Despite unprecedented challenges due to global boundaries and barriers, effective software teams succeed by implementing ambidextrous coping strategies to address emergent, complex problems in global software development.

Software development increasingly requires globally-distributed teams as organizations seek to deliver high quality software to global users and customers at lower development costs. This globalization increases the complexity of coordination in the software development effort, which in turn negatively influences project outcomes. Geographic distance, time separation, cultural differences, language differences, organizational boundaries, and functional boundaries, inherent in global contexts, represent significant global boundaries and barriers. A key challenge for information systems organizations today is to overcome these global boundaries and barriers and deliver quality software on time and within budget. However, coping with this challenge is a daunting task because organizations have a limited understanding of what makes global software development projects successful.

To shed light on how organizations cope with global boundaries and barriers to succeed in software development, we studied 22 globally-distributed software development projects (See the methodology sidebar for the details about our research methods). As shown in Figure 1, we conceptualize global software development as an I-P-O (input-process-output) model. In this model: global situational factors represent various global boundaries and barriers; task and team processes refer to coordination and communication processes normally required by software development; and project outcomes include software quality and on-time/within-budget completion. Global situational factors hinder traditional task and team processes, potentially causing poor project outcomes. However, effective global software teams adopt “coping strategies” to adjust their task and team processes to mitigate these negative effects on project outcomes. We found that effective global software teams tailor their software development and project management processes to cope with the difficulties of global contexts. More important, we found that effective coping strategies exhibit ambidextrous properties – i.e., strengthening
disciplined, rigorous processes while simultaneously building flexibility to quickly adapt and change these processes when needed.

Figure 1: Model of Globally-Distributed Software Development

Note: This paper is currently being reviewed by *Communications of the ACM*. If you want to get the full paper, please contact Professor Gwanhoo Lee (*glee@american.edu*).