

Streamlining Branching Scenario Planning and Design

Christy Tucker, *Syniad Learning*

Sponsored by



Presented by



Technology Provider



STREAMLINING BRANCHING SCENARIO PLANNING AND DESIGN

Christy Tucker, Learning Experience Design Consultant at Syniad Learning, LLC

Table of Contents

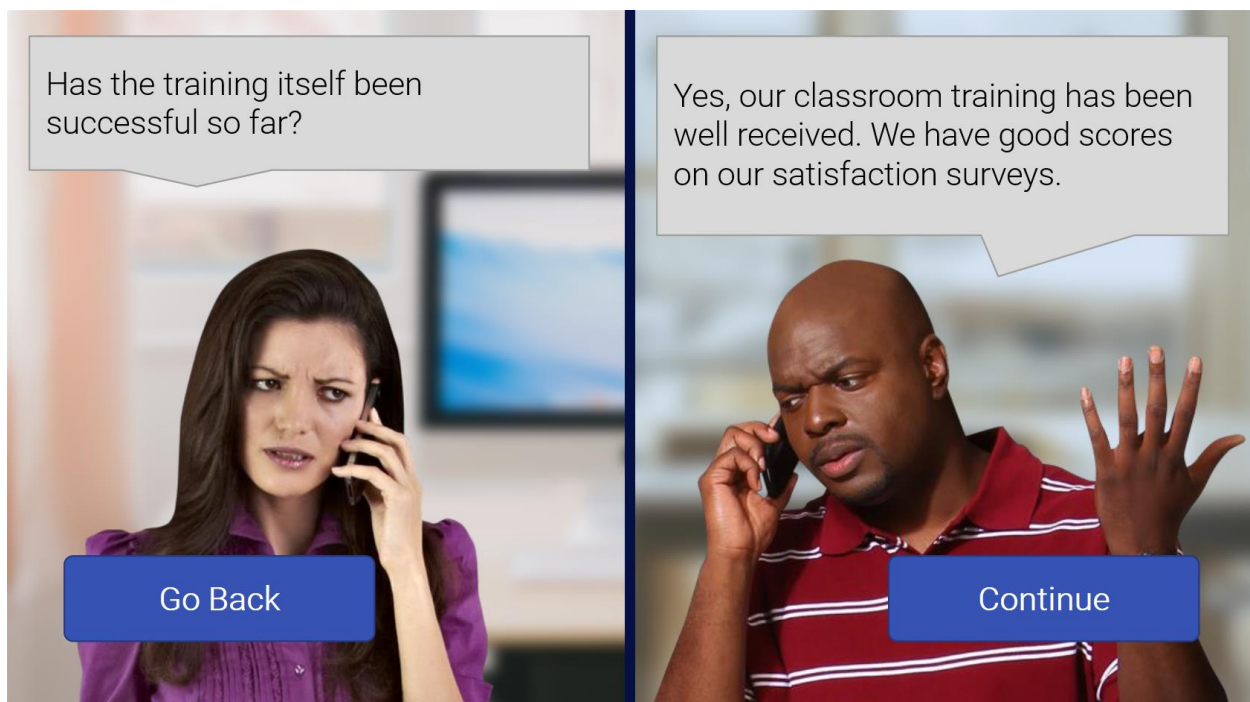
Branching Scenarios Defined	2
Analysis and Planning.....	3
Begin with the end in mind	3
The photo test	3
Identify your objective	3
Identify mistakes.....	3
Identify consequences.....	4
Questions to ask SMEs.....	4
If your SME gets stuck	5
Scenario concept and summary	6
Overall concept and problem summary	6
Outline of the ideal path.....	6
Mistakes.....	7
Organizing mistakes and consequences.....	8
Branching Structure	9
Number of options.....	9
Time cave: Broad but shallow	10
Branch and bottleneck.....	11
Branch and bottleneck example.....	12
Cutscenes.....	12
Opportunities to fix mistakes	13

Twine.....	14
Writing the scenario.....	14
Option 1: One decision point.....	14
Option 2: Ideal path	14
Prototype.....	15
References and Further Reading.....	16
Blog posts.....	16
Contact me.....	16

Branching Scenarios Defined

How I define it: A non-linear scenario where learners make decisions and see the consequences of those decisions.

