www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(4), Sep-Dec 2025, 335-344. ISSN: 2007-1558
https://doi.org/10.61467/2007.1558.2025.v16i4.1020

A Bayesian Equilibrium in the Game of Effort Estimation in Agile Global Software Development

María Guadalupe Medina-Barrera[1], Damián Emilio Gibaja-Romero[2], Rosa María Cantón-Croda[2]
[1] Tecnológico Nacional de México IT de Apizaco, México.
[2] UPAEP, México.
guadalupe.mb@apizaco.tecnm.mx, damianemilio.gibaja@upaep.mx, rosamaria.canton@upaep.mx

Abstract. This research examines leader–team interactions in agile software development within a global context, with a focus on effort estimation. Drawing on principal–agent theory, we analyse the interaction on the assumption that the Scrum Master guides the development team under imperfect information. We model the interaction as a sequential game with incomplete information. In the first stage, the Scrum Master allocates resources to the development team; in the second stage, the team exerts effort. Both parties are characterised by types that capture their knowledge and skills. As these types are private information, we derive the Bayesian Nash equilibrium to determine the equilibrium effort levels.
Keywords: Agile Global Software Development, Bayesian Equilibria, Effort Estimation, Incomplete Information Games.

Article Info
Received February 25, 2025
Accepted July 6, 2025

1 Introduction

Effort estimation (EE) is increasingly necessary since demand for software solutions is growing worldwide across multiple social and economic sectors (Yousif et al., 2024; Brem et al., 2021). Consequently, skilled developers with knowledge of emerging and innovative technologies are needed as resources to respond to the needs of the software industry. Moreover, skilled workers have boosted the adoption of different work schemes like home office, freelancers and open-source development communities in the software industry (Lisboa de Andrade et al., 2024; Zöller et al., 2020); reason why Global Software Development (GSD) environments have become attractive for software companies to carry on projects regardless of where developers are located (Zamir et al., 2025; Lopes et al., 2023). GSD generates benefits such as cost reduction and provides a large and skilled labor pool (Yadav, 2016).

Nevertheless, EE is a complex task since GSD involves people with different objectives and processes (Grande et al., 2024; Zöller et al., 2020; Dantas et al., 2018). In other words, while agile methodologies promote close collaboration between co-located teams, GSD seeks to incorporate human resources from different organizational and geographic locations, so team members' sourcing is determined by physical and legal dimensions (Smite et al., 2014). For example, offshore insourcing means a company moves software development to a branch, while offshore outsourcing transfers software development to an external third party (Britto, 2015). Hence, GSD faces various cultural and technical barriers to establishing collaborative work (Lisboa de Andrade et al., 2024; Butt et al., 2024). So, EE in merged environments such as Agile Global Software Development (AGSD) represents a greater challenge since EE is critical for software development to cope with delivery times (Rodríguez Sánchez et al., 2023; Piñeros Rodríguez et al., 2023; Mohagheghi & Jørgensen, 2017).

It is worth emphasizing that teamwork is paramount since EE lies in the relationship of mutual dependence between the project leader and the development team to accomplish the project's objectives within agile environments. However, in AGSD, the leader and the team members may have different objectives since they may belong to different organizations with different objectives (Sandeep et al., 2022). For example, in the Scrum methodology, the Scrum master (SM) focuses on producing the highest business value. At the same time, the development team (DT) wants to maximize its profits by exerting a minimum effort since it might be composed of smaller teams located in different places. Hence, when there is no proper communication between the SM and the DT (Digital.ai, 2020; Mutiullah et al., 2018), coordination and collaboration issues may arise in AGSD to the detriment of EE (Mishra et al., 2025; Constantino et al., 2020; Stray & Moe, 2020). Thus, agile methodologies aim to overcome

the previous challenges to access the benefits of building innovative and high-quality software at a lower cost (Mishra et al., 2025; Yaseen et al., 2025; Rosado Castillo, 2024; Constantino et al., 2020; Stray & Moe, 2020; Hossain et al., 2009; Cristal et al., 2008; Sutherland et al., 2007).

In summary, the leader and team members' decision-making play a major role in EE. So, we study EE by considering that the leader and the team interact strategically in AGSD. Based on the principal-agent models (Eisenhardt, 1989), we consider that the SM (principal) guides the DT (agent) in developing a software project in the presence of imperfect information (their features are private information) when they interact in a sequential game. The SM provides resources to the DT in the first stage, while the DT sets its EE during the second stage. We analyze the subgame perfect Nash equilibrium as the solution concept to understand agents' decision-making.

Regarding our understanding, we are the first to analyze EE in an AGSD context through the lens of game theory. In this sense, this study is a novel contribution to the EE in AGSD literature. Our contributions rely on the fact that there is a need for a systematic method to deal with dependencies between the parties involved in AGSD concerning EE (Wickramaarachchi & Lai, 2017). Although EE can be done through a centralized or distributed process (Britto, 2015), such processes do not capture the strategic interactions during AGSD (Mishra et al., 2025; Yaseen et al., 2025).

The lack of information also characterizes AGSD since the SM and the DT typically have not previously worked together when the latter is outsourced. Alongside cultural and linguistic differences, it is common not to have complete information about the skills and abilities of all the agents involved in a project (Yousif et al., 2024; Noll et al., 2010). Thus, agency issues (like moral hazard and adverse selection) may emerge, leading to unrealistic EE since there is no proper information about the skills and abilities of the agents involved in the project (Eisenhardt, 1989; Yaseen et al., 2025; Rosado Castillo, 2024). The empirical evidence supports the presence of agency problems in AGSD since developers engage in decision hijacking, free-riding, and gold plating during software development (Stray & Moe, 2020; Moe, 2013; Gulliksen Stray et al., 2011). For example, Shmueli and Ronen (2017) study the tendency to develop software by applying risky practices to project schedules, quality, and costs. In this regard, Moe et al. (2010) point out strategic misbehavior, where team members pursue their objectives without discussing them with their partners; in other words, they prioritize what they consider important instead of achieving the project's objectives. In addition, outsourced developers might simultaneously work on several projects with different incentives for each one, which biases EE (Conoscenti et al., 2019).

