

Problem Requirement Rough Draft.

NB: Client email approval required. Remember, this is your agreement with the client as to what will get done!

A requirements document outlines what you are doing for your project and what your client will be getting once you are finished. By now, you should have met with your client and discussed what you will be creating for them. The requirements document is a contract of work, spelling out with great detail what you will be working on for the year.

The requirements document is the "what" document describing your project. The different requirements should be at the individual task level. You might create user stories (for something like a kanban board), or you might create descriptions of requirements at the functional level. Either way, make sure that you do these at a sufficient level of granularity such that you can mark the project tasks as complete as you go.

Please use the IEEE Std 830-1998 to format

Team Name

Gantt Chart

Lets do this.

Problem Requirements

-BE SPECIFIC

-BE PRECISE

-no loose language...

-GANTT CHART (see the sheets template)

-STRETCH GOALS

-not a technical description, somewhere in between the problem statement solution and the technical documentation.

Table of Contents

1. Introduction

The BoxSand project aims to develop an open source Learning Module System that will be used by students to learn course material and complete online homework. The project, titled BoxSand, provides Open Educational Resources with a primary goal to provide students and instructors with an all-in-one online learning environment. This includes links to videos, practice problems, open source textbooks, educational websites, simulations, and more. The main goal of BoxSand is to improve student performance by providing access to free and open source resources all in one location. It aims to engage students and provide them with feedback while using the student's site interaction and engagement to track success and improve content.

1.1 Purpose

Since this is the first large-scale overhaul of the BoxSand project, this development team and this cycle should mainly focus on developing long-term and overarching development goals and procedures for future project development. Additionally, this development team will also create an initial proof of concept for site functionality demonstration and future feature integration

1.2 Scope

This development team will produce the BoxSand OpenStax site. At its core BoxSand is driven by a need to create an all-in-one open source learning website that is free and available for use by both students and instructors. The long term goal is to have a modular website where instructors can create a course in any subject and contribute and edit content. The goal for this development team is to set up an environment where this is possible. The development team will design, document, and begin implementation of that framework. This includes the tracking of student interaction with the site while using the OpenStax suite of technologies and adding a homework system.

- a) The website that will be developed must provide access to the OpenStax Physics textbook within the site itself.

- b) Provide a homework system within the site that allows an instructor of a course to provide questions with answers, Assign a value to the question and, assign a group of questions or a single question as an assignment to a course.
- c) Instructor must also be able to assign reading homework from the textbook for students within a course
- d) An instructor must be able to generate a downloadable gradebook of student scores

1.3 Definitions, acronyms, and abbreviations

This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

Please see Appendix A for a glossary of terminology.

1.4 References

1.5 Overview

2. Overall description

2.1 Product perspective

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

3. Specific requirements

3.1 External Interfaces

- a) Name of item;
- b) Description of purpose;
- c) Source of input or destination of output;
- d) Valid range, accuracy, and/or tolerance;
- e) Units of measure;
- f) Timing;
- g) Relationships to other inputs/outputs;
- h) Screen formats/organization; i) Window formats/organization;
- j) Data formats;
- k) Command formats;
- l) End messages.

3.2 Functions

- a) Validity checks on inputs
- b) Exact sequence of operations
- c) Responses to abnormal situations, including

- i) Overflow
 - ii) Communication facilities
 - iii) error handling and Recovery.
- d) Effects of parameters.
- e) Relationship of outputs to inputs including
 - i) input / output sequences
 - ii) formulas for input to output conversion

3.3 Performance requirements

- a) Static numerical requirements
- b) The number of simultaneous users to support
- c) Amount and type of information to be handled.

3.4 Logical database requirements

- a) Types of information used by various functions.
- b) Frequency of use.
- c) Accessing capabilities.
- d) Data entities and their relationships.
- e) Integrity constraints.
- f) Data retention requirements.

4 Design Constraints

- a) This should specify design constraints that can be imposed by other standards
Hardware limitations Etc.

4.1 Standards compliance

- a) report format.
- b) Data naming.
- c) Accounting procedures.
- d) Audit tracing.

4.2 Software System attributes

1. There are a number of attributes of software that can serve as requirements. It is important that Required attributes of a specified so that their achievement can be objectively verified
 - a) Reliability.
 - b) Availability.
 - c) Security.
 - d) Maintainability.
 - e) Portability.

4.3 Organizing the specific requirements

- a) System mode.
- b) User class.
- c) Objects.

- d) Feature.
- e) Stimulus.
- f) Response.
- g) Functional hierarchy.

4.4 Additional comments

5 Supporting Information

5.1 Table of contents and index.

5.2 Appendixes

- a) Sample input/output formats, descriptions of cost analysis studies or result of user surveys.
- b) Supporting or background information that can help the readers of the SRS.
- c) A description of the problems to be solved by the software.
- d) Special packaging instructions for the code and the media to meet security export, initial loading, or other requirements

Appendixes (we need to do the right format listed in that doc.

<http://www.cse.msu.edu/~cse870/IEEEExplore-SRS-template.pdf>

Appendix A - Glossary of Terminology (alphabetize after def...)

Project BoxSand or BoxSand

OER

OpenStax

Course

Instructor

Student

Administrator

Teaching Assistant (TA)

Assignment

Learning Module

Daily Learning Guide

Gradebook

Reading

Homework

Quiz

Question

Page

User

Unregistered

Registered
Organization
Tutor
Tracking
Content
Resources
Media
AsyncSync
Adaptive Learning
Virtual Whiteboard
Chat client
Physics with Algebra
Physics with Calculus
eCampus
Flipped Classroom
Kaltura
YouTube
MasteringPhysics
PeerCeptiv
Facebook
Canvas
Oregon State University
OSU Physics Department
Drupal
Docker
VM
Wireframe
Vagrant
Ansible
Ruby
Rails
HTML
CSS
JS
Rbenv
Node (NodeJS)
Client-side (client)
Server-side (server)
Customer

Sketch
Sparfa
BigLearn
AWS
Linode
Dev
Production
Live
GitHub
GitLab
OSL
CMS
Bootstrap

Index