Coordination of verification activities with incentives: a two-firm model

Aditya U. Kulkarni, Christian Wernz & Alejandro Salado

Research in Engineering Design

ISSN 0934-9839

Res Eng Design
DOI 10.1007/s00163-020-00352-7

Springer

Research in Engineering Design
https://doi.org/10.1007/s00163-020-00352-7

ORIGINAL PAPER

Coordination of verification activities with incentives: a two-firm model

Aditya U. Kulkarni[1] · **Christian Wernz**[2] · **Alejandro Salado**[1]

Received: 25 April 2019 / Revised: 7 October 2020 / Accepted: 26 October 2020

Abstract
In systems engineering, verification activities evaluate the extent to which a system under development satisfies its requirements. In large systems engineering projects, multiple firms are involved in the system development, and hence verification activities must be coordinated. Self-interest impedes the implementation of verification strategies that are beneficial for all firms while encouraging each firm to choose a verification strategy beneficial to itself. Incentives for verification activities can motivate a single firm to adopt verification strategies beneficial to all firms in the project, but these incentives must be offered judiciously to minimize unnecessary expenditures and prevent the abuse of goodwill. In this paper, we use game theory to model a contractor-subcontractor scenario, in which the subcontractor provides a component to the contractor, who further integrates it into their system. Our model uses belief distributions to capture each firm's epistemic uncertainty in their component's state prior to verification, and we use multiscale decision theory to model interdependencies between the contractor and subcontractor's design. We propose an incentive mechanism that aligns the verification strategies of the two firms and using our game-theoretic model, we identify those scenarios where the contractor benefits from incentivizing the subcontractor's verification activities.

Keywords Systems engineering · Verification · Testing · Incentives · Multiscale decision theory

1 Introduction

Verification activities aid in managing project risk while improving confidence in a system meeting its requirements (Salado 2015). They are critical to system development because they shape the uncertainty associated with the functioning or performance of the system that is being developed (Walden et al. 2015; Tahera et al. 2019). Most companies have not adopted a structured approach to verification (Shabi et al. 2017). As a result, verification activities consume a significant amount of resources during systems' lifecycle throughout the industry (Tahera et al. 2017). Hence, the importance of discovering the scientific foundations of verification activities in systems engineering has been recognized by multiple authors, and research on developing robust decision-making frameworks for verification activities is an active research area (Engel and Barad 2003; Shabi and Reich 2012; Salado and Kannan 2019; Xu and Salado 2019).

Planning and executing verification activities becomes more difficult as the number of firms participating in system design increases (Nagano 2008). An important impediment in multi-firm or multi-team systems engineering projects is the misalignment of individual interests, which prevents the implementation of Pareto-optimal solutions (Collopy 2012). The research community has looked to game theory (Lewis and Mistree 1997; Xiao et al. 2005; Ciucci et al. 2012) and incentive theory (Vermillion and Malak 2018a, b) to better understand how conflicting interests affect project outcomes in systems engineering.

While game theory (Osborne and Rubinstein 1994) is concerned with equilibrium behavior of cooperative or non-cooperative actors in a given scenario, incentive theory (Laffont and Martimort 2009), a subset of game theory, focuses on how supervisors can use incentives to influence the behavior of subordinates in an implicit or explicit organizational hierarchy. This makes incentive theory a well-suited modeling approach for predicting the equilibrium behavior

✉ Alejandro Salado
asalado@vt.edu

[1] Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA, USA

[2] Department of Health Administration, Virginia Commonwealth University, Richmond, VA, USA

Published online: 12 November 2020

of agents under different incentive schemes in systems engineering projects. Though multiple authors have used incentive theory to understand how incentives can improve product quality through verification activities in supply chains (Baiman et al. 2000, Balachandran and Radhakrishnan 2005; Zhu et al. 2007), to the best of our knowledge, the application of incentives to improve coordination of verification strategies in systems engineering projects has not been explored yet.

To understand how incentive mechanisms can improve coordination of verification strategies in multi-firm systems engineering projects, in this paper, we use game theory to model a contractor-subcontractor scenario, where the contractor outsources the development of a component to the subcontractor. We focus on those scenarios where the subcontractor is confident of its component design meeting requirements and does not want to conduct any more verification activities, whereas the contractor prefers that the component design undergo further verification. For these scenarios, we propose an incentive mechanism that ensures the subcontractor is sufficiently motivated by incentives to verify its design. Using our model, we then determine when it is profitable for the contractor to incentivize the subcontractor's verification activities.

Unlike traditional manufacturing environments where quality assurance engineers optimize verification strategies by understanding the aleatory (i.e., random) uncertainty of the production process, verification strategies in systems engineering rely on an understanding of the epistemic uncertainty (i.e., lack of knowledge) of the system design meeting its requirements (Sentz and Ferson 2002). Due to epistemic uncertainty, firms can merely maintain beliefs over the state of their design and never achieve certainty. However, the uncertainty—both aleatoric and epistemic, can be reduced, and thereby the belief distributions can be narrowed (Salado and Kannan 2018). To model the epistemic uncertainty faced by engineers in system design, in our model, we assume that each firm maintains a belief in its design meeting requirements.

An important distinction between the beliefs of the two firms in our model is that the subcontractor's belief is defined for the component design, whereas the contractor's belief is defined for the overall system design. Hence the contractor's belief is a function of the subcontractor's belief. This interdependence gives rise to two challenges: (1) how to address the possibility of the subcontractor misrepresenting its beliefs for monetary gain, and (2) how to model the interdependence between the beliefs of the firms. To address the possibility of the subcontractor misrepresenting its beliefs, we develop a reward/penalty mechanism, specific to our model, which ensures the subcontractor cannot gain by misrepresenting its beliefs to the contractor. To model the interdependence between the contractor's system and the subcontractor's component, we use multiscale decision theory (MSDT). MSDT uses influence functions to model the dependence between a subordinate's output, in this case the subcontractor's component design, and its supervisor's output, in this case the contractor's system design (Kulkarni and Wernz 2020). Another benefit of MSDT is its scalability and the two-firm model developed in this paper can be readily extended in future research, as previously demonstrated (Wernz and Deshmukh 2010).

The remainder of this paper is organized as follows. Section 2 provides an overview of the literature. In Sect. 3, we develop the single-firm model, where only the contractor works on the system design. Here, we define a single firm's verification strategy as a function of its verification costs and its belief in the system design meeting requirements. In Sect. 4, we extend the single-firm model to the two-firm model, where the contractor delegates the design of a component to a subcontractor but offers no additional incentives for verification. We use the two-firm model to determine the two-firm verification strategy as a function of each firm's belief in its design meeting requirements. In Sect. 5, we determine how incentivizing the subcontractor's verification activities can benefit the contractor. In this section, we develop an incentive mechanism whereby the contractor can adequately incentivize the subcontractor's verification activities while ensuring the subcontractor does not gain by reporting a false belief value to the contractor. In addition, we study how the variation in the two firm's model parameter values affects the verification strategy and incentive space for the two firms. Finally, we conclude by summarizing all the insights in Sect. 6.

2 Literature review

Literature on incentivizing verification in systems design is sparse and often narrows in on specific topics, such as testing of design alternatives (Schumacher and Schlapp 2017), unforeseeable changes in product design (Sommer and Loch 2009), information sharing (Schlapp et al. 2015), or ensuring cost and time compliance (Mihm 2010). The majority of the literature on verification only considers single-firm verification activities. A multi-firm model for incentivizing verification activities that is scalable has not yet been developed. Using MSDT, we can achieve both, a general-purpose two-firm model, which has the potential to be scaled up to capture multi-firm interactions (Wernz and Deshmukh 2012).

A number of modeling approaches have been developed to determine optimal verification strategies for a single-firm. Ahmadi and Wang (1999) formulated a nonlinear program that minimizes verification costs for the desired level of system verification. Boumen et al. (2008, 2009) used dynamic programming to determine risk-minimizing verification

strategy for lithographic machines. To allocate resources for verification in an efficient manner, Shabi and Reich (2012) developed an analytical model to determine the optimal verification strategy given the design maturity under cost and risk constraints. Using Monte Carlos simulation, Engel and Barad (2003) determined the probability distribution of the residual risk of a cost-minimizing verification strategy. In a follow-up paper, Barad and Engel (2006) extend their work by analyzing their prior results for different objective functions. To account to uncertain data inputs, Engel and Last (2007) applied fuzzy logic and compared it to the probabilistic approach in the aforementioned models.

Though the number of works on single-firm verification is large, prior works on single firm models of verification have relied heavily on aleatory interpretations of probability. However, recent works have acknowledged that uncertainty faced by designers is epistemic in nature (Dai et al. 2003; Huang and Zhang 2009), and engineers make design decisions using their subjective beliefs on the current state of the system design (Eifler et al. 2010; Wynn et al. 2011). Unlike aleatory uncertainty, epistemic uncertainty arises due to a fundamental lack of knowledge about the system, or process, under study (Sentz and Ferson 2002; Schlosser and Paredis 2007). In this regard, belief distributions are more appropriate in representing the uncertainty faced by designers. To the best of our knowledge, only recent works by Salado et. al (Salado et al. 2018; Xu and Salado 2019; Kulkarni et al. 2020) have explicitly modeled a firm's belief over the possible states of its design. In this paper, we continue this line of work by using belief distributions to model a firm's belief in the state of its design. A firm's belief distribution is then used as a basis to determine its optimal verification strategy.

Though the literature on incentives for verification in systems engineering is sparse, similar problems have been explored, in detail, in the quality control literature for supply chains (Emons 1988; Reyniers 1992; Reyniers and Tapiero 1995; Baiman et al. 2000; Balachandran and Radhakrishnan 2005). In this literature, the focus is on minimizing the adverse effects of information asymmetry between the supplier and the buyer. Here, the buyer cannot observe the extent to which the supplier has verified its products, but the buyer is assumed to have the capability to test the supplier's product for defects. The buyer can then choose the level of resources it will spend on testing, or sampling, the supplier's goods for defects, or choose to incentivize the supplier's quality control. Furthermore, in supply chain contracts, verification activities are often not contracted upon, and instead, the contracts only specify product quality level the supplier must meet (Starbird 2001).

Verification in systems engineering projects, such as satellite design, is fundamentally different from verification in supply chains since systems engineering projects often involve complex and costly designs that require the participation of engineers from multiple disciplines. Specific verification activities are often specified by contracts. Furthermore, unlike supply chains, in systems engineering projects, the contractor may not have the ability to directly verify the subcontractor's design and may only discover an erroneous component design when the entire system design is verified (e.g., discovering errors in embedded systems through hardware-in-loop simulations). This motivates our model scenario, where the contractor can only discover an error in the subcontractor's component by verifying the entire system design, and must thus determine if incentivizing the subcontractor would be more beneficial.

3 The single-firm model

We first consider the scenario where only the contractor is engaged in the system design. The system design process often consists of multiple development phases, where a development phase is defined to consist of design and production activities, such as system architecture, tradespace exploration, or manufacturing, followed by verification activities, such as testing, demonstration, analysis, or inspection (Salado 2018). Design and production activities are carried out with the intention of creating a design that meets its requirements. However, for various reasons, the design at the end of a design phase may not meet all the requirements that were set for it at the start of the development phase. Verification activities are thus executed to confirm or deny if the design at the end of a design phase meets all the requirements that were set for it at the start of the development phase. Furthermore, verification activities are often chosen based on the requirement to be verified along with budget and time constraints.

For the single-firm model, we restrict our attention to a single development phase and a single requirement, which we refer to as the requirement of interest. We assume that based on the requirement of interest, budget and time constraints, the contractor has already determined the appropriate verification activities. The decision problem faced by the contractor is whether to carry out verification activities in the current development phase, or postpone verification of the requirement of interest to the next development phase. In our model, we assume that that contractor's decision to verify or not is based on its belief in the current design satisfying the requirement of interest and two high-level costs associated with the verification activities: setup cost to execute the verification activity and expected cost to repair an erroneous design. These conditions are similar to those in prior works (Engel and Barad 2003; Shabi and Reich 2012). The main difference in our model is that we explicitly model a firm's confidence in the state of its design using belief distributions.

