Globalization and the Future Developer

Audris Mockus

audris@avaya.com

Avaya Labs Research
Basking Ridge, NJ 07920
http://mockus.org/

yes, globalization is inefficient, but it happened, so get over it

Transfer of ownership

Globalization

Transfer work to where software developers are

deal with distributed groups?

→ collocate

Understand training/mentoring

Understanding developer productivity

References:[5, 3]

Software Engineering and Call Centers

Call center reporting systems are crucial

- Customer quality: e.g, dropped calls, time on wait
- Performance of agents: e.g.,
  - Number of customers served
  - Number of “successful” interactions
  - Time to complete the call

Call Center Agent

Call center measures are far from perfect

but business case is strong

hire **on demand** with **hourly** precision

individuals are completely **exchangeable**

Software Engineer as a Call Center Agent

Comparable software productivity measures are available

- Modifications to the code
- Issues resolved
- Time to resolve
- Centrality of the tasks
- Inverse of the amount of code added

Developer learning

Types of developer participation

Newcomer → core team

Newcomer challenges

Resources/tools

References: [6, 2, 4, 1, 7]

References

[1] Andrew Begel and Beth Simon. Novice software developers, all over again. In *International Computing Education Research Workshop*, Sydney, Australia., 2008.

[2] B. Dagenais, H. Ossher, R. K. E. Bellamy, M. P. Robillard, and J. P. de Vrie. A qualitative study on project landscape. In *CHASE'09*, Vancouver, Canada, May 17, 2009 2009.

[3] J. D. Herbsleb and A. Mockus. An empirical study of speed and communication in globally-distributed software development. *IEEE Transactions on Software Engineering*, 29(6):481–494, June 2003.

[4] AJ Ko, R DeLine, and G Venolia. Information needs in collocated software development teams. In *29thInternational Conference on Software Engineering*, pages 344–353. ACM Press, May 20–26 2007.

[5] Audris Mockus and David M. Weiss. Globalization by chunking: a quantitative approach. *IEEE Software*, 18(2):30–37, March 2001.

[6] Yunwen Ye and Kouichi Kishida. Toward an understanding of the motivation open source software developers. In *ICSE 2003*, pages 419–429, Portland, Oregon, 2003.

[7] Minghui Zhou, Audris Mockus, and David Weiss. Learning in offshored and legacy software projects: How product structure shapes organization. In *ICSE Workshop on Socio-Technical Congruence*, Vancouver, Canada, May 19 2009.

Audris Mockus
Avaya Labs Research
233 Mt. Airy Road
Basking Ridge, NJ 07920
ph: +1 908 696 5608, fax:+1 908 696 5402
http://mockus.org, mailto:audris@mockus.org

Audris Mockus is interested in quantifying, modeling, and improving software development. He designs data mining methods to summarize and augment software change data, interactive visualization techniques to inspect, present, and control the development process, and statistical models and optimization techniques to understand the relationships among people, organizations, and characteristics of a software product. Audris Mockus received B.S. and M.S. in Applied Mathematics from Moscow Institute of Physics and Technology in 1988. In 1991 he received M.S. and in 1994 he received Ph.D. in Statistics from Carnegie Mellon University. He works in the Software Technology Research Department of Avaya Labs. Previously he worked in the Software Production Research Department of Bell Labs.