7 Requirements Validation and Negotiation

Validation and negotiation during requirements engineering is meant to ensure that the documented requirements meet the predetermined quality criteria, such as correctness and agreement (see section 4.6). The introduced principles and techniques can be used to validate and negotiate individual requirements or entire requirements documents.

7.1 Fundamentals of Requirements Validation

During the requirements engineering activity, it is necessary to review the quality of the requirements developed. Among others, the requirements are presented to the stakeholders with the goal to identify deviations between the requirements defined and the stakeholders' actual wishes and needs.

Approving requirements

During requirements validation, the decision of whether a requirement possesses the necessary level of quality is made (see chapter 4) and whether the requirement can be approved to be used for further development activities (such as design, implementation, and testing). This decision should be made on the basis of predefined acceptance criteria.

Goal of validation

The goal of requirements validation is therefore to discover errors in the documented requirements. Typical examples of errors in requirements are ambiguity, incompleteness, and contradictions (see section 7.3).

Error proliferation

Requirements documents are reference documents for all further development activities. Therefore, errors negatively affect all further development activities. A requirements error that is discovered when the system is already deployed and operating requires all artifacts affected by the error to be revised, such as source code, test artifacts, and architectural descriptions. Correcting errors in requirements once the system is in operation therefore entails significant costs.

Legal risks

A contract between client and contractor is often based on requirements documents. Critical errors in requirements can lead to the fact that contractual agreements cannot be met, e.g., scope of supply and services, expected quality, or completion deadlines.

7.2 Fundamentals of Requirements Negotiation

Contradictory requirements cause conflicts.

If there is no consent among the stakeholders regarding the requirements and thus the requirements cannot be implemented collectively in the system, a conflict arises between the contradictory requirements as well as between the stakeholders that demand contradictory requirements. For example, one stakeholder could demand the system to shut down in case of a failure, whereas another stakeholder could require the system to restart.

Risks and opportunities of conflicts

The acceptance of a system is threatened by unresolved conflicts because unresolved conflicts cause the requirements of at least one group of stakeholders to not be implemented. In the worst case, a conflict causes stakeholder support to cease, causing the development project to fail (cf. [Easterbrook 1994]). Other than posing risks, conflicts can also be an opportunity for requirements engineering because conflicts between stakeholders require a solution that can potentially help discover new ideas for development and can illustrate different options (cf. [Gause and Weinberg 1989]). Therefore, treating and resolving conflicts openly during requirements engineering can increase acceptance.

Goal of requirements negotiation

The goal of negotiation is to gain a common and agreed-upon understanding of the requirements of the system to be developed among all relevant stakeholders.

Reducing costs and risks in late phases

Requirements validation and negotiation is an activity that must be performed (to a varying degree of intensity) throughout the entirety of requirements engineering. The validation and negotiation of requirements therefore causes additional effort and therefore additional costs. However, the advantages gained by performing requirements validation and negotiation as described in the previous sections (reduction of overall cost, increase in acceptance, supporting creativity and innovations) is usually significantly higher than the costs that arise due to the increased effort.

7.3 Quality Aspects of Requirements

A major aim of using quality criteria (e.g., completeness, understandability, agreement) in requirements validation is to be able to check requirements systematically (see section 1.1.2). In order to assure an objective and consistent validation, it is necessary that each quality criterion is concretized and refined. In correspondence with the overall goals of the requirements engineering process, the validation is carried out with the following goals:

- *Content:* Have all relevant requirements been elicited and documented with the appropriate level of detail?
- *Documentation:* Are all requirements documented with respect to the predetermined guidelines for documentation and specification?
- *Agreement:* Do all stakeholders concur with the documented requirements and have all known conflicts been resolved?

Three quality aspects

Each of the three goals implies an individual approach that focuses on specific aspects of the quality of the requirements. Therefore, the following three quality aspects have been defined:

- Quality aspect "content"
- Quality aspect "documentation"
- Quality aspect "agreement"

A requirement should be approved for further development activities only if all three quality aspects have been checked. The quality aspects are described in detail in the following sections and made concrete through different fine-grained quality criteria (with no claim of completeness).

7.3.1 Quality Aspect "Content"

The quality aspect "content" refers to the validation of requirements with respect to errors in the content. Errors in requirements with regard to content negatively influence the subsequent development activities and cause these activities to be based upon erroneous information.

