Capstone Project Proposals

2020/01/31

Overview

A project proposal is the document that describes the what, why, how, and when of creating a product or service. It's really two different documents combined:

1. **Product Design Specification:** a description of what product or service you're going to build.
2. **Project Management Plan:** A description of how and when you're going to to build it.

Here's an example outline of project proposal. There are different formats, different labels, and different ideas of what's more important than others, but they all contain basically the same thing:

- **Overview / Executive Summary**
- **Product Design Specification**
    - Concept of operations / User stories
    - Stakeholders
    - Requirements
    - Specifications
    - Deliverables
    - Initial product designs
    - Verification plans
- **Project Management Plan**
    - Timeline, with milestones
    - Budget
    - Team and development process

Formatting

All the things you should already know to do for a formal document:

- Sane margins and a cohesive font choice..
- Title, author, date, version number.
- Number the pages. Best is a format like "i/j" pages so the reader knows where they are.
- I strongly suggest a table of contents (automatically generated, of course) if there are more than about ~4 pages, or there are a lot of sections.

Writing Style

Please don't use the passive voice. Your document should be written in a direct and *concise* fashion. Some of the best PDS are just a few pages long. There's no reason to sound "fancy" or "scientific", which are code words for convoluted, passive, hard to read sentences which people think makes you sound smart. They don't. People who write this way often actually don't know what they're trying to say, and are hiding it with poor writing. Use direct, short, active sentences. Use "we" and "our". Does that sentence sound weird? Take a step back, and say it more simply.

Overview / Executive Summary

This is "30 second elevator pitch". It's the 10,000 ft view of your project. It should answer the quick "who, what, why, and when." It should use as little technical jargon as possible, and clearly describe to a technically competent reader what you're doing. They should understand what you're going to do by the time they're done with the executive summary. It should be 1 to 3 (at most) paragraphs.

Product Design Specification

This is your "standard" PDS. The important thing to remember is that it's a contract between you and your industry sponsor (and faculty advisor) that codifies what you're going to design and build. After reading the PDS, a reader should be crystal clear on what your product/service is, What parts of it you're going to deliver, and how you're going to approach the problem.

Concept of operations / User stories

Don't ignore this often ignored section. The Concept of operations (often called "conops") and user stories describe to your reader in a concise format what you're being asked to create. This is section is *not* what you're going to do, it's a description of what the industry sponsor is asking for, who is going to use it, and how they're going to use it. You should be answering questions like:

- What does this thing/service do?
- Who uses it? Who are the stakeholders?
- Why would they want it? What does it do for them?
- How does the sponsor get it to the users?
- How do the users use it? What does the user need to use it?
- Can it be reused? What does it look like if the project is successful?

Stakeholders

This is a formal list of the stakeholders in the project. Often this is the industry sponsor, the users of the project, and feel free to list yourselves as the project team. Stakeholders may be more than you think; is the purchaser of this system the same as the users? I.e., who makes the decision to purchase the system? Who uses it? Who has to service it? Who determines when it's at the end of its useful life?

Requirements

These are the requirements of your product. Often called "User Requirements" or "Marketing requirements". Don't forget that requirements are abstract, verifiable, unambiguous and written as a concise statement, traced to a user need, and realistic. Although not required, "must", "should", and "may" are useful statements if this is a large project that won't all get done in the next 5 months.

REMEMBER: Unless the user is specifically requiring the "how", this is *not* the "how" section. The requirements are about what requirements your project must deliver on, in an abstract way. "Must last 3 hours when on battery power" is a good example requirement, "Uses 2.5 Ah LiPo battery" is *not*.

Specifications

Specifications and requirements are often mixed up. Specifications are the had numbers that you have to hit. Sometimes you won't need a specification section, the requirements are good enough. If you have a lot of hard requirements, however, it might be a good idea to have the requirements be your abstract, user-centered requirements list and then add a specification section that lists the hard specifications.

