(last revision May 10th , 2017)

Name of Project

Author

Team

Issue Date of Baselined Document

Subsequent issue dates listed earliest to latest

INTRODUCTION (required)

This is a brief description of the project, at most 2 paragraphs. You should point to key documents, e.g., requirements documents, architecture review documents, state the problem you are trying to solve and state success criteria. Briefly state what is the problem you are trying to solve and perhaps a user story of how it will be used.

ROLES AND RESPONSIBILITIES (required)

These vary for each type of product and, for small projects, folks may serve multiple roles. This is a list of common roles we have used for software development:

Development Lead (only 1 name)
Buildmeister (only 1 name)
Architect (only 1 name)
Developers (multiple)
Test Lead (only 1 name)
Testers (multiple)
Documentation (multiple, usually ALL)
Documentation Editor (only 1 name)
Designer (only 1 name)
User advocate (only 1 name)
Risk Management (only 1 name)
System Administrator (only 1 name)
Modification Request Board (1 leader, multiple representatives)
Requirements Resource (usually 1 name)
Customer Representative (multiple)
Customer responsible for acceptance testing

METHOD (required)

These are unique to software development, although there may be some overlap.

- Software:
  - Language(s) with version number including the compiler if appropriate
  - Operating System(s) with release number
  - Software packages/libraries used with release/version number
  - Code conventions – this should preferably be a pointer to a document agreed to and followed by everyone
- Hardware:
  - Development Hardware
  - Test Hardware
  - Target/Deployment Hardware
- Back up plan (individual and project)
- Review Process:
  - Will you do architecture, usability, design, security, privacy or code reviews?
  - What approach will you use for the reviews (formal, informal, corporate standard)?
  - Who is responsible for the reviews and resolving any issues uncovered by the reviews?
  - Code readings?
- Build Plan:
  - Revision control system and repository used
  - Regularity of the builds – daily
  - Deadlines for the builds – deadline for source updates
  - Multiplicity of builds
  - Regression test process – see test plan
- Modification Request Process:
  - MR tool
  - Decision process (board – if more than paragraph should point to alternate description)
  - State whether there will be two process streams one during development and one after development

Virtual and Real Work Space

It is great to have a project room, it is also great to use a wiki or some document repository system so long as it is private. Google docs is not private but may be sufficient for a class or university project.

COMMUNICATION PLAN (required)

"Heartbeat" meetings

This section describes the operation of the "heartbeat" meetings, meetings that take the pulse of the project. Usually these meetings are weekly and I prefer to have them early in the day before folks get into their regular routine, but this is not necessary. The meeting should include only necessary individuals – no upper level management or lurkers. It should have a set agenda, with the last part of the meeting reviewing open issues and risks. It should be SHORT, thirty minutes or less is ideal. Notes should be provided after the meeting and issues should be tracked and reviewed each meeting, usually at the end.

Status meetings

Status meetings have management as their target and should be held less frequently than heartbeat meetings, preferably biweekly, monthly or quarterly depending on the duration of the project. It is solely to provide status for the project. If issues arise, they should be addressed at a separate meeting (see next item). These should be short. This section should describe the format and periodicity.

Issues meetings

If a problem does arise, never surprise your manager. Schedule a meeting at his or her earliest convenience. This section describes how alerts will arise and the governance of when to trigger an alert – usually after a discussion at the heartbeat meeting.

TIMELINE AND MILESTONES(required)

This section should be crisp containing 4-10 milestones for the duration of the project, each of which would trigger a re-issuing of this document to report on progress. Each milestone should define a 100% complete item, should list the critical participants and list begin time and end time. Each time you re-issue this document you should highlight changes with italics or bold – colors will not show up on a photocopy.

TESTING POLICY/PLAN (optional–software relevant)

This should probably point to a plan or the document would get unwieldy. At the very least it should describe when testing begins.

RISKS (required)

Ideally it occurs because of an established risk management program. If that does not exist, do the best you can to enumerate the risks and explain how they will be track and monitored.

ASSUMPTIONS(required)

It may be clear to the project insiders what assumptions are being made about staffing, hardware, vacations, rewards, … but make it clear to everyone else and to the other half of the project that cannot read your thoughts.

DISTRIBUTION LIST

Who receives this document?

MORE OPTIONAL SECTIONS:

These should be self-explanatory.

Worry beads

I add this section to describe the things as manager I am most worried about at the time of latest document issue. This section is useful because it helps you to focus on the parts most likely to fail. Sometimes, I segment the worries by time scale: day, week, month, quarter … lifetime.

Documentation Plan

Many years ago we had much too much documentation, now we have precious little – this has to change. Write documentation as if you'll need to personally support the project forever – you just might need to and you'll be glad you took the time to document the obvious, the not so obvious and the obscure. As an example, it's useful to document alternate architectures and designs you did not pursue along with the rationale. What were the "gotchas" you were trying to avoid?

Build Plan

When builds and testing become complex, this might be a separate section or point to a separate document.