

## Final Presentations

**Released:** Wednesday, Apr 17, 2024

**Due:** 9:30 am Wednesday, **Apr 24**, 2024

### Learning Objectives

- Reflect on your semester-long design process and generalize them towards actionable lessons learned.
- Describe concepts of cross-team collaboration, multi-team development, design evolution, and testing.

### Recommended Resources

- Lecture slides throughout the semester
- Recommended reading from the lectures (see the lecture slides and course website)
- Example talk: [https://canvas.cmu.edu/files/11086390/download?download\\_frd=1](https://canvas.cmu.edu/files/11086390/download?download_frd=1)

### Tasks

For your continued growth as a software engineer, you will need to continuously reflect on how effective the techniques that you use turn out to be, and how you can improve your design process. The final presentation is dedicated to practicing this process of retrospective reflection and identification of areas of improvement. To accomplish this, the major part of your presentation is to share what went wrong, what went right, and what you can learn from this experience for future projects.

Please prepare a **15 min** presentation (make sure you finish on time!) that focuses on sharing your lessons learned from the project. There will be a **3~4 min** Q&A session after your talk. For each lesson learned, please include and clearly separate your **observations** from your **interpretation** towards more generalizable guidelines that you or other students can apply for future projects.

The **observation** aspect should include a concrete description of the situation (i.e., available design options and the context that is necessary to understand these options), the decision you made (e.g., selection of process, method, or design decisions), and the consequences you observed. The **interpretation** aspect should include a reflection on whether you think the decision you made was appropriate and your lesson learned from this experience, which should be actionable and based on your observations. You can see an example of a talk like that here:

[https://canvas.cmu.edu/files/11086390/download?download\\_frd=1](https://canvas.cmu.edu/files/11086390/download?download_frd=1)

We encourage **every team member** to participate in the presentation. You do not have to evenly split your time, and it is up to you if you want to divide the talk into sections or present each section as a “dialogue” between multiple speakers.

Please include insights on the following topics. Sample questions that you could discuss during your talk are listed below (you do **NOT** need to answer every question listed). You are also free to deviate from these questions if there are more interesting things to discuss in each category.

### **Task 1: Your Lessons Learned from Cross-Team Communication**

Observations: How did you structure your communication? How did teamwork across multiple teams go? What challenges did you face when communicating with other teams?

Interpretations: What have you learned from communicating with other teams? What advice would you give to next year’s students? How would you improve cross-team communication when working on your next cross-team project?

### **Task 2: Your Lessons Learned from Multi-Team Service Development**

Observations: Which design principles or design techniques did you apply to structure the high-level architecture of the system and to ensure compatibility between services? How well did integration go? Were there any issues caused by implicit assumptions, interface incompatibility, or other integration issues?

Interpretations: What have you learned from developing a service in a multi-service project? What advice would you give next year’s students? Which design techniques would you use in your next large-scale project and how can you improve the architecture consistency in multi-team projects?

### **Task 3: Your Lessons Learned from Design Evolution**

Observations: Which design decisions have changed throughout the project? How well did the consequences of your design decisions align with your evaluation before implementation?

Interpretations: If you could go back in time, which design decisions that you made would you change? What have you learned from designing a software system over multiple iterations under changing requirements? What advice would you give to next

year's students? How would you improve your design process to be more effective for evolving architectures (e.g., which design techniques would you use in your next large-scale project)?

#### **Task 4: Your Lessons Learned from Evaluating Large-Scale Software Systems**

Observations: Which design evaluation & testing techniques have you used throughout the project? Did your test find any bugs that you didn't notice before? Were your quality attribute evaluations different from what your design evaluations predicted? Did the process of writing test cases change your perception of design?

Interpretations: If you could go back in time, would you spend more or less time on design evaluation and/or testing? What have you learned from testing quality attributes and from testing the integration of multiple services? What advice would you give to next year's students? How would you improve your design evaluation and testing techniques for your next large-scale project?

#### **OPTIONAL Task 5: Comparison with Recommendations from Literature**

To receive 5 bonus points, for at least one of the lessons learned, please describe what related software design literature (e.g., research papers, books, blog posts, guidelines from companies, or podcasts on the topic of software design) recommends and whether this is consistent with your own recommendation or if your recommendation deviates from the literature recommendation (if you deviate, please explain why). Please link the source of the recommendation from the literature that you are referring to. To practice studying available literature yourself, please pick a literature recommendation that was not explicitly mentioned in the course lectures.

Possible starting points for you to identify available literature are recommended readings from the lectures. Also, Google Scholar is a popular search engine for academic literature. Note that you do not have to conduct a complete literature review on the topic. It is sufficient to find one good source.

This task is **optional**. You can still receive full credit without doing this.

#### **Tips**

You are also welcome to include other content in your presentation and you are free to choose a format and structure for your talk. However, since you only have 15 minutes to talk about a semester-long project, we recommend that you focus on the important

insights & lessons learned. Make sure that the observations you include are relevant to your interpretations, and the interpretations clearly follow from the described observations.

As you can see from the grading rubric, we do not evaluate slide design or verbal delivery, so please focus your time on the quality of the content rather than visuals.

### **Deliverables**

Submit your slides as a PDF file to Gradescope and verbally deliver your presentation in class. We recommend that you use your own laptop to present your talk by connecting to the room AV system.

### **Grading**

This assignment is out of **100** points. For full points, we expect:

- **(50 pt)** Insights clearly demonstrate a deep understanding of software design by referring to appropriate concepts, design principles, and challenges of software design
- **(20 pt)** All four insight categories are included in the presentation.
- **(10 pt)** Interpretations are separated from observations.
- **(10 pt)** Interpretations are clearly based on observations.
- **(10 pt)** Presentation ended on time.
- **(5 BONUS pt)** for including a comparison with existing literature on software design