Client Meeting 2 Transcript

Participants: Winn, Stella, Animesh, Lulu, Jenna, Federico, Dip

---

Lulu: Meeting in session. Basically, we've been working on some models for the product so far. We have a couple we could show you if you're interested—like a MoSCoW table.

Winn: Yeah.

Federico: Should we start with the models or the MoSCoW table?

Lulu: We'll just show the MoSCoW table.

Jenna: We made some updates from the Word doc.

Winn: Yeah, sure.

Jenna: And we've put it in a better format so it's easier.

Winn: Which group are you?

Stella: The first one.

All: Group 6.

(MJ walks in.)

Lulu: Basically, we just clarified a couple of things from your notes, like how the updating works for the subject coordinator. It's not automatic—they get the option to approve updates.

Winn: Yes.

Jenna: And we removed the student, because they don't actually use the system.

Federico: The secondary actors.

Winn: That will be easier, yep.

Federico: And we only kept the "must haves" for now.

Winn: Yes, definitely.

Lulu: So are you happy with the workable system, and then we can expand further later?

Winn: Yes, exactly.

Lulu: So that's all—we feel pretty much on the same page with that.

Winn: That's great, that's wonderful.

Federico: Then we have the motivation model. Are you guys familiar with it?

Winn: Yeah.

Federico: We start at the top with the "use scale" product for our project. It should be awesome, it should *feel* awesome. There are three main actors: students (but only when signing in), subject coordinators, and the admin.

Jenna: For the emotional aspect.

Federico: And they each have their respective to-dos and feelings.

Lulu: These are like their user stories and how they're going to interact with the system. The admin also has all their capabilities outlined.

Jenna: And those are the use cases.

Lulu: We included the subject so you can be proud and ensure it's widely adopted.

Winn: Yeah, sure.

Lulu: We've got some front-end design mockups.

Jenna: We've got two options. We heard you say "drag and drop" a lot, so this one is based off Draw.io. There are different filters you can go through—you can search, and things will be tagged—but you can also expand different sections of assignments, add a new row with a new task, and then drag and drop the level into the AI use scale. The rest will populate.

These boxes will be editable (for adding specific assignments), renameable, and exportable. So this is the drag-and-drop version.

We also have one without drag-and-drop—we want to see which one you prefer.

Animesh: This other one is really similar to the Excel template you gave us. You have the option to add new rows, but instead of drag-and-drop you get a pop-up with templates to select and insert.

Jenna: So this one shows less information on the screen at once, but the other one is more intuitive. Do you have a preference?

Animesh: It will also be long, so there will be a scroll bar.

Winn: Mmm.

Lulu: I've got a side-by-side. This one has the repository with drag-and-drop, as opposed to a pop-up.

Winn: So that one's more like building a table—one row at a time—while this one is dragging components in.

Jenna: But you could also go row at a time.

Lulu: It will have the headers automatically populated.

Winn: Remember, there are two things:

- For the scale itself, we usually organise based on levels.
- For the student version (where it's mapped to assessment tasks), we start with the assessment task, not the levels.

So I think one is good for building the scale itself, but the other is more intuitive if I'm thinking about assessment tasks.

Jenna: This could be similar: when you press "new row," it asks you to input the assessment type first, and only then creates the row.

Winn: Yeah, because the subject coordinator would likely create the assessment task first, and then think about which level fits.

Federico: Quick question: when you say "create the assessment task," you're basically just naming it and then picking the level for it?

Winn: Exactly.

Federico: So it's a two-part process.

Winn: Exactly. Create a row (a task of a particular assessment), then assign a level.

Lulu: So are you envisioning them creating a full table first with the tasks outlined, and the rest blank, and then dragging levels into it?

Winn: Exactly. I imagine they'll create the assessment first, then create rows for specific tasks, and then fill in those rows. When filling in, they can drag the level in or select from a dropdown. From there, some details will automatically fill in, and they can edit the rest. Once they've created assessment tasks with levels, the system would automatically generate the actual scale.

Jenna: Yep—just based on the levels they actually used.

Winn: Exactly.

Jenna: What if, for this one, you create all new rows and name them for assignment types first, and then once that step is done, the drag-and-drop appears? Would that be better since there's less information at once?

Winn: Yes, definitely. Create the assessment first, then think about the levels.

Animesh: So are you guys happy with that design?

Jenna: If we make some changes during the week, I can email you for a quick check before we start building.