We adopt three additional assumptions about the verification activity to build an analytically tractable model. First, the verification activity has a fixed setup cost; second, the verification activity will certainly reveal whether or not the system design meets the requirement of interest; and third, the contractor knows the expected cost of repairing the system design if it is found not to meet the requirement of interest. In reality, the setup costs of verification activities may vary based on how thorough the contractor wants to be in detecting errors in design, verification activities do not always reveal an error in system design, and the contractor does not know the expected cost of repairing the system design before the error is identified. In addition to analytical tractability, the assumptions above greatly simplify the insights on the tensions between two firms with respect to design verification.

The single-firm model scenario can be illustrated with the following example. The system is a prosthetic robotic arm, and the requirement of interest defines a strict weight limit for the robotic arm. The verification activity involves the contractor measuring the weight of all components in the robotic arm. From the previous development phase, the contractor knows the weight of the robotic arm with a certain precision. In the current development phase, the contractor chooses a new design for the pneumatic system in the robotic arm. The contractor computes the approximate weight of the new robotic arm design using its knowledge from the previous development phase and the weight specifications of the new pneumatic system. However, the new pneumatic system has resulted in minor alterations in other components, and these alterations may have caused the new robotic arm design to violate the weight requirement. The contractor now uses its confidence in the new design meeting the weight requirement to decide between verifying the new design now or postponing the verification to the next development phase. If the contractor chooses to verify, then the contractor faces a fixed verification cost in terms of time and effort required to dismantle the arm and measure its components, and it is reasonable to assume in this scenario that the contractor will know the expected repair cost of altering the new design to make it meet the weight requirement.

3.1 Model parameters under complete information

Using MSDT terminology, we refer to the contractor as SUP. We divide SUP's development phase into two time periods, the design period, where SUP executes design activities, and the verification period, where SUP may or may not execute the predefined verification activity. We will use subscript I for all variables associated with the start of SUP's development phase. Similarly, we will use subscript D for all variables associated SUP's decision point, where SUP decides to either verify or not verify the system design, and subscript E will be used for all variables associated with the end of SUP's development phase. In this paper, we restrict our attention to those scenarios where the state of SUP's design can be broadly categorized as either satisfactory (meets the requirement of interest) or unsatisfactory (does not meet the requirement of interest). The state of SUP's design is denoted by $S_t^{SUP} \in \{0, 1\}$, where $t \in \{I, D, E\}$. Here, $S_t^{SUP} = 1$ implies that SUP's system design is in a satisfactory state at point t and $S_t^{SUP} = 0$ implies that SUP's system design is in an unsatisfactory state at point t.

Though SUP may not intend to violate the requirement of interest through its choices directly, SUP's design choices to meet other system requirements may cause the current design to violate the requirement of interest. For example, the robotic arm could have a durability requirement for the arm material. To meet this requirement, SUP uses a metal alloy that increases the weight of the arm beyond its allowable limit. In general, the development of new systems has uncertainty associated with meeting requirements. To model this, we assume that there is a positive probability of SUP violating the requirement of interest in the current development phase. We denote this probability as ε^{SUP}, and we refer to ε^{SUP} as the probability of SUP making a design error, where a design error implies the design does not meet its requirements. In our model, ε^{SUP} is a measure of SUP's design skills: lower the value of ε^{SUP}, the more skilled SUP is in designing the system. It follows that if SUP's design is in the satisfactory state at the start of the development phase, then with probability ε^{SUP} it will be in the unsatisfactory state at the end of the development phase. Furthermore, we assume that if SUP's design is in the unsatisfactory state at the start of the development phase, it will certainly be in the unsatisfactory state at the end of the development phase.

We denote SUP's decision to verify the design in the current development phase by v^{SUP}, and its decision to postpone the design verification to the next development phase by $-v^{SUP}$. As per our model assumptions, we consider two high-level verification costs: set up cost and expected repair cost. Irrespective of the state of the design, SUP incurs a fixed setup cost of c^{SUP} when it chooses to verify its design. If SUP chooses to verify the design, then SUP will incur an expected repair cost of r^{SUP}. Another cost parameter we consider in our model is the cost of postponing verification to the next development phase. If SUP chooses to postpone verification to the next development phase, and the current design is in an unsatisfactory state, then SUP will incur repair costs in the next development phase. To capture the possible costs associated with delaying verification to the next development phase, we assign a cost l^{SUP} to $S_E^{SUP} = 0$. Figure 1 graphically depicts the single-firm model, where SUP knows the true state of its design at all times.

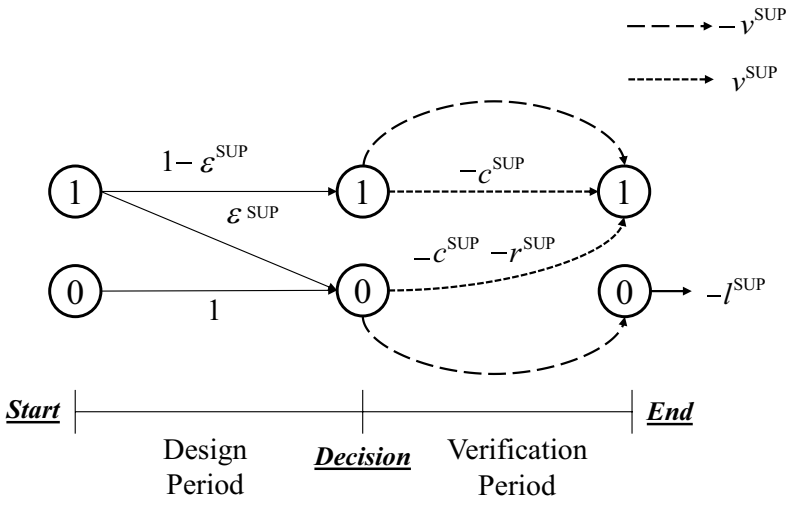

Fig. 1 Single-firm model where SUP knows the true design state at all times

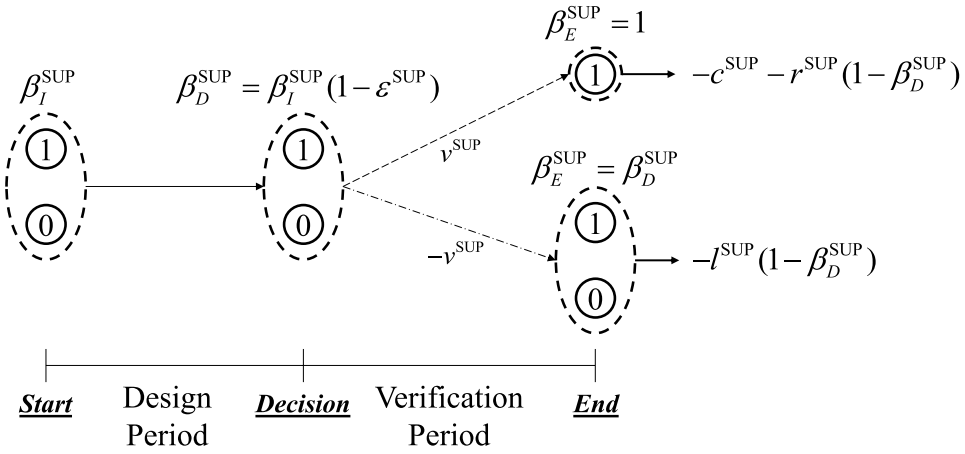

Fig. 2 Using belief distributions to capture SUP's imperfect information

3.2 Modeling imperfect knowledge with belief distributions

The scenario depicted in Fig. 1 is where SUP knows the true state of its design at all points on the time horizon. Here, SUP will verify the design only if the design is in the unsatisfactory state at the end of the design phase, $S_{\mathrm{D}}^{\mathrm{SUP}} = 0$, and if the cost of verification and repair in the next development phase is greater than the cost of verification and repair in the current development phase, $l^{\mathrm{SUP}} > c^{\mathrm{SUP}} + r^{\mathrm{SUP}}$. In reality, SUP cannot execute such a strategy since SUP has imperfect knowledge about the true state of its design prior to verification. To model SUP's imperfect knowledge in the state of its design, we use belief distributions. SUP's belief in the satisfactory state of its design at point $t \in \{I, D, E\}$ is denoted by $\beta_{\mathrm{t}}^{\mathrm{SUP}} \in [0, 1]$. In this paper, we assume that SUP's belief in the unsatisfactory state of its design at point t is $1 - \beta_{\mathrm{t}}^{\mathrm{SUP}}$. This enables us to define expected rewards for SUP's decision based on SUP's belief values. However, in general, it is not required for belief values to sum to 1 (Sentz and Ferson 2002).

SUP's initial belief in the satisfactory state of its design, $\beta_{\mathrm{I}}^{\mathrm{SUP}}$, represents SUP's confidence in the satisfactory state of the design based on the current state of knowledge of SUP. In the current design iteration, $\beta_{\mathrm{I}}^{\mathrm{SUP}}$, is transformed by the design activities into $\beta_{\mathrm{D}}^{\mathrm{SUP}}$. Since the probability of SUP making a design error is $\varepsilon^{\mathrm{SUP}}$, we know

$$\beta_{\mathrm{D}}^{\mathrm{SUP}} = \beta_I^{\mathrm{SUP}}(1 - \varepsilon^{\mathrm{SUP}}). \tag{1}$$

In addition, we know that if SUP chooses to verify its design then $\beta_{\mathrm{E}}^{\mathrm{SUP}} = 1$ since SUP identifies and fixes all errors in design; if SUP chooses not to verify its design, then $\beta_{\mathrm{E}}^{\mathrm{SUP}} = \beta_{\mathrm{D}}^{\mathrm{SUP}}$, since SUP's belief is unchanged after the design period. Figure 2 graphically depicts the single-firm model after accounting for SUP's belief in the true state of its design.

3.3 SUP's optimal verification strategy

We now determine SUP's optimal verification strategy as a function of its belief and discuss the different aspects of this strategy.

Proposition 1 *Given SUP's initial belief in the satisfactory state of the system design* $\beta_{\mathrm{I}}^{\mathrm{SUP}}$, *SUP's optimal strategy is to verify the system design if* $\beta_{\mathrm{I}}^{\mathrm{SUP}} < \frac{1}{(1-\epsilon^{\mathrm{SUP}})}\left(1 - \frac{c^{\mathrm{SUP}}}{l^{\mathrm{SUP}} - r^{\mathrm{SUP}}}\right) = \beta_{*}^{\mathrm{SUP}}$. □

Proof of Proposition 1 is provided in the "Appendix". The values $\beta_{\mathrm{I}}^{\mathrm{SUP}} \in [0, \beta_{*}^{\mathrm{SUP}})$, where β_{*}^{SUP} is a decision threshold in the belief state probability space, represents those scenarios where SUP's confidence in the satisfactory state of the design is low, and hence SUP will verify the design, whereas $\beta_{\mathrm{I}}^{\mathrm{SUP}} \in [\beta_{*}^{\mathrm{SUP}}, 1]$ represents those scenarios where SUP's confidence in the satisfactory state of the design is high, and hence SUP will prefer to postpone verification to the next design phase.

If $\beta_{*}^{\mathrm{SUP}} \geq 1$ or $\beta_{*}^{\mathrm{SUP}} \leq 0$, then the single-firm model implies that SUP would, with certainty, either always verify the system design or always not verify the system design, respectively. However, in reality, firms do not always exhibit such black-or-white behavior with respect to verification activities. Indeed, observations imply that firms often verify their design, but not necessarily at optimal times (Yamada et al. 1995; Engel and Barad 2003; Boumen et al. 2008a, b; Shabi and Reich 2012). Thus, to avoid exploring trivial scenarios, we will henceforth assume that $0 < \beta_{*}^{\mathrm{SUP}} < 1$.

