



A pattern in the making: Bridging the SCM maturity gap

”Keep your patterns close, but your anti-patterns closer”

Jonas Bovin, ALOC A/S

Pattern form



IF you find yourself in CONTEXT
for example EXAMPLES,
with PROBLEM,
entailing FORCES

THEN for some REASONS,
apply DESIGN FORM AND/OR RULE
to construct SOLUTION
leading to NEW CONTEXT and OTHER PATTERNS



- Heterogeneous Software development Team
 - Developers with a background in finance, banking or economics
 - Coders
 - UI-designers and Usability experts
- Without correct use of SCM
 - High workload, highly inefficient
 - Error prone
 - Two people on every integration



- Team unaware of unused SCM potential
- Team unable to communicate about SCM
- Low level of developer confidence in Perforce use
- Progress is slow
- Results vary



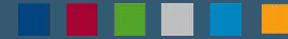
- Due to the high workload, time consuming training was out of the question
- Some sub-projects had risks that had to be handled
- The team was aware of the symptoms and was actively seeking help - they did not know what to ask for though
- Team used SCM only as a revision repository

Reasons



- Patterns for SCM are easier to grasp than the underlying implementation
- The very idea behind a pattern language is to help communication across professions
- If you cannot fill the gap - bridge it...
- P4V and P4Win can be extended

Form/Design

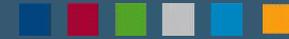


- The existing P4V tools could be used as a platform for development
- Added extra information to paths through XML
- Added XML-policy constraint language
- GUI showing solely patterns' actions or policy's content
- Visually aided actions
- Terminology / vocabulary subset of patterns



- A context sensitive P4V integrated visual tool
- Can be launched from path, changelist or job (context)
- Depending on context different patterns / actions are available
- Pattern description shown
- Clear indication of what each function does

The Pattern Plug-in



perforce:1667, jonasb - ALOC Pattern Integration

File Help

Available Patterns

- Software Configuration Management
 - Branch patterns
 - Task Branch
 - Project Branch
 - Development Line
 - Policies
 - devel/interim
 - devel/interim5
 - devel/main
 - devel/main5

Description

Context

You have an evolving codeline that has code intended to work with a future product release. When you are working in a dynamic development environment, many people are changing the code. Team members are working toward making the system better, but any change can break the system, and changes can conflict. This pattern helps you balance stability and progress in an active development effort.

Problem

How do you keep a rapidly evolving codeline stable enough to be useful?

Solution

Institute policies that are effective in making your main development line stable enough for the work it needs to do. Do not aim for a perfect active development line but for a mainline that is usable and active enough for your needs.

Context

P4 'depot' path: `///depot/alloc/products/tms/programs/devel/interim`

Create Development Line from Main Line

Catchup changes from Main Line

Catchup

Catchup changelist from Main Line

Integrate completed package to Main Line

Integrate package

Deliver completed package to Main Line

✓ Resolve accept theirs

Deliver package

ALOC SCM Pattern Integration for P4

The Pattern Plug-in



The screenshot shows the 'ALOC Pattern Integration' application window. The title bar reads 'perforce:1667, jonasb - ALOC Pattern Integration'. The interface is divided into several sections:

- Available Patterns:** A tree view under 'Software Configuration Management' showing 'Branch patterns' (Task Branch, Project Branch, Development Line) and 'Policies' (devel/interim, devel/interim5, devel/main, devel/main5). 'devel/interim' is selected.
- Description:** A text area containing:

Policy for devel/interim

Integration branch

Owner

Martin Hyldgaard (martinhy@aloc.dk)

Restrictions

integration from //depot/aloc/products/tms/programs/devel/main/... is allowed

integration from //depot/aloc/products/tms/programs/devel/interim5/... is allowed

Goal

Eventually all changes to this line should be integrated to //depot/aloc/products/tms/programs/devel/main/
- Context:** P4 'depot' path: //depot/aloc/products/tms/programs/devel/interim
- Policy for devel/interim:**
 - Title: devel/interim
 - Description: Integration branch
 - Owner: martinhy
 - Restrictions: Disallow all actions except for:
 - Action/From table:

Action	From	
integration	//depot/aloc/products/tms/programs/devel/main/...	Delete
integration	//depot/aloc/products/tms/programs/devel/interim5/...	Delete
 - Buttons: + Add restriction, Save policy, Revert Policy
 - Based upon (pattern): Development Line

ALOC SCM Pattern Integration for P4



- Users benefit from SCM with minimal effort
- The team eagerly explores more areas to improve upon
- When their needs change they can describe what they need to change
- All of the team members can catch-up, merge and branch
 - even baseless merges

The future of Pattern Plug-in



- Server trigger checking for policies
- Process handling
- Re-implementation using P4 WSAD

Questions?

ALOC



E-mail: jonasb@aloc.dk