Analytical Modeling of Software Development Teams in Globally Distributed Projects

Ricardo M. Czekster, Paulo Fernandes, Afonso Sales, and Thais Webber

Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681 – Prédio 32 – 90619-900 – Porto Alegre, Brazil
{ricardo.czekster, paulo.fernandes, afonso.sales, thais.webber}@pucrs.br

Abstract—Global software engineering is an area of increasing research challenges, in which teams are dispersed in multiple sites collaborating across borders and time zones. In spite of its potential competitive advantages, globally distributed projects must deal with difficulties when distributing resources such as teams with cultural diversities, different skills and experience levels. Both industry and academia demonstrate a special interest in the modeling and prediction, mainly representing systems in order to extract interesting indices, for example, evaluating whether or not a project will succeed. This paper demonstrates the usefulness of analytical modeling techniques in order to predict the outcome of geographically-distributed projects. We focus our attention to the participants interaction and its interplay when it affects team productivity. The models are parametrized considering single-site and multi-site scenarios, varying resources availability, teams expertise and support levels. Performance indices from both scenarios are presented and conclusion indicates possible model extensions.

Keywords—Geographically-Distributed Project; Global Software Development; Analytical Modeling

I. INTRODUCTION

In recent years the adoption of global software development (GSD) has become a common practice in the software industry. However, several challenges arise such as communication, coordination and cultural diversity across multiple sites in different geographical locations.

Effective local interactions within a multi-site context plays an important role in GSD, directly impacting in teams productivity. To reduce the need of external support, research results point out that the project must be divided in self contained units, loosely coupled, to maximize work periods [1], [2]. In conjunction with availability and participants expertise, these characteristics have been subjected to many researches in past years, where there is a lack of quantitative analysis to discover their influence on GSD contexts.

The use of analytical modeling in software engineering contexts has been successfully used before to provide quantitative performance measures [3], [4]. There are basically three ways to obtain results: (i) monitoring - providing empirical results from analysis of team behaviors and interactions [5], [6]; (ii) simulation - analysis of the evolution and intercommunication of software development processes in order to help project managers comprehend the impacts related to the global context [7], [8], [9]; and (iii) modeling - identification of entities and relations considering spatial and temporal boundaries in distributed projects [10].

Related works concerning stochastic models and simulation are developed towards to the specification of the dynamics of software projects [11], and the usage of analytical models to analyze teams productivity variability [12]. Considering the fact that the performance analysis of geographically dispersed teams is emerging [13], [14], [2], [15], [1], advances are still needed for the quantitative evaluation of such systems using stochastic modeling as a tool.

Stochastic Automata Networks (SAN) [16], [17] is a powerful modeling formalism based on Markov chains that provides a high-level description (abstraction). Its basic idea is to represent a system by a set of modules with an independent behavior and occasional interdependencies. Furthermore, SAN is a suitable formalism for modeling globally distributed projects due to the fact that development teams can be smoothly abstracted in a modular way. Basically a module is described by a stochastic automaton depicted by a state-transition diagram, where the transitions are labeled with probabilistic and timing information. A SAN model has a set of events which triggers changes of the state of one or more automata. Each event has an estimated duration, which indicates how often this event occurs. Solving an analytical model numerically, one can obtain its steady-state probabilities [18], [19] and, hence, it is possible to extract measures of interest (e.g., performance indices) about the system under evaluation.

Both industry and academia have a special interest in the modeling and prediction of the behavior of software development processes, teams compositions and evaluation, i.e., estimating performance indices in assumed scenarios with a variety of parameters (e.g., different skills, experience levels and availability for collaboration). In GSD teams, the participants spend large amounts of their time interacting and communicating, and it is well known that despite best efforts at communicating among dispersed sites, GSD brings more challenges than single-site development [14], [1].

This paper demonstrates the usefulness of analytical modeling within geographically-distributed projects, presenting the solution of models for both single-site (Section II) and multi-site (Section III) software development projects. We focus our attention on the impact of the interactions on team members’ productivity. It is not the goal of this paper to analyze the
impact of coordination or cultural diversity, even though these aspects are often quite relevant. However, this first modeling effort only aims on the impact of communication issues that are the central concerns in GSD projects.