For $\beta_{*}^{\mathrm{SUP}} > 0$, it must be true that $l^{\mathrm{SUP}} > c^{\mathrm{SUP}} + r^{\mathrm{SUP}}$. Note, l^{SUP} is SUP's estimate of the sum of verification setup costs and the potential rework costs further downstream in the design process that SUP will incur if it does not correct the system design in the current development phase. In general, the cost of reworking the design will increase as the design matures (to start with, more development activities will need to be repeated the later in the development process a given rework action is undertaken). In addition, we assume that the verification setup costs in l^{SUP} are at least equivalent to c^{SUP}, since the comparison is performed among equivalent verification activities (that is, the same verification activity performed at different developmental stages). Hence, it is true that $l^{\mathrm{SUP}} > c^{\mathrm{SUP}} + r^{\mathrm{SUP}}$. Similarly, for $\beta_{*}^{\mathrm{SUP}} < 1$, it must be true that $c^{\mathrm{SUP}} + r^{\mathrm{SUP}}\epsilon^{\mathrm{SUP}} > l^{\mathrm{SUP}}\epsilon^{\mathrm{SUP}}$. The amount $c^{\mathrm{SUP}} + r^{\mathrm{SUP}}\epsilon^{\mathrm{SUP}}$ is SUP's expected cost from the additional verification activity when SUP has complete confidence in the ideal state of the system design at the start of the development phase, or $\beta^{\mathrm{SUP}} = 1$. The amount $l^{\mathrm{SUP}}\epsilon^{\mathrm{SUP}}$ is then SUP's expected downstream costs of verification and rework if SUP chooses not to execute the additional verification activity in the current development phase given $\beta^{\mathrm{SUP}} = 1$. It follows that if $l^{\mathrm{SUP}}\epsilon^{\mathrm{SUP}}$ is strictly lesser than $c^{\mathrm{SUP}} + r^{\mathrm{SUP}}\epsilon^{\mathrm{SUP}}$, then it will not optimal for SUP to execute the additional verification activity for high values of β^{SUP} that are sufficiently close to 1.

4 The two-firm model

The single-firm model implies that a firm is likely to postpone verification activities to the next development phase when: (1) the verification costs between the current development phase and the next development phase are comparable, or (2) the firm is confident of not making any errors in the current design phase. Though this strategy may be optimal on a single-firm level, it may not be optimal in a multi-firm scenario. To illustrate this point, consider the robotic arm example with contractor delegating the design of the pneumatic components to a subcontractor, with the requirement of interest being the durability of the robotic arm. Stress and vibrational tests are conducted to determine if the robotic arm meets the durability requirement. Say, the current and the next design phase are prototype models for both the contractor and subcontractor; the subcontractor provides a prototype of the pneumatic components, and the contractor integrates it into the prototype of the robotic arm for further design and testing. Based on prior experience, the subcontractor is confident that its pneumatic components will meet the durability requirements set by the contractor, and thus it prefers not to spend any resources on executing stress tests on the pneumatic components. However, the contractor is not certain if the pneumatic components will withstand the required level of stress as when they are integrated into the robotic arm and would prefer if the subcontractor performed the stress tests on its components. In this scenario, the verification strategies of the two firms are not aligned since the information each of them possess are on different scales (system vs component).

To study how individual firm interest affects verification strategies in multi-firm settings, we now consider a two-firm scenario where the contractor outsources the design of one component to a subcontractor. Once again, we focus on a single development phase and a single requirement, referred to as the requirement of interest. Each firm's design phase and decision-making is modeled using the single-firm model developed in the previous section. We adopt all the single-firm model assumptions in building the two-firm model. Both firms determine their verification strategies based on their respective belief in their respective designs. We assume that the subcontractor provides its component design to the contractor, for testing purposes or otherwise. This component design is information about the product, which can be a mathematical model, a prototype, and even the final product

(eventually, in the last time period/interval). Furthermore, we assume that the subcontractor does not possess system-level information, but will convey its belief about the component meeting the requirement of interest to the contractor due to contractual requirements. Hence, the subcontractor decides whether or not to verify the component design based on component-level parameters. However, the contractor uses the subcontractor's belief to form a belief of the overall system design meeting the requirement of interest and will decidewhether or not to verify the system design based on system-level parameters. We couple the beliefs of the two firms using MSDT, as described later. In reality, inter-firm communication may not happen for all design phases, but our assumption helps set the stage for a discussion on incentives for verification activities.

In this paper, we adopt six assumptions to build an analytically tractable two-firm model. First, the subcontractor's verification costs are part of its budget, and not explicitly paid for by the contractor. Since the contractor pays for the overall component design, we assume that the verification costs for each firm are endogenous. Second, if one firm does not meet its requirements, then the overall design does not meet the requirement of interest. The contractor is able to flow down[1] the requirement of interest suitably to the subcontractor, but there is no margin for error for either firm. Third, verification on the contractor's level is comprehensive: the contractor will verify the entire system design when it chooses to verify. This is not true in general since the contractor can verify individual components. Fourth, the contractor has the monetary resources to incentivize the subcontractor to verify its component design, but the contractor will only do so if its expected reward strictly increases by incentivizing the subcontractor. In general, verification activities are negotiated at the start and additional incentives are usually not offered. This assumption sets the state to determine the effectiveness of explicit incentives for verification activities in systems engineering.

The next two assumptions we adopt are to ensure that the two-firm model is consistent with the assumptions of the single firm model. The fifth assumption we adopt is that if the subcontractor's component has an error, then the contractor will detect this error when it verifies the system design. However, the contractor does not have the capability to characterize the exact nature, or location, of this error, and hence will send the component back to the subcontractor for component-level testing and potential repair. The sixth assumption is that when an error is found in the subcontractor's component after verification at the system level, the subcontractor rectifying the design to correct the error will result in design changes on the contractor's level as well. The final assumption restricts our model to those scenarios where the contractor's system design is strongly coupled with the subcontractor's component design (Terwiesch et al. 2002, Mihm et al. 2003). Furthermore, these two last assumptions imply that the subcontractor will consider the expected cost to repair the design further in the development process when it determines whether or not to verify the component.

[1] By requirements flow down, we refer to the contractor defining a requirement for the subcontractor's design based on a requirement set for the system design. That is, the process of decomposing a system requirement into subsystem or component requirements and allocating them to the corresponding subsystems or components. For example, the contractor sets a weight limit for the subcontractor's design based on a weight limit set for the system design.

4.1 Model parameters

Using MSDT terminology, we will continue to refer to the contactor as SUP, and we will henceforth refer to the subcontractor as INF. We will initially assume that SUP provides no additional incentive for verification activities to INF, and we will later relax this assumption to study the benefits of incentivizing verification for both SUP and INF. A firm, in general, will be referred to as firm $x \in \{\text{SUP}, \text{INF}\}$. Table 1 summarizes the notation that we will use for the two-firm model.

A notable difference between the two firm and single-firm model is that INF's time horizon is shorter than the design period of SUP, since SUP integrates INF's component design into the system design once INF completes its work. Figure 3 graphically represents the two-firm model when both firms have imperfect information about the state of their design prior to verification.

An important aspect of the two firm model is the probability of a firm making a design error. INF's probability of making a design error is similar to the single-firm model, where the probability of INF making an error is a characteristic of INF alone. Whereas, SUP's probability of making a design error is considered to be affected by the component SUP delegates to INF. This results in $\beta_{\text{D}}^{\text{SUP}}$ being a function of $\beta_{\text{E}}^{\text{INF}}$. We will model the relationships between all the error probabilities and derive the precise relationship between $\beta_{\text{D}}^{\text{SUP}}$ and $\beta_{\text{E}}^{\text{INF}}$ by using MSDT's influence function approach as described next.

4.2 Influence of INF's beliefs on SUP's beliefs

There may be several reasons for which delegating a component design to INF may be beneficial to SUP, such as leveraging INF's specialization or simply compensating for lack of resource capacity for SUP. By delegating a component design to INF, SUP works on designing fewer components. This implies that the probability of SUP making a design error with design delegation is at most ϵ^{SUP}, but can be lower. However, this is true only when INF designs

Table 1 Summary of notation for the two-firm model

Notation	Description
$x \in \{\text{INF}, \text{SUP}\}$	Refers to a generic firm
$t \in \{I, D, E\}$	Point on a firm's time horizon, where I is associated with the start, D is associated with the verification decision point and E is associated with the end
$S_t^x \in \{0, 1\}$	State of firm x's design at point t, where 1 denotes a satisfactory state and 0 denotes the unsatisfactory state
ε^x	Probability of firm x making a design error during its design period without delegation
l^x	Net cost of verifying and repairing a faulty design in the next design phase rather than the current design phase
c^x, r^x	Cost of setting up verification and expected repair cost for fixing all errors in design in the current design phase
$v^x, -v^x$	Firm x's decision to verify or not verify, respectively
$\beta_t^x \in [0, 1]$	Firm x's belief in the satisfactory state of its design at point t
β_*^x	Belief threshold for firm x such that it is optimal for firm x to verify its design if $\beta_B^x < \beta_*^x$

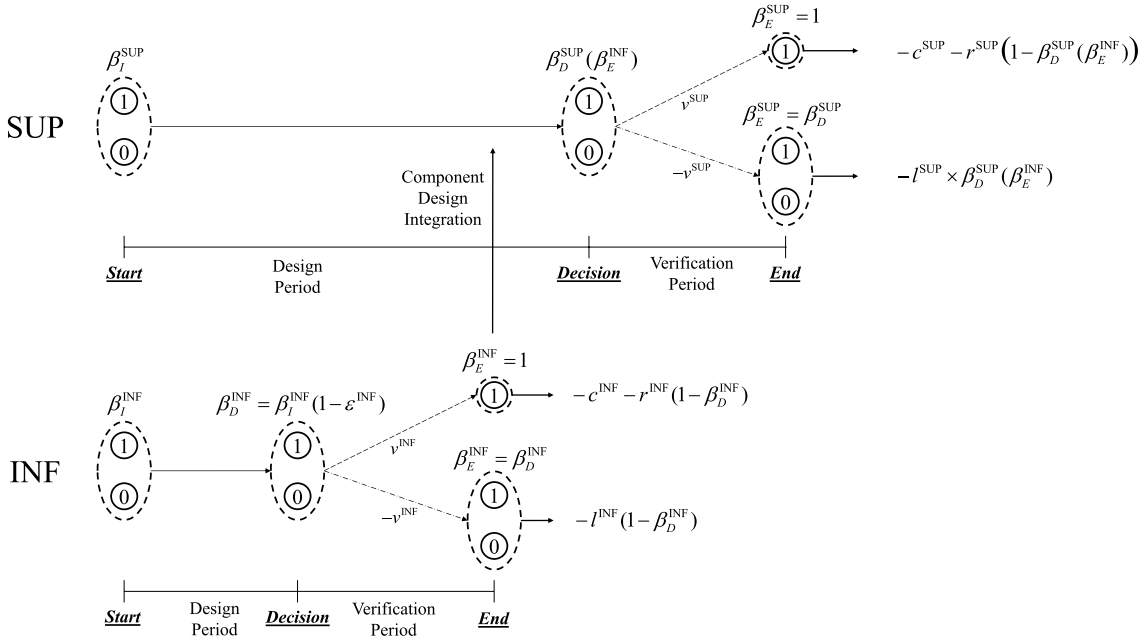

Fig. 3 Two firm verification model with imperfect information

a component that meets its requirements. For if INF designs an unsatisfactory component, then SUP's overall system design is also unsatisfactory.

