

## Basic Project Skills

- Change Control
  - goals, issues, approaches
- Version Control
  - goals, issues, approaches
- Software Builds
  - goals, issues
- Project Scheduling Exercise
- Project Post Mortems

9/3/2007

Basic Project Skills

2

## change control

- who can make what changes, where, when?
- sometimes, some change is good
  - it represent progress as work is completed
  - such changes should be facilitated
- sometimes, some change is bad
  - changes can be disruptive to the product ●
  - we need processes to detect & prevent these
- hopefully these processes are adaptive
  - adjusting the burden in response to the risk

9/3/2007

Basic Project Skills

3

## change control mechanisms

- may be performed by version control tools
  - may control who can modify which files
  - may notify interested parties of changes
  - these features are usually configurable
- may be managed by human processes
  - publication and objection
  - designated component reviewers
  - change control boards
- should have mechanism/policy separation ●

9/3/2007

Basic Project Skills

4

## the laws of version control

- All of our work products are versioned ●
  - we can tell what version we are dealing with
- All official changes are tracked ●
  - we know exactly what changes were made
  - we know who made each change, when, why
- We can reconstruct any version at any time ●
  - not just the current version, any prior version
- Files exist in multiple parallel branches ●
  - each of which has its most current version

9/3/2007

Basic Project Skills

5

## version control procedures

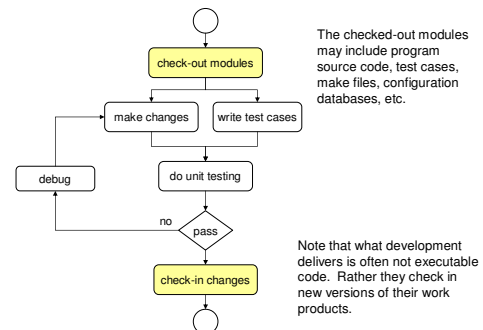
- never deliver a work product directly ●
  - rather, deliver a version-controlled file
  - ensures proper recording of all work
- build from the version controlled files
  - extract specific (or current default) versions
- associate versions with deliverables
  - release has a list of all versions used to build it
  - test/bug reports associated w/specific releases
  - bug fixes are associated w/new file versions
  - work product approvals specify a version ●

9/3/2007

Basic Project Skills

6

## software development process



9/3/2007

Basic Project Skills

7

## Conflicting Updates

- People may work at cross purposes
  - independent changes to same module
  - different understandings of how things work
- File Change Notifications
  - subscribe to notifications for selected files
- File Locking
  - at check out time, or independently
  - locks can be advisory or enforced
- Change merge assistance
  - automatic difference analysis, proposed merge

9/3/2007

Basic Project Skills

8

## the laws of software building

- The correctness of software should be a function of the underlying code, not the manner in which it was built.
- All builds (of the same software versions) should yield the same executable bits. No matter who builds them, when they are built, or where they are built.
- Test what you build. Ship what you test.

9/3/2007

Basic Project Skills

9

## Standard Build Environment

- there is no universal build environment
  - different tool-sets, releases, and platforms
- all of these can affect a built program
  - include files define types, constants, macros
  - libraries contain a huge amount of code
  - different compilers produce different code
- we need a standard build environment
  - compilers, headers, libraries, build tools, ...
  - it must be used for all product builds
  - it must be archived for future use

9/3/2007

Basic Project Skills

10

## Standard Build Procedures

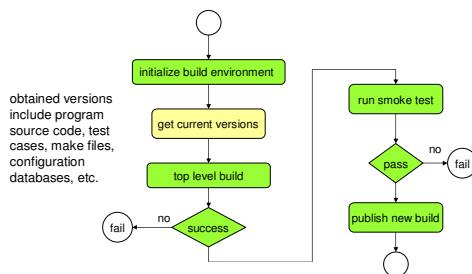
- software may be very complex to build
  - many components must be built
  - each must be built with correct options
  - pieces must be built in the correct order
  - components must be properly combined
- entire build process must be automated
  - from extraction to installing the binaries
  - to ensure it is done properly/same every time
- build scripts are part of the code

