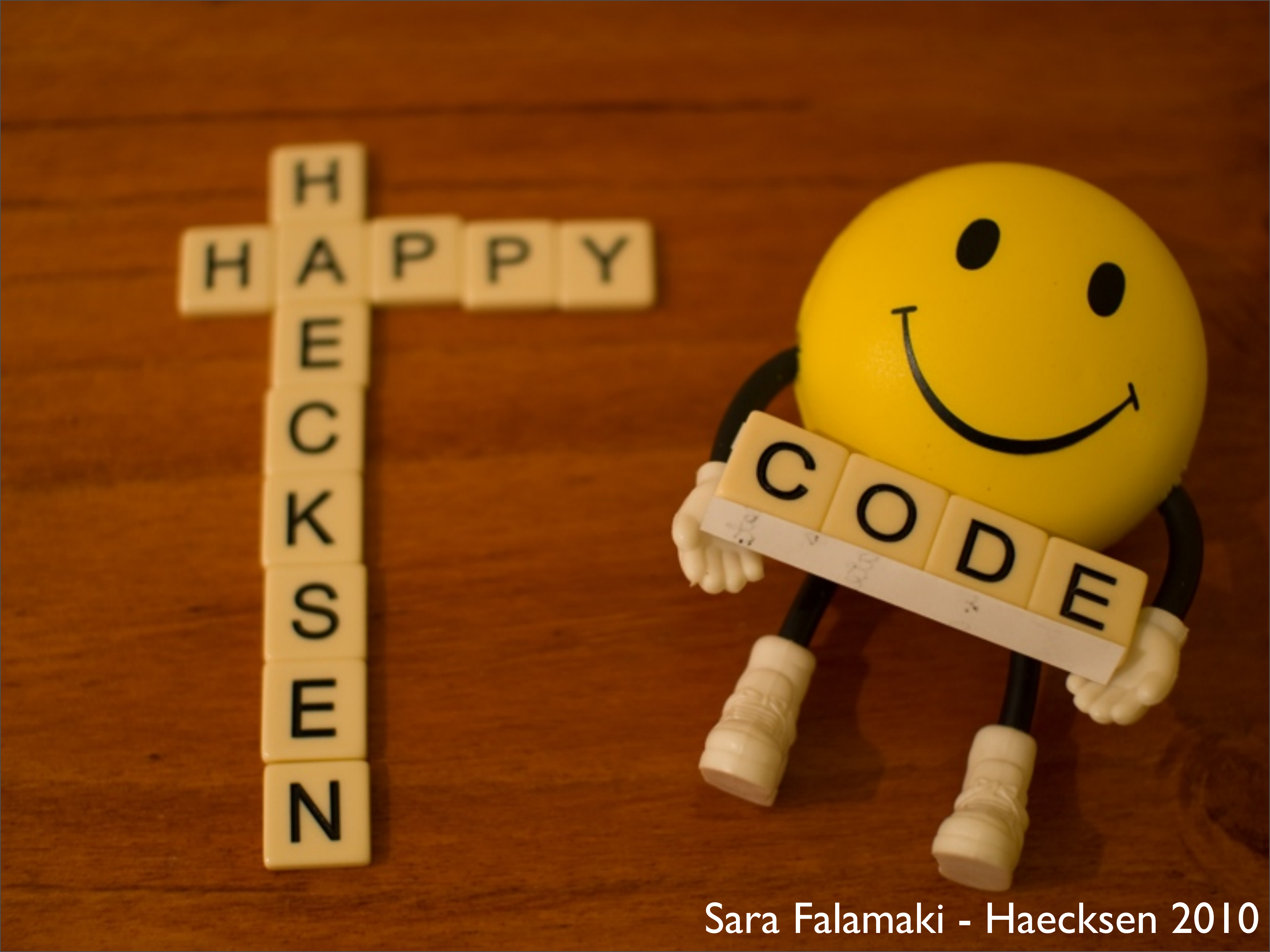


Sara Falamaki - Haecksen 2010

Happy Code?

Happy Code?

- Correct

Happy Code?

- Correct
- Understandable

Happy Code?

- Correct
- Understandable
- Efficient

Happy Code?