Test criteria of the quality aspect "content"

Errors in requirements with regard to content are present when specific quality criteria for requirements (see section 4.6) or for requirements documents (see section 4.5) are violated. The validation of requirements with regard to the quality aspect "content" is successful once requirements

validation has been applied to the following error types and no significant shortcomings have been detected:

- *Completeness (set of all requirements):* Have all relevant requirements for the system to be developed (for the next system release) been documented?
- *Completeness (individual requirements):* Does each requirement contain all necessary information?
- *Traceability:* Have all relevant traceability relations been defined (e.g., to relevant requirements sources)?
- *Correctness/adequacy:* Do the requirements accurately reflect the wishes and needs of the stakeholders?
- *Consistency*: Is it possible to implement all defined requirements for the system to be developed jointly? Are there no contradictions?
- *No premature design decisions:* Are there any forestalled design decisions present in the requirements not induced by constraints (e.g., constraints that specifiy a specific client-server architecture to be used)?
- *Verifiability:* Is it possible to define acceptance and test criteria based on the requirements? Have the criteria been defined?
- *Necessity:* Does every requirement contribute to the fulfillment of the goals defined?

7.3.2 Quality Aspect "Documentation"

The quality aspect "documentation" deals with checking requirements with respect to flaws in their documentation or violations of the documentation guidelines that are in effect, such as understandability of the documentation formats and the consideration of organizational or project-specific guidelines regarding the documentation of requirements but also the structure of the requirements documents.

Implications of the violation of documentation guidelines

Ignoring the documentation guidelines can, among other things, lead to the following risks:

- *Impairment of development activities:* It may be impossible to carry out development activities that are based upon a specific documentation format.
- *Misunderstandings:* Requirements may not be understandable or may be misunderstood by the people that need to comprehend them. As a result, the requirement may be unusable.

- *Incompleteness:* Relevant information is not documented in the requirements.
- *Overlooking requirements:* If requirements are not documented at the position that they are supposed to in the requirements document, these requirements may be overlooked in subsequent activities.

Test criteria of the quality aspect "documentation"

Requirements validation with regard to the quality aspect "documentation" is successful when requirements validation has been applied to the following error types and no significant shortcomings have been detected:

Four test criteria of the quality aspect "documentation"

- *Conformity to documentation format and to documentation structures:* Are the requirements documented in the predetermined documentation format? For instance, has a specific requirements template or a specific modeling language been used to document the requirements? Has the structure of the requirements document been maintained? For instance, have all requirements been documented at the position defined by the document structure?
- *Understandability:* Can all documented requirements be understood in the context given? For instance, have all terms used been defined in a glossary (see section 4.7)?
- *Unambiguity:* Does the documentation of the requirements allow for only one interpretation or are multiple different interpretations possible? For instance, does a text-based requirement not possess any kind of ambiguity?
- *Conformity to documentation rules:* Have the predetermined documentation rules and documentation guidelines been met? For instance, has the syntax of the modeling language been used properly?

7.3.3 Quality Aspect "Agreement"

The quality aspect "agreement" deals with checking requirements for flaws in the agreement of requirements between stakeholders.

Last opportunity for changes

During the course of requirements engineering, stakeholders gain novel knowledge about the system to be developed. Due to this additional knowledge, the opinion of the stakeholders regarding a requirement that has already been agreed upon can change. During requirements validation, stakeholders have the opportunity to requests changes without impairing the subsequent development activities.

Requirements validation with regard to the quality aspect "agreement" is successful when requirements validation has been applied to the following error types and no significant shortcomings have been detected:

Three test criteria of the quality aspect "agreement"

- *Agreed:* Is every requirement agreed upon with all relevant stakeholders?
- *Agreed after changes:* Is every requirement agreed upon with all relevant stakeholders after it has been changed?
- *Conflicts resolved:* Have all known conflicts with regard to the requirements been resolved?

7.4 Principles of Requirements Validation

Considering the following six principles of requirements validation increases the quality of the validation results:

- *Principle 1:* Involvement of the correct stakeholders
- *Principle 2:* Separating the identification and the correction of errors
- *Principle 3:* Validation from different views
- *Principle 4:* Adequate change of documentation type
- *Principle 5:* Construction of development artifacts
- *Principle 6:* Repeated validation

The individual principles are explained in the following sections.