This paper is organized into four sections. The second section describes the game-theoretical model concerning EE in an AGSD context. Then, we compute the Bayesian subgame-perfect Nash equilibrium in the third section. Finally, in the last section, we discuss the conclusions and future work.

2 The Model

We study EE by considering that an on-site scrum master and one offshore development team interact in a single sprint of an outsourced Scrum environment. So, the SM and the DT have different priorities because they belong to different organizations or places (Conoscenti et al., 2019). In this sense, our methodology is based on establishing a game-theoretical model since Game Theory is the branch of mathematics that studies conflicts between decision-makers with different objectives (Bildirici et al., 2024), as it happens in agile methodologies (Yousif et al., 2024; Wickramaarachchi & Lai, 2017). To simplify the model, we consider the DT a single decision-maker because its members self-organize, share goals, and make decisions collectively (Stray & Moe, 2020; Srivastava & Jain, 2017). Below, we describe the basic elements of our model.

The set of players is $J = \{SM,\ DT\}$, the agents involved in the sprint. Given the planned *sprint backlog* under the AGSD setting, its completion depends on the interaction between agents in J. So, *SM* and *DT* develop different activities during the sprint. First, we assume that SM is a *servant leader*; that is to say, she is always willing to support DT by removing obstacles and conflicts (Srivastava & Jain, 2017; Villegas Gómez et al., 2016). Mathematically, servant leadership means that SM provides a support vector $s = (r, c)$ where r and c are the resources and communication facilities SM wants to provide for the sprint development. Thus, the set of all SM's actions is $A_{SM} = \{s = (r, c) \in \mathbb{R}^2 | r, c \geq 0\}$.

Regarding the offshore team, DT calculates the resources it needs to build the *sprint backlog* based on its knowledge and previous experience with leaders from other GSD projects. Hidayatu et al. (2020) characterize DT's behavior by considering the resources (h_r) and communication expenses (h_c) it needs to develop the project. We denote by $h = (h_r, h_c)$ as the type vector of DT. The set of all possible types of DT is $H \subseteq \mathbb{R}^2$. Given the outsourced environment, we assume that SM does not know the DT's type; hence, h is private information (Conoscenti et al., 2019). Note that h is the realization of the random vector $H = (H_r,\ H_c)$. As it

is common in games with incomplete information (Bildirici et al. 2024), we assume h is drawn from a common knowledge probability function $z : h_r \times h_c \rightarrow [0,1]$. The type vector h influences the effort that DT wants to exert during the sprint. We consider that DT has three effort alternatives: high-effort e_H, low communication effort e_{LC}, and overall low-effort e_L. Hence, the set of all possible DT actions is $A_{DT} = \{e_H, e_{LC}, e_L\}$.

The payoffs of the players map an action vector $(s, e) \in A_{SM} \times A_{DT}$ into a real number, i.e., $u_{SM} : A_{SM} \times A_{DT} \rightarrow \mathbb{R}$ and $u_{DT} : A_{SM} \times A_{DT} \rightarrow \mathbb{R}$. Such payoff functions summarize the agents' preferences.

2.1 Payoffs construction

Payoffs depend on an action profile (s, e). Moreover, SM and DT benefit from interacting with each other; i.e., they get revenue from fulfilling the sprint objectives while incurring costs associated with their activities. So, payoffs are the difference between the revenue and the costs.

Concerning the SM, we assume that the costs of providing resources and communication are represented by a nonlinear function $C(r, c) = (rc)^2/\tau$, whereτ is the environmental stability during the sprint. The function $C(r, c)$ is aligned with empirical evidence that suggests a nonlinear relationship between r and c that increases when SM provides more resources or communication facilities. Also, the costs diminish as the stability increases, given the absence of discussions and misunderstandings between SM and DT (Ziauddin & Zia, 2012).

Finally, we consider that the SM revenues are proportional to the DT's effort since reaching the sprint's goals depends on the DT activities (Lisboa de Andrade et al., 2024). By the previous discussion, the SM benefit is shown in equation 1.

$$u_{SM}(s, e; h) = \begin{cases} (rc)e - \frac{(rc)^2}{\tau} & si\ e = e_H, \\ (rc)e - \frac{(r)^2}{\tau} & si\ e = e_{LC}, \\ (rc)e & si\ e = e_L\ . \end{cases} \quad (1)$$

The benefit of DT is associated with the effort it can exert. If DT exerts high effort, we assume it gets the same benefits as SM, plus a monetary incentive I that represents the fact that its activities contribute to fulfilling the sprint goal (Lisboa de Andrade et al., 2024). Concerning low effort, such action summarizes how DT values the SM's support during the sprint. Specifically, we assume that e_{LC} means that DT does not value communication, while e_L implies that DT does not get any value from the support vector s. Moreover, we assume that DT faces costs driven by the complexity of the sprint backlog, which is represented by γ. As usual, we consider that DT faces a quadratic cost function over e (Ziauddin & Zia, 2012); hence, the DT's payoff is given by equation 2.

$$u_{DT}(s, e; h) = \begin{cases} I + (rc)e - \gamma e^2 & si\ e = e_H, \\ I + (r)e - \gamma e^2 & si\ e = e_{LC}, \\ I + e - \gamma e^2 & si\ e = e_L\ . \end{cases} \quad (2)$$

2.2 The Game

We say that SM and DT interact in an offshore EE game. Given the features of a sprint, we model the interaction between the SM and the DT as a three-stage game.

