Updated 02/15/2021

CADLearning®

Customer Lesson Planning Guide

4D Technologies

Table of Contents

TASK: Planning Your Lessons

Planning is an essential step in the video creation process, and especially when you are teaching how to perform a task in a software. You must know what you are going to say and do before you record; therefore, you will need to prepare ahead of time. The CADLearning model embraces these philosophies when creating video-based lessons:

- Roles & Goals: The program functions that are utilized by an individual depend on that user's role in the company. Fulfilling that role requires that the user learns the workflows necessary to perform their specific duties. Therefore, learning paths guide users as they master those workflows within a defined goal. Each goal is further broken down into logical medallions, badges, and lessons that provide incremental achievements that are validated and measured for 100% competency.
- Microlearning: Atomized lessons teach only one task or concept at a time and run an average length of two to three minutes. Task-based lessons will require use of the software, whereas concept lessons will require a PowerPoint presentation.
- Good Pedagogy: Every lesson begins by stating what will be taught and why, then includes every step in the procedure, stated clearly and concisely, and ends with a concluding statement that reiterates what was taught.
- Interface Interaction: Since every lesson teaches one specific feature or function, the parts of the interface that will be required must be tested and practiced ahead of time; this includes having all necessary files ready and preconfigured to the point at which they will be used at the beginning of the lesson. It also requires a specific style that follows the interface steps exactly.
- Incorporating Program Files: Demonstrating the functionality of a software means having all the necessary program, exercise, and demo files planned and ready for recording. Preparing these files properly takes careful consideration, especially when logins are required or examples of corporate/client documents are used. This will reduce or eliminate the need to blur confidential information within the videos.

Put simply, your task is to plan microlearning lessons that meet the requirements of a user's job role, follow good pedagogy, and accurately portray processes and procedures according to the product and its interface.

ROLES & GOALS: Custom Learning Solution

Most organizations have a unique set of workflows that lead to specific outcomes. No one person is responsible for all deliverables of a project, but each role has specific tasks and unique competencies that define it. Thus, creating custom role- and goal-based paths will ensure that the exact set of knowledge and skills relevant to your organization is tested and maintained.

Roles and Goals customize learning solutions that are unique to each user, team, and organization. By identifying required job competencies and mapping those competencies to desired outcomes, learning paths are created that provide incremental achievements, which are validated and measured for 100% competency.

Creating a Custom Roles-Based Workflow

1. Identify job roles
2. Map those roles to goals
3. Break goals down into specific tasks that produce a desired outcome
4. Match competencies that are actionable and measurable for each desired outcome
5. Add learning objectives

Identify Roles

The key to truly effective learning paths is to create a hierarchical structure of the knowledge and skills in your organization. To do so, you must first start at the highest level of the defined job role and responsibilities. CADLearning Roles can be assigned at an organization level or chosen by an individual, and are made up of one or more goals, which include a series of related medallions and badges.

Working backward is the key to building more effective learning paths. Start by listing out the jobs in your organization. For example, roles could include a Detailer, HVAC Installer, Pipe Fitter, or Mechanical Engineer. Make a list of the roles in your organization.

Set Goals

Look at each role, job description, and defined set of responsibilities. Ensure that these descriptions align with the tasks performed on project deliverables. Within the roles of every job, those requirements define a series of goals. Ask yourself the question, “The person in this role needs to be able to do *what*?” For example, the job responsibility might state, “working with BIM,” but the actual goal is to generate a facility’s energy analysis from a model. In this way, you identify what you want your users to achieve. For our Mechanical Engineer, they could be workflow-specific, such as Energy Analysis, Plumbing Design, or Mechanical Design.

Now list the goals that you want to achieve for one of the roles in your organization. For our example, “Energy Analysis” would be the name of the goal.

Organize into Medallions

Once you have determined the goals essential to a role, you can drill into the desired outcomes. What are the tasks that need to be accomplished in order to achieve those outcomes? Also, think about the core skills that each employee needs to achieve that outcome. Each of these is a task. In order to be able to perform an energy analysis using Revit, our Mechanical Engineer needs to master Model Preparation, Energy Analysis Settings, and Energy Simulations. Each of these tasks will be a medallion.