To capture the SUP's value from delegating a component design to INF, we use MSDT's influence function approach as follows. Let $\varepsilon_{\text{final}}^{\text{SUP}}$ denote the probability of injecting or making an error at the end of the SUP's design phase before SUP has accounted for INF's beliefs. We define

$$\varepsilon_{\text{final}}^{\text{SUP}} = \varepsilon^{\text{SUP}} + f_{\text{INF}}(S_{\text{E}}^{\text{INF}}). \tag{2}$$

The function $f_{\text{INF}}(S_{\text{E}}^{\text{INF}})$ is referred to as the influence function in MSDT literature, and it quantifies the value of INF's work on SUP's design based on the final state of INF's component. To model the benefit of INF designing a satisfactory component to SUP and the certainty of the overall system design being unsatisfactory when INF's component design is unsatisfactory, we define $f_{\text{INF}}(\cdot)$ as follows.

$$f_{\text{INF}}(S_{\text{E}}^{\text{INF}}) = \begin{cases} -\theta & \text{if } S_{\text{E}}^{\text{INF}} = 1 \\ 1 - \varepsilon^{\text{SUP}} & \text{if } S_{\text{E}}^{\text{INF}} = 0. \end{cases} \tag{3}$$

From the unit measure axiom of probability, it follows that $1 - \varepsilon^{\text{SUP}} \geq \theta \geq 0$. When SUP accounts for INF's beliefs, $\varepsilon_{\text{final}}^{\text{SUP}}$ is defined by.

$$\varepsilon_{\text{final}}^{\text{SUP}} = \varepsilon^{\text{SUP}} + f_{\text{INF}}(S_E^{\text{INF}} = 1)\beta_E^{\text{INF}} + f_{\text{INF}}(S_E^{\text{INF}} = 0)(1 - \beta_E^{\text{INF}}). \tag{4}$$

Using Eqs. (2) and (3), we can now define $\beta_{\mathrm{D}}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$ as follows

$$\beta_{\mathrm{D}}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}}) = \beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon_{\mathrm{final}}^{\mathrm{SUP}}),$$

$$\Rightarrow \beta_{\mathrm{D}}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}}) = \beta_{\mathrm{I}}^{\mathrm{SUP}}\left(1-\left(\varepsilon^{\mathrm{SUP}}+f_{\mathrm{INF}}(S_{\mathrm{E}}^{\mathrm{INF}}=1)\beta_{\mathrm{E}}^{\mathrm{INF}}+f_{\mathrm{INF}}(S_{\mathrm{E}}^{\mathrm{INF}}=0)(1-\beta_{\mathrm{E}}^{\mathrm{INF}})\right)\right),$$

$$\Rightarrow \beta_{\mathrm{D}}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}}) = \beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta\beta_{\mathrm{E}}^{\mathrm{INF}}-(1-\varepsilon^{\mathrm{SUP}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})),$$

$$\Rightarrow \beta_{\mathrm{D}}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}}) = \beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)\beta_{\mathrm{E}}^{\mathrm{INF}}. \tag{5}$$

In the single firm model, SUP's belief was transformed by the factor $\varepsilon^{\mathrm{SUP}}$. Whereas, in the two-firm model, as described by Eq. (5), SUP's belief is transformed by the factor $(1-\varepsilon^{\mathrm{SUP}}+\theta)$ and INF's belief $\beta_{\mathrm{E}}^{\mathrm{INF}}$. In this regard, the factor θ is SUP's assessment of the benefits of delegating the component design to INF. Since delegation of design activities ensures that the probability of SUP making a design error effectively reduces from $\varepsilon^{\mathrm{SUP}}$ to $\varepsilon^{\mathrm{SUP}}-\theta$. It follows that $\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)$ represents SUP's belief in not making any design errors in SUP's portion of the design activities.

4.3 Optimal verification strategy without incentives

We now characterize each firm's optimal verification strategy and discuss the different aspects of these strategies. We begin with INF's verification strategy, followed by SUP's.

Proposition 2 *Given that SUP will not incentivize INF to verify its component design and INF's initial belief in the satisfactory state of the component design if* $\beta_{\mathrm{I}}^{\mathrm{INF}}$, *it is optimal for INF to verify the component design if* $\beta_{I}^{\mathrm{INF}} < \frac{1}{(1-\varepsilon^{\mathrm{INF}})}\left(1-\frac{c^{\mathrm{INF}}}{l^{\mathrm{INF}}-r^{\mathrm{INF}}}\right) = \beta_{*}^{\mathrm{INF}}$. □

Proposition 3 *Given that INF's final belief in the satisfactory state of the component design is* $\beta_{\mathrm{E}}^{\mathrm{INF}}$, *and thus SUP's belief in the satisfactory state of the system design at the end of SUP's design period is* $(1-\varepsilon^{\mathrm{SUP}}+\theta)\beta_{\mathrm{E}}^{\mathrm{INF}}$, *it is optimal for SUP to comprehensively verify the system design if* $\beta_{\mathrm{I}}^{\mathrm{SUP}} < \frac{1}{(1-\varepsilon^{\mathrm{SUP}}+\theta)\beta_{\mathrm{E}}^{\mathrm{INF}}}\left(1-\frac{c^{\mathrm{SUP}}}{l^{\mathrm{SUP}}-r^{\mathrm{SUP}}}\right) = \beta_{*}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$. □

Proposition 2 follows from the single firm model results. Proof of Proposition 3 is provided in the "Appendix". SUP's optimal verification strategy is similar in structure to the single-firm model, with the notable difference being SUP's decision threshold β_{*}^{SUP} is now a function of $\beta_{\mathrm{E}}^{\mathrm{INF}}$ in addition to being a function of SUP's cost and skill parameters.

We visualize SUP's verification strategy with a two-dimensional phase diagram, where one axis is INF's reported final belief in the satisfactory state of the component design at the end of its design phase, $\beta_{\mathrm{E}}^{\mathrm{INF}}$, and the other axis is SUP's initial belief in the satisfactory state of the overall system design, $\beta_{\mathrm{I}}^{\mathrm{SUP}}$. To illustrate SUP's phase diagram, we use the following notional values: $c^{\mathrm{SUP}} = \$\,2000$, $r^{\mathrm{SUP}} = \$\,1000$, $l^{\mathrm{SUP}} = \$\,4000$, $\varepsilon^{\mathrm{SUP}} = 0.3$ and $\theta = 0.1$. The phase diagram resulting from these values is shown in Fig. 4. We see that SUP's phase diagram consists of two distinct regions when SUP chooses not to incentivize INF for its verification activities. The bottom left region is where SUP's optimal strategy is to verify the design, and the top right region is where SUP's optimal strategy is to not verify its design. The nonlinear boundary between the two regions is the curve $\beta_{*}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$ as defined by Proposition 3.

There are two potential benefits for SUP in incentivizing INF to verify its component design. The first potential benefit is SUP avoiding unnecessary verification of the system design. Ideally, INF always verifies its design, $\beta_{\mathrm{E}}^{\mathrm{INF}} = 1$, and so SUP does not verify its design when $\beta_{\mathrm{I}}^{\mathrm{SUP}} > \beta_{*}^{\mathrm{SUP}}(1)$. However, the single-firm model implies that in some cases, INF will find it optimal to postpone verification. The adverse effects of INF adopting a locally optimal strategy on SUP is illustrated in Fig. 4 by the substantial region above $\beta_{\mathrm{I}}^{\mathrm{SUP}} = \beta_{*}^{\mathrm{SUP}}(1)$ where SUP verifies its design only because INF chooses to postpone verification.

The second potential benefit of incentivizing INF to verify its design is the minimization of expected repair costs for SUP. If $\beta_{\mathrm{I}}^{\mathrm{SUP}} < \beta_{*}^{\mathrm{SUP}}(1)$, then SUP will verify its design irrespective of INF's final reported belief $\beta_{\mathrm{E}}^{\mathrm{INF}}$. By doing so, SUP will incur a fixed setup cost c^{SUP}. However, SUP's expected repair costs are dependent on the system design not meeting the requirement of interest. By incentivizing INF, SUP ensures that INF's component design meets the requirement of interest, and hence SUP's expected repair costs are minimized.

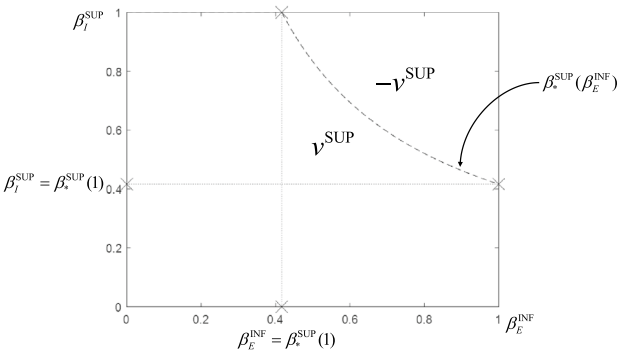

Fig. 4 Phase diagram of SUP's verification strategies under no incentives

5 Incentivizing verification activities

5.1 Optimal incentive for INF's verification activities

We now characterize the minimum necessary incentive required to motivate INF to verify its design. We also discuss an incentive mechanism that SUP can use to ensure INF does not abuse the provision for verification incentives. For the incentive mechanism, we will restrict our attention to those scenarios where INF repairing an error in its design provides a reasonable signal to SUP. For example, if INF discovers that its pneumatic component cannot stand the stress test, then it takes a significant amount of time for INF to correct the error, and thus SUP knows that INF found an error in its design. This assumption eliminates those scenarios from our study where SUP incentivizes INF to verify its component design, INF discovers an error in its component design and INF repairs the error without SUP's knowledge.

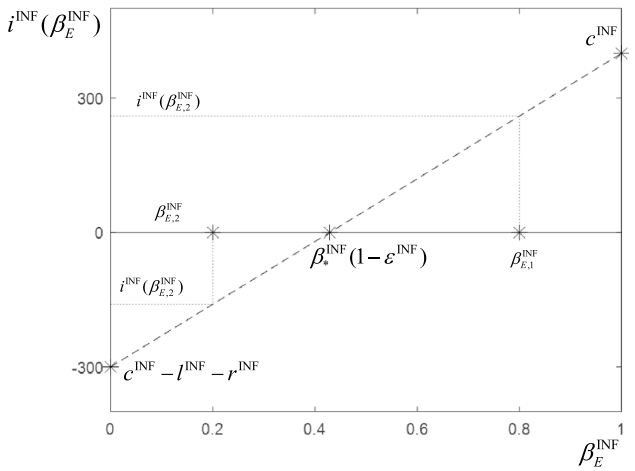

Fig. 5 Change in minimum verification incentive with increasing INF belief

Proposition 4 *If* $\beta_{\mathrm{I}}^{\mathrm{INF}} \geq \beta_*^{\mathrm{INF}}$, *then* $\beta_{\mathrm{E}}^{\mathrm{INF}} = \beta_{\mathrm{I}}^{\mathrm{INF}}(1-\epsilon^{\mathrm{INF}})$, *and the minimum necessary incentive required to motivate INF to verify its component is* $i^{\mathrm{INF}}(\beta_{\mathrm{E}}^{\mathrm{INF}}) = c^{\mathrm{INF}} - (l^{\mathrm{INF}} - r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})$. □

Proof of Proposition 4 is provided in the "Appendix". The optimal verification incentive $i^{\mathrm{INF}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$ consists of the reward component c^{INF} and the penalty component $(l^{\mathrm{INF}} - r^{\mathrm{INF}})$. This reward/penalty nature of $i^{\mathrm{INF}}(\cdot)$ arises from the need to balance the interests of both firms. If INF's design has no errors, then INF's belief in the satisfactory state of the system design is justified, and INF would be justified in expecting SUP to completely reimburse INF's setup cost. However, if INF's design has an error, then SUP would be justified in levying a penalty on INF since delivering an error-free design was INF's responsibility. The optimal value of the penalty is then $(l^{\mathrm{INF}} - r^{\mathrm{INF}})$. This is so since INF avoids incurring $(l^{\mathrm{INF}} - r^{\mathrm{INF}})$ in the future by accepting SUP's incentive to verify the component design in the current development phase. However, this gain belongs to SUP since INF's original strategy was to postpone verification and it was SUP that directed INF to verify its design.

We now illustrate the variation in i^{INF} with respect to $\beta_{\mathrm{E}}^{\mathrm{INF}}$ using the following notional values for INF: $c^{\mathrm{INF}} = \$400$, $r^{\mathrm{INF}} = \$100$ and $l^{\mathrm{INF}} = \$800$. Figure 5 graphs the values of $i^{\mathrm{INF}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$ with respect to $\beta_{\mathrm{E}}^{\mathrm{INF}}$ for these notational values. As illustrated in Fig. 5, when INF has complete confidence in the satisfactory state of the design, $\beta_{\mathrm{E}}^{\mathrm{INF}} = 1$, then SUP must offer to completely reimburse INF's setup cost c^{INF} to motivate INF to verify the component design. In this case, complete reimbursement of the setup cost is necessary since it is not in INF's interest to incur any verification costs when $\beta_{\mathrm{E}}^{\mathrm{INF}} = 1$, and is sufficient since INF will not expect to incur any repair costs when $\beta_{\mathrm{E}}^{\mathrm{INF}} = 1$. As INF's belief in the satisfactory state of the component design decreases, the incentive SUP has to offer also decreases. This is true for when INF's belief in the satisfactory state of the system design is less than 1, INF believes there is a chance of its component design not meeting the requirement of interest. Since delivering a component design that meets the requirement of interest is INF's responsibility, it is in INF's interest to accept an incentive amount less than c^{INF} as long as it is commensurate with INF's belief in the satisfactory state of the system design, where the commensurate incentive for $\beta_{\mathrm{E}}^{\mathrm{INF}} < 1$ is equal to $c^{\mathrm{INF}} - (l^{\mathrm{INF}} - r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})$.