During the first stage, nature determines the offshore DT's type h through a probability distribution z. The SM does not observe the DT's type, while DT does it at the end of this stage. In the second stage, SM sets the support vector s that she wants to provide concerning the iteration development. All players observe such a vector, and the game moves to the third stage. Finally, in the third stage, the offshore DT exerts effort by considering the sprint features and the resources provided by the SM. Then, the offshore DT establishes the effort to develop the *sprint backlog*.

After the sprint ends, agents observe the support vector provided by the SM and the effort exerted by the DT. Hence, the payoffs of each agent depend on the action profile (s, e). Given a type h, the payoffs are denoted $u_{SM}(s, e; h)$ and $u_{DT}(s, e; h)$ for the SM and DT, respectively. Figure 1 illustrates the EE game.

2.3 The Solution Concept

Subsection 2.2 describes a sequential game where DT has private information concerning its type during the third stage. Since the SM does not know the DT's type, her payoff is not deterministic. Consequently, the solution concept should consider the sequential structure and expected payoffs since SM does not know the type of DT. So, we consider a Bayesian Sub-Game Perfect Nash equilibrium as the solution for the EE game (Gavidia-Caldero et al., 2020). Before defining such an equilibrium concept, we first introduce additional notation.

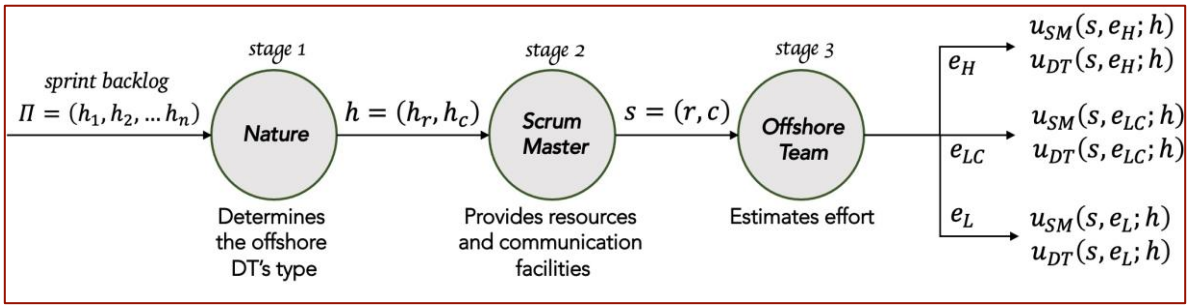

Fig. 1. The EE game in AGSD.

First, it is worth recalling that the SM has a unique type (she is a servant leader). Consequently, an action of SM is also a strategy; the reason why the set of SM's pure strategies is $S_{SM} = A_{SM}$. Regarding DT, pure strategies depend on the realization of its type; mathematically, a pure strategy ϕ for DT is a decision rule that maps (h, s) into an action in A_{DT}. So, DT's decision rule is a function $\phi : H \times \mathbb{R} \to A_{DT}$. Let S_{DT} denote the set of DT pure strategies. A profile of pure strategies is a vector (s, ϕ), and the set of all strategies' profiles of the EE game in AGSD is $S = S_{SM} \times S_{DT}$.

A Bayesian Sub-Game Perfect Nash Equilibrium is a profile of pure strategies that induces a Nash equilibrium at each stage. So, agents do not have incentives to change their strategy since the equilibrium strategies provide the largest possible payoff.

Definition 1. *A profile* $(s^*, \phi^*) \in S$ *is a Bayesian Sub-Game Perfect Nash Equilibrium, if and only if*

a. $E[u_{SM}(s^*, \phi^*)] \geq E[u_{SM}(s, \phi^*)]$ *for all* $s \in S_{SM}$*, where* $E[u_{SM}]$ *is the expected utility of SM, and*
b. $u_{DT}(s^*, \phi^*; h) \geq u_{DT}(s^*, \phi; h)$ *for all* $\phi \in S_{DT}$ *and* $h \in H$.

The solution concept states that DT assesses SM's actions and chooses the action that maximizes his payoff at the end of the third stage. Then, SM seeks to maximize her expected utility by setting the support it provides at the second stage. Also, DT observes resources provided by SM and chooses the strategy that induces the largest possible payoff for it.

3 The Bayesian Equilibria

We compute the Bayesian Sub-Game Perfect Nash Equilibria through the backward induction process (Gavidia-Calderon et al., 2020). So, we first solve the third stage and then compute the Nash equilibria of stage 2.

3.1 Second Stage equilibria

Assume that SM sets the support vector $s = (r, c)$ at the end of stage two, and DT observes it. Also, DT knows its type vector $h = (h_r, h_c)$. Hence, during the second stage, DT chooses a level of effort from A_{DT}. So, in the third stage, the backward induction process searches for the DT's strategies that maximize its payoff. Proposition 1 summarizes our findings.

Proposition 1. *There is a unique Nash equilibrium for DT at the third stage. Specifically, the DT's equilibrium strategy is given by equation 3.*

$$\phi_{DT}^{*}(s,h)=\begin{cases} e_H & si\ r \geq h_r,\ c \geq h_c, \\ e_{LC} & si\ r \geq h_r,\ c < h_c, \\ e_H & si\ r < h_r \end{cases}. \tag{3}$$

Proof. For each decision rule $\phi_{DT}(s,h) \in S_{DT}$, we have to must prove that $u_{DT}\left(\phi_{DT}^{*}(s,h)\right) \geq u_{DT}\left(\phi_{DT}(s,h)\right)$. Given the parameters of the model, strategy ϕ_{DT}^{*} establishes an effort level that depend on the relationship between the team's type and the resources providing by SM. Table 1 summarizes the payoff that the development team can get by exerting different effort levels at each case. Then, we compare the corresponding payoffs.

