

## ARTICLE REPRINT: **ARE WE HAVING FUN YET?**

---

### Abstract

When **training** professionals talk, there is often a great deal of focus on making training “fun” for learners. When **line managers** talk, they always want to cut out the “fun” and just get to the content.

This article explores the ways learning activities can improve learning and how they can become a waste of time and energy—and even interfere with learning.

---

### Thanks for your interest!

The following pages contain an extended version of an article that was previously published in “Building Capability,” the PRH Consulting quarterly newsletter.

We are interested in your feedback—please contact us at one of the numbers below to let us know what you think about the article. Also, we invite you to browse the other articles and presentations available in the “Library” section of our website.

*Peter R. Hybert*, Founder and Principal Consultant

**PRH Consulting Inc.**  
20 Danada Square West, #102  
Wheaton, IL. 60187  
[www.prhconsulting.com](http://www.prhconsulting.com)  
(630) 682-1649

## Are We Having Fun Yet?

By Peter R. Hybert

### *I Wonder...*

I have never been entirely comfortable with “having fun” used as a measure of an effective training program. Still, in many cases training evaluation is nothing more than a customer satisfaction rating of how much fun the trainees had.

On the other hand, there is a great deal of corporate training that is really nothing more than conveying uninteresting information using the means of someone talking at “learners.” Certainly nobody is having fun in that situation and they probably aren’t learning much either.

There is nothing magic about creating effective training. First you need something that people don’t already know (or know how to do) that would benefit them (or the company) for them to learn. The learners then need to know what they are supposed to learn and they need a good reason for trying to learn it. Then they will pay attention and invest the effort needed to learn new information and/or skills. Learning requires the learners to actually *use* the new stuff, though—if you don’t try the new thing, you can’t learn it. (And if you try it but don’t get any feedback or coaching, you will learn only inefficiently if at all.)

It can be very rewarding to try a new skill, see yourself improve, and then imagine how it will make your job go better, easier, faster, etc. In some cases it could be considered fun. But most of the time, the term “the learning curve” is more like the necessary “pain before the gain.” The payoff is later. Real learning is usually more like work than fun. You can have some fun at work but work is not a game.

Why do so many people think training should be fun? Lot’s of reasons. People like to have fun so trainers do get positive feedback for entertainment. Fun is a way to keep people paying attention. It can be a way to sneak in some application of the new learning. But if it is just activity for the sake of activity, it is probably missing the point.

### *Why Do We Have Training in the First Place?*

Ultimately, training is an expense...well, really an investment. From the standpoint of the business it takes time, dollars, focus, etc. that could be used for other things (for example, selling, making, or repairing products—things the customers will pay for). That means that a business must evaluate the investment in light of the return compared to other investments. So, if we can get the same learning to happen with less time (i.e., less investment) that is better just like it may make sense to take *more* time if it leads to better results.

You can appear to cut training costs but, if the learning doesn’t happen, you have actually increased your costs—all you did was convert your training investment to waste. For awhile many companies acted as if computer-based training was the best approach for everything simply because it seemed to deliver the same learning for much less cost per person. (I think this thinking has

recently been at least somewhat corrected.) Computer-based training doesn't automatically guarantee either cost savings or improved performance. For example:

- Computer-based training is not really that cheap. Even simply converting your existing training to computer-based delivery takes time and effort.
- Lots of computer-based training is not effective. Maybe it isn't the right content to deliver via computer, maybe the design/development is poorly done, maybe the company doesn't have the infrastructure needed to support it. Whatever the reason, much of this investment seems to yield less return than expected in terms of improved performance.
- Training that is effective in a controlled pilot isn't necessarily effective when released. Often there is all kinds of set up and coaching available during a pilot. But on the job, people are often given a login and turned loose and, no surprise, the effectiveness plummets. Also, because the training can be done on a computer, employees are often expected to complete it on their own time which seems like a cost savings but, besides being a little uncool, can result in reduced focus and effectiveness.
- Activities on the computer are difficult to build. It can be easy to do simple activities (e.g., matching, labeling diagrams, multiple-choice, etc.) that are OK for basic recall checks. But if you want to practice a more complex performance, you need a complex (and expensive) simulation. A bunch of good test scores on simple recall activities doesn't mean people can actually do a job task better.

The bottom line is that training effectiveness should be evaluated based on the capabilities the learners leave with and, assuming they are comparable, then the cost of imparting the capabilities should be considered. Figuring out the cost is easy though—even if you include the cost of all employees time, amortization of facilities, and out-of-pocket cost, it is really a fairly straightforward exercise for an accountant. In fact, even adding opportunity cost (possible revenue lost while employees are off the job in training) or “softer” benefits, such as employee satisfaction can be factored into the equation fairly simply. But getting an objective measure of capabilities at the conclusion of training is not easy.<sup>1</sup>

Without digressing into another topic altogether (i.e., how to quantify capability) the problem can be broken into

- How to measure whether a capability has been learned.
- How to assess how completely a capability has been learned (e.g., the number, type, and degree of difficulty of the different situations in which the capability was demonstrated).
- How to account for variations between individuals (ability to learn, experience).

