Towards Governance Schemes for Distributed Software Development Projects

Narayan Ramasubbu, Rajesh Krishna Balan
Singapore Management University
[nramasub, rajesh] @ smu.edu.sg

ABSTRACT

Growth in the adoption of distributed software development business models continues to outpace the development of robust governance schemes for them. In the absence of specific well-tested governance schemes, distributed software teams continue to employ the normative process frameworks, and software engineering practices that were developed in the co-located context. In this workshop paper, we present our position on the research direction to develop and test governance schemes specifically suited for distributed software development projects. We draw examples from our recent body of research work to come up with an actionable road map for researchers and practitioners who are working towards establishing governance schemes for distributed software teams.

Categories and Subject Descriptors
D.2.9 [**Management**]: software process models, software quality assurance

General Terms
Economics, Management

Keywords
Software process, software engineering economics, quality management, offshore software development, empirical analysis

1. INTRODUCTION

Spurred by the desire to leverage global resources and the technological advancements to accomplish it, software development projects are increasingly being distributed across multiple development centers across the globe. It is well documented that distributed teams bring new challenges to project management [1, 7, 14, 28]. There is also an emerging body of research contributions to addresses these challenges [2, 3, 12, 17]. In this workshop paper we present our position on the research opportunities that exist and the research direction necessary to develop, implement and test software engineering governance schemes that specifically address the challenges of distributed development projects.

ICSE '08 workshop session , May 10–18, 2008 Copyright info …..

present along with our position on how we could advance the state-of-the-art in each of these areas. We specifically draw examples from our recent body of work in globally distributed software development.

2. "DISTRIBUTEDNESS" AND PROJECT PLANNING

One of the first activities in project planning is the estimation of the cost, time and project personnel necessary for the project. While there is a large body of work on software project estimation in the current software engineering literature [4, 6], very few models account for "distributedness". Accounting for "distributedness" in current software project estimation requires modeling the spatial and temporal separation of team members along with the other typical software engineering parameters such as requirements ambiguity, process maturity, logic and data complexity, etc. We believe that there is a tremendous research opportunity to revamp the extant software project estimation models to suit distributed development by including the spatial and temporal characteristics of distributed teams. In our own work, we are in the process of developing empirically validated "distributedness" measures that can be then used to build specific project estimation models based on the COCOMO platform [5].

Another research opportunity in accommodating "distributedness" in project planning arises from the need to devise suitable labor division policies for distributed teams. Since Conway [9], there is a conjecture that the architecture of a system is mirrored closely in the organization of the team that builds the system. Subsequently, software engineering studies that focused on the team organization and coordination have developed empirical guidelines for software teams working in specific contexts [15, 16, 22]. There is a need to verify these coordination guidelines in a variety of settings such as custom software development, packaged product development in the presence of significant cultural, spatial and temporal differences. Such attempt at generalization of the division of labor and team organization guidelines will be one of the significant first steps towards achieving normative governance schemes for distributed software development projects.

3. "DISTRIBUTEDNESS" AND PROJECT EXECUTION

We identify two areas for fruitful near term research in accommodating "distributedness" in software project execution: 1) Key processes necessary for distributed development 2) Process model choice in a distributed setting. We explain these in turn.

There are numerous software development process frameworks and methodologies available for software teams [11, 13, 18]. The comparative strengths and weaknesses of the prevalent software development processes have also been extensively studied by researchers [20, 23, 24]. However, there has been little work in exploring the factors that contribute to the selection of a specific development process for a specific software project especially accounting for the distributed environment in which the project is executed. It is not fully clear which process models will be suitable for a distributed project and how a project manager can make a priori choice of a process model before the start of the project. Also, it is important to understand the key process areas that are specifically needed for distributed teams and incorporate these key process areas in to the normative process models used for executing distributed software development. For example, the need for having a "work continuity and synchronization" process does not arise in co-located settings but is crucial for distributed teams.