[View an example of a branching scenario to try it out yourself.](#) This scenario will be referenced throughout as a sample of the process.



Analysis and Planning

Begin with the end in mind

At the end of the training, what do you want people to do differently? It's important to ask what you want learners to **DO**, not what you want them to **KNOW**. [Cathy Moore](#) has been talking about this for years. If we're aiming for behavior change, then we need to focus on what behaviors we want. It's not enough to simply increase awareness.

The photo test

[Julie Dirksen](#) talks about using the "photo test." If you took a photo or video of the desired behavior, what would it look like? For example, a client might ask you for training on "quality customer service" or "better communication between nurses and patients." As part of your analysis, ask what that really means. It's not enough to just get a list of principles or broad best practices. You need specifics and examples.



"Quality customer service" might mean cashiers asking customers if they found everything they were looking for and calling for someone to get it if they missed something. That's a specific behavior we can observe and assess.

"Better communication between nurses and patients" might mean asking open-ended questions to learn what concerns are most important to the patient. That's another behavior we can observe.

Identify your objective

Based on the behavior you identified, create your learning objectives. Here's an example objective. This is the objective for the sample scenario linked earlier.

Instructional designers will screen clients by following a repeatable process and asking probing questions to uncover the business need.

Identify mistakes

Ask your SMEs questions about mistakes. In a branching scenario, it's not enough to know what the right behavior looks like. You need to know the wrong behavior you need to change too.

If you have access to learners or people who have recently learned the skill, ask them too. They may have more insight than the SMEs.

The mistakes you identify become the distractors in the questions for your branching scenario. The mistakes and places people get stuck help you determine where to put decision points. If certain parts of the process are fairly clear and unproblematic, you can make those sections of the scenario passive review. That way, you can focus on what you really need to meet your objectives in the scenario.

Sometimes SMEs have a hard time switching from talking about abstract principles to describing behaviors. If they answer your questions about mistakes and consequences with broad answers, keep probing for specific examples and behaviors. You may have to ask these questions several different ways to get what you need.

Identify consequences

For each mistake you identify, find out the consequences. Ask your SMEs and sources this question.

What are the consequences if people make this mistake?

The consequences become the feedback in your scenario. Asking a patient a closed question rather than an open-ended one results in a one-word answer. Forgetting to ask customers if they found everything results in lost sales and less satisfied customers.

Questions to ask SMEs

- What are the common mistakes people make?
- Where do people get stuck in this process?
- Tell me more about that mistake. What do you think is going through people's heads when they do that?
- What does it look like when they make this mistake?
- What does that consequence look like in practice?
- Can you give me an example of how someone used this technique successfully? What were they able to accomplish by doing it right?
- Tell me about a time when you saw this happen in a real situation.
- What happened next?

If your SME gets stuck

What if you can't get anything useful from the SME? They're too busy, or they only give you one or two-word answers, or everything is abstract rather than concrete.

For whatever reason, if you're having trouble drawing information out from a SME, start writing something yourself. Do your research—review existing training materials, online articles, books, blogs, etc. Make your best guess and start writing a scenario as best you can.

The trick is, it doesn't matter if it's wrong. At this stage, you're just trying to get something other than a blank page. Ask the SME to review it and point out all your errors. Even a recalcitrant SME will have a hard time not correcting your mistakes—and now you suddenly have more realistic mistakes or consequences.

I've used this trick a few times, and I've discovered that the blank page is often the stumbling block.

Scenario concept and summary

I create a summary of the scenario and the narrative. This is included in the design document and signed off by the client and SME before I start writing. I want everyone on the same page for the basic concept of the scenario.

This includes:

- **Characters:** Give the characters names and roles. The backstories can be very thin, but giving them names makes them more concrete.
- **Problem:** I summarize the problem the main character faces. This should be a relevant, realistic problem like what learners will face on the job.
- **Rough outline of the ideal path:** The “ideal path” is what learners would do if they do everything perfectly. Part of the structure of a branching scenario is this “best case scenario.” At the design document or planning stage, I create a rough outline of the steps, but I don’t flesh it all out until later.
- **List of mistakes and consequences:** This list comes from your conversations with the SME and other stakeholders.

