Development Tools

PSU CS 300 Lecture 2-2

Bart Massey
Assoc Prof Computer Science
Portland State University
<bart@cs.pdx.edu>
Modern SE Is Tool-Driven

• In OSS, and in general
• Many kinds
  – Compilers, debuggers…
  – SCMS, build, configuration, bug-tracking…
  – Email, Web…
PL Tools Have Evolved

- Compilers are better
  - Optimizing, correctness
- Languages are better
  - GC, static types, expressiveness
- Debugging is way better
- Auxiliary tools are there
SE Tools Are “Standard”

- “Make” has always been with us
- Modern SCMS rules
- IDEs help navigate projects
Debugging Tools

- For C++
  - Boehm GC, autopointers
  - Gdb
  - Valgrind
  - -Wall
  - Unit test frameworks
SCMS

• aka VCS, SCS...
• Record project history
• Manage sharing of work products
• Allow divergent development
• Allow regression
Centralized vs Distributed SCMS

- **Centralized (svn) serves when**
  - Folks are always well-connected
  - Branching is simple
  - An admin is available

- **Distributed (hg) relaxes these**
Project Communications

• cf Brooks' notebooks etc
• On small project
  – email list, phone numbers
  – Wiki is good (eg bartforge)
  – Issue tracking may be wiki or separate BTS
• Scale carefully!
Configuration Management

- These days, may be IDE-driven
- cf autotools, “manual”
- Building, packaging, deploying is hard
Watch For Tool Opportunities

- Patch management system?
- Contributor management tools?
- Metacoding?
- Wacky processes?