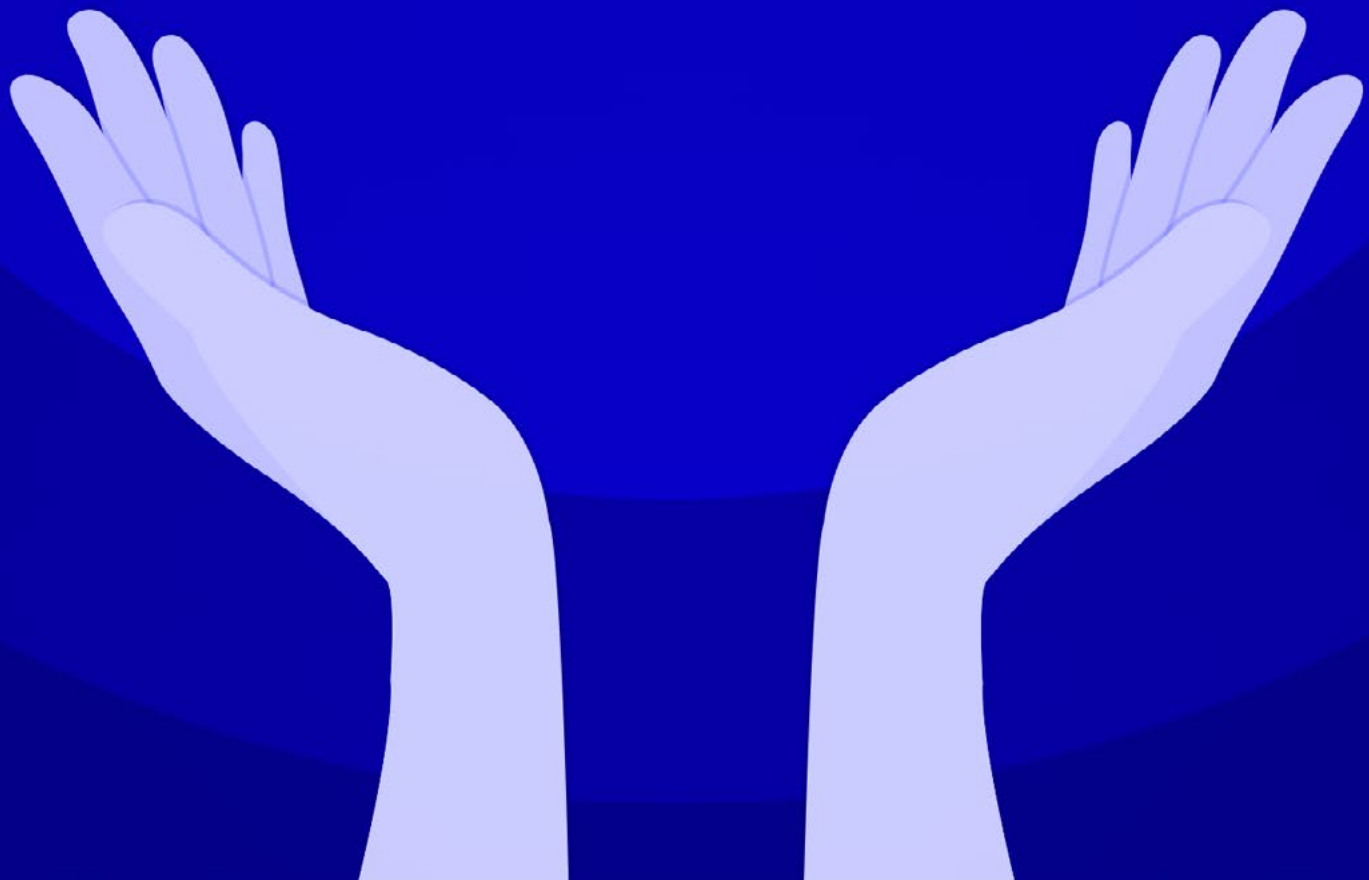
E-BOOK

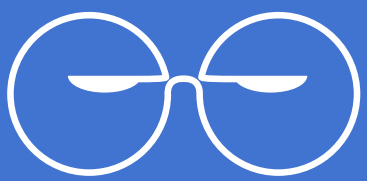
Train Your Teams to Fly

How hands-on learning can break barriers, build better teams, and get results that stick