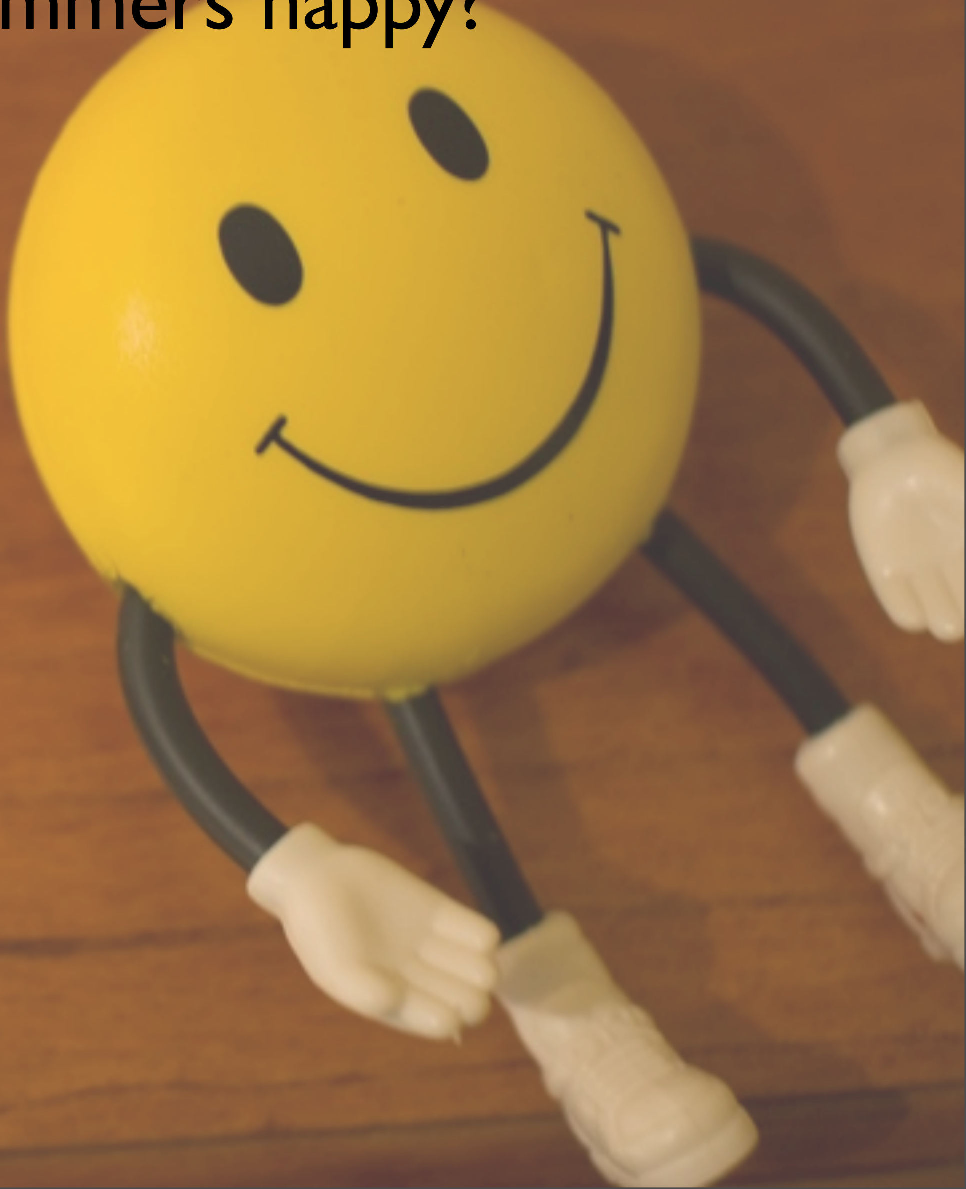
- Correct
- Understandable
- Efficient
- Testable

Happy Code?

