Software Project Documentation

No one likes creating documentation, but everyone enjoys the benefits that comes from having appropriate documentation. Software projects run smoother and future enhancements are easier when there is written documentation to go along with the code.

Certain types of documents, in one form or another, are found on the best run software projects. This pages provides links to templates, examples and checklists for the most common types of documents used on software projects.

Just because a particular document is mentioned here though, doesn't imply it is needed for all projects. Furthermore, just because a particular resource (template, example or checklist) includes certain content, doesn't mean that this content is needed for all projects all the time. The types of documents created and their content are dictated by the needs of the project.

For each type of document mentioned here, there is:

- A brief description describing the content and purpose of the document.
- A template that shows major section headings and a brief explanation of what is needed in each section.
- One or more examples that illustrate what an actual instance of the document might look like. Examples are good for clarification and inspiration, not the final word on what is needed or allowable. As you might expect, examples are narrowly focused. What is appropriate for one project might not be appropriate for another.
- A quality assurance (QA) checklist that lists criteria, usually in the form of brief questions, for assessing a document. A checklist can provide guidance during development or serve as a standard for evaluating completed work.

For convenience, the documents are organized into two groups:

1. Process Documents
2. Product Documents

Process Documents

Vision Statement

Template | Example | QA Checklist

The vision statement sets the direction for a project by specifying what will be accomplished. It defines the scope of a project. A good vision statement helps you make decisions regarding priorities and what to include and what to exclude.

Project Charter

Template | Example, Example | QA Checklist

The project charter defines the scope of the project and provides justification and rational for pursuing it. If the project is approved, much of the information contained in the project charter will be expanded upon and further refined in the project plan.

Software Requirements Specification (SRS)

Template (Template-Lite) | Example, Example1, Example2, Example3, System Shalls | QA Checklist

The software requirements specification lists the functional and non functional requirements along with any implementation constrains. The requirements document serves a diverse audience ranging from non-technical clients to programmers. To meet the needs of this diverse group, requirements are commonly expressed at progressive levels of detail. Most requirements documents will include a list of general product features as well as the detailed system behavior needed to deliver these features. Detailed system behavior is often expressed with use cases or usage scenarios.

Software Project Management Plan (SPMP)

Template (IEEE Template) | Scrum Planning Guide | Example, Example | QA Checklist

Project planning is the process of defining the expected outcomes of the project and devising a course of action for accomplishing them. The project plan documents the results of the planning process.

Expect the project plan to be updated and refined continuously throughout the project as conditions change and more is known about the project. Few projects start with all the information needed to plan the project in detail from the beginning. More often, the project plan starts with a high-level breakdown of known work, a course grain schedule and only ballpark estimates for cost and duration. Over time as more is understood about the project, these and other components of the project plan are progressively refined.

Release Plan

Template-Lite, Template for Tracking Estimates over time | Example | QA Checklist

The release plan is a high-level schedule that stretches for the duration of the project. There is one per project and it specifies the timing of iterations and a rough allocation of product features to iterations.

Iteration Plan

Template | Example, Example | QA Checklist

An iteration plan defines the activities that will be performed during an iteration. There is one per iteration and they specify the detailed tasks for an iteration, and in some cases an assignment of tasks to individuals

Memo of Understanding

A memo of understanding is a less formal way of documenting assumptions and intentions. Verbal agreements are often documented in a memo of understanding.

Project Success Criteria

Template | Example | QA Checklist

Project success criteria describes how the results of the project will be measured. If the project is scheduled to be complete on July 1, will it be considered a failure if it is finished on July 2nd? It will if the results of the project were intended for a trade show schedule for July 1st. In other cases, being a week late might be tolerable. The project success criteria defines in a measurable verifiable way what constitutes project success. tbd

Project Closure Report

Project Metrics

Change Request

Template | Example | QA Checklist

Change Control Log

Template | Example | QA Checklist

Issue Log

Template | Example | QA Checklist

Status Report

Template | Spreadsheet for tracking estimates and actuals | QA Checklist

Product Documents

Architecture and Design

Template | Example1, Example2 | QA Checklist

The purpose of the architecture/design document is to explain the organization of the code. A well written architecture document will reduce the amount of time it takes programmers new to a project to read and understand the code at the level needed to make modifications and enhancements.

The architecture/design document should identify major system components and describe their static attributes and dynamic patterns of interaction.

Software architecture and designs are typically expressed with a mix of UML models (class and sequence diagrams being the two most common) and prose. Dataflow diagrams are also helpful for understanding the interaction between components and overall flow of data through the system.

Architecture is high-level design so the principles found in both types of documents are the same but how they are expressed might vary depending on the level of the design being captured. For example, communication protocols between architecture components are relevant for an architecture document but probably not a design document were communication between components is usually accomplished via shared variables/memory or procedure calls.

Coding Standards

Template | Example | QA Checklist

Test Plan

Plans, Testing Doc | A1,A2; B1, B2, B3 | QA Checklist

Test Case Specification

Template | Example | QA Checklist

User Guide

Template | Example, Example | QA Checklist

The user guide explains how to use the software from the user's perspective. A well written user guide will welcome first-time users by providing basic information about how to get started quickly, but also include more in-depth information for power users wanting to understand how to use the more advanced features of the software.

System Documentation

Template | Example, Example | QA Checklist

The system documentation (aka installation guide, administrator's Manual, etc.) explains how to install and configure the software.

Other: activitylog.xls,