Requirements Specification

Ze Shi (Zane) Li

thesegalgroup.org

lize@uvic.ca

Outline

Why have a requirements specification/document

Quality attributes for requirements and reqts specification

RD Template

Requirements Document - Project Name (date created)

Revision History

Name	Date	Reason for Changes	Version

Requirements Documentation

Communication tool between multiple stakeholders

Communicates understanding of requirements

Often used as a contract

Baseline for change control

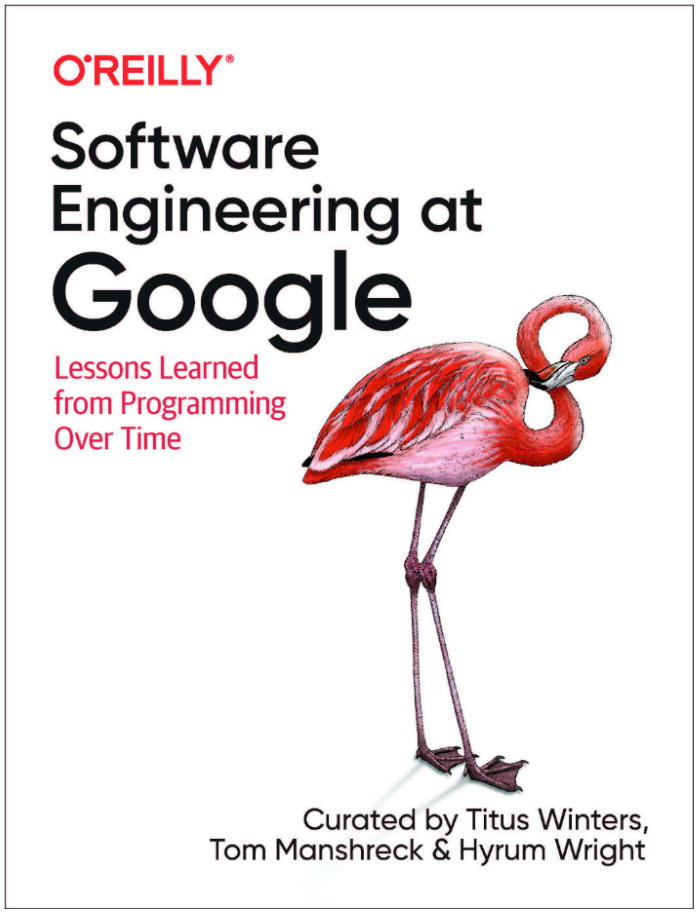
The RD: Communication tool between multiple stakeholders

Requirements documentation as a form of Knowledge Management

How developers see users	How users see developers
Users don't know what they want. Users can't articulate what they want. Users have too many needs that are politically motivated. Users want everything right now. Users can't prioritize needs. Users refuse to take responsibility for the system. Users are unable to provide a usable statement of needs. Users are not committed to system development projects. Users are unwilling to compromise. Users can't remain on schedule	Developers don't understand operational needs. Developers place too much emphasis on technicalities. Developers try to tell us how to do our jobs. Developers can't translate clearly stated needs into a successful system. Developers say no all the time. Developers are always over budget. Developers are always late. Developers ask users for time and effort, even to the detriment of the users' important primary duties. Developers set unrealistic standards

The RD: Communication tool between multiple stakeholders

Requirements documentation as a form of Knowledge Management



Read online

Purchase from O'Reilly

online version

Knowledge management within your Team

Table 3-1. Group interaction patterns

Recommended patterns (cooperative)	Antipatterns (adversarial)
Basic questions or mistakes are guided in the proper direction	Basic questions or mistakes are picked on, and the person asking the question is chastised
Explanations are given with the intent of helping the person asking the question learn	Explanations are given with the intent of showing off one's own knowledge
Responses are kind, patient, and helpful	Responses are condescending, snarky, and unconstructive
Interactions are shared discussions for finding solutions	Interactions are arguments with "winners" and "losers"

SRS contents

Functionality

Data Requirements

External interfaces (user, hardware, software, communication)

Performance

Design constraints

Other requirements

Functional requirements

inputs => outputs

user interface structure and behavior

data processing

error handling

Non-functional requirements

For example?

Non-functional requirements

physical environment

users and human factors

documentation

resources

quality assurance

security

system interfaces

Data Requirements

Data Requirements

- Logical Data Model
  - ER-Diagram
- Data Dictionary
  - Length, value, description, composition
- Reports
  - Example report
- Data Integrity, Retention, and Disposal

Quality characteristics of Requirements Documentation

- Unambiguous
- Complete
- Verifiable
- Consistent
- Modifiable
- Traceable
- Ranked for importance
- Correct

Quality characteristics of requirements specification

Unambiguous

Each statement can be read in exactly one way

Complete

Includes all the significant requirements, e.g. related to functionality, performance, design constraints, attributes or external interfaces

Verifiable

A process exists to test specification of each requirement

Quality characteristics of requirements specification

Consistent

Three types of conflict which can occur are:

different terms used for the same object: e.g. "a P45" and "a tax form" might be used to describe the same form.

characteristics of objects conflict: e.g. in one part of the requirements document, "a red light will indicate a fault", while in another part, "a blue light will indicate a fault".

logical or temporal faults: e.g. "A follows B" in one part, "A and B occur simultaneously" in another.

Modifiable

Easy to use organization, easy to change without difficulty

Quality characteristics of requirements specification

Traceable

The origin of each requirement is clear ("backward traceability")
Facilitates referencing to future documentation related to RS ("forward traceability") such as test plans, design specs

Correct

contains as much but no more than what is required (i.e. expresses only the real needs of stakeholders)

Ranked by importance

Contains priority information for each requirement