Sustainable software development

From hacking...

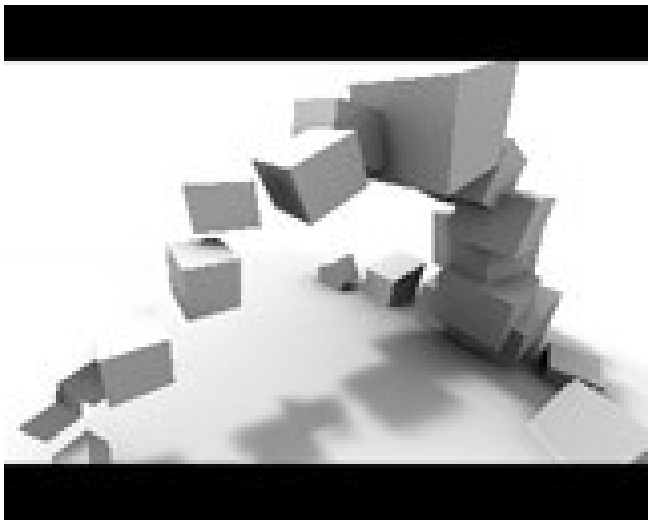
...to craftsmanship

About me

Mikael Kalms, Technical Director at DICE

Programming since 1990

I code for fun



I code for cash

Why are you here today?

What's this 'sustainable' about?

Never get permanently stuck due to your habits or your craftsmanship.

You should be able to accomplish anything, as long as you have enough time available.

Goals

Spend time on the hard stuff, not the stupid stuff.

Work effectively...

… on small and large programs.
… on new and old programs.
… alone and with other programmers.

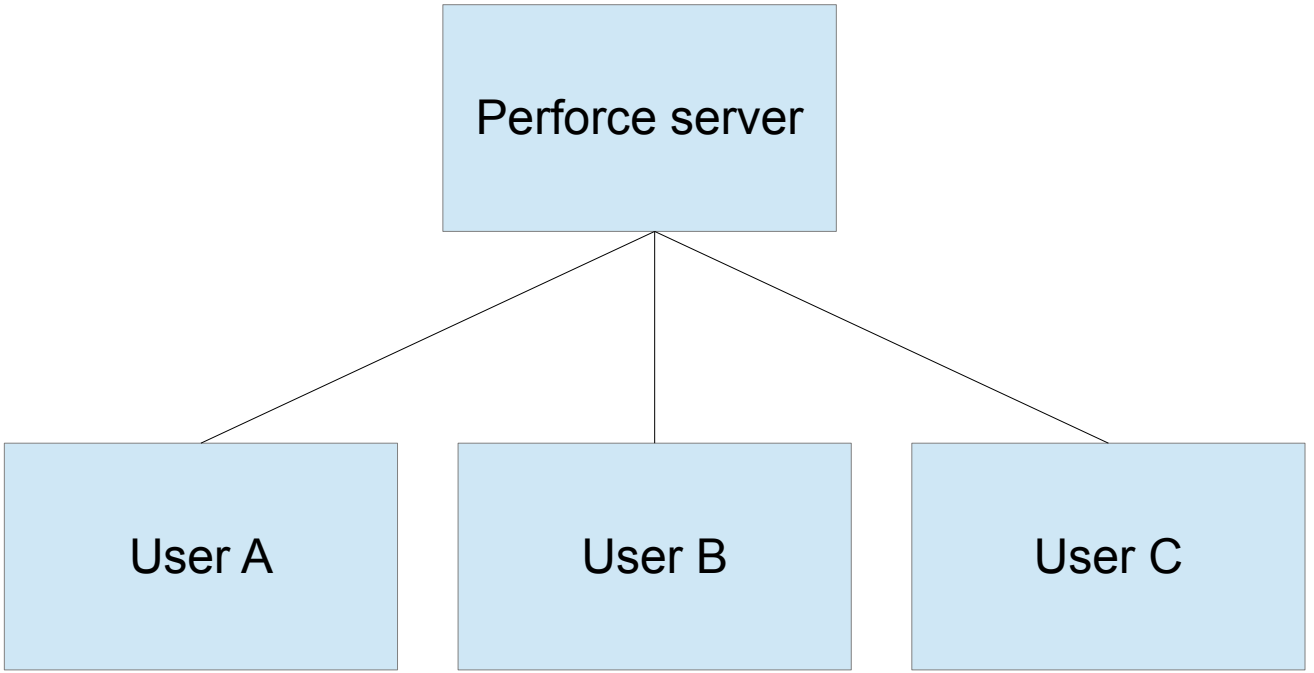
Version Control System

”Do I have the latest version of file X...
...or do you?”

”I changed some stuff, my program doesn't work,
and now I can't remember what I changed!”

^^^ This is stupid stuff. Use a VCS to avoid it. ^^^

Example VCS: Perforce



Perforce, in one sentence

Perforce is ”a file system with history”

Tour de Perforce

I will show you what it's like in practice.

You will now criticize my code.

Grab handouts.

It's time to dissect my Matrix-Vector multiply code.

You will now dive into larger codebases.

It's time to take a look at a larger program.

I will now criticize your code.

:)

That's all folks.

Questions?

You can reach me at mikael@kalms.org.

Strategies, 1/5

Naming is important

- what does this function/variable do?
- what does this function/variable NOT do?
- pure, or with side effects?

Strategies, 2/5

Code should be easy to use correctly, and hard to use incorrectly

- Structure/flow
- Type system
- Scope of variables/functions
- Dual functions (create/destroy)

Strategies, 3/5

Know what happens where

- Preprocessor
- Compiler
- Linker
- Runtime

Strategies, 4/5

Know your tools

- Version control system
- Editor
- Programming language
- Debugger

Strategies, 5/5

Know your pace

- Write testable chunks of code
- Test early, test often
- Re-run your tests!