Deliverables

This critical section specs out what you're going to deliver to the industry sponsor by the end of your

capstone. This must include:

- Project proposal
- Weekly Progress Reports
- Final report
- ECE Capstone Poster Session poster
- Any other documentation or activities (presentations) your *faculty advisor* requires

And, *oh*, *right*, you should probably also deliver the project you're building for the industry sponsor. This is very different for different capstones, so here's an example listing of some of the things you might be expected to deliver:

- Detailed design documentation, explaining how your what your design does, how your design works, and why you made the decisions you did
- Simulations, including both the simulation inputs and outputs
- Electrical CAD: Schematics and board layouts, including output files (Gerbers, PDFs, etc)
- Mechanical CAD: enclosures, mechanisms, including output files (STLs, PDFs, etc)
- Bill of materials and pricing
- Discussion on manufacturing, servicing, updates, upgrades, etc.
- Short user manuals documenting how to use your creation
- Version control, including checked in previous revisions of your design (e.g., a Git repo).
- Intellectual Property information

Initial product design

OK, NOW, FINALLY, describe what you think you're going to make. Remember, this is a *proposal*. You literally don't know all the answers, so make sure your product design section is chock full of questions and discussions. Remember, *you don't have to know all of the answers!* This is a learning process. Even in industry, there should be questions and discussions in the design section.

- What are you proposing to make?
- How are you going to make it?
- What are the big risks? How are you going to answer them quickly?
- What are the questions you have still need to be answered?
- How much of this do you think you can get done in 5 months?

Things to include here:

- Hardware architecture
    - Block diagrams (L0, L1, maybe even L2)
- Software architecture
    - Languages and development environments
    - Data flow diagrams
- User interface / experience
    - User stories of how the end-user actually uses your product
- Other considerations
    - Security (physical, electronic, and software)
    - Regulatory compliance
- Back up plans
    - What if things go wrong? What can you fall back on?

While we encourage you to use MS Visio or Inkscape or LucidCharts to draw diagrams, do *not* be afraid to include some hand sketches here! Photos of whiteboard drawings, for example, are worth a thousand words and are much, much faster than, say, a full SolidWorks drawing. Crop them down and try to make them as presentable as possible.

Verification plans

Once you've built your Thing™, how do you know that it works, or it fulfills the requirements? It might be as easy as filling out a bullet point of features, but usually this involves a test where you formally record the results. This way you have proof that the system met its "must" requirements (or not) before hand off to the industry sponsor. Write simple test plans, and have pass/fail criteria. Including this in your PDS will help the sponsor understand what you're going to deliver in more detail. Also, it might not be possible to create a *detailed* test plan right now, because you might not yet know how you're really going to solve this. But a short section on what and how to test your final product should be possible.

Project Management Plan

Now describe, in detail, how you're going to actually do this.

Timeline, with milestones

Use a Gantt chart. It can be simple (Word table or Excel spreadsheet) or it can be fancy (MS Project, ProjectLibre, or some online Gannt program like Asanna and InstaGantt). However you do it, you'll need carefully think about how much time things are going to take, multiply it by a factor of 3 for the fact it's *way* more complicated and harder than you think, and then by a factor of 4 because you theoretically only have 10 hours a week to work on Capstone so you're 0.25 FTE (full time equivalent), and *now* you should be panicking about getting your project done and handed off before the end of May. Your timeline should be realistic: it's going to take time to get PCBs, components, and development boards. Things are going to go wrong, and you've underestimated how much time some parts of your capstone will take. Try to factor that in by providing wiggle room in your schedule for unexpected delays.

Milestones to include will depend on your capstone, but might include: research phase done, initial design done, first schematic, development tools up and running, board layout, first working prototype code, first integrated prototype, etc.

DO NOT FORGET TO INCLUDE TIME FOR:

- Writing the project proposal
- Writing the final report
- Writing / presenting other documents
- Creating the final poster

Budget and Resources

What's your budget? What do you need to buy? And when? And what other resources do you need that your industry sponsor has told you they would provide? List it here.

This section should also include resources *you* are bringing to the table. For example, are you using the EPL? List that as a resource. Where are you going to set up and build your project? Do you need a locker to store your capstone in? A work area at PSU?

Team and development process

Finally, a short section on your team and what development process you're going to use. Try to include:

- Who are the team members and what are their skills? Just some bullet points are fine, no need to include full resumes..
- What are each of the team members going to do? You might not know yet, of course.
- Who is going to be the point person to be communicating with your industry sponsor and faculty advisor?
- Is there a team leader for your group? Who is that?
- What collaboration tools are you going to use?
- Are you going to follow a method, like Agile, or spiral, or ?