Planning and Improving Global Software Development Process Using Simulation

Siri-on Setamanit
School of Business Administration
Portland State University
631 SW Harrison St., Portland, OR
1-503-997-2746
sirion@pdx.edu

Wayne Wakeland
Systems Science PhD
Portland State University
1604 SW 10th Ave., Portland, OR
1-503-725-4975
wakeland@pdx.edu

David Raffo
School of Business Administration
Portland State University
631 SW Harrison St., Portland, OR
1-503-725-8508
davidr@sba.pdx.edu

ABSTRACT

Global software development poses a number of challenges and difficulties as well as significant potential benefits. In order to be successful, companies need to adapt and improve their processes to support this kind of development. Strong project planning and management is also required. Software process simulation modeling has been used to address a variety of issues in software development projects ranging from strategic management, project planning and control, process improvement, to training and understanding. We believe that a hybrid simulation model combining system dynamics and discrete-event models is needed to effectively model global software development projects. In this paper, we describe such a model, and focus on the use of the model to support project planning and process improvement in global software development. Example questions/issues that can be addressed by our GSD model are provided, along with an illustrative application.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management---cost estimation, software process models; I.6.5 [**Simulation and Modeling**]: Model Development; I.6.3 [**Simulation and Modeling**]: Applications; I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis.

General Terms

Management, Measurement.

Keywords

Global Software Development, Distributed Development, Software Process Simulation Modeling, Hybrid Simulation Model.

1. INTRODUCTION

Global software development (GSD) has become a dominant paradigm in the software industry. Almost 100 nations are now participating in GSD [6] expecting to gain benefits including reduction in time-to-market, reduction in development costs, better use of scarce resources, and business advantages from proximity to customers [5, 6, 9, 12, 19]. Unfortunately, GSD also poses challenges and difficulties due to geographic dispersion, time-zone differences, cultural differences and language differences. This can hinder the project from realizing the full potential benefits.

Although several companies report success in their GSD project [2, 3, 5], simply deploying GSD does not guarantee positive results. In order to be successful, companies need to adapt their current processes to support the new distributed development paradigm. This includes:

- The incorporation of new methods, processes, and tools into software development operations
- Strong project planning
- Strong management tracking and control

One area that has shown some success helping companies to address these issues is software process simulation modeling (SPSM). SPSM has been used to address several aspects of software development including [17]:

- Strategic management
- Project planning
- Project management and control
- Process improvement and technology adoption
- Process understanding
- Training and learning

Furthermore, SPSM has also been used to support companies at different levels of the Capability and Maturity Model ranging from designing and defining development process (levels 2 and 3), quantitative process management (level 4), and continuous process improvement (level 5) [24].

Thus, it seems likely that SPSM can also be used to improve GSD processes and support the organizational decision-making necessary for companies to gain the full benefits of GSD. Section 2 provides an overview of our GSD model. Section 3 discusses the questions or issues related to GSD projects that can be addressed by our GSD model. Section 4 provides an example of using the model. Conclusions are given in Section 5.

GSD'06, May 23, 2006, Shanghai, China.

2. A GSD SIMULATION MODEL

2.1 Hybrid Simulation Model

We believe that a hybrid simulation model combining the system dynamics and discrete-event paradigms is the most appropriate for representing GSD project [22]. One strength of the system dynamics paradigm is its ability to capture the dynamic nature of software development projects. Therefore, it is suitable for modeling continuous factors and their interactions, including communication, coordination, cultural issues, learning curve, changing staff levels, and dynamically varying productivity. Discrete-event simulation has the ability to capture actual process level details, and the ability to represent each work product of the development process as being unique--through the use of attributes attached to each work product, such as size and complexity. Thus, it provides the ability to explicitly represent the process structure and mechanisms used to transfer work products and to coordinate activities. These two paradigms complement each other and allow our GSD model to capture the actual development process executing within a continuously changing project environment.

2.2 Important Factors

In our previous work [22], we have identified the factors that can affect the performance and the productivity of GSD projects. These factors can be categorized into three categories, as follows:

2.2.1 Fundamental Factors

The fundamental factors relate to the primary characteristics of GSD projects, including communication problems, coordination and control problems, cultural differences, language differences, and time-zone differences. A project manager has little or no control over these factors; however, by using the right strategy and tool support, the negative impact of these factors can be reduced.