Overall concept and problem summary

The summary includes the name and role of the main character, plus any other critical characters. I describe the problem the main character faces and how it will be addressed. This is just a few sentences to give the overall feel of the scenario without getting into much detail.

Sophie is an instructional design consultant. She's tired of spending hours and hours writing proposals for clients who don't end up hiring her or really aren't a good fit in the first place. She has been contacted by Robert about a potential project. Sophie will attempt to follow a new process for screening clients to see if this is actually a good fit for her skills and to establish a professional relationship with Robert.

Outline of the ideal path

I start with a rough outline or checklist of steps in the ideal process. In this example, I'm creating a course on screening potential consulting clients, and I have a process with 4 steps. Each of these steps will be a decision point in the scenario.

It's possible that when I write the scenario that I'll realize I need to add another choice in this process, but this gives me the basic flow.

1. Send client initial screening questions.
2. Review client responses for fit and feasibility.
3. Learn more about client needs during preliminary phone call.
4. Propose a short road mapping engagement.

Mistakes

Based on my analysis (including conversations with SMEs, learners, and/or other stakeholders), I also create a list of mistakes or errors people could make. This list tends to be fairly fluid for me; I try to brainstorm more mistakes and problems that I'll actually use in the scenario. Some mistakes might be critical for the learning objectives, while others might be possible options.

I try to include both major, critical errors and some errors that are partially correct or in the gray area. Sometimes this list of mistakes also includes notes on consequences, although usually I have that in my notes from the SME.

I find it helpful to include both the outline and list of mistakes in the design document if possible. I haven't always done it that way, but it seems to help set clear expectations with SMEs and clients.

- Agreeing to a client request for a project before screening for fit (critical–must include)
- Sending client screening questions without a budget question
- Ignoring red flags in client responses (not enough money, unrealistic timeline, etc.)
- Rejecting a client because they don't know what they want (that's what road mapping is for)
- Jumping right into asking about project logistics without understanding goal/problem
- Writing a big proposal for free

Organizing mistakes and consequences

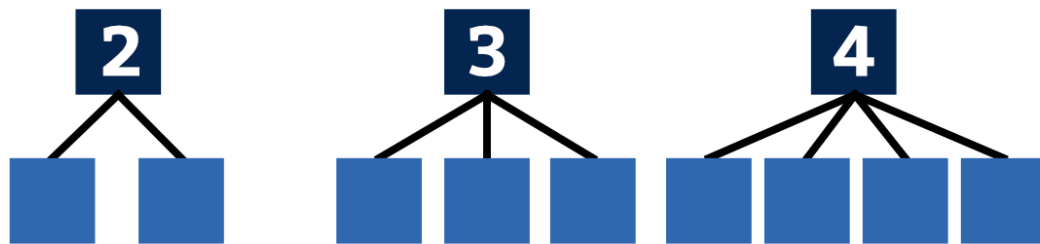
You might create a table for yourself like this to organize information you collect during interviews. This shows the objective plus mistakes and consequences.

Objective	Mistake	Consequence
Instructional designers will screen clients by following a repeatable process and asking probing questions to uncover the business need.	Agreeing to a client request for a project before screening for fit (critical–must include)	Bid too high and don't get the project Underestimate the scope and bid too low, losing money
	Sending client screening questions without a budget question	Client has very low budget; waste time on a client who can't pay

Branching Structure

The branching structure affects the complexity and the number of decision points. A decision point is each place in the scenario where learners make a decision.

Number of options



How many options in branching scenario decisions?

Multiple choice questions often have 4 choices in an attempt to reduce guessing. However, 4 choices isn't the best option for traditional multiple choice questions either. In fact, research shows that 3 options is the best for multiple choice. (Baghaei, P., & Amrahi, N. (2011), Rodriguez, M. C. (2005))

While this research is about quizzes and tests, I think we can apply it to branching.

The fastest way to reduce the complexity of your branching scenarios is to only use 3 choices for each decision point, not 4. Writing wrong answers and consequences is usually the hardest part, so you can just cut that down.