As illustrated in Fig. 5, we see that $i^{\mathrm{INF}}(\beta_{\mathrm{E}}^{\mathrm{INF}}) < 0$ when $\beta_{\mathrm{E}}^{\mathrm{INF}} < \beta_*^{\mathrm{INF}}(1-\epsilon^{\mathrm{INF}})$. This implies that it is in SUP's interest to charge a penalty if INF seeks an incentive from SUP when $\beta_{\mathrm{E}}^{\mathrm{INF}} < \beta_*^{\mathrm{INF}}(1-\epsilon^{\mathrm{INF}})$, since for these belief values it is INF's interest to verify the component design even without incentives. In reality, SUP will simply not incentivize the INF when $\beta_{\mathrm{E}}^{\mathrm{INF}} < \beta_*^{\mathrm{INF}}(1-\epsilon^{\mathrm{INF}})$. However, the lack of incentives for belief values $\beta_{\mathrm{E}}^{\mathrm{INF}} < \beta_*^{\mathrm{INF}}(1-\epsilon^{\mathrm{INF}})$ would encourage INF to always report a final belief value $\hat{\beta}_{\mathrm{E}}^{\mathrm{INF}} > \beta_*^{\mathrm{INF}}(1-\epsilon^{\mathrm{INF}})$, since it is in INF's interest to elicit incentives for verification activities from SUP irrespective of whether or not INF deserves it. That means, that it is always in INF's best interest to report $\hat{\beta}_{\mathrm{E}}^{\mathrm{INF}} = 1$ to receive the maximum incentive.

To avoid encouraging INF in reporting a false belief value, it is in SUP's interest to convert the incentive mechanism $i^{\mathrm{INF}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$, which relies on the truthfulness of the information provided by INF, into another incentive mechanism that does not depend on INF reporting its belief values truthfully to SUP, but instead relies on INF's choice on whether or not to participate in the incentive mechanism. We now

present one such incentive mechanism, which we denote by Γ.

Proposition 5 *When INF chooses to participate in the incentive mechanism* Γ, *then*,

(1) *SUP will compensate for INF's setup cost,* c^{INF}, *upfront, and*
(2) *When INF discovers errors in design during verification, INF will pay* $(l^{\text{INF}} - r^{\text{INF}})$ *to SUP as a penalty fee.*

The incentive mechanism Γ discourages INF requesting verification incentives when $\beta_I^{\text{INF}} < \beta_*^{\text{INF}}$, and provides INF with the minimum necessary incentives when $\beta_I^{\text{INF}} \geq \beta_*^{\text{INF}}$.

Given that INF's final belief in the satisfactory state of the system design is $\beta_{\text{E}}^{\text{INF}}$, INF's expected incentive from participating in the incentive mechanism Γ, denoted by $\text{E}(i^{\text{INF}}|\beta_{\text{E}}^{\text{INF}})$, is

$$\text{E}(i^{\text{INF}}|\beta_{\text{E}}^{\text{INF}}) = c^{\text{INF}} - (c^{\text{INF}} + a^{\text{INF}})(1 - \beta_{\text{E}}^{\text{INF}})$$

$$\Rightarrow \text{E}(i^{\text{INF}}|\beta_E^{\text{INF}}) = c^{\text{INF}}\beta_E^{\text{INF}} - a^{\text{INF}}(1 - \beta_E^{\text{INF}}).$$

Since $\text{E}(i^{\text{INF}}|\beta_{\text{E}}^{\text{INF}}) = i^{\text{INF}}(\beta_{\text{E}}^{\text{INF}})$, the incentive mechanism Γ provides the minimum incentive necessary to motivate INF to verify its component design by relying on observable events (the result of the verification activity) rather than using INF's reported value of $\beta_{\text{E}}^{\text{INF}}$. This ensures that INF has no incentive to falsify its belief value, and thus will report its true belief value to SUP. Note, the penalty in the mechanism Γ can be imposed since we assume SUP will know if INF finds an error in its component.

An advantage of using Γ is that SUP does not need to know ε^{INF}. However, the incentive mechanism Γ requires SUP to know the amount $l^{\text{INF}} - r^{\text{INF}}$ to define the penalty amount because INF will find the incentive mechanism acceptable when the penalty amount is close to INF's estimate of $l^{\text{INF}} - r^{\text{INF}}$. There are several methods to obtain or estimate $l^{\text{INF}} - r^{\text{INF}}$, such as through business intelligence, direct negotiation, or SUP's historical records of INF's past performance data, but they are outside of the scope of this paper.

5.2 Optimal two-firm verification strategy with incentives

When SUP offers no incentives to INF, INF's decision to verify the component design or not is final. Whereas, with incentives, SUP may request INF to verify its design after INF has decided to postpone verification activities to the next development phase. This sets up a coordination challenge between the two firms with respect to verification activities. If SUP knows the set of belief pairs $(\beta_I^{\text{SUP}}, \beta_{\text{E}}^{\text{INF}})$ for which it is profitable to incentivize INF to verify its component design beforehand, then coordinating verification strategies becomes easier in the two-firm scenario. Toward this end, we now determine the belief pairs $(\beta_I^{\text{SUP}}, \beta_{\text{E}}^{\text{INF}})$ for which incentivizing INF to verify its design is beneficial for SUP. In this regard, we state the following two propositions.

Proposition 6 *If* $\beta_{\text{E}}^{\text{INF}} \geq \beta_*^{\text{INF}}(1 - \varepsilon^{\text{INF}})$ *and if any one of the following two conditions are true*,

(i) $\beta_{\text{I}}^{\text{SUP}} < \beta_*^{\text{SUP}}(1)$, *or*,
(ii) $\beta_{\text{I}}^{\text{SUP}} < \min\{\beta_*^{\text{SUP}}(\beta_E^{\text{INF}}), \beta_\diamond^{\text{SUP}}(\beta_{\text{E}}^{\text{INF}})\}$ *and* $\beta_I^{\text{SUP}} \geq \beta_*^{\text{SUP}}(1)$,
where $\beta_\diamond^{\text{SUP}}(\beta_{\text{E}}^{\text{INF}}) = \frac{(l^{\text{SUP}} - r^{\text{SUP}}) + c^{\text{INF}} - (l^{\text{INF}} - r^{\text{INF}})(1 - \beta_{\text{E}}^{\text{INF}})}{(1 - \varepsilon^{\text{SUP}} + \theta)(l^{\text{SUP}} - r^{\text{SUP}}\beta_{\text{E}}^{\text{INF}})}$,
then it is optimal for SUP to verify the system design.

Proposition 7 *Given* $\beta_{\text{E}}^{\text{INF}} \geq \beta_*^{\text{INF}}(1 - \varepsilon^{\text{INF}})$, *it is optimal for SUP to incentivize INF's verification activities when one of the following conditions is true*,

(1) $\beta_\nabla^{\text{SUP}}(\beta_E^{\text{INF}}) < \beta_I^{\text{SUP}} < \beta_*^{\text{SUP}}(1)$,
(2) $\beta_\diamond^{\text{SUP}}(\beta_E^{\text{INF}}) < \beta_I^{\text{SUP}} < \beta_*^{\text{SUP}}(\beta_E^{\text{INF}})$, *or,*
(3) $\beta_I^{\text{SUP}} > \max\{\beta_*^{\text{SUP}}(\beta_E^{\text{INF}}), \beta_\triangle^{\text{SUP}}(\beta_E^{\text{INF}})\}$,
where $\beta_\nabla^{\text{SUP}}(\beta_E^{\text{INF}}) = \frac{c^{\text{INF}} - (l^{\text{INF}} - r^{\text{INF}})(1 - \beta_E^{\text{INF}})}{r^{\text{SUP}}(1 - \varepsilon^{\text{SUP}} + \theta)(1 - \beta_E^{\text{INF}})}$ *and* $\beta_\triangle^{\text{SUP}}(\beta_E^{\text{INF}}) = \frac{c^{\text{INF}} - (l^{\text{INF}} - r^{\text{INF}})(1 - \beta_E^{\text{INF}})}{l^{\text{SUP}}(1 - \varepsilon^{\text{SUP}} + \theta)(1 - \beta_E^{\text{INF}})}$

Proofs of Propositions 6 and 7 are provided in the "Appendix". From the single-firm model, we know it is in INF's interest to verify the component design, even without incentives, when $\beta_{\text{I}}^{\text{INF}} < \beta_*^{\text{INF}} \Rightarrow \beta_{\text{E}}^{\text{INF}} < \beta_*^{\text{SUP}}(1 - \varepsilon^{\text{INF}})$. Furthermore, due to the penalty imposed by the incentive mechanism Γ, only when $\beta_{\text{E}}^{\text{INF}} > \beta_*^{\text{SUP}}(1 - \varepsilon^{\text{INF}})$ will INF be willing to accept incentives for verification from SUP. Hence, valid belief pairs $(\beta_{\text{I}}^{\text{SUP}}, \beta_{\text{E}}^{\text{INF}})$ that SUP needs to consider for coordinating the incentive-backed two-firm verification strategy are those where $\beta_{\text{E}}^{\text{INF}} > \beta_*^{\text{SUP}}(1 - \varepsilon^{\text{INF}})$ and $0 \leq \beta_{\text{I}}^{\text{SUP}} \leq 1$. In addition, since INF will not verify its component design without incentives for $\beta_{\text{E}}^{\text{INF}} > \beta_*^{\text{SUP}}(1 - \varepsilon^{\text{INF}})$, we know that for all valid belief pairs there are only four potential verification strategies that SUP needs to consider for incentive purposes: $(v^{\text{SUP}}, -v^{\text{INF}})$, $(v^{\text{SUP}}, v_i^{\text{INF}})$, $(-v^{\text{SUP}}, -v^{\text{INF}})$ and $(-v^{\text{SUP}}, v_i^{\text{INF}})$, where v_i^{INF} denotes that INF verifies the component design with incentives.

We now illustrate the incentive-backed two-firm verification strategy using the phase diagram method presented in Sect. 4. For the purpose of this illustration, we use the notional values presented in Table 2. The phase diagram is graphed in Fig. 6. The incentive-backed optimal two-firm

Table 2 Parameter values for the two firms

Parameter	SUP's value	INF's value
c^x	\$2,000	\$400
r^x	\$1,000	\$100
l^x	\$4,000	\$550
ε^x	0.3	0.1
θ	0.1	NA

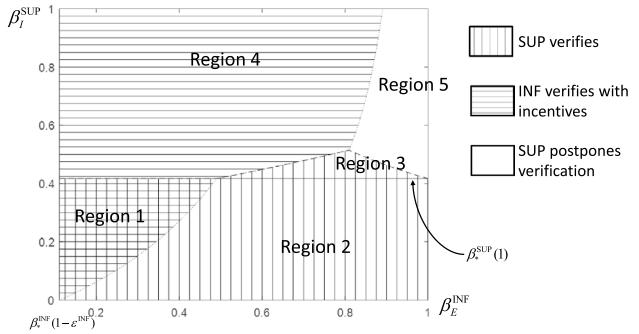

Fig. 6 Effect of incentivizing INF's verification activities on SUP's strategy

verification strategy is illustrated with hatches in the phase diagram. The vertical hatch denotes that SUP will verify the system design, irrespective of whether or not it incentivizes INF to verify the component design. The horizontal hatch denotes that INF will verify the component design when SUP offers the minimum incentive for verification.