2.2.2 Strategic Factors

The strategic factors are related to high level issues that the project manager must address when managing a GSD project. Decisions regarding these issues significantly impact the performance of the GSD project. There are five such factors, including development site, product architecture, task allocation strategy, distribution overhead, and distribution effort loss.

2.2.3 Organizational Factors

The factors in this category are concerned with the impacts from virtual teams, including team formulation and team dynamics.

Additional explanation of these factors may be found in [22]. Our GSD model incorporates all of the factors mentioned above. The next section describes the GSD model structure, and how it incorporates these important factors.

2.3 GSD Model Structure

At a high level, the GSD model has three major components: Discrete-event (DES) sub-model, System Dynamics (SD) sub-model, and Interaction Effect (IE) sub-model.

The DES sub-model includes a global DES sub-model and a site-specific DES sub-model for each development site. Each development site may have different process steps depending on task allocation strategy. The site-specific DES allows us to capture the impact of these differences. Different time-zones are also modeled. Work products are passed from one site to another in order to capture the effect of distribution overhead and distribution effort loss. The global DES sub-model aggregates the information from the site-specific DES sub-models to determine overall project progress.

The SD sub-model includes a global SD sub-model and a site-specific sub-model for each development site. The global SD sub-model captures the overall project environment, including the planning and controlling activities. The global SD sub-model has three modules: Human Resource (HR), Planning, and Control. The Human Resource module acts as an interface between HR module from each development site and the other modules within the global SD sub-model. The Control module receives information about the project progress (from the global DES sub-model) and then evaluates whether adjustments to the schedule or the work rate are needed. The Planning module monitors and identifies the workforce level required to meet the overall project schedule.

Each development site has its own site-specific SD sub-model. The site-specific SD sub-model represents aspects that may be different between development sites, including HR, productivity (PD), manpower allocation (MP) and defect generation and detection rates (QA). The HR module deals with human resource management, which includes hiring, training, assimilation, and transferring human resources in a particular site. The PD module models the rate at which the developers at a particular site can develop software (productivity rate). The MP module assigns workforce to different activities. The QA module models defect generation, detection, and correction rates. These modules function as if there were only one development site in the project. For example, site-specific productivity assumes that developers are working with others from the same site.

The IE sub-model represents the interaction effects, when staff from different sites need to collaborate or work closely together such as during follow-the-sun development. When developers work with their colleagues from the same site, information such as productivity and defect rates will be sent from site-specific SD sub-model. However, when developers have to collaborate with their colleagues from other sites, their productivity will be different. The IE sub-model modifies the productivity before sending it to the DES sub-model. Fundamental factors and organizational factors are mainly captured in this sub-model. Additional discussion about IE sub-model is provided in Section 2.4.

Figure 1 shows the overall GSD model structure with two development sites.

2.4 Interaction Effect (IE) Sub-model

The structure of the interaction effect (IE) sub-model is based on literature regarding global software development, distributed development, and virtual teams. In this paper, we focus on the interaction effect on productivity rate since it has the strongest impact on project performance.

Software development requires tight coordination among various efforts in order to be successful [18]. Difficulty in coordination leads to losses in productivity [4, 25]. Due to distance and time-zone differences, coordination in multi-site development team is difficult, and can negatively affect productivity. The IE sub-model calculates the coordination efficiency of the distributed team (relative to the coordination efficiency of the single-site team), and then applies the coordination effect to the productivity before sending it to the DES sub-model. If the coordination efficiency of the distributed team is lower, the productivity will be lower. Figure 2 shows the interaction effect on productivity rate. There are two factors that determine the coordination efficiency: communication frequency and trust.

