Chapter 2.17
Managing Intellectual Capital and Intellectual Property within Software Development Communities of Practice

Andy Williamson
Wairua Consulting Limited, New Zealand

David M. Kennedy
Hong Kong Institute of Education, Hong Kong

Ruth DeSouza
Wairua Consulting Limited, New Zealand

Carmel McNaught
Chinese University of Hong Kong, Hong Kong

INTRODUCTION

In this article, we will develop a framework for educational software development teams that recognizes the conflicts and tensions that exist between the different professional groups and will assist software teams to recognize the intellectual capital created by individuals and teams. We will do so by recognizing the inherent relationship between the tangible elements of intellectual property and the intangible organizational assets that form the basis of intellectual capital and by discussing how knowledge generated by a project team can become an explicit asset.

BACKGROUND

Universities are increasingly becoming developers of complex software-based applications. In-house development ranges from teaching aids and online learning resources to large information systems products that could ultimately become successful commercial ventures. Increased product complexity is easily recognized, yet research shows that the organizational aspects of a software development project are more likely to affect performance and outcomes than technical issues (Xia & Lee, 2004). Successful development and deployment of today's complex educational systems and environments comes with an imperative for an array of different and unique skill sets for the various stages of each project. One can view a software development team as a microcosm of the wider community of practice of software development professionals who work in information and knowledge management in higher education. As Wenger (1998) observes, such communities of practice are not random but constructed around required skills and through a process of negotiation based on mutuality and accountability.

Workforce mobility has increased: academic staff members regularly and easily move between institutions; development and design staff have many opportunities for contract-based work, move to other academic institutions or into the private sector. The ideas that lie behind a successful process or product are increasingly drawn from a wider pool of talent and, as people move around, these ideas are being taken with them and disseminated through informal and new work practices into a wider community of practice. How then does a team, formed to design and develop a technology-rich educational or systems environment, manage and control issues of intellectual capital and intellectual property such that all of those who contribute throughout the life of a project are acknowledged and rewarded fairly and appropriately for that contribution, even after they have left the project?

Team Formation and Relationships

Additional complexity leads to specialization (Jacobson, Booch & Rumbaugh, 1998). New ways of working bring with them a shift in power, where the academic expert will often lack the technical skills, time or resources to turn ideas into reality. Instead, they must rely on a team of experts from other disciplines to interpret their ideas, evolve them, and deliver the finished product. As complexity increases, communication between team members becomes paramount; specialist educational designers are required to translate pedagogy into functional specifications that can be understood by software developers and graphic designers. Modern software teams are project-based, where resources come and go as required.

Software development communities of practice exist within a larger organizational context. Roles and responsibilities will vary and are negotiated depending on the toolset and architecture used, the size of the project, and the culture of the organization (Phillips, 1997; Williamson et al., 2003). Project team members can be full- or part-time employees (academic or non-academic) or contractors retained specifically for the project. As such, these roles exhibit complex relationships and interfaces between each other and the project. In Figure 1, a range of typical roles and relationships found in a tertiary education software development project are shown.

During the various stages of the development process, various players move into prominent roles. One way to illustrate this shifting set of work responsibilities is to list the main players at each stage of the process. We will do this using the classic instructional systems design (ISD) model (Dick & Carey, 1990) as it is so well known. (There are many other models, many of which are discussed in Bannan-Ritland, 2003.) The key players at each stage of the ISD model are listed in Table 1. In reality, each team parcels out the work depending on the skill set of individuals in the team.

It is important to be aware of the different communities of practice that exist in this field and ensure that the role of individual team members is able to be promoted appropriately. Professional

Related Content

A Survey on IoT Authentication Security Service: Open Issues, Security Threats, and Future Solution Direction
Mihir Mehtaand Kajal Patel (2022). *International Journal of Systems and Software Security and Protection (pp. 1-13).*
www.irma-international.org/article/a-survey-on-iot-authentication-security-service/295101

Test-Driven Development: An Agile Practice to Ensure Quality is Built from the Beginning
S. Mark (2007). *Agile Software Development Quality Assurance (pp. 206-220).*
www.irma-international.org/chapter/test-driven-development/5076

Improving the Bluetooth Hopping Sequence for Better Security in IoT Devices
Matt Sinda, Tyler Danner, Sean O'Neill, Abeer Alqurashiand Haeng-Kon Kim (2018). *International Journal of Software Innovation (pp. 117-131).*
www.irma-international.org/article/improving-the-bluetooth-hopping-sequence-for-better-security-in-iot-devices/210459

An Empirical Study on Novice Programmer’s Behaviors with Analysis of Keystrokes
Dapeng Liu, Shaochun Xuand Huafu Liu (2013). *International Journal of Software Innovation (pp. 68-87).*
www.irma-international.org/article/empirical-study-novice-programmer-behaviors/77619

Design and Evaluation of Automated Scoring: Java Programming Assignments
Yuki Akahane, Hiroki Kitayaand Ushio Inoue (2015). *International Journal of Software Innovation (pp. 18-32).*
www.irma-international.org/article/design-and-evaluation-of-automated-scoring/133112