IT CAPSTONE PROJECT & Research 2

JOHN MICHAEL LEGASPI,DIT(c)

WORKSHOP

CONTENTS OF THE LESSONS

Software Requirement and Specification

Software Development

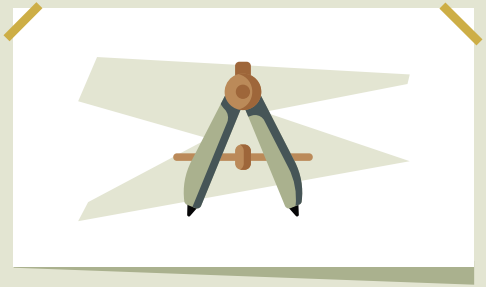
Software Requirements are a way to identify and clarify the ***why***, ***what*** and ***how*** of a business's application. When documented properly, software requirements form a roadmap that leads a development team to build the right product quickly and with minimal costly rework.

The actual types of software requirements and documents an IT organization produces for a given project depend on the audience and the maturity of the project. In fact, organizations often draft several requirements documents, each to suit the specific needs of ***business leaders, project managers and application developers.***

What is this topic about?

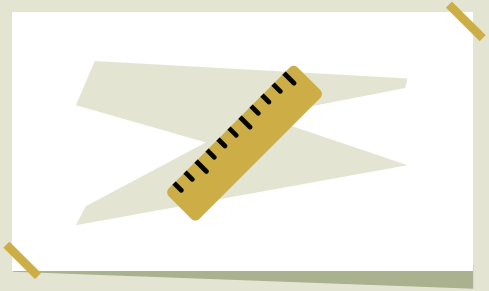
BUSINESS

Business needs drive many software projects. A business requirements document (BRD) outlines measurable project goals for the business, users and other stakeholders.



USER

User requirements reflect the specific needs or expectations of the software's customers.



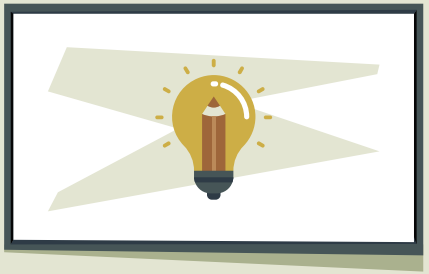
SOFTWARE

the team should devise a software requirements specification (SRS) that identifies the specific features, functions, nonfunctional requirements and requisite use cases for the software.

Software Products

Generic Products

- Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
- The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.



Customized Products

Software that is commissioned by a specific customer to meet their own needs.

- The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.

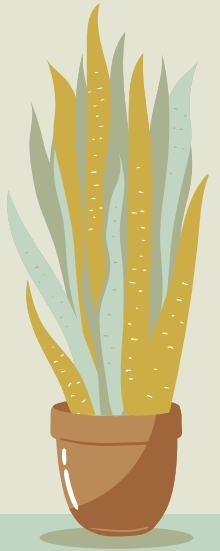
Sample Statement on Business Requirement

FORMAT

"The [project name] software will [meet a business goal] in order to [realize a business benefit]."

Example

"The **laser marking software** *will allow the manufacturing floor to* **mark text and images on stainless steel components using a suitable laser beam** *in order to* **save money in chemical etching and disposal costs."**



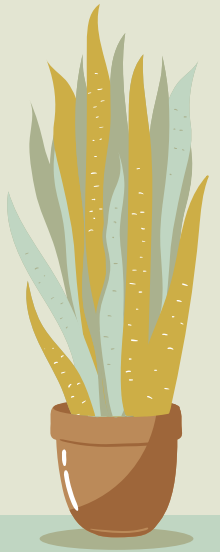
Sample Statement on User Requirements

Format

*"The **[user type]** shall **[interact with the software]** in order to **[meet a business goal or achieve a result]."***

Example

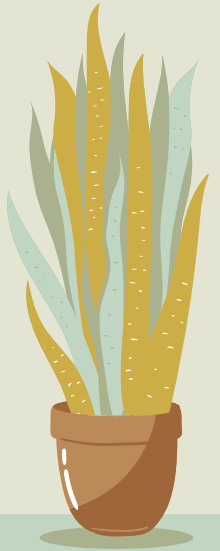
*"The **production floor manager** shall* **be able to upload new marking files as needed** *in order* **to maintain a current and complete library of laser marking images for production use**."*



Software Requirement

Software requirements typically break down into:

- ✓ **functional requirements**
- ✓ **nonfunctional requirements**
- ✓ **domain requirements**



Functional requirements.

