new assignment - implementation
design due 5pm

4156 Thursday 10/17/13 page 1

before writing any code

setup version control
(aka configuration management)

http://ase.cs.colombia.edu/stash
uses JIRA groups & git
(book uses subversion)

who has used git before this class?
subversion? another?
which is better?

- version control systems manage changes to ~~collections of~~ documents, software, web sites, etc.

check out
commit
revert (keeps backup)

logs all revisions with commit comments
can often show "diff"

useful for single developer,
more useful for team

share with colleagues
commit log shows who committed -
each change

some systems lock files and -
only allow one checkout
at a time

problems with this?

others allow multiple checkouts
and try to merge revisions

there can be conflicts that
need to be manually resolved

should update before starting
to edit each file to make
sure starting from latest

centralized vs. decentralized
(distributed)
(SVN) (git)

single master ~~copy~~ vs. clone
repository whole repository

tag each release (or important milestone) so can always find it later, makes snapshot of all files

may need to branch
- ~~[illegible]~~ fix bugs in version 1.0
- while continuing development on version 2.0 in trunk (master branch)

so branch for version 1.1

eventually merge branch with trunk in most cases

---

back to before writing any code

need to choose a build tool (book uses ant, also mentions maven)

some frameworks (~~[illegible]~~
come with a specific build tool
Java play uses SBT
Ruby on Rails has rake
Python Buildout?

4156 Thursday 10/17/13 page 4

→ who has NOT used a build tool?

some IDEs also come with a specific build tool (or have one hardwired)

build tools : build script

build automation

- compile source code into ~~binaries~~ executable form
- find dependencies (libraries)
- package ~~libraries~~ for deployment
- generate documentation
- "clean"

- ~~may~~ run ~~tes~~ regression test suite
- run application itself

→ continuous integration

rebuild whole system (or incremental)

~~on~~ nightly or even every commit

may run all tests again too

more later in semester

commits frequent, more than once per day notice I mentioned IDES

another thing to do before starting
to write code is choose an IDE

usually same IDE for a small team
but may not be practical for
a large or distributed team

Who has not used an IDE?
~~(emacs~~
not going to discuss IDEs further
unless real need - then maybe
a tutorial on some Friday

finally, last thing to do before
starting to code? - what is it?

determine coding conventions -
or pick style checker to enforce
standard coding conventions

most languages or frameworks
have standard conventions
companies also make own conventions

4156 Thursday 10/17/13 page 6

coding conventions generally address:

- commenting + documentation
- consistent indentation
- avoid obvious comments
- ~~coding~~ code grouping
  blank lines between tasks w/in same function
- consistent naming scheme

  2 popular options

  camel Case
  under_scores

- DRY principle - dont repeat yourself
  DIE - duplication is evil

  same pieces of code should not be repeated over & over

- avoid deep nesting
  for sake of readability

- limit line length
  (max 80 chars for terminal windows)

- file & folder organization
  often supplied by framework
  or IDE (or can mimic)

- consistent temporary names
  these can be short
  even single letter

  i, j for loop counters
  ret for return variables
  ~~[illegible] for~~.

- capitalize SQL special words
  to distinguish from
  table & column names

- separation of code & data

  - e.g., html is data in
    web applications