# Presentations

Throughout your time here at Revature, you will need to make various presentations, at least one for each of the major projects you are assigned. These skills are expected in modern programmers because as you grow in your career, you will need to make presentations showcasing the projects and plans for which you and your team are responsible.

The intended audience for these presentations may vary significantly. It may be a technically experienced audience like your team or manager when explaining how to implement a new technology or advocating for a particular design or feature in a project. However, you will also likely be expected to present to non-programmers, including business analysts, sales representatives, managers, department heads, and others explaining the capabilities of your proposed software/implementations and how the end user can utilize it. Mixed audiences are also highly possible, so the below will provide a structure and suggestions that should work across various levels of programming experience. Still, you should always keep whom you are presenting to in mind and adjust these guidelines to reach your intended audience best.

## Basic Format

The basic format we expect during showcases combines live demos and slides. The most common ordering of these elements is starting with your slide deck so you can introduce the context for the project, discuss your designed solutions to the project's requirements, and demonstrate the project in action. Finally, after the demo, you should address future plans and iterations alongside offering an opportunity to take questions from the audience.

The above order is helpful, particularly for new applications or proof of concept presentations that address use cases widely understood by your audience and where the features and design choices are broad enough that they can easily be discussed abstractly before the application is demonstrated. However, in cases where the presentation focuses on highly technical aspects of the application or where the application addresses a need that may not be easily understood, it may be better to bring forward the demonstration of the application and then discuss how the solutions were achieved.

### Suggested Presentation Flow:

- What the project's name is and who the team behind it is. This can be an opening slide with the project name and the name of those who worked on it.

- What was the problem presented for you to solve/What were you and your team tasked with achieving? This provides the context for everything that follows.

- How did you and your team organize to provide the problem's solution, and what did you use for it? Talk about your SDLC process, break down the team structure, and mention the different technologies leveraged for the solution.

- What was your solution? This part should be an overview of what you did. This is likely to overlap with the demo, but this is where you prepare the audience on what to watch for in the demo. It also lets you mention cumbersome things to demo or that would require many repetitions.

- Live Demo. Once you have covered the outline, you demo the result of your project.

- Future Iterations & Challenges. After the demo is an excellent time to return to address challenges faced and future development plans. See below for more on how to handle this section of the presentation.

- End Slide prompting for questions. This section should note the end of the presentation and, if applicable, that the floor is open for questions.

## Components of a Successful Presentation

Below is a discussion of the various elements critical to a successful presentation. Each section will discuss the expectations for including that element in your presentation and guidelines for effectively integrating that element.

### Attitude

When it comes to giving a presentation, the attitude of the presenters is essential. If you are bored, nervous, or do not care about your presentation, the audience will notice and reflect that. Therefore, you want to ensure that you present the project with genuine professionalism, excitement, and enthusiasm. You should avoid overly exuberant declarations or actions to not come across as fake or manic either.

Humor is especially tricky in this regard. It is beneficial when used appropriately in keeping an audience's attention. However, humor is highly subjective and difficult to achieve with an unknown audience universally. Inside jokes, highly technical jokes with a non-technical audience, or anything pejorative may be amusing for the presenters but will likely alienate audience members. Ultimately, humor is a tool best used only sparingly, if at all, and must be appropriate for the professional context.

### Slides

Slides are very often misused. This has led many people to dislike PowerPoint presentations in general. The vital thing to note is that slides are not the basis for your presentation; they supplement it as a visual aid. To this end, the slides should be constructed with that in mind.

- Slides are a visual medium: you should avoid walls of text and make effective use of imagery to supplement your speech.

- When text is on a slide, it should supplement what you are saying and not be repeated verbatim by the speaker. An audience can read faster than you can talk so reading from a slide simply results in a bored audience waiting for you to catch up.

- Slides should not be identical slide after slide; this predictability aids in an audience growing bored. Instead, mix things up so each slide feels fresh.

- Just as slides should not be identical, they should have a common theme; it should not feel like each slide was pulled from a random slide deck.

For pre-themed slides, [https://slidesgo.com](https://slidesgo.com/) is a free source of visually attractive slide templates that can aid your endeavors.

### Live demo

The live demo portion is critical to showing your work in action. Ensure a logical flow to your demonstration, which will also be discussed in the narrative structure portion. In addition, the live demo should be a chance to show off your project's functionality.

Only some things, though, work well in a live demo. If, for example, you implemented security so that ten different people logging in would have access to ten different things, that is likely to grow stale quickly. Show off just one of these instances and mention the others. In cases where the specifics are less demo friendly, speak about them in the slides.

### Challenges

Inevitably a feature or even the whole project just does not go as planned. It is common to want to defend your work and justify the issues, and often developers seek to do so in a showcase. It would be best if you did not do this. The showcase is not a place to air grievances, criticize other teams/members or make excuses.

1. The audience is likely completely unaware that the feature you demonstrated only had half the functionality you planned. Telling them this only reduces your stature in their eyes, not enhances it.

2. Criticizing the work or choices of past teams, or worse, current teammates comes across as wholly unprofessional, no matter how justified you may feel it is. But, again, there are better times and places to address it than presentations.

3. Constructive criticism is essential to the success of developers and teams but should be reserved for the Sprint Retrospective meeting or its equivalent, not a presentation or showcase.

If you are missing a key feature, it is better to address it not as a failure of the current iteration but as a feature to be implemented in a future sprint. Frame it as to be done, not as something failed to be done. This can address audience concerns about why a feature mentioned as one of the critical use cases is missing, and not hurt their overall impression of the project.