Break Down into Badges

Badges are where the tactical aspects of the workflow are defined as competencies. In all cases, the candidate must produce an outcome that is actionable and measurable and prove they are competent to complete it via an assessment. An actionable and measurable outcome is met when a skill is performed within a software program, which should be directly related to a workflow. Objectives for each badge need to complete the following statement: "At the successful completion of this badge, the user will be able to…" do what?

For example, in order for our Mechanical Engineer to master the Model Preparation medallion, we must think about the specific competencies necessary to prepare the Revit model to undergo an energy analysis. These become the learning objectives. For example, at the successful completion of this badge, a Mechanical Engineer performing an energy analysis in Revit will be able to prepare a Revit model by:

1. adjusting model properties, including wall, floor, roof, window, and door properties, to ensure their type and materials reflect the built environment.
2. working with rooms and spaces to ensure calculations for volume and air flow are correct for determining heating and cooling loads.

These two objectives now become two badges that lessons can be assigned to, named "Adjusting Model Properties" and "Working with Rooms and Spaces".

Identify and Assign Lessons

Once you have organized your goals, medallions, and badges for each role, it is time to plan the lessons that will include the knowledge and skills necessary to fulfill those badges. For example, the Adjusting Model Properties badge will be the first badge our Mechanical Engineer will work toward achieving. It will include these CADLearning lessons:

- Understanding Wall System Families and Types
- Understanding Wall Type Properties
- Understanding Floor Type Properties
- Understanding Roof Type Properties
- Modifying Door and Window Properties

Notice that, in order to perform the modification of the door and window properties, a user must understand what all the different types of properties are that are involved in the process. Also, each lesson teaches only one aspect of what it takes to adjust model properties—this is *microlearning*.

Working with the Spreadsheet

You have been provided with a *Consulting Production Planning Template* that will help you plan, organize, and track your lessons, both existing lessons that are already in the CADLearning Library and any gap content that is determined to be necessary to create. **Reach out to the CADLearning Team, who will help you to develop this content based on your objectives.***

1. 4D & Customer: Start with the **Roles and Goals Brainstorming** sheet. In the **Discovery Center** section, brainstorm the roles, goals, medallions, and badges that you need in your organization.
2. 4D & Customer: To the right of that, organize your roles, goals, medallions, and badges into an assigned column for each (see examples).
3. 4D: Copy and paste the lesson list from the product library and assign relevant information to each lesson that will be used, including the badges, video length, and URL.
4. 4D: For each role, duplicate the Role Data tab, rename it, and fill in relevant role information.
5. Click the **Lesson Tracking** sheet. Assets are tracked in each column as follows:

    - **Topic:** enter the topic name that the lesson belongs to, such as Getting Started, Plumbing, etc. (The topic may or may not have the same name as the badge.)
    - **Lesson Title:** enter the names of all the lessons you plan to place in each badge.
    - **Lesson Number:** enter the production file name for the lesson, as instructed in the *Welcome Guide*.
    - **Badges**: For Badge 1, put in the badge information for that lesson, if applicable. If you have more than one badge for a lesson, use the **Badge 2** and **Badge 3** drop-downs.
    - **Raw Asset Status (RSA Developer):** (pink) fields indicate the status of the creation of your raw assets, including **Outline Written**, **Captured**, and **Delivered**. Use these columns to keep track of your lesson creation process (see this guide and the *Recording Videos Guide*).
    - **Lesson Video Status: 4D USE ONLY!** (blue) fields indicate which step in the 4D production process the lesson is in, such as Script Written, Copy Edited, Comments, Audio Recorded, Video Recorded, Video Edited, Approved, and Completed.
    - **Questions:** under Written, indicate with a Y when a question has been created for that lesson. Under Copy Edited, copy editor will indicate with a Y once the question has been edited. All lessons ***must*** have at least one question (see *Questions for Assessments Guide*).
    - **Approval:** For client approvals of the videos. Under Video Link, a link will be provided for each video that does require approval. (Not that not all videos will require approval.) Comments are used if the video is NOT approved, enter a comment explaining why.
    - **Video Times: 4D USE ONLY!** Video length data entered by production team.
    - **Exercise File:** contains the name of associated demo/exercise file or files that accompany the lesson.
    - **URL:** The 4D Production Team may enter an associated lesson URL, if applicable.
    - **Lesson Notes:** The 4D Production Team may enter notes about the lesson.

