Requirements Engineering

2018/2019
Course 7

Course 7 outline

- Quality Assurance for Requirements
- Requirements Negotiation

Introduction

- Requirements engineering is the initial part of the software development process, and all later steps of the development are influenced by the requirements.
- The quality of the requirements is an important factor for the overall quality of the developed system.
- Quality assurance (QA) is an important but elusive part of software development.
- Traditionally, QA techniques have mainly focused on the later development phases such as the implementation phase and the related testing activities.
- It is common to do QA only by means of testing (and maintenance approaches), called an opportunistic approach.
- Many studies show that late, opportunistic QA leads to stressful and costly test and maintenance phases.

Introduction

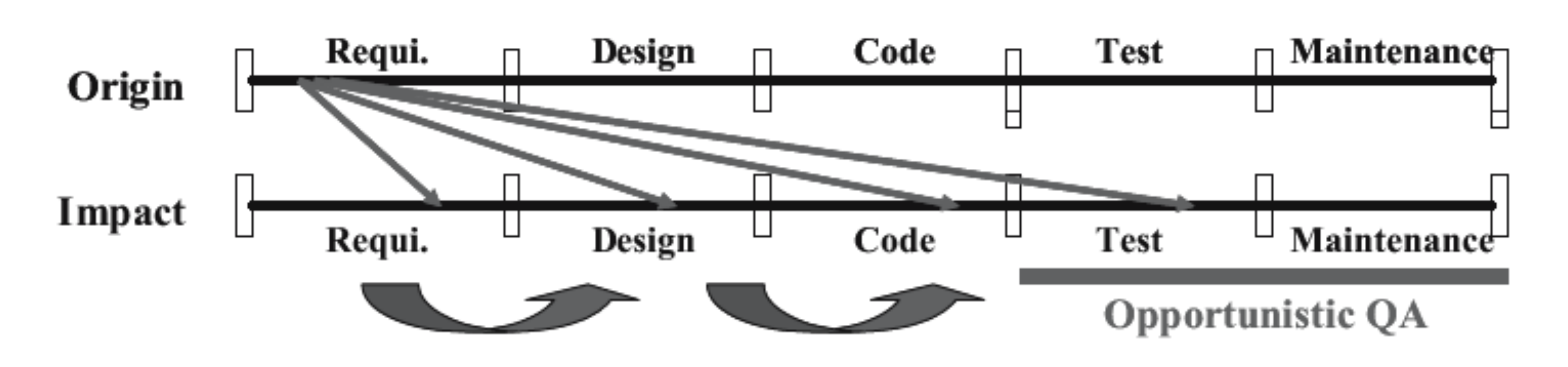
- If the quality assurance is only performed in the test and maintenance phase, the success of the project is dependent on the ability of the requirements engineers, designers and programmers to produce good working products, suitable for the rest of the development.
- Without intermediate QA (a quality gate for the intermediate work products), it may be possible that the design and implementation are based on the wrong requirements.
- It may lead to high rework effort as not only the code but most often the overall system architecture and design have to be revised due to requirements defects.
- Requirements deficiencies are the prime source of project failures.
- Over 40% of problems in the software development cycle result from low quality requirements.

Introduction

- QA should start earlier.
- Issues should be resolved in the phase of their origin to avoid costly testing and rework.
- Testing and rework can account for up to 40-50 % of the development effort.
- Removing defects early in the development process is more cost effective than addressing the defects during testing or maintenance.
- Correcting a defect late in the process gets more expensive as development effort has already been spent and more artifacts are affected.
- A requirements issue can become up to 100 times more expensive if it is detected during design or implementation, compared to detecting it in the requirements phase.

Introduction

- QA techniques for requirements are one of the most promising and cost effective techniques to ensure successful development and to prevent avoidable rework in later phases.
- QA in the requirements phase should be performed whether high quality is required or not.
- QA becomes even more important if high quality is a key success factor.