Table 1. DT payoff analysis.

Case condition	High effort $u_{DT}(s,e_H;h)$	Low communication effort $u_{DT}(s,e_{LC};h)$	Low effort $u_{DT}(s,e_L;h)$
$r \geq h_r,\ c \geq h_c$	$I+\frac{r^2c^2}{4\gamma}$	$I+\frac{r^2(2c-1)}{4\gamma}$	$I+\frac{2rc-1}{4\gamma}$
$r \geq h_r,\ c < h_c$	$I+\frac{r^2(2c-c^2)}{4\gamma}$	$I+\frac{r^2}{4\gamma}$	$I+\frac{2r-1}{4\gamma}$
$r < h_r$	$I+\frac{2rc-r^2c^2}{4\gamma}$	$I+\frac{2r-r^2}{4\gamma}$	$I+\frac{1}{4\gamma}$

At case 1, we observe that $u_{DT}(s,e_H;h) \geq u_{DT}(s,e_{LC};h)$ and $u_{DT}(s,e_H;h) \geq u_{DT}(s,e_L;h)$ because $u_{DT}(s,e_H;h)$ does not consider any substraction. In the second case, analogously, the development teams' payoff $u_{DT}(s,e_{LC};h)$ does not suffer any substraction, which implies that $u_{DT}(s,e_{LC};h) \geq u_{DT}(s,e_H;h)$ and $u_{DT}(s,e_{LC};h) \geq u_{DT}(s,e_L;h)$. Finally, in the third case, the previous phenomenon prevails for *DT* when she gets the payoff $u_{DT}(s,e_L;h)$; hence, we have that $u_{DT}(s,e_L;h) \geq u_{DT}(s,e_H;h)$ and $u_{DT}(s,e_L;h) \geq u_{DT}(s,e_{LC};h)$.

Then, in any case, the strategy ϕ_{DT}^{*} provides DT the largest possible payoff for the DT in comparison with other strategies. Thus, ϕ_{DT}^{*} is the equilibrium decision rule for DT during stage 2.

■

Proposition 1 points out the importance of providing support for the DT. Particularly, we observe that the SM should align with the DT's type to drive a high effort level.

3.2 First stage equilibrium

We continue with the backward induction process. So, we consider stage 2 equilibrium to compute the Nash equilibrium at stage 1, where the SM chooses the resources and communication facilities it provides. Note that ϕ_{DT}^{*} depends on DT's requirements vector $h=(h_r,h_c)$. However, SM does not observe $h=(h_r,h_c)$. In other words, SM is unsure about her benefits because the effort level depends on the relationship between the DT's type and the resources that SM provides. Hence, the SM expected utility function is written in equation 4.

$$E[u_{SM}] = u_{SM}(s,e_H)Pr[e_H] + u_{SM}(s,e_{LC})Pr[e_{LC}] + u_{SM}(s,e_L)Pr[e_L]. \tag{4}$$

Before computing the first-stage equilibrium, we need to rewrite the expected utility function more operationally. Note that the payoff of SM allows us to expand the expected utility as it is shown in expression 5.

$$E[u_{SM}] = \left((rc)e_H - \frac{(rc)^2}{\tau}\right)Pr[e_H] + \left((rc)e_{LC} - \frac{r^2}{\tau}\right)Pr[e_{LC}] + \big((rc)e_L\big)Pr[e_L] \quad \textbf{(5)}$$

Recalling that the type vector h is drawn from a common knowledge probability distribution z, the expected utility function can be rewritten by describing each probability event in terms of h and s. By applying Proposition 1, we can specify the agent's payoff concerning each probability event, which is illustrated in expression 6.

$$E[u_{SM}] = \left((rc)e_H - \frac{(rc)^2}{\tau}\right)Pr[r \geq h_r,\ c \geq h_c] + \left((rc)e_{LC} - \frac{r^2}{\tau}\right)Pr[r \geq h_r,\ c < h_c] + \big((rc)e_L\big)Pr[r < h_r] \quad \textbf{(6)}$$

We assume a multivariable uniform distribution function over the interval [0, a] to simplify the probability. Then, we get expression 7, which shows the expected utility function regarding the model's parameters. So, it represents an operational function that we analyze in Proposition 2. Specifically, we show the existence of a unique Nash equilibrium at stage 1 of the EE game.

$$E[u_{SM}] = \left((rc)\frac{rc}{2\gamma} - \frac{(rc)^2}{\tau}\right)\left(1 - \frac{r+1}{2\gamma a}\right) + \left((rc)\frac{r}{2\gamma} - \frac{r^2}{\tau}\right)\left(\frac{r}{2\gamma a}\right) + \left((rc)\left(\frac{1}{2\gamma}\right)\right)\left(\frac{1}{2\gamma a}\right) \quad \textbf{(7)}$$

Proposition 2. *The equilibrium support vector of the scrum master is unique, and it is shown in equation 8*

$$s^* = (r^*, c^*) = \left(\frac{c^2(\tau-2\gamma)(1-2\gamma a) - \sqrt{\big(c^2(2\gamma-\tau)(1-2\gamma a)\big)^2 - 6\tau c\big((\tau c-2\gamma) - c^2(\tau-2\gamma)\big)}}{3\big((\tau c-2\gamma) - c^2(\tau-2\gamma)\big)},\ \ \frac{\tau(r^2+1)}{2r(r+1-2\gamma a)(\tau-2\gamma)}\right) \quad \textbf{(8)}$$

when $\tau < 2\gamma$ *and* $2\gamma a > (r+1)$*, or when* $\tau > 2\gamma$ *and* $2\gamma a < (r+1)$.