7.4.1 Principle 1: Involvement of the Correct Stakeholders

The choice of stakeholders for requirements validation depends on the goals of the validation as well as the requirements that are to be audited.

When assembling the auditing team, at least the following two aspects ought to be considered.

Independence of the auditor

Generally, it should be avoided that the author of a requirement is also the person to validate it. The author will make use of his or her prior knowledge when reading or reviewing the requirement. This prior knowledge can negatively influence the identification of errors because potential erroneous passages of the requirements documentation or the requirements are implicitly and subconsciously amended by the author's own knowledge and can thus easily be overlooked.

Internal vs. external auditors

Suitable auditors can be identified within or outside of the developing organization. Internal audits are performed by stakeholders that are members of the developing organization and can be used to validate intermediate results or preliminary requirements. An internal validation is easy to coordinate and organize because the stakeholders are available from within the organization. An external audit requires a higher degree of effort because it identifies auditors and (potentially) hires them for payment. In addition, external auditors have to become familiar with the context of the system to be developed. Due to the high effort, an external audit should be performed only on requirements that exhibit a high level of quality.

7.4.2 Principle 2: Separating the Identification and the Correction of Errors

Basic principle

Separation between identifying errors and actually fixing them has proven itself in the domain of software quality assurance. The same principle can be applied to requirements validation. During validation, the flaws identified are documented immediately. After that, each flaw identified is double-checked to determine whether it really is an error.

Concentrating on error identification

Separating error identification and error correction allows auditors to concentrate on the identification. Measures to correct the errors are taken only after identification measures have been completed. This has the advantages that the resources available for error correction can be used purposefully, that premature error identification does not create additional errors, and that no alleged error is fixed that did not need fixing because further investigation of the error may result in the fact that an alleged error is in fact no error at all. That way, potentially present significant errors are less likely to be overlooked because the auditor is concentrating on fixing a previous error instead of identifying new ones.

7.4.3 Principle 3: Validation from Different Views

Perspective-based validation

Validating requirements from different views is another principle that has proven itself in practice. In this principle, requirements are validated and agreed upon from different perspectives (e.g., by different people, see section 7.5.4). Comparable methods are used in other disciplines as well. For instance, in a legal trial, circumstances are often reported from the perspective of different people so that a sound overall picture can be gained.

7.4.4 Principle 4: Adequate Change of Documentation Type

Strengths and weaknesses of documentation types

Changing the documentation type during requirements validation uses the strengths of one documentation type to balance out the weaknesses of other documentation types. For instance, good understandability and expressiveness are strengths of natural language texts. However, their weakness is potential ambiguity and difficulty in expressing complex circumstances. Graphic models are able to depict complex circumstances rather well, but the individual modeling constructs are restricted in expressiveness.

Simpler identification of errors

Transcribing a requirement that is already documented in another form of documentation simplifies finding errors. For instance, ambiguities in natural language requirements can be identified much easier by transcribing them into a model-based representation.

7.4.5 Principle 5: Construction of Development Artifacts

Suitability of the requirements for design, test, and manual creation

Constructing development artifacts aims at validating the quality of requirements that are meant to be the basis of creating design artifacts, test artifacts, or the user manual. During the course of the validation, the activities usually carried out during subsequent phases to construct respective development artifacts are carried out for small samples. For instance, the auditor intensively deals with a requirement by creating a test case. This way, errors (e.g., ambiguity) can be identified in the requirement. This kind of validation, however, demands a lot of resources because subsequent development activities must be executed at least in part.

7.4.6 Principle 6: Repeated Validation

Validation occurs at a distinct point in time during the development process and relies on the level of knowledge of the auditors at that point in time. During requirements engineering, the stakeholders gain additional knowledge about the planned system. Therefore, a positive validation of requirements does not guarantee that requirements are still valid at a later point in time. Requirements validation should occur multiple times in the following cases (among others):

- Lots of innovative ideas and technology used in the system
- Significant gain of knowledge during requirements engineering
- Long-lasting projects

- Very early requirements validation
- Unknown domain
- Requirements that are to be reused

7.5 Requirements Validation Techniques

In the following sections, techniques for requirements validation are introduced. Often, manual validation techniques, which are also known by the general term *review*, are used for requirements validation. Three major types of reviews can be differentiated:

- Commenting
- Inspections
- Walk-throughs

Along with reviews, the following three techniques have proven themselves to be useful for requirements validation:

- Perspective-based reading
- Validation through prototypes
- Using checklists for validation

In the following, these six techniques are described. Prior to applying any of these techniques, preparatory steps need to be taken as needed, such as identifying and inviting the right stakeholders or organizing suitable rooms and supplies.