CONTENTS

Why Learning is Hard According to Science

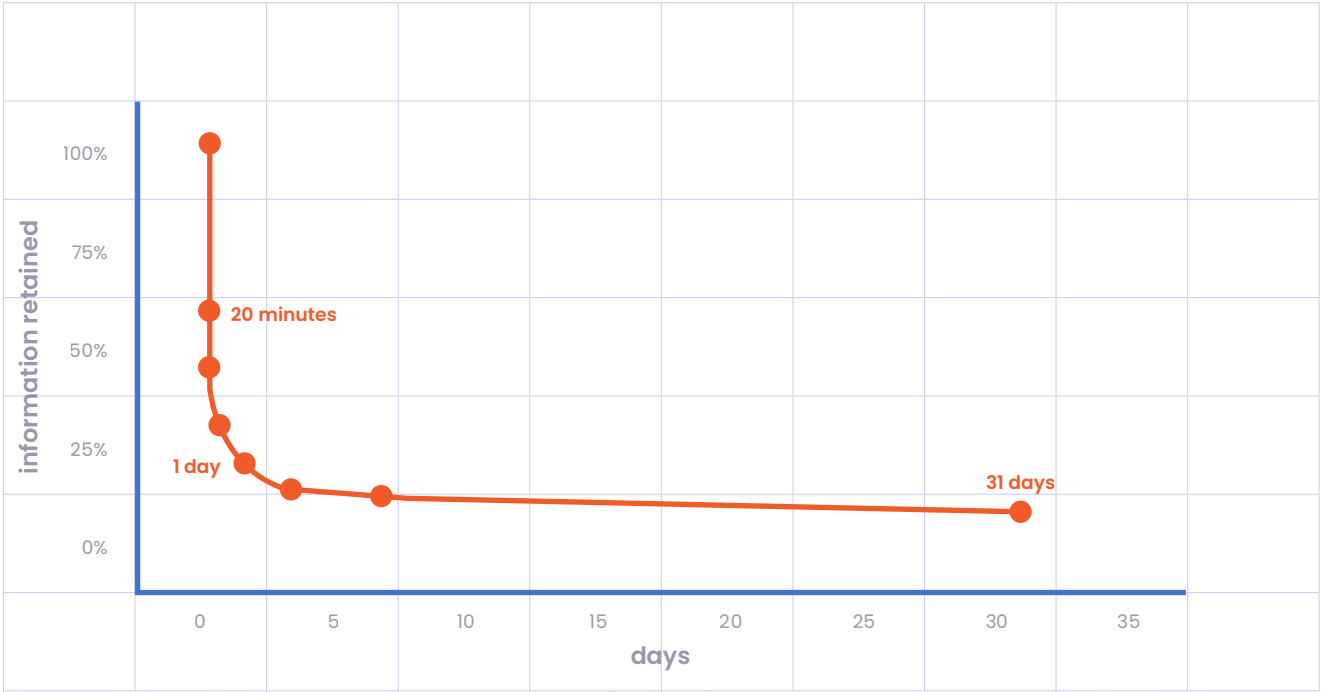


In the late 19th century, the German psychologist Hermann Ebbinghaus became deeply curious about the nature of human memory and how it affects our ability to learn something new. He didn't just want to understand how well the brain could remember learned skills, but was looking to create an empirical model that could predict how effectively the brain could retain that information — a sentiment data scientists today would appreciate.

Ebbinghaus conducted his first experiments on himself. He began by memorising a series of meaningless syllables, for example: DIF, LAJ, LEQ, MUV, WYC, DAL, SEN, KEP, NUD...and so on. Then he tried to recall them at various points of time— immediately after memorising them, then 20 minutes later, then an hour, a day, and finally, a whole month later.

When he documented the results of the experiment, an interesting pattern seemed to form. The amount of information he could retain dropped at an exponential rate, which meant that just 20 minutes after he'd memorised the syllables, he'd already forgotten nearly half of them. By the end of two weeks, he could remember less than 10% of all the syllables.