Proof. Now, we compute the first-order condition to find the critical points of the expected utility function. Concerning r, equation 9 summarizes the first order condition.

$$\frac{\partial E[u_{SM}]}{\partial r} = \frac{rc + 3r^2(\tau c - 2\gamma) + rc^2(\tau-2\gamma)(4\gamma a - 3r - 2)}{4\gamma^2\tau a} = 0 \quad \textbf{(9)}$$

Note that equation 9 has two zeros, which are the critical points of the expected utility. Expressions 10 and 11 show the previous critical points.

$$r_1^* = \frac{c^2(\tau-2\gamma)(1-2\gamma a) - \sqrt{\big(c^2(2\gamma-\tau)(1-2\gamma a)\big)^2 - 3\tau c\big((\tau c-2\gamma) - c^2(\tau-2\gamma)\big)}}{3\big((\tau c-2\gamma) - c^2(\tau-2\gamma)\big)} \quad \textbf{(10)}$$

$$r_2^* = \frac{c^2(\tau-2\gamma)(1-2\gamma a) + \sqrt{\big(c^2(2\gamma-\tau)(1-2\gamma a)\big)^2 - 3\tau c\big((\tau c-2\gamma) - c^2(\tau-2\gamma)\big)}}{3\big((\tau c-2\gamma) - c^2(\tau-2\gamma)\big)} \quad \textbf{(11)}$$

Both r_1^* and r_2^* must satisfy that $\big(c^2(2\gamma-\tau)(1-2\gamma a)\big)^2 < 3\tau c\big((\tau c-2\gamma) - c^2(\tau-2\gamma)\big)$ and $(\tau c-2\gamma) - c^2(\tau-2\gamma) \neq 0$. Then, we compute the second derivative of the expected utility to verify if r_1^* and r_2^* maximize the SM's payoff (see expression 12). Next, we substitute r_1^* into $\frac{\partial^2 E[u_{SM}]}{\partial r^2}$, and get the equation 13. Secondly, by introducing r_2^* into $\frac{\partial^2 E[u_{SM}]}{\partial r^2}$ and simpliflying, we have equation 14.

$$\frac{\partial^2 E[u_{SM}]}{\partial r^2} = \frac{3r + \big((\tau c-2\gamma) + c^2(\tau-2\gamma)\big) + c^2(\tau-2\gamma)(2\gamma a-1)}{2\gamma^2\tau a} \quad \textbf{(12)}$$

$$\frac{\partial^2 E[u_{SM}]}{\partial r^2}(r_1^*) = -\frac{\sqrt{\left(c^2(2\gamma-\tau)(1-2\gamma a)\right)^2 - 3\tau c\left((\tau c-2\gamma)-c^2(\tau-2\gamma)\right)}}{2\gamma^2\tau a} \tag{13}$$

$$\frac{\partial^2 E[u_{SM}]}{\partial r^2}(r_2^*) = \frac{\sqrt{\left(c^2(2\gamma-\tau)(1-2\gamma a)\right)^2 - 3\tau c\left((\tau c-2\gamma)-c^2(\tau-2\gamma)\right)}}{2\gamma^2\tau a} \tag{14}$$

By expressions 13 and 14, we have that $\frac{\partial^2 E[u_{SM}]}{\partial r^2}(r_1^*) < 0$ and $\frac{\partial^2 E[u_{SM}]}{\partial r^2}(r_2^*) > 0$, respectively. So, r_1^* maximizes the SM's payoff, and r_2^* minimizes the SM's payoff. Therefore, r_1^* is the unique optimal strategy for SM concerning the SD resources provided to DT.

Next, we look for the communication investment c at equilibrium. As before, we follow the criterion of the first derivative, which implies to solve equation 15. The equation $\frac{\partial E[u_{SM}]}{\partial c} = 0$ has a unique critical point, that we show in equation 16.

$$\frac{\partial E[u_{SM}]}{\partial c} = \frac{r^2 c(r+1)}{\gamma\tau a} + \frac{-2r^2 c(r+1)+r+r^3}{4\gamma^2 a} + \frac{r^2 c(\tau-2\gamma)}{\gamma\tau} \tag{15}$$

$$c_1^* = \frac{\tau(r^2+1)}{2r(r+1-2\gamma a)(\tau-2\gamma)} \tag{16}$$

only if $\tau > 0$ and $r > 0, \tau - 2\gamma \neq 0, r + 1 - 2\gamma a \neq 0$.

Now, we compute the second derivative to determine the nature of c_1^*. Expression 17 shows the evaluation of the critical point in the second derivative.

$$\frac{\partial^2 E[u_{SM}]}{\partial c^2} = \frac{r^2(\tau-2\gamma)\left(2\gamma a-(r+1)\right)}{2\gamma^2\tau a} \tag{17}$$

Since parameters γ, τ, and a are positive, we conclude that $\frac{\partial^2 E[u_{SM}]}{\partial c^2}(c_1^*) < 0$ when $\tau < 2\gamma$ and $2\gamma a > (r+1)$, or when $\tau > 2\gamma$ and $2\gamma a < (r+1)$. So, c_1^* maximizes the SM's payoff when such conditions are met. Therefore, c_1^* is the optimal SM strategy concerning communication expenses.

■

Theorem 1 summarizes Propositions 1 and 2. So, a unique Bayesian Sub-Game Nash equilibrium exists in the EE game when an AGSD environment is considered.

