using quantification against itself

get ready: 3 slides per minute + a demo





.....





quantify development of a system that quantifies unplugging

quantify development

seamlessly log code creation in near-real-time

unplug user0\$ git shadow activate OPEN QUESTIONS

- Do programmers of different skill levels actually write code differently?
- Can micro-level code churn, ownership, and other established metrics aid in fault prediction?
- Can granular coding metadata (like the exact time of day a line was written) be combined with growing QS datasets to provide novel insights?



git-shadow

~/unplug README.md LICENSE.md analysis.py





Jonathan Foote

git-shadow



git-shadow



ifoote

demo

(time permitting)



quantify unplugging

quantify log analyze gamify





Nexus5



example

(time permitting)

"How long did it take you to code that?"

Total time coding (days, H:M:S): 1 day, 4:48:51 Total commits: 2757

\$./shadow_analysis.py



Android app (H:M:S): 5:19:47 Total commits: 376

\$./shadow_analysis.py — android



OS X 'dump' function (days, H:M:S): 0:27:29 Total commits: 73

\$./shadow_analysis.py -L :'def dump':unplug





Thanks for your time

http://foote.pub