Quality Assurance

- The term issue is used as an umbrella term for all matters that should be resolved in the requirements context.
- The terms defects, errors, faults or problems are other words used with a similar meaning.
- In the case of requirements, it is sometimes unclear whether an issue really is a defect.
- Example:
  - if two stakeholders disagree on one aspect of a requirement, this is an issue that should be resolved, but would usually not be referred to as a defect in the traditional sense.
  - If it is not resolved, at least one stakeholder will reject the system in acceptance test.
- Contradicting requirements are closer to the conventional interpretation of a defect.

Quality of Requirements

- The quality of requirements is dependent on the various stakeholders and their perspective.
- Several different views need to be considered in order to define what quality means in a certain context:
  - user view: evaluates the quality of a software product with respect to its fitness of purpose to fulfill certain user tasks.
  - manufacturing view: focuses on the product view during production and after delivery (i.e., adherence of standards). It evaluates whether the product was build right the first time.
  - product view: The focus is on internal quality aspects of the product that can be measured. It is assumed that ensuring certain internal quality aspects has an impact on the external quality and the quality in use of the product.
  - value-based view. It relates quality to cost. It considers quality as something the customer is willing to pay for.

Quality of Requirements

- Mapping the views on the quality of requirements reveals relevant stakeholders and needed QA for the requirements.
  - user view: the requirements should describe what the user requires of the final system.
  - product view: they should be described in a way that allows the developers to produce the software effectively and efficiently.
  - manufacturing view: the requirements engineers have to follow certain standards when specifying the requirements to ensure the quality of the requirements right from the start.
  - value-based view: the customers have to decide on the value of each requirement and whether the implementation cost is motivated.

Quality of Requirements

- Standards are a starting point for defining the quality of requirements and requirements specifications.
- There exist a number of processes, guidelines, and best practices on how to perform good requirements engineering.
- The advocates of these approaches argue that adhering to the process facilitates requirement engineering and minimizes later quality problems.
- The IEEE 830 standard for requirements specification is used as a starting point for an initial set of quality criteria.
- The standard may be extended to provide a more complete picture of relevant quality aspects of requirements, especially to address customer and user needs (value-based and user-view on requirements quality).

Quality attributes for requirements

- Correctness (IEEE, user-view):
  - The requirements that are implemented have to reflect the expected (intended) behavior of the users and customers.
  - Everything stated as a requirement is something that shall be met by the final system to fulfill a certain purpose (suitability).
- Unambiguity (IEEE, product-view):
  - The requirements should only have one possible interpretation.
  - One requirement might be unambiguous to a certain group of stakeholder but has a different meaning in another. It is important to involve all stakeholders in the requirements engineering process to gain a common understanding.
- Completeness (IEEE, product-view):
  - All important elements that are relevant to fulfill the different user's tasks should be considered. This includes relevant functional and non-functional requirements and interfaces to other systems, the definition of responses to all potential inputs to the system, all references to figures and tables in the specification, and a definition of all relevant terms and measures.

Quality attributes for requirements

- Consistency (IEEE, product, manufacturing - view):
  - The stated requirements should be consistent with all other requirements, and other important constraints such as hardware restrictions, budget restrictions, etc.
- Ranked for Importance / Stability (IEEE, product, value-based, user view):
  - Each requirement specifies its importance and/or its stability. Stability expresses the likelihood that the requirement changes, while importance specifies how essential the requirement is for the success of the project (from a value-based and a user point of view).
- Verifiability (IEEE, product view)
  - All requirements should be verifiable.
  - There exists a process for a machine or a human to check (in a cost effective way) whether the requirement is fulfilled or not.

Quality attributes for requirements

- Modifiable (IEEE, product view):
  - All requirements should be modifiable.
  - The structure of the requirements and the requirements specification allow the integration of changes in an easy, consistent and complete way.
- Traceable (IEEE, manufacturing view):
  - All requirements should be traceable, that is, it should be possible to reference the requirement in an easy way.
  - It should be possible to identify the origin of a requirement.
- Comprehensibility (New, manufacturing, user, value-based view)
  - The requirements are specified and phrased in a way that is understood by all involved stakeholders.

