Accepted to the 50th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA'24).

Not All Conflicts Are the Same: An Empirical Study of Requirement Conflicts in Practice

Antonia Welzel
Dept. of Computer Science and Eng.
Chalmers | University of Gothenburg
Gothenburg, Sweden
welzel@chalmers.se

Rebekka Wohlrab
Dept. of Computer Science and Eng.
Chalmers | University of Gothenburg
Gothenburg, Sweden
wohlrab@chalmers.se

Richard Berntsson Svensson
Dept. of Computer Science and Eng.
Chalmers | University of Gothenburg
Gothenburg, Sweden
richard@cse.gu.se

***Abstract*—Requirement conflicts commonly occur in software development, especially for complex systems that involve many requirements. Resolving these conflicts can be very time-consuming and costly. Moreover, due to the contextual nature of requirements conflicts, resolution strategies are needed that can be tailored to a specific conflict and its context. Currently, there is a lack of research on what categories of conflicts exist and how practitioners manage these conflicts. To enable more adaptive resolution strategies, the aim of this research is to map what types of requirement conflicts are encountered and how they are managed in practice. Through an interview study with eleven participants from eight companies in six domains, we identified three levels of conflict types that connect to four types of causes. These types and causes revealed two main dimensions that impact conflict management. One dimension is related to the nature of the conflict, either technical or social, and the other dimension is related to the nature of the requirement scope, which is either too constraining or too undefined resulting in stakeholders making assumptions that create conflicts. We found that these two dimensions impact what conflict resolution strategies are most commonly used for different types of conflicts.**

***Index Terms*—Requirement Conflict, Conflict Types, Conflict Causes, Conflict Resolution, Interview Study**

I. Introduction

Systems today need to fulfil several different quality attributes and requirements to satisfy the needs of stakeholders as well as different system contexts [1], [2]. Conflicts between requirements then arise when the requirements are not compatible or contradictory [3], making it impossible to find a solution in such a way that all requirements are achieved [4].

Different aspects of requirement conflicts and their nature have been identified to develop heuristics for conflict management and support rule-based approaches [5]–[7]. However, while these methods can be effective for managing conflicts, they generally do not consider the contextual factors around requirement conflicts such as the cause of the conflict or the dynamic nature of the system and business environment, which can increase the complexity of requirement conflicts [8]. Moreover, there are few empirical studies focusing on the different factors of conflicts and conflict management in practice. One prominent example is an empirical study conducted by Hadar *et al.* [9] where they interviewed different companies to understand the challenges and attitudes towards requirement inconsistencies in practice. They found that there are inconsistencies between the strategies suggested in theory for managing inconsistencies and what is applied in practice. Therefore, more practice-based evidence is needed for effective conflict management strategies.

In order to account for the various factors and circumstances affecting requirement conflicts, conflict management strategies need to be able to adapt to be effective in different contexts. However, developing these strategies requires a better understanding of the attributes of conflicting requirements in different contexts and their impact on managing conflicts. These insights would extend the understanding of what aspects are important to manage conflicts between requirements. Therefore, to gain in-depth knowledge of the types of requirement conflicts experienced in practice, we conducted an interview study with ten semi-structured interviews. The research setting consisted of eight companies and six different domains. We provide empirical insights into the types of conflicts and how they are typically resolved. We identified two dimensions to requirement conflicts according to their causes which impact what strategies are typically used to resolve the conflict. One dimension for categorising conflicts and their management strategies pertains to the requirement scope either being too constrained or too undefined. The second dimension relates to the nature of the conflict, which is either more technical relating to the software internally or social if it is external to the software.

The remainder of the paper is structured as follows: Sect. II discusses related work. Sect. III presents the research methodology. In Sect. IV, the findings of the study are described and further discussed in Sect. V. Lastly, conclusions and suggestions for future research are given.

II. Related Work

Multiple attempts have been made to categorise requirement conflicts to help with the identification and resolution of conflicts, such as through heuristics and models [10]–[12]. The types of conflicts have been based on, for instance, the nature of the requirements in conflict. Butt *et al.* [7] type requirement conflicts as either 'mandatory', 'essential', or 'optional' depending on the level of importance of the requirements in conflict. Conflicts between NFRs have also been categorised

as 'absolute', 'relative', or 'never' conflicts [13]. Furthermore, the relationships between imprecise requirements have been classified as either conflicting, cooperative, mutually exclusive, or irrelevant to develop methods for conflict detection, resolution, and impact assessment [14].

Aside from the nature of the requirements in conflict, requirement conflicts have also been categorised based on the semantics of the requirements. For example, Easterbrook [15] distinguishes between (i) conflicting interpretations, i.e., discrepancies in the perceptions of what the current requirements are, (ii) conflicting designs, i.e., discrepancies about how the system should be, or (iii) conflicting terminology, i.e., discrepancies in the terms used to describe the system and requirements. These conflicts can then be not-interfering, partially-interfering, and mutually exclusive depending on their severity. Moreover, Kim *et al.* [16] define two types of requirement conflicts, activity and resource conflicts, based on the conflict cause. They describe the requirement's authoring structure as 'Action (Verb) + Object (Object) + Resource (Resource)' and the conflict type is subsequently based on whether a conflict arose due to a misalignment in one of these three aspects, such as requirements having opposite actions involving the same object, same action involving different objects, or relying on the same resource which is limited to some requirements.