9/3/2007

Basic Project Skills

11

## software build process

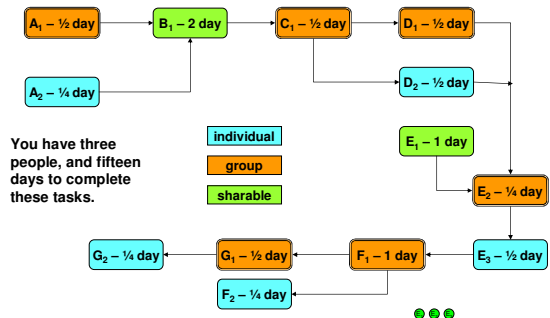


9/3/2007

Basic Project Skills

12

## Project Scheduling Exercise



9/3/2007

Basic Project Skills

13

## Post Mortems

- every project is a learning opportunity
  - improve our skills with existing methodology
  - try new techniques, confront new problems
- efficient/effective learning is not automatic
  - causes and solutions are not always obvious
  - analysis and understanding take time & work
- post-mortems are pro-active learning
  - set aside time for reflection & exploration
  - gather & examine perspectives in a group
  - produce written report & recommendations

9/3/2007

Basic Project Skills

14

## For next lecture

- McConnell 3.1. 20
  - Good overview quality and what affects it
- What is Quality
  - good discussion of an imprecise concept
- What is S/W Quality Assurance
  - excellent background and overview
- What is a post-mortem
  - good introduction to goals and process
- Anatomy of a Retrospective
  - example of a facilitated difficult post-mortem (skimming this briefly will be fine)

9/3/2007

Basic Project Skills

15

## Back up slides

## subversion - keywords

- Automatic module labeling
  - macros that are included in the file text
  - automatically expanded when file is extracted (… if they are enabled by the **svn:keywords** property)
- Standard keywords
  - \$Date: date of the last change \$
  - \$Revision: current version number \$
  - \$Author: person who made last change \$
  - \$URL: location of file and repository \$
  - \$Id: all of the above \$

9/3/2007

Basic Project Skills

17

## subversion - branches

- support parallel code evolution
  - concurrent releases
  - variant products
- implemented as copies of an original tree
  - `svn copy original new-copy`
  - perform branch development on new sub-tree
- copy changes from one branch to another
  - `svn merge -r first:last repository-files`
  - take those changes, make them to local copy

9/3/2007

Basic Project Skills

18

## subversion - locking

- proactively prevent conflicting changes
  - only owning user is allowed to change file
- locking commands
  - `svn lock file --message "text"`
  - `svn commit file`
- lock abandoning
  - `svn unlock file`
- lock breaking
  - `svn unlock file --force`

9/3/2007

Basic Project Skills

19

## make – implicit rules

- standard actions for default transformations  
%.O: %.c  
cc -c \$(COPTS) \$<
- value
  - greatly shorten makefile (dependencies only)
  - put action in one place for easier update
- automatic variables for implicit rules
  - \$@ the desired target
  - \$< the first prerequisite
  - \$? the newer-than-target prerequisites
  - \$\* the stem (without suffix)
  - %% the target member name (for archives)

9/3/2007

Basic Project Skills

20

## make - variables

- user variables
  - form: \$(name)
  - can be set explicitly (=, :=, ?=, +=)
  - can be set on command line, in environment
- typical uses
  - lists of file names target: \$(objs) \$(libs)
  - directory paths cc -I\$(include) ...
  - commands/options \$(CC) -\$(COPTS) ...

9/3/2007

Basic Project Skills

21

## make – control directives

- **include** rules from another file
  - processed as though part of the makefile
  - can be read conditionally, based on variables
  - often used for definitions or modularity
- **if, ifdef, ifeq, ifndef, ifneq, else, endif**
  - conditional processing of rules and directives
- nested **make** invocations
  - actions can include **make** commands
    - with another makefile or in another directory

9/3/2007

Basic Project Skills

22

## make – phony targets

- typically operations
  - clean, extract, install, etc.
- if the file does not exist ...
  - actions taken when ever target is requested
  - .PHONY directive in case file might exist
- or create a file to record action taken
  - dependency rules can see if it is still up to date
  - e.g. “backup”

9/3/2007

Basic Project Skills

23

## Anecdotal Process Improvement

- study a defective product
  - to understand why it came out defective
- figure out where our process failed
  - what steps allowed this defect to happen
- update our process
  - to prevent such mistakes in the future
- this approach works
  - but some major problems remain un-fixed
  - some fixes don't wind up making a difference

9/3/2007

Basic Project Skills

24