Quality attributes for requirements

- Feasibility (New, value based, product view):
  - All requirements can be implemented with the available technology, human resources and budget.
  - Moreover, all requirements contribute to the monetary success of the system, that is, they are worth to include in the system.
- Right Level of Detail (New, user, manufacturing, value-based view):
  - The information given in the requirements is suitable to gain the right understanding of the system and to start implementation.
  - There are no unnecessary implementation or design details specified in the requirements.
- Relationships among the attributes:
  - ambiguous requirements are also difficult to understand.
  - if the requirements are not traceable, the verifiability, modifiability and the comprehensibility can be affected.

Requirements Quality Strategy

- Developing software without any defects is impossible.
- It is possible to achieve an optimal compromise between the desired quality and available resources, considering the specific context factors and quality need of a company or a project.
- Many factors influence the importance of different quality attributes in a specific context.
- During the requirements engineering phase, it is important to define a quality strategy that addresses those quality issues that can easily be verified and validated in the requirements phase.
- Other quality aspects that cannot be efficiently addressed during the requirements phase should be left for later phases.

Requirements Quality Strategy

- A quality strategy defines how, when and where different QA approaches, in combination with other approaches in the software development process, are used to assure high quality.
- This includes the planning of resources (which approach is applied when and how much effort should be spent) and the definition of an optimized combination of the different QA approaches with the aim of achieving the desired quality at the desired cost.
- Technical aspects of quality assurance:
    - Basic strategies. They represent those strategies in place in a company or a project that define how to perform QA in the requirements phase. In that sense they represent the current state of the practice in a certain context.

Due to the lack of sophisticated quality strategies, ad-hoc approaches are most frequently applied.

Requirements Quality Strategy

- Technical aspects of quality assurance:
  - The simplest but also the least systematic strategy is to state that everything in the requirements specification should be verified or that all quality issues should be tackled in later development phases.
  - Experience based strategies give hints on what to address in the requirements based on the experience of earlier projects. Such basic strategies should be considered when creating a more sophisticated quality strategy. They provide valuable input on where to start from and what has paid of in the past.
- Coverage criteria. It defines which aspects of the requirements should be covered by the QA approach.

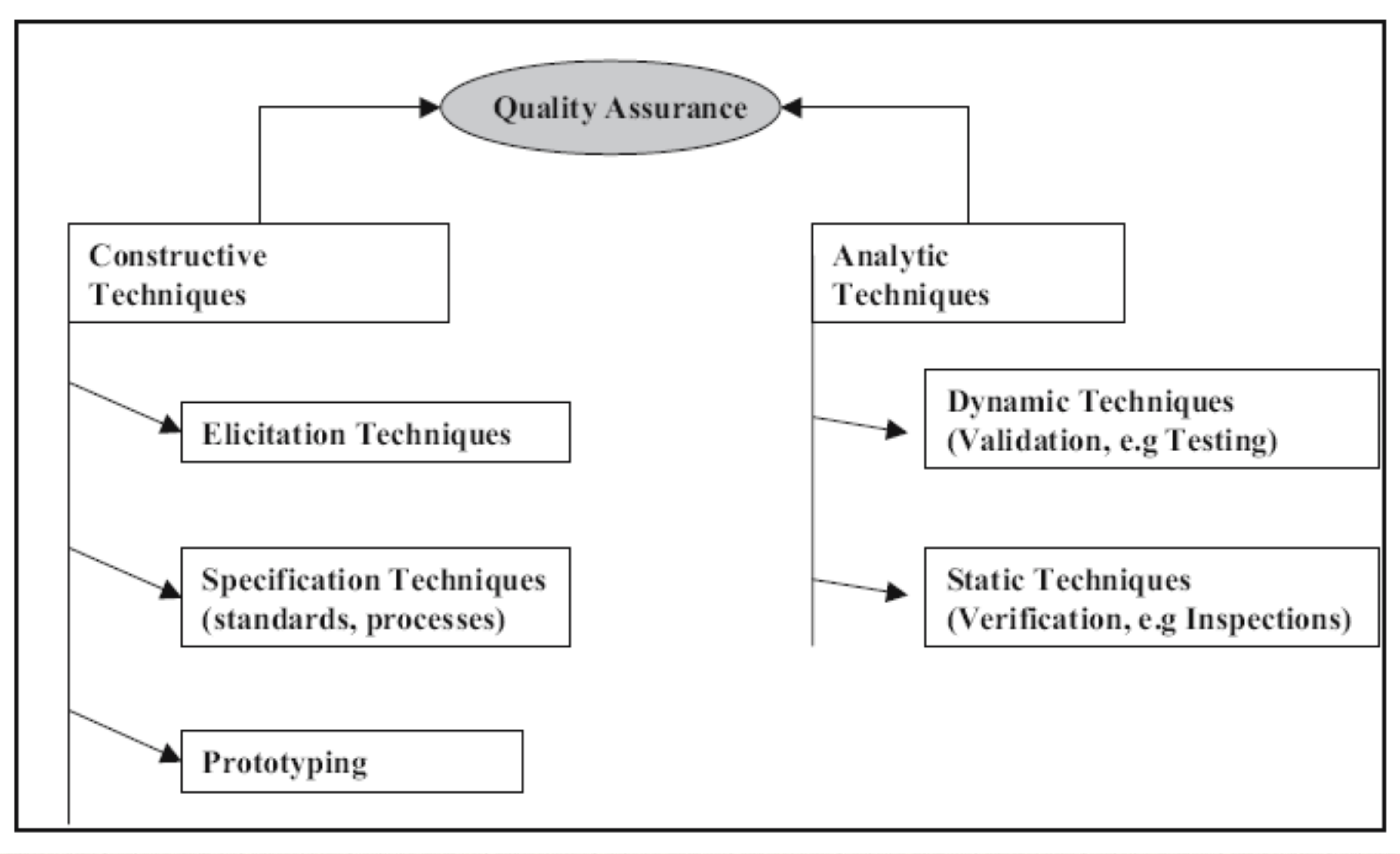
One example of a coverage criterion is that all requirements are covered by at least one test case.

