Computer Science Capstone Design

Assignment: Design Review 2

Presentation: 100pts

Overview

In real CS corporate practice, Design Reviews are a common way to update everyone in the group/division/company apprised of progress on your project. Specifically, there are three main goals that you are trying to achieve in a Design Review:

Efficiently remind people of what your project is about and communicate current work status to others within the organization.

Gather input and critical feedback from these groups that could help improve the design or project outcome.

And (always always) to *sell your project*; to remind the audience how vital/cool/necessary your project is, what a competent team you are, and how very much you deserve continued support. In real life, this last "political" point is at least as important as the other two put together!

Because of this somewhat "internal" focus of mid-project design reviews, the end-user client may or may not typically be invited to the Design Review in a real corporate setting; the decision on whether to do this depends somewhat on the nature of the project and on the involvement, technical proficiency, and desires of the client. **For our Capstone course, however, *you should always invite your client to attend all design reviews*.** Let them know at least a week in advance, so they can block out the time!

The Assignment

In this assignment, you will prepare and present the next formal Design Review for your project. The overall content focus for this and all other Design Reviews is the same:

- Intro: Intro the overall project area. Sell it as a vital/valuable market. Intro client's business and his/her needs.
- Problem and Solution Statement: Remind us of what's broken/inefficient, and what your vision for a solution is.
- The Meat. What you focus on here depends on where the DR falls in the project process. An early review might focus on user studies and envisioned product, later reviews might focus on requirements, software architecture, or particular stages of implementation. More details on this in the outline below.
- Update on Risks/Challenges/Resolutions. Again, what exactly happens here depends where you are on the project. Basically gives and update on status of project risks, plus any particular obstacles you've encountered/solved since the last review.
- Update on the project plan and schedule: Update us on how it's going and what's left to do.
- Conclusion: summarize and wrap it all up nicely.

Again, this is the basic outline for *any* Design Review; every single review is going to have the Intro and Problem/Solution pieces, and the schedule and conclusion pieces at the end. What varies between reviews as the project goes along is where you place the focus of discussion in the central "payload" part. Obviously, this should be on things you have been working on since the last review! So for this second review, the focus will be on what you've been doing so far this semester: Designing the software architecture and other implementation level decisions, combined with substantial implementation to arrive at an alpha prototype.

Design Review 2: Content Outline

As just discussed, every Design Review has the same basic outline, aimed at reminding the audience of the project, then updating them on recent progress. For Design Review 2, this means you'll focus on your software design and implementation efforts. The basic content and flow of info is about the same as within the Software Design document; **consult the guidelines posted for that document for more detailed discussion of what the following sections contain!** I'm just going to outline it here. Obviously, you have much less time for detail than you do in the written document. Your goal is to summarize your work effectively and compactly in oral presentation form, which is a vital skill for success in professional practice.

Introduction (< 1 minute)

The usual. Begin by introducing yourselves briefly: Go through each team member's name and role(s) on the project, as well as your team name, client, faculty mentor.

Problem Statement (about 2-5 minutes, depending on domain complexity)

As we've said from the start, this is an absolutely key section.   If you don't explain and motivate your project *very very clearly* here, you'll be in grave danger of having lost your audience. Lacking a clear idea of what you're doing, they literally won't be able to grasp the rest of your talk. And lacking a strong motivation of why this project matters, what its impact for the client and/or society might be, they won't care enough to listen. *This is a part of your oral presentations that you should further perfect with each Design Review* so that it's really strong by the time the Capstone Conference rolls around.

As discussed in the spec for the written document, begin by talking about the overall business area that your client is in: introduce the area, explain briefly how it works, and try to give some motivating info on how big/active/important that sector is.   How many people are involved, how much data is transacted, etc. If it's a complex area, graphics that help you explain processes, entities or data flow can help support you. THEN introduce your introduce your sponsor and the organization they're attached to, and say how they and their organization contribute within the larger picture of the sector you intro'd. What do they produce, how does it fit into that larger sector, and what is the volume/importance/user base of their part? What is the process by which your client produces whatever data/product that they are producing?