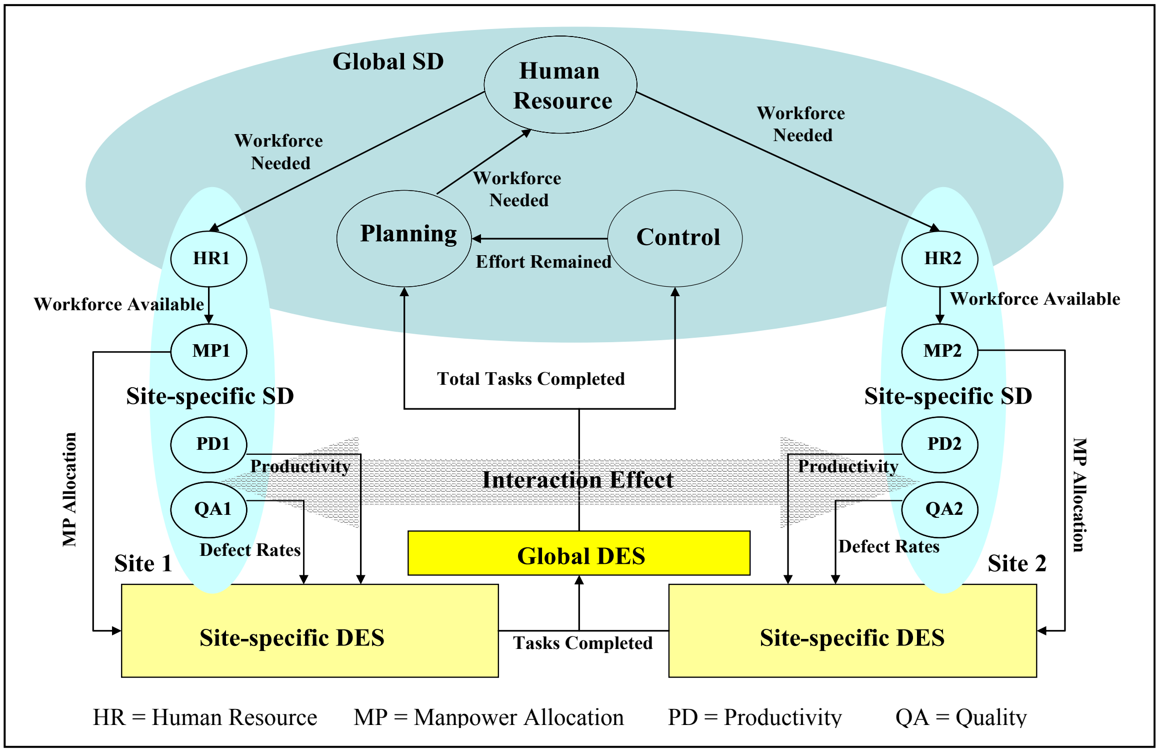

Figure 1. Overview of the GSD model structure

2.4.1 Communication Frequency

Several studies of software development projects [8, 11, 18, 21] found that informal, unplanned, and ad hoc communication is extremely important in supporting collaboration. Discussion with peers is the most used and valued coordination techniques [18]. This indicates that teams with frequent communication among the members tend to coordinate better. Unfortunately, the distance between team members negatively affects the amount of communication [1]. The IE sub-model will determine the relative frequency of communication when team members are at different sites compared to when they are at the same site. The relative communication frequency will positively impact the coordination efficiency.

2.4.2 Trust

Trust is a basic feature of social situations that require cooperation and interdependence [15]. Teams with higher trust tend to coordinate better, thus achieve better performance [14]. In the GSD model, trust is modeled as a dynamic variable. The initial level of trust is determined by culture (individualist or

