Identifying Risks and Risk-Mitigation Strategies for Collaborative Systems during Requirements Engineering: A Goal-Oriented Approach

Kirthy Kolluri, Lawrence Chung
Department of Computer Science
The University of Texas at Dallas
Richardson, TX
{kirthy.kolluri, chung}@utdallas.edu

Tom Hill
Fellows Consulting Group
Dallas, TX
{tom}@fellowsconsultinggroup.com

***Abstract*— If risks are not identified, they are unlikely to be addressed, possibly resulting in undesirable consequences, such as fatal accidents or even loss of human lives. Furthermore, risks should be considered not only in terms of system behavior but also events occurring in the system environment – i.e., in a collaborative setting in which the system and its environment work together towards certain goals the system is intended to help achieve. In this paper, we present a goal-oriented risk analysis framework, *Murphy+G*, in which non-functional requirements (hereafter, *NFRs*) are treated as softgoals to be achieved and systematically addressed in terms of both the system and its environment during requirements engineering, by adopting what is called the *Reference Model.* A study of a smartphone app, *Theia,* which is intended to help blind people navigate indoors, is used for the purpose of both illustration and experimentation. In this study, NFRs (e.g., safety, reliability, timeliness, etc.) are treated as softgoals, and risks (e.g., fall down, injury, etc.) are identified, along with risk-mitigation strategies for both the system and its environment, with the help of an activity-oriented ontology. To see both the strengths and weaknesses of Murphy+G, a systematic methodology for risk analysis for collaborative systems, a controlled experiment has been carried out, in terms of three different versions of Theia implementations. Feedback from students show improvements on the accuracy of the risk analysis and the risk mitigation strategies devised, as well as enhanced users' experience with respect to increased confidence in navigating indoors, in a safe, timely, and reliable manner.**

Keywords- risks, risk analysis, risk-mitigation, non-functional requirements (NFRs), softgoals, Reference Model (WRSPM Model), ontology

I. INTRODUCTION

The statement "A Risk unidentified is a Risk unaddressed" expresses the importance of risk identification, analysis and addressing the risks that are identified. Risk, which is defined as a situation or event where something of human value (including humans themselves) can be put at stake and where the outcome is uncertain"[14], is a phenomenon faced or caused by a person. In general, each action performed may have one to many risks associated with them which can range from a normal risk (e.g., missing route) to a serious risk (e.g., loss of life). The usage of smartphone apps (Collaborative Systems) has been on the rise and studies show that people spend an average of 3 hours and 10 [1]minutes using their phone out of which 2 hours and 51 minutes is being spent on smartphone apps, hence many people are inclined towards using smartphone apps for day-to-day activities [20]. The Reference Model emphasizes that the requirements are satisfied by the collaboration between the user and the events in its environment, hence the term collaborative system. A previous study describes the importance of comprehensively identifying the key NFRs (Non-Functional Requirements) and appropriately handling them [10]. For example, with apps such as Theia, a smartphone app for helping blind people navigate indoors, there is always chance for the user (blind person) getting hurt, if risks, which are related to user goals, e.g., safety, reliability, timeliness, etc. (the so-called NFR softgoals [11]) are not identified or not appropriately handled. Lack of systematic methodologies to identify the most important NFR softgoals related to a requirement, the risks associated with them, and ways to mitigate them can lead to mishaps and undesirable things from happening.

We extend our previous research, the Murphy framework [12] [13], with goal-orientation. This new framework, the *Murphy+G* supports goal-oriented risk identification and analysis for collaborative systems. Two technical contributions have been made in this paper, including 1) a goal-oriented, activity-driven ontology for the Murphy+G framework which captures the most important goals, actions and risks related to the smartphone app (*Theia*), and 2) a technique to combine the Reference Model with the NFR framework for identifying the most important ways to achieve NFR softgoals (called *operationalizing softgoals*) in the context of the Reference Model, where requirements are met in terms of specification and domain, and choosing the most feasible risk-mitigation strategies.

