

Continuous Improvement Processes: Why They do not Work and How to Fix Them

Continuous improvement processes are all around us during the engineering design process. Indeed, any meeting with a focus on review and improvement of a design may be viewed as part of a continuous improvement process. Yet many of us often look at the changes that result and wonder if they really are for the good. We can certainly see continuous degradation processes working in many areas of our lives—television and radio, the evolution of web sites, the tax laws—and we could all add to the list. The question is, are we part of the solution or part of the problem?

To get to the root of the issue, we need to understand the mathematics of a continuous improvement process. Let us say a group of three engineers are working on a product design, and decide that it might be good to make some changes. The three are quite happy with the design as it stands, but feel it could be better. So they go over the product, and a number of changes are suggested. Then they go through the proposed changes one by one, and agree that many should be made. With each proposed change, the group agrees by some mechanism, such as majority vote, that the change should either be made or not. Clearly, if the design starts off good, and a series of changes are made, each of which improves the product, it should be better at the end of the process. But, not so fast. How many of you have gone through just such a process only to wind up with a design that you think is quite a bit worse than it was at the start? I surely have. It might surprise you to find that it could well be the case that everyone thinks the design is worse for the changes, even while the group agreed that each individual change improved it. How can this happen?

Problems arise because there are competing issues that the designers are trying to cope with. For example, these could include issues of reliability, manufacturing, and customer appeal. For the sake of simplicity here, let us assume that there are no technical issues. The designers are in complete agreement on the technical performance of the product. Now let us take a series of three changes, A , B , and C , on the status quo, S , which may address the competing issues. So the possible end states can be designated S (status quo with no changes), S_A (S modified by change A), S_B , S_{AB} , S_C , S_{AC} , S_{BC} , and S_{ABC} , depending upon which changes the group adopts. Let us say that the preference orderings on these changes by three designers are the following:

Jan: $S_{AB} > S_A > S > S_B > S_C > S_{AC} > S_{BC} > S_{ABC}$

Pat: $S_A > S > S_B > S_C > S_{BC} > S_{AC} > S_{ABC} > S_{AB}$

Michael: $S_B > S > S_C > S_{CB} > S_{AC} > S_{ABC} > S_{AB} > S_A$

Where, for example, $S_A > S$ would be read S_A is preferred to S , which is to say that the designer feels that product S_A is better than product S . From these preferences, it is easy to see that all three designers have a rather strong preference for S over S_{ABC} . Now let us see how they would choose by simple majority vote. By a vote of 2:1, the group would adopt change A over the status quo. Then they would consider adding change B . Change B (from state S_A to state S_{AB}) is also adopted by a vote of 2:1. And finally, change C , leading to the final state S_{ABC} , is adopted over S_{AB} by a vote of 2:1. So the group moves from S to S_{ABC} even though everyone in

the group would agree that the product is worse for these changes. What is more insidious is that everyone in the group will walk away from the process thinking that it was fair and honest and that, somehow, the product must actually be better. And, of course, the process will not stop with only three changes. It will go on through changes D , E , F , and so on, in what we could appropriately call a continuous degradation process.

Some people will be quick to argue that I have done something wrong here. If only I used a different selection process, a different form of voting, whatever. But I can assure you that I have not made a mistake, nor would a different selection process fix this mess. There is a proof (Arrow's Impossibility Theorem) that it is impossible to find a selection process that will prevent these problems from arising. Some might argue that I cooked the example by picking a rare pathological case. But, once you understand how selection processes work, you will see that it is harder to cook the preferences to get a good outcome than a bad one, such as this example. Further, as more and more "improvements" are offered, or as more people get involved in the process, the more likely it is that the product will get worse for the changes. And the likelihood that a continuous improvement process will morph into a continuous degradation process gets higher with every change. If you question this, I refer you to the extensive literature on the subject.¹

The implications of these mathematics are profound. Our continuous improvement processes are often making things worse rather than better. I give you the following challenge: step back and look at the continuous improvement processes in which you have been involved, not just for product design, and ask yourself if the current state is really better than the starting state. As you take this moment of introspection, be sure to recognize the possibility that the process is likely to have been a degradation process. Can we fix this?

The first step toward fixing any problem is recognizing that there is a problem. How could it be that a process that has received so much attention by so many people for so long could be so wrong? Maybe you just do not believe me. If not, solutions I would offer are superfluous. Take time to reflect on this. Nothing is as simple as it might seem. Processes developed in the absence of a sound theoretical base, and I can assure you that there is no sound base underpinning continuous improvement processes, just are not likely to be valid. Nature provides us with too many things that can go wrong.

What we know from game theory, which is the mathematics of group behavior, is that group selection processes are unstable in the sense that the outcome of a group process is as much dependent (perhaps even more so) upon the rules of the interaction between the members of the group as it is upon the preferences or desires of the individuals who comprise the group. Just how unstable can these processes be?

