A Critical Bug! SR2 Was Long Ago... Now What?

Steve Francisco (IBM)
Markus Knauer (EclipseSource)
EclipseCon Europe 2014
Who Would Need Fixes on Old Code?

Commercial products
• Paying customers expect fixes without being forced to upgrade
• Support durations is often 5 years or more

Custom development tools
• Some industries require supported tools for years, even decades

Teams locked into specific Eclipse versions
• Some won’t have a choice for the version they are based on
• Part of a larger project
• Providing add-ons to 3rd party software
What Could Go Wrong Anyway?

- Security bugs may require changes to plug holes
  - POODLE, Heartbleed, ShellShock, etc
- New OS versions arrive
- New hardware platforms arrive
- Shifting standards
- Government requirements change
Traditional Choices for Support

Participate in the projects you rely on
- Add committers
- Influence changes from within

Bring a branch into your own development team
- Control all changes
- No need to earn commit rights
- Still need to invest in development expertise

Keep current with project releases
- Adopt new releases frequently
- Can be disruptive

Hope it’s stable and won’t need much work
- Good luck with that....
Running Out of Options

Eclipse developers fix bugs against HEAD – backports aren’t routinely done

Expensive to apply fixes in unfamiliar code
  • Setting up the build environment
  • Maintaining the hardware
  • Understanding the code logic
  • Performing the right tests
How To Get Support

Two ways:

1. Pay an expert to provide you with support
   - Find a domain expert willing to sign a contract
   - Pay them a reasonable amount for the service

2. Do-it-yourself
   - Handle just the fixes you need
   - Not too expensive if the infrastructure is stable and reliable

Eclipse Long Term Support (LTS) makes both of these options possible.
What LTS Provides

LTS provides two important ingredients to the support story

1. A marketplace to connect users with service providers

2. A stable infrastructure for maintaining & building legacy Eclipse content
LTS Marketplace

• Users can find companies or individuals willing to provide support

• LTS does not dictate any terms, costs or rules

• Service providers can custom-fit a contract for each client

• Service providers are expected to get paid for their time and expertise

• LTS Marketplace is free to use, but the services that clients get are not free
1. Find a maintenance provider

2. Contact them with your needs

3. Negotiate terms
LTS Infrastructure

• The LTS Infrastructure consists of:
  • Git repository
  • CBI build with Hudson front end
  • Signing services for bundles, Win & MacOSX
  • Full time engineer to maintain systems

• Only LTS Members have access to the infrastructure

• Membership is not free – annual dues are paid to cover operating costs
LTS Workflow

Bug reported in bugs.eclipse.org

Fix the bug
Push changes to LTS Git

Merge changes after Gerrit review

Build with Hudson using CBI

Deliver fixes in products
Making Fixes

Maintenance Committers
• Can be named by LTS Members
• Can commit changes to LTS Git
• Do not need to be committers on the project they support
• Eclipse committers are automatically maintenance committers in LTS

Change Management
• Changes in company forges do not need approval (PMC or otherwise)
• Changes in LTS-Central require Gerrit review to be merged

• Since LTS Git is not public, changes must also be made available in another way (Bugzilla attachment, etc) to satisfy EPL
Any Project? Really?

Projects can be included in LTS if they meet these criteria

- Source code is available in a Git repository
- Uses Maven or Tycho for build
- Build scripts are available in case other build tools are used
Projects Currently In LTS

Eclipse Equinox
Eclipse Project
  + JDT Project
  + PDE Project
Eclipse RAP
How Are Builds Done?

• Hudson CI manages builds

• Currently LTS Central builds are done every 6 hours

• Private forges can be built on demand
What Have We Got So Far?

- A safe Git repository for source control
- Authority to make code changes
- Ability to build on demand
- Freedom from maintaining all of the above

- Everything you need for supporting code... except the domain expertise
Domain Expertise

All of this is useless without the know-how to fix a bug.

If you are comfortable making fixes, then **Self-Service** is for you. Make any changes you need and let LTS store and build them. *(LTS Membership required)*

Those without the know-how can pay a **maintenance provider** for the service. LTS Marketplace connects you with the domain expertise. *(No LTS Membership required – your maintenance provider is a member)*
Example of LTS in Action

Timeline of a Fix for Bug 424118

2014-01-29 17:51  Production critical bug 424118 in RAP 2.1.0
2014-01-29 18:45  RAP team finishes analysis
2014-01-30 11:58  First version of bug fix pushed
2014-01-30 17:07  Second version for corner cases pushed
2014-01-30 17:13  LTS build finishes successfully
2014-01-30 17:41  Fix delivered to customer

The fix is now available to the public with RAP 2.1.2
Business Opportunities

Are you a subject matter expert? Ready to make some money?

**Become a maintenance provider** listed on the LTS Marketplace!

Looking for support so your project can remain viable in the long term? Need a safety net?

**Find a maintenance provider** at the LTS Marketplace!
Who Is Steering LTS?

The LTS Steering Committee Members

[Logos of various companies such as CA Technologies, IBM, INNOOPRACT, PolarSys, and SAP]
Links & Resources

LTS web page: http://lts.eclipse.org
LTS Charter: https://www.eclipse.org/org/workinggroups/ltsiwg_charter.php
Contact us: lts@eclipse.org
Mailing List: lts-iwg@eclipse.org

All photos copyright Elena Elisseeva – http://www.elenaphoto.com