In our research work we address these two aforementioned research gaps through empirical studies of process choices made my distributed teams in the offshore software development setting. We have developed a set of new key process areas that specifically address the needs of distributed product development teams and implemented the distributed process framework at a real world product development setting [26]. We continue to study the effects of the adoption of the new process framework and will be able to report the results during the workshop. In a second study [25], we analyzed the process choices made by two firms for executing more than a hundred offshore software development projects to link the process choice decision with the eventual project performance outcomes. Our process choice model uses six easily measurable inputs that can be obtained at the starting phase of a software project. We then used the model to predict the development process choice that a new project should use. Using the predictive capability of our model we demonstrate the causal linkages between the development process choice and the performance indicators of the project. Through this stream of research we are thus attempting to build a process choice governance scheme in the context of distributed development.

4. "DISTRIBUTEDNESS" AND PROJECT REFLECTION

Project reflection or project postmortem refers to the activities taken to understand and record the learning points from the project for potential future use [8]. Recording and organizing learning points from a project in a co-located setting is much easier than in a distributed setting. In a distributed context, gathering learning points becomes challenging because of the difficulty in mutual trust development [19], existence of knowledge partitioning problems [29], and other coordination challenges [10]. Hence to create a learning organization in the distributed context, it is necessary to revamp the current process prevalent for software project reflection.

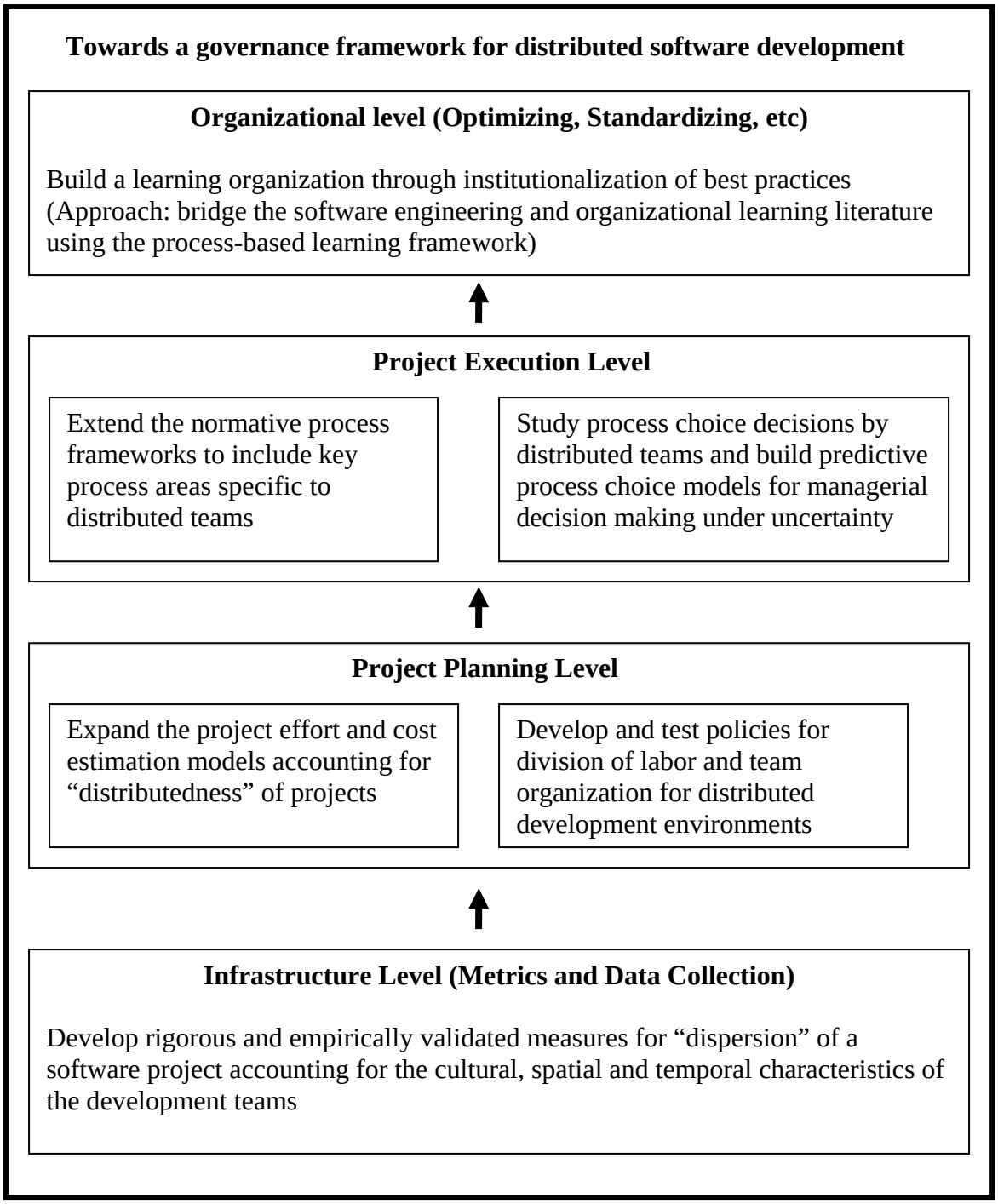
Our approach to creating a learning organization (in the software development context) is called process-based learning [27]. Process-based learning refers to the development of both conceptual (know-why) and operational (know-how) knowledge through a set of activities that are launched through the regular key process areas of software development process frameworks. Rather than having an end-of-the project structured reflection session, we advocate ongoing process-based learning sessions through out the project life cycle. Specifically, we build on the theme that routines or processes, which are the temporal structures in organizations through which work is accomplished, can be an important source of organizational learning [21].