Various scenarios are evaluated and analyzed showing the trade-off of choosing different team sizes and compositions. Conclusions point out future works directed to the extension of models in order to capture advanced characteristics such as cultural issues and diverse communication problems that arise in GSD projects.

II. SINGLE-SITE DEVELOPMENT CONTEXT

This section presents an analytical model of a development team where its members are located in a single-site. It corresponds to a self-contained environment, modeling a leader, who can also be seen as a project manager, supplier manager or R&D manager; and members that assume different roles in a project such as developer, designer, architect, QA expert, and so on.

Fig. 1 shows the development team interaction pattern for our single-site context, i.e., the relations between leader and members. On this perspective, we consider a team composed of single members to a maximum size of $N$ members attached to a leader. The development team behavior states that the members work on their assigned tasks, performing local cooperation with other members, and interact with the leader. The leader has an holistic project view and the capability to reassign tasks, to consider minor decisions and to carry out new (small) developments, according to project demands.

The mapping of the development team interaction pattern (Fig. 1) to a SAN model is straightforward, as presented in Fig. 2. The main entities (leader and member) are replaced by automata. The abstraction, for instance, represents the leader having two major assignments within a project: some of the time is spent on Management ($Mg$) and the rest in Collaboration ($Co$) with the members. Each member is represented by four states that corresponds to Working ($Wk$), Waiting ($Wt$) for cooperation, Collaborating ($Co$) with the leader and Reworking ($Rw$). The specific descriptions for all states are better explained in Table I.

Assuming that a member is in its working state ($Wk$), it transits to the waiting state ($Wt$) when it needs more information or help related to its assignment or reviewing. At this point, the member is actually waiting to collaborate with the leader, so, they must synchronize, enabling communication exchange (i.e., the leader must not be collaborating with other members). In real world projects, this state could be replaced by other states representing tasks that are independent of leader feedback, however in earlier phases of software development some tasks can be strongly dependent of managers decisions.

In this model we focused on that dependent behavior in order to analyze productivity indices in these extremal conditions. After collaboration ($Co$), the member can resume working ($Wk$) or realize that some parts of a task needs improvements (or some bug has to be corrected, for example). In that case, the member performs rework ($Rw$) in its assignments and only then starts to work again ($Wk$), repeating this process iteratively. In the same way, after collaboration ($Co$), the leader returns to the management state ($Mg$).

---

We assume in our abstraction that the workload (tasks) are equally distributed among the members. We are not considering levels of complexity for each task, they are equalized,
i.e., the duration and effort required for its completion is comparable among other important tasks, such as component design, testing or implementation. We also assume that the members have the same set of skills and individual expertise. In addition, any questions, remarks and task conflicts must be properly evaluated by the leader. This assumption avoids the situation where the members spend time reworking some tasks, assuring a satisfactory project evolution from the leader’s perspective. Whenever members need assistance, they stop working and wait for collaboration with the leader, i.e., the leader must be in the management (Mg) state and not in collaboration with other members.

Our single-site model has four events (me, s, a and r) to represent the expertise, the capability to solve issues demanded by members, the availability of the leader and the amount of time that rework takes to be done, respectively. Every state change and interactions among leader and members happen due to the occurrence of these events. Table II shows all events and its description from the model depicted in Fig. 2.

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>me</td>
<td>Member’s Expertise: expresses the level of expertise of a member to deal with its own issues.</td>
</tr>
<tr>
<td>a</td>
<td>Availability: symbolizes the leader’s availability to cooperate with a member.</td>
</tr>
<tr>
<td>s</td>
<td>Support: characterizes the leader’s expertise in solving issues demanded by members.</td>
</tr>
<tr>
<td>r</td>
<td>Rework: represents the level of experience and skills of a member for reworking in a given task.</td>
</tr>
</tbody>
</table>

This section showed the single-site context model and explained how the leader interacts with the members in a project. Section II-A presents the estimated durations of each event and the single-site scenarios to be analyzed.

A. Assigning parameters

Describing only the entities and its relationships in a model is insufficient, lacking the durations that every entity rests in a given state. These information are also named as model parameters and is paramount in analytical modeling, due to its expressiveness. According to estimations based on empirical experience, the main idea is to assign durations to every state in the model, i.e., frequencies at every connection among states. Table III shows the estimated durations for the events in the single-site model, assuming as reference an eight-hour workday.