The existing categorisations provide frameworks for classifying requirement conflicts and support conflict management. However, to the best of our knowledge, no current categorisations are taking into account the larger context of requirement conflicts, such as the system or organisational environment, to guide conflict management. Additionally, very few studies consider the cause of the conflict for deciding what conflict management strategies to apply. In our study, we look at these aspects and contribute with a new perspective on conflict types and conflict resolution based on the cause.

III. Research Method

The aim of this research is to understand how conflict management takes place in practice considering the different contexts of conflicts. One aspect of these contexts is the different types of requirement conflicts and how these might impact conflict management. Therefore, the research question in focus for this study is the following: ***How are different types of requirement conflicts managed in practice?***

In order to get a deeper understanding of requirement conflicts in practice and the different contexts in which they take place, a qualitative research approach was taken to explore and generate new insights about conflict types in practice. The study was designed as an interview study and semi-structured interviews were chosen for data collection to enable flexibility as well as consistency and comparability between the interviews [17].

A. Data Collection

Ten semi-structured interviews were conducted with eleven participants from eight companies to gather empirical insights. The interviewees came from six different domains and had between five and twenty years of experience in the area of requirements management. An overview of the interviewee characteristics is shown in Table I. The interviews were conducted in person or online. They lasted between 45 to 60 minutes. Eight of the interviews were held in English and two in Swedish depending on the language preferred by the interviewee. Participants E5 and E6 were interviewed together. Nine of ten interviews were recorded with the participant's permission and afterwards transcribed for analysis. Detailed notes were taken during the interview that was not recorded.

TABLE I
INTERVIEWEES' CHARACTERISTICS

ID[1]	Role	Years of Experience	Domain
A1	CEO	5 years	Management Systems
B2	Cybersecurity Engineer	5 years	Automotive
C3	Head of Onboard Sales and Services Digital	7 years	Freight Transport
D4	Product Owner	9 years	Management Systems
E5	Quality Assurance Manager	20 years	Medical
E6	Project Manager	6 years	Medical
F7	System Architect	20 years	Automotive
G8	Product Owner	15 years	Telecommunications
H9	System Architect	15 years	Automotive
H10	IT Director	18 years	Automotive
H11	Cluster Leader in Digital Finance	15 years	Financial Systems

[1]Letter denotes company, number denotes interviewee

Participants B2, F7, G8, H9, H10, and H11 were sampled based on convenience sampling [18]. The remaining five participants were sampled according to criterion sampling [19] to capture other industries and domains as well. The aim of the selection was to capture as many empirical software contexts as possible, therefore the criteria were companies of different sizes and domains, and secondly, their products or services involve software. There was no set number of participants determined at the beginning of the study and interviews were held until no new themes were identified from the interviews and we considered data saturation to be reached [20].

The interviews were conducted by one interviewer (first author) and performed according to the interview guide prepared beforehand. The complete interview guide can be found in [21]. The questions were formed based on the areas of interest for this study such as conflict types and management as well as two general questions on the interviewee's background. During the interviews, the context of the study was first presented to the interviewees, who were then asked questions about their backgrounds. Moreover, we also asked about and discussed the definition of the terms 'requirement' as well as 'requirement conflict'. Thereafter, we asked questions about the participants' experience with requirement conflicts and their management such as what conflicts they encountered, how they were identified and resolved as well as what they considered to be the conflict's cause. After the initial discussions about conflicts experienced by the interviewees, they were also shown a set of examples of requirement conflicts from a list of ten examples that had been prepared before the

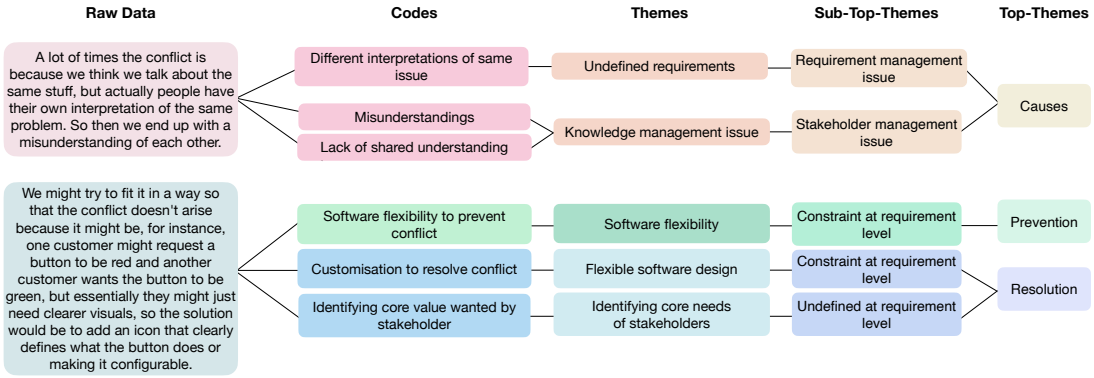

Fig. 1. Coding Examples

interviews [21], which were created during discussions within the author group and partially based on examples found during the initial literature review [22]–[24]. The interviewees' previous answers influenced what examples were shown during the interview. Conflict types that had already been named were not discussed again. Instead, the examples were used to probe the interviewees for different conflict types. When the interviewees were shown examples, they were first asked whether they were familiar with the conflict it presented. Additionally, we asked specific questions about the context and management of the example conflict as specified in the interview guide [21]. Conflicts that had not been experienced or observed were noted as not being relevant for the interviewee and they were asked about why this was the case. Overall, these examples served to stimulate discussion to identify other potential types of conflicts that had not been previously mentioned.

