





SCM Rules of Thumb



Piccolo Engineering, Inc.

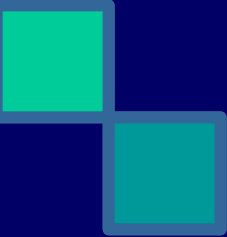



Find an expert; make them as necessary.

- ✍ Who's the person who makes sure you're playing to the strengths of the software?*
 - ✍ At least one person should RTFM from cover to cover once.*
- 

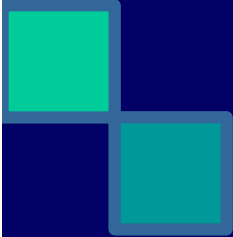



Basic things to train for...

- 
- ✍ *Basics for an administrator: setup, backup, testing restores.*
 - ✍ *Basics for a developer: be able to explain how do “add/delete/edit” a file and how to avoid destroying other people’s work.*
 - ✍ *Basics for a project lead: know how to make/deal with codelines and branches.*
- 








Keep the “daily grind” simple

- 
- ✍ You should know the basics of your configuration. (“Working on //depot/main/... for new release work”)*
 - ✍ Know, off top of head, commands to add/edit/delete/submit/revert a change, and how to avoid stomping other folks’ work.*
- 




Use default configurations when possible.

-  *Use SCM primitives for NT/Unix “shared” work – don’t outsmart it.*
 -  *Type real change descriptions, not just “another change” or “fixed something”.*
 -  *Don’t complain if SCM system locks at submit-time instead of checkout-time. (Read enough to understand why.)*
- 




Set up new development projects (and releases) correctly

- ✍ Pick new pathnames carefully;*
 - ✍ Use SCM primitives ("p4 integrate") to create new codeline.*
 - ✍ If it's too hard, get help, read a manual, or e-mail Tech Support.*
 - ✍ Don't even start work unless the backup/restore strategy is robust.*
- 



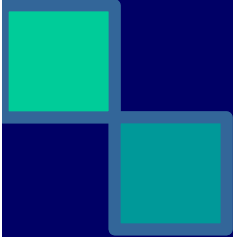



For god's sake, don't...

- ✍ Don't create a branch using the "add" operation.*
 - ✍ Don't destroy other people's work.*
 - ✍ Don't type in something like "modified a file" into a change description.*
 - ✍ Don't make the same change in two places.*
 - ✍ Don't wait until the crash to test your "restore" strategy - and make sure it addresses a "Loma Prieta" surprise.*
- 




For god's sake, don't...

-  *Don't delete revisions.*
 -  *Don't make a check-in contain more than one unit of functionality or one bug fix.*
- 
- 




Know the foibles of the tool you're using.

- ✍ *ClearCase and Perforce have a two-step "label" operation. ("Oops")*
- ✍ *Perforce doesn't lock a file on "edit", but at submit time.*
- ✍ *Labels are handy.*

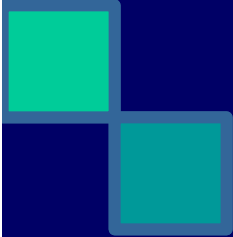


When things are “too” hard,
ask if it’s you or the tool.

- ✍ If merging changes is a horror show, you’re either using the tool improperly or you’re using VSS.*
 - ✍ If restoring from a backup is time-consuming or error-prone (or doesn’t work), change your backup strategy.*
 - ✍ Type in change descriptions. Don’t be lazy.*
- 



Pay for a shoulder to cry on

- 
- ✍ Call the Tech Support folks for this product and figure out what their strengths and weaknesses are.*
 - ✍ Do this before your big emergency.*
- 