Ok, now the audience hopefully fully understands what your client does and how it matters. Now you can go on to describe what's broken, why you were hired. This should be easy if you've already described the workflow/dynamics of how your client's production/business process works: you then just have to explain what's bad/inefficient about it. Describe the problem in overall terms briefly, then get down to bulleting out a few specific things that are not satisfactory. By the end of this, your audience should be really clear on what needs fixing.

Solution Overview (about 2 minutes, depending on solution complexity)

Now you need to outline your plan for fixing the problems you just outlined. Again, this is easy if you've done a good job of describing the business workflow, and then pointing out specific problems with it: you just connect your solution right into this discussion. Again, begin with a broad statement of your overall solution, e.g., "The solution that we envision to address the client needs just outlined is to transition the entire workflow to a secure, highly reactive web application that …".   This is a great place to have process- or data-flow graphics you used to describe the clients business re-appear, with the elements you are fixing/adding clearly highlighted.   Then present a list of bulleted specific features of your solution; choose them so that it's pretty clear to listeners that your solution features will certainly address the problem issues bulleted out earlier. Explain as much as needed (e.g. walk us through figures, whatever) to make it clear.

DO NOT succumb to the temptation to get too detailed or technical here! You want to present the vision in a way that any reasonably tech-competent listener could relate to. More detail will come in future sections.

If you've done your work in the Problem Statement and Solution, your audience will:

a) understand the problem domain and what problems your client has

b) will have a solid overall idea of what you have in mind to fix it,

c) and will (!!) be strongly convinced that your solution vision will actually fix the clients problems completely and elegantly.

You might close with a statement previewing the intended effect for your client, e.g., "In sum, implementing and deploying our proposed solution should < save the client xx dollars, reduce time of processing by Y, significantly streamline order processing, whatever>".

OK, this is the end of the critical info. You're about 3-6 minutes into the talk and everyone is hopefully absolutely clear on what you're up to, and is hooked on your project. Time to get to detail of this particular Design Review! As always, smoothly lead in with a nice segue: "Now that we've established what we're doing, let's look in more detail at some project details and status".

Implementation Overview (about 1 minutes)

It's a good idea to begin the discussion of the implementation overview with a brief review of your requirements. Look at the requirements acquisition discussion you presented in the last DR, and condense that down to a slide: just briefly summarize your requirements acquisition process and the key requirements (maybe 4-6) that came out. This just helps remind the audience what to look for in your implementation.

Now go on to present an overview of the architectural structure of your solution. A graphic is highly useful here: you just sketch out the main "high level functional elements/modules" and how they interact. So for a web app might have a front end (using some frameworks), a back end (using some langs/frameworks/DB), and a hosting solution. Walk through each of these pieces and talk about them: what alternatives existed and/or why you made the high level design decisions (e.g. regarding frameworks, tools, hosting) that you did.

At the end of this discussion, your audience should very clearly understand the high level implementation approach you chose, and have the feeling that you did your research and have good reasons for all of those high-level design decisions.

Implementation Details: a quick tour of the guts (about 2-3 minutes)

Now that everyone understands what the overall architectural shape of your implementation is, it's time to give a little more implementation detail. This is the part where you can get detailed, technical and geeky. After a nice transition, dive us down for a look at the more detailed architecture within each of the main functional elements that you introduced in the last section. For instance, what are the main modules that make up the front end (e.g. User Profiles, Authentication, etc.)…and how do they interact to produce the behavior of the front end. Then "zoom in" to each of those higher level modules to show some more of what its design looks like in more detail. You don't have time or need to get into really low level details (no code!), but you want to at least give an idea of how you tackled design/implementation of that element. For instance, for a DB element, you could show the ER diagram for the DB schema, verbally discuss the design thinking or process that shaped development of the schema, and then highlighting a few keys tables or other important features of the schema to "sample" some detail. Or for an "video conversion module", you'd maybe review the key functions it has to provide, then show the public functions (API) it exposes to the rest of your system that allow it to satisfy those needs, and then maybe add in a few highlights of how it works internally. In short, what you are aiming to do is to communicate *the nature/thoroughness/detail of your design process,* rather than somehow showing every low level detail of your design.