Without incentives, INF will not verify its design for *the entirety* of SUP's phase diagram graphed in Fig. 6. Whereas, with incentives, we see in Fig. 6 that INF will verify its design for a large portion of the phase diagram. SUP's phase diagram can be divided into five distinct regions. In regions 1 and 2, $\beta_{\mathrm{I}}^{\mathrm{SUP}} < \beta_{*}^{\mathrm{SUP}}(1)$, and hence SUP has to verify its design irrespective of INF's strategy. In region 1, it is profitable for SUP to incentivize INF to verify its design. By doing so, SUP is able to minimize its repair costs. However, in region 2, we see that INF's belief in the satisfactory state of the component design is sufficiently high for SUP to overlook incentivizing INF. The benefits of incentivizing INF are highlighted by regions 3, 4 and 5. Prior to incentivizing INF, SUP would verify the system design for all $\beta_{*}^{\mathrm{SUP}}(1) < \beta_{\mathrm{I}}^{\mathrm{SUP}} < \beta_{*}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$. Whereas, with incentives, when $\beta_{*}^{\mathrm{SUP}}(1) < \beta_{\mathrm{I}}^{\mathrm{SUP}} < \beta_{*}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$, SUP verifies the system design only in region 3.

In region 4, INF's belief in the satisfactory state of the system design is low, and hence SUP prefers to incentivize INF since SUP believes that INF will find an error in its design. Whereas, in region 5, INF's belief in the satisfactory state of the system design is high, and this leads SUP to believe that incentivizing INF will result in a loss of c^{INF} for SUP rather than INF finding an error in its design. Unlike regions 4 and 5, region 3 is a challenging region for SUP. Here, INF's belief is sufficiently low to influence SUP to verify the system design even though $\beta_{\mathrm{I}}^{\mathrm{SUP}} \geq \beta_{*}^{\mathrm{SUP}}(1)$. However, INF's belief is not low enough for SUP to find it profitable to incentivize INF. Hence, in region 3, SUP will verify the system design without incentivizing INF.

5.3 Effect of parameter values on two-firm strategy

We now discuss the effect of varying model parameter values on the incentive-backed two-firm verification strategy. Of all the model parameters that affect SUP's phase diagram when SUP chooses to incentivize INF, we consider only c^{SUP}, r^{SUP}, l^{SUP}, θ and c^{INF} for our study. We ignore $\varepsilon^{\mathrm{SUP}}$ since the effect of $\varepsilon^{\mathrm{SUP}}$ on SUP's decision can be deduced by varying θ. Whereas, we ignore l^{INF} since it is only meaningful for SUP to incentivize INF when l^{INF} is low, and it is in INF's interest to verify the component design for high l^{INF} values. We consider the parameter values listed in Table 2 as the baseline values for our analysis.

Figure 7 graphs SUP's phase diagrams for the different values of c^{SUP}. As illustrated in Fig. 7, changing the value of c^{SUP} does not affect the two-firm strategy for all valid belief

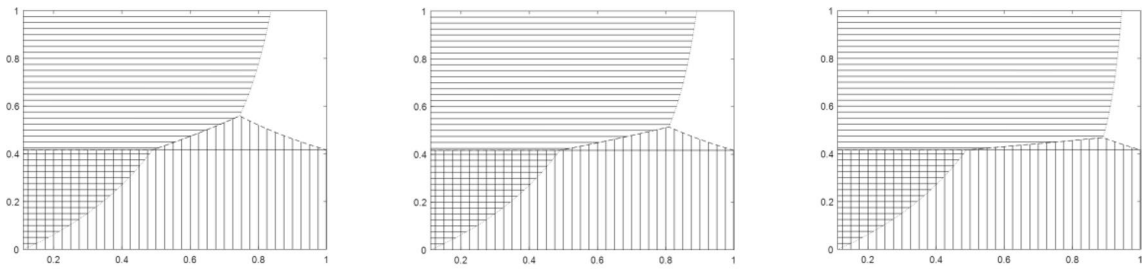

Fig. 7 Change in SUP's phase diagram based on SUP's setup cost

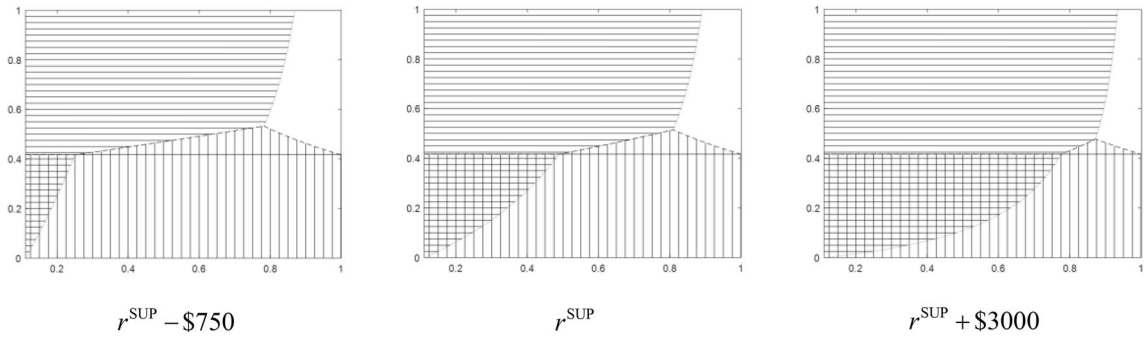

Fig. 8 Change in SUP's phase diagram based on SUP's repair costs

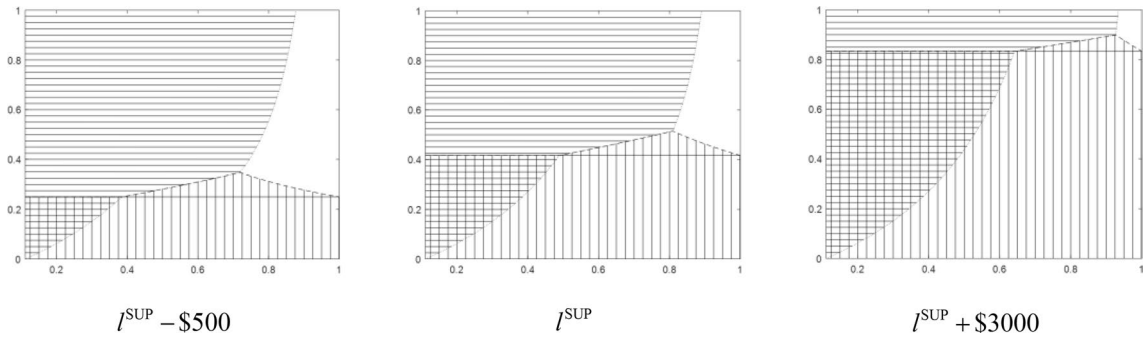

Fig. 9 Change in SUP's phase diagram based on the potential increase in future costs

pairs with $\beta_{\mathrm{I}}^{\mathrm{SUP}} < \beta_{*}^{\mathrm{SUP}}(1)$. However, for all valid belief pairs with $\beta_{\mathrm{I}}^{\mathrm{SUP}} \geq \beta_{*}^{\mathrm{SUP}}(1)$, we see that as c^{SUP} increases, the area of region 4 increases while the areas of regions 3 and 5 decrease. That is, as c^{SUP} increases, the number of belief pairs for which incentivizing SUP is the optimal strategy increases. This is so since SUP will postpone verification when INF verifies its design when $\beta_{\mathrm{I}}^{\mathrm{SUP}} \geq \beta_{*}^{\mathrm{SUP}}(1)$, and as c^{SUP} increases, SUP finds it in its interest to avoid incurring a high verification setup cost by incentivizing INF to verify its design.

Next, we studied the effect of r^{SUP} on the two-firm strategy by varying the value of r^{SUP} while fixing all other parameter values at their baseline. Figure 8 illustrates the effect of varying r^{SUP} on the two-firm strategy. As illustrated in Fig. 8, varying r^{SUP} has resulted in a noticeable change throughout SUP's phase diagram. Specifically, as r^{SUP} increases, the areas of regions 1 and 4 increase, while the areas of the other regions decrease. That is, as r^{SUP} increases, the number of belief pairs for which for SUP incentivizing INF to verify its design increases. This is so since SUP wishes to minimize the expected repair costs on its end by incentivizing INF to repair all errors in the component design.

The effect of changing l^{SUP} on the phase diagram is illustrated in Fig. 9. We see that as l^{SUP} increases, SUP will prefer to verify the system design. Hence, SUP is will incentivize INF to minimize SUP's expected repair costs. Figure 10 illustrates the effect of a variation in θ on the two-firm strategy. As θ increases, SUP's net probability of making an error in the system design decreases. Hence, we see in Fig. 10, as θ increases, SUP will incentivize INF to postpone its verification activities to the next design phase.

Finally, the variation in the two-firm strategy due to a change in c^{INF} is illustrated in Fig. 11. As one would expect, with an increase in c^{INF}, SUP will incentivize INF to verify its component design. This is so, since with increasing c^{INF}, the valid belief pairs $(\beta_I^{\mathrm{SUP}}, \beta_E^{\mathrm{INF}})$ for which SUP finds it profitable to incentivize INF's verification activities decreases.

6 Conclusion

Our work seeks to lay a foundation for the theoretical understanding of verification activities in multi-firm projects. Toward this end, we used belief distributions to model each firm's epistemic uncertainty in the true state of its design. We considered three high-level verification costs: setup cost, repair or rework cost and the cost of postponing verification of a faulty design. Analysis of our single firm model showed that a firm will postpone verification activities if (1) it has high confidence its ability to not make an error in design,

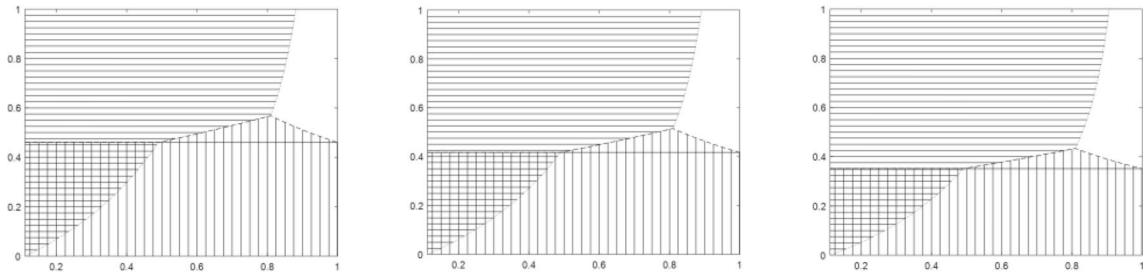

Fig. 10 Change in SUP's phase diagram based on the influence parameter theta

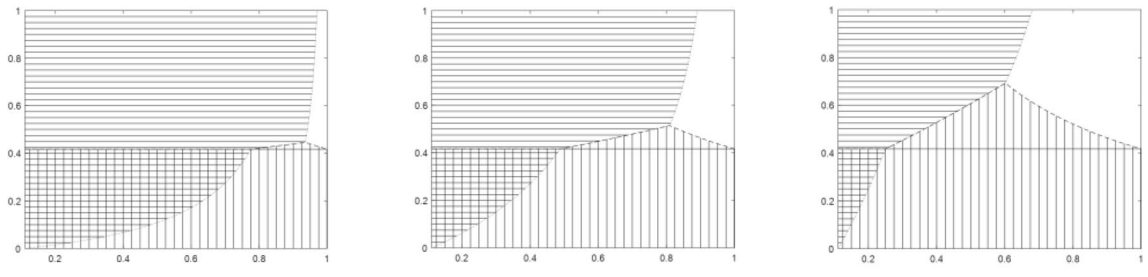

Fig. 11 Change in SUP's phase diagram based on INF's setup cost

and (2) if it has a high belief in its design meeting requirements and the cost of postponing verification of a faulty design is not significantly greater than the current setup and repair cost. The intuitive nature of these results led us to believe that the foundational single-firm model was suitable for extension to multi-firm models.

The two-firm extension studied how the subcontractor's preference to postpone verification activities adversely affects the contractor. Specifically, we showed that the subcontractor postponing verification activities forces the contractor to verify the system design in a large number of scenarios, where the contractor would have preferred to postpone verification activities had the subcontractor chosen to verify its design. Given this conflict of interest, we developed an incentive mechanism by which the contractor could suitably motivate the subcontractor to verify its design while ensuring the subcontractor didn't abuse the incentive mechanism. Using the two-firm model, we then identified the scenarios where the contractor would benefit from incentivizing the subcontractor's verification activities. Results of the parameter variation of our model showed that the contractor is motivated to incentivize the subcontractor for two reasons: (1) to minimize its own repair costs, and (2) to improve its own belief in the system design meeting requirements so as to postpone verification activities.