We are varying the levels for every event estimation except for rework (which has a fixed value). Note that we have exaggerated the range assigned to these parameters in order to analyze more effectively the impact of them in our study.

Therefore, combining the estimated durations possibilities presented in Table III, we assemble eight possible scenarios as follows: ILA, ILB, IHA, IHB, ELA, ELB, EHA, and EHB. For instance, ILA scenario represents a team with inexperienced members (I), a leader that provides a low support (L) - i.e., an inexperienced leader in solving issues demanded by members - but often available to cooperate with the members (A). The rest of the scenarios follows the same idea. We choose to vary the parameters of the three major aspects that our model captures, looking for results that promote a better understanding on how those variables behave together.

<table>
<thead>
<tr>
<th>Event</th>
<th>Estimated durations for an eight-hour workday</th>
</tr>
</thead>
<tbody>
<tr>
<td>me</td>
<td>(I)inexperienced: member demands cooperation in average every 90 minutes, i.e., a member with a lack of required skills (or low expertise) to accomplish its own responsibilities.</td>
</tr>
<tr>
<td></td>
<td>(E)xpert: member requires cooperation in average once a day. An expert member presents a high level expertise related to its role.</td>
</tr>
<tr>
<td>s</td>
<td>(L)ow: leader requires in average 90 minutes for solving issues demanded by members.</td>
</tr>
<tr>
<td></td>
<td>(H)igh: leader requires in average 30 minutes for solving issues demanded by members.</td>
</tr>
<tr>
<td>a</td>
<td>(A)vailable: leader cooperates with a member every 30 minutes.</td>
</tr>
<tr>
<td></td>
<td>(B)usy: leader presents low availability due to management duties. A busy leader cooperates only once a day.</td>
</tr>
<tr>
<td>r</td>
<td>(R)ework: member requires in average 120 minutes to review/correct its tasks.</td>
</tr>
</tbody>
</table>

The parametrization completes the model, which can now be subjected to specialized tools for numerical solution (in our case, PEPS software tool [18]) in order to extract performance indices and measures. Thus, Section II-B presents some performance indices from the aforementioned scenarios emphasizing estimations of team members’ productivity.

B. Measuring productivity

An effective manner to measure productivity is to concentrate on accomplished project tasks. There are different factors that affects productivity such as perceived schedule versus actual schedule, increased interactions due to project milestones and tasks misconceptions or changes in requirements. The measure of team productivity is also influenced by the developers morale, as they must constantly feel that the project is smoothly progressing [2].

In our model, productivity is captured by the working (Wk) probability, where team members generate output and contribute to project completion in a promptly manner. The duration spent on this state reflects the amount of work that has been produced as they influence the other states for each member, i.e., the time is balanced with waiting for the leader (Wl), collaborating (Co) and performing rework (Rw).

The following figures and tables demonstrate the main results obtained from the single-site model, varying the team size from two to nine members. According to Sangwan et al. [2], it is suggested that development teams be no larger than ten participants. Fig. 3 and Fig. 4 depict the team leader’s responsibilities analysis and tables show the team members’
performance indices (represented by the probabilities of being in states \(Wk\), \(Wt\), \(Co\) and \(Rw\)).

Fig. 3 presents the results for expert team compositions (E) varying leader support (H or L) and availability (A or B). In order to highlight the increase (+) or decrease (−) for measures of the members, we have calculated an indice (%) called \(\Delta\) that relates the second with the ninth member together, showing the gain or loss accordingly. For example, considering EHA scenario in Fig. 3 (a), the probability for the working state (\(Wk\)) measure for \(N=2\) members corresponding to 84.62%, decreases to 79.80% when \(N=9\) members, i.e., \(\Delta≈-5.70\%\).

Moreover, Fig. 3 (a) shows other interesting results for comparing team leader’s responsibilities analysis and its impact on the team members. Note that the increase of the waiting probability (\(Wt\)) is far more dramatic in ELA than EHA scenario, corresponding to a fall of \(\approx31\%\) of work and an augmentation of \(\approx384\%\) of waiting due to low support provided by the leader to the team members.