The proposed approach is illustrated using a smartphone application, *Theia*, for helping blind people navigate indoors. A scenario using Theia, for helping blind people navigate indoors is used as the running example all through this paper for easy understandability. Stevie is a blind student who wants to attend a class in room 3.415. He uses a smartphone application to

DOI reference number: 10.18293/SEKE2023-227

navigate from his current location to his class, room 3.415. He uses voice instruction to provide his destination to Theia. To navigate to his destination Stevie must walk 10 steps forward and turn right but we will elaborate only on the walking 10 steps forward part for performing goal-oriented risk analysis. The rest of this paper is structured as follows. Section 2 discusses related work and adopted models for this work. Section 3 presents the Murphy+G framework. Section 4 describes the experiments conducted and their results. Section 5 discusses our observations and threats to validity. Finally, Section 6 summarizes the paper and future work.

II. Related work and adopted models

We believe that this paper is one of the first to extend the NFR Framework with the Reference Model and an activity-oriented ontology, which corresponds to it. There has been some work on goal-oriented risk analysis (e.g., see [1, 3, 6, 7, 11]) in literature. Our work is similar especially to [1] [8], in the sense that it provides a qualitative goal-oriented risk analysis, with an ontology, but, among other things, without adopting the Reference Model or an activity-oriented ontology.

For this work we have adopted various models for representing functional requirements, specification and domain assumptions, non-functional requirements (softgoals), and models to represent risks for which risk-mitigation strategies are chosen to help avoid those risks.

A. For Requirements, Specification and Domain Assumptions:

A picture of the *Reference Model* is shown in fig. 1, which consists of Environment (E) and System (S). The environment consists of Domain Assumptions (W or D) and Requirements (R) and System consists of Program (P) and Machine (M) and the Specification (S) is present in the intersection of E and S. The environment has environment events e_{h} and e_v, and the system has system events s_{h} and s_v (hidden (h) and visible (v) events respectively). The user is a designator with an intention, designated to perform any kind of actions in the environment. The Reference Model [4, 5] emphasizes that the user requirements are satisfied not by the system alone but also by the system's collaboration with the events in its environment. Hence, we use the term collaborative system for those kinds of systems.

B. For Non-Functional Requirements:

For representing NFRs, there are several goal-oriented frameworks, including KAOS [15], i* [16], and the NFR framework [11], each with its own significance, quality, and features. Adopting the NFR Framework [11] and other similar works, we treat NFRs, such as safety, reliability, timeliness, etc., as softgoals, which convey the sense that they typically have no clear-cut definition or absolute criteria to determine their satisfaction. Hence, instead of logical "satisfy", we use "satisfice" with different degrees of contribution. Each softgoal can be either AND or OR decomposed into sub-softgoals or contribute towards satisficing another softgoal either fully or partially positively (MAKE or HELP), or fully or partially negatively (BREAK or HURT). A label propagation mechanism evaluates the effect of a decision on upper softgoals, with a label - Satisficed, Denied, Conflict, or Undetermined. Softgoals and relationships between softgoals are visualized by using Softgoal Interdependency Graph (SIG) (see Fig. 3 for examples).

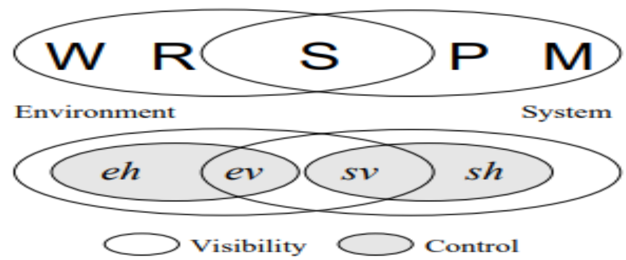

Fig. 1: Reference Model showing Domain (D or W), Requirement (R), Specification (S), Program (P) and Machine (M) along with their events

Operationalizing softgoals are concrete functional requirements (FRs) (risk-mitigation strategies in this paper), which are implemented as features in the projected software system to address various risks (e.g., In Theia, to address the risk "Incorrect number of steps taken", risk-mitigation strategies could be any features that provide help with counting steps), which can satisfice the NFR softgoals (e.g., user goals such as safety, reliability, etc.).