This decaying exponential graph, known as the Ebbinghaus Curve or the 'Forgetting Curve', models how the brain forgets information over time. It also explains why learning is so hard.



THE EBBINGHAUS FORGETTING CURVE

The thing about learning and memory is that—unlike the common comparisons to computer storage—human brains don't store data perfectly. Our minds are constantly shifting information around, keeping what's most useful, tossing out what isn't, and yet somehow never letting you forget memories from high school that make you wince.

Rather, human memory works a lot more like a muscle: the more you exercise it, the stronger it gets. When you learn something once and never touch it again, you're bound to forget it. But repeatedly reinforcing those memories over time ensures you'll remember it, because you've taken the time to impress it in your mind.

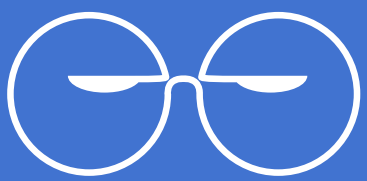
Think about it—when's the last time you crammed last-minute for an exam, and were able to recall what you studied even a few days after?

But learning doesn't have to suck. The reason most people hate learning new skills is that it's hard, or boring, or just unintuitive. Even visual learning, often considered the Holy Grail of education, has its limits.

What you need is something more visceral, more tangible, something that doesn't just ask you to see or hear something, but do.

And that, folks, is what this ebook is all about.

Hands-On Learning: The Secret Sauce You Didn't Know You Needed



Traditional learning—students sitting at desks facing a blackboard, a solitary teacher explaining stuff without much interaction—is a relic of the Industrial Revolution era, when thousands of youths had to be quickly and efficiently prepared for jobs at factories.

But this style of teaching has long since become outdated, largely because 'one-way learning' is so focused on rote memorisation. It might teach you the 'what' and the 'why,' but rarely explains the 'how'. Despite this, many institutions and corporate training programs haven't really evolved past these outmoded techniques.

> **What makes active learning so much more effective than the passive alternative is the fact that it's learner-centric.**

But in recent decades, new research has given us a look at how a more active approach to learning can make a massive difference to how quickly and effectively you learn. One study showed that students who engaged in learning hands-on are 1.5 times more likely to pass a course than those who didn't. It's even been found that learning by doing activates the sensory and motor-related parts of the brain when students were asked to recall what they learned.

BUT WHAT EVEN IS HANDS-ON LEARNING?

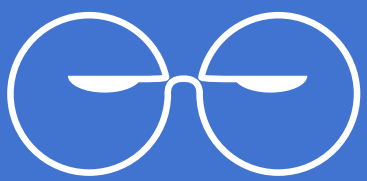
Hands-on or active learning is the process of actively performing the thing or concept you're learning about. If you want to learn how to secure a container image in Kubernetes, the best way to learn about it is to actually take an insecure image and perform every step of the process yourself.

What makes active learning so much more effective than the passive alternative is the fact that it's learner-centric. In passive learning, the instructor and learner take on the roles of master and apprentice, where one teaches and the other absorbs. In active learning, the instructor serves more as a facilitator, guiding the learner to knowledge instead of giving it to them straight away.

Since there's a barrier to progress—you can't finish a lesson without performing the assignment/task—the student necessarily has to understand and implement what they've learned in order to get to the next step. And that makes all the difference to how much you remember.

The longer you spend learning something, the less you'll be affected by the Ebbinghaus Curve, and the more information you retain over time. And the best part? This information is directly applicable in the real world.

Let’s Talk About Why Corporate Training Sucks



If you're reading this ebook, chances are you're working in a corporate environment, juggling ten different responsibilities while still trying to have a life. You might even be a team leader who's in charge of a major project that your team isn't quite ready for, and you're trying to figure out how to train 100+ people in 5 different disciplines.

But there's a serious problem with corporate training programs. You've probably faced it, too. Maybe it's PTSD from school, or maybe it's just that learning can be so boring, but nobody seems to like training in a corporate environment. It's been estimated that only 7-9% of skill acquisition in organisations comes from formal training.