ELA and EHA scenarios present a trade-off that must be observed according to team size. If the team size increases, the team members tend to wait more for the team leader to be available and less time being productive. The team leader’s responsibilities are also affected. For small sized teams, the leader is usually available to respond to members. However, as the team grows in terms of size, the leader ceases doing management (\(Mg\)) - as demonstrated in the graphics in Fig. 3 (a) - and starts doing more collaboration (\(Co\)).

The graphics also shows that when \(N=9\) members, ELA scenario reaches \(\approx87\%\) of the time under collaboration, whereas for EHA scenario, this indice is \(\approx45\%\). If the leader provides high support (H) to the team members, even when the team size is increased, the leader can still supply requisitions keeping a higher \(Mg\) probability than \(Co\). However, the leader begins to perform more collaboration than management from \(N=4\) members, when we have a team leader that provides low support (L) to the team members.

In contrast to Fig. 3 (a), where the leader is often available (A) to collaborate with the team members, Fig. 3 (b) shows the results for a busy (B) leader. In EHB scenario, when team size is increased, the team productivity decreases \(\approx10\%\), whereas waiting probability augments about \(11\%\).

Another interesting fact is presented in ELB scenario, where the variation (\(\Delta\)) is more intense. The team productivity decreases \(\approx28\%\) while the waiting increases about \(\approx31\%\), and both probabilities began somewhat similar (\(\approx43\%\) and \(\approx47\%\)) for \(N=2\) members. The team members spent most of the time waiting for a response from the leader, instead of working. For the ELB scenario this fact is more evident as team size is increased (for \(N=9\) members, the waiting probability is twice higher than the working probability). The graphics in Fig. 3 (b) shows that a busy leader does not spend much time in collaboration (\(Co\)), but rather stays in management (\(Mg\)). The consequence is that the members’ productivity drops considerably when comparing with the results presented for EHA and ELA scenarios.

---

**Fig. 3.** Expert team analysis (single-site context)
Fig. 4 shows the results for inexperienced team compositions (I) varying leader support (H or L) and availability (A or B). In ILA scenario - Fig. 4 (a), even for small team sizes (for N=2 members), the leader has greater probability of being collaborating than managing due to the fact that the team members are inexperienced. Despite that fact, for teams having more than five members, the leader has more than 90% of probability of being cooperating, impairing the management responsibility (reaching ≈4% for N=9 members).

Analyzing team members’ performance in Fig. 4 (a), we observe that when the participants are inexperienced, it is better to have a team leader which provides high support (H) than low support (L) to the team members. This is evident in IHA scenario, where the productivity (W(k)) is equal to 48.60% for N=2 and 27.59% for N=9 members, whereas, in ILA scenario, the productivity varies from 31.67% to 10.59% respectively. However, the waiting probabilities (W(t)) are more evident for both scenarios, since the variation (Δ) is beyond 150%, which means that for a team with nine members, for instance, a substantial part of a workday is spent waiting for the leader to collaborate.

Fig. 4 (b) presents the scenarios with a busy team leader which provides high and low support. Even though the leader is frequently busy, i.e. having less time for collaboration (Co) with members, in ILB scenario from six to nine members the leader starts performing more collaboration (Co) than management (Mg). Analyzing team members’ performance for inexperienced teams having a busy leader (IHB and ILB scenarios), we notice high waiting probabilities (W(t)), more than 80%, for teams bigger than five members.

Observing both Fig. 3 and Fig. 4 and analyzing only when N=9 members, the best case scenario is EHA, which has 79.80% of members’ productivity, whereas ILB scenario reaches only 6.64% (the worst case scenario). These results indicate that it is better to compose a team with experienced members and an available leader who provides high support to them than inexperienced team members with a busy leader who provides low support. This is quite obvious in practice, however, now it is possible to understand the difference between these scenarios from a quantitative point of view and how team size affects the team members’ performance indices.