B. Data Analysis

The data from the interviews were analysed using a thematic analysis based on the six steps by Braun and Clarke [25].

Data Familiarisation. In order to familiarise ourselves with the data, the transcriptions and notes from the interviews were read actively multiple times.

Generating Codes. In step two, initial codes were generated. Open codes were used to analyse the results to maintain an explorative approach and identify as many patterns as possible.

Generating Themes. The identified codes were then grouped into different themes based on the patterns that could be identified among the codes. Themes were generated throughout the interview process as new codes emerged. The codes and subsequent themes were generated by the first author.

Reviewing Themes. The identified themes were continuously reviewed and refined to create distinct and coherent themes. After coding several interviews, existing themes and their corresponding codes were reviewed and revised if needed to ensure that each code presented one clear idea. With the review of the identified themes after each interview, we could also ensure sufficient data were collected.

Defining Themes. The themes were further refined and defined to accurately reflect what each theme represents. This step was performed within the author group after all interviews had been conducted. Two complete coding examples are shown in Fig. 1.

Producing the Report. In the final step, the results from the analysis were summarised and written down as shown in Fig. 2 and described in Section IV.

C. Threats to Validity

Validity threats are discussed based on Runeson *et al.* [17].

External Validity. The chosen research design and number of participants imply a low external validity. To minimise the impact of this, we interviewed individuals in different roles, domains as well as companies. Furthermore, the purpose of this study's research design is not to be generalisable but rather to enable a more explorative approach and gain deeper insights into the topic.

Reliability. A set of example conflicts was shown to the participants during the interviews to gain insights into specific scenarios. Due to time constraints, only a subset of the examples was shown and which ones were chosen was based on the interviewee's role and their previous answers, so that the same conflict scenarios were not discussed again. This increases the risk of bias impacting the reliability of this research. Therefore, to increase the reliability, the interview material used in the study is included and the collection as well as analysis of data are described in detail with examples.

Internal Validity. Since different subsets of examples were discussed during the interviews, the internal validity is affected in terms of establishing causality between conflict types, causes, and conflict management. Nonetheless, the research aim was to collect significant insights into conflicts and conflict management in practice rather than making inferences about all types of conflicts that exist. The internal validity was further upheld through a clear chain of evidence as described in the method. Additionally, internal validity was maintained through peer debriefing to reduce the risk of bias [17] and the thematic analysis was performed among three researchers to include different perspectives and maintain objectivity.

Construct Validity. In order to achieve a higher level of construct validity of the research, the definitions for both

conflicts between requirements as well as the term requirement itself were discussed during the interviews to ensure that the interviewer and interviewees shared the same understanding of these concepts [17]. Moreover, some of the examples shown during the interviews can be considered very generalised which presents another issue to the research validity. However, the examples were kept at a generally understandable level to initiate discussions about different types of requirement conflicts and their management.

IV. Results

To answer the research question of this study, it is necessary to understand what types of requirement conflicts exist in practice and their current management strategies. Fig. 2 shows an overview of our findings and includes all the themes identified during the thematic analysis. We present the types of conflicts in Sect. IV-A, their causes in Sect. IV-B, and the strategies for resolving conflicts are described in Sect. IV-C.

A. Types of Requirement Conflicts

The study showed that there are different perspectives for defining requirement conflicts. Our interviewees defined conflicts as a contradiction between two or more requirements. Many interviewees also noted that conflicts are contextual and that the definition of a conflict is dependent on, for example, the role and subsequent perspective an individual takes. H9 explained, *"It is quite easy to spot the conflict I think for a specific [requirement], because [a] conflict is also contextual. So, depending on your role looking into those inputs coming into your specific context, you will be able to assess if that is a conflict within your specific consideration context"*. Consequently, to categorise a requirement conflict, we need to consider the contexts in which conflicts occur and on what level.

At the core of all the mentioned conflicts, there is a conflict between either non-functional requirements (NFRs) or functional requirements (FRs). Therefore, three 'core' conflicts were identified, i) *conflict between NFRs*, ii) *conflict between NFR and FR*, and iii) *conflict between FRs*, that all participants have experienced or witnessed to different extents. NFRs were often found to conflict with FRs or other NFRs due to them representing a factor that the software needed to comply with such as laws, standards, or business values. The resolution of conflicts with NFRs was therefore based on how strict or important they are and required a trade-off or compromise.

Apart from the conflicts based on the set of requirements, the interviews revealed different types of conflicts on a higher contextual level that involved other factors as well, as seen in Fig. 2. These types are described in more detail below.

1) Software Level Conflict Types: The interviews revealed three types of conflicts on the software level.

