



Risks Implied by Bad Requirements Specifications













3/-12





12/42

9/12

In the following we shall discuss: ... in the sense of "finding out what the exact requirements are" "Analysing an existing requirements/feature specification" — later. Recall: one and the same content can serve both purposes; only the title defines the purpose then. (ii) kinds of requirements (i) desired properties of To maximise confusion, we may occasionally (inconsistently) call it requirements specification or just specification – should be dear from context... Note: In the following (unless otherwise noted), we discuss the feature specification i.e. the document on which the software development is based. requirements specification documents, hard and soft. open and tacit. functional and non-functional. requirements specifications, (iv) documents of the requirements analysis: (iii) (a selection of) analysis techniques dictionary, requirements specification ('Lasternheft'), feature specification ('Pfichtenheft') 13/12

A requirements specification should be complete call/equirements (existing in somebody's fail/equirements (or ...) should be present. correct – it correctly represents the wishes/needs of the customer, relevant – things which are not relevant to the project should not be constrained. traceable, comprehensible the sources of requirements are documented, requirements are uniquely identifiable, neutral, abstract - a requirements specification does not constrain the realisation more than necessary.

Requirements on Requirements Specifications

Requirements Analysis...

consistent, free of contradictions - each requirement is compatible with all other requirements, otherwise the requirements are not realisable. testable, objective

 the final product can objectively be checked
 for satisfying a requirement.

Correctness and completeness are defined relative to something which is usually only <u>in the customer's bad</u>. \rightarrow is is difficult to be sure of correctness and completeness.

- "Dear customer, please tell me what is in your head" is in almost all cases not a solution! It's not unusual that even the customer does not precisely know ...!
 For example, the customer may not be aware of contradictions due to technical limitations.

14/42

14/42

Requirements on Requirements Specifications



Requirements on Requirements Specification Documents

The representation and form of a requirements specification should be:

understand the requirements medication	all affected people should be able to	not unnecessarily complicated –	 easily understandable, 	
need inne care any affort	specification should be easy and should not	creating and maintaining the requirements	 easily maintainable – 	

 Cprecise. The requirements specification should not introduce new unclarities or rooms for interpretation (→ testable, objective).
 easily usable – storage of and access to the requirements specification should not need significant effort.

It is not trivial to have both, low maintenance effort and low access effort.

15/42

value low access effort higher. a requirements specification document is much more often read than changed or written (and most changes require reading beforehand).

"the list of participants should be sorted by immatic • Precise, non-abstract: "the list of participants should be sorted by Precise, abstract: lowest number first'

where T is the type of participant records, c compares immatriculation number numerically." public static <T> void Collections::sort(List<T> list, Comparator c);

A requirements specification should always be as precise as possible (--) testable objectivel timed not devote eacity over solution; precisely characterizing acceptable solutions is often more appropriate.
 Being too specific may limit the design decisions of the developers, which may cause unnecessary costs.

Idealised views advocate a strict separation between requirements ("what is to be done?") and design ("how are things to be done?").

16/42

Content

Pitfall: Vagueness vs. Abstraction

Consider the following examples:

Vague (not precise):

"the list of participants should be sorted conveniently"

Introduction

Vocabulary: Requirements (Analysis)
 Importance of Requirements Specifications

Requirements Specification
 Requirements Analysis
 Desired Properties
 Kinds of Requirements
 Analysis Techniques

Documents
 Dictionary

→• Specification

17/-0

Requirements Specification Languages





(A Selection of) Analysis Techniques

24/-0

Content

25/12

26/42

Tell Them What You've Told Them...

Requirements Documents for important lage, for • negotation design & implementation documentation testing, delivery, re-use, re-implementation.

A Requirements Specification should be • correct, complete, relevant, consistent, neutral, traceable, objective.

 Note vague vs. abstract
 Requirements Representations should be easily understandable, precise, easily maintainable, easily usable

Distinguish hard / soft functional/ non-functional open / tacit

It is the task of the analyst to elicit requirements.
 Natural language is inherently imprecise, counter-measures:
 natural language patterns.

Do not underestimate the value of a good dictionary.

40/12

References

41/42

References

Arenis, S. F., Westphal, B., Detsch, D., Muitz, M., and Andiaha, A. S. (2014). The wireless fire alarm system Ensuing conformance to industrial attandards through formal welf-relation. In Jones, C. B., Philapsaud, P., and Sun, J., entors, JM 2014: *Cime Wirelow-19: Philipsene Technology Symposium Singapore*, *May* 12-16, 2014. Proceedings, volume 8442 of LNCS, pages 658–672. Springer.

Boehm, B. W. (1979). Guidelines for verifying and validating software requirements and design specifications. In EURO IFIP 79, pages 711–719. Elsevier North-Holland.

Brooks, F. P. (1995). The Mythical Man-Month: Essays on Software Engineering. Anniversary Edition. Addison-Wesley.

Garchua, R., Ma, L., Nusselbeh, B., Piwek, P., de Roeck, A., Rouncefield, M., Sawyer, P., Willis, A., and Yang, H. (2009), Making tacht requirements explicit, talk.

IEEE (1990). IEEE Standard Glossary of Software Engineering Terminology. Std 610.12-1990.

IEEE (1998). IEEE Recommended Practice for Software Requirements Specifications. Std 830-1998.

Ludewig. J. and Lichter, H. (2013). Software Engineering. dpunkt.verlag. 3. edition. Rupp, C. and die SOPHISTen (2009). Requirements-Engineering und -Management. Hanner. Sthe dition.

42/-0