Theorem 1. *The EE game in AGSD has a unique Bayesian Sub-Game Perfect Nash Equilibrium* (s^*, ϕ^*)*; expressions 18 and 19 show the equilibrium strategies for each agent.*

$$\phi_{DT}^*(s,h) = \begin{cases} e_H & si\ r \geq h_r,\ c \geq h_c, \\ e_{LC} & si\ r \geq h_r,\ c < h_c, \\ e_L & si\ r < h_r \quad , \end{cases} \tag{18}$$

$$s^* = (r^*, c^*) = \left(\frac{c^2(\tau-2\gamma)(1-2\gamma a) - \sqrt{\left(c^2(2\gamma-\tau)(1-2\gamma a)\right)^2 - 6\tau c\left((\tau c-2\gamma)-c^2(\tau-2\gamma)\right)}}{3\left((\tau c-2\gamma)-c^2(\tau-2\gamma)\right)},\ \frac{\tau(r^2+1)}{2r(r+1-2\gamma a)(\tau-2\gamma)} \right), \tag{19}$$

when $\tau < 2\gamma$ *and* $2\gamma a > (r+1)$*, or when* $\tau > 2\gamma$ *and* $2\gamma a < (r+1)$.

It is important to note that strategy ϕ_{DT}^* shows how reactive the development team is concerning the support provided by the scrum master. In other words, the equilibrium strategy says that resources and communication are crucial to encourage DT to exert high-

level effort. If resources are not enough, the DT immediately chooses a low-level effort. Since the DT's type represents how it values the support infrastructure, the SM should guarantee an appealing environment to reach the objectives of the sprint. In this sense, the SM equilibrium strategies also summarize how she should incorporate complexity (γ) and stability (τ) into providing resources and communication. Figure 2 shows a simulation of SM decision-making when the previous parameters change. The images show how communication changes with respect to stability when complexity is fixed. We observe that communication increases as resources increase in an environment of high stability and low complexity, suggesting a strong servant leadership (Holtzhausen & de Klerk, 2018). In contrast, a high level of complexity tends to diminish communication as resources increase, even with high levels of stability. Such a scenario represents the challenges of effectively using resources (associated with the optimal point where communication is maximized in Figures 2.b and 2.c) (Srivastava & Jain, 2017).

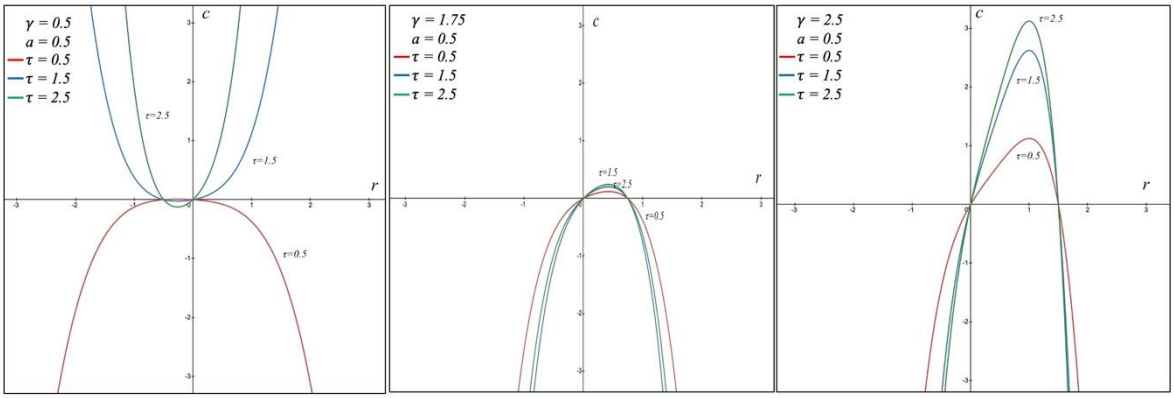

Fig. 2. SM equilibrium strategies as τ change with different levels of γ, low (a), medium (b) and high (c).

4 Conclusions

In this research, we study the impact of leader-team interaction on effort estimation and support provision during Agile Global Software Development (AGSD). We model such interaction using a sequential game-theoretical approach to analyze players' strategies in the presence of incomplete information. As the scrum master has limited information about the team's skills at an offshore site, principal-agent issues arise during the EE process. Findings are important for those companies that move software development to a branch or an external third party; that is, companies developing software under offshoring or outsourcing arrangements.

Based on the agile SD approach, the scrum master is responsible for supporting the team by removing obstacles. We investigate the magnitude of support provided and the effort estimations at equilibrium. The support basket in AGSD includes SD resources and investment in communication to overcome problems due to the distance between sites collaborating on a project. Therefore, we analyze players' strategies (support and effort) at equilibrium during a single sprint where the scrum master is on site, and the development team is located offshore. We model such interaction as a strategic model, the EE game in AGSD, where the scrum master has incomplete information about the team's abilities. Here, nature plays first in determining the type of team; thus, players make decisions strategically to maximize their profits. Then, the scrum master supports her expectation of benefit since she does not know the team's type. Lastly, the team observes the support and chooses the effort level for the sprint. We find the solution by computing the Bayesian Sub-Game Perfect Nash Equilibrium and the players' payoff functions described in section 2.1 for the EE game model.

The team's optimal strategy is unique at the game's third stage. Then, a high effort level comes when both SD and communication resources are enough to deal with the *sprint backlog* complexity based on the team's experience. On the contrary, effort decreases in communication or development, depending on which resources the team considers useless. At the game's second stage, the scrum master's optimal strategy is unique and consists of a support vector providing SD and communication resources. However, the scrum master's strategies are highly dependent on the availability of a database with reliable effort records. Theorem 1 shows players' strategies at equilibrium in the EE game in AGSD. In future works, we intend to extend this theoretical model to a dynamic framework that allows us to analyze EE and support provision over several sprints by updating iteration velocity in AGSD.