The analysis of our single-firm and two-firm models revealed that incentivizing verification activities can be beneficial for the contractor. However, we have made significant assumptions to ensure analytical tractability for our models. Specifically, we have restricted our study to high-level cost parameters, and we have assumed that the firms will be able to quantify the probability of making errors during the design process. In addition, we also assumed that when SUP incentivizes INF to verify its design, INF cannot lie about finding an error in its design to SUP. We have adopted these assumptions with the knowledge that our work provides the foundation for future extensions where our model assumptions can be relaxed to derive more general results. An interesting extension would be to generalize the manner in which we have coupled the beliefs of the two firms using MSDT. Specifically, the coupling we use in this paper is linear in nature, whereas, in general, the coupling of beliefs would be non-linear.

In conclusion, we have developed a mathematical model of verification that incorporate belief distributions and thereby could model the evolving knowledge of an agent has about the state of its system's design. By using the MSDT modeling approach to determine optimal incentives for verification, a problem that has previously been unexplored in systems engineering, we are providing a framework that can determine optimal incentive for verification in a two-firm

setting. This work builds a foundation for future extensions on incentives for verification in multi-firm networks, which design complex engineering systems, and entail complex contractor-subcontractor interaction relationships.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant Nos. CMMI-1762883 and CMMI-1762336. The authors thank the reviewers for their comments, which have significantly improved the manuscript, as well as for suggestion the durability example to contextualize the model setup presented in Sect. 4.

Appendix

Proof of Proposition 1

SUP's expected reward is dependent on its verification strategy and its belief in the satisfactory state of the design at the end of the design period, β_D^{SUP}. Since β_D^{SUP} is a transformed value of SUP's initial belief in the satisfactory state of its design, β_I^{SUP}, SUP's expected reward associated with each verification strategy as a function of β_I^{SUP} are given by

$$\text{E}(R^{\text{SUP}}|\beta_{\text{I}}^{\text{SUP}}, v^{\text{SUP}}) = -c^{\text{SUP}} - r^{\text{SUP}}(1-\beta_{\text{I}}^{\text{SUP}}(1-\varepsilon^{\text{SUP}})), \text{ and} \tag{6}$$

$$\text{E}(R^{\text{SUP}}|\beta_{\text{I}}^{\text{SUP}}, -v^{\text{SUP}}) = -l^{\text{SUP}}(1-\beta_{\text{I}}^{\text{SUP}}(1-\varepsilon^{\text{SUP}})). \tag{7}$$

SUP will prefer to verify the system design only if

$$\text{E}(R^{\text{SUP}}|v^{\text{SUP}}) > \text{E}(R^{\text{SUP}}|-v^{\text{SUP}})$$

$$\Rightarrow \beta_I^{\text{SUP}} < \frac{1}{(1-\varepsilon^{\text{SUP}})}\left(1-\frac{c^{\text{SUP}}}{l^{\text{SUP}}-r^{\text{SUP}}}\right) = \beta_*^{\text{SUP}}. \tag{8}$$

Proof of Proposition 3

For SUP, we know that SUP will consider $\beta_{\text{E}}^{\text{INF}}$ since INF communicates $\beta_{\text{E}}^{\text{INF}}$ to SUP at the end of INF's design phase. Thus, in addition to its endogenous cost and skill parameters, a rational SUP will also use $\beta_{\text{E}}^{\text{INF}}$ to determine the optimal verification strategy. Using the definition of $\beta_{\text{D}}^{\text{SUP}}$ from Eq. (5), SUP's expected reward associated with each of its verification strategies is defined by,

$$\begin{aligned}&\text{E}(R^{\text{SUP}}|\beta_{\text{I}}^{\text{SUP}}, v^{\text{SUP}}, \beta_{\text{E}}^{\text{INF}})\\ &\quad = -c^{\text{SUP}} - r^{\text{SUP}}(1-\beta_{\text{I}}^{\text{SUP}}(1-\varepsilon^{\text{SUP}}-\theta)\beta_{\text{E}}^{\text{INF}}) \text{ and,}\end{aligned} \tag{9}$$

$$\text{E}(R^{\text{SUP}}|\beta_I^{\text{SUP}}, -v^{\text{SUP}}, \beta_{\text{E}}^{\text{INF}}) = -l^{\text{SUP}}(1-\beta_{\text{I}}^{\text{SUP}}(1-\varepsilon^{\text{SUP}}-\theta)\beta_{\text{E}}^{\text{INF}}). \tag{10}$$

SUP will then prefer to comprehensively verify the component design only if

$$\text{E}(R^{\text{SUP}}|\beta_I^{\text{SUP}}, v^{\text{SUP}}, \beta_E^{\text{INF}}) > \text{E}(R^{\text{SUP}}|\beta_I^{\text{SUP}}, -v^{\text{SUP}}, \beta_E^{\text{INF}}),$$

$$\Rightarrow \beta_{\text{I}}^{\text{SUP}} < \frac{1}{(1-\varepsilon^{\text{SUP}}+\theta)\beta_{\text{E}}^{\text{INF}}}\left(1-\frac{c^{\text{SUP}}}{l^{\text{SUP}}-r^{\text{SUP}}}\right) = \beta_*^{\text{SUP}}(\beta_{\text{E}}^{\text{INF}}). \tag{11}$$

Proof of Proposition 4

The minimum necessary incentive makes INF indifferent between verifying and not verifying the component design. Let i^{INF} denote the optimal verification incentive, or the minimum necessary incentive, offered by SUP to INF to make INF indifferent between verifying and not verifying the component design, and let v_i^{INF} denote INF's decision to verify the component design when SUP offers INF i^{INF} to verify its design. When $\beta_{\text{I}}^{\text{INF}} > \beta_*^{\text{INF}}$, INF's expected reward associated with v_i^{INF} and $-v^{\text{INF}}$ are given by,

$$\text{E}(R^{\text{INF}}|\beta_{\text{I}}^{\text{INF}}, v_i^{\text{INF}}) = -c^{\text{INF}} - r^{\text{INF}}(1-\beta_I^{\text{INF}}(1-\varepsilon^{\text{INF}})) + i^{\text{INF}}, \text{ and} \tag{12}$$

$$\text{E}(R^{\text{INF}}|\beta_{\text{I}}^{\text{INF}}, -v^{\text{INF}}) = -l^{\text{INF}}(1-\beta_{\text{I}}^{\text{INF}}(1-\varepsilon^{\text{INF}})). \tag{13}$$

The optimal value of i^{INF} is such that

$$\text{E}(R^{\text{INF}}|\beta_{\text{I}}^{\text{INF}}, v_i^{\text{INF}}) = \text{E}(R^{\text{INF}}|\beta_{\text{I}}^{\text{INF}}, -v^{\text{INF}})$$

$$\Rightarrow i^{\text{INF}} = -l^{\text{INF}}(1-\beta_{\text{I}}^{\text{INF}}(1-\varepsilon^{\text{INF}})) + c^{\text{INF}} + r^{\text{INF}}(1-\beta_{\text{I}}^{\text{INF}}(1-\varepsilon^{\text{INF}}))$$

$$\Rightarrow i^{\text{INF}} = c^{\text{INF}} - (l^{\text{INF}} - r^{\text{INF}})(1-\beta_{\text{E}}^{\text{INF}}). \tag{14}$$

Proof of Propositions 6 and 7

Given a valid belief pair $(\beta_I^{\text{SUP}}, \beta_E^{\text{INF}})$, SUP's expected rewards for the four verification strategies are defined as.

$$\begin{aligned}&\text{E}(R^{\text{SUP}}|\beta_{\text{E}}^{\text{INF}}, \beta_{\text{I}}^{\text{SUP}}, -v^{\text{SUP}}, -v^{\text{INF}})\\ &\quad = -l^{\text{SUP}}(1-\beta_I^{\text{SUP}}(1-\varepsilon^{\text{SUP}}+\theta)\beta_{\text{E}}^{\text{INF}}),\end{aligned} \tag{15}$$

$$\mathrm{E}(R^{\mathrm{SUP}}|\beta_E^{\mathrm{INF}},\beta_{\mathrm{I}}^{\mathrm{SUP}},-v^{\mathrm{SUP}},v_i^{\mathrm{INF}}) = -l^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta))-c^{\mathrm{INF}}+(l^{\mathrm{INF}}-r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}}), \quad (16)$$

$$\mathrm{E}(R^{\mathrm{SUP}}|\beta_{\mathrm{E}}^{\mathrm{INF}},\beta_I^{\mathrm{SUP}},v^{\mathrm{SUP}},-v^{\mathrm{INF}}) = -c^{\mathrm{SUP}}-r^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)\beta_{\mathrm{E}}^{\mathrm{INF}}), \text{ and} \quad (17)$$

$$\mathrm{E}(R^{\mathrm{SUP}}|\beta_E^{\mathrm{INF}},\beta_I^{\mathrm{SUP}},v^{\mathrm{SUP}},v_i^{\mathrm{INF}}) = -c^{\mathrm{SUP}}-r^{\mathrm{SUP}}\left(1-\beta_I^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)\right)-c^{\mathrm{INF}} + (l^{\mathrm{INF}}-r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}}). \quad (18)$$

From the two-firm model without incentives, we know that SUP will prefer the strategy $(-v^{\mathrm{SUP}},-v^{\mathrm{INF}})$ over $(v^{\mathrm{SUP}},-v^{\mathrm{INF}})$ when $\beta_{\mathrm{I}}^{\mathrm{SUP}}>\beta_*^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}})$. Furthermore, SUP will prefer the strategy $(-v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$ over $(v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$ when

$$\mathrm{E}(R^{\mathrm{SUP}}|\beta_{\mathrm{E}}^{\mathrm{INF}},\beta_I^{\mathrm{SUP}},-v^{\mathrm{SUP}},v_i^{\mathrm{INF}})>\mathrm{E}(R^{\mathrm{SUP}}|\beta_{\mathrm{E}}^{\mathrm{INF}},\beta_{\mathrm{I}}^{\mathrm{SUP}},v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$$

$$\Rightarrow -l^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta))>-c^{\mathrm{SUP}} - r^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta))$$

$$\Rightarrow \beta_{\mathrm{I}}^{\mathrm{SUP}}>\frac{1}{(1-\varepsilon^{\mathrm{SUP}}+\theta)}\left(\frac{c^{\mathrm{SUP}}}{l^{\mathrm{SUP}}-r^{\mathrm{SUP}}}\right)$$

$$\Rightarrow \beta_{\mathrm{I}}^{\mathrm{SUP}}>\beta_*^{\mathrm{SUP}}(1).$$

Since $\beta_*^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}})\geq\beta_*^{\mathrm{SUP}}(1)$, we know that for all valid belief pairs with $\beta_{\mathrm{I}}^{\mathrm{SUP}}<\beta_*^{\mathrm{SUP}}(1)$, the optimal two-firm strategy is either $(v^{\mathrm{SUP}},-v^{\mathrm{INF}})$ or $(v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$. Similarly, the optimal two-firm strategy is either $(v^{\mathrm{SUP}},-v^{\mathrm{INF}})$ or $(-v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$ for all valid belief pairs with $\beta_*^{\mathrm{SUP}}(1)<\beta_I^{\mathrm{SUP}}<\beta_*^{\mathrm{SUP}}(\beta_{\mathrm{I}}^{\mathrm{INF}})$, and the optimal two-firm strategy is either $(-v^{\mathrm{SUP}},-v^{\mathrm{INF}})$ or $(-v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$ for all valid belief pairs with $\beta_I^{\mathrm{SUP}}>\beta_*^{\mathrm{SUP}}(\beta_{\mathrm{I}}^{\mathrm{INF}})$.