Functional requirements are statements or goals used to define system behavior. Functional requirements define *what* a software system must do or not do. They are typically expressed as responses to inputs or conditions.

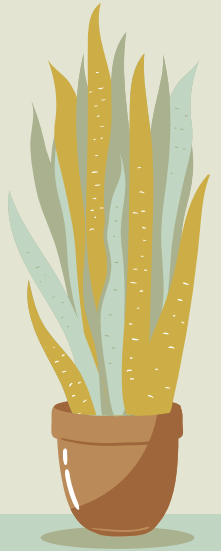
If an alarm is received from a sensor, the system will report the alarm and halt until the alarm is acknowledged and cleared.

Nonfunctional requirements (NFRs).

Nonfunctional requirements relate to software usability. Nonfunctional software requirements define *how* the system must operate or perform. A system can meet its functional requirements and fail to meet its nonfunctional requirements.

An example nonfunctional requirement related to performance and UX could state:

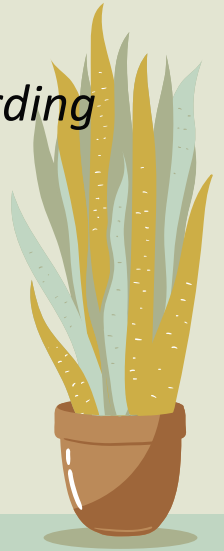
The pages of this web portal must load within 0.5 seconds



Domain requirements

Domain requirements are expectations related to a particular type of software, purpose or industry vertical. Domain requirements can be functional or nonfunctional.

The software must be developed in accordance with ISO 9126-1 regarding the basic safety and performance for medical electrical equipment.

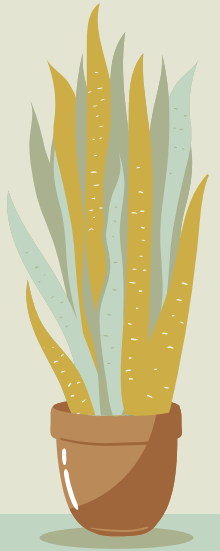


Software Requirement

Format : "The [feature or function] shall [do something based on user inputs and provide corresponding outputs].”

"The laser marking operation shall translate AutoCAD-type vector graphics files into laser on/off control signals as well as X and Y mirror control signals used to operate the laser system."

"The software provides a visual feedback to the operator, who shall track and display the current state of the marking cycle overlaid on a graphic product image displayed on a nearby monitor in real time."



Characteristics of good software requirements

Clear and understandable. Software requirements must provide the utmost clarity. Write requirements in plain language, free of domain-specific terms and jargon. Clear and concise statements make requirements documents easy to evaluate for subsequent characteristics.

Correct and complete. The document should accurately detail all requirements. If it's a BRD, the document should detail all business goals and benefits. If it's an SRS, it should describe all features and functionality expected from the system.

Characteristics of good software requirements

Consistent, not redundant. Software requirements documents are often long and divided into multiple parts -- each with its own specific requirements. Consistent requirements have no conflicts, such as differences in time, distance or terminology.

Unambiguous. No software requirement can leave room for interpretation. Even clear statements can still be subject to multiple interpretations, which leads to implementation oversights.

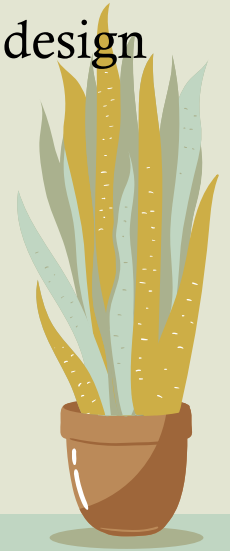
Characteristics of good software requirements

Design-agnostic. Software requirements documents should illustrate an end result. Like an architectural diagram, the different types of requirements together detail what the development team should build and why, but rarely explain how.

Measurable and testable. The goal of a requirements document is to provide a roadmap for implementation.

Characteristics of good software requirements

Traceable. It's hard to know when developers are actually done with a software project. Ideally, there will be a direct connection between requirements documents and finished code; a project manager should be able to follow the provenance of a project from a requirement to a design element to a code segment, and even to a test case or protocol.




05.
Presentation
Next session, each team will present proposed Software Requirements

QUESTIONS ?

UP NEXT .....

ISO 9126

Software Engineering Metrics

SQA

Software Quality Assurance

Software Requirements

Identify what specification of the Software

Program Code

Program Code Quality Management

THANK YOU