7.5.1 Commenting

Individual validation of requirements

During commenting, the author hands his or her requirements over to another person (e.g., a co-worker). The goal is to receive the co-worker's expert opinion with regard to the quality of a requirement. The co-worker reviews the requirement with the goal to identify issues that impair requirement quality (e.g., ambiguity or errors) with respect to predetermined quality criteria. The identified flaws are marked in the requirements document and briefly explained.

7.5.2 Inspection

Typical phases of an inspection

Inspections of software or any other type of product are done to systematically check development artifacts for errors by applying a strict process [Laitenberger and DeBaud 2000].

An inspection is typically separated into various phases [Gilb and Graham 1993]: planning, overview, defect detection, defect correction, follow-up, and reflection. For requirements validation, the planning, overview, error detection, and error collection phases are relevant (see principle 2, separating the identification and correction of errors in section 7.4.2). Individual preparation is an obligatory part of inspections. An inspection session usually serves the purpose of collecting and evaluating error indications. Occasionally, performing dedicated inspection sessions is omitted when performing inspections.

Planning

Among other things, the goal of the inspection, the work results that are to be inspected, and the roles and participants are determined during the planning phase.

Overview

In the overview phase, the author explains the requirements to be inspected to all team members so that there is a common understanding about the requirement among all inspectors.

Error detection

In the error detection phase, the inspectors search through the requirement for errors. Error detection can be performed individually by each inspector or collaboratively in a team. Individual inspection has the advantage that each inspector can concentrate on the requirements. On the other hand, team inspections have the advantage that communication between the inspectors creates synergy effects during error detection. During the course of error detection, any errors that are found are purposively documented.

Error collection and consolidation

In the error collection phase, all identified errors are collected, consolidated, and documented. During consolidation, errors that have been identified multiple times or errors that aren't really errors are identified. The latter can be the case if, for instance, an inspector makes wrong assumptions about a requirement or interprets some constraint the wrong way. Along with consolidation, the identified errors and correcting measures are documented in an error list. Inspections are also known as *technical reviews*.

Roles during inspection

For an inspection to be performed, the following roles must be staffed with suitable personnel:

- *Organizer:* The organizer plans and supervises the inspection process.
- *Moderator:* The moderator leads the session and ensures that the predetermined inspection process is followed. It is advisable to select a neutral moderator because the moderator could potentially balance out opposing opinions of authors and inspectors.
- *Author:* The author explains the requirements that he created to the inspectors in the overview phase and later on is responsible for correcting the errors identified.
- *Reader:* The reader introduces the requirements to be inspected successively and guides the inspectors through them. The role of the reader should be given to a neutral stakeholder so that the inspectors can center their attention on the requirements instead of on the interpretation of the author. Often, the moderator is also the reader.
- *Inspectors:* The inspectors are responsible for finding errors and communicating their findings to the other members of the project team.
- *Minute-taker:* This person takes minutes of the results of the inspection.

7.5.3 Walk-Through

Lightweight inspection

In requirements validation, a walk-through is a lightweight version of an inspection. A walk-through is less strict than an inspection and the involved roles are differentiated to a lesser degree. During a walk-through, at least the roles of the reviewer (comparable to the inspector), author, and minute-taker, and potentially the moderator, are staffed.

Discussion of the identified flaws in quality during a group session

The goal of a walk-through of requirements is to identify quality flaws within requirements by means of a shared process and to gain a shared understanding of the requirements between all the people involved. To prepare for a walk-through, the requirements to be validated are handed out to all participants and inspected. During the walk-through session, the participants discuss the requirements to be validated step-by-step, under guidance of the moderator/reader. Usually, the author of a requirement is the one who introduces the requirement to all other participants. This way, the authors have the opportunity to give additional information to the group along with the actual requirement (e.g., alternative requirements, decisions, and rationale for decisions). A minute-taker documents the flaws in quality that have been identified during the session.