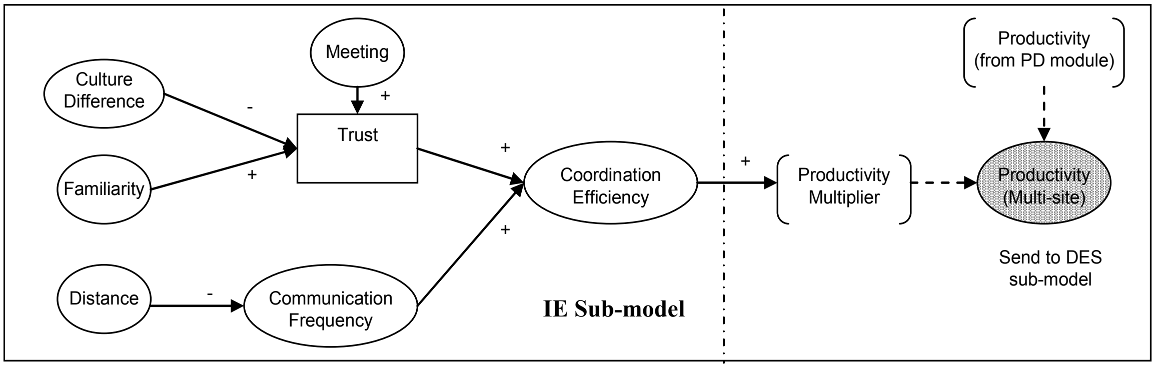

Figure 2. The interaction effect on productivity rate

collectivist as defined by Hofstede [13]) and team member familiarity [7]. People from same culture develop trust more quickly than people from different cultures. In addition, individuals from an individualistic culture tend to be more ready to trust others than individuals from a collectivist culture [10, 20]. Trust will be higher when team members become more familiar with each other during a project. However, without face-to-face communication, trust can also decrease overtime.

3. THE USE OF GSD MODEL

In this paper, we focus on the use of the GSD model for project planning and process improvement.

3.1 Project Planning

The GSD model can assist the project manager in planning. Example project planning questions that the GSD model can address are listed below.

What is the forecasted project performance in terms of cost, quality, and schedule?

At a basic level, the project manager can tailor the GSD model to a specific project in order to estimate the performance of the project in terms of effort spent, duration, and quality. In addition, the project manager can vary parameters such as the number of developers, and see the impact.

The GSD model can also be used to forecast the staffing levels needed across time. This will help the project manager to plan his/her resource allocation and cope with resource constraints.

Should the project be distributed across multiple sites or should it be centralized at a single site?

The GSD model is designed to be flexible and expandable. At the simplest level, the GSD model can be used to represent a single site development project (with only one site-specific SD and one site-specific DES sub-model). The project manager can run the model, and see the performance of the project with one site. He/she can then add another site and see the effect on the project performance.

Which development site should be included in the project?

The site-specific SD and the site-specific DES sub-models allow the GSD model to capture different characteristics of the development sites such as communication infrastructure, number of staff, level of experience, productivity rate, culture, native language, and etc. The project manager can vary parameters to represent different development site and decide which development sites should be included in the project. For example, the project manager can evaluate whether it is worthwhile to switch to low cost (but potentially lower productivity) development site.

How should work be divided across sites? What task allocation strategy should be used?

The site-specific DES sub-model allows us to capture the detail process steps in each development site. In addition, the work product transfer between sites can also be represented by passing entities in the model. Therefore, the project manager can experiment with the GSD model by changing process steps or the task allocation strategy in order to identify the best alternative.

3.2 Process Improvement

The GSD model can support process improvement by analyzing the impact of potential process changes. The project manager can compare the current performance (as-is) versus the expected performance after the improvement (to-be) and then decide whether or not to implement the change. The following paragraphs provide example process improvement questions that the GSD model might be able to address.

Where should the testing be performed? Should it be centralized?

The discrete-event sub-model represents actual process steps, which allows the project manager to change the location where the testing will be performed. By comparing the project performance for different configurations, the project manager will be able to determine where the testing should be performed. This sort of analysis can also be done for other software development activities.

What is the impact of adding an additional process step (such as inspection)? Can we minimize or skip a portion of an inspection?

Again, the project manager can easily model such a change by modifying the discrete-event sub-model to reflect the particular process change of interest. By comparing the project performance after the process change to the baseline performance, the project manager can determine whether such a change might be worthwhile or not.

What is the impact of implementing new collaborative support tool?

By implementing collaborative tool, it is expected that the developers will be able to better communicate and coordinate with their colleagues at different sites. The effect of this tool can be captured in the interaction effect (IE) sub-model (e.g. coordination efficiency). This allows the project manager to determine whether the new tool actually improves the overall project performance, before making the investment.

Would it be worthwhile to improve communication infrastructure between development sites?