The *variability conflict* was identified from the interviews with interviewees A1, B2, C3, D4, E5, and E6. The conflict occurs between requirements from different configurations or variants, such as in different ECUs in the automotive domain. Moreover, it also represents conflicts between customisability and standardisation. For example, companies offering one software platform for multiple different customers need to find a balance to manage conflicts between requirements that satisfy individual stakeholder's needs while maintaining a reasonable degree of product standardisation.

Implication conflicts stem from conflicts between the requirements of the existing software and new requirements. For instance, the requirements that come from the existing system in the form of legacy code can conflict with requirements of new features or implementations, such as that *"it is a function that works as it should but there might be some dependencies that make things difficult, for example, old code"* (D4). We found that this type of conflict can occur in systems with many external dependencies when their requirements do not align or when one requirement cannot be realised until another requirement is fulfilled.

Moreover, *technical constraint conflicts* arise when some requirements are in conflict with constraints and can therefore not be fulfilled under the current circumstances. For example, multiple processes require a certain percentage of a CPU, which is in conflict with constraints that are mandated by hardware limitations.

2) Stakeholder Level Conflict Types: The interviews showed that there are distinct conflict types for conflicting requirements involving different stakeholders which arise in the interactions with them. These conflicts tend to involve cognitive or social issues. *Customer conflicts* are misalignments between the requirements that were expected by the customer and the ones that were performed or understood by the organisation working with them.

Moreover, *team conflicts* are conflicts between requirements within companies and different teams. This conflict can occur when teams are not communicating or coordinating with each other effectively and they end up defining requirements that contradict another team's requirements. Additionally, D4 explained that it can also be a conflict between the views and goals within or between teams that results in conflicting requirements. While the team and customer conflicts are very similar in terms of the cognitive misalignment they represent, the dynamic between the conflicting requirements from teams is different since they are working within the same scope of the software and therefore ideally with similar goals and values, which does not necessarily apply to requirement conflicts with external stakeholders.

Furthermore, we identified *integration conflicts*, which arise due to contradictions between requirements set by suppliers or the organisation integrating an external component. When the supplier's components do not align with the functions of the system integrating them, they cannot be merged with the system which results in conflicts.

3) Business Level Conflict Types: Lastly, our interviewees mentioned requirement conflicts on the business or operational level. One of the conflicts that constrain the solution space due to non-technical aspects is a *cost conflict*, which arises from cost constraints that do not allow all requirements to be fulfilled and therefore cause them to be in conflict.

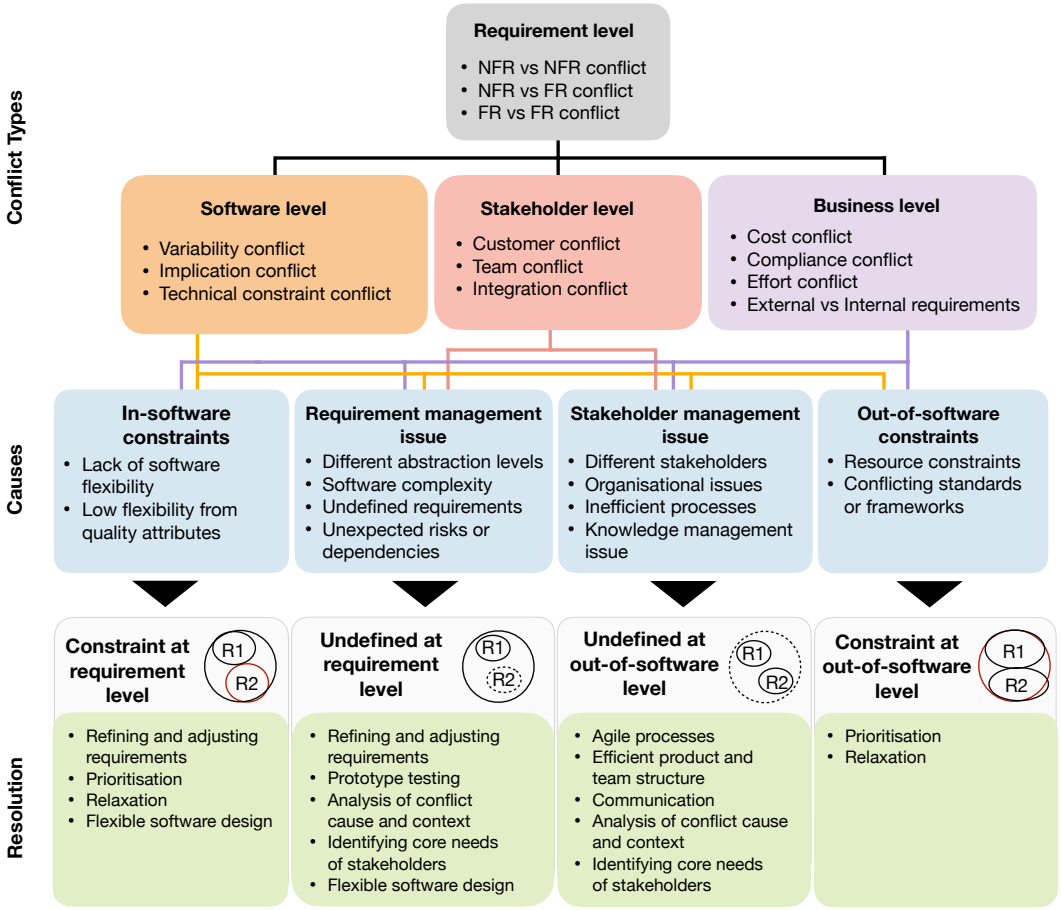

Fig. 2. Overview of Our Interview Findings on Requirement Conflicts

Additionally, a *compliance conflict*, which involves requirements for compliance to certain standards or laws, is a specific type of conflict that was identified during the study. The compliance conflict also relates to compliance with company values and goals. This conflict was found to be typically an issue between quality attributes.