---

<sup>1</sup> It can be made to look easy by simplifying the problem, for example, by creating a multiple choice test and then comparing scores at the end. But even then there are distortions built into the measurement. And the real goal of enabling people to do something on the job is not often measured well by a multiple choice test.

- How to compare two training programs that may deliver different learning results—for instance, if program A teaches 80% of the capability but program B teaches 70% of the capability but includes some additional (“nice to know”) content and costs 60% of A.

Ultimately, it comes down to a qualitative decision based on a wide range of factors (including administrative issues such as ease of delivery, availability, etc.)<sup>2</sup>.

### *A Closer Look at Activities*

So what are we to do? We advocate approaching training like an engineer approaches a design problem. Define the results that you want, create a process for getting those results, then execute that process, measuring results as you go. (If you are thinking this sounds like Instructional Design, it is!)

Since we are focusing on activities here, that is, where the learner is performing something (e.g., trying out or practicing a new skill—demonstrations where the instructor or computer is showing something are another subject) we will skip a description of the overall instructional design process other than to say that all training design decisions need to start with the job performance. So the activities need to be based on the job performance<sup>3</sup>. Going back to an earlier paragraph, we asserted that the learner has to try out the new learning (i.e., knowledge, skill, task) in order to really learn it. That is where the activities should come in. They should be opportunities for the learner to try out some or all of what they are trying to learn and, where appropriate, get feedback and coaching so they can improve.

Below is a list of generic types of learning activities often used in training programs. There are multiple variations of each but these types are based on the instructional purpose. (Additional activities, such as ice breakers, may be included in the training but their purpose is other than learning.)

---

<sup>2</sup> Of course, studies can be configured to address the above but to arrive at an answer that is truly objective would require an effort far beyond that which most companies are willing to invest.

<sup>3</sup> Even activities that teach enabling knowledge or skills need to be designed as “stepping stones” to the end performance.

| Activity Type            | Description  | For Example...   |
|--------------------------|--|--|
| Concept Illustration     | A brief activity to demonstrate a concept or impress on the learners the importance of an upcoming point.  | <ul style="list-style-type: none"> <li>• "Telephone" game to illustrate how communication can get garbled</li> <li>• Have participants solve a brainteaser to illustrate "out of the box" thinking</li> </ul>  |
| Recall check             | Providing the learner with a stimulus and asking them to provide a response  | <ul style="list-style-type: none"> <li>• Label a diagram</li> <li>• Jeopardy-style game</li> <li>• Classical quiz questions (e.g., matching, true-false, etc.)</li> </ul>  |
| Skill practice           | Providing the learner with a situation and asking them to try out the skill they just learned  | <ul style="list-style-type: none"> <li>• Giving typing students a letter to type</li> <li>• Giving a database design student the assignment to create a series of reports</li> <li>• Word problems in a math class</li> <li>• Role play to practice asking "closed probe" questions</li> </ul> |
| Application (Case)       | Providing the learner with a situation and asking them to describe how they would approach it—this requires a combination of recall and strategy | <ul style="list-style-type: none"> <li>• Case studies in an MBA class</li> <li>• Scenario in which they have to create a strategy</li> </ul>   |
| Application (Simulation) | Providing the learner with a situation in an environment that emulates the real job  | <ul style="list-style-type: none"> <li>• Breaking a database design class into teams to first role play a client meeting to determine requirements and then design and present a prototype database by a deadline</li> </ul>   |

It is interesting to note that all the activities above, except for concept illustration activities, can be used as pre-tests, examples or demonstrations, practice exercises, or tests. We suggest developing them to be semi-standalone so they can be re-used as needed in other settings.

### *Design Guidelines*

Of course, there is no one way to design training. Lots of approaches work (and sometimes, even a poor approach will work if it is implemented and reinforced well.) In general, we tend to use simulations as culminating activities in which a range of prior learning is integrated. We will teach the underlying knowledge and skills using skill practices where needed to build the components of capabilities. Then we would practice putting it all together in a simulation.