Fig. 5 shows the last case to analyze for single-site development context, where the impact on team members’ productivity is considered according to members’ expertise and leader’s availability. The figure distinctly points out that, for small sized teams, it is better to have an available leader that provides high support to an inexperienced team (ILA scenario) than a busy leader providing support to an experienced team (EHB and ELB scenarios). On the other hand, as team size increases, it is possible to disregard available leaders only if the members are experienced, independently of the level of support given by the leader (EHB and ELB scenarios). This result is emphasized in Fig. 5 by the significant decrease of productivity of IHA (Δ=−43.23%) in contrast to the graceful degradation of performance presented in ELB (Δ=−28.19%) and EHB (Δ=−10.47%) scenarios.
This section presented an analytical model for a single-site development team and performance indices for a set of scenarios, varying members’ expertise, the leader support and availability. It is well-known that members’ expertise plays an important role on the development team productivity [20], [21]. We observed in these indices a trade-off between the level of members’ expertise and the leader’s availability, that directly impacts the members’ productivity. Since our major interest is to study these set of interactions in GSD, Section III presents an analytical model of a development team in a multi-site context.

III. MULTI-SITE DEVELOPMENT CONTEXT

This section presents an extension of the single-site model and an abstraction for a project in a multi-site context. The idea is to instantiate multiple teams (composed of a team leader and team members) having a central authority, named central team, responsible for the overall project management (e.g., project planning, software design, quality assurance, and general support to development teams).

Fig. 6 graphically shows the development teams interaction pattern for a globally distributed project within S dispersed sites. In our model, we assume that the development work is divided into modules, which are assigned to sites (teams) in such a way that the development can be made independently with few interactions. Thus, we consider the teams self-contained, i.e., teams only work with tasks pertained under their module assignments. The interactions occur between team members and team leader, and between team leader and central team, i.e., team members do not interact directly with the central team. Summarizing, we assume that each site has a development team composed of one single member to a maximum value of N members attached to a leader, which is linked to a central entity to collaborate.

Fig. 7 presents the SAN model for the multi-site context, showing the additional states for the leader and for team members, and the interplay between them. The leader is modeled with an extra state $Ex$, which indicates that the leader is interacting with the central team, becoming unavailable to collaborate with the members. In a similar way, we modeled $Wr$ state for the members, indicating a waiting period on which they wait for the leader to resume collaboration. $Wr$ state forces a synchronization between the leader and members to effectively cooperate.

Table IV expresses the state names for the multi-site model, describing the main abstraction for the team leader and members. Note that $Wk$, $Wt$, $Co$ and $Rw$ are similar states in essence with the single-site version presented in Table I.

The local behavior of a team is similar to the one already described by the single-site case, where members are working, so they can wait for the leader, move to cooperating, then working or reworking, depending on the case. The leaders are also managing or cooperating with the members, and so on.

It is worth noticing that our abstraction is powerful enough to consider the global context as being represented by the addition of new states and events, allowing the model to simply capture new characteristics. Looking closely Fig. 7, it is notable that the events ($le$ and $es$) are fully synchronized, i.e., from cooperating, leaders move to external collaboration ($Ex$) at the same time members move from cooperating to waiting response ($Wr$).
### Table IV
**State Names for Multi-Site Model - Fig. 7**

<table>
<thead>
<tr>
<th>State</th>
<th>Abstraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mg</td>
<td>Managing: in addition to local responsibilities, the leader also collaborates with other leaders.</td>
</tr>
<tr>
<td>Co</td>
<td>Collaborating: similar to single-site development context.</td>
</tr>
<tr>
<td>Ex</td>
<td>External Collaboration: represents the cooperation between the leader and central team, discussing major project decisions and researching alternative technical/design solutions.</td>
</tr>
<tr>
<td>Wk</td>
<td>Similar to the single-site model (see Table I).</td>
</tr>
<tr>
<td>Wt</td>
<td>Waiting Response: expresses that the member is waiting for an external response from central team in order to resume working.</td>
</tr>
</tbody>
</table>

### Table V
**Description of Events from Multi-Site Model - Fig. 7**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>me</td>
<td>Similar to the single-site model (see Table II).</td>
</tr>
<tr>
<td>a</td>
<td>Leader’s Expertise: expresses the level of expertise of the leader to deal with members’ requests without any external support.</td>
</tr>
<tr>
<td>r</td>
<td>External Support: represents the central team ability (experience, time-zones and cultural diversities barriers) to solving issues demanded by leaders.</td>
</tr>
</tbody>
</table>

Table V presents the events showed in the model of Fig. 7. They correspond to the new activities performed by the leaders and members in the global context, since there is a correlation with the states where external support is present.