*Note that some columns/fields may not be used at all, and some may be added, depending on the contracted assets.

MICROLEARNING: Atomizing the Content

Each lesson should:

- Explain ONLY one aspect of the program
- Run an **average length of 2 to 3 minutes**
- Ideally, every one of your lessons will be less than 5 minutes in length
- Some lessons may be as short as a minute or less
- Lengthier topics may take as long as 8 minutes

IMPORTANT: If a particular lesson is longer than 8 minutes, it should be broken into separate lessons. In general, the shorter the lesson, the better. We refer to this as "atomizing the content."

How to Atomize Your Content

Think of the functionality of the program first: Each single option within your program could have its own lesson. Whether you turn that option on or off has an effect on how the program works, therefore it is a teachable lesson.

Of course, this may not always be practical—you must take into consideration common practices and workflows within your industry, as well as how the processes work within the program. However, keeping this "single option" concept in mind as you are planning your lessons will help you to break the processes down into easily digestible pieces.

Furthermore, the lessons do need to be as short as possible, but it is more important that each lesson cover only one task or concept within the program. For example, you may think of creating pipes in Revit as one task, however, our Revit library has at least 44 lessons about pipe. *Understanding Pipe Settings, Modeling Piping, Creating Piping Systems, Creating Pipe Connectors, etc.*, are all separate lessons having to do with pipe creation.

If you need help in determining whether a lesson is too long or short, or whether it is a knowledge lesson or task lesson, please consult with the Content Team, who will be more than happy to guide you.

Knowledge-Based Lessons

Knowledge lessons are used to answer the question: *What is it*? They can explain general topics and theories, both software-specific and non-software-specific. They can also explain workflows or processes, and they can be used to compare and contrast concepts across products.

EXAMPLES:

Software-specific concepts are high-level procedures within the program. In Revit, for example, families are used to define content within a model. Users must understand what a family is in order to be able to use the program.

Non-software-specific/general theories are concepts that users must master within their industry. Again, in Revit, architects need to understand rules of composition, such as the rule of thirds, to create renderings of their models with aesthetically pleasing image composition.

Workflows explain the high-level procedures that a program requires in order to accomplish a task. For example, in Revit, you must create a wall before you can create a door within that wall. The wall must exist before Revit will let you create the door.

Process explains the linear steps that users must complete within a program before they can perform the next step. For example, in Revit, you must create reference planes, then you dimension those reference planes, then you add parameters to those dimensions, and finally, you host geometry to those reference planes in order to create parametric geometry. Focus on how the program forces a user to accomplish a task in the program, as opposed to design processes, which can vary.

Compare/Contrast lessons explore the differences in the ways that various programs can accomplish the same task. They can also explain cross-product functionality. For example, both AutoCAD and Revit can be used to design buildings, but they accomplish this in completely different ways. AutoCAD is basically an electronic pencil that uses the CAD approach to create a set of drawings, while Revit focuses on creating a building model that the documentation can be generated from.

Knowledge lessons are typically accompanied by a PowerPoint presentation. See the *PowerPoint Presentation Guide* for specific instructions.

Task-Based Lessons

Task lessons are used to demonstrate the procedures within your application that are used to accomplish the task that is used to fulfill the concept. Task lessons are the atoms in “atomization” and will be the majority of the lessons you create. For example, in Revit, how to place a connector or valve in a pipe is a single task.

PEDAGOGY: Approach to Learning

A well-organized pedagogical approach ensures that each lesson addresses a specific outcome that is measurable and attainable. To do this, we recommend you write a S.M.A.R.T. objective for each. Smart Objectives are Specific, Measurable, Actionable, Relevant, and Timely. They state clearly and simply what the student is about to learn.

