Policy Implementation Strategies Using ClearCase

- Policy Implementation Strategies
- Kinds of Policies
    - Process Shepherding
    - Access Control
    - Entry & Exit Quality Criteria
    - Automatic Communication
- Policy Interactions
    - Miscable Policies
    - Selectable & Defeatable Policies
    - Project Specific Policies

Copyright 1995 Howard Cohen

413 Central Avenue
Alameda CA 94501
(510) 521-0124

http://www.timefold.com

Policy Implementation Strategies: The Big Picture

- You can meet more of your business needs by implementing some policies using ClearCase facilities.
- Policies should be designed to help people avoid making mistakes, and thereby should improve quality and minimize wasted time.
- Learn how ClearCase works and use built-in facilities with skill before crafting a new methodology on top of ClearCase. Learn how Atria uses ClearCase.
- Every policy you implement through ClearCase will affect performance; some can have a drastic effect.
- You should maintain a policy manual which defines the policies, including their benefits, and what one must do to subscribe or unsubscribe.

Copyright 1995 Howard Cohen

413 Central Avenue
Alameda CA 94501
(510) 521-0124

http://www.timefold.com

Policy Implementation Strategies: The Technology Side

- Standardize how policy implementation is managed, so that it is easier to develop and maintain policy software.
- Establish meta-data and view naming conventions to simplify policy implementation.
- Strive to implement policies such that they don't interfere with each other.
- Focus first on policies which protect the codeline, like triggers to protect the `cleartool rmelem`, `rmbranch` and `rmver` commands.
- Communication policies (automatic email) are easy to implement and make the system more helpful quickly.
- Housekeeping policies can be added when the tasks are clear; nothing will make them clearer than doing them by hand for a while.
- You can apply policies to objects and to object types.

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124
http://www.timefold.com

Policy Implementation Strategies: The Human Side

- Policies which are perceived as enforcement will meet with more resistance than those perceived as assisting, facilitating, and tracking.
- Communicate the benefits of policies, not just the rules.
- Some policies which could be implemented through ClearCase work equally well based on personal discipline, when work is subject to peer and management review.
- End-users won't know where ClearCase ends and the policy implementation begins. Starting with few policies increases the perceived robustness of the combined system in general.
- The most powerful tools are words: having everyone understand and be able to discuss codeline management concepts will empower people to use ClearCase better, and with fewer policies needed to help them.

Copyright 1995 Howard Cohen

413 Central Avenue
Alameda CA 94501
(510) 521-0124

Kinds of Policies

- There are a few kinds policies you might want to implement:
  - Access Control
  - Process Shepherding
  - Entry/Exit Quality Criteria
  - Automatic Communication

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124

Access Control

Definition: **Access control** means limiting who can perform what operations.

Example:

*During the merge we only want **jen**, **paul** and **don** to be able to make changes in the rel2 project.*

- You can lock the REL2_BASE label and the rel2 branch type except for jen, paul and don.
- Alternatively, a less restrictive approach is to create a branch-type trigger to prevent checkins except for these individuals.

Example:

We don't want anyone but our codeline administrator to be able to remove elements, branches or versions.

- You can create a global trigger which rejects these operations unless the user is the codeline administrator.

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124

Process Shepherding

Definition: **Process shepherding** means that the system is taught about a process, and won't allow missing or out-of-order steps.

Example:

Suppose a bugfix process includes steps to get a review before files can be checked in.

- You can create a global trigger which refuses checkins unless a "Review is done" attribute is present on the checked-out file.
- You could create a global trigger which prevents a person from applying the "Review is done" attribute to their own checkouts.
- It would be nice if you supply a simple tool to apply the "Review is done" label to a checked-out file.

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124

Entry/Exit Quality Criteria

Definition: **Quality criteria** are hurdles you set up to keep defects from entering a product. The criteria must be met before some operations are allowed.

Example:

Code must lint cleanly before it can be checked in.

- You can create a checkin trigger which runs lint and rejects the checkin if lint fails.

Example:

Meta-data types shall subscribe to the conventions and standards required by the policy software.

- You can create preop type-triggers to examine the names and reject those that don't subscribe.

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124

Automatic Communication

Definition: You use **automatic communication** to spontaneously give people the information they need to succeed.

Example:

All concurrent users of a file within the same branch shall be notified that others are using the file (including who and why).

- You can create a checkout trigger which uses cleartool lscheckout to find all concurrent users of the file (on the same branch type), and then composes a suitable email message.

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124

Miscable Policies

Definition: Policy implementations which can coexist with others are said to be **miscable**. Policy implementations are only miscable through discipline & adherence to standards when the policies are implemented.

Rules for miscable policies:

- Preops should only perform analysis and should exit non-zero to reject an operation.
- Preops should never modify meta-data or take other actions which could require “undoing”.
- Postops may manipulate meta-data.
- All policy implementations should have a simple means to disable them.

Benefits of miscable policies:

- Easier to debug policy implementation.
- No surprises due to trigger firing order.
- Simple model helps keep policy implementations from becoming too complex.

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124

Selectable & Defeatable Policies

Definition: A **selectable** policy is **off** by default, but can be enabled if desired.

Definition: A **defeatable** policy is **on** by default, but can be turned off if desired.

- It is easy to make triggers selectable:

```
#!/bin/sh
if [“$FOOPOLICY_ON”!= “Y” ]
then
        exit 0
fi
```

... remainder of policy trigger code, off by default

- It is easy to make triggers defeatable:

```
#!/bin/sh
if [ “$BARPOLICY_OFF” = “Y” ]
then
        exit 0
fi
```

... remainder of policy trigger code, on by default

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124

Project-Specific Policies

Definition: A **project-specific policy** is a policy that applies to one or more projects (or streams or branches, etc.), but not all.

- Not all policies are always appropriate. Sometimes it makes sense to defeat or enable a policy only during a critical phase for a given work effort.
- Global triggers can be coded to examine the branch type of a file (or a view name -- even better) to decide whether to take effect.
- You can attach triggers directly to branch types, and even instances of branches if needed, but this can be hard to manage except for simple cases.
- Don't code project/branch/stream names into triggers -- use a file or meta-data to create "data-driven" solutions (which are configured by manipulating data as opposed to changing code). Document how the data affects the tools.

Copyright 1995 Howard Cohen
413 Central Avenue
Alameda CA 94501
(510) 521-0124

Timefold(tm)

- Owned and operated by Howard Cohen.
- Offers focused training in the concepts and strategies of using ClearCase.
- Compliments but does not replace the excellent tactical training offered by Atria Software, Inc.
- Seminars include:
    - ClearCase Fundamentals
    - Project Concepts and Strategies
    - Advanced ClearCase Administration
    - Policy Implementation Strategies
- For more information:
    - http://www.timefold.com (Timefold(tm) Web)
    - info@timefold.com (Automated reply)
    - hoco@timefold.com (Howard Cohen)
    - (510) 521-0124

Copyright 1995 Howard Cohen

413 Central Avenue
Alameda CA 94501
(510) 521-0124