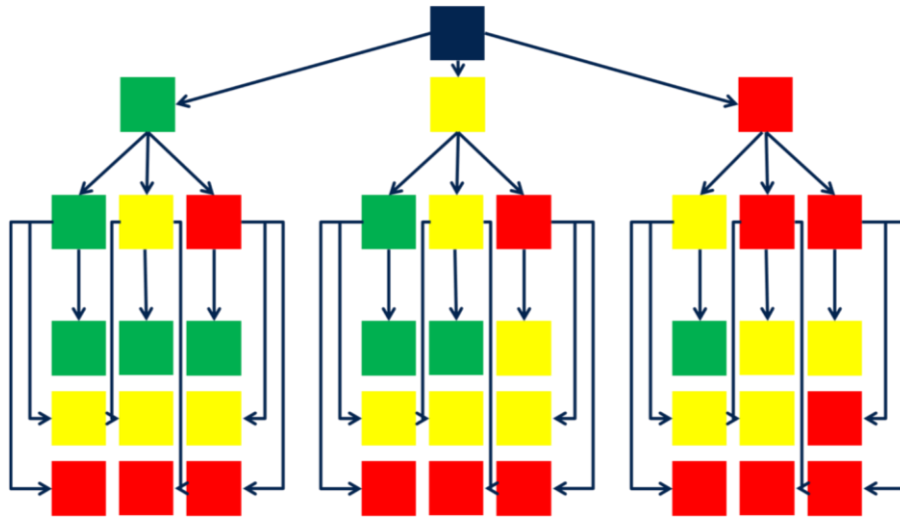
I usually aim for a Good or Best choice, a Bad choice, and an OK choice, although I'm not rigid about that.

In branching, sometimes you might have just 2 choices, and that's OK. I'd rather see two realistic choices than 2 realistic ones plus one ridiculous choice.

Time cave: Broad but shallow

This is the traditional structure for branching. Each choice leads to more choices, with no rejoining or reusing choices. This leads to numerous endings.

In this image, green are good choices, yellow are OK, and red are bad choices.

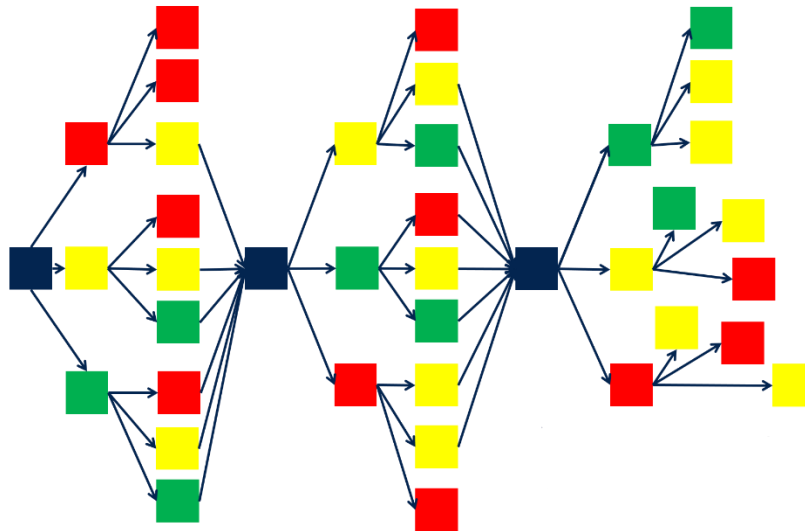


In the example above, players make 3 total decisions, but it's already 40 screens ($1+3+9+27$). The number of screens in a time cave increases exponentially. To add a fourth decision in the path, you need to add 81 more screens. By necessity, a time cave is generally limited to a maximum of 3 decisions in each path. It's broad, giving you lots of alternative endings, but it's not very deep. It doesn't do well for showing a longer process or change in a character.

When do you use this? Only if the scenario is really short. I have used this if you only need to make 2 decisions in a row, and it's very shallow. Otherwise, I don't recommend this structure.

People get stuck creating branching because this is what they think they need to do, and it quickly gets out of hand.

