????? ????? (BUILT-IN FAILURE)

This pattern was written by Eugene Wallingford, based on ideas from Kent Beck's THREE BEARS.

Learning comes from experience, and much useful experience comes from failure. But a learner who lacks confidence will fear failure, and this fear impedes or even prevents learning.

Confident learners use failure and frustration as investments whose payoff comes in future success. They know that a "wrong answer" offers the opportunity to discover a misunderstanding and to arrive at a better understanding of the topic. These knowledge "repairs" will lead to improved performance over time.

However, many learners do not start out as confident. Traditional schooling typically discourages or punishes failure through grading schemes and recognition of academic achievement. Employees may fear that failure will be seen as a sign of inability by their employers and lead to fewer workplace rewards. As a result, the confidence to fail is rare and hard to develop.

φφφ

Therefore, remove the fear of failure as a barrier to learning by making failure a part of the goal.

Create an environment in which failure is an expected and desired outcome of the learning activity. Build an activity that requires learners to reflect on both the "correct" and "incorrect" answers as a way to better understand the topic. Make sure that all learners will encounter the negative outcome and that no one will be stigmatized by not reaching the right answer.

φφφ

THREE BEARS builds failure into the process of learning to recognize a point along a continuum. MISSION IMPOSSIBLE poses a problem that cannot be solved with a naive understanding of the topic, encouraging students to explore the topic more deeply. Mistake [JBx] asks students directly to make and deal with errors.

THREE BEARS

This pattern was written by Kent Beck and revised by Eugene Wallingford.

Some problems, inherently sap a learner's confidence. Many problems challenge the learner to find a solution positioned along some continuum. Solving these problems effectively requires that the learner discover a point or a range along the continuum that satisfies the demands of the problem. But finding such a solution requires that the learner have experience with many problems, balancing the demands of each in a particular solution. Until they have sufficient experience, they are likely to be unsuccessful finding the right balance.

How often should a developer refactor a program? How strictly should a musician follow the rhythm of the piece? How often should a point guard shoot the basketball?

The process of learning to find such balances creates substantial barriers to the learner gaining experience. The learner will likely be unsuccessful on the first few attempts, unlike many other learning activities. Even if the learner stumbles into the right balance, chances are that the learner will not recognize that the balance has been struck, or why.

φφφ

Therefore, ask the learner to create solutions that lie at both extremes, as well as at some balance point. The extreme answers will certainly be "wrong" for the given problem, but they give the learner permission to explore the boundaries of the continuum.

First, define the continuum for the learner. The simplest approach is to explain the *reductio ad absurdum* at both extremes.

Second, conduct an experiment that gives the learner a chance to locate the balance for problems whose solutions lie in three different places: at one end of the continuum, somewhere in the middle, and at the other end of the continuum.

Third, conduct a review that gives the learner an opportunity to reflect on the experiment.

The reductio ad absurdum strategy usually gives the learner enough background to begin learning the continuum. You might also pose a set of questions that will be asked of the resulting balance. For example, in reviewing the frequency of refactoring, you could ask "Was the team able to get into a good flow while programming?", "Was there sufficient time for testing?", and "Did the team deliver its product?"

Your experiment should ensure that the learner experiences all three options close enough in time to accurately compare them. Scope the topic to something that can be accomplished in less than an hour, if possible.

Reviewing the experiment is critical if the learner is to understand how well the solutions balance the problem's demands. A useful technique is to have the learner briefly present the three solutions to other learners, and then have the rest of the group guess which was which. This can help learners who have not yet learned the true boundaries of the continuum.

Even still, some learners have difficulty getting past the fact that they "have to do it right" eventually.

Some topics are more complex. You may find that reducing a problem to a single continuum oversimplifies the topic so much that the learner arrives at a simplistic understanding. In such cases, you will want to follow up this experience with others that address the problem's other facets.

φφφ

Walt Disney once used this technique when he found himself dissatisfied with the features of his staff's animations. Finally, in frustration, he told the animators to exaggerate the movement of all their characters. The result was just what Disney was looking for. [Thomas 1981]

THREE BEARS has been used to teach requirements engineering. The instructor asks the learners to write stories that will define the system: one too large in scope to be useful, one too small, and one just right.

Similarly, THREE BEARS can help learners to explore the ethical continuum that faces computing professionals. Groups of three are asked to write stories about obviously ethical applications of computing technology, obviously unethical applications, and applications that are still unsettled. Later, the groups share their stories with each other and try to place the stories on the continuum. Interesting discussions usually follow as the groups disagree with one another about the relative placement of their stories.

Many experienced learners routinely use THREE BEARS in their own learning all the time. For example, a Smalltalk programmer might learn the constraint-ish ValueModel framework by deliberately writing systems that use it too much. Some programmers learn object-oriented programming by writing some programs in which every variable is an instance of a different class and other programs that use too few classes and objects.

MISSION IMPOSSIBLE

This pattern was written by Alan O'Callaghan and revised by Eugene Wallingford.