Finally, if you really must address a challenge, address it in a way that emphasizes the solution, not the problem itself. If a critical feature had a significant delay and resulted in it only being partially implemented, address how you worked to overcome it and the solution the team produced to solve it. This allows you to manage the concern while highlighting your success in dealing with it rather than just dwelling on the issue itself.

## Narrative Structure

When you are doing a project demo, it can sometimes be unclear to an audience member what the purpose of it is without additional context. A narrative structure can help with this and provide a logical flow to the demo.

For example, someone jumping between three distinct features may provide information about those features. Still, if you instead weave a narrative, it can give the use and why we care about those three features context, making the features far more memorable and impactful.

Example:

"Today we want to introduce Jane, they are a customer of the client, and when they first open the app, they see that they are prompted to log in. Having the customer log in allows us to maintain a history of their past orders and support tickets so our service agents can quickly assist Jane should she need it. Upon logging in, Jane wants to check her reward point balance, which we can see in the top left of the screen. Jane notices she has enough points to redeem for a free product. By clicking the redeem button, we can see a list of all redeemable rewards; those requiring more points than jane has are greyed out and un-selectable...."

In the above example, we know a few things from the context given. First, it is an app used by a customer, we track order and support ticket history internally for our service agents, and customers must log in to the app. Once inside, we have a feature that displays reward points and a redeem feature that displays the rewards available to customers. Finally, we navigate through the app as an actual user would, further understanding the app's use and purpose.

Without the narrative context, we still know parts of that, but if the audience is unfamiliar with the business and project, they might be lost on what precisely some of these features are. If we then demonstrated the reward points counter, the login, and the redeem option, we might also lose the audience in terms of how it all ties together.

A narrative structure gives an easy-to-understand context for the project and how it works and an easy-to-follow logical flow.

## Practice

Once you set up the outline of your presentation, created the slides, and decided on a flow for the live demo, we must practice. Invariably something will happen when doing it live. The more you have practiced the presentation, the less likely it is to happen, or if it does, the better you can recover.

Practice allows you also to find things that do not work or will break that you might have missed beforehand. For instance, you might realize that you have no mock data in the demo, which could be a significant issue in a live scenario. Additionally, something that seemed OK written down might come across as awkward or clumsy said aloud. Finally, the more you practice, the more polished and professional the final presentation will be.

Rarely, over-practicing can also be a problem. If you have practiced your presentation more than a dozen times, it can become so memorized and repetitive that it can come across as dry and unexciting because you have become bored with it. The usual recommendation is to practice a standard presentation between 5-8 times, possibly more if it is highly technical, but usually not more than 10-12 at most.

### Timing

Time limits are essential; you need to be able to address the topics and efficiently convey your message. Talking endlessly and repeating yourself all make for a dull presentation. Keep it short, snappy, and engaging, and not longer just for the sake of being long. Also, be aware that well-done practice sessions will generally be faster than the live presentation. When you are in front of an audience, nerves, pauses for audience reactions, and other things will eat away at the time. So, make sure you are giving yourself a buffer.

### Time-Warping (The Cooking Show Method)

Sometimes you will find when doing a practice run that you are losing time waiting. The page is taking a while to load, the callout to an external service can occasionally hang, or there is a bug that made it into the system, so a page fails to load one out of five times. The latter would be something to fix, but code freeze has taken place for this sprint, and you want to avoid that happening during the showcase.

Much like a holiday cooking show demonstrating how to make a delicious turkey puts the uncooked turkey in one oven and then opens another to show the result, we can do the same. You can identify spots during practice and have those pages loaded up with the results ready. For example, this is useful when you want to show a record being created but want to avoid filling out 10+ required fields and have everyone watch you type in the information for a minute. You can start it off, showcase the save button, then switch tabs to a completed record to continue with the next part of the demo.

### Scripts

Scripts are a double-edged sword in presentations and should be avoided wherever possible for the actual presentation. Writing one ahead of time to figure out the flow of what you want to say, followed by practice, allows you to separate yourself more from the script at presentation time. In addition, you need to address the audience rather than looking at a script off to the side.

Additionally, a reason to move away from scripts is that if you memorize a script and lose your spot, and can start a mental chain reaction, and suddenly you are blanking in the middle of it all.

Finally, script reading, aside from being painfully obvious, is also often very wooden and makes you unable to convey the excitement and engagement with the audience we want you to have.

I recommend creating a general outline of what you want to say and then trimming it down to a series of bullet points. You can glance over to remind yourself of the topics you want to touch on, and trust your expertise in the project to elaborate as needed on them. Practice will further help you become comfortable with what you will be saying.

An exception to the above may be someone that is becoming conversant with the language that the presentation will be given in. For example, someone that takes long breaks in sentences or many verbalized pauses ("uh," "um" etc.) may want to consider using a script, so they do not have to translate the words they want on the spot.

## Q&A

Finally, at the end of a presentation, you will often be asked questions, certainly in a project showcase. This can be far-ranging, and you might want to delve into the specifics of certain features. When a team is answering questions, we do not want team members speaking over one another, nor do we want everyone to remain quiet, expecting someone else to answer. To solve this, we use the team lead as a router, channeling the question to the person most familiar with the topic. In larger teams with multiple sub-teams, the team lead will direct it to the relevant sub-team lead, who then directs it further as needed.

A word of caution, however: whoever is routing the questions should be passing all questions to others instead of answering the questions themselves. It can be tempting for the router to answer all the questions they know the answer to, but this can make the team appear unengaged or unknowledgeable. Also, you may only be able to anticipate some of the questions, which may not all be technical questions. So have plans for whom to pass random questions to and what team members are comfortable answering non-technical questions about the experience or process the team went through.

In cases where the question is asked of multiple people, the team lead should still be directing it, calling on people to give their answers in turn. This allows for a neat and orderly Q&A.