

# Global Software Development: The Bell Labs Collaboratory

David Atkins<sup>1</sup>, Mark Handel<sup>2</sup>, James Herbsleb<sup>1</sup>, Audris Mockus<sup>1</sup>, Dewayne Perry<sup>3\*</sup>, Graham Wills<sup>1</sup>

<sup>1</sup>Bell Laboratories  
Lucent Technologies  
263 Shuman Boulevard  
Naperville, IL 60566 USA  
+1 630 713 1869, +1 630 713 4070  
{datkins, jherbsleb, audris, gwills}@lucent.com

<sup>2</sup>School of Information  
University of Michigan  
3218 Computing Center Building  
Ann Arbor, MI USA 48109  
+1 734 764 6131  
handel@umich.edu

<sup>3</sup>Electrical & Computer Engineering  
The University of Texas at Austin  
Austin TX 78712  
+1 512 471 2050  
perry@ece.utexas.edu

Software development is a global enterprise for many large corporations. Searching for talent across national boundaries, and integrating groups thrown together by mergers and acquisitions are but two of the many forces conspiring to change the organizational context of software development (e.g., [1]).

For the past three years, a group of researchers from Bell Labs and the University of Michigan have been working to understand and address global development issues. The project has four concurrent threads of activity:

**Empirical studies of global development.** The problems of global development are varied. We conducted over 200 structured interviews at 14 sites on three continents, with people at all levels in the organization from developers to executives. In addition to the obvious problems of time zone, limited bandwidth connectivity, language and culture differences, we found the chief victim of global development to be speed [4]. Changes which cross sites take much longer than changes which are all at a single site. The difference appears not to be due to the size or complexity of cross-site changes, but rather to communication and coordination issues.

Most pressing among these communication and coordination issues [3] are 1) what we are calling issue resolution paralysis, induced by the inability to identify the right person, initiate communication, and have an effective interchange, and 2) a complete lack of informal “corridor talk” among people at different sites, which results in a surprisingly powerful impediment to the flow of information.

**Collaboration tools for awareness and communication.** In order to address these problems, we have developed several tools for collaborating over distance in software development. The tools include

- Experience Browser, which allows the user to explore data in the change management system through a visual interface to find people experienced in various parts of the code.
- Rear View Mirror, an instant messaging tool that supports persistent team chat rooms, a lightweight tool for informal conversation.
- CalendarBot, a web-based multi-user calendar tool that helps keep everyone informed of items such as travel and vacation plans.

All of these tools are currently in use within Lucent.

**Organizational models for distributing work over sites.** There are many possible ways to distribute development work over sites (see [2, 5]). For example, one might

- develop different subsystems at different sites,
  - execute different process steps at different sites,
  - develop a core product at a single site, and customize for different markets and customers at satellite sites,
  - locate different maintenance releases at different sites.
- These techniques require different mechanisms for coordinating the work, and differ in the circumstances in which they are most likely to be effective.

**Best practices for global development.** We have observed a number of practices that have been quite effective in improving communication. Among them are practices about

- establishing trust across sites by being even more responsive to remote colleagues than to local ones
- setting up liaisons at each site to facilitate cross-site communication
- establishing etiquette for answering e-mail and voice mail messages in a timely way
- planning for travel early in the relationship between sites – everything will work better after people have met.

Our future work will focus on continuing to introduce tools and models to practice, and to measure the results with analyses of MR data and survey results.

## REFERENCES

1. Carmel, E., Global Software Teams. 1999, Upper Saddle River, NJ: Prentice-Hall.
2. Grinter, R. E., J. D. Herbsleb, and D. E. Perry. The Geography of Coordination: Dealing with Distance in R&D Work. in *GROUP '99*. 1999, Phoenix, AZ
3. Herbsleb, J. D. and R. E. Grinter. Splitting the Organization and Integrating the Code: Conway's Law Revisited. in *21st ICSE 1999* Los Angeles, CA: ACM Press 85-95.
4. Herbsleb, J. D., et al. An Empirical Study of Global Software Development: Distance and Speed. in *ICSE*. 2001. Toronto, Canada: IEEE Press
5. Mockus, A. and D. M. Weiss. Globalization by Chunking: A Quantitative Approach. *IEEE Software* January - March, , 2001.

---

\* Work done while at Bell Labs.