By improving communication infrastructure between development sites, work product can be transferred faster and the developers can communicate more efficiently. The distribution overhead captured in the GSD model includes the time it takes to actually transfer the work product. Therefore, the project manager can experiment with the distribution overhead parameters and determine whether the improvement in project performance would justify the investment to improve the infrastructure.

In addition to project planning and process improvement, the GSD model can also aid in understanding the project development process, and to facilitate training. The GSD model offers a safe learning environment for people to practice and learn about project management and the complex interactions that occur within GSD projects. Management trainees can see the impacts of their decisions, and also learn how to better predict the behavior of projects.

4. EXAMPLE

To illustrate the functionality and the usefulness of the GSD model, we created a simple example project. There are 5 phases in the development process including Requirements (REQ), Design (DES), Coding (CODE), Testing (TEST), and Rework (RWK). Due to resource constraints, the main development site is unable to complete the project within the necessary time window. Therefore, the project manager needs to find additional development sites to help perform the work. There are 2 alternative sites:

Site A (offshore):

- Eight hour time-zone difference (no overlap working hour)
- The culture and native language of the developers are different.
- The programmer wage is lower than the current site and Site B.

Site B (near-shore):

- Only four hour time-zone difference (50% overlap working hours)
- The culture and native language of the developer are the same.
- The programmer wage is higher than Site A.

Both sites have advantages and disadvantages that may or may not offset each other. For Site A, we can benefit from 16 hour development time per day by using follow-the-sun development strategy. However, there may be more coordination and communication problems since the developers have different culture and language. This difficulty may require additional effort to complete the project.

On the other hand, the development time per day if using Site B is lower. However, the overlap working allows synchronous communication between two sites, which can result in better coordination and communication. In addition, since culture and language of the developers are the same, miscommunication and coordination problems tend to be lower, resulting in higher productivity. One can see that it is not obvious which development site would be better. This is where our GSD model comes in. We can use the GSD model to analyze and compare the performance of each alternative before making a decision.

4.1 Result

We first configured the GSD model to have two development sites-- the current site and Site A. We ran the model (30 replications) to obtain the expected project performance, including effort, duration, and quality (number of latent defects). We then re-configured the GSD model with Site B rather than Site A. We ran 30 replications, and studied the project performance in each case. We also performed hypothesis tests to determine if the differences in project performance are statistically significant. The results are shown in Figure 3 using the familiar box and whisker plots to show the variation.

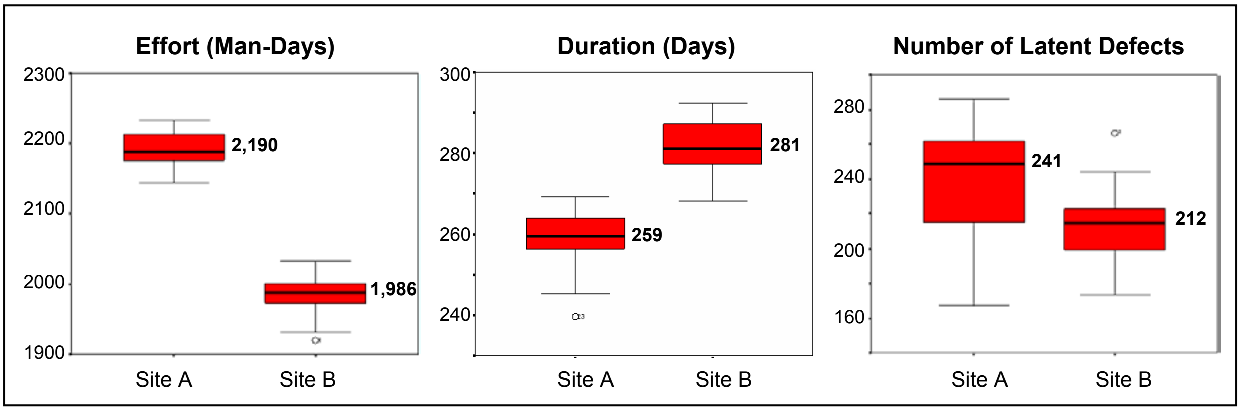

Figure 3. Comparison of effort, duration, and defects using Site A vs. Site B

4.1.1 Effort

The total effort required to complete the project is higher when using Site A (204 man-days more). The difference is significant at .05 level. Working with Site A requires additional effort for coordination. In addition, miscommunication tends to be higher, which results in higher defects. These defects also require additional effort to rework. However, since the programmer wage in Site A is lower, we cannot automatically conclude that Site B would cost less than Site A. The GSD model also records effort spent at each development site. Figure 4 shows the effort distribution between sites for each alternative (Site A vs. Site B).