C. For Risks:

There are various models for problem (risk in this paper) representation and root cause analysis, including Fish Bone Diagram [22], Problem Interdependency Graph (PIG) [19], Fault Tree Analysis (FTA) [18], Cause Effect Graph [20], Failure Mode and Effect Analysis (FMEA)[17]. While FTA, is used when information is available about AND/OR logical relationships among root causes, Fish Bone Diagram is used when the problem is about relationships among root causes. Causal Mechanism graph help design better assessment models and risk control approaches since causal mechanisms capture the causal entities that determine software dependability, while FMEA reviews components and subsystems to identify potential failure modes in a system and their causes and effects, while providing a scale to understand the likelihood of failure, while our risk analysis uses an activity-oriented ontology to systematically identify risks.

We adopt and use PIG, since it not only resembles a SIG, but also since problems (potential problems) are similar to risks. The problems (risks in this work) and relationships between problems are represented using a Problem Interdependency graph. Problems represent potential problems until they are realized using label propagation, hence we consider them as risks for this work.

III. Murphy+G: A Goal-oriented framework for utilizing goals and reference model for risk analysis

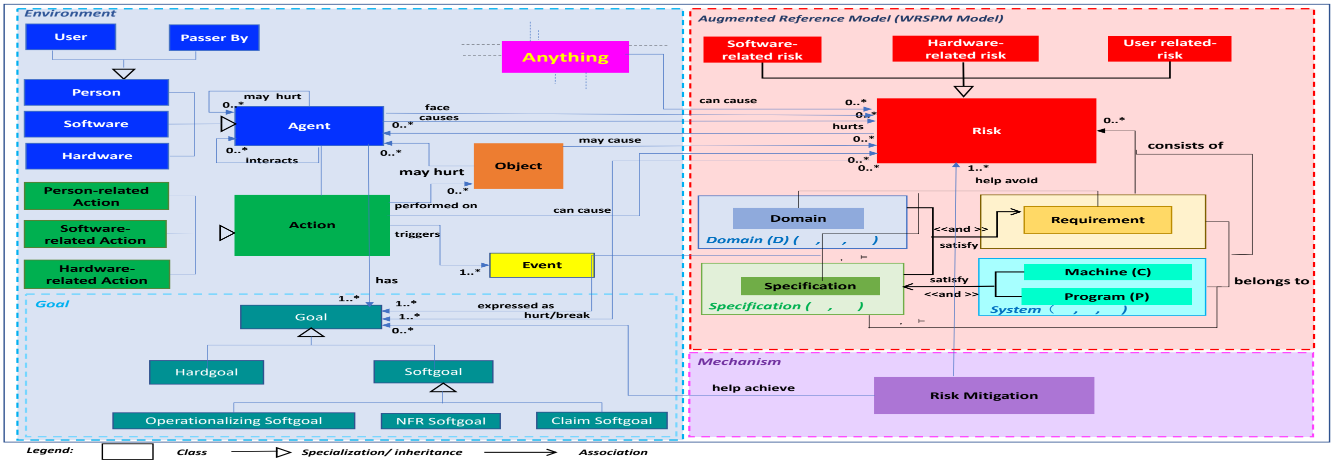

Fig. 2: Ontology of the Murphy+G Framework

Murphy+G framework offers an activity-oriented ontology. The framework then introduces goals related to requirements by extending NFR framework with the Reference Model. Additionally, the framework offers (SIG-PIG)+Reference Model (RM) Graph which represents the process graphically, thereby bringing in the notion of Reference Model to SIG-PIG to analyze and identify Specification and Domain Assumptions to satisfice the Requirement (FR).

A. Ontology of the Murphy+G Framework:

Fig. 2 shows a high-level domain independent ontology which includes some concepts that we have discussed in section II. *Ontology* is meant to refer to the categories of essential individual concepts, relationship between the individual concepts and constraints on individual concepts and on the relationships between individual concepts, as in [9]. It depicts salient concepts such as actions, agents, risks, goals, concepts related to the Reference Model (requirements, specification, etc.).