Effort conflicts could also be identified from the interviews. These conflicts relate to effort either in the form of time constraints, due to, for example, deadlines set by stakeholders, or in terms of the required competence and resources such as developers or specific expertise. For example, G8 stated *"we also need to find the right people to do it, people with high experience involved in decisions, but at the same time you then lose them somewhere else"*.

Finally, the results from the interviews showed that a *conflict between internal and external requirements* denotes another type of conflict. This is a conflict between the need and ability to deliver value. These conflicts often arise from operational requirements contradicting with system requirements, or when existing process requirements are in conflict with the introduction of new technology (C3). With requirements originating from the customer, this conflict type also often raises the question of providing some value to customers now or more value later. For instance, *"the customers want new features, nicer designs and faster systems and (...) the internal requirements, these are important things as well such as code quality and maintainability, so it is a discussion between these two interests, like how much can we do here and now to actually give customers something new while at the same time making sure that our internal requirements are met"* (D4).

B. Causes of Different Types of Conflicts

The interviewees often mentioned that the resolution of a conflict varied for each case. Their management strategy was based on relevant properties of the conflict such as the cause. The causes for requirement conflicts that were identified during the study are grouped into four categories; in-software constraints, requirement management issue, stakeholder management issue, and out-of-software constraints, see Fig. 2. The causes were categorised in one dimension as more technical (in-software constraints and requirement management issue) or more of a social issue (stakeholder management issue and out-of-software constraints). The second dimension relates to the source of conflict being either due to a cause constraining the scope which restricts the solution space (in-software and out-of-software constraints), or a cause that makes the scope more undefined and unclear leading to false assumptions and

knowledge gaps (requirement and stakeholder management issue). The most relevant causes are described below.

1) In-software Constraints: For causes representing constraints on the in-software level, we found that a *lack of software flexibility* can lead to technical constraints or limitations in how much variability the system can offer and its ability to adapt to different circumstances or stakeholder needs. *Low flexibility of certain quality attributes* of a system was viewed by seven participants to be another cause for conflict. The quality attributes are more fixed if they represent, for example, technical constraints, company and brand values, or have legal implications. Low flexibility from one quality attribute typically leads to a conflict due to a compromise between the attributes or blocking of one or more requirements. For instance, H9 commented *"there is a lot of safety consideration when it comes to our solutions and that might result in reducing different or other quality aspects of the system like availability or performance"*.

2) Requirement Management Issue: Another group of causes is the one that leads to 'undefinedness' on the requirement level. *Different abstraction levels* of requirements is one of these causes. G8 stated *"it is not uncommon to get really high-level requirements and you do not find conflicts there, then when you look at what the requirement means you realise there are conflicts"*. We found that different abstraction levels are often associated with the different roles in an organisation and subsequently different levels of perspectives as well as requirements. The level of requirement is typically dependent on its source. For example, non-technical stakeholders will often put forward high-level requirements that need to be defined further to understand their implications and identify conflicts since there is often room for interpretation.

Undefined requirements were identified as another cause in this category. They can lead to misinterpretations and uncertainty about what requirements need to be fulfilled. For instance, too high-level requirements present an issue since they can lack specific information leading to conflicts later on between requirements on a more detailed level since the higher level requirements were interpreted differently among stakeholders. B2 stated *"if your requirements towards your supplier are too high level, if it is not detailed enough, then it opens room for interpretations"*. Too high-level requirements can also mean different implementations when there are different configurations in the software as they do not specify how they are to be implemented. For example, B2 explained, *"let's say they told us to implement TLS Protocol 1.2, but then how you implement TLS 1.2 could be very different based on the configuration you do"*. Therefore the need for clear requirements increases in this case as well.

We found *unexpected risks or dependencies* such as unknown requirements that become visible at runtime to be another reason for conflicts that contribute to an undefined software scope. E5 and H10 mentioned this to be a cause for particularly customer and resource conflicts where insufficient scoping and boundary testing of requirements contributed to this issue.

3) Stakeholder Management Issue: Causes in this cluster involve uncertainty or indefiniteness at the out-of-software level and subsequently for the scope of the software. One of these is *different stakeholders* and viewpoints. For example, conflicts can occur due to a software system having to cater to customers from different industries and therefore requirements. Additionally, different goals or processes among teams, due to the different roles or value propositions they might have, can also lead to conflicts.

Inefficient processes for requirement management and communication are another external issue to the software that can cause conflicts. For instance, B2 explained that there will be conflicting requirements when teams work independently with interconnected functionality and there are no efficient requirement management processes or communication in place to prevent them. Issues with requirement management can also cause misalignment because of missing functionalities that were not gathered in time. Many interviewees mentioned that conflicts often arise when requirements are not allocated or communicated at the right time. Moreover, all participants in this study worked in an agile manner for some or all of their projects and their descriptions of the cases of inefficient processes leading to conflicts could also be interpreted as a lack or misuse of agile processes.

