

---

# When to Use Simulation Training, and When Traditional Training Is the Better Call

## Key Highlights

- The real question is not simulation or traditional training; it is which method builds the outcome you need. Simulation builds procedural skill in a specific system. Classroom sessions, e-learning, and job aids do other jobs better.
- Simulation earns its place when a task is high-stakes or high-frequency, cannot be safely practised on the live system, or is rolled out at scale and changes often. That is where hands-on practice pays back.
- Traditional methods are the right call for concepts and the “why”, for behaviour and soft skills, for awareness, and for one-off or low-stakes tasks. Building a simulation for those wastes the effort.
- Interactive content is labour-heavy to produce: a widely-used benchmark puts simulation-level development at 490 or more hours per finished hour, far above basic e-learning. Match the investment to the stakes.
- The strongest programmes blend. Teach the concept, practise the task on a realistic clone, then support the rare task in the live system. Each method does what it is best at.

## Introduction

### The question clients keep asking

“Should we use simulation, or stick with classroom training?” It comes up in almost every rollout conversation, and it is the wrong question as phrased. The honest answer is another question: what are you trying to build? A finance clerk who needs to post an invoice correctly under time pressure has a different need from a manager learning why a new approval policy exists, and the two are not served well by the same method.

There is a cost to getting this wrong in both directions. Simulation-level content is expensive to produce, so you cannot default to it for everything. Equally, a one-off classroom session will not build the recall a daily transaction needs, so you cannot default away from it either. The skill L&D leaders are asking for is not a verdict on simulation. It is a way to decide which method fits which need.

### Start with the outcome, not the method

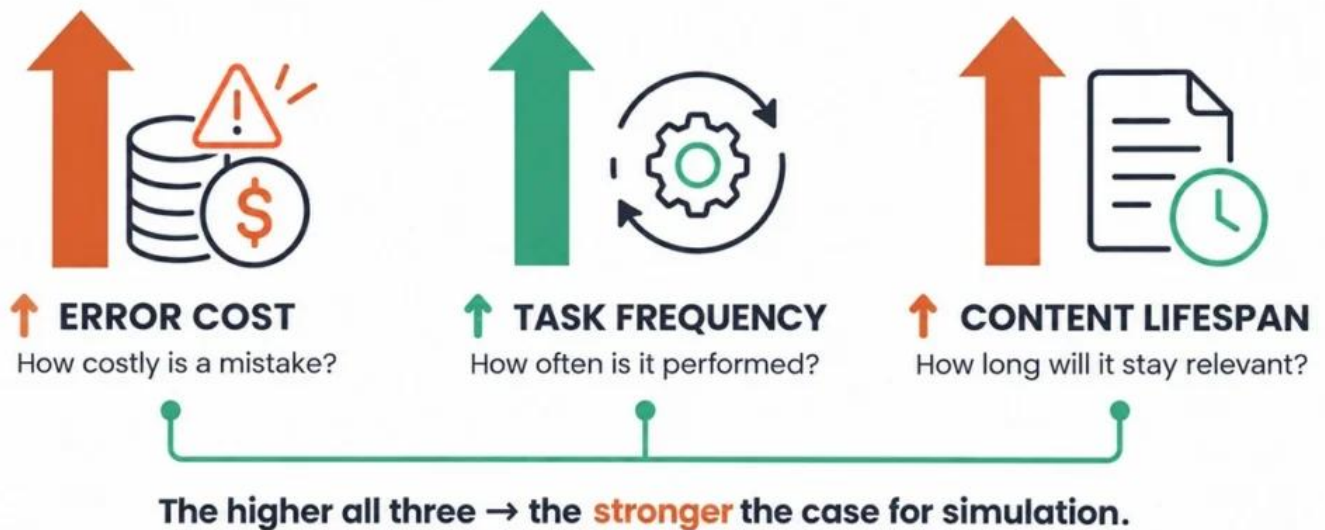
Every training method is good at something and poor at something else, so the first move is to name the outcome and let it choose the method, rather than the other way round.

The old distinction still does the work here. Knowledge is facts, concepts, and the reasoning behind a process. Skill is the ability to carry out a task. Behaviour is how people act on the job. Kirkpatrick’s levels are a useful discipline at this point, because they push you to measure the outcome you want rather than attendance or a completion tick.

[Systems training](#) is mostly a skill problem. The test is whether someone can take a real case through the software, correctly, without help. Skills are built by doing, not by watching, and the evidence for that is unusually solid. Learners who practise retrieving and performing a task

retain far more than those who only review it, and the gap widens with time (Roediger and Karpicke, 2006). A large review of learning techniques rated practice testing, the act of doing and retrieving rather than rereading, one of only two strategies with high utility for durable learning (Dunlosky et al., 2013). That is the case for simulation. But not every training need is a skill need, and that is where judgement starts.

## When Does Simulation Pay Back?



## Where simulation is the right tool

Simulation is the strong choice when the outcome is procedural competence in a particular system, and several conditions push the decision further in its favour.

The task is high-stakes or high-frequency. Medication administration, financial postings, patient identification, order entry: tasks where an error is costly or the action is repeated all day are exactly the ones worth drilling until they are automatic. The first mistake should happen in a simulator, where it costs nothing.

People cannot safely practise on the live system. You cannot turn a ward loose on real patients or a sales team loose on live pipeline. An anonymised, editable clone gives genuine practice with no real consequences, which is what Assima Train is built to provide.