By the end of this part, the technically savvy in the audience should have a clear idea of key implementation pieces, how implementation of each piece was approached and some sampling of what its final internals look like. This will allow them to ask specific questions and give specific technical input on your project.

Again, what you will NOT do here is to go into incredible code-level detail --- excruciating walk-though of endless UML, internal code structure, etc. It would just break your momentum, put your over time limit, and be boring to all. This is more about *elucidating and defending your design process* than describing your design details. Your Design Document should have all of this detail; a nice touch is to just invite the audience to have a look at your Software Design document for more detailed info on code structure.

Challenges and Resolutions (about 2 minutes)

In the first DR, this section focused on risks and feasibility issues. Now you are in the thick of implementation, so you should focus on coding challenges here. Briefly introduce a few of the challenging questions or problems that have come up in your implementation (e.g. "Computing spatial layout of nodes in our graphical state view"), briefly describe the problem, then present your solution. In any case, your goal here is to show that you have recognized some key implementation challenges and are well into solving them. If there are remaining challenges that you *don't* yet have solutions for, mention them here too…maybe someone in the audience will have an idea to help you out!

Schedule (about 1 minute)

As usual, offer a short discussion of your project plan, as it stands right now. A Gantt chart is highly recommended, with a "now" line running through it. Go over your main functional milestones…which ones you're through and what's coming. Close with some summary statement of where you are "going well", "somewhat behind, but we think we can catch up", whatever.

Conclusion (about 1 minute)

Finish your talk by providing a solid summary of your presentation: This is where you wrap it all up nicely and bring it all together. Start by briefly restating the importance of the domain, your client's business and processes and what was inefficient about them. Then review your solution vision, and go on to review what key topics you've discussed in this Design Review; do NOT review the details of those topics (you did that already in the middle part), just review what you talked about and the overall outcomes: Requirements acquisition, development of detailed Functional, Performance, Environmental requirements, and risks/feasibility.

End your conclusion with a sentence or two about what's coming up for you, what you'll be focusing on in the next development phase. End the presentation with an energetic note that conveys professionalism and confidence that you'll be able to solve the problems and satisfy your client.

Closing comments and Logistics

The times given for each of the topical areas to discuss are, of course, only nominal. It will be up to each team to tailor the amount of discussion in each area to the project…and, of course, to the overall time you have for the presentation. The point is that, if you go overtime, you'll bore your audience and drown them with detail (not to mention losing points for overtime); and if you use less than your allotted time, you've left some valuable time on the table that you could have used to explain some key points in your project more clearly. So you'll want to map out your time, try it, and adjust until you get it right.

You will be delivering this Design Review in the same overall format as the first DR: we split up into separate presentation rooms, with about 4-5 teams per room.

Some more details:

- The detailed room assignments for teams and faculty mentors will be announced shortly before the Design Review. Again, there will be three rooms, with reviews running in parallel.
- **All members** of your team must participate in the presentation.
- Consider asking your mentor to walk through a finished draft of your presentation with you, to comment on your slides, topics, granularity and other presentation details. Nobody is so good at presentations that we can't

use some feedback! Your mentors will literally expect finished presentations to review and give feedback on. Have all the figures in there, have the slides all organized, and know what you plan to talk about for each slide.

- Each team will have a **total of 15 minutes** for their presentation, including some time for questions. **Plan to talk for about 12 minutes**, to leave a few minutes for questions and critique.
- Dress up appropriately, meaning looking professional and credible.

Deliverables

DR 2 Presentation: Refine and practice your talk to prepare for the delivery of the formal Design Review in front of classmates, clients, and CS mentors.

There are two deliverables for this assignment:

- Your formal Design Review presentation, given at the designated time.
- Your "Team Facesheet", in hardcopy again, presented to mentor at DR, so that your mentor can comment on each of your individual performances during the talk.