That Real World Thing

Being a Software Developer

Sean Handley

October 16, 2008

Structure

Operation

Day to Day Pointers

Technicality

Good Coding Practices

Context

Why You Code

Brain

Those Gray Matters

Wrap

Write

- A day in the life of a developer will involve many
  - Conversations.
  - Decisions.
  - Changes.
  - Plans.
- Scribble it down as it happens.
- The more zealous, the better.
- Get names, places, dates, agreements, milestones and contact details.
- Do it now.

Speak

- Many of the best ideas arrive spontaneously. Share your thoughts.
- Don't be afraid to voice doubts. Even if they turn out to be unfounded, a discussion helps to give the specification shape and keep everyone on the same page.

Relay

- Think twice before every phonecall, e-mail or meeting and ensure that
  - Everyone who should be involved is involved and has all the right information.
  - The right people have been CC'd (or BCC'd if you're being crafty).
  - The right people are told the right things at the proper level of detail (your manager will be annoyed if you go around telling people how chaotic things actually are behind the scenes some days...).
  - Don't hoard information. Buffer for a while, perhaps, but remember to flush.

Assimilate

- Read code written by other devs on your team.
- Ask them lots of questions.
- If you feel it's warranted, offer up improvements.
- Do mention to other devs if you're planning to re-work code that is traditionally their territory.
- Have other devs look over your code, especially at the beginning of your time with the company.
- If the company doesn't have a set of common practices or internal documentation, use your newness as an opportunity to put such a project in place. Your fresh eyes on the codebase will prove invaluable.

Describe

- Talk to the other developers about what they're working on. You may be able to help and, if they go off sick, you know what issues remain open.
- Share solutions to common problems so new developers don't hit the same brick walls.
- Ensure you use the right words with the right people at the right time.

Raptors

- One *goto* can't hurt...

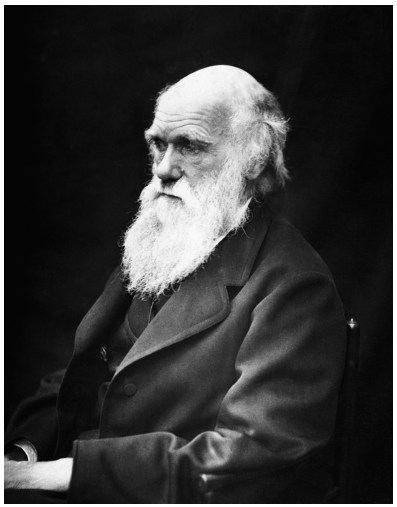
Conventions

- Every development team has a set of conventions.
- Learn the naming rules.
- Familiarise yourself with the general development lifecycle and release strategy.
- Know the release schedule and team milestones.

Comments

- Maintaining someone else's badly-commented/undocumented code is a nightmare.
- Don't let that be your code.
- If you need to read a section of code more than once, it needs a comment.
- Document well - if the technology does inbuilt document (like Javadoc or .NET XML doc) then document all public members.

Revolution



- You need to realise when you're painting into a corner.
- Rewriting parts of a project may well become inevitable in order to keep it maintainable.
- Learning from old mistakes is what makes a solution smarter. Do it. Viva la revoluçion.

Philosophise

- Take the principles of Unix with you. They're useful in any project, proprietary or not.
  - Design components to do one thing well.
  - Store as much detail in data structures as possible so logic can be robust and dumb.
  - Be modular.
  - Clear code is better than clever code.
  - There is never 'one true way'.

Apps

- Write lots of small, focused test and helper apps.
- It's always better to automate where you can.
- If you find yourself spending a large amount of time doing a particular task, look into automating some or all of it.
- Don't work hard. Work smart.

Dig

- Learn the finer points of the technology.
- Familiarise yourself with common library components of the language you're using. Proficiency with this will save you a lot of time.
- Make use of the features of your development environment. If you're using an IDE, go through the menus and learn what all the features do.
- If you're using a text editor, see that it has helpful features like syntax highlighting, code folding etc. Vim, Emacs, SciTE, ConTEXT and Notepad++ are all good.

Change

Model 123 with Odyssey® Styling

- Change is inevitable (unless a vending machine is involved).
- Find out what changes and encapsulate it.

Role

- Bear in mind your manager's perspective of your role in the company.
- Developer time is finite and serves a wider purpose.
- Milestones achieved can clinch business deals and seal the company's fate.
- Developers like details. Managers like a broader perspective.
- Get to know your manager and speak to them on a regular basis.

Lingo

- What you say vs what you mean.
- You say "it works" but...