An agent is someone/something that has the capability to perform an action, oftentimes with some intention (goal, in this paper), which might be attributed to by another agent. Our ontology recognizes people, software and hardware as three types of agents. As for goals, we consider softgoals; softgoals can be an operationalizing softgoal (a feature that can be implemented in the system to be developed; e.g., screen tapping mechanism – where the user taps the screen for every step taken, etc.). Risks are potential problems that the agent (Person) may face, and a risk hinders the achievement of a goal. Hence, we come up with risk mitigation strategies which help achieve the user's goals by alleviating some risks faced by the user.

B. Process offered by the Murphy+g Framework:

The Murphy+G process consists of 5 steps:

Step 1: **Capture most important NFR Softgoals.** In this step, most important users' goals are explicitly captured as softgoals from the stakeholders by using a questionnaire. The questionnaire is not shown here due to space limitation. Based on the answers provided by the users' of the app, the most important NFRs (Safety, Timeliness, Reliability, etc.) for this app are captured.

Since most of the users' (e.g., blind person) express their goals as being safe (**Safety**), have an application and smartphone that they can rely on (**Reliability**) (e.g., smartphone app, smartphone, etc.) and being able to reach their destination on time (**Timeliness**), we have capture these softgoals as the most important at the highest level for this work. These softgoals are represented using a cloud symbol, as shown in step 1 in fig. 3.

Step 2: Decompose softgoals into sub-softgoals and identify risks associated with the softgoals. As discussed in step 1, Safety, Reliability and Timeliness are captured as the most important softgoals at the highest level. Due to space restriction, we will discuss only safety softgoal here. Safety is the type of the softgoal, and User is the topic as discussed in [11]. A softgoal can be decomposed either using type or topic. Firstly, Safety[User] is refined using an AND decomposition by topic, i.e, Safety[User] is decomposed into Safety[Blind Person] and Safety[Passer By] sub-goals, where the former is further decomposed by type into Attentiveness[Blind Person] and Committing Less Errors[Blind Person].

Step 3: Identify potential problems associated with sub-softgoals. For the low-level sub-softgoals identified in Step 2, which are Attentiveness[Blind Person] and Committing Less Errors[Blind Person], risks that could impede with achieving the goals are identified in step 3. The risks relevant to the goals are identified, which may be further decomposed into sub-risks (if applicable), until they are at a basic level. To address these identified risks, some functional features (operationalizations), could be implemented in the projected software system, to help avoid the identified risks, as shown in step 3 in fig. 3. In this example, ***Wrong Action performed*** could break the *Committing Less Errors[Blind Person]* softgoal. ***Wrong Action Performed*** risk is further OR decomposed into ***Walk Incorrect Number of***

