Rebase, Transplant.
- Push rights on server:
 - Restricted to Scrum Masters and QA managers.

Mercurial Deployment

- Some statistics:
 - DEV has around 85000 changesets
 - Each repository is about 1.4GB
 - Around 47000 files in the full solution
 - Between 5 to 10 big features developed in parallel at any time

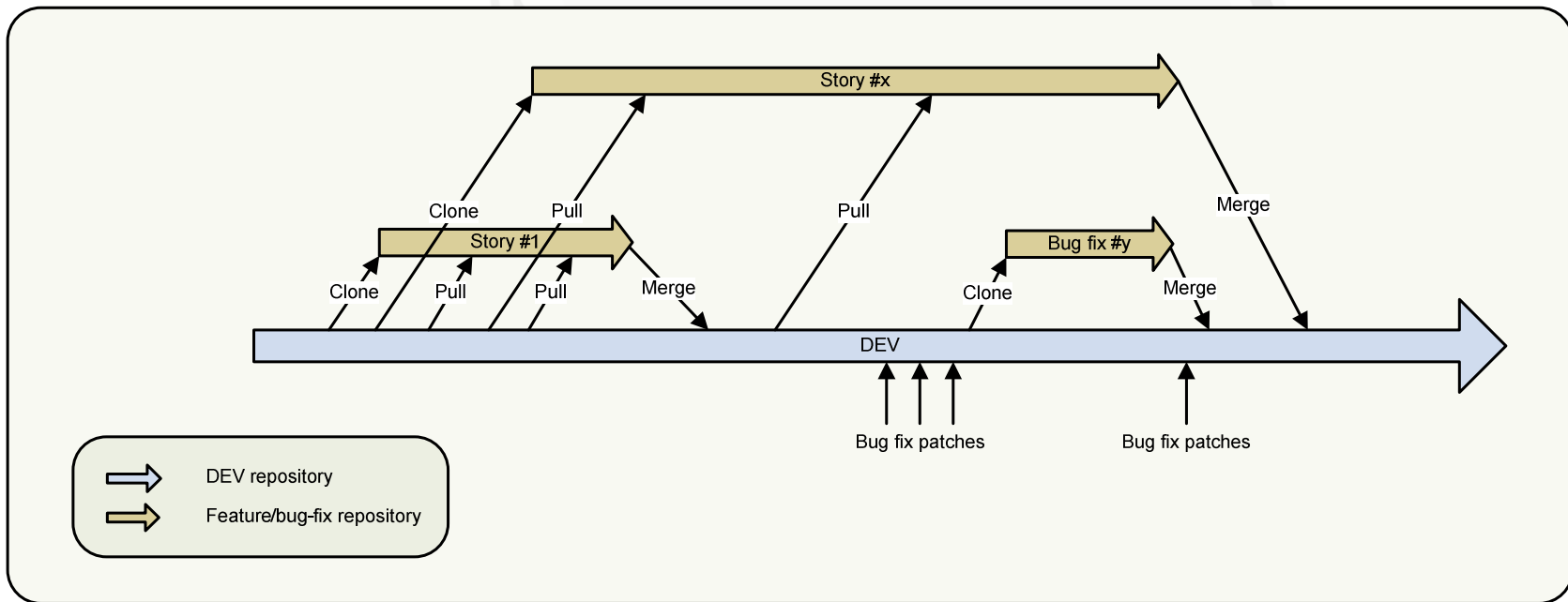
Outline

- Introduction
 - Background
 - Terminology
- Before Mercurial
 - In the beginning...
 - The migration
- Current approach
 - Mercurial deployment
 - Development process
- Wrapping up

Development process

- Where we develop new code?
 - If it is a small coding task: we use patches.
 - If it is normal/big: we create feature repository or bug-fix repositories.
 - For huge features, we use the concept of EPIC story repository.