One can see that the effort expended at main site is approximately the same for two alternatives. The additional effort required for alternative 1 is mainly spent at Site A. In this example, if the wages at Site A are approximately 80% of the wages at Site B, the cost to hire the developers will be about the same.

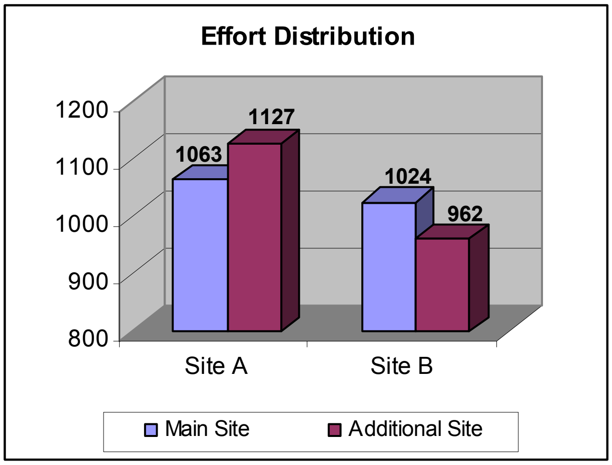

Figure 4. Effort distribution comparison

4.1.2 Duration

The duration with Site A is shorter than the duration with Site B (259 vs. 281 days). The difference is statistically significant at .05 level. This may due to the benefits of having more development time per day (16 hours). Although, we have to expend more effort when using Site A, the larger number of work hours reduces the project duration. It should be noted that there is about a 4% probability that the duration with Site A will be longer than the duration with Site B.

4.1.3 Quality

We measure the quality of the software as the number of defects escaped or latent defects. The latent defects with Site B (mean = 212, std. dev. = 21) is lower than the latent defects with Site A (mean = 241, std. dev. = 30). The difference is statistically significant at .05 level. As mentioned before, difference in culture and language, and difficulty in communication and coordination are likely to result in more defects being generated. Since both Site A and Site B have about the same capability to detect and correct defects, the site with the higher number of defects injected will be likely to also have the higher number of defects escaped. Nevertheless, it should be noted that there is a 15% chance that the quality will be lower (more latent defects) when using Site B versus Site A.

4.2 Discussion

Neither Site A nor Site B performs the best on all three performance measures. Adding Site B to the project results in less effort required and higher quality software (lower defects), but the project duration will be longer. On the other hand, adding Site A contributes to shorter project duration, but higher effort and lower quality. The project manager has to make a trade-off between these three performance measures and determine which alternative will better meet their objectives. For example, if the goal is to reduce the cycle-time, adding Site A will best satisfy this objective. However, if the project manager is more concerned with the quality of the software, he may choose to add Site B instead. In addition to the simple comparisons above, there are several ways to compare the simulation result and evaluate alternatives such as using a utility function, using financial performance measures, and using Data Envelopment Analysis (DEA). We will not go into detail about these. Thus, Interested readers may find additional information and examples in [23].

Again, the example provided above is just a hypothetical example to illustrate how the project manager can use the GSD model to answer his/her question of interest. The example was calibrated with information from the literature and the industry standard data collected by Capers Jones [16]. We are currently collecting real-world data in order to calibrate the model to specific real world projects.

5. CONCLUSION

In this paper, we present our GSD model that features an innovative hybrid architecture which combines the system dynamics paradigm with the discrete-event paradigm. The hybrid feature allows the GSD model to better represent the GSD project. We believe that such a model can be used as a decision support tool to help project managers plan, manage, and improve global software development processes. Examples are provided to illustrate types of managerial questions that can be addressed, and to show how the GSD model can be used to help address these questions.