Back in 2019, we'd been brought on to train a major technology solutions company in cloud security, Kubernetes security, and threat modeling. More importantly, though, we needed to help them fix a systemic issue plaguing the company: people were constantly skipping mandatory training programs. Feedback from their teams revealed a host of issues that are prevalent across thousands of companies around the world.

And if your organisation doesn't recognise them right now, they could potentially sink thousands of man-hours and hundreds of thousands of dollars on training programs that don't work.

THE 4 BIGGEST PROBLEMS WITH CORPORATE TRAINING

- ***'Check-the-box' mandatory training***

  This is perhaps the most common: programs that exist purely to meet compliance requirements. These perfunctory efforts at training to tick a box might get the job done on paper, but amount to very little in the real world.

  When the end result is not knowledge but a certification, how can you expect your team members to actually learn anything useful?

- ***The 'one-size-fits-all' approach***

  Some people learn faster than others. Others have shorter attention spans. Still others aren't comfortable with the language your company uses to communicate.

  These are all real problems employees face when they undergo training, and they're often overlooked because corporate training is all about scale and

efficiency. But that simply isn't how learning works, and it's important to give people the opportunity to learn at their own pace.

- ***Learning fatigue***

  A symptom of the compliance training issue, learning fatigue happens when training is seen as an obstacle to get past rather than an opportunity to acquire new skills.

  Tight deadlines, obtuse progress tracking mechanisms, and a saturation of content can make employees dread a training program rather than welcome it. Over time, this can lead to them taking training less and less seriously, which is bad for both morale and productivity.

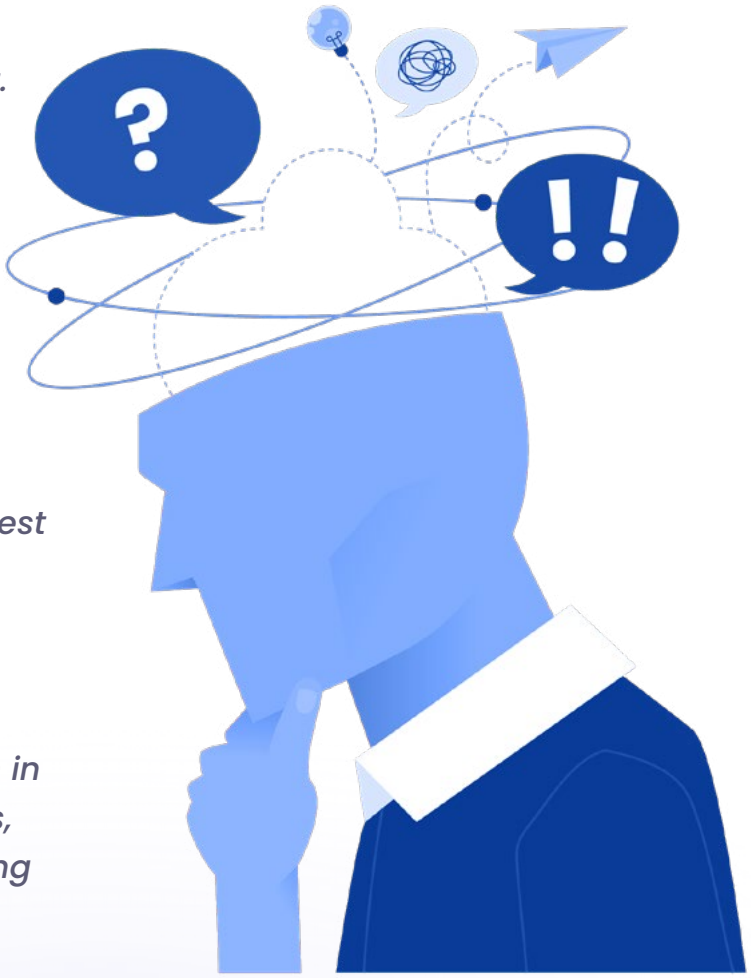
- ***Outdated or irrelevant material***

  If there's one thing that can completely sap someone's interest in a training course, it's bad content. If your team members have to sit through hours of training that's not relevant or immediately applicable to them in their day-to-day responsibilities, you can count on them not taking those programs seriously.