Finally, *knowledge management issues* such as misunderstandings, bad documentation, and generally a lack of common definitions lead to conflicts between requirements as well based on the findings from five interviews. Usually, conflicts in the form of misalignments between stakeholders occur due to misunderstandings or information discrepancies. F7 stated that *"bad product documentation opens up for conflicts since expectations do not align with the descriptions"*.

4) Out-of-software Constraints: One of the causes grouped in this category that impose constraints from outside the software are *resource constraints* which contribute to conflicts by creating constraints either in the form of cost, time, or hardware limitations.

Additionally, *conflicting standards or frameworks* were mentioned as a cause specifically of the compliance conflict by B2, C3, E5, E6, G8, and H9. An example of this was given by B2 operating in the automotive domain where a conflict was identified between GDPR and the tachograph, which regulates how much a driver can drive and requires the system to collect some personal data such as the driver's location or the car's vehicle identification number to help identify the vehicle. The need to collect and store personal data interferes with the GDPR and therefore creates a conflict between their requirements for the system.

C. Resolution of Different Types of Conflicts

Based on the identified causes, it was possible to establish themes for the resolution strategies that were mentioned foremost for the respective cluster of causes. We identified resolution strategies linked to each cause, however, some strategies can be used for several causes while others are connected to a specific cluster of causes. The strategies have

been grouped and mapped to these causes, as shown in Fig. 2. The most frequently mentioned ones are described below.

1) Constraint at Requirement Level: One solution that was brought up for constraining causes at the requirement level is *prioritisation*. It was one commonly named solution for many types of conflicts by all interviewees and was often linked to causes relating to constraints. This resolution strategy includes evaluating requirements according to their value, effort, and impact. Additionally, the term 'balancing' was mentioned and refers to a form of requirement *relaxation* where one or multiple requirements in the conflict are weakened to enable a resolution. This was brought up primarily in relation to quality attributes and NFRs as well as resources. F7 stated that *"in the end you need to reach an acceptable level for all quality attributes and some are more critical than others"*. Requirement prioritisation and relaxation were also considered to be a collaborative activity that required communication and involving the right stakeholders to collect the necessary information and have a sufficient overview of the conflict to determine the value and impact of different solutions.

Another solution mentioned by eight interviewees was *refining and adjusting requirements* which is described in more detail in Sect. IV-C2. Requirements in conflict with a technical constraint usually need to be adjusted to resolve the conflict since the constraint is not flexible. Therefore, this strategy is important for either getting a more defined scope or redefining it to adjust for constraints.

2) Undefined at Requirement Level: The resolution strategies mentioned for causes on the requirement level that lead to an undefined scope and subsequent solution space have a strong focus on gaining a better understanding of the conflict and the involved requirements. The importance of managing and maintaining scope was mentioned by multiple interviewees. Therefore, *refining and adjusting requirements* to keep a clear scope as well as rescope when necessary was one resolution strategy for managing conflict causes that lead to an undefined scope. For instance, due to the different abstraction levels in software development, it is important to break requirements down so they, and subsequently the scope, become more defined. However, this often requires a deeper understanding of the product and involved stakeholders.

Additionally, the importance of *identifying the actual needs of stakeholders* was found to be another central factor in resolving conflicts that occur due to a lack of scope clarity or alignment. A1 mentioned *"oftentimes there are requirements from customers where you then have to try to break down what they really want"*. Subsequently, rephrasing and adjusting requirements can then resolve a conflict to make the requirement a more accurate reflection of what need it represents.

Furthermore, another resolution strategy we identified is having a good overview of the conflict, as *analysing the context and cause of the conflict* facilitates its resolution. H10 commented *"if you just look at things one by one it does not work because if you have a budget for requirement A and you get a requirement B that does not have a budget it will not work (...) and this is a kind of skill you need to have if you know to put all the components together and identify the bottlenecks"*. Another central aspect of this is involving different and, more importantly, the right stakeholders, which is also closely tied to resolution strategies relating to social causes such as an effective team structure and processes.

Lastly, a *flexible software design* was another important means for resolving conflicts involving an undefined scope at the requirement level based on the interviews with A1, B2, D4, E5, and E6. The interviewees reported that already considering the emergence of conflicts in the software design helps resolve many conflicts, however it can be relatively resource intensive. Moreover, we found that higher software flexibility through customisability, such as different software versions, is also instrumental for conflict resolution. This includes being able to manage different configurations more effectively that can cause variability conflicts.

3) Undefined at Out-of-software Level: Communication was considered by all interviewees as an important resolution strategy, particularly in relation to the social causes that are also characterised by the resulting uncertainty in the scope. Joint reviews or workshops were often used as a tool to resolve conflicts or to identify, for instance, the right level of requirements to help resolve integration conflicts (B2). Moreover, multiple participants mentioned the importance of people being able to see the value of a solution to resolve a misalignment between stakeholders, where a significant factor was having discussions to share different views on a conflict and in the end reach a common understanding. Communication became a more influential factor in conflicts that were considered more complex as information transparency and the involvement of different stakeholders became more important.

