ASSIGNMENT 2

COMPARISON OF VARIOUS SOFTWARE REQUIREMENT SPECIFICATION FORMATS

By : Sakshi S Nayak

15IT235

INTRODUCTION

The requirements phase is the most critical phase of the software development life cycle (SDLC). Wrong or missing requirements lead to wrong or incomplete product, no matter how good the subsequent phases are. The quality of the requirements phase affects the overall quality of the subsequent phases and hence, the software product. Writing good software requirements specification (SRS) is an important determinant of software quality . The SRS document defines the capabilities of the provided software . Therefore, if an analyst or a developer does not share the same understanding about requirements, the outcome of the development process will not satisfy the customers' needs . The more progress in the software development life cycle, the more defects will emerge. Consequently, the earlier the detection of requirements defects, the much money and time of rework can be saved. To achieve this, it is important to study defects in the requirements phase and defect detection techniques, especially in the SRS document.

A standard SRS should not go beyond specifying requirements, in particular it should not include any design or implementation details, e.g. specific screen design, although the user interface section may include logical formats or layouts. A good SRS should have the characteristics of:

1)Correct
2)Unambiguous
3)Complete
4)Consistent
5)Ranked for importance
6)Verifiable
7)Modifiable
8)Traceable.

COMPARISION OF IEEE SRS FORMAT WITH OTHER SRS's

- User stories encourage the team to defer collecting details. An initial place-holding goal-level story can be written and then replaced with more detailed stories once it becomes important to have the details. This technique makes user stories perfect for time-constrained projects. A team can very quickly write a few dozen stories to give them an overall feel for the system. They can then plunge into the details on a few of the stories and can be more informative for creating the document than following IEEE SRS format which feels compelled to complete an IEEE 830–style software requirements specification.

- The IEEE SRS has only product requirements, not project requirements, i.e. it doesn't contain costs , schedules, development methods, acceptance procedures, etc.

- The business requirements specification is not considered in the IEEE SRS .In practice, the business requirement plays a important role while developing a product as business requirements has budget constrains which would give the investor a clear cut of how much would the project cost in the initial stages of developing a product.

- The IEEE SRS doesn't contain Document Approvals for the project i.e. It must contain a part where the client , investor, and the developers must be officially linked to the project which could be done by adding a new section called the Document Approvals where the main actors can sign .

- There are few subheadings which are repetitive and hence unnecessary. For Example- operating environment and hardware and software interfaces.

- IEEE SRS format has the revision table only at the starting of the SRS .Many companies have the table at the header of each page to keep in track of the version and the changes already made and which would allow us not to get confused while writing the SRS.

- IEEE SRS does not contain important topic like Architecture diagram, database design which is covered by IBM SRS.

- The IEEE SRS is quite lengthy as it contains many subsections under a section which are less significant or trivial compared to TCS and IBM SRS.

ADVANTAGES AND DISADVANTAGES OF VARIOUS SRS FORMATS

IEEE FORMAT

Advantages
1. Very detailed and concise
2. Useful for the developers to get a good idea of what the client wants to see in the final product
3. There is a provision to add requirements later by including them in the "To Be Discussed" section
4. Evolving documentation
5. Follows a standard document convention that has to be followed by all the maintainers

Disadvantages
1. It is hard for the client to specify some requirements(like communication interfaces) in such detail at the beginning phase
2. Many sections may be marked under "To Be Discussed" remain unfilled during the initial phases
3. Takes a considerable amount of time to prepare

4. SRS will undergo many revisions due to refinement of requirements
5. Less focus on UML/class diagrams, though they help the developer visualize the final product

BELITSOFT FORMAT

Advantages
1. Has extra details about memory constraints and special operations by the user
2. The “Apportioning” section is a helpful addition as it maps the requirements to the software that will carry out the given operations to satisy the requirement
3. “Site adaptation requirements” help in ensuring the end product will be compatible with more systems by specifying any data or initialization sequences that are specific to a given site
4. “Usability requirements” include measurable effectiveness, efficiency, and satisfaction criteria in specific contexts of use.
5. Provides the verification approaches and methods planned to qualify the software
6. Supporting info contains : Sample input/output formats, descriptions of cost analysis studies, or results of user surveys; A description of the problems to be solved by the software; Special packaging instructions for the code and the media to meet security, export, initial loading.

Disadvantages
1. Much more complicated and time consuming to create. No user documentation. 2. Harder for the client to specify such requirements detail at the beginning phase(more revisions)

ZILDOR FORMAT

Advantages
1. Less complicated than Format 2 to create
2. “Key Milestones” give the client a good idea of what to expect and when. This avoids unrealistic expectations
3. “Project Proposal” provides a complete summary of the given problem statement and what the end product will be capable of
4. System features are split into core and additional so that developers can assign priorites. 5. Key Resource requirements divide the huge problem into managable logical chunks and state the necessary required expertise, internal/external resources to satisfy the requirements and the associated constraints

Disadvantages
1. Many revisions can still arise.
2. Harder for the client to specify such requirements in such detail at the beginning phase

UNIVERSITY OF TEXAS FORMAT

Advantages
1. Simpler format than 1,2 and 3, easier to make

2. Client can specify functionality without going into the specific sub-requirements
3. Includes Legal information and standards

Disadvantages
1. No diagrams to aid developers think of the final product
2. More focus on non functional than functional requirements.
3. Support Information can be vague since there is no proper definition

IBM FORMAT

Advantages
1. Simplest format
2. Use-case/class/ER/Web Architecture diagrams aid the software developers
3. Includes Legal information and standards, not mentioned in others

Disadvantages
1. Maybe too simple
2. Supplementary requirements might mix up functional and non-functional requirements
3. No mention about user documentation