We show that process-based learning conceptual framework, can be utilized to investigate the effect of investments in software process improvement schemes and the development of an organization wide process standard. In doing so, we used an empirical measure for "distributedness" of a software project which accounted for the overall work dispersion across multiple development centers and also the intensity of integration activities between these development centers.

We believe that the process-based learning conceptual framework is just a starting point in investigating distributed software development through the lens of organizational learning phenomena. Further interdisciplinary investigations linking software engineering economics and organizational learning literature holds immense potential especially while striving towards a comprehensive governance scheme for distributed development.

5. RESEARCH ROADMAP

Based on the above discussions, the research roadmap we are pursuing with the goal to bring forth a body of work that contributes to the development governance schemes for distributed development is shown below.



6. CONCLUSION

While distributed development is becoming pervasive through out the software service and product development industry we believe that there is significant research work that needs to be done before we could confidently give specific managerial directions and advice to govern distributed software development. In this workshop paper, we presented our views on how we could move closer towards a governance framework for distributed software development research work in the near term: 1) develop reliable and empirically tested dispersion measures that can be the basis for other software engineering activities 2) extend current project planning models related to cost

estimation and team organization to accommodate the dispersion measures developed in step 1. 3) extend current normative process frameworks to accommodate key process areas specific to distributed environments, and 4) bridge the gap between software engineering and organizational learning approaches to build organizational level practices suitable for distributed software development. We also drew examples from our recent research work to highlight that our research road map can be operationilized through empirical studies. We are hopeful that the broader research community would share our vision and contribute to the advancements of the individual research themes that we have highlighted.

8. REFERENCES

[1] H. P. Andres, "A comparison of face-to-face and virtual software development teams," *Team Performance Management*, vol. 8, pp. 39-48, 2002.

[2] J. M. Bhat, M. Gupta, and S. N. Murthy, "Overcoming requirements engineering challenges: Lessons from offshore outsourcing," *IEEE Software*, vol. 23, pp. 38-44, 2006.

[3] R. Blackburn, S. Furst, and B. Rosen, "Building a winning virtual team: KSAs, selection, training and evaluation," in *Virtual Teams that work: creating conditions for virtual team effectiveness*, C. B. Gibson and S. G. Cohen, Eds. San Francisco, CA: Jossey-Bass, 2003.

[4] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches — A survey," *Annals of Software Engineering*, vol. 10, pp. 177-205, 2000.

[5] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby, "Cost models for future software life cycle processes: COCOMO 2.0," *Annals of Software Engineering*, vol. 1, pp. 57-94, 1995.

[6] L. C. Briand, K. E. Emam, D. Surmann, I. Wieczorek, and K. D. Maxwell, "An assessment and comparison of common software cost estimation modeling techniques," in *21st International Conference on Software Engineering*, Los Angeles, 1999, pp. 313 - 322.