References

Bildirici, F., Codal, K. S., & Medeni, T. D. (2024). Using Agile Story Points and Game Theory Together: Better Software Planning and Development in Agile Software Development. *arXiv preprint arXiv:2409.12196*.

Brem, A., Viardot, E., & Nylund, P. A. (2021). Implications of the coronavirus (COVID-19) outbreak for innovation: Which technologies will improve our lives?. *Technological Forecasting & Social Change*, 163,1–7. https://doi.org/10.1016/j.techfore.2020.120451

Britto, R. (2015). Knowledge Classification for Supporting Effort Estimation in Global Software Engineering Projects (Licentiate dissertation). Blekinge Institute of Technology, Sweden.

Britto, R., Mendes, E., & Börstler, J. (2015). An empirical investigation on effort estimation in agile global software development. In *IEEE 10th International Conference on Global Software Engineering* (pp. 38–45). IEEE Press.

Butt, S.A., Naz, S., Piñeres-Espitia, G., Ariza-Colpas, P., & Piñeres-Melo, M.A. (2024). The Importance of Robust Communication in Large-Scale Agile Development. *Procedia Computer Science*, 236, 224-232. https://doi.org/10.1016/j.procs.2024.05.025

Constantino, K., Zhou, S., Souza, M., Figueiredo, E., & Kästner, C. (2020). Understanding Collaborative Software Development: An Interview Study. In *ACM/IEEE 15th International Conference on Global Software Engineering (ICGSE)*, Seoul, Republic of Korea, (pp. 55–65). https://doi.org/10.1145/3372787.3390442

Cristal, M., Wildt, D., & Prikladnicki, R. (2008). Usage of scrum practices within a global company. In *IEEE International Conference on Global Software Engineering* (pp. 222-226). IEEE Press.

Conoscenti, M., Besner, V., Vetro, A., & Méndez Fernández, D. (2019). Combining data analytics and developers feedback for identifying reasons of inaccurate estimations in agile software development. *Journal of Systems and Software*, 156, 126–135. https://doi.org/10.1016/j.jss.2019.06.075

Dantas, E., Perkusich, M., Dilorenzo, E., Santos, D.F.S., Almeida, H., & Perkusich, A. (2018). Effort Estimation in Agile Software Development: An Updated Review. *International Journal of Software Engineering and Knowledge Engineering*, 28(11n12), 1811-1831. https://doi.org/10.1142/S0218194018400302

Digital.ai. (2020). 14th annual State of Agile Report. Retrieved June 22, 2020, from https://digital.ai/catalyst-blog/the-14th-annual-state-of-agile-report/

Eisenhardt, K. M. (1989). Agency theory: An assessment and review. *Academy of Management Review*, 14, 57–74.

Gavidia-Calderon, C., Sarro, F., Harman, M., & Barr, E. T. (2020). Game-theoretic analysis of development practices: Challenges and opportunities. *Journal of Systems and Software*, *159*, 110424. https://doi.org/10.1016/j.jss.2019.110424

Grande, R., Vizcaíno, A., & García, F.O. (2024). Is it worth adopting DevOps practices in Global Software Engineering? Possible challenges and benefits. *Computer Standards & Interfaces*, 87, 103767. https://doi.org/10.1016/j.csi.2023.103767

Gulliksen Stray, V., Moe, N. B., & Dingsøyr, T. (2011). Challenges to teamwork: A multiple case study of two agile teams. In A. Sillitti, O. Hazzan, E. Bache & X. Albaladejo (Eds.), *Agile Processes in Software Engineering and Extreme Programming XP 2011*, (Lecture Notes in Business Information Processing, Vol 77, pp. 146–161). Springer Berlin Heidelberg.

Hidayati, A., Budiardjo, E.K., & Purwandari, B. (2020). Hard and Soft Skills for Scrum Global Software Development Teams. In *ICSIM'20 Proceedings of the 3rd International Conference on Software Engineering and Information Management*, (pp. 110-114). https://doi.org/10.1145/3378936.3378966

Holtzhausen, N. & de Klerk, J. J. (2018). Servant leadership and the scrum team's effectiveness. *Leadership & Organization Development Journal*, 39(7), 873–882. https://doi.org/10.1108/LODJ-05-2018-0193

Hossain, E., Babar, M. A., & Paik, H. (2009). Using Scrum in global software development: A systematic literature review. In *Fourth IEEE International Conference on Global Software Engineering* (pp. 175-184). IEEE Press.

Keil, P. (2005). Principal agent theory and its application to analyze outsourcing of software development. *ACM SIGSOFT Software Engineering Notes*, 30(4), 1–5.

Lisboa de Andrade, A. S., Jackson, V., Prikladnicki, R., & van der Hoek, A. (2024). On meetings involving remote software teams: A systematic literature review. *Information and Software Technology*, 175, 107541. https://doi.org/10.1016/j.infsof.2024.107541

Lopes, T., Ströele, V., Braga, R., David, J.M.N., & Bauer, M. (2023). A broad approach to expert detection using syntactic and semantic social network analysis in the context of Global Software Development. *Journal of Computational Science*, 66, 101928. https://doi.org/10.1016/j.jocs.2022.101928

Mishra, D. (2025). Developing a knowledge-based perspective of coordination in global software development, *VINE Journal of Information and Knowledge Management Systems*, 55(2), 287–309. https://doi.org/10.1108/VJIKMS-08-2022-0270

Moe, N. B., Dingsøyr, T., & Dyba, T. (2010). A teamwork model for understanding an agile team: A case study of a scrum project. *Information and Software Technology*, 52, 480–491. https://doi.org/10.1016/j.infsof.2009.11.004

Moe, N. B. (2013). Key challenges of improving agile teamwork. In H. Baumeister & B. Weber (Eds.), *Agile Processes in Software Engineering and Extreme Programming XP 2013* (Lecture Notes in Business Information Processing, Vol 149, pp. 76–90). Springer Berlin Heidelberg.

