Computer Science Capstone Design

Assignment: Design Review 3

DESIGN
REVIEW

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. Just exactly what you are wanting feedback on (and thus focusing on in your talk) depends on that project stage you are at in your development.
- 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*.** It certainly is not mandatory for them to come (as opposed to Capstone, which is coming up), but it's polite to invite them. 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 a formal Design Review for your project. The overall content focus for this and all upcoming 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 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 third review, the focus will be on the implementation work that you've done since the last Design Review, namely implementing your product into (hopefully) a fully functional beta prototype. Accordingly, the focus will be on (a) Decisions made/changed/updated in the implementation stage, (b) issues you ran into during implementation and how they were resolved, and (c) showing off the current final product.

Design Review 3: 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 3, this means you'll focus on your implementation and the beta prototype that has resulted, as well as your plan (i.e. testing and refinement) leading into the last part of the term. Of course, you'll want to wrap all of this into a "good story": as usual, you'll start with your motivation/introduction of the problem, require the requirements, outline the design/architecture, and then tell the story of what happened in implementation and how that turned out.

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 2-3 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.

You should have this fairly well nailed down by now: 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, e.g., how many people work in the area, how many people does it affect, how many use products like this, how much money is in it…anything of that sort that makes it clear that it's a vital problem. If it's a complex area, graphics that help you explain processes, entities or process flows relevant to understanding how business in this sector works can help support you. THEN introduce your introducing 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 it 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 (and only now!) are you ready to go on to describe the problem: 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 1-2 minutes, depending on how you tackle this)

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. 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 earlier to describe the client's business processes 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.

Since this is DR3 and you have a fairly complete beta prototype already out there deployed for testing, the "solution vision" part of the talk is a fine place to slip in some screen shots of your application here. If you do this, make sure that your discussion of those screen shots focuses on those features of your application that directly address the "what's broken" bullets you had in the problem statement! The point of your solution statement is to introduce and convince how your solution vision (potentially embodied in real software now) solves the client's specific problems. If you choose this route, you *could* actually decide to go on and show a fairly complete demo here, rather than later on in the talk. It's your call how you want to approach this…

In sum, if you've done your work in the Problem Statement and Solution Vision, 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 5 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".

Requirements/Specs review (about 1 minute)

To start a detailed walkthrough of how you built a house, it's nice to begin with a review of what was planned originally, i.e. the blueprints. Start 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 of condensed key requirements, and walk through them to remind the audience of specific capabilities that you'd required for the product. Just helps remind the audience what to look for in your implementation.

Architecture and Implementation Review (about 1-2 minutes)

Now go on to briefly review the logical architecture of your problem. Frameworks and tools used, overall models or design patterns expressed, and major functional elements in your implemented system. Graphics are highly useful here, as you just sketch out the main "high level functional elements" and walk through 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 briefly.

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.

Finally, follow the architectural overview up with another slide on "Implementation Overview". This is just an update on what you presented in introducing your software design plan in DR2. Give us a nice graphic showing what modules/classes/libraries that you wrote, briefly introducing what kinds of tasks each of them is responsible for. A nice way to bring this together is to follow up with a walk-through of a common use case, tracing through how various steps/actions the user would be taking are passed between and handled by the various modules that you just introduced. Your goal is to give listeners a clear idea of how all of these individual modules that you've described work together to produce the overall external behavior required for the system in a clean and obvious way.

What you will **NOT** do here is to go into incredible code-level detail --- UML, internal class structure, method headers. You just don't have the time and it would only break your momentum…not to mention being boring. As you finish overviewing the major pieces and how they work, you can let the audience know that your final as-build report will have all of the low level detail, and invite them to check that out after you finish up your project if they want to know more.

Prototype Review (about 4 minutes)

THIS IS THE MEAT! You've worked hard all semester to get a nice beta prototype up…the audience is surely dying to see what you've done. If you haven't chosen to insert a walk-through of your prototype earlier on (e.g., as part of an extended solution vision) then this is the time to show it off. **Warning: Live demos are dangerous!** If something breaks or the network goes down, you're sunk. On the other hand, for some products, it's the cool dynamic aspects that are the most impressive. Ask yourself whether the live visual flow is critically important/impressive for your product. If not, a series of well-chosen screenshots to present and discuss are probably a better bet.

In any case, your goal of the demo is to convince the audience that you have elegantly satisfied the requirements with a solution that will address the client's problem. This means: focus your screenshot and discussion on those aspects of the product that clearly address the problems the client needs solved. Again, a great way to organize such a demo walk-through is by coming back to the usage cases that you (hopefully) introduced earlier: set up a scenario of a user wanting to do whatever it is they need to do, then walk through them using your application to do so. This is *way* better than just giving a random walk through various features of your application! It clearly indicates how your application nicely solves the user problems you outlined in your intro.

Challenges and Resolutions (about 1 minute)

This section is about the same as the second DR: review coding quandaries or challenges you had, and talk about how they were ultimately resolved. In DR2, there were often problems that you were still trying to figure out how to solve; remind us of them and how that worked out now. Or maybe some new problems came up as you completed implementation; this is the time to tell us about those and how they were resolved. One would hope that you have no more "open" (unsolved) problems at this late day…but if you do, it's best to be honest and give an update and plans to solve. In any case, your goal here is to review the tricky challenges you faced and give some defense of your choice of solution. If you do this well, the audience will feel that you have conscientiously identified and tackled some hard issues in a reasonable way.

Testing Plan (1-2 minutes)

At this DR, your team should have (a) deployed your beta prototype for testing, (b) have created a testing plan (earlier assignment), and (c) be engaged in executing that testing plan. Hopefully this is exactly what the schedule you just presented showed…which gives you the perfect segue to talk about your testing plan a bit. Introduce your overall testing plan (a graphic is useful here), and walk us through the key elements briefly: what the element is, what sorts of testing it involves, and what you expect to get out of it. End with a summary of how you'll respond to test outcomes: fix problems as they come in, wait and fix them in phases during testing, whatever. You aim is to convince the audience that you have a solid testing plan, and are on the way to producing a refined, well-tested product.

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, what you're working on now, and what's left before your product delivery date. Close with some summary statement of where you are "going well", "somewhat behind, but we think we can catch up", whatever.

Conclusion (< 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 skim over the key highlights you presented in this DR; 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.

End your conclusion with a sentence or two about what you'll be focusing on in the last few weeks of the project, and say something about how your client is reacting to the prototype that you've delivered for testing and refinement. End the presentation with an energetic note that (hopefully) can now speculate confidentally on the fantastic and revolutionary impacts your software product is going to have on how the client does business: time saved, dollars saved, new capabilities enabled, etc.

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 other DRs: 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 3 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.