The rollout is large, distributed, and changing. When thousands of people across sites and languages need to reach the same standard, and the system ships updates through the year, simulation scales the practice in a way classroom delivery cannot, and editable clones keep it current with an edit rather than a re-record.

That is simulation's zone: a specific, repeatable, consequential skill that has to be built and kept current for a lot of people. It is not the whole of training.

## Where traditional methods still win

This is the part vendors tend to skip, and it is the part that earns trust. Several needs are served better, or only, by methods other than simulation.

---

Concepts and the “why”. If people need to understand a policy, the rationale behind a redesigned process, or how the pieces of a system fit together, a classroom discussion, a workshop, or a well-made e-learning module does that better than a click-through. Simulation teaches the how. It rarely explains the why.

Behaviour and soft skills. Negotiation, difficult conversations, coaching, leadership judgement: a software simulation cannot build these. Role-play, facilitated discussion, and coaching can. Reaching for systems simulation here is a category error.

Awareness and information. Telling people what is changing, running compliance awareness, or announcing a new process needs a short video, an email, or a light module, not a built simulation. Do not spend build effort to pass on something people only need to be told once.

One-off, low-stakes, or simple tasks. If a task is rare and straightforward, a job aid, a quick-reference card, or an in-app tip often beats a full simulation on cost and is quicker to keep current. Save the heavier method for the tasks that carry real risk.

Systems that are not built or stable yet. You cannot clone what does not exist. Early in a programme, process design and concept training come first, and simulation follows once there is a system worth cloning.

There is a hidden risk in ignoring all this. Aim a simulation at the wrong outcome and it will underperform, and the lesson L&D takes away is that “simulation does not work”, when the truth is that it was pointed at a target it was never built to hit.

## **The cost test: does a simulation earn its place?**

Interactive content is not cheap to build, and pretending otherwise leads to bad decisions. A benchmark the industry has used for years, the Chapman Alliance study, puts development at roughly 49 hours per finished hour for the simplest e-learning, rising to 490 or more for simulation-level content, with the most advanced work higher still. That effort has to be justified by something: high stakes, high volume, or long reuse.

A workable rule of thumb: the more an error costs, the more people perform the task, and the longer the content stays relevant, the more a simulation pays back. A high-risk transaction done daily by thousands justifies it easily. A rare, trivial task for a handful of people usually does not.

Two things shift the maths in simulation’s favour. [No-code authoring](#) lets subject-matter experts build content without scarce specialist developers, and editable clones turn maintenance into an edit rather than a re-record. Together they lower the threshold at which a simulation is worth building. They do not remove the judgement, though. A cheaper simulation is still the wrong tool for a concept or a soft skill.

**See where hands-on practice pays back. Our ROI calculator estimates the time and cost you could recover.**

[Calculate your ROI](#)

## **The answer is usually a blend**

In practice it is rarely either/or, and the mature move is to layer methods so each does the job it is suited to.

---

Teach the concept and the “why” with the lightest effective method, whether that is a short workshop, a discussion, or an e-learning module. Build the skill with hands-on practice on a realistic clone before go-live, using [Assima Train](#). Then support the rare or unfamiliar task in the live system at the point of need, with Assima Assist prompting a user through a step and [Assima In-App Search](#) bringing the right process guide onto the screen they are already working in, so nobody has to memorise everything. [Sodexo](#) runs exactly this blend: the global facilities-management group builds its own content and mixes classroom sessions with self-service e-learning on clones of the applications it is rolling out, so unit managers become self-sufficient quickly and lean on the help desk far less, and it reports around 30 per cent faster onboarding as a result.

The evidence supports blending, with a caveat worth stating plainly. The US Department of Education’s meta-analysis found that blended approaches outperformed purely face-to-face instruction, but the authors were careful about why. Blended designs tended to add practice, time, and interaction, and that, rather than the technology itself, is what moved the results. The same logic applies to simulation. Its value comes from letting people practise, not from being software. Keep that straight and you will use it for the right reasons.

## 1. UNDERSTAND



## 2. PRACTISE



## 3. PERFORM

### A short way to decide

When the question comes up, five checks settle most cases:

1. What outcome do I need: knowledge, skill, or behaviour? Skill points towards simulation. Knowledge and behaviour often point elsewhere.

- 
1. How costly is an error, and how often is the task done? High stakes and high frequency favour drilled practice.
  1. Can people safely practise on the live system? If not, a clone is the way to give real practice.
  1. How many users, how distributed, and how often does the system change? Scale and frequent change favour a maintainable simulation.
  1. Is the system built and stable enough to clone? If not, start with concept training and process design.

If the answers describe a high-stakes, high-volume, system-specific skill that cannot be rehearsed live, simulation is the clear choice. [AkzoNobel](#) fits that description exactly: training more than 100,000 employees and customers on a new application across 80 countries and 13 languages would have meant standing up and maintaining thirteen separate training clients, one per language. A single set of editable clones, localised from one source, did the job instead. If they describe concepts, behaviour, awareness, or a rare and simple task, reach for a lighter method and keep the simulation budget for where it counts.

## Conclusion

The question clients ask, simulation or traditional training, has a better version hiding inside it: what am I trying to build, and which method builds it best? Simulation is a powerful tool used deliberately and a waste used by default. Match the method to the outcome, let stakes and scale decide how much to invest, and blend so each method carries the load it is suited to. Do that, and the debate stops being simulation against tradition and becomes what it should have been all along, a design decision.

**See Assima on your own systems. Book a demo for hands-on practice your team actually retains.**