An aspect related to coverage that should be considered is the depth of the QA approach. Depth defines the level of detail to which the requirements are verified or validated or the quality level to be achieved. The greater the depth, the more resources are required for QA and the more sophisticated QA approaches are required.

Requirements Quality Strategy

- Technical aspects of quality assurance:
  - The most important elements of a requirements quality strategy are the potential quality assurance approaches and methods that can be used to ensure the different quality characteristics of the requirements.
- The QA approaches are the technical core element of the quality strategy as they represent the means of achieving good requirements quality.

QA Approaches for Requirements



QA Approaches for Requirements

- Constructive approaches ensure that mistakes are minimized during the creation of a work product (e.g. the requirements specification).
- They prevent issues from being introduced.
  - Examples: style guidelines on how to specify requirements, templates for the requirements specification, elicitation approaches and prototyping.
- Analytical approaches are performed on the completed artifact or a self contained part of it with the aim to detect issues.
- The latter approaches can be further divided into static quality assurance approaches and dynamic quality assurance approaches (including formal methods).
- Dynamic approaches require an executable version of the system.

  E.g. Testing
- Static quality assurance approaches can be performed without executing code.

  E.g. Inspections and formal verifications

QA Approaches for Requirements

- In most cases there is no executable code available during the requirements engineering phase.
- Usually only static approaches are applicable during requirements.
- QA in the requirements phase means that requirements issues are prevented from being introduced (i.e. during elicitation).
- With the help of constructive approaches omissions and ambiguous requirements are addressed.
- The validation process of requirements is based on a requirements document and tries to resolve issues within this document. Here, the analytical approaches are applied.

Elicitation Techniques

- It is a constructive approach.
- The elicitation step is important to the overall quality of the requirements and the acceptance of the final system.
- Quality attributes ensured using elicitation techniques:
  - *Comprehensibility*: by developing a common terminology and ensuring that the different stakeholders speak the same language, comprehensibility is improved.
  - *Completeness*: if the elicitation is performed correctly, all the (relevant) stakeholders, and their individual stakes, should be identified. Elicitation activities contribute to higher quality in that they support the requirements engineers in the identification processes.
  - *Verifiability* and *feasibility*: by involving the relevant stakeholders, quality can be assured. By involving the testers the attribute verifiability is improved, and by involving the developers feasibility is improved.
  - *Correctness*: the elicitation process should be driven by the business concerns. Suitability, as part of correctness, is supported by this, as it is then more likely that the developed software will bring a real financial benefit in the context of use.