Many teaching situations are limited in duration. The instructor can choose to omit important concepts in order to fit the available time, but then the learners will not have been exposed to the full subject. Alternatively, the instructor can choose to generalize the material to the point that the whole subject can be covered in the available time. However, such generalizations can oversimplify a rich, subtle topic to the point that the learners think they have mastered it, even though they do not yet have sufficient experience with its details.

Any sufficiently complex topic can be understood at many levels of abstraction. When a generalization is supported by understanding at deeper levels, then abstraction can be a powerful tool. But often new learners arrive at an abstraction not via generalization from a deeper understanding but from a simplification of something they do not yet understand. Such simplistic truths are dangerous, because they lead learners to construct simplistic solutions that do not really solve problems. Worse, the learners' lack of experience prevents them from recognizing the shortcomings in their thinking.

φφφ

Therefore, present the learner with a problem that seems straightforward to solve but whose complete solution requires a much deeper understanding than the basic concepts afford.

Choose a problem that at first glance suggests a solution based directly on the general concepts that the learners have encountered. However, a complete solution to the problem should require careful consideration of a number of issues. Indeed, make the development of a complete solution not normally be possible due to insufficient time to study the full range of issue, insufficient information available to the learner, or the lack of any solution at all, despite first impressions.

Follow up the exercise with a brief summary that explains why the problem was "impossible".

The contrast between the learner's initial reaction ("This is easy!") and the result of some study ("This is a more difficult problem than first we thought!") is crucial to the success of this pattern. It creates in the learner a recognition that the subject is more subtle than originally thought. The instructor's explanation at the end should make sure that the learner understands both the impossibility of the problem and the role played by their still naive understanding in not seeing it.

Use this technique just after the learner has conquered a logical unit of material. It can be used to form a link between the learning of basic concepts and the more advanced topics needed to master the "impossibility" of the problem.

You should be able to present the problem in a short form, and yet it should be complete enough that the learner has sufficient information to begin work. The learner should be able to appreciate the unforeseen subtlety of the problem within about 45 minutes, or the learner will begin to lose interest in the problem.

MISSION IMPOSSIBLE makes learners suspicious about their understanding of basic concepts so that they continually question those concepts and improve their understanding of them. Learners occasionally need to be "shocked" into deeper thinking about what they are doing in order to appreciate subtleties. This becomes even more important when such ideas as "objects model the real world" can be understood in a naive way that disarms the learner in the face of real problems.

Misused, or overused, the pattern can destroy a learner's confidence in what she is learning. Some learning contexts create unstated expectations that the student that they will be "spoon-fed" instruction. Many university students come from schools in which rote learning is the norm. In industry, new ideas are often viewed only as tools or as programming techniques and so require "instruction", not "education". MISSION IMPOSSIBLE requires initiative and risk-taking on behalf of the learner, and therefore may not be appropriate in such contexts.

This pattern follows General Concepts First [Seminars] and is a form of Repeat Topics [Seminars] that aims for a deeper understanding of the repeated topics.

φφφ

In historic tradition, Zen masters pose questions such as "What is the sound of one hand clapping?" in order to encourage their students to lose their worldly inhibitions and achieve enlightenment.

MISSION IMPOSSIBLE has been used in teaching of object-oriented concepts both to university students and to software professionals. Often, the basic truths about objects that distinguish them from structured methods are expressed in a way that leads the learner to underestimate the intellectual effort needed to master object-oriented concepts. A common example is Meyer's "Objects are there for the picking" in response to the question, "Where do I find the objects?") Such a situation creates a perfect opportunity to apply this pattern.

For example, a one-hour tutorial [O'Callaghan 199x] uses MISSION IMPOSSIBLE to help learners realize that even relatively simple programs becomes tortuous when they apply a naive notion of object in both analysis and design. This tutorial uses a simple scenario from [Cook 1994] to help the learners see the need for transformation from the objects in the analysis model to those in the design model.

Bibliography

[JBx] Bergin, Joe, “Mistake”, a pattern in *XXXXX*, www.pace.edu/xxxxx

[Cook 1994] Cook, S., and Daniels, J., *Designing Object Systems*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[O’Callaghan 199x] O’Callaghan, Alan, “title”, OOPSLA Educators Symposium, 199x.

[Thomas 1981] Thomas, Frank, and Johnston, Ollie, *Disney Animation: The Illusion of Life*, Abbeville Press, New York, 1981.

A Note on the Names

THREE BEARS derives from a German fairy tale in which a little girl encounters a number of situations in whic her three choices “too hot, too cold, just right”, “too hard, too soft, just right”, and so on.

MISSION IMPOSSIBLE was originally called KOBAYASHI MARU, based on a story line in the second Star Trek movie. The current name comes from a television series and movie of the same name. The metaphor in Mission Impossible is slightly less accurate to the essence of this pattern than the one to Kobayashi Maru, but it has a wider appeal and so is preferred.