7.5.4 Perspective-Based Reading

Check requirements from a defined perspective.

Perspective-based reading is a technique for requirements validation in which requirements are checked by adopting different perspectives [Basili et al. 1996]. Typically, perspective-based reading is applied in conjunction with other review techniques (e.g., during inspections or walk-throughs). Focusing on particular perspectives when reading a document verifiably leads to improved results during requirements validation. Possible perspectives for validation, for instance, emerge from the different addressees of a requirement [Shull et al. 2000]:

- *User/customer perspective:* The requirements are checked from the perspective of the customer or the user to determine whether they describe the desired functions and qualities of the system.
- *Software architect perspective:* The requirements are checked from the perspective of the software architect to ascertain if they contain all necessary information for architectural design (e.g., if all relevant performance properties have been described).
- *Tester perspective:* The requirements are checked from the perspective of the tester to establish whether they contain the information necessary to derive test cases from the requirements.

Perspective quality aspects

The three quality aspects (see section 7.3) also describe three possible perspectives for requirements validation:

- *Content perspective:* With the content perspective, the auditor verifies the content of requirements and focuses on the quality of the content of the documented requirement.
- *Documentation perspective:* With the documentation perspective, the auditor ensures that all documentation guidelines for requirements and requirements documents have been met.
- *Agreement perspective:* With the agreement perspective, the auditor checks if all stakeholders agree on a requirement, i.e., if the requirements are agreed upon and conflicts have been resolved.

In addition, further perspectives that emerge from the individual context of the development project can be created as need be.

Define validation directives for each perspective.

During perspective-based validation, each auditor is assigned a perspective (at the proper point in time) from which she reads and validates the requirement. For each perspective defined, detailed instructions for performing the validation should be laid down because the auditor might

not be familiar with all relevant details of her assigned perspective. It is advisable to associate questions with each validation instruction that must be answered by the content of the requirements or by the auditor after she has read the requirement, respectively. In addition, validation instructions can be amended with a checklist that summarizes the most important content aspects that ought to be addressed by a requirement with regard to the appropriate perspective.

Follow-up

During the course of the follow-up to a perspective-based reading session, the results of the chosen perspective are analyzed and consolidated. On the one hand, the results of the perspective-based reading contain answers to the predefined questions, and on the other hand, open issues that the auditors noticed while reading may be present. The consolidation can be done as a group effort, similarly to a review.

Support of other techniques

Perspective-based reading can be both an independent technique for requirements validation and a support technique for other validation techniques, such as inspections or reviews of requirements documents by means of perspective-based reading.

7.5.5 Validation through Prototypes

Requirements validation by means of prototypes allows auditors to experience the requirements and to try them out. Experiencing requirements directly through prototypes [Jones 1998] is the most effective method to identify errors in requirements. Stakeholders can try out the prototype and compare their own idea of how the system ought to be implemented with the prototype at hand and thereby find discrepancies between their ideas and the current implementation.

Evolutionary vs. throw-away prototypes

Depending on the further use of the prototype, one can distinguish between throw-away prototypes and evolutionary prototypes [Sommerville 2007]. Throw-away prototypes are not maintained once they have been used. Evolutionary prototypes are developed with the goal to be developed further and improved in later steps. In contrast to throw-away prototypes, implementation plays a much more significant role here. Therefore, the effort to create evolutionary prototypes is much higher.

Selection of relevant requirements

Before a prototype can be implemented, the requirements that shall be validated through the prototype must be selected. The set of requirements to be validated is limited by development resources (e.g., time, money, etc.) that can be allocated for validation. For example, a selection criterion can be the criticality of a requirement.

Preparation of the validation

The following preparations have to be made in order to validate requirements by means of prototypes:

- *Manual/instructions:* The users of the prototype must be supplied with the necessary information so that they can use or apply the prototype. This can be done by means of a manual or by means of proper instruction.
- *Validation scenarios:* Validation scenarios that the users of the prototype can perform with the prototype should be prepared. A validation scenario defines, for example, all relevant data sets or user interactions.
- *Checklist with validation criteria:* For requirements validation, a checklist of validation criteria should be created according to which the prototype (and by proxy, the requirements) can be validated.