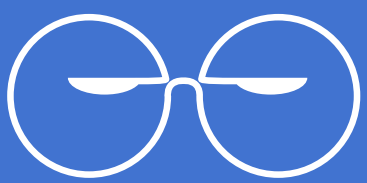
So there you have it. Corporate training has long suffered from problems of scale and an obsession with efficiency that prioritises numbers over the needs of the learners. Bad training programs waste employees' time and the company's money with little to show for it.

But it doesn't need to be this way. Training can be learner-centric and cost-effective, all without spending hundreds of man-hours on methods that don't work.

And it all starts with hands-on learning.



How to Get Corporate Training Right



Let's say your team is building an app on AWS, and they need to train in Application Security so that it's secure at launch. We'll take you step-by-step through the process of finding, implementing, and analysing the effectiveness of an AppSec training program so you know exactly what worked and what didn't.

So, where do you start?

While we're big fans of active learning, it's not as simple as just picking the first program that uses 'hands-on' as a buzzword. To get the most out of your training programs, it helps to break down the entire process into 3 phases:

- *Before training*
- *During training*
- *After training*

This process is recommended by psychologists and training researchers, and has been found to be highly effective in training large teams.

At each phase, there are important steps you need to take to ensure that the training isn't just happening, but helping your team effectively solve their day-to-day problems.

PHASE 1: BEFORE TRAINING

Training Needs Analysis

This is the first phase, and it involves a whole lot of planning. Researchers say the most important step to take at this point is conducting a Training Needs Analysis (TNA). It helps you answer 4 key questions:

1. ***What is the expected outcome of the training?***
2. ***How is the training going to be designed and delivered?***
3. ***How will the training be evaluated?***
4. ***What factors will help or hinder the training?***

Let's go back to the example where your team needs to build a secure AWS application. Answering the first question is straightforward: the training should result in every team member acquiring the skills to securely build and deploy a cloud-native app securely on AWS.

Job Task Analysis

The second question needs to consider a few factors like the various disciplines your team is divided into—developers, DevOps engineers, security engineers, etc. It wouldn't make sense for your developers to take courses on building DevSecOps pipelines, while your DevOps engineers probably don't need lessons on secure

coding practices. Identifying what kind of training each team member needs is what's called a Job -Task Analysis. But it's not enough to simply ask your team, "What training do you want to take?" Research has shown that people are often not able to articulate what training they really need, making a more thorough analysis necessary.

It's crucial for your organisation to find the courses best suited for each team member so they train only in the most relevant, time-sensitive skills to help them do their job.

The third question has a lot to do with the material of the course, and here is where you need to evaluate if the lessons are truly hands-on or not. Do they offer some way to practice or apply the skills you learn in each course? How closely do the lessons and practical exercises relate to real-world use cases and scenarios?

For example, every course on AppSecEngineer comes with hands-on labs so you get to try out what you see in our videos. Whether it's learning to configure AWS services securely or automate DevSecOps, our courses show you how it's done using true-to-life scenarios and models.

Organisational Analysis

Finally, the fourth question. This one's a bit more complicated, and depends a lot on the work culture and environment at your organisation. You'd need to conduct an Organisational Analysis to determine what your strategic priorities are with the training, the resources available, limitations, and fostering an environment conducive to learning.

Another major issue to consider is skill decay, where employees tend to forget learned skills if they don't immediately get to use them. (The Ebbinghaus Curve rears its ugly head again...) While hands-on learning helps in mitigating this, it greatly helps to schedule training within a period of a few weeks to when they'll need to apply those skills for actual work-related tasks.

PHASE 2: DURING TRAINING

Now we're on to the second phase, where the training actually gets executed. If you've followed all the steps in Phase 1, your training program is now officially ready for takeoff.

Learning Climate

If you're in charge of the team undergoing training, it's your responsibility to foster a positive Learning Climate amongst your colleagues. Studies have found that the conduct of supervisors can have a significant impact on the effects of

employee training, both for better and worse.

Positive affirmation, preparation, and guidance have been found to amplify the effectiveness of training, while negative comments or a lack of involvement have resulted in the exact opposite. A team leader's presence is as crucial to the outcome of a training program as the program itself.

In fact, guidance and positive reinforcement from a supervisor can be a huge source of motivation, giving employees new impetus to take training seriously. Don't ever doubt the power of good leadership, folks.

