Software Configuration Management
Issues with Industrial Opensourcing

Lars Bendix
Department of Computer Science
Lund University
Sweden

Jan Magnusson
Sony Ericsson
Lund
Sweden

Tero Kojo
Nokia
(FinPro)
Helsinki
Finland
What is SCM?

Traditionally:
• configuration identification
• configuration control
• configuration status accounting
• configuration audit

Pragmatically (Wayne Babich):
• co-ordinating a team
• communication in a team
Issues with OSS

SCM-related issues from [14] and [15]:
• custom changes need to be maintained
• difficult to get changes accepted
• backward compatibility concerns
• familiarize with submission process
• maintenance effort
• creating forks/variants
• “upgrading” strategy
Types of involvement

Service:
- “passive”
- well specified requests
- little code (short time)

Development:
- “active”
- open/fuzzy requests
- much code (long time)

Owner:
- designer/architect
- CCB chairman
- technical integrator

As-is usage:
- glue code
- updates

Modified usage:
- modified code
- many/few changes
- updates
Experience and strategies

Service:
• ease of setting up (test & code review)
• ease of providing feedback (bug reports)
• ease of contributing (bug fixes)

Development:
• ease of communication
• ease of collaboration (co-ordination)
• ease of contributing (commit vs. patch)
• separation of logical tasks
Ways of working

a)

b)
Experience and strategies

Owner:

• set-up for low “entrance fee” (tools and processes)
• support for CCB (communication and decision)
• provide a designer/architect
• provide/support a technical integrator
• bottlenecks impede continuous/quick integration
Experience and strategies

As-is usage:
• update strategy

Modified usage:
• “update” strategy
• minor ("stable") modifications
• major modifications
• (not) contributing back
Ways of “updating”
Discussion questions I

How important is (the capability of ) the VC tool?
What is the real problem(s)? Commit vs. patch?

How important is the (right) process?
Can it be enforced?

Why do/don’t companies contribute changes?

Should Linus Torvalds stop being schizophrenic?

How to handle small/large refactorings?
Discussion questions II

Does co-ordination change in a distributed set-up?
If yes, why and how?

How to substitute physical vicinity/proximity in the communication?

Is there a CCB? If so, what is its role and who is “on it”?

How can people communicate through the repository?
And what can/cannot they communicate?

What is the difference between OSS and Agile?
Concluding remarks

Setting up:
• low “entrance fee”
• moderator

Participating:
• create patches
• keep tasks distinct
• keep in sync

Using:
• contribute – or sweat