Specification Techniques

- It is a constructive approach.
- Certain specification techniques, best practices and standards can help to ensure the quality of the requirements.
- Standards and templates contribute to better requirements (considering the quality characteristics), in the following way:
  - *Completeness*: in the case that the requirement engineers adhere to the recommendations in the standards and apply the pre-defined templates it can be ensured that all relevant aspects of a requirements document are considered, i.e. completeness of the document.
  - *Understandability* and *modifiability*: the structure provided by templates and standards ensures that requirements document look similar over different projects in a company. Standardization of requirements documents prevents ambiguities within the documents and improves the understandability as well as the modifiability, as elements that need to be changed can be found more easily.

Specification Techniques

- Specifying functional requirements using use cases and related scenarios ensures also the comprehensibility of the requirements from the start, as use cases and scenarios are easy to understand for technical and nontechnical stakeholders.
- This also supports the attribute right level of detail.
- In addition, use cases seem to be valuable source for the definition of acceptance and system test cases.
- Specifying the requirements in a structured, scenario-oriented way improves their verifiability.

Prototyping

- It is a constructive approach that can be used to support elicitation.
- A prototype is an executable version of the system under development, though restricted in one way or another.
- Examples:
  - A user interface prototype implements parts of the user interface, the structure and navigation, but will not have all the functionality.
  - A performance prototype focuses on memory and CPU load and might have no user interface at all.
- A prototype typically targets the following quality attributes:
  - *Inconsistencies* and *incompleteness*: the process of developing a prototype will, in itself, reveal inconsistencies and incompleteness of the requirements and thus improves their quality.
  - *Correctness*: correctness is improved by letting the different stakeholders work with and evaluate a concrete object rather than the abstract requirements.
  - *Feasibility*: by trying out different solutions, already in the requirements phase, feasibility is improved. A lot of time and money can be saved if dead-ends are detected at an early stage.

Requirements Inspections

- An analytical approach.
- Inspections are a valuable means to ensure the quality of a software product right after its creation.
- In general, inspections aim at minimizing the issues of a certain product being propagated to later phases, as the issues are addressed in the same phase in which they are introduced.
- Requirements inspections are one of the most cost effective QA approaches, as they prevent issues from being propagated from the requirements to other artifacts and cause follow-up defects and avoidable rework.
- Elements of an inspection:
  - process
  - roles
  - reading techniques
  - the information on how the results of the inspection are documented.

The Inspection Process

- A basic inspection process contains four main steps:
  - planning (managing the organizational issues of an inspection),
  - detection (inspectors search for issues in the document under inspection),
  - collection or meeting step (moderated meeting merging the results of the inspectors into approved defect list)
  - correction (where the author has to resolve all the identified issues).
- These steps are common for almost all instantiations of the inspection process.
- Several inspection processes mention additional steps such as the overview meeting or the follow-up meeting.
- Each phase of the process can be implemented in different ways depending on the level of detail with which the requirements should be inspected.
  - Eg, in the case that the requirements should be checked only from an abstract viewpoint, the individual preparation phase of the process could be skipped and the requirements would be discussed during a meeting with certain experts.

Reading Techniques

- The detection step is the most important, but also the most difficult step, in a requirements inspection.
- In this step, the inspectors identify requirements issues.
- A reading technique supports the inspectors in performing this step.
- It represents a series of steps or procedures that guide an inspector in acquiring a deeper understanding of the requirements under inspection and detecting issues in them.
- There are different kinds of reading techniques that can be used during a requirements inspection:
  - ad-hoc reading (reading without further guidance based only on experience),
  - checklist-based reading (using a list of questions to point to potential issues in the requirements),
  - scenario-based reading (using a step-wise description to guide the inspector during the defect detection step).

