SAP and the Eclipse LTS program

Walldorf, July 11, 2011

Karsten Schmidt

Jutta Bindewald

Thomas Rastetter

Michael Schuhmacher

Jens Hänelt

Introduction

The presentation will focus on ...

- ... the challenges we see in supporting the **Eclipse** OSS components we are using within our products
- ... how the Eclipse LTS program may be a means to tackle these challenges

It will NOT cover ...

- ... maintenance / support for OSS in general
- ... any historical developments
- ... the various use cases and scenarios
- ... the SAP-internal decision processes
- ... the LTS program itself
- ... the Business Model

The basic problem

SAP products are supported for at least 9 years

A tricky ménage à trois

SAP
open source
eclipse

Slides from EclipseCon 2010

General Challenges

Challenge / requirement	Addressed by Eclipse LTS program?
Only committers can check in code • Impossible to have committers in all used projects • Project teams not interested in fixing ancient versions • No guarantee that critical (for SAP or our customers) bugs are fixed	**Maintenance Commmitter** concept
Costs and complexity of infrastructure • Infrastructure: source control, build, tests, signing, source / binary version management, supported (historical) platforms • Forking and replication of heterogeneous infrastructure is complex and expensive	Provisioning of a **central shared homogeneous infrastructure**

General Challenges (2)

Challenge / requirement	Addressed by Eclipse LTS program?
Availability of source code and patches • Fixing a bug twice is waste – no matter for whom the first fix was made	All source code will be **publicly available** (but binaries only to participating parties)
Legal requirements • Modifications require publishing and license additions • IP cleanliness	Code available under **EPL, IP Process**

Challenges of a 3rd party approach

Challenge / requirement	Addressed by Eclipse LTS program?
Fragmented expertise • Companies specialize on few projects / technologies • Leads to many small support contracts with many small companies • Not feasible for complex SAP support infrastructure	**„Maintenance Service Provider“** layer
Vendor lock-in • Vendor may own all patches • Infrastructure behind closed doors	**Code** available under **EPL, Binaries** under **EBL, Shared Infrastructure**
Meeting the SLAs / handling emergency situations • In very critical cases 3rd party vendor may not be able to be fast enough (?)	**Infrastructure is available to SAP itself**

Open Questions

Challenge / requirement	Addressed by Eclipse LTS program?
„Critical“ bugs / fixes • Example: Security Vulnerabilities • Bug and fix must be hidden until it is active at the customer sites	Some concept like (temporarily) **„hidden branches / build results / bugs / Git clone …“**? Addressed by Eclipse Security Policy?
Merging of branches • Vendor specific branches will over time become un-manageable • Discovery and sharing of patches will become difficult • Regular merges into something like 3.6.x are needed	To be discussed

Next steps?

- **Clarify the requirements of all interested parties**
- **Clarify initial funding and reimbursement models**
- **Clarify business model for all participants (customers, suppliers, Eclipse Foundation)**
- **Get commitments from customers and suppliers**
- **Define time plan**

Thank you!