Mohagheghi, P. & Jørgensen, M. (2017). What contributes to the success of IT projects? An empirical study of IT projects in the Norwegian public sector. *Journal of Software*, 12(9), 751–758. https://doi.org/10.17706/jws.12.9.751.758

Mutiullah, J., Ayesha, W., Muhammed, S., Sherjeel, I., Muhammed, A., & Adnan, K. (2018). A review of popular agile software development technologies. *Journal of Information Technology & Software Engineer*ing, 8(4). https://doi.org/10.4172/2165-7866.1000245

Noll, J., Beecham, S., & Richardson, I. (2010). Global software development and collaboration: Barriers and solutions. *ACM Inroads*, 1(3), 66–78. https://doi.org/10.1145/1835428.1835445

Piñeros Rodríguez, C.A., Sierra Martínez, L.M., Peluffo Ordóñez, D.H., & Timana Peña, J.A. (2023). Effort Estimation in Agile Software Development: A Systematic Map Study. *INGE CUC*, 19(1), 22-36. https://doi.org/10.17981/ingecuc.19.1.2023.03

Rodríguez Sánchez, E., Vázquez Santacruz, E.F., & Cervantes Maceda, H. (2023). Effort and Cost Estimation Using Decision Tree Techniques and Story Points in Agile Software Development. *Mathematics*, *11*, 1477. https://doi.org/10.3390/math11061477

Rosado Castillo, N.I., & Aguileta Güemez, A.A., & Aguilar Vera, R.A. (2024). Team Composition in Software Engineering Development Phases: A Systematic Literature Review. *International Journal of Combinatorial Optimization Problems and Informatics*, 15(5), 237-253. https://doi.org/10.61467/2007.1558.2024.v15i5.584

Sandeep, R.C., Sánchez-Gordon, M., Colomo-Palacios, R., & Kristiansen, M. (2022). Effort Estimation in Agile Software Development: An Exploratory Study of Practitioners' Perspective. In P. Adam, J. Aleksander, L. Ivan & Y.N. Yen (Eds.), *Lean and Agile Software Development*, (LNBIP, Vol. 438, pp. 136-149), Springer Cham. https://doi.org/10.1007/978-3-030-94238-0

Shmueli, O., & Ronen, B. (2017). Excessive software development: Practices and penalties. *International Journal of Project Management*, 35(1), 13-27. https://doi.org/10.1016/j.ijproman.2016.10.002

Smite, D., Wohlin, C., Galvin ̧a, Z., & Prikladnicki, R. (2014). An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering*, 19, 105–153. https://doi.org/10.1007/s10664-012-9217-9

Srivastava, P. & Jain, S. (2017). A leadership framework for distributed self-organized scrum teams. *Team Performance Management*, 23, 293–314. https://doi.org/10.1108/TPM-06-2016-0033

Stray, V., & Moe, N.B. (2020). Understanding coordination in global software development: A mixed-methods study on the use of meetings and Slack. *The Journal of Systems & Software*, 170, 110717. https://doi.org/10.1016/j.jss.2020.110717

Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). Distributed Scrum: Agile project management with outsourced development teams. In *40th Hawaii International Conference on Software Systems (HICSS'07)* (pp. 274a-274a). IEEE Press.

Villegas Gómez, E., Ruiz Rodríguez, J. M., & López Gallego, F. (2016). El conflicto en el desarrollo ágil: una perspectiva desde el scrum. *Revista Gestión y Región*, 21,121–149.

Wickramaarachchi, D. & Lai, R. (2017). Effort estimation in global software development - a systematic review. *Computer Science and Information Systems*, 14(2), 393–421. https://doi.org/10.2298/CSIS160229007W

Yadav, V. (2016). A flexible management approach for globally distributed software projects. *Global Journal of Flexible Systems Management*, 17(1), 29–40. https://doi.org/10.1007/s40171-015-0118-9

Yaseen, M., Alroobaea, R., & Alsufyani, H. (2025). Structural association of requirements engineering challenges in GSD: interpretive structural modelling (ISM) approach. *Requirements Engineering*. https://doi.org/10.1007/s00766-025-00435-8

Yousif, J.H., & Al-Kindi, G., & Srivastava, D.K. (2024). Global Software Development for Smart Cities. In H. Susheela, K. Vidhu, G. Rupali, D. Srivastava & H.Y. Jabar (Eds.), *5G Enabled Technology for Smart City and Urbanization System*, Chapman and Hall/CRC: New York. https://doi.org/10.1201/9781003467892-13

Zamir, S., Rehman, A., Moshin, H., Zamir, E., Abbas, A., & Al-Yarimi, F.A.M. (2025). Integrating Pull Request Comment Analysis and Developer Profiles for Expertise-Based Recommendations in Global Software Development. *IEEE Access*, 13, 16637–16648. https://doi.org/10.1109/ACCESS.2025.3532386

Ziauddin, S. K. T. & Zia, S. (2012). An effort estimation model for agile software development. *Advances in Computer Science and its Applications*, 2(1), 314–324.

Zöller, N., Morgan, J.H., & Schröder, T. (2020). A topology of groups: What GitHub can tell us about online collaboration. *Technological Forecasting & Social Change*, 161, 120291. https://doi.org/10.1016/j.techfore.2020.120291