Checklist based reading

- It is based on checklists containing questions that should be answered during the issue detection.
- These questions focus on certain quality aspects that are relevant for the requirements under inspection.
- The checklist approach tells an inspector what to check.
- The reviewers get no guidance or hints on how to answer the questions in the checklist.
- There exist no standard checklist that can be applied in all contexts (e.g., checklist for use cases).
- A checklist is company- and sometimes even project-specific.
- The checklist has to be tailored to the context and characteristics of the company and the project.

Checklist based reading

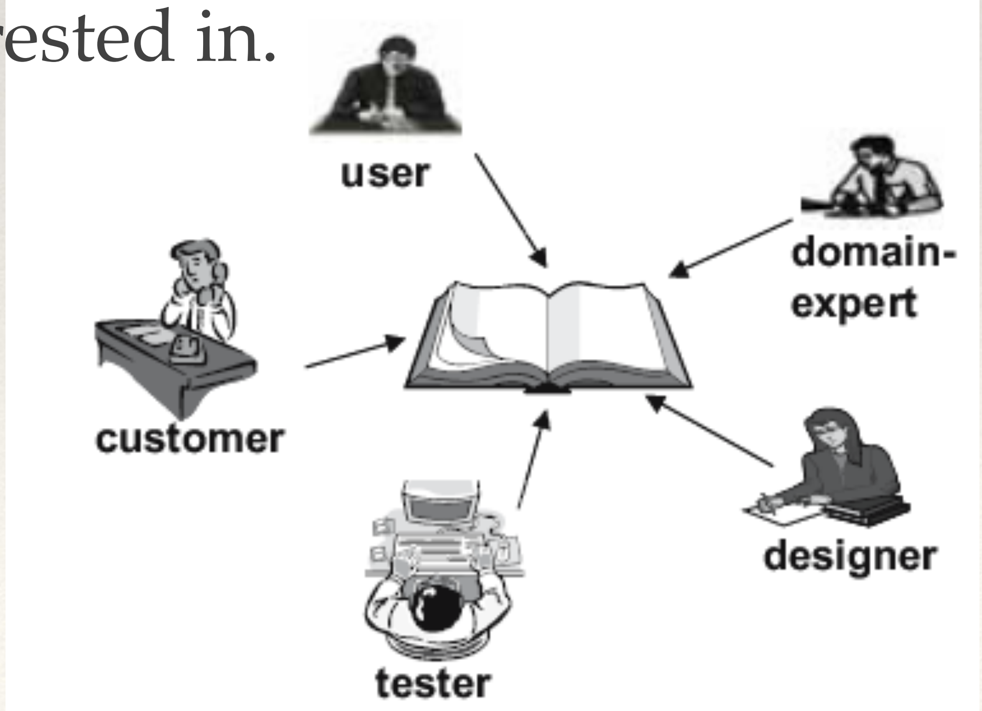
- Weaknesses:
  - The checklist questions are often extremely general.
  - The checklist questions are often not up to date.
  - It provides little support for how to perform the analysis. Concrete guidance on how to use the checklist is missing.
- Alternative approaches (scenario-based approaches) were developed.

Scenario-based Reading

- Inspectors are guided by a scenario that tells them what to look for during the inspection and how to perform the inspection.
- The scenario guides the inspector to actively work with the requirements, resulting in a deeper understanding of the requirements and their interrelationships.
- The scenarios focus the attention of the inspectors on the essential quality aspects and on the essential parts of the requirements under inspection that need the most thorough investigation (i.e., taken from prioritization techniques).
- Scenario-based approaches:
  - Perspective-based reading
  - Traceability-based reading
  - Defect-based reading
  - Usage-based reading

Perspective-based reading (PBR)

- The requirements are inspected from the viewpoint of different stakeholders.
- Different stakeholders have different interests in the requirements.
- Assumption: the requirements are of good quality if all stakeholders who use the requirements for their specific tasks, agree on the requirements quality (find no serious issues in them).
- In each company context, the involved perspectives are different. The first essential step is identification of the potential perspectives and the quality concerns these perspectives are interested in.