Branch and bottleneck



In a branch and bottleneck structure, you branch into different options for a while, but then all paths return to a single bottleneck. The bottleneck is an event or choice that happens the same in every path.

In the example above, there are 39 screens. Most paths have 6 decisions (plus two bottlenecks), although a few shorter paths end in failure.

The terminology “branch and bottleneck” comes from Sam Kobo Ashwell in his post on [Standard Patterns in Choice-Based Games](#). Although he’s writing about structures for games and interactive fiction, his concepts also apply to branching scenarios for training.

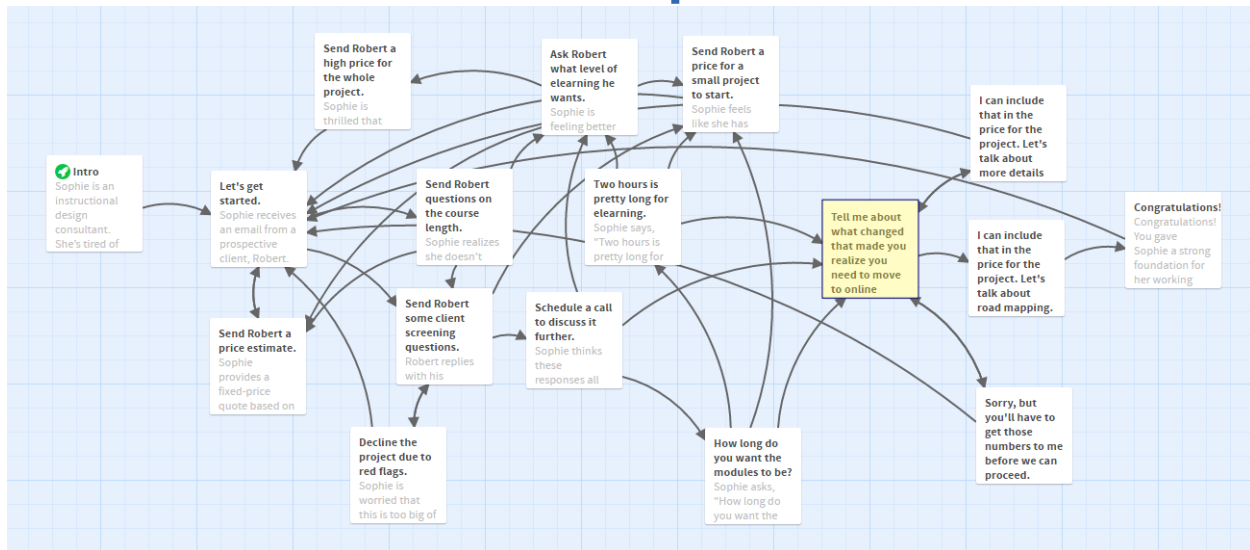
The branch-and-bottleneck structure is most often used to reflect the growth of the player-character: it allows the player to construct a somewhat-distinctive story and/or personality, while still allowing for a manageable plot.

Sam Kobo Ashwell

Ashwell notes that this structure often relies on states to maintain continuity in the narrative. You might track an overall score or adjust another part of the [consequences or intrinsic feedback](#) based on earlier choices.

Branch and bottleneck is a good structure if you have two options that could be done in either order (where you then backtrack to the other branch after completing one). It’s also good if certain key events will drive the plot forward regardless of prior decisions.

Branch and bottleneck example



In practice, the branch and bottleneck doesn't usually look as "clean" as the previous graphic. I usually reuse some choices. That gives players a chance to correct their mistakes without continuing down the wrong path indefinitely. It doesn't immediately force them back to the right path or to restart; it lets them learn from the feedback and adjust.

In my client screening branching scenario, I have one major bottleneck near the end, highlighted in yellow ("Tell me about what changed..."). Several other paths are failures that cause a restart. Mostly, the paths cross and converge multiple times as players can correct mistakes and get back to the ideal path.

The ideal path is 4 decisions deep. The complete scenario ended up as 20 slides in Storyline, 1/6 of the number of slides it would have taken to use a time cave structure.