Practice Makes Perfect

The foundation of any good hands-on learning is repetition. The fastest way to learn a new skill is to perform it by hand, and the best way to retain that skill for an extended period is to practice, practice, practice. Whether it's in a simulated environment or the real world, the sharpest skills are the ones you use the most.

To use our example of building an AWS app again, it's important to encourage your team members to engage with lab exercises where they'll learn how to manage secrets in AWS, use different AWS services like IAM, S3, and EC2, and automate security scans with various tools.

In fact, AppSecEngineer even offers labs that simulate the process of attacking and defending insecure AWS apps, giving learners a comprehensive Purple Team approach to cloud security best practices.

Learning From Mistakes

Want to know the best part about using simulated models for training? It's a safe, easy way to learn from your mistakes. If your team made a serious error on a live application accessible to the public, it could have dire consequences.

On the other hands, making a mistake in a simulated lab environment is actually a good thing, because it's a harmless way to learn what not to do.

Training that doesn't use real-world models is like teaching someone to fly a plane with a textbook. If flight simulators can save lives, hands-on learning will save your company.

PHASE 3: AFTER TRAINING

This is the final phase of the training process, but just because the actual training is over doesn't mean it ends here. At this point, your job will be too look back at the last two phases and evaluate what went right and wrong, and prepare to make changes the next time you need to train your team again.

Training Evaluation

Think back to the Training Needs Analysis from Phase 1—specifically, the first question: "What is the expected outcome of the training?" The main objective of your evaluation is to see if the training gave the results you were expecting.

Going back to our example, you need to determine if your team learned the necessary skills to securely build and deploy an app in AWS. This includes team members specialising in other disciplines who took courses on DevSecOps or AppSec. Were there problems your team couldn't solve even after getting trained?

Here are some of the most important questions to consider:

1. ***Did the training teach the skills necessary for the learners to do their jobs better?***
2. ***How thoroughly did the material cover the subject? Were there any gaps in knowledge?***
3. ***Did the hands-on exercises closely represent real-world challenges?***
4. ***What was the learners' experience with the training?***

By speaking to the team members and systematically answering these questions, you'll come to a conclusion that tells you if the training program was worth it or not.

Beating Skill Decay

The last step in the training process is all about ensuring your team doesn't forget the skills they so painstakingly acquired in the last few weeks. According to some studies, trainees that don't get to apply what they learned forget up to 90% of it within one year of being trained. Sound familiar?

That's why it's so important to make sure your team get to utilise their skills in real- world circumstances soon after they've learned them. It's quite possibly the best way to reinforce their training, owing in no small part to the confidence your team members get when they see all their hard work come to fruition in a tangible form.

But what do you do when these opportunities don't happen? If your team can't immediately their newly learned skills, it certainly helps to keep practicing every once in a while. One of the many benefits of hands-on learning is that you can revisit topics you learned months or years prior and brush up on them with little to no friction.

That's why AppSecEngineer offers unlimited practice time on all our hands-on labs. Your AWS Network Security getting kind of rusty? Fire up those labs and pick it up right where you left off. Follow easy, step-by-step instructions so you never get lost.

You'll be learning so fast, you'll forget what the Ebbinghaus Curve ever was.

Hands-on Training is the Difference Your Team Needs

If there's one thing you should take away from this ebook, it's that hands-on learning really works. It's a way to test ideas and experiment without the associated risks, to learn new skills and reinforce familiar ones, and develop experience in new domains without needing to explore untested waters.

But it's also a lot harder to create material for active learning , which explains why you don't see it more often in the corporate space.
That's where AppSecEngineer totally changes the game. We've created a space where your lessons go hand-in-hand with practical models you can play around and experiment with. More importantly, we've made it incredibly accessible to people who are dipping their toes into AppSec for the first time.

Teams can train in 7 different domains of AppSec, including Kubernetes, Cloud, and Threat Modeling, and training can scale from teams of 5 to 200+ with zero friction.

When it comes to skilling your whole organisation in Application Security, nothing comes close to AppSecEngineer.

But don't take our word for it. Give it a go.

Get a Demo

Take the Free Trial