Traceability-based reading

- During an inspection traceability links can help to guide the inspectors through the requirements.
- Eg., the quality attribute of consistency is directly related to the ability to trace one requirement to another.
- Having well defined links between the requirements, it is possible for the inspector to follow these links and check that the requirements work together in a consistent and correct way.
- The defect detection step gets more efficient as the inspectors do not have to think of potential relationships between requirements but can follow the links between them.
- It is also possible for the inspector to judge whether certain functions are completely realized with the different requirements described in the specification by following the traceability links and judging whether the sum of the requirements results in the desired support for the user.
- Finally, the traceability links indicate requirements that are highly related to each other and therefore help the inspectors to judge the maintainability and understandability of the requirements.

Requirements-Based Testing

- Testing is usually performed at the end of the development process when executable system parts are available.
- Test cases are usually defined and run on the system to validate whether the system fulfills its specification.
- The test cases derived from the requirements are used during the acceptance and system test phase.
- Testing is often perceived as the pure execution of the test cases at the end of the development cycle.
- Testing is more than running the test cases and looking for failures in the final software.
- Two steps, test planning and test case creation can and should be integrated in the development process much earlier than they are usually integrated.
- It is recommended that test planning and test case creation should be performed as soon as the requirements, or a self-contained sub-set, are defined.

Requirements-Based Testing

- The idea of early test case creation is similar to the idea of perspective-based inspections.
- Through the early construction of the test cases, the test engineers gain a better understanding of the requirements and are able to identify weaknesses and potential issues within the requirements.
- Test engineers bring in a completely new perspective on the requirements which also contributes to identify requirements issues during the early test case creation.
- For example, if the test engineers have difficulties in deriving the acceptance test case from requirements it might be necessary to refine the requirements, to add missing information or to remove/restate the requirement as it is not possible to test them.
- Early test case creation helps to improve the quality of the requirements by identifying correctness, completeness, ambiguity, consistency and verifiability issues during the specification of the test cases.

Automated Approaches and Formal Methods

- Due to the abstract and informal nature of most requirements documents it is difficult to apply any automated tools to ensure their quality.
- For simple issues, such as grammar or spelling defects, there are tools available. Removing such issues from the requirements typically improve their comprehensibility.
- For one quality attribute, unambiguity, more tool support is available. The idea of tools that address ambiguity flaws is the identification of certain patterns and keywords in the requirements that point to potential risk areas (i.e. areas where more than one interpretation of the requirements is possible).
- These tools identify, based on a glossary, phrases that are marked as weak or subjective, for example, "if possible", "may", "could", "optionally", etc. The tools parse the requirements document based on the pre-defined glossary and provide a list of all occurrences of the weak-phrases in the document.

Automated Approaches and Formal Methods

- Further automation is possible when the requirements are defined in a formal way.
- The use of formal languages deals with requirements issues by avoiding the imprecise nature of natural language.
- Requirements are specified in a semantically well-defined way, typically mathematically based.
- Benefits of using formal methods:
  - The communication between the stakeholders is more precise, and thus, misunderstandings and ambiguities can be reduced.
  - It is possible to check the completeness and the consistency of the requirements document, and automated proof of safety properties is possible.
  - The requirements engineer can perform simulations of the future system, when the language is supported by a tool.
- Drawbacks:
  - They are difficult to learn and difficult to understand for a person without the necessary background (clients, end-users).
  - The first version of the requirements might be formulated in natural language (for the customer), and then be translated into the formal version.

Requirements Negotiation

- Conflicts play an important role in software engineering although they are often neglected or badly handled by existing development methods.
- Conflicts arise almost inevitably as project stakeholders such as future system users, customers, developers, or maintainers frequently pursue mismatching goals:
    - Future system users are typically interested in many features, high level of service, or early availability.
    - Customers focus on cost effectiveness, compliance with standards, or budget/schedule constraints.
    - Developers typically want flexible contracts and stable requirements.
- Many existing developing methods neglect or do not explicitly address conflict handling and resolution.