Cutscenes

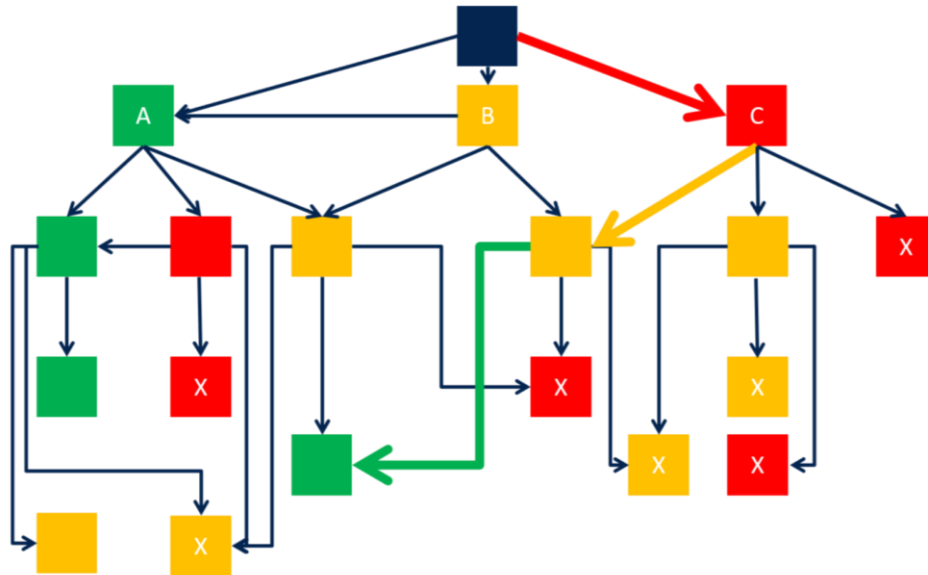
Another way to reduce the complexity is by using cutscenes.

In video games, cutscenes are short, non-interactive scenes that "cut away" from the action. They can move the plot forward, show conversations, or build emotion. We can borrow this technique from video game design when we develop branching scenarios for learning.

In branching scenarios with simulated conversations, sometimes we see every single exchange built as an interaction. With that approach, if you're controlling a character, then every single sentence that comes out of that character's mouth has to be a decision point. By using a cutscene, you simplify the scenario and reduce how many

slides you need to build. You can just show some less critical content, and save the decisions for the critical points that align to your objectives.

Opportunities to fix mistakes



How long should we let learners go down the wrong path? You don't have to build everything out 6 layers deep for the worst possible choices.

I don't usually like forcing people into getting immediately back on the right path, but I do like to give people opportunities to fix their mistakes. Most of the time in scenarios, we're working with gray area. In real life, we often have opportunities to change paths and correct mistakes. Where a single isolated mistake can be corrected, the cumulative effect of several wrong answers is the real concern.

I give people opportunities to get back on the right path by making better choices. If they keep going down the wrong path, they have to restart and try again. I won't force them to correct; learners need the opportunity to fail.

In this example, there are good (green), OK (yellow/orange), and bad (red) choices. If you choose C (red) at the beginning, you may reach a poor ending after just 2 choices. However, if you improve your choices, you can get back to a good (green) choice by correcting your mistakes.

This also helps manage the complexity by reusing choices. You aren't building completely new slides for every single choice; sometimes you just point back to an earlier choice.

Twine

The free, open source tool Twine is designed for creating nonlinear, interactive stories and games. It saves time in writing, prototyping, and reviewing.

Download the program from <https://twinery.org/>.

See my post [Twine makes branching scenarios easier](#) for step-by-step directions to get started in Twine.

Writing the scenario

After your planning and considering the structure, it's time to start writing. Where do you start? What do you write first?

