# Design Reviews[1]

## What is a design review?

A design review is a technical meeting that is held at an early stage in the software development process. A design review generally involves a presentation to an audience of stakeholders of a planned software design at a time early enough in the process to allow for changes.

Design reviews can fill a number of purposes, such as:

- Ensuring that the proposed design will meet the requirements.
- Ensuring that the design can be implemented in the required time.
- Ensuring that the design will have the needed interfaces with other systems.
- Ensuring the design will be consistent with technical goals of the organization.
- Ensuring the design will follow established software design principles.

Design reviews can be at various levels, corresponding to the level of detail of the design being evaluated. At the highest level, the overall architecture is reviewed; lower level design reviews focus on specific features or specific technical aspects of a system design. Design reviews can also differ with respect to the audience and timing. Very early in the design process it is often useful to present a design to intended users or customers to determine if the requirements will be met. Later in the design process the audience is usually limited to the team that will be doing the implementation.

## Why use design reviews?

Design reviews are useful in making sure that all participants in the software development process are "on the same page", that they understand how the software will be structured and how the various components will interact. Design reviews are useful in identifying design deficiencies and improvements or alternative approaches to the design.

## Who should attend a design review?

The attendance at a design review depends on the level of the review. A high-level design review can include developers, the technical management of the project, users or customers, and participants from related development teams that will make use of the final product. Lower level design reviews should include the team that will be implementing the design plus others affected by aspects of the design, such as users of a networking interface or a database design.

## When should design reviews be held?

Generally it is important to hold design reviews in the early stages of new software, laying out the framework in which subsequent software development will proceed. Design reviews should occur after the requirements are (fairly well) understood but before the code is written. Any project with more than one developer should hold design

[1] http://swiki.ucar.edu/sea-best-practices/16

reviews in the early development stages of the software. If changes are being made to an existing software base, any substantial changes to the design should be reviewed by the relevant stakeholders. Even on established software projects, if several members of the development team are new to the project, a design review should be held to ensure that the team is in agreement as to the technical direction of the project.

## How to conduct a design review

**Agenda:** A design review should be tightly structured to ensure that it results in useful feedback. An agenda should be distributed prior to the meeting. This agenda should identify the topics that will be presented and the specific issues that require discussion and feedback.

**Presenter:** Ordinarily the design is presented by the author of the design or the person most technically aware of all aspects of the design. More than one person may present a shared design.

**Presentation:** The objective of the presentation is to discuss the design in sufficient detail to enable the audience to understand how the design is intended to work, at a much higher level than a code review. Usually UML diagrams are provided and discussed, with sufficient detail to show the critical components of the design. However, these diagrams should omit extensive details that could distract the audience. Highly detailed UML diagrams can be kept on hand in case they are needed to answer questions raised in the discussion.

**Invitees:** The audience for a design review should consist of stakeholders (i.e. people whose work is affected by the design being reviewed) and all attendees should have a technical awareness of the issues being discussed. The audience should be kept small enough to ensure that all participants can express technical concerns with the design.

**Requirements:** It is important that design reviews take place at a time when the software requirements have been formulated, although these requirements may not be finalized; and, in fact, one consequence of a design review may be to change the requirements if it is found that the existing requirements cannot be met. A written copy of the requirements (in their current state) should be available for discussion during the design review.

**Background information:** Prior to the meeting, attendees should be given a written statement of the proposed design, including UML and other diagrams and any relevant requirements. In addition, it is often useful to provide preliminary design information, such as the results of a prototyping effort, relevant performance measurements, and even customer feedback, at a design review. Visual aids can help clarify complicated aspects of the design.

**Outcome:** An attendee involved in the design should write minutes, and/or make notes of any issues that arise during the meeting. Action items should identify persons responsible for resolving issues identified in the meeting. A summary of the meeting results and follow-up issues should be distributed to attendees, and others affected by the results, after the meeting.

# Design Reviews in CS493

## Schedule for design reviews

Friday, November 4: Thomas
Monday, November 14: Jesse
Monday, November 21: Alex

## Format of design review in CS493

The person being reviewed will share their schedule, SRS document, specification, environment survey, and one module design with the class.

Before class, the reviewers will read the module design and the specification, referring to the SRS document and schedule where necessary. The reviewers will prepare questions to bring to class.

During the class period, the first 40 minutes should be spent in discussion with the person being reviewed, and the reviewers should get answers to their questions as well as any clarifications they may have. The reviewers should also verbally offer suggestions for how the design of the project could be improved.

The next 10 minutes should be spent by all members of the team writing up the answers to the following questions. Your answers may be of any length, but should be justified by reference to specific parts of the design document (and specification and use cases, if appropriate). Write these answers in a GoogleDoc and share that document with the presenter and instructor (ShereenKhoja@gmail.com).

1. Are there any inconsistencies in the design? That is, does the document contradict itself?
2. Are there omissions in the design? That is, are there elements that are mentioned but never discussed, or obvious pieces that are missing?
3. Are any parts of the design unclear? The standard should be that given the design document, a competent programmer could code the project. Note that "clear" does not mean that the document must be very detailed. We assume that a decent computer scientist can fill in missing details, provided the overall document is clear enough.
4. Are there technical errors in the design? Is there any statement of fact that you know is false?
5. Has thought been given to testing? How would you test this design? What, if anything, could be done to make the design easier to test?
6. Does the design make realistic assumptions about the environment? That is, will the developer have trouble getting access to important external components (e.g., specialized hardware) and are the systems the project needs to interact with suited to the purpose?

7. Does the plan seem realistic? Are tasks at a reasonable level of granularity and is it clear what each task means? Do the time estimates seem appropriate? Do any parts of the plan seem risky in the sense that they are likely to become a bottleneck to further progress?
8. Any other comments?

Please note that reviews should be written with a professional tone, and suggestions, technical or otherwise, are most welcome. By writing a review for another developer, you are contributing to their project by becoming collaborators. You essentially become part of their team. Although you do not contribute to development, your insights are very valuable.

**Bonus:**

You get another FREE change request based on issues found in the Design Review. The change request must be tied directly to at least one problem listed in one of the GoogleDocs written by your peers.