Reality

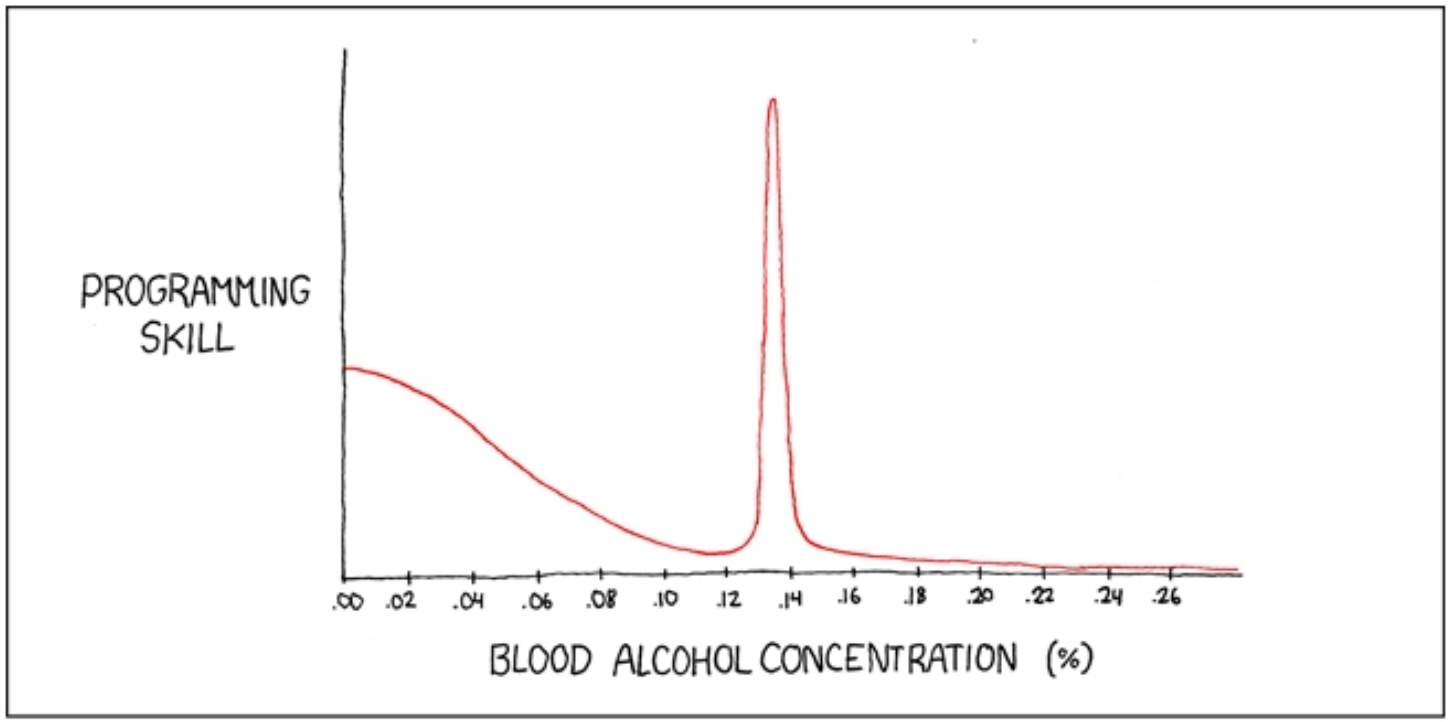
Persist

- Be persistent and proactive.
- Take ownership of projects - responsibility is a great motivator.
- Chase down human bottlenecks. Some people don't like returning e-mails or calls. Be persistent (but stay polite), get them to commit to targets, keep chasing them until they deliver.

Cognition

- Driving when tired leads to crashes. Same with coding.
- If you're not on your toes, stick to lighter duties.
- Don't rewrite the core libraries at 5PM because you'll just have to rewrite them the next day.
- Drink lots of water. Your brain will thank you, even if your bladder doesn't.
- Coffee doesn't actually make you a better coder. Although alcohol might (disregard Windows ME here...).

Ballmer Peak



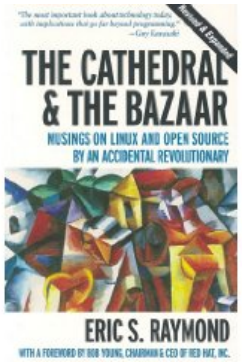
Concentration

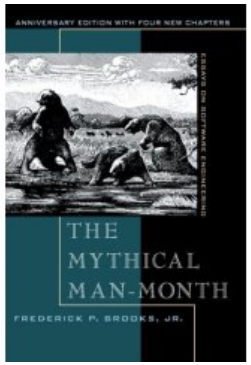
- Coding for hours on end can put you into *Code Mode*.
- Code Mode is where your brain gets so tied up with the details of your program that it struggles to do anything else.
- I've had the phone ring while I was coding. Answered it, had a conversation, put the phone down and completely forgot who it was or what they wanted straight away.
- Lesson learned? Write things down, think twice, be aware of the context switch.

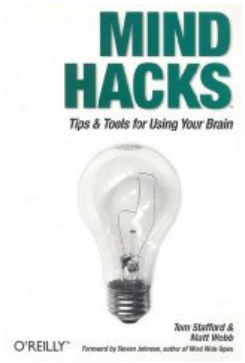
In Short

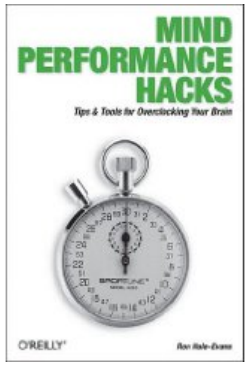
- Operation
  - Stay on top of info and be a careful, descriptive communicator.
- Technicality
  - Endeavour to find the right solution for the right problem and see that it is implemented with the best technical finesse you can muster.
- Context
  - Understand how you slot into the big picture.
- Brain
  - Listen to your brain and keep it happy.

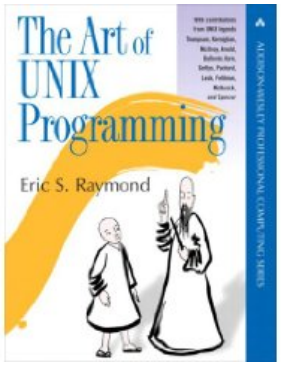
Read


THE CATHEDRAL & THE BAZAAR
MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY
ERIC S. RAYMOND


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.


MIND HACKS
Tips & Tools for Using Your Brain
O'REILLY


MIND PERFORMANCE HACKS
O'REILLY


The Art of UNIX Programming
Eric S. Raymond

Questions?