Seeing inside: Using social network analysis to understand patterns of collaboration and coordination in global software teams

Kate Ehrlich, Giuseppe Valetto, Mary Helander
IBM Research
{katee@us.ibm.com; gvaletto@us.ibm.com; helandm@us.ibm.com}

Abstract

One of the pervasive challenges facing any software development team is getting the right level and timing of communication to ensure that people are able to coordinate their work effectively. Communication issues are difficult to address because important aspects of communication are largely invisible to both management and the individuals on the team. Communication challenges are further exaggerated in global software teams, because of the different time-zones, cultures, and working environments. Social Network Analysis (SNA) is an established method for revealing patterns of human communication and decision-making. This tutorial introduces students to basic concepts in SNA, illustrates how SNA can be used to understand the dynamics of and address common communication problems in global software teams, and provides structured exercises in data capture, analysis and interpretation.

1. Introduction

As global software teams grow in number and importance, the social and behavioral challenges of managing and coordinating their work assumes a greater significance in the overall success rate of these projects [1]. But it can be difficult to see the challenges in a way that lends itself to change. Social Network Analysis (SNA), which has been successfully applied to collaboration issues in teams of varying sizes [2], can make an important contribution to understanding the dynamics of global software teams. One of the benefits of this approach is that it makes collaboration and communication more tangible and visible to everyone on the team. The ensuring clarity can motivate the changes in the team that are necessary to re-align the team structure with the work structure.

1.1 Software development as a socio-technical system

It has long been recognized that the structure and dynamics within the development organization are at least as influential as technical considerations for the success of software projects, and that those social and technical concerns interact in very interesting ways. This interplay between tasks, organization and people can be described as a socio-technical system. One way of analyzing and assessing software teams is to look at their communication structure, mapped against the inherent relationships between the software artifacts [3].

Concepts of alignment and congruence date back to 1968 when Melvin Conway [4] posited that any piece of software reflects the organizational structure that produced it. Even though the concept has been around for a long time, factors of alignment or congruence are rarely considered when putting together and managing the software team. This may be because factors, such as expertise, availability or location overly constrain choices about team membership and management. But, the absence of reliable information about the relationship between people, and, the relationship between people and software, further hampers the ability to structure and manage projects around alignment. Measures of socio-technical congruence are derivable from data kept in software repositories commonly supporting the development process.

This tutorial will introduce the basic concepts of SNA and provide concrete examples of how this approach has been successfully applied. The concepts will provide a lead into exercises in applying SNA to the communication challenges of global software teams. We introduce an approach in which SNA is closely related to the analysis of software artifacts and activities, as a way to illuminate how social network data can be mined from data repositories and how the analysis can provide insight into the organization of work in global software development projects.
2. Tutorial Organization

**Introduction to social network analysis.** This section of the tutorial will introduce basic concepts in social network analysis and how networks operate in business collaboration. We will examine productive and less productive patterns of collaboration and explore some of the actions that can be adopted by individuals, teams and organizations to improve collaboration.

Exercises to reinforce the learning portion will include: reading and interpreting social network diagrams, reviewing case studies to select remedial actions.

**Constructing social networks and other metrics from software engineering repositories.** In this section we will focus on tools and methods for deriving social network data from software repositories. We will review some of the common challenges of communication and collaboration for global software teams and examine how these data could be used to address these problems.

Students will undertake exercises in data capture and analysis.

3. Presenters

**Dr. Kate Ehrlich** is a researcher in the Collaborative User Experience group at IBM Research where she uses Social Network Analysis as a research and consulting tool to gain insights into patterns of collaboration in distributed teams. She has used SNA with over 60 groups covering team dynamics, knowledge and information flow, collaboration, communities, innovation, software development and governance structure.

**Dr. Giuseppe (Peppo) Valetto** is a post-doc in the Software Quality and Testing group at IBM Research, where he is investigating the construction and analysis of socio-technical networks that can be derived from information kept in repositories commonly employed in software development projects.

**Dr. Mary Helander** is a Math Scientist at IBM Research. She has 20+ years of industry and academic experience in operations research, software engineering and network optimization. Over the past two years, she has focused her work on network algorithms and analytics toward SNA. Prior to joining IBM, Mary was an Associate Professor and Director of the Applied Software Engineering Lab at Linköping University in Sweden. She also taught extensively and did research at Northeastern University and the University at Buffalo.

4. Level

This is an introductory level course that is suitable for anyone managing, participating in, or studying global software development teams.

5. Attendee Background

Some knowledge of statistics and familiarity with software engineering repositories and tools (e.g. CVS, Bugzilla, etc.) is helpful but not required.

6. References