Winn: Yeah, sure. Sometimes it depends on the actual prototype—if we can try it, we can give better feedback.

Dip: Should I show the prototype? This is a very basic prototype—like a proof of concept. You can log in, then create a template (say, for a maths assessment). I've got two pre-made for the demo. Right now it's just a text box, but later you'll be able to drag. I'm mostly backend, so the way it works is it generates the actual template in JSON, and then sends that back through the backend to create whatever format you'd like.

Winn: A few things. First, we have to use the terms more carefully—it gets confusing. When you say "template," I think of something the *admin* creates. A template means something to start from and finalise.

The admin might create templates for common assessment types, so coordinators don't have to start from scratch. Then subject coordinators create the actual thing. Later, if they want to reuse it, they can save it as their own template.

Animesh: So like a basic template that the admin creates, and then the subject coordinator can play around with it.

Jenna: Right—so this is sort of a template the admin chooses.

Winn: I think the full saved version is the template. For example, if I teach the same subject again and again, I'd want to reuse the same scale by saving it as a template.

Jenna: Would it be sufficient if templates were linked to all the information, so everything autofills? Or would you prefer the template to be very clear before you click it?

Winn: I think the template is basically one full version of the scale. If I'm teaching the same subject again, I'll reuse it, but I might modify it. I'd base it on the previous version and then save it as a new one.

Lulu: So it's the whole thing.

Winn: Yes.

Lulu: Can I just clarify—what's your term for the final product that the subject coordinator creates? Is it the use scale?

Winn: When we say "use scale," it's only the levels. The whole product is the "AI Use Declaration Form" (for students to fill in). That's easier for them to understand.

Lulu: So the whole thing with tasks and levels—

Winn: That's the declaration form.

Jenna: And export both every time?

Winn: Exactly—export both.

Jenna: We might need to make some amendments.

Lulu: So overall, are you happy with this editing setup—with the repository of use scales being collapsible?

Jenna: And when you first use it, it's a template, and if you want to change things, you use drag-and-drop.

Winn: Exactly. The idea is to take the minimum number of steps to achieve what I want. It has to be intuitive—so I don't need to learn the UI. I just open it, add or change things, and move on. If I modify something, it should be easy to find the resources I need.

Jenna: That's where the repository comes in for modifications.

Winn: Exactly. I imagine coordinators care more about functionality than presentation.

Lulu: Ok, cool—that sounds good.

Winn: The only tricky part is subjects. All use scales need to be attached to a subject. The subject could just be a tag. The other tricky part is versioning. If I make changes, am I saving to an existing version, overriding, or creating a new one?

For example, if I drag from a template already being used, I shouldn't override it. I should only be able to save a new version. That's something to think about on the backend.

Lulu: Cool. And as for the repository—you're still happy with the AI use scales that have already been defined, and that new ones can be added to it?

Winn: I think both. The system needs to know whether it's added by the admin or by a coordinator. Admin ones should be shared with everyone.

Lulu: Like a personal repository.

Winn: Yes. I'd base mine on the admin one, but the system should show differences between the admin's version and mine. That way, if the admin updates the default, coordinators are notified: do you want to adopt the new version, or keep your old one? If they keep the old version, it should highlight that it's based on an older default.

Lulu: That sounds good.

Winn: All good. Anything you want to add?

Stella: Actually, I was thinking we can categorise the levels based on types of assessment. But what if a coordinator thinks their assignments are different from the conventional ones, and wants to add more? I think it'd be easier if we just had features. For example, typical written assessments—but you can still view all types of levels.

Lulu: So if you're making something for a specific assessment, the most relevant ones show up, but you can also search the side?

Stella: Yes. Right now, it's not quite visible on the page what the other options are. I'd need to check all the other assessments to find what I want.

Winn: There are pros and cons. We don't expect coordinators to be experts, so this arrangement can guide them. For example, written assessments usually match certain levels. Coding assessments, different ones. But coordinators might want to drag across types. So it's good to indicate which levels are designed for which assessment types. Some levels might even be suitable for both coding and written assessments.

Stella: So can we have another tab that shows all the skills and variables?

Jenna: And maybe a filtering system—you could filter by “Level N” and see all the types at that level.

Winn: Yeah, so filtering is important. Always show the level, and which type it suits.

Lulu: So you’ve got the five main levels as headings, and the descriptions change based on the assessment type.

Winn: Exactly.

Lulu: We’re thinking you can scroll through to see more options. Cool.

All: Thank you.

Lulu: Meeting over.