Development process



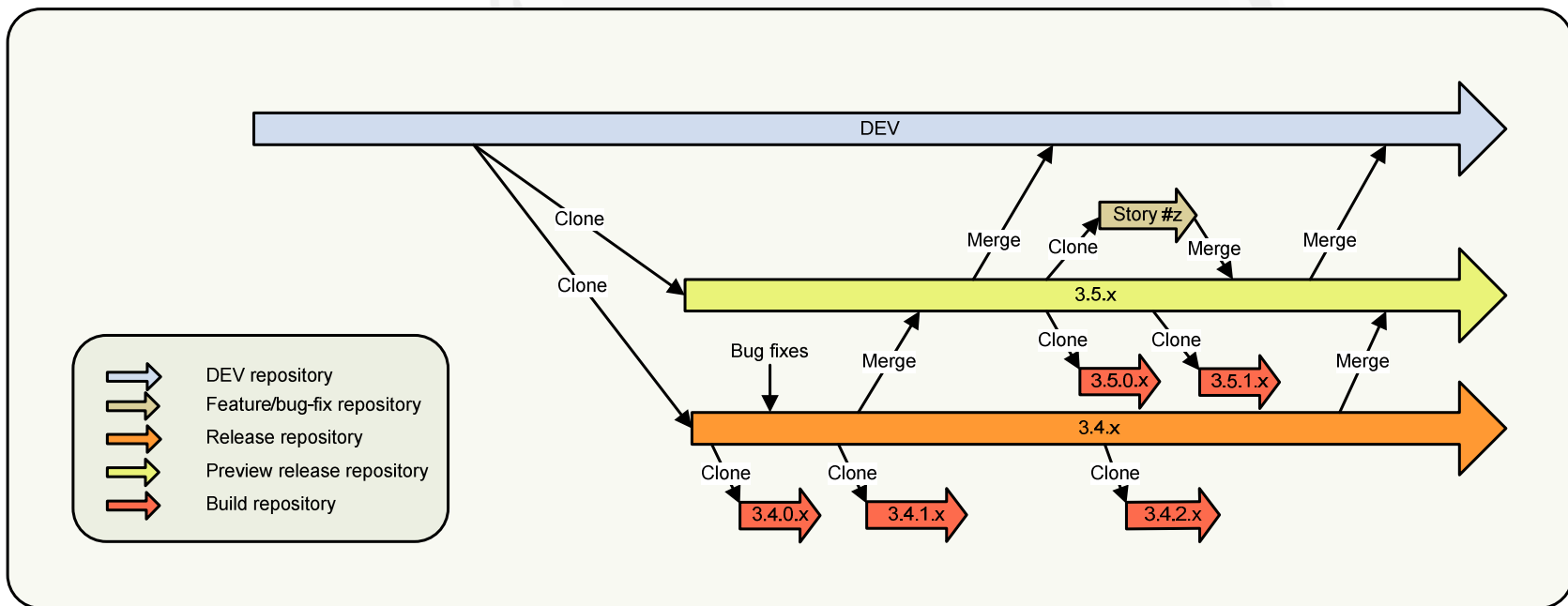
Development process

- Two types of releases:
 - Normal releases:
 - Only receive bug-fixes.
 - EVEN version numbers, *e.g.* 3.4.x.
 - Preview releases:
 - Cloned from normal releases.
 - Receive bug-fixes and new features.
 - ODD version numbers, *e.g.* 3.7.x.
 - Only one preview release alive at any given time.

Development process

- Full overview of repositories:
 - DEV
 - Releases repositories
 - Normal repositories, e.g. 3.4.x, 3.6.x
 - Preview repositories, e.g. 3.7.x
 - Build repositories
 - E.g. 3.4.0.x, 3.4.1.x, 3.6.1.x, 3.7.4.x, etc.
 - Feature repositories
 - Bug-fix repositories