Using the event estimated durations for multi-site model presented in Table VI, we can configure eight possible scenarios as follows: AIL, AIH, AEL, AEH, BIL, BIH, BEL and BEH. For example, BIL represents a team in which the leader is busy to cooperate with the members, also this team has an inexperienced leader and it receives a low support from central team. These scenarios are parametrized with constant values (in average) to the team expertise, team experience, and level of support provided by the leader, which refers to the events me, r and s respectively.

Section III-B presents, based on this parametrization, the numerical solution of the SAN model [18] enabled the estimation of performance indices from those scenarios, emphasizing the team members’ productivity in a multi-site context.

#### B. Measuring Productivity

The productivity estimation follows the principles established for the single-site development context presented in Section II-B. As mentioned before, the working state Wk plays a major role in our analysis since it represents the fact that a given member is actually working (i.e., producing). Similar to the graphics shown in the previous section, we present the results for the multi-site development context showing the team leader’s responsibilities analysis on top (graphically) and the team members’ performance indices on bottom, varying from two to nine members in a site (team). Indice Δ again corresponds to the variation of these extremal cases, relating...
the increase or decrease of team productivity (i.e., working probability $W_k$) or the amount of effort wasted on waiting (probability of being in $W_t$) as teams size augments.

![Diagram](image)

Not surprisingly, Fig. 8 shows that the best case of team members’ productivity (AEH scenario) is when the leader is classified as expert, provided for a high external support, and is plenty available to cooperate with the members in the site. For $N \geq 3$ members, the leader is proportionally performing more collaboration activities ($Co$) than management ($Mg$). This behavior can be explained by the fact that high expert leaders require less external support (however, if needed, the support can be slow or very fast). The worst case considering members’ productivity (BIL scenario) shows that for $N \geq 3$ members the leader spends more time waiting for external support than performing management or cooperation activities. In large teams, for instance, when $N=9$ members, the leader has approximately the same chance of being under management and cooperation, as shown in Fig. 8 (b).

![Diagram](image)

Fig. 8. The best/worst productivity scenarios (multi-site context)

Afterwards, the team members’ productivity is analyzed by looking at the corresponding tables in Fig. 8. Despite being the best case (AEH scenario) in relation to the worst case (BIL scenario), the amount of waiting is significantly different for both cases. For the best case, indice $\Delta$ is superior to 280%, whereas, for the worst case, it is much less drastic (about 45%). Accordingly, the productivity ($W_k$) for $N=9$ members remains near 38%, however it drops to 7.28% for the same team size and the team members persist waiting ($W_t$) for most of the time (82.89%).

The diversity of scenarios enables more comprehensive studies on the effect of availability in globally distributed projects, as shown in Fig. 9. The graphics shows that the team members’ productivity observed for a scenario with expert leaders provided for a low external support is similar to a scenario with inexperienced leaders provided for a high external support. This result is observed in scenarios that have leaders with the same level of availability, e.g., AEL and AIH scenarios, and similarly to BEL and BIH scenarios. This is a remarkable result produced by our model since it analytically shows the sensitivity of leaders’ availability on the team members’ productivity. Hence, a team with an available leader most of the time is more productive than otherwise.

![Diagram](image)

Fig. 9. Impact on the productivity considering leader’s expertise and external support (multi-site context)

Fig. 10 and Fig. 11 compare team productivity in high and low external support scenarios. Our aim is to study the effect of external support in a multi-site context, since it is an important issue in globally distributed projects.
In Fig. 10, we analyze the team members’ productivity in high external support (AIL and BEH) scenarios in order to inspect some relevant decisions such as which type of leader is better: an available and inexperienced leader or a busy and expert one? In this case, our results show that it is better to have an available leader even combined with inexperience, since the leader provides high external support, compensating the lack of experience. This fact is confirmed for all team sizes (i.e., from two to nine members) in these scenarios. However, for AIL scenario, the variation $\Delta=-46.90\%$ is considerably greater than $\Delta=-28.60\%$ for BEH scenario, i.e., the decrease of team members’ productivity is emphasized much more in AIL than BEH scenario.