6. REFERENCES

[1] Allen, T.J. *Managing the Flow of Technology*. MIT press, Cambridge, MA, 1977.

[2] Bass, M. and Paulish, D., Global Software Development Process Research at Siemens. in *The 3rd International Workshop on Global Software Development*, (Edinburgh, Scotland, 2004), 8-11.

[3] Battin, R.D., et al. Leveraging Resources in Global Software Development. *IEEE Software* (March/April. 2001), 70-77.

[4] Brooks, F.P. *The Mythical Man-Month*. Addison-Wesley, Reading, MA, 1975.

[5] Carmel, E. *Global Software Teams*. Prentice Hall PTR, Upper Saddle River, NJ, 1999.

[6] Carmel, E. and Tija, P. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press, Cambridge, UK, 2005.

[7] Consortium, S.P. Measurement for Distributed Teams, Software Productivity Consortium, Herndon, Virginia, 2002, 68.

[8] Curtis, B., Krasner, H. and Iscoe, N. A Field Study of the Software Design Process for Large Systems. *Communications of the ACM*, *31*, *11* (1988), 1268-1287.

[9] Gorton, I. and Motwani, S. Issues in Co-operative Software Engineering Using Globally Distributed Teams. *Information and SoftwareTechnology*, *38* (January. 1996), 647-655.

[10] Gudykunst, W.B., Matsumoto, Y., Ting-Toomey, S., Nishida, T., Linda, K.W. and Heyman, S. The influence of cultural individualism-collectivism, self construals, and individual values on communication style across cultures. *Human Communication Research*, *22*, *4* (1996), 510-543.

[11] Herbsleb, J.D. and Grinter, R.E., Splitting the Organization and Integrating the Code: Conway's Law Revisited. in *International Conference on Software Engineering (ICSE'99)*, (Los Angeles, CA, 1999), ACM Press, 85-95.

[12] Herbsleb, J.D. and Moitra, D. Global Software Development. *IEEE Software* (March/April. 2001), 16-20.

[13] Hofstede, G. *Culture's Consequences: Comparing values, behaviors, institutions, and organizations across nations*. Sage Oublications, Inc., Thousand Oaks, CA, 2001.

[14] Jarvenpaa, S.L. Communication and Trust in Global Virtual Teams. *Journal of Computer Mediated Communication*, *3*, *4* (1998).

[15] Jennings, E.E. *Routes to the Executive Suite*. McGraw-Hill, New York, 1971.

[16] Jones, C. *Applied Software Measurement: Assuring Productivity and Quality*. McGraw-Hill, New York, 1997.

[17] Kellner, M.I., Madachy, R.J. and Raffo, D.M. Software Process Simulation Modeling: Why? What? How? *Journal of Systems and Software*, *46*, *2/3* (April 15. 1999).

[18] Kraut, R.E. and Streeter, L.A. Coordination in Software Development. *Communications of the ACM*, *38*, *3* (1995), 69-81.

[19] Norbjerg, J., Havn, E.C. and Bansler, J.P., Global Production: The Case of Offshore Programming. in *Wirtschaftsinformatik '97*, (Physica-Verlag, Berlin, 1997).

[20] Pearce, W.B. Trust in interpersonal communication. *Speech Monographs*, *41* (August. 1974), 236-244.

[21] Perry, D.E., Staudenmayer, N.A. and Votta, L.G. People, Organizations, and Process Improvements. *IEEE Software*, *11*, *4* (1994), 36-45.

[22] Raffo, D. and Setamanit, S., A Simulation Model for Global Software Development Project. in *The International Workshop on Software Process Simulation and Modeling*, (St. Louis, MO, 2005).

[23] Raffo, D.M. and Kellner, M.I. Empirical Analysis in Software Process Simulation Modeling. *Journal of Systems and Software*, *49*, *9* (2000).

[24] Raffo, D.M., Vandeville, J. and Martin, R. Software Process Simulation to Achieve Higher CMM Levels. *Journal of Systems and Software*, *46*, *2/3* (April 15. 1999).

[25] Steiner, I.D. Models for Inferring Relationships Between Group Size and Potential Group Productivity. *Journal of Behavioral Science*, *5*, *11* (1966), 273-283.