Option 1: One decision point

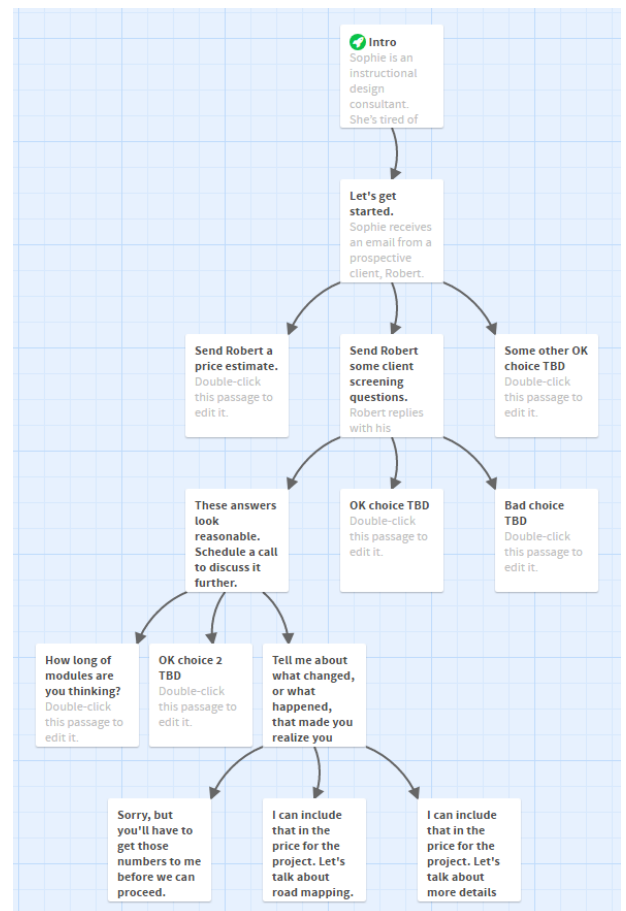
Cathy Moore recommends building out one decision point with all consequences first. If this is a large scenario and your clients haven't done scenarios before, that may be a useful prototype to get buy in and make sure the tone is right.

Option 2: Ideal path

If the whole scenario isn't very big (I have some that are 15 screens or less), and if your stakeholders have done branching before, you might not need to give them that single decision point prototype.

For me, writing the ideal path first makes the most sense. I have found that it's easiest to write the ideal path from start to finish first. I note decision points and sometimes draft bad choices along the way, but I don't fully write anything else until I finish the ideal path. The ideal path is the one that follows the best choices at every step and reaches the best possible conclusion.

You may want to do this as an outline in Twine first, not a full script. A sample ideal path with placeholders is shown here.



Once the first alternate path is written, go back to the beginning of the scenario. Find the first decision point that isn't fully fleshed out, and start writing there. As before, I try to write one complete path from start to finish.

When you're writing your mistakes and consequences, this is when you think about those structures I talked about earlier. Does it make sense to create a bottleneck at a particular choice? Where are you going to give people the opportunities to correct mistakes?

Prototype

One advantage of Twine is that it provides a fully functional prototype, where links can be clicked to test out the scenario.

View a [sample Twine prototype](#).

References and Further Reading

Baghaei, P., & Amrahi, N. (2011). The effects of the number of options on the psychometric characteristics of multiple choice items. *Psychological Test and Assessment Modeling*, 53(2), 192-211.

Rodriguez, M. C. (2005). Three Options Are Optimal for Multiple-Choice Items: A Meta-Analysis of 80 Years of Research. *Educational Measurement: Issues and Practice*, 24(2), 3-13.

Blog posts

I have written about many of these topics on my blog. Check out these posts for further reading, including the full collection of my posts on [Storytelling and Scenarios for Learning](#).

- [How to Get Started Writing a Branching Scenario](#)
- [Planning a Branching Scenario](#)
- [3 Tricks for Working with SMEs on Branching Scenarios](#)
- [What to Write First in a Branching Scenario](#)
- [Writing Mistakes and Consequences](#)
- [Managing the Complexity of Branching Scenarios](#)
- [How many options in branching scenario decisions?](#)
- [How Long Should We Let Learners Go Down the Wrong Path?](#)
- [Branch and Bottleneck Scenario Structure](#)
- [Branching Scenario Prototype in Twine](#)
- [Twine makes branching scenarios easier](#)
- [Branching Scenario in Storyline](#)
- [Color-Coded Branching Scenario Flowchart](#)

Contact me

Email: christy@christytuckerlearning.com

Blog: christytuckerlearning.com

Twitter: [@christyatucker](https://twitter.com/christyatucker)

LinkedIn: <https://www.linkedin.com/in/christytucker/>