Another factor here is documentation which can further aid in the understanding of the reasoning and decision-making behind requirements and design choices to support conflict management and also to avoid overlaps in requirements from different stakeholders. However, implicit to effective communication is also involving the right stakeholders. This leads to the importance of an *effective product and team structure* which was another resolution strategy mentioned in seven interviews. Since a network of people is typically involved in conflict management and decision-making, it is important to have the right structure where the right people can be included or teams within an organisation can coordinate to resolve resource related conflicts. C3 reported that for some conflicts a change in the system or operational processes is necessary to resolve a conflict. Moreover, G8 mentioned a bottom-up approach to team structure and software design to manage conflicts as it is possible to work with more defined requirements early on and create a clearer scope. However, G8 highlighted that this can also confuse since the boundaries to some requirements such as NFRs are derived from top-level goals or requirements that have to then be managed with assumptions as well.

4) Constraint at Out-of-software Level: Prioritisation was also mentioned as a common solution to conflicts with constraining out-of-software causes to achieve the requirements

that are considered to be most important for the system and its stakeholders. For instance, legal requirements are often prioritised to ensure compliance. Furthermore, we found relaxation of requirements to be another resolution strategy to resolve external constraints such as relaxing cost constraints to reach a higher level of system performance.

V. Discussion

In this study, we identified ten conflict types that can be grouped into three levels: software level conflicts, stakeholder level conflicts, and business level conflicts. However, we found that these types did not impact the management of requirement conflicts in practice. Instead, how they were managed was generally dependent on the conflict causes which we grouped into four clusters as seen in Fig. 2.

Based on our findings, there were some unique connections between the types and causes of conflicts, which shows the complex relationship between the two concepts. Conflicts that take place on the business and software levels were linked to causes in all four clusters, while conflicts on the stakeholder level were primarily connected to the dimension of causes contributing to an undefined requirement scope. Since conflict types on the stakeholder level were considered to be caused by process and interaction issues, it indicates that these conflicts might be more cognitive in nature. Therefore, the human factor has more of an impact and subsequently needs to play a bigger role in conflict management as well. Furthermore, many of the types, as well as causes of conflicts that were identified, were very connected to the non-technical aspects of software and requirement engineering such as stakeholder management and organisational or legal constraints.

Our findings from the interviews show the importance of looking at the bigger picture when it comes to requirement conflicts. Existing research [7], [13], [14] mainly focuses on the nature of conflicting requirements. Yet, these related studies do not address other contextual factors that generally affect requirements. Some works [15], [16] consider the semantics of requirement conflicts such as perception of requirements, and Kim *et al.* [16] include the cause of the conflict in their categorisation. However, larger scale issues that might be impacting requirement conflicts are generally not captured. We found that the requirements themselves might not be the only issue, but possibly reflect a bigger conflict that affects the software and organisation as a whole. Understanding the type of requirement conflict and cause that is encountered can then enable more effective conflict resolution.

A main finding in this study concerning the causes of requirement conflicts is the distinction between technical and social causes as well as the requirement scope being either too undefined or constrained, which subsequently impacts the conflict and its management. The differentiation of causes for requirement conflicts being either social or technical has been discussed in previous research. For instance, Robinson *et al.* [26] identifies both technical and social difficulties leading to conflicts. They consider, for example, too many or complex requirements as technical difficulties, and conflicting interests between stakeholders or changing expectations as social difficulties.

To the best of our knowledge, there is no previous study differentiating between requirement conflicts according to the nature of the requirement scope. An undefined scope causes conflicts when multiple stakeholders do not have aligned expectations and a common understanding of the requirement scope. Additionally, a lack of processes in place to manage this uncertainty further contributes to the conflict, since, for example, customers might change their minds later on. This definition of conflict was found in other domains such as on a more general interpersonal level by Rijsberman [27] who defines conflicts based on the state of the solution space as being either simple and well-defined, making it clear what the solution to the conflict is, or complex and ill-defined, which involves undefined objectives and values of stakeholders making it unclear what the solution is. However, it has not been discussed in the context of requirements and conflicts between them. Therefore, for the domain of requirements engineering, it indicates a new way to approach conflict management.

VI. Conclusion

Due to the context-dependent and dynamic factors of requirement conflicts and their environment, conflict management strategies need to be able to consider and adapt to these aspects to be effective in practice. This study provides a general overview of types of conflicts and conflict management in practice. We found two dimensions to the identified conflict types and causes; technical or social and undefined or constrained. We also found that conflict types are not the determining factor for selecting management strategies, instead, the causes for conflicts represented a significant factor for connecting the types to different resolution strategies. This study along with previous research shows that the nature of the conflict is significant to its management. In future work, we aim to further address this by exploring adaptive resolution strategies tailored to specific conflict situations. We also plan to further examine the role of contextual factors of requirement conflicts and their impact on conflict resolution in practice, such as investigating the effects of specific resolution strategies in relation to the conflict context. Additionally, it would be interesting to explore the use of AI for conflict management and to help identify different types of conflicts.

Acknowledgement

Thank you to the participants of the study for their insights. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

[1] R. V. Anand and M. Dinakaran, "Handling stakeholder conflict by agile requirement prioritization using apriori technique," *Computers & Electrical Engineering*, vol. 61, pp. 126–136, 2017.