Put simply, a Smart Objective finishes the statement, “At the end of this lesson, you will be able to…”

In general, this statement will refer to a piece of knowledge or a task. To ensure you are teaching that knowledge or task comprehensively, we recommend you write a logical outline:

1. Introduction: define the task and how it works in the program (S.M.A.R.T. objective)
2. Explain why the task is important/advantages of completing the task
3. Note where the task/command name/interface elements is/are in the interface as you…
4. …Demonstrate the task, step by step
5. Conclusion: reiterate what the task is and why it is important

From this outline, further documentation can be created, such as a narration script or transcript of instructions. We also recommend that you add your S.M.A.R.T. objectives to the list of lessons in your *Roles and Goals Planning Spreadsheet*, though this is optional.

Style

CADLearning lessons are instructional in nature, and so they are written in the second person point of view (only the pronoun "you" is allowed) using a conversational tone. They sound like a natural speaking voice, yet thoroughly cover the procedure.

- Do not use "In this lesson…" in the introduction. Simply begin by defining the program function you are going to teach and why it is important.
- Do not make direct reference to any other lesson names or topics, or irrelevant functionality, including release numbers and dates. Also avoid referencing the product; do so only when necessary and use "the program" in your statement.
- Make the lesson as short as possible, using concise sentences. Avoid excess verbiage, such as, instead of, "click the **OK** button," or "click **OK** to close the dialog," just say, "Click **OK**".
- Include **EVERY** step in the procedure.

Interface

Correctly referring to the interface of the program is essential for calling out the procedure properly, and for eliminating ambiguity.

- Name the tools by their correct names, EXACTLY as they appear in the interface.
- Use the specific order: **ribbon > tab > panel > tool** because that is the order in which one would have to click to reach the tool—what we call "top-down selection". For example, when selecting a tool from the ribbon, use, "On the ribbon, **3D Model** tab, **Create** panel, click **Extrude**."
- In tree views or drop-down menus, use the term "expand" when describing the function of opening a level in the tree or opening the drop-down menu. For example, use "In the **Model Browser**, expand **Extrusion1**, then select **Sketch1**."

Program Files

Demo Files and Exercise Files

As you are demonstrating the product of your specialty, you may need to prepare a variety of files ahead of time that will be used during the workflow within the program. For example, if you are planning a lesson about how to add a window to a wall, you will need to have a file prepared ahead of time that includes a windowless wall. A more complicated example would be a lesson where a spreadsheet is used to drive the features of a parametric design—in that case, you would need to prepare both the design file and the spreadsheet file.

Demo files: program files and reference files that are used during the step-by-step workflow that is demonstrated in the video.

Exercise files: program files and reference files that are necessary to mimic the procedures as demonstrated in the video, and which are configured so that the student can begin at step 1 and continue to perform all necessary steps as shown in the lesson.

Note that your demo files and exercise files may be the same files, but not always.

Considerations for Naming and Saving Demo and Exercise Files:

IMPORTANT: Realize that, as you are recording, everything that you show in the interface will be seen in the video! Be aware of displaying corporate or personal information that may be sensitive in nature.

Strongly Recommended Best Practices:

> **Set up a generic email** if your product requires an online sign-in and you *do not* want your personal or professional email to show in the video, such as “poweruser567@gmail.com”.
>
> **Set up a generic folder structure** in your local system for any demo or exercise files that you will be accessing during the course of recording the lesson. Use folder names such as “my_project”, and for file names, use generic terms that summarize the lesson objective. For example, if you are demonstrating how to create a new project, use “MyProject”.

While it is possible to blur some sensitive information within the videos, this is a time-consuming post-production process that, if needed excessively, could result in additional production costs and delays in receiving your deliverables.

CHECKLIST

Please refer to this Lesson Planning Checklist to ensure that you meet all of the requirements for creating questions. (You may wish to print this page and keep it handy as a reference.)

- ☐ Identify job roles
- ☐ Map those roles to goals
- ☐ Identify job tasks that produce a desired outcome
- ☐ Match competencies that are actionable and measurable for each desired outcome
- ☐ Add learning objectives—best practice is to have learning objectives for every medallion, badge, and lesson
- ☐ Create an outline for each lesson that follows good pedagogy, adheres to the CADLearning style, and exactly follows the interface
- ☐ Plan and prepare all program files necessary for step-by-step demonstration
- ☐ Check for the display of sensitive private or corporate information that appears/will appear in the recorded video

Questions? Contact support@cadlearning.com