- Correct
- Understandable
- Efficient
- Testable
- Supportable

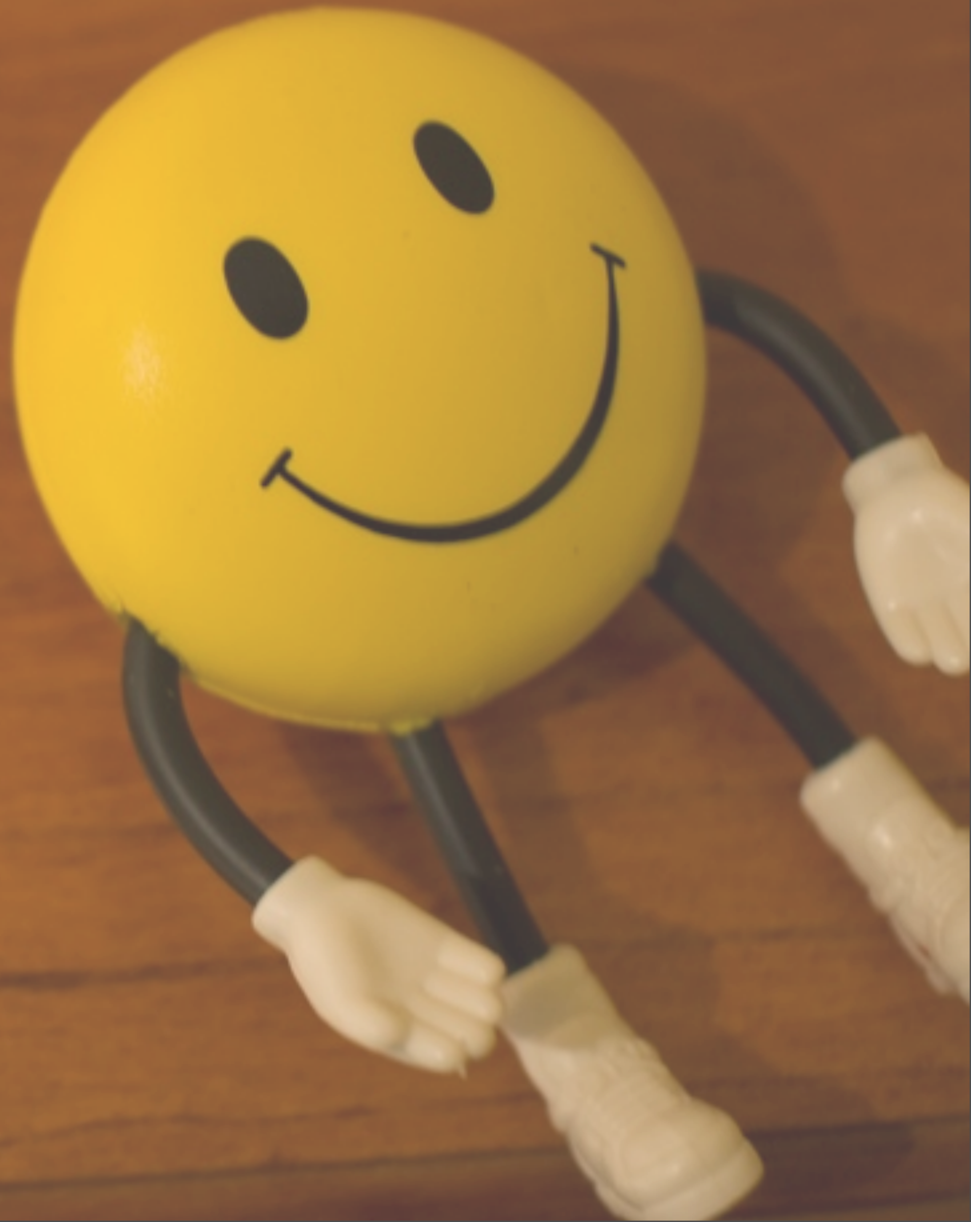
Happy Programmers

What makes programmers happy?



Happy Programmers?

- Good Tools
- Good Policies
- Good Colleagues!
- Happy Code!



The Right Tools

The Right Tools

- Version Control
- Knowledge Management System
- Bug/Ticket/Task tracking system
- Build System
- Testing System
- Development tools

Version Control

- Centralised (CVS, SVN)
- Distributed (Git, Darcs, Mercurial, Bazaar)
- Google for a comprehensive comparison of different systems

Version Control

- Dealing with Conflicts: *“My current team has a bizarre fear of concurrent edits. Apparently they used some sort of lame source control software in the past that let multiple people edit a file, and whoever checked in last won!” - Kathryn from #linuxchix*
- Managing trees (branches, tags)
- Porting bug-fixes to different versions

Knowledge Management System

- Any is better than none
- Wikis, Content Management Systems (CMS), Document management systems (Sharepoint), combined solutions (Redmine, fogBugz, Jira)
- Should be searchable and organisable

Knowledge Management System

What goes in one?