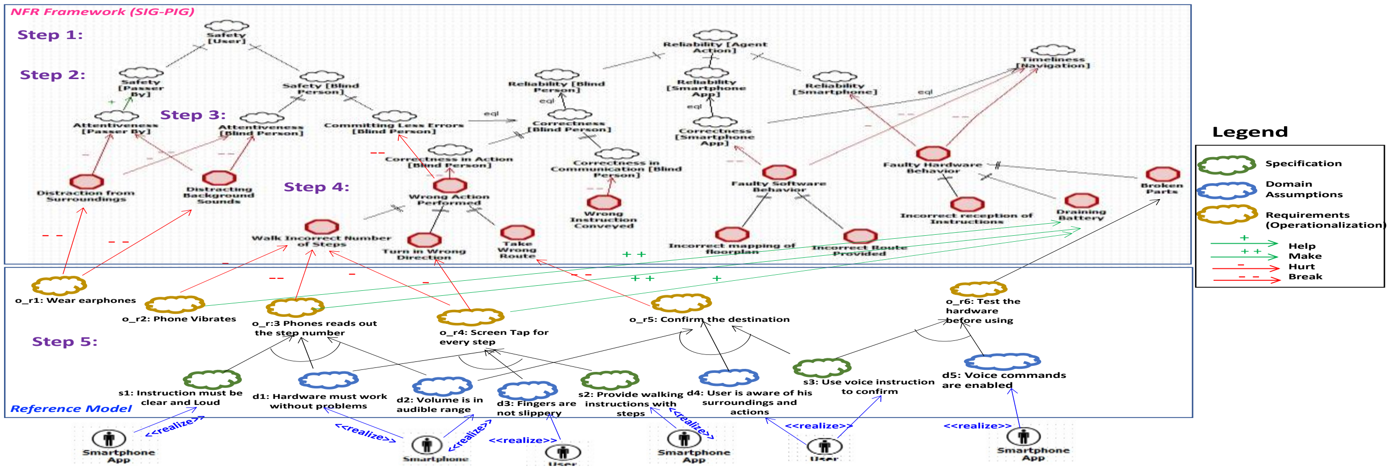

Fig. 3: An example showing (SIG-PIG)+Reference Model (RM)

Steps, ***Turn in Wrong Direction*** or ***Take wrong Route*** **sub-risks**.

Step 4: Find operationalizing softgoals (FRs) for the risks. This step identifies the risk-mitigation mechanisms, which are the operationalizing softgoals or the functional requirements that may be implemented in the projected system.

For each of the leaf risks, an operationalization represented by the yellow cloud as shown in fig.3, helps alleviate that particular risk. As discussed in the example in step 3, we OR refine ***Wrong Action Performed*** into ***Walk Incorrect Number of Steps***, ***Turn in Wrong Direction*** or ***Take wrong Route***. Considering Walk Incorrect Number of Steps, **o_r**1*: Phone Vibration*, **o_r2:** *Phone reading out the step number* and **o_r3:** *Screen Tap for every step taken* can be some risk mitigation strategies that can help avoid "Walking wrong number of steps" risk. All these risk-mitigation strategies either hurt or break the risks, which satisfies the risk-mitigation strategy.

Step 5: Identify corresponding specification and domain assumptions. In this step, we bring in the specification and Domain assumptions that satisfy the requirements**.** The Reference Model, states that the specification (S) and the domain assumptions (D) together satisfy the requirement [4, 5].

$$S, D \vDash R$$

Hence, every requirement must have a S and D that satisfies it. The requirement (risk-mitigation strategy) must be satisfied by S, D to help alleviate the risk. For a requirement to be satisfied, there must be some domain assumptions that must hold, and some system specification that needs to be implemented for a requirement (operationalization/ risk-mitigation strategy) to be satisfied. Using the example in Fig. 3, and as discussed in step 2.3, let us consider the risk-mitigation strategy **o_r2:** *Phone reading out the step number*, the domain assumptions **d1** is that the hardware must work without any problems and **d2** is that the volume is in audible range for the user to hear. The corresponding specification, **s1** is that the instruction must be loud and clear

After identifying specification and domain for the all the operationalizing softgoals, the best solution which helps avoid that particular risk must be selected to be implemented. For that, we perform a trade-off analysis to explore among alternatives to select the option that helps alleviate the problem.

C. Perform trade-off analysis for risk mitigation strategies:

In this step, all the risk mitigation strategies identified are considered and a trade-off analysis is performed to explore among alternatives. These risk-mitigation strategies are analyzed using various important NFR softgoals, and the operationalization/risk-mitigation strategy that helps avoid the risk in a better and an efficient way is selected during the requirements engineering phase and is implemented as a feature in the projected software system. For example, in Table. 1, we have operationalizing softgoals on one side of the table and softgoals on the other.

If we consider "**Phone reads out step number**" feature, it is given a contribution of ++ since the blind person is not involved in performing any action, hence the chance of a risk happening is less when compared to "**screen tap for every step taken**" which was given a +**,** which means that this feature is helpful partly (sometimes). If the user must tap the screen for every step taken, there may be a chance for some risks to arise such as blind person forgetting to tap the screen, screen tap not being recorded, etc.

