

2008

# Detecting Coordination Problems in Collaborative Software Development Environments

Chintan Amrit

*University of Twente, Netherlands, c.amrit@utwente.nl*

Jos Van Hillegersberg

*University of Twente, Netherlands, j.vanhillegersberg@utwente.nl*

Follow this and additional works at: <http://aisel.aisnet.org/eis2008>

---

## Recommended Citation

Amrit, Chintan and Hillegersberg, Jos Van, "Detecting Coordination Problems in Collaborative Software Development Environments" (2008). *EIS 2008 Proceedings*. 7.

<http://aisel.aisnet.org/eis2008/7>

This material is brought to you by the Benelux (BENAIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in EIS 2008 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Detecting Coordination Problems in Collaborative Software Development Environments

Chintan Amrit and Jos Van Hillegersberg

Department of IS&CM  
University of Twente, Netherlands  
{c.amrit, j.vanhillegersberg}@utwente.nl

## Extended Abstract.

The classic paper by Conway (1968) states organizations which design systems are constrained to produce designs which are copies of the communication structure of these organizations [1] (this has come to be known as Conway's Law). In other words, if the teams involved in software production have shortcomings in their interpersonal relationships, the resulting technical architecture of the software is likely to be flawed. Since Conway, many researchers have invented various more detailed Process Patterns which describe the preferred relationships between team communication structure (the social network) and technical software architecture. Currently a wide collection of such Process Patterns exist suggesting optimal team compositions and task assignments depending on the modularity of the technical architecture. However, these Process Patterns usually have limited empirical validation and are very hard to implement and monitor in practice. The increasing use of agile methods and globally distributed development have made the implementation and monitoring of Process Patterns even harder. As a result, a software project manager does not know which Process Patterns are suitable to his project and currently has no methods and tools to successfully use Process Patterns. The TESNA project (Technical Social Network Analysis) we report on in this paper addresses these issues. We propose a method and a tool that a project manager can use in order to detect Social Technical Structure Clashes (STSC).

STSC occurs if and when there exists a Socio-Technical Pattern (patterns involving problems related to both the social and technical aspects of the software process) that indicates that the social network of the software development team does not match the technical dependencies within the software architecture under development. These STSCs are thus indicative of coordination problems in a software development organization. We claim that continuous and early detection of STSCs can help project managers in monitoring the software development process and enable them to take action whenever a STSC occurs. We test the method and tool in a case study of a small and innovative software product company. The approach is illustrated below in Figure 1. The results of the study show that the method and tool support the project manager to implement Socio-Technical Patterns and use them to redesign the social network and/or technical architecture during development. Surprisingly, even in the relatively small development team that collaborates closely, STSCs occurred that the project manager was astonished by.

**Keywords:** Coordination, Collaborative Software Development, Software Patterns, Project Management

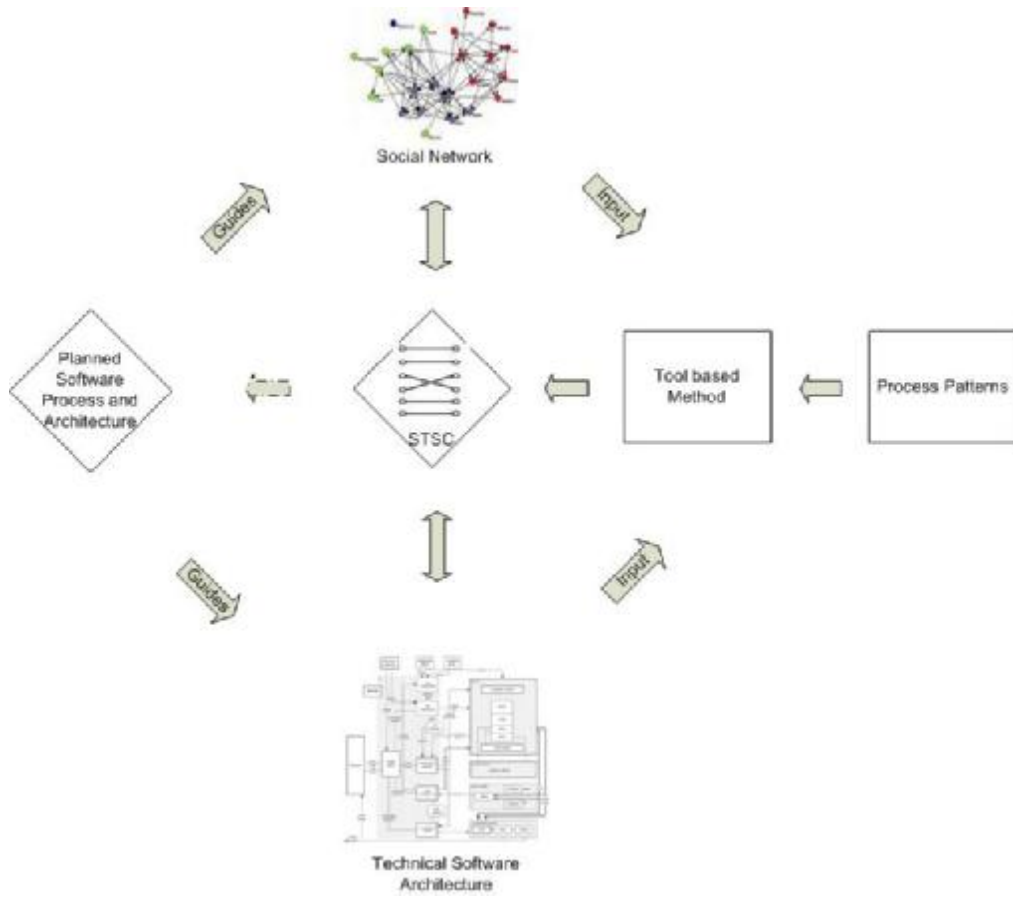


Figure 1: Socio-Technical Structure Clashes and the planned Software Process

1. Conway, M., *How do Committees Invent*, in *Datamation*. 1968. p. 28-31.