[2] R. Ali, F. Dalpiaz, and P. Giorgini, "A goal-based framework for contextual requirements modeling and analysis," *Requirements engineering*, vol. 15, pp. 439–458, 2010.

[3] S. Robertson and J. Robertson, *Mastering the requirements process: Getting requirements right*. Addison-wesley, 2012.

[4] A. Salado and R. Nilchiani, "The concept of order of conflict in requirements engineering," *IEEE systems journal*, vol. 10, no. 1, pp. 25–35, 2014.

[5] W. Guo, L. Zhang, and X. Lian, "Automatically detecting the conflicts between software requirements based on finer semantic analysis," *arXiv preprint arXiv:2103.02255*, 2021.

[6] A. Salado and R. Nilchiani, "The tension matrix and the concept of elemental decomposition: Improving identification of conflicting requirements," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2128–2139, 2015.

[7] W. H. Butt, S. Amjad, and F. Azam, "Requirement conflicts resolution: Using requirement filtering and analysis," in *Computational Science and Its Applications-ICCSA 2011: International Conference, Santander, Spain, June 20-23, 2011. Proceedings, Part V 11*, Springer, 2011, pp. 383–397.

[8] M. Roy, N. Deb, A. Cortesi, R. Chaki, and N. Chaki, "Requirement-oriented risk management for incremental software development," *Innovations in Systems and Software Engineering*, vol. 17, no. 3, pp. 187–204, 2021.

[9] I. Hadar, A. Zamansky, and D. M. Berry, "The inconsistency between theory and practice in managing inconsistency in requirements engineering," *Empirical Software Engineering*, vol. 24, no. 6, pp. 3972–4005, 2019.

[10] A. Salado and R. Nilchiani, "A set of heuristics to support early identification of conflicting requirements," in *INCOSE International Symposium*, Wiley Online Library, vol. 25, 2015, pp. 266–279.

[11] B. Boehm and H. In, "Identifying quality-requirement conflicts," *IEEE software*, vol. 13, no. 2, pp. 25–35, 1996.

[12] A. Van Lamsweerde, R. Darimont, and E. Letier, "Managing conflicts in goal-driven requirements engineering," *IEEE transactions on Software engineering*, vol. 24, no. 11, pp. 908–926, 1998.

[13] D. Mairiza and D. Zowghi, "Constructing a catalogue of conflicts among non-functional requirements," in *Evaluation of Novel Approaches to Software Engineering: 5th International Conference, ENASE 2010, Athens, Greece, July 22-24, 2010, Revised Selected Papers 5*, Springer, 2011, pp. 31–44.

[14] X. F. Liu and J. Yen, "An analytic framework for specifying and analyzing imprecise requirements," in *Proceedings of IEEE 18th International Conference on Software Engineering*, 1996, pp. 60–69.

[15] S. Easterbrook, "Resolving requirements conflicts with computer-supported negotiation," *Requirements engineering: social and technical issues*, vol. 1, pp. 41–65, 1994.

[16] M. Kim, S. Park, V. Sugumaran, and H. Yang, "Managing requirements conflicts in software product lines: A goal and scenario based approach," *Data & Knowledge Engineering*, vol. 61, no. 3, pp. 417–432, 2007.

[17] P. Runeson, M. Höst, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.

[18] S. J. Stratton, "Population research: Convenience sampling strategies," *Prehospital and disaster Medicine*, vol. 36, no. 4, pp. 373–374, 2021.

[19] L. A. Palinkas, S. M. Horwitz, C. A. Green, J. P. Wisdom, N. Duan, and K. Hoagwood, "Purposeful sampling for qualitative data collection and analysis in mixed method implementation research," *Administration and policy in mental health and mental health services research*, vol. 42, pp. 533–544, 2015.

[20] G. Guest, A. Bunce, and L. Johnson, "How many interviews are enough? an experiment with data saturation and variability," *Field methods*, vol. 18, no. 1, pp. 59–82, 2006.

[21] A. Welzel, *Interview guide*, 2024. [Online]. Available: https : / / figshare . com / articles / journal_contribution / Interview_Guide_pdf/25049972.

[22] T. Moser, D. Winkler, M. Heindl, and S. Biffl, "Requirements management with semantic technology: An empirical study on automated requirements categorization and conflict analysis," in *Advanced Information Systems Engineering: 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings 23*, Springer, 2011, pp. 3–17.

[23] J. C. Maxwell, A. I. Antón, and P. Swire, "A legal cross-references taxonomy for identifying conflicting software requirements," in *2011 IEEE 19th international requirements engineering conference*, IEEE, 2011, pp. 197–206.

[24] A. Sardinha, R. Chitchyan, J. Araújo, A. Moreira, and A. Rashid, "Conflict identification with ea-analyzer," *Aspect-oriented requirements engineering*, pp. 209–224, 2013.

[25] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[26] W. N. Robinson, S. D. Pawlowski, and V. Volkov, "Requirements interaction management," *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, pp. 132–190, 2003.

[27] F. R. Rijsberman, *Conflict management and consensus building for integrated coastal management in Latin America and the Caribbean*. Inter-American Development Bank, Sustainable Development Department, 1999.