Similarly, for the "Phone vibrating for every step taken", there may be some risks such as the blind person may walk very fast, or the smartphone app may not be able to detect the steps, or the blind person may not be able to recognize the vibration, etc. Hence, with regards to Safety softgoal, **Phone reading out the** *step number is the best risk-mitigation strategy that could be implemented. In a similar fashion, trade-off analysis was performed by taking all other softgoals into consideration.* We have identified that *"phone reading out the step number"* while the person is walking as the best risk-mitigation strategy to be implemented to address the risk *"Walking incorrect number of steps"*.

Validation Tools: Murphy+G Assistant has been implemented to validate the risk identification and analysis process and to devise risk-mitigation strategies. We have adopted the Murphy Assistant tool from our previous work [12, 13] and extended it

Softgoal / Operationalization/ Risk-Mitigation Strategy	Safety (User)	Timeliness	Reliability	Easy to use
Screen tap for every step taken	+	-	-	- -
Phone reads out step number	+ +	+ +	+	+ +
Phone Vibrates for every step	+	+	-	+

TABLE 1: An example table showing the trade-off analysis for selecting the best risk-mitigation strategy

with the concept of goals, which is a semi-automated risk analysis tool, where the user of the application has to setup the ontology, provide a requirement and the most important softgoals before performing risk identification and analysis. Fig. 4 shows an example snapshot of the validation tools in action. For this process, we developed a windows application using the .NET framework. For storing all ontological concepts entered by the user, a Microsoft SQL Server Local Database is used. Murphy Assistant is a prototype tool which supports the concepts of Murphy framework. We also developed two versions of Theia, with the risk-mitigation strategies devised using the results from the Murphy+G Assistant. The underlying code and the snapshots of the tool in action can be found at

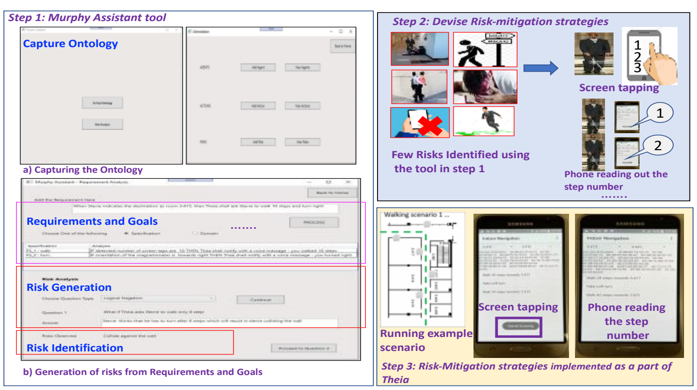

Fig. 4: Snapshots of the prototypes used - Murphy+G Assistant, and three versions of Theia implemented (showing two versions here)

https://github.com/indoornavigation0/Murphy.git.

IV. EXPERIMENTATION AND VALIDATION

The authors of the paper has conducted various controlled experiments, while also involving feedback from students (more on this in the next section), to see the strengths and weaknesses of the Murphy+G framework, using the results from the Murphy+G tool, that the authors experimented with. The risk-mitigation strategies from the tool have been considered and three different implemented versions of the smartphone app (Theia) were used for experimentation.

Experimental setup:

We have developed three versions of Theia aditionally, apart from one version 1 in which, smartphone "**reads the step number**" as the user walks. In version 2, we have implemented the second risk-mitigation strategy devised for a risk in our running example, "**screen tapping"**, where the user taps the screen for every step taken. In version 3, "**phone vibrating"** for every step taken is implemented. These three versions of Theia are tested with approximately 60 students (20 students for each version).

For the experimentation, undergraduate, masters-level and Ph.D.-level students, aged around 18-40, were involved as subjects, who were blind-folded and asked to use Theia for performing basic activities such as speaking out his/her current location, choosing a destination, walking forward, stopping at a specific point, turning at the right place in the right direction as and when indicated by the application. The experimentation took approximately 1 hour in total for each participant for monitoring the usage of app and gathering the feedback from the participant, concerning the general feelings/comments, the risk-mitigation features implemented, the weaknesses/limitations and recommendations. Two students were present (one ahead and one behind) all the time within a proximity to the participant to ensure the safety of the student while walking.