For all valid belief pairs with $\beta_*^{\mathrm{SUP}}(1)<\beta_{\mathrm{I}}^{\mathrm{SUP}}<\beta_*^{\mathrm{SUP}}(\beta_{\mathrm{I}}^{\mathrm{INF}})$, SUP will prefer $(v^{\mathrm{SUP}},-v^{\mathrm{INF}})$ over $(-v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$ when $\mathrm{E}(R^{\mathrm{SUP}}|\beta_E^{\mathrm{INF}},\beta_I^{\mathrm{SUP}},v^{\mathrm{SUP}},-v^{\mathrm{INF}})>\mathrm{E}(R^{\mathrm{SUP}}|\beta_E^{\mathrm{INF}},\beta_I^{\mathrm{SUP}},-v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$

$$\Rightarrow -c^{\mathrm{SUP}}-r^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)\beta_{\mathrm{E}}^{\mathrm{INF}}) + l^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)) + c^{\mathrm{INF}}-(l^{\mathrm{INF}}-r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})>0$$

$$\Rightarrow \beta_{\mathrm{I}}^{\mathrm{SUP}}<\frac{(l^{\mathrm{SUP}}-r^{\mathrm{SUP}})+c^{\mathrm{INF}}-(l^{\mathrm{INF}}-r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})}{(1-\varepsilon^{\mathrm{SUP}}+\theta)(l^{\mathrm{SUP}}-r^{\mathrm{SUP}}\beta_{\mathrm{E}}^{\mathrm{INF}})} = \beta_{\diamond}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}}).$$

Thus, when $\beta_*^{\mathrm{SUP}}(1)<\beta_{\mathrm{I}}^{\mathrm{SUP}}<\beta_*^{\mathrm{SUP}}(\beta_{\mathrm{I}}^{\mathrm{INF}})$, SUP's optimal verification strategy is to verify the system design when $\beta_{\mathrm{I}}^{\mathrm{SUP}}<\min\{\beta_*^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}}),\beta_{\diamond}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}})\}$.

For all valid belief pairs with $\beta_{\mathrm{I}}^{\mathrm{SUP}}<\beta_*^{\mathrm{SUP}}(1)$, SUP will prefer $(v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$ over $(v^{\mathrm{SUP}},-v^{\mathrm{INF}})$ if $\mathrm{E}(R^{\mathrm{SUP}}|\beta_{\mathrm{E}}^{\mathrm{INF}},\beta_{\mathrm{I}}^{\mathrm{SUP}},v^{\mathrm{SUP}},v_i^{\mathrm{INF}})>\mathrm{E}(R^{\mathrm{SUP}}|\beta_{\mathrm{E}}^{\mathrm{INF}},\beta_{\mathrm{I}}^{\mathrm{SUP}},v^{\mathrm{SUP}},-v^{\mathrm{INF}})$

$$\Rightarrow -c^{\mathrm{SUP}}-r^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta))-c^{\mathrm{INF}} + (l^{\mathrm{INF}}-r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})+c^{\mathrm{SUP}} + r^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)\beta_{\mathrm{E}}^{\mathrm{INF}})>0$$

$$\Rightarrow \beta_{\mathrm{I}}^{\mathrm{SUP}}>\frac{c^{\mathrm{INF}}-(l^{\mathrm{INF}}-r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})}{r^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)(1-\beta_{\mathrm{E}}^{\mathrm{INF}})} = \beta_{\triangledown}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}}).$$

Similarly, for all valid belief pairs with $\beta_{\mathrm{I}}^{\mathrm{SUP}}>\beta_*^{\mathrm{SUP}}(\beta_{\mathrm{I}}^{\mathrm{INF}})$, SUP will prefer $(-v^{\mathrm{SUP}},v_i^{\mathrm{INF}})$ over $(-v^{\mathrm{SUP}},-v^{\mathrm{INF}})$ if $\mathrm{E}(R^{\mathrm{SUP}}|\beta_{\mathrm{E}}^{\mathrm{INF}},\beta_{\mathrm{I}}^{\mathrm{SUP}},-v^{\mathrm{SUP}},v_i^{\mathrm{INF}})>\mathrm{E}(R^{\mathrm{SUP}}|\beta_{\mathrm{E}}^{\mathrm{INF}},\beta_{\mathrm{I}}^{\mathrm{SUP}},-v^{\mathrm{SUP}},-v^{\mathrm{INF}})$

$$\Rightarrow -l^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta))-c^{\mathrm{INF}} + (l^{\mathrm{INF}}-r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})+l^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)\beta_{\mathrm{E}}^{\mathrm{INF}})>0 + l^{\mathrm{SUP}}(1-\beta_{\mathrm{I}}^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)\beta_{\mathrm{E}}^{\mathrm{INF}})>0$$

$$\Rightarrow \beta_{\mathrm{I}}^{\mathrm{SUP}}>\frac{c^{\mathrm{INF}}-(l^{\mathrm{INF}}-r^{\mathrm{INF}})(1-\beta_{\mathrm{E}}^{\mathrm{INF}})}{l^{\mathrm{SUP}}(1-\varepsilon^{\mathrm{SUP}}+\theta)(1-\beta_{\mathrm{E}}^{\mathrm{INF}})} = \beta_{\triangle}^{\mathrm{SUP}}(\beta_{\mathrm{E}}^{\mathrm{INF}}).$$

References

Ahmadi R, Wang RH (1999) Managing development risk in product design processes. Oper Res 47(2):235–246

Baiman S, Fischer PE, Rajan MV (2000) Information, contracting, and quality costs. Manage Sci 46(6):776–789

Balachandran KR, Radhakrishnan S (2005) Quality implications of warranties in a supply chain. Manage Sci 51(8):1266–1277

Barad M, Engel A (2006) Optimizing VVT strategies: a decomposition approach. J Oper Res Soc 57(8):965–974

Boumen R, de Jong IS, Vermunt J, van de Mortel-Fronczak J, Rooda J (2008a) Risk-based stopping criteria for test sequencing. IEEE Trans Syst Man Cybernet Part A Syst Humans 38(6):1363–1373

Boumen R, de Jong IS, Vermunt J, van de Mortel-Fronczak J, Rooda J (2008b) Test sequencing in complex manufacturing systems. IEEE Trans Syst Man Cybernet Part A Syst Humans 38(1):25–37

Boumen R, De Jong IS, Mestrom J, Van De Mortel-Fronczak J, Rooda J (2009) Integration and test sequencing for complex systems. IEEE Trans Syst Man Cybernet Part A Syst Humans 39(1):177–187

Ciucci F, Honda T, Yang MC (2012) An information-passing strategy for achieving Pareto optimality in the design of complex systems. Res Eng Design 23(1):71–83

Collopy P (2012) A research agenda for the coming renaissance in systems engineering. In: 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition

Dai Z, Scott MJ, Mourelatos ZP (2003) Incorporating epistemic uncertainty in robust design. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference

Eifler T, Engelhardt R, Mathias J, Kloberdanz H, Birkhofer H (2010) An assignment of methods to analyze uncertainty in different stages of the development process. In: ASME 2010 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers Digital Collection

Emons W (1988) Warranties, moral hazard, and the lemons problem. J Econ Theory 46(1):16–33
Engel A, Barad M (2003) A methodology for modeling VVT risks and costs. Syst Eng 6(3):135–151
Engel A, Last M (2007) Modeling software testing costs and risks using fuzzy logic paradigm. J Syst Softw 80(6):817–835
Huang H-Z, Zhang X (2009) Design optimization with discrete and continuous variables of aleatory and epistemic uncertainties. J Mech Design 131(3)
Kulkarni AU, Wernz C (2020) Optimal incentives for teams: a multiscale decision theory approach. Ann Oper Res 288:307–329
Kulkarni AU, Salado A, Wernz C, Xu P (2020) Is verifying frequently an optimal strategy? A belief-based model of verification. In: ASME 2020 International Design Engineering Technical Conference and Computers and Information in Engineering Conference (IDETC/CIE 2020), St. Louis, MO, (USA)
Laffont J-J, Martimort D (2009) The theory of incentives: the principal-agent model. Princeton University Press, USA
Lewis K, Mistree F (1997) Modeling interactions in multidisciplinary design: a game theoretic approach. AIAA J 35(8):1387–1392
Mihm J (2010) Incentives in new product development projects and the role of target costing. Manage Sci 56(8):1324–1344
Mihm J, Loch C, Huchzermeier A (2003) Problem-solving oscillations in complex engineering projects. Manage Sci 49(6):733–750
Nagano S (2008) Space systems verification program and management process: importance of implementing a distributed-verification program with standardized modular-management process. Syst Eng 11(1):27–38
Osborne MJ, Rubinstein A (1994) A course in game theory. MIT Press, Cambridge
Reyniers DJ (1992) Supplier-customer interaction in quality control. Ann Oper Res 34(1):307–330
Reyniers DJ, Tapiero CS (1995) Contract design and the control of quality in a conflictual environment. Eur J Oper Res 82(2):373–382
Salado A (2015) Defining better test strategies with tradespace exploration techniques and pareto fronts: application in an industrial project. Syst Eng 18(6):639–658
Salado A (2018) An elemental decomposition of systems engineering. In: 2018 IEEE International Systems Engineering Symposium (ISSE), IEEE
Salado A, Kannan H (2018) Properties of the utility of verification. In: IEEE International Symposium in Systems Engineering. Rome, Italy
Salado A, Kannan H (2019) Elemental patterns of verification strategies. Syst Eng 22(5):370–388
Salado A, Kannan H, Farkhondehmaal F (2018) Capturing the information dependencies of verification activities with bayesian networks. In: Conference on Systems Engineering Research (CSER). Charlottesville, VA, USA
Schlapp J, Oraiopoulos N, Mak V (2015) Resource allocation decisions under imperfect evaluation and organizational dynamics. Manage Sci 61(9):2139–2159
Schlosser J, Paredis CJ (2007) Managing multiple sources of epistemic uncertainty in engineering decision making. SAE Trans 1340–1352
Schumacher G, Schlapp J (2017) Delegated testing of design alternatives: the role of incentives and testing strategy. https://ssrn.com/abstract=3020025. Accessed 15 Jan 2019
Sentz K, Ferson S (2002) Combination of evidence in Dempster–Shafer theory, Citeseer
Shabi J, Reich Y (2012) Developing an analytical model for planning systems verification, validation and testing processes. Adv Eng Inform 26(2):429–438
Shabi J, Reich Y, Diamant R (2017) Planning the verification, validation, and testing process: a case study demonstrating a decision support model. J Eng Des 28(3):171–204
Sommer SC, Loch CH (2009) Incentive contracts in projects with unforeseeable uncertainty. Prod Oper Manage 18(2):185–196
Starbird SA (2001) Penalties, rewards, and inspection: provisions for quality in supply chain contracts. J Oper Res Soc 52(1):109–115
Tahera K, Earl C, Eckert C (2017) A method for improving overlapping of testing and design. IEEE Trans Eng Manage 64(2):179–192
Tahera K, Wynn DC, Earl C, Eckert CM (2019) Testing in the incremental design and development of complex products. Res Eng Design 30(2):291–316
Terwiesch C, Loch CH, Meyer AD (2002) Exchanging preliminary information in concurrent engineering: alternative coordination strategies. Organ Sci 13(4):402–419
Vermillion SD, Malak RJ (2018) A game theoretical perspective on incentivizing collaboration in system design. Disciplinary convergence in systems engineering research, Springer, Berlin, pp. 845–855
Vermillion SD, Malak RJ (2018) A theoretical look at the impact of incentives on design problem effort provision. In: ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection
Walden DD, Roedler GJ, Forsberg K, Hamelin RD, Shortell TM (2015) Systems engineering handbook: a guide for system life cycle processes and activities. Wiley, USA
Wernz C, Deshmukh A (2010) Multiscale decision-making: bridging organizational scales in systems with distributed decision-makers. Eur J Oper Res 202(3):828–840
Wernz C, Deshmukh A (2012) Unifying temporal and organizational scales in multiscale decision-making. Eur J Oper Res 223(3):739–751
Wynn DC, Grebici K, Clarkson PJ (2011) Modelling the evolution of uncertainty levels during design. Intern J Interact Design Manuf (IJIDeM) 5(3):187
Xiao A, Zeng S, Allen JK, Rosen DW, Mistree F (2005) Collaborative multidisciplinary decision making using game theory and design capability indices. Res Eng Design 16(1–2):57–72
Xu P, Salado A (2019) A concept for set-based design of verification strategies. INCOSE Int Symp 29(1):356-370. https://doi.org/10.1002/j.2334-5837.2019.00608.x
Yamada S, Ichimori T, Nishiwaki M (1995) Optimal allocation policies for testing-resource based on a software reliability growth model. Mathe Comput Model 22(10–12):295–301
Zhu K, Zhang RQ, Tsung F (2007) Pushing quality improvement along supply chains. Manage Sci 53(3):421–436