[7] E. Carmel, *Global software teams: Collaborating across borders and time zones*. Upper Saddle River, NJ: Prentice Hall, 1999.

[8] B. Collier, T. DeMarco, and P. Fearey, "A defined process for project post mortem review," *IEEE Software*, vol. 13, pp. 65-72, 1996.

[9] M. E. Conway, "How do committees invent?," *Datamation*, vol. 14, pp. 28-31, 1968.

[10] C. D. Cramton, "The mutual knowledge problem and its consequences for dispersed collaboration," *Organization Science*, vol. 12, pp. 346-371, May-June 2001 2001.

[11] T. Demarco and B. W. Boehm, "The agile methods fray," *IEEE Computer*, vol. 35, pp. 90-, 2002.

[12] C. Ebert and P. D. Neve, "Surviving global software development," *IEEE Software*, vol. 18, pp. 62-69, 2001.

[13] A. Fuggetta and G. P. Picco, "An annotated bibliography on software process improvement," *ACM SIGSOFT software engineering notes*, vol. 19, pp. 66-68, 1994.

[14] I. Gorton and S. Motwani, "Issues in co-operative software engineering using globally distributed teams," *Information and Software Technology*, vol. 38, pp. 647-655, 1996.

[15] J. D. Herbsleb and R. E. Grinter, "Splitting the organization and integrating the code: Conway's law revisited," in *21st International Conference on Software Engineering*, Los Angeles, CA, 1999, pp. 85-95.

[16] J. D. Herbsleb and A. Mockus, "Formulation and preliminary test of an empirical theory of coordination in software engineering," in *11th ACM SIGSOFT international symposium on foundations of software engineering*, Helsinki, Finland, 2003, pp. 138-147.

[17] J. D. Herbsleb, D. J. Paulish, and M. Bass, "Global software development at Siemens: experience from nine projects," in *International Conference on Software Engineering*, St. Louis, MO, USA, 2005, pp. 524-533.

[18] W. S. Humphrey, *A discipline for software engineering*. Boston, MA: Addison-Wesley Professional, 1994.

[19] S. L. Jarvenpaa and D. E. Leidner, "Communication and Trust in Global Virtual Teams," *Organization Science*, vol. 10, pp. 791-815, 1999.

[20] D. H. Kitson and S. M. Masters, "An analysis of SEI software process assessment results: 1987-1991," in *15th International Conference on Software Engineering*, Baltimore, MD, 1993, pp. 68-77.

[21] B. Levitt and J. G. March, "Organizational learning," *Annual Review of Sociology*, vol. 14, pp. 319-40, 1988.

[22] A. Mockus and D. M. Weiss, "Globalization by chunking: A quantitative approach," *IEEE Software*, vol. 18, pp. 30-37, March-April 2001 2001.

[23] M. C. Paulk, "Extreme programming from a CMM perspective," *IEEE Software*, vol. 18, pp. 19-26, Nov 2001 2001.

[24] M. C. Paulk, "How ISO 9001 compares with the CMM," *IEEE Software*, vol. 12, pp. 74-83, Jan 1995 1995.

[25] N. Ramasubbu, R. Balan, and G. K. Tayi, "An Empirical Model of Software Development Process Choice," in *International Symposium of Information Systems* Indian School of Business, Hyderabad, 2007.

[26] N. Ramasubbu, M. S. Krishnan, and P. Kompalli, "Leveraging global resources: A process maturity framework for managing distributed development," *IEEE Software*, vol. 22, pp. 80-86, 2005.

[27] N. Ramasubbu, S. Mithas, M. S. Krishnan, and C. Kemerer, "Project Performance in Offshore Software Development: An Empirical Analysis of the Role of Process-Based Learning Routines," in *2004 Academy of Management Conference Presentation* New Orleans, 2004.

[28] S. Sarker and S. Sahay, "Information systems development by US-Norwegian virtual teams: implicatios of time and space," in *35th Annual Hawaii International Conference on System Sciences*, Hawaii, 2002, pp. 1-10.

[29] A. Tiwana, "Knowledge partitioning in outsourced software development: A field study," in *International conference on information systems*, Seattle, 2003.