Performing the validation

The auditor should validate the prototype without being influenced; i.e., the auditor should execute the validation scenarios independently and by herself. This ensures that the validation results are created without bias.

During validation, the auditors can and ought to execute alternative and deviant scenarios and should use the prototype exploratively and experimentally once the required validation scenarios have been covered. For example, error cases that have remained hidden until then can be identified. For experimental validation of the prototype, the auditor needs to know the scope of the prototype, i.e., the set of requirements that have been considered when the prototype was created. Without knowledge of the implemented requirements, an auditor cannot decide whether an identified error can be traced back to a missing requirement or if the requirement has been consciously omitted in the prototype.

Documentation of the validation results

Requirements validation through prototypes therefore permits two types of result documentation:

- *Protocol of the auditor:* The auditor documents the results and experiences made during the validation of the prototype, e.g., by means of validation scenarios as well as a checklist that he has been supplied with.
- *Observation protocol:* The auditor can be observed by a second person. The second person creates a so-called observation protocol. This protocol can disclose additional important symptoms for errors in requirements. For example, when the auditor hesitates at a certain step in the validation scenario while using the prototype and the observer

documents this, it may be an indication for missing apparentness and as such an indication for impaired understandability of the prototype. Under certain circumstances, it may be advisable to record the validation on video because the validation situation can be analyzed in detail during the follow-up. For example, a video recording can show the realization of requirements pertaining to anthropometric properties (such as ergonomics) or intuitive use and can be investigated in detail.

Analysis

The results of the validation are analyzed after validation is complete. Change suggestions for the requirements are consolidated. If significant changes to the requirements are necessary, it may be advisable to revise the prototype and validate anew.

7.5.6 Using Checklists for Validation

A checklist comprises a set of questions and/or statements about a certain circumstance. Checklists can be applied whenever many aspects must be considered in a complex environment and no aspect must be omitted. A checklist for requirements validation contains questions that ease the detection of errors [Boehm 1984]. Using checklists for requirements validation is very common in practice. Checklists can be used in all previously introduced techniques for requirements validation.

Creating checklists

Before a checklist can be used, every single question or statement must be defined. The sources for questions and statements in the following list can be used to create checklists to support requirements validation:

- The three quality aspects of requirements (see section 7.3)
- Principles of requirements validation (see section 7.4)
- Quality criteria for requirements documents (see section 4.5)
- Quality criteria for individual requirements (see section 4.6)
- Experiences of the auditors from prior projects
- Error statistics [Chernak 1996]

Improving checklists

Checklists are not necessarily complete. When using a checklist, one should always look for opportunities to improve the checklist for future use. For example, if a question was forgotten, the checklist should be amended to contain the extra question. Ambiguous questions or questions that are not understandable must be marked and revised. Outdated or no longer valid questions should be removed.

Checklists as a guideline

Checklists can support requirements validation in different ways. They can serve as a guideline for the auditor, who can follow the checklists at her own discretion (e.g., during a review).

Checklists as a means of structuring

The checklist can define a list of questions that must be strictly adhered to. These questions must be answered by the auditor to validate the requirements. In this case, the checklist serves as a measure to approach the validation in a structured manner. For example, the checklist may detail the exact process that the auditors are asked to apply, which guarantees that every auditor validates the requirements in the same way. This makes the results more comparable.

Hybrid forms of checklist application are also possible. For example, a checklist can contain obligatory questions for perspective-based reading and can contain suggestions that the auditor may or may not follow.

Successfully applying checklists

Applying checklists for requirements validation successfully depends on the manageability and complexity of the checklist. A large amount of questions can make it more difficult to use the checklist because the auditor does not have a steady overview of the questions and is thus forced to consult the checklist frequently. It is therefore advisable to design the checklist to not be longer than a single page [Gilb and Graham 1993]. In addition, questions that are formulated altogether too generically or abstractly can make it more difficult to use the checklist. For example, the question "Is the requirement formulated appropriately?" can lead to a multitude of different answers, depending on what the auditor considers an appropriately formulated requirement. The questions therefore ought to be as precise as possible.

7.6 Requirements Negotiation

To negotiate the requirements of a system to be developed, it is necessary to identify conflicts and to resolve those conflicts. This is done by means of systematic conflict management. The conflict management in requirements engineering comprises the following four tasks:

Four tasks of conflict management

- Conflict identification
- Conflict analysis
- Conflict resolution
- Documentation of the conflict resolution

These four tasks are explained in the following sections.