Development process

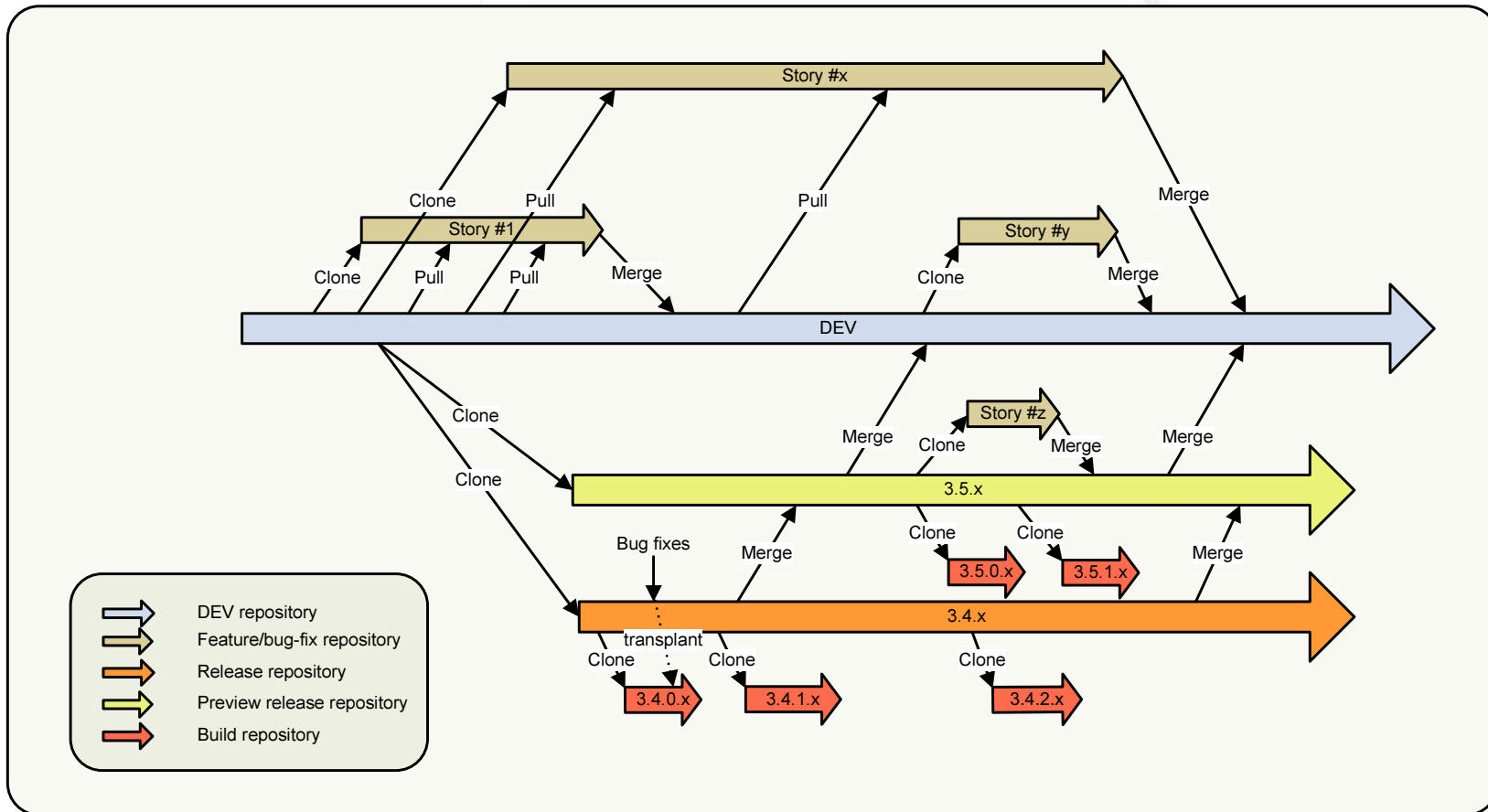


Development process

Hot-fixing:

- Using hg transplant to cherry-pick specific changes from releases into build repositories
- Versioning is increased on build repository

Putting it all together





Mercurial Geek Night II

WRAPPING UP



More information

- isonet homepage:
 - <http://www.isonet.ch>
- Mercurial tutorial by Joel Spolsky:
 - <http://hginit.com>
- Mercurial kick-start by aragost Trifork:
 - <http://mercurial.aragost.com/kick-start>

Contact

- Please get in touch if you have more questions:
 - Email: gc@isonet.ch
 - twitter.com/gnz

Q&A



Thank you!