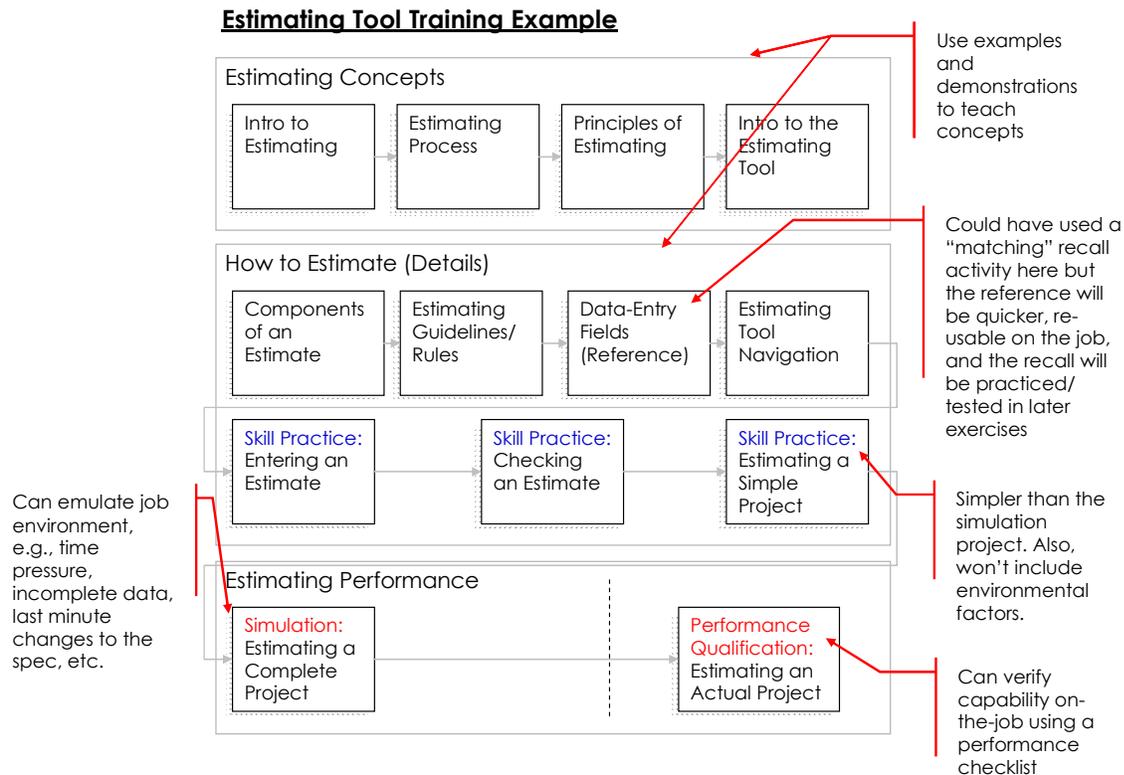
A key design principle is targeting your use of activities to where they will be of most benefit. Clearly, activities take more time than lecture (or reading). But where they are necessary, they ensure

learning really happens. And simulations, if well-designed, are key to transfer—to ensuring the learning is actually applied on the job.

### An Example Instructional Process

There are always judgement calls when designing anything and training is really no exception. We don't agree with using hard and fast rules (such as "every objective needs to have an exercise and test question") because a smart designer may come up with a way to test a number of objectives in a single exercise. (Simulations usually test multiple objectives.) Or one objective may be subordinate to another meaning, if you can meet the second objective you can meet the first—so you may only need to test the second.

But, the disclaimer notwithstanding, here is a (simplified) example of how you might approach teaching someone how to do an estimate using the companies estimating tool. (By the way, we use an object-based design/development method to allow us to restructure the flow as needed.)



### Conclusion

There is a great deal more that could be said about designing and developing effective activities. Just because you decide to use a simulation doesn't mean it will work well—there is some art to it. And fun still matters, especially with simulations—if nothing else, learners have an expectation that they should be engaged during the learning process

With simulations, we suggest adding the fun by building in challenges from the real world. That means things like where all the data isn't available. Or where there is some conflict or disagreement. With a recall check, you may have to force the fun, for example, by turning it into a competitive game. Concept illustration activities are usually fun to begin with...but you have to be careful that they don't really end up being much ado about nothing. If you use a lot of class time to make an obvious point you may be taking time from a more valuable activity.

Ultimately, if you start from performance requirements, if you use activities sparingly and for strategic purposes, and if you have a logical sequence building to capability, you will be able to make a strong case for keeping the activities you need for learning to happen from being cut from the training program as a time saver.

But sometimes, even when you do all your homework the business will need to put the squeeze on training time. Since the first thing on the chopping block will be the activities, because of the time they consume, below is a set of responses to typical objections. If this were a class, maybe we would have you role play them!

| Argument for Cutting   | Response  | Comments  |
|--|---|---|
| They will get the application on the job.  | Actually, they will do the learning (i.e., "trial and error") on the first few attempts. There is some business risk in that.   | Sometimes, you can afford the risk—there may be a coach or supervisor checking their work. But often the risk is underestimated.  |
| These are "sharp people," if they can't figure it out, they shouldn't have this job.   | There is a difference between "getting it" and being able to do it. The activity will give them a chance to build skill and fluency.  | Your client has forgotten how much they have learned to get to this point and aren't aware of the number of things they think about subconsciously during the performance.  |
| Instead of everyone in the class trying it, what about just picking a couple of people and then have everyone discuss how it went. | See previous.   | It is easier to watch someone doing something and assume you can probably do it than to actually do it!   |
| Cut down the number of examples/ attempts—just give them the idea of what it is like.  | We have designed this program for the minimum number of activities—it would be better to eliminate lower priority content (capabilities) than do a partial job of learning more capabilities. | The number of situations or practice applications is a key design decision and does largely depend upon judgement. You need to be confident that you have considered this question carefully during the design process. |