7.6.1 Conflict Identification

Conflicts can arise during all requirements engineering activities. For example, different stakeholders can utter contradicting requirements during elicitation.

Conflict identification in all requirements engineering activities

Conflicts between requirements and conflicts between stakeholders are often not obvious due to different reasons. During the entire requirements engineering process, the requirements engineer should pay attention to potential conflicts so that they can be identified, analyzed, and resolved early on.

7.6.2 Conflict Analysis

Determining the conflict type

During conflict analysis, the reason for an identified conflict must be determined. According to [Moore 2003], different types of conflicts exist.

Data conflict

A *data conflict* between two or more stakeholders is characterized by a deficit of information, by false information, or by different interpretations of some information. For example, take the following requirement: "R131: The reaction time of the planned system shall not exceed one second". A data conflict between two stakeholders with regard to this requirement can arise from the fact that one stakeholder considers a reaction time of 1 second to be too slow while another stakeholder does not believe that a reaction time of 1 second is feasibly implementable (i.e., it is too short).

Conflict of interest

A *conflict of interest* between two or more stakeholders is characterized by subjectively or objectively different interests or goals of stakeholders. A conflict of interest between two or more stakeholders can arise, for instance, when one stakeholder primarily focuses on keeping the costs of the planned system at a minimum while another stakeholder primarily desires a high level of quality. A conflict of interest between these two stakeholders arises when the first stakeholder rejects a requirement due to estimated costs and the second stakeholder insists on implementing it due to quality reasons.

Conflict of value

A *conflict of value* is characterized by differing underlying values stakeholders have regarding some circumstance (e.g., cultural differences, personal ideals). For instance, a conflict of value arises when one stakeholder favors open source technologies while another stakeholder favors closed sources technologies.

Relationship conflict

A *relationship conflict* is characterized by strong emotions, stereotypical relationship concepts, deficient communication, or negative inter-

personal behavior between stakeholders (e.g., insults, disrespect). For instance, a relationship conflict arises when two stakeholders of equal rank or position (e.g., department leaders) reject each other's requirements and try to distinguish themselves by forcing their requirements onto the project.

Structural conflict

A *structural conflict* is characterized by unequal levels of authority or power. For instance, a structural conflict can arise between an employee and his superior if the superior invariably rejects requirements that the employee has defined because he does not recognize the employee's competence to delineate requirements.

Mixed reasons for conflicts

Often, it is difficult to unambiguously classify emerging conflicts. For example, a conflict can be a relationship conflict with clear structural components. Similarly, a conflict of interest can be a conflict of values as well. Therefore, it is advisable to analyze an identified conflict with respect to all types so that all possible reasons for the conflict can be determined and suitable resolution strategies can be selected.

7.6.3 Conflict Resolution

Good conflict resolution is a success factor.

Conflict resolution is very important for requirements negotiation because the strategy of conflict resolution has a big influence on the willingness of the people involved (e.g., customers, consultants, or developers) to continue working together. For example, a conflict resolution considered unfair by at least one party of the conflict can lead to a decreased engagement and willingness to collaborate in the project. On the other hand, a resolution that is considered fair by all parties can increase the willingness to cooperate because this signals that everyone's ideas about the planned system are being considered.

Involvement of the relevant stakeholders

Independently from the selected resolution strategy, it is essential to involve all relevant stakeholders. If not all relevant stakeholders are considered, some opinions and viewpoints will remain unconsidered. The conflict will therefore only be resolved in part or incompletely. In the following paragraphs, different conflict resolution techniques are introduced.

Agreement

With the conflict resolution technique *agreement*, all conflict parties negotiate a solution to the conflict. The parties exchange information, arguments, and opinions and try to convince one another of each other's viewpoints in order to come to an agreeable solution.

Compromise

With the conflict resolution technique *compromise*, all conflict parties try to find a compromise between alternative solutions. In contrast to an

agreement, a compromise consists of an amalgamation of different parts of the alternative solutions. Also, a compromise can mean that all alternative solutions as proposed thus far are discarded and entirely new solutions are creatively developed.