Fig. 5 explains the results of the experience of the students using each of the risk mitigation strategies for addressing the risk "***Incorrect number of steps walked***". When students tested the feature of phone vibration, for each step taken, there were both good and bad observations. The bad observations were that some students were partly able to follow the vibration given by the phone for every step taken. While using this feature, a few students were paying more attention to confirm if the phone has given a vibration or not. In that confusion, they missed route, walked in zig-zag pattern, walked too fast sometimes, stopped walking, and just moved hands, etc. which resulted in some good number of risks. Similarly, some students who focused on the feature well were able to use the most part of feature without committing any mistakes, hence alleviating some risks.

There were both positive and negative observations with the students that tested screen tapping for every step taken to keep track of number of steps. Students were more confident when they used this feature. However, there were some risks that were faced by the user as a result of their actions (e.g., not paying attention to the step number, etc.). Some users did not tap the screen accurately, some tapped the screen twice for one step taken, and some did not tap in the right

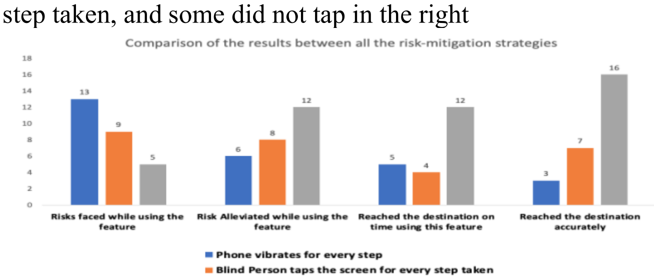

Fig. 5: Graph comparing the results obtained using three versions of the risk mitigation strategies implemented in Theia

spot on the screen which resulted in the step not being counted or being counted more than once respectively, which resulted in walking incorrect number of steps forward and making incorrect turns. Users who walked very slow to tap the screen for every step taken took more navigation time to reach their destination.

For the “smartphone reading the step number” for every step taken, the students were much more comfortable and confident using this feature compared to the other two. Since the steps taken were read out loud, most of the students had their focus on the steps they were taking. There were a smaller number of errors committed, hence a smaller number of risks. When the background noise was too loud, some students were not able to follow the step number and missed the route, walked in a zig-zag pattern trying to pay attention to the volume. Some students used old phones whose volume was not clearly audible. Those students were more comfortable using the screen tapping feature.

Overall, based on our observation, the students committed a smaller number of risks and alleviated a greater number of risks when the phone read the step numbers for the users. Most of them were able to reach their destination in a timely manner. The only issues with this feature were some hiccups with the hardware and hardware behavior.

Threats to Validity: The results from the risk mitigation strategies depends on the individuals (students) who participated in the experiment using three versions of Theia. The experience of using the Theia app with various risk-mitigation strategies also varies from individual to individual. We are yet to receive our IRB approval to test our smartphone application with real blind people. We feel that testing with real blind people may give us an edge over blind-folded people, especially with identifying a variety of risks they face and while using the app overall.

V. Conclusion

In this paper, we have proposed a goal-oriented framework, Murphy+G, which extends the NFR Framework with the Reference Model for identifying risks and risk-mitigation strategies for collaborative systems. More specifically, this paper has presented 1) an ontology which comprises of key concepts such as action, risk, agent, etc.; 2) Capture most important NFR softgoals by using a questionnaire, 3) Identify risks associated with the softgoals while devising risk-mitigation strategies, 4) Extend the NFR Framework with the Reference Model and 5) Perform trade-off analysis to identify the best solution to help alleviate those risks. We used three versions of Theia to see the strengths and weakness of the Murphy+G framework, and we have observed that the features implemented as risk mitigation strategies have indeed increased the confidence of the users to reach their destination in a safe, timely and reliable manner. As future work, we plan to apply our approach to a variety of domains which involve collaboration between the user and the software system (e.g., autonomous vehicles domain) for performing risk analysis and providing risk mitigation strategies. A graphical tool for risk analysis is also underway. Experimentation and validation of the risk mitigation strategies and different versions of Theia developed must be tested with real blind people upon the IRB approval as real blind people can give the authors much better insights of what changes must be made to make the app help meet blind people’s goals.