![Team members’ productivity analysis](image)

Fig. 10. Team members’ productivity on a high external support scenario (multi-site context)

And, is it also better to have an available and inexperienced leader than a busy and expert one in low external support scenarios? It is true for very small teams. Nevertheless, as team size increases, the team members’ productivity is better in overall having an expert leader (despite being busy). In this case, the leader’s experience improves the team members’ productivity even provided for a low external support in comparison with the productivity of a team with an available but inexperienced leader. Fig. 11 shows the team members’ productivity for these two scenarios (AIL and BEL), where the team size varies from two to nine members.

This section presented an analytical model for a multi-site development team and it also presented performance indices for a set of scenarios, varying the leader’s availability and expertise, and external support provided by a central team. As shown in the single-site results (Section II-B), we also observed in these indices the existence of a trade-off between leader’s expertise and external support provided by the central team, impacting on team members’ productivity.

![Team members’ productivity analysis](image)

Fig. 11. Team members’ productivity on a low external support scenario (multi-site context)

And, is it also better to have an available and inexperienced leader than a busy and expert one in low external support scenarios? It is true for very small teams. Nevertheless, as team size increases, the team members’ productivity is better in overall having an expert leader (despite being busy). In this case, the leader’s experience improves the team members’ productivity even provided for a low external support in comparison with the productivity of a team with an available but inexperienced leader. Fig. 11 shows the team members’ productivity for these two scenarios (AIL and BEL), where the team size varies from two to nine members.

This section presented an analytical model for a multi-site development team and it also presented performance indices for a set of scenarios, varying the leader’s availability and expertise, and external support provided by a central team. As shown in the single-site results (Section II-B), we also observed in these indices the existence of a trade-off between leader’s expertise and external support provided by the central team, impacting on team members’ productivity.

**IV. Conclusion**

In this paper we present an abstraction for development teams in single and multi-site contexts and we analytically solve a set of scenarios to investigate some performance indices. For our analysis, it is important to remark that basically three parameters play a direct influence in team members’ productivity: leader’s availability, (local/external) support and expertise. We are aware that this broad view does not capture all important dimensions of global software development projects such as specific cultural and language barriers, communication patterns considering social networks, or other difficulties in teams coordination. Also, one can model the software development process focusing on the flow of requirements engineering, development and testing as well as analytically study the project schedule evolution. However, we were particularly interested in evaluating how the communication aspects affect team members’ productivity in specific globally distributed scenarios.

Note that representing many member tasks in a single state (working state $W_k$) in the model is an abstraction that seems to be a large constraint to measure productivity. Nevertheless, this limitation may not be as bad as it may first appears to be [20], since we have been able to actually extract important team performance indices with a reasonable quality.

We stress the fact that our models are open to further modifications in future works, i.e., new states could be defined, new interactions patterns and different sets of estimated durations. A natural future work would be to improve the proposed models to capture more intricate behaviors, for instance, having team members with different skills and experiences. We could also evaluate models where teams members gain project experience and confidence to address more complex tasks. In fact, it is just an initial effort aiming to introduce analytical model techniques in the global software engineering area. Therefore, we were satisfied to consider that the team members perform the same tasks without an implicit learning process.

We are aware of the multitude of studies contemplating real project data, measurements and observations. Another idea of future work is to choose a set of geographically-dispersed projects available for reference and assign real parameters...
for our models. This extension will provide insight as how to increase productivity in global environments and how the efforts could be directed to promote faster quality project workflow. As a matter of fact, the concept of productivity could also be extended to comprise additional information and provide more qualitative estimations of teams performance.

Anyway, as said before, the main contribution of our work is to introduce the analytical modeling techniques, SAN formalism in particular, in order to provide reliable predictions in GSD endeavors. However, as any other initial approach, the use of a formal modeling to GSD projects is only starting and much research remains to be done.

ACKNOWLEDGMENT

Authors receive grants from Petrobras (0050.0048664.09.9). Paulo Fernandes is also funded by CNPq-Brazil (307272/2007-9). Afonso Sales receives grants from CAPES-Brazil (02388/09-0). The order of authors is merely alphabetical.

REFERENCES