

Computer Science Capstone Design



Assignment: Technical Demos

(100 pts)

Overview

One popular definition of “good business” centers around the concept of maximizing productivity while minimizing risk; this has led to entire industries and areas of research that focus specifically on risk management. There are areas in any project that are “technically risky”, which is essentially a fancy way of saying “not sure it will actually work” or – more likely – “not sure how much effort/time it will take”. Obviously, the less uncertainty you have on these fronts, the less risk there will be in your project.

Managing Risk with Technical Demos.

During your feasibility analysis, you (hopefully) came up with a set of alternative solutions to a challenge, did some research and maybe checked out some sample code, and then made a decision of what alternative you were going to go with. Even if you did a great job, however, that was just sort of “armchair engineering: You googled, you read, you educated yourself and analyzed the options to make the best *apparent* choice. But was it the right choice? Maybe there were flaws or missing details in your analysis that --- when you get down to implementing the system --- would suddenly reveal themselves as show-stoppers, a true nightmare that could sink or gravely delay your project. This is where technical demos come in: they allow you to reduce this substantial technical risk by developing small “proof-of-concept” demos of key technical elements. There are three specific goals:

- The demo proves that the functionality in question can even be implemented (in the give language/framework). Your feasibility analysis said “probably doable”; here you prove it.
- Even if it is obviously technically feasible, the demo will give you a real basis for estimating how hard implementation will be...and how long it will take. This is key to developing an accurate project schedule and planning project staffing.
- Finally, the demo will help one or more of the team members to develop expertise in the specific language/framework/technologies, which will jump-start actual implementation.

What you will do in this closing assignment is to work with your mentor develop a “Demonstration Plan”, which is simply a list of key skills/functionalities that you have explored and wish to present as proof-of-concept for key aspects of your project. You will then present these to your mentor for evaluation of your effort.

In terms of Requirements Engineering overall, the development of demos like this is an important phase, and often helps to drive out a last burst of hidden functional and performance requirements.

Assignment specifics: What to do

Obviously, the technical challenges for every project will be different but, in general, the technical challenges you identify will be driven by the work you did in your Feasibility Study: In that study, you investigated possible technical solutions to tackling various functionalities that you foresaw needing to provide in your project. You might have even done some simple “hello world” testing out of those functionalities. But will they actually work for the job that you are selecting them for on this project? At the end of every section in the Tech Feasibility document, you should have outlined how you plan to partially implement and test drive the key technologies that you’ve selected for your project. This becomes the default base content for your technical demo. Of course, there may have been other technically challenging issues that emerged since then, so feel free to modify. The point is that you should be demonstrating that you have every technical worry in the project already solved, so that you’re ready to put it all together into an integrated product and first prototype very quickly in Spring.

Every team will be asked to work up an agreed-upon “demonstration plan” with their CS faculty mentor, that essentially just lists out what proof-of-concept demos you will show during your meeting. There is a basic template (MSword) for developing a “Demonstration Plan” linked from the course website. You will download this template, then flesh it out with the technologies/capabilities that (a) are critical for your project and (b) that you are going to prove you have handled by showing a little demo you’ve developed. Hopefully these two lists overlap quite a bit...but be honest, if there are critical functionalities that you have *not* demo’d and that therefore remain a risk, then it’s better to be up front about it.

Specifically, here’s what you need to do:

1. Download the template and flesh out the missing parts. In the first part you name (C1, C2, etc.) the technical challenges that you’ve identified and briefly state what you feel would be convincing to demo its mastery. Then, you get to the lower meat: give a step by step outline for one or more “demo sequences” that collectively cover/demo all of the C1,C2...Cn challenges you listed above. It’s like a “flight plan”...it outlines exactly what you’ll be showing off and how. Any reader should be able to clearly visualize what you have in mind.
2. Send the modified plan to your mentor **at least one week in advance** of your planned demo date. Your mentor may ask you to add/upgrade certain missing/weak demos, so be sure to start on this right away to allow time for upgrades. You must receive an explicit “OK, I approve this demo plan” from your mentor **before you schedule a demo date!**
3. Make arrangements to meet with your mentor, in order to present your demos live and in-person. **Bring a hardcopy of your agreed-upon Demonstration Plan with you** for your mentor; it is meant to be used by your mentor for his/her notes, and ultimately as a sort of grading sheet.

Deliverables

- A final PDF version of your Demonstration Plan, **approved by your mentor**, by the end of finals week. Bring hardcopy to demo session planned with team mentor.
- A live in-person demonstration of your software demos for your CS Faculty mentor, following the established Demonstration Plan. Don't forget to invite your client, if there is any chance at all he/she could attend.