Knowledge Management System

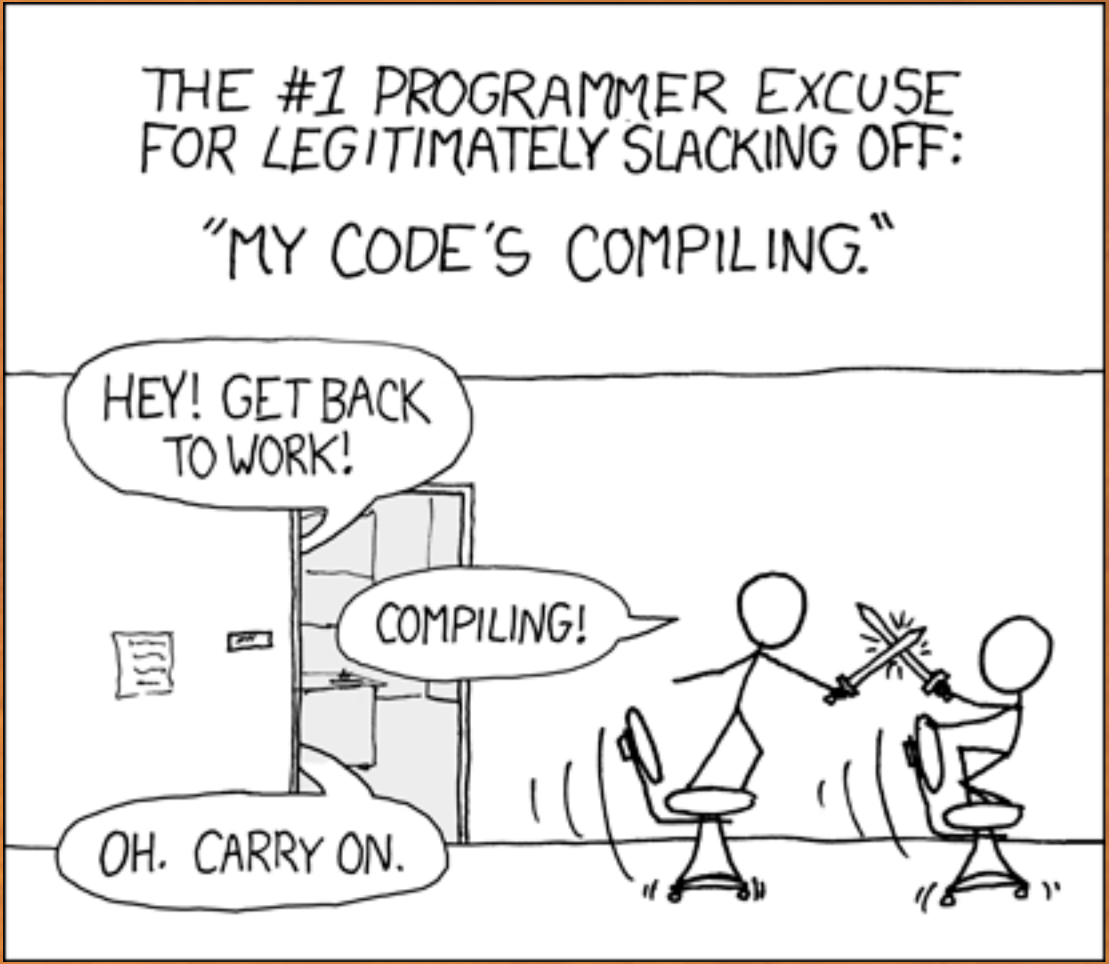
- Specification
- Important decisions
- Design documents
- Information about target platforms
- Essential information for employees

Task Tracking System

- Lets you manage Bugs, Tasks and Tickets
- Integration with Version Control System is useful
- Helps with project management
- Often bundled with VCS, wiki or both.

Build System

- Ideally a one-step build process
- Easy building for multiple different targets (if relevant)
- Integration with test system and version control system.


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

Testing System

- Ideally should be in place before you start coding
- “Continuous Integration”
- Nightly builds/Nightly tests
- Staging system

Testing System

Testing System

- “Does it work?” testing - run it, and see it doesn’t crash straight away
- Unit testing
- Regression testing
- Performance testing

Development Tools

- The right language for the job
- IDEs - love them or hate them
- Profilers, debuggers, monitoring tools
- Network debugging tools

Discussion!

- What tools make *you* happy? :-)

Slides + More Info

http://sara.falamaki.id.au/moin/Writing/HappyHackersHappyCode