References

[1] Asnar, Y., Giorgini, P. & Mylopoulos, J. Goal-driven risk assessment in requirements engineering. Requirements Eng **16**, 101–116 (2011). https://doi.org/10.1007/s00766-010-0112-x

[2] https://en.wikipedia.org/wiki/Murphy%27s_law, last edited on 12 February,2023

[3] A. Cailliau and A. van Lamsweerde, "A probabilistic framework for goal-oriented risk analysis," 2012 20th IEEE International Requirements Engineering Conference (RE), Chicago, IL, USA, 2012, pp. 201-210, doi: 10.1109/RE.2012.6345805.

[4] C. A. Gunter, E. L. Gunter, M. Jackson and P. Zave, "A reference model requirements and specifications," in IEEE Software, vol. 17, no. 3, pp. 37-43, May-June 2000, doi: 10.1109/52.896248.

[5] Zave, P., Jackson, M. (1997). Four dark corners of requirements engineering. ACM Trans. Softw. Eng. Methodol., 6, 1-30. DOI:https://doi.org/10.1145/237432.237434

[6] Lamsweerde, A.V.. (2013). Risk-driven engineering of requirements for dependable systems. 10.3233/978-1-61499-207-3-207.

[7] Cailliau, A., van Lamsweerde, A. Assessing requirements-related risks through probabilistic goals and obstacles. Requirements Eng **18**, 129–146 (2013). https://doi.org/10.1007/s00766-013-0168-5

[8] Sales, T.P., Baião, F., Guizzardi, G., Almeida, J.P.A., Guarino, N., Mylopoulos, J. (2018). The Common Ontology of Value and Risk. In: , et al. Conceptual Modeling. ER 2018. Lecture Notes in Computer Science(), vol 11157. Springer, Cham. https://doi.org/10.1007/978-3-030-00847-5_11

[9] Greenspan, S.J., Mylopoulos, J., Borgida, A.. (1994). On formal requirements modeling languages: RML revisited. 135-147. 10.1109/ICSE.1994.296773.

[10] Mehta, R., Wang, H., Chung, L. (2012). Dealing with NFRs for Smart-Phone Applications: A Goal-Oriented Approach. In: Lee, R. (eds) Software Engineering Research, Management and Applications 2012. Studies in Computational Intelligence, vol 430. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30460-6_8

[11] Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J. (2000). The NFR Framework in Action. In: Non-Functional Requirements in Software Engineering. International Series in Software Engineering, vol 5. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-5269-7_2

[12] Kolluri, K., Ahn, R., Hill, T., & Chung, L. (2021). Risk Analysis for Collaborative Systems during Requirements Engineering. In Proc., International Conference on Software Engineering & Knowledge Engineering (SEKE 2021) (pp. 297-302).

[13] Kolluri, K., Ahn, R., Rauer, J., Chung, L., & Hill, T. (2022, July). Identifying Risks for Collaborative Systems during Requirements Engineering: An Ontology-Based Approach. In International Conferences on Software Engineering and Knowledge Engineering (Vol. 2022).

[14] Rosa, E., “Metatheoretical foundations for post-normal risk." Journal of Risk Research 1 (1998): 15-44.

[15] A. van Lamsweerde, "Requirements Engineering in the Year 00: A Research Perspective," Proc., 22nd Int. Conf. on Soft. Eng., 2000. pp. 1-15

[16] E. Yu, P. Giorgini, N. Maiden and J. Mylopoulos, Social Modeling for Reqs. Eng., The MIT Press. 2011.

[17] Stamatis, Diomidis H. Failure mode and effect analysis: FMEA from theory to execution. Quality Press, 2003.

[18] L. Xing and S. V. Amari, "Fault Tree Analysis," Handbook of Performability Engineering, Springer London, 2008.pp.595-620

[19] Supakkul, S., Chung, L.: Extending problem frames to deal with stake Holder problems: An agent-and goal-oriented approach. In: Proceedings of the 2009 ACM symposium on Applied Computing. (2009) 389-394

[20] https://www.pewresearch.org/internet/fact-sheet/mobile/