Suppose, in our example, just as Pat was about to suggest

¹For example, see Saari and Sieberg, 2004, "Are Pair Wise Comparisons Reliable?," *Res. Eng. Design*, **15**, pp. 62–71.

change A , she dropped her pencil and, while bending down to pick it up, Michael suggests change B . Of course, the proposal to make change B will be defeated as S is preferred 2:1. Now Pat, pencil in hand, suggests change A , which is adopted, leading to state S_A . Next, change C is proposed. But state S_A is preferred to state S_{AC} , so change C is defeated and the final state is S_A . S_A results rather than S_{ABC} simply because Pat dropped her pencil. This is indeed the sort of instability that occurs in every such meeting. The results are changed by a sneeze, a cough, a pencil drop. And the changes are big. It certainly scares me to think that “improvements” hinge on such trivial events as a pencil drop, or just by who speaks first.

So how do we fix it? We have to address the two fundamental problems with continuous improvement processes: (1) they are most often group processes, which have inherent issues, and (2) they are continuous, that is, they use sequential decision making, which also have inherent problems. The more we can move away from group processes and continuous processes, the better off we are.

Let us look at the continuous aspect of continuous improvement first. A quick glance back at our example shows that we got into a lot of trouble by considering the changes sequentially. If we had merely listed the eight possible end states and chosen from them, things would have been quite different. Different, yes, but right, no. Nonetheless, you should do your best to minimize sequential decision making. But there are problems, and we will come back to them shortly.

A big problem is that of choosing among the end states. No method of selection can be guaranteed to be free of trouble. But some methods are better than others. And the best method, that is, the method that allows the least offensive outcomes, is the Borda method. In the Borda method, each person in the group first rank orders all possible outcomes and then assigns a score to each, beginning with 0 for the least preferred outcome, 1 for the next least preferred, then 2, 3, and so on. These scores are added for all persons in the group to get the group score. If we had done that for our design example, S would have a score of 17, S_A a score of 13, and S_{ABC} a score of 3, and we would have concluded that we should leave the product alone. The Borda method does not guarantee selection of the “best” alternative, but it has the rather nice property that it will not select the worst alternative (as we see from the example, other methods can).

Of course, rank ordering the alternatives gets to be a problem as the number of alternatives grows. Note that, for three suggested changes to the product there were 8 (that is 2^3) end states. For only 10 changes ($2^{10}=1,024$), it becomes pretty much impossible to rank order the end states. The only practical way around this is to find an index (a real scalar) such that all the end states can be mapped onto the real number line, preferably by computer, in such a way that they precisely reflect the preferences of each of the persons in the group. And still, we are left with the problem of aggregating the resulting scores, and the vagaries associated with such aggregations. In the end, it may help to appoint a “dictator,” namely, a person in the group who does all the decision making

on which changes to adopt. It is only in this way that the decision making relative to such changes can be made rational, and only in this way that the accumulated changes can be guaranteed of actually making things better—albeit (necessarily) only in the eyes of the dictator. But at least they will be better in the view of that person.

Unfortunately, continuous improvement processes are both continuous and group processes, and there is no way to get around this or the problems that come with such processes. But I can suggest a procedure that can keep you from doing utterly foolish things. First, set a baseline, such as state S in our example above. Second, make a change only if the group, by whatever selection method chosen, determines that the change results in an improvement over both the current state, S_i , and the baseline state. This way, no change will be accepted that results in a degradation over the baseline state. Third, whenever, but only whenever every member of the group feels that a new state, S_j , is a better state than the baseline state, S , update the baseline state to state S_j , and use the updated baseline state in successive tests. This double test on every change does not assure continuous improvement. Indeed, the group may well make a series of changes such that they would agree that state S_{i+2} or S_{i+3} is worse than state S_i , but at least it would not be worse than state S (note that the change to state S_{i+1} is made because it is deemed an improvement over S_i).

The scary thing is that we have only two choices—to leave a product alone or to change it. If we want it better, we have to change it. Yet, if we make more than one change, we risk actually making the product worse. And we see that mathematics is not so kind as to allow our simple product improvement processes to work. Nor should you try to fix such processes by adding complexity to the decision process, namely adding nuances like fancy or multilayered voting methods, colored hats, reflexive processes, toss-the-ball talking, and so on. Any attempt to fix this through complexity merely adds opportunity for things to go south faster. In the end, you want to keep your processes as simple as possible, with as few steps as possible, and with as few people involved as possible. If you want to have a continuous improvement process that has any degree of reliability, you should follow these rules:

1. Set a baseline state against which you will compare changes.
2. Group changes into bundles as much as possible.
3. Use the Borda method to select changes.
4. Use a double test—elect to make a change only if it results in a better state than both the current state and the baseline state.
5. Update the baseline state upon unanimous consent of the group.

So I have not solved all your problems. But at least now you know about one that you probably did not think you had, and you have some tools to stay out of deep trouble.

George A. Hazelrigg