Requirements Negotiation

- Software engineering is a highly collaborative process and identifying shared or opposed interests is a necessity for project success.
- The objectives of customers, users, or developers have to be understood and reconciled to develop mutually acceptable agreements.
- It does not mean that stakeholders will always agree.
- The result of negotiation is also to understand why stakeholders disagree.
- Identified disagreements represent major risks and need to be addressed by project management.
- The primary purpose of requirements negotiation is to identify and resolve conflicts among stakeholders.
- It contributes to the goal of defining feasible and mutually satisfactory requirements that accommodate all stakeholders goals and expectations.

Requirements Negotiation

- Issues to be addressed:
  - How can conflicts be identified?
  - How can the identified conflicts be resolved?
  - How can stakeholders find feasible alternatives?
  - Who is in charge of the negotiation (the stakeholders themselves or a facilitator)?
  - How can the negotiation be supported with tools or other means?
- Requirements negotiation can make use of negotiation methods and tools from a wide range of disciplines and domains.
- There are different perspectives on negotiation and different aspects are emphasized.

Requirements Negotiation

- Negotiation is traditionally viewed as "the actual interactions among participants that lead to mutual commitment" starting "when participants begin communicating their goals, and ending (successfully) when all agree to a specified contract."
- A requirements engineering perspective: "in general terms, requirements negotiation can be seen as an iterative process through which stakeholders make tradeoffs between requested system functions, the capabilities of existing or envisioned technology, the delivery schedule and the cost."

Curtis, B., Krasner,H., Iscoe, N.,, A field study of the software design process for large systems, Communications of ACM, 1988

Negotiation Process

- Three general phases:
  - pre-negotiation
  - negotiation
  - post-negotiation
- Pre-negotiation: important activities
  - the definition of the negotiation problem,
  - the identification and solicitation of stakeholders,
  - the elicitation of goals from stakeholders,
  - and the analysis of goals to find conflicts.

The results of this phase are the issues and conflicts involved.

An issue: is "a topic of discussion that is of particular interest in a negotiation. Each issue has a range of alternatives or options, one of which must ultimately be agreed upon by the negotiators in order to achieve a compromise."

Kersten GE, Noronha SJ, Negotiations via the World Wide Web: A crosscultural study of decision making. 1999

Negotiation Process

- Negotiation:
  - This phase involves the actual conduct of the negotiation and the definition of agreements.
  - Based on the elicited goals and the identified conflicts stakeholders seek mutually beneficial solutions that are acceptable to all parties.
  - This activity is about structuring issues and developing alternatives to solve problems.
  - Depending on the type of conflict and problem at hand different strategies can be adopted for dealing with the conflicts (competing, collaborating, compromising, avoiding, etc).
  - This involves trade-offs in which stakeholders give up partly on some issues so as to gain on other issues.
  - Examples
    - problem-solving by identifying and adopting solutions that satisfy the goals of the parties;
    - persuading other negotiators to concede.

Negotiation Process

- Post-Negotiation:
  - In this phase stakeholders (or automated tools) analyze and evaluate the negotiation outcomes and suggest re-negotiation if necessary.
  - For example, it can be determined if the current agreement satisfies the preferences of the counterparts and if a better solution would be possible for one negotiation party, without causing loss to the other side.
  - It can also involve quality assurance reviews of the negotiation results.
  - Another important aspect of post-negotiation is to secure commitment of stakeholders over time. For example, by monitoring existing agreements and initiating re-negotiation in case agreements become obsolete due to new developments.

Negotiation Tool Support

- EasyWinWin
  - It is a requirements negotiation approach.
  - It is based on Boehm's negotiation model.
  - The individual objectives of stakeholders are captured as win conditions.
  - Conflicts among win conditions, risks, and uncertainties are recorded as issues.
  - Options are proposed to reconcile issues.
  - Agreements are developed out of win conditions and out of options by taking into account the preceding decision process and rationale.
  - The tool helps a team of stakeholders to gain a better and more thorough understanding of the problem and supports cooperative learning about others' viewpoints.
  - The requirements negotiation approach also includes steps for elicitation and analysis.
  - http://csse.usc.edu/csse/research/easy_win_win/