Voting

In the conflict resolution technique *voting*, all conflict parties vote on solution alternatives. The alternatives that are up for voting are presented to all relevant stakeholders. Each stakeholder casts her vote for an alternative and the alternative with the most votes is accepted as the resolution for the conflict.

Definition of variants

In the conflict resolution technique *definition of variants*, the system is developed in a way that permits the definition of variants by deriving variants, by selecting parameters that define system variants, or by selecting variable system properties. This way, the system can satisfy the different interests of the stakeholders.

Overruling

In the conflict resolution technique *overruling*, a conflict is resolved by means of the hierarchical organization. This means that a conflict party with higher organizational rank or position wins the conflict by overruling objections of organizationally lower parties. If both parties have the same organizational rank, the conflict is resolved by a superior stakeholder or some third-party decider. This conflict resolution technique is only advisable if other resolution techniques have failed (e.g., no compromise could be found) or are not applicable due to limitations of resources (e.g., time).

Consider-all-facts

In the conflict resolution technique *consider-all-facts (CAF)*, all influencing factors of a conflict are being investigated so that as much information about the conflict can be collected as possible. This information is used during resolution. By prioritizing the influence factors, the relevance is determined. Based on the results of this technique, the plus-minus-interesting conflict resolution technique can be applied.

Plus-minus-interesting

In the conflict resolution technique *plus-minus-interesting (PMI)*, all positive and negative repercussions of a solution alternative are investigated so that positive and negative repercussions can be evaluated. Positive repercussions are placed in the category "plus" and negative repercussions are placed in the category "minus". Repercussions that are neither positive nor negative are placed in the category "interesting". Repercussions in the category "interesting" cannot be evaluated yet and must be investigated further to determine if their influence is positive or negative.

Decision matrix

In the conflict resolution technique *decision matrix*, a table is created that contains solution alternatives in the columns and all relevant decision criteria in the rows. The decision criteria can be identified by means of the

technique "consider-all-facts". For each combination of criterion and solution alternative, an assessment is made, for instance by means of a point-scale ranging from irrelevant (0 points) to relevant (10 points). Table 7-1 shows a decision matrix.

	Solution alternative 1	Solution alternative 2	Solution alternative 3
Criterion 1	3	6	2
Criterion 2	5	4	10
Criterion 3	10	3	5
Sum	18	13	17

Table 7-1 *Decision matrix*

In order to find a solution, the sums of the columns are calculated; i.e., the assessments of the criteria of each solution alternative are summed up. The solution alternative with the highest score is accepted as the decision. In the example shown in table 7-1, this would be solution alternative 1.

7.6.4 Documentation of the Conflict Resolution

Risks of missing conflict documentation

Conflicts cannot be avoided during requirements engineering. A resolution to a conflict must always be traceably documented. If a conflict resolution is not properly documented, the following threats (among others) to the project may arise:

- *Handling conflicts repeatedly:* A certain conflict can arise a second time during the requirements engineering process. Without proper documentation of the conflict resolution, the conflict must be analyzed and resolved anew. This causes additional effort and can potentially lead to additional conflicts or abrogate previous resolutions.
- *Inappropriate conflict resolution:* During the requirements engineering process, the resolution of a conflict can turn out to be wrong or unsuitable. In this case, the conflict must be investigated and resolved anew. Without proper documentation, relevant information that has been considered during the initial analysis and resolution can be overlooked and the new conflict resolution can once again lead to false results.

In both cases, proper documentation of the conflict and its resolution supports the requirements engineering process and ensures that relevant information already known can be considered.

7.7 Summary

The quality of the elicited and documented requirements must be assured during requirements engineering so that it can be guaranteed that the requirements meet the desires and ideas of the stakeholders adequately. Therefore, it is necessary to validate the requirements with regard to the quality of their content, their documentation, and their agreement with respect to the different stakeholders. There are different techniques that can be selected and purposively combined for requirements validation, depending on the project peculiarities and project goals. Among the most common validation techniques for requirements are the different types of requirements reviews (e.g., commenting, inspection, walk-through) as well as perspective-based reading and validation through prototypes and checklists.

For requirements negotiation, it is necessary to identify conflicts between stakeholders, analyze them, and resolve them in a suitable manner. A systematic conflict management supports analysis and resolution of the conflicts that have been identified over the course of requirements validation or other requirements engineering activities.