Chapter 82

Media Richness, Knowledge Sharing, and Computer Programming by Virtual Software Teams

Idongesit Williams

https://orcid.org/0000-0002-5398-3579

Aalborg University, Denmark

Albert Gyamfi

https://orcid.org/0000-0002-3839-4178

Aalborg University, Denmark

ABSTRACT

Software programming is a task with high analyzability. However, knowledge sharing is an intricate part of the software programming process. Today, new media platforms have been adopted to enable knowledge sharing between virtual teams. Taking into consideration the high task analyzability and the task characteristics involved in software development, the question is if the media richness of the current media platform is effective in enabling knowledge sharing among these virtual teams? An exploratory research was conducted on a software company in Denmark. The data was gathered was analyzed qualitatively using narrative analysis. This paper concludes, based on the case being investigated, that rich media does not fit the task characteristics of a software programmer. It further concludes that Media richness does affect knowledge sharing in these virtual teams. This is because the current lean media actually enables knowledge sharing as it fits the core characteristics of the software programming process.

INTRODUCTION

This chapter explores how media richness influences knowledge sharing activities in virtual teams involved in software programming. The task of interest is the process of programming or coding of the software. In recent times, large corporations have been involved in the decentralization of the operational aspects of their companies. Such corporations include SAP, IBM and General etc. (Lepsinger, 2015). The facilitation of their virtual operations has been enabled by the opportunities enabled by globalization and the evolution of the Internet. Based on our previous knowledge of the first author in software development and a brief practice in the trade, it was intriguing to learn that software development companies have been embracing virtual teams (Ramesh & Dennis, 2002). These companies have different software developers in different locations collaborating online to implement or programme software. This, of course, was not the intriguing part. That was because software development has always been developed with computers. So having such collaborative development effort with different computers connected to each other over the Internet was not that exciting. What was exciting though was how media richness enabled knowledge sharing within a virtual software development team as they implement or code the software. In a software development team domicile in a particular location, knowledge sharing is not difficult. Knowledge could be shared via face-to-face interaction using a device with rich media; placing comments in a programming code to explain a code in the software, documentation of daily team meetings, one on one tutorials in a company training session and codes and other relevant materials stored in repositories (see examples (Ramesh & Dennis, 2002; Ying, Wright, & Abrams, 205; DeRosa, 2014)). In this way, various members of the team are able to learn something new from one another. The new knowledge will help them implement their codes. If this scenario is transposed online, the need for this kind of knowledge sharing activities will still exist.

But knowledge sharing, as a result of the complexity in communication programming languages outside the software designed for it, might be challenging. This will definitely be the case in a scenario where two remote software developers working on the same project for the same company run into challenges on how to implement the codes. Depending on the level of information richness they need from each other in order to collaborate, different media platforms can be used. However, the more they try to unravel how to solve a programming code, rich media platforms may not be helpful. But lean media platforms might be helpful because they could read each other's codes, ponder on it and possibly verbalize their thoughts. A media platform used for such an activity should be able to enhance knowledge sharing in less ambiguous and equivocal manner. But in a situation where the software developers are discussing the requirement specifications they were given or they are presenting their thoughts in a scrum meeting. The nature of information being transmitted may not necessary be complex. The other team members will not only be interested in hearing what their colleague says, they will want to take not of non-verbal gestures. Here they can sense if they are on the same page or there are minor differences in opinion. In such a situation, rich media will be helpful. Therefore the media needs for the transmission of different forms of information by the team will differ. The fact that knowledge sharing in software teams, and by extension virtual software teams, require different set of media is supported in literature. The literature states that no media tool supports every facet of the operational dynamics in the development of a software (see (Bindrees, Pooley, Ibrahim, & Taylor, 2014)). For example, virtual teams use collaborative platforms such as GitHub to share programming codes and other media platforms such as WebEx, adobe connect or GoToMeeting, to support their daily tasks (DeRosa, 2014). Therefore one cannot dispute the use of multiple media platforms by these virtual software teams. But this raises a

Related Content

A Hybrid Approach for Feature Selection Based on Genetic Algorithm and Recursive Feature Elimination
Pooja Rani, Rajneesh Kumar, Anurag Jainand Sunil Kumar Chawla (2021). *International Journal of Information System Modeling and Design (pp. 17-38).*
www.irma-international.org/article/a-hybrid-approach-for-feature-selection-based-on-genetic-algorithm-and-recursive-feature-elimination/276416

Resource Scheduling Techniques in Utility Computing: A Survey
Inderveer Chanaand Tarandeep Kaur (2014). *International Journal of Systems and Service-Oriented Engineering (pp. 44-65).*
www.irma-international.org/article/resource-scheduling-techniques-in-utility-computing/114606

Secure Embedded Systems: Concepts and Issues
Ali Ahmadiniaand Ahmed Saeed (2018). *Cyber-Physical Systems for Next-Generation Networks (pp. 207-221).*
www.irma-international.org/chapter/secure-embedded-systems/204674

E-CARe: A Process for Engineering Ubiquitous Information Systems
Ansem Ben Cheikh, Agnès Front, Jean-Pierre Giraudinand Stéphane Coulondre (2013). *International Journal of Information System Modeling and Design (pp. 1-31).*
www.irma-international.org/article/e-care/80194

Location-Awareness with Action Systems
Luigia Petre, Kaisa Sereand Marina Waldén (2012). *Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications (pp. 463-483).*
www.irma-international.org/chapter/location-awareness-action-systems/66483