



Learning to Learn: A New Look at Product Development

By Nick Zeniuk

At Ford Motor Company, we know how to design cars. We have the engineering, the technology, the Computer-Aided Engineering (CAE) tools. But we haven't been able to adapt the human element to produce the kind of behavior that will enable us to create the superb, special type of product that we're looking for.

We've been benchmarking the Japanese, particularly Toyota. Their capabilities are tremendous; and yet, on a technological basis, there's no difference. Our engineers are as good if not better. The difference is that they've developed a different process of communication and behavior—how they think, vision, and interconnect. Once you get the behavior right, you can take the engineering tools and apply them more effectively.

At Ford, the Lincoln Continental team has been trying to do something dramatically different with product development. The program consists of a cross-functional team of approximately 200-300 engineers, planners, manufacturing and finance people, etc. We're talking about a major program—the responsibilities are very heavy, and the expectations are very high. A core team of six to eight people decided to apply systems thinking and the discipline of mental models to help us think differently about problem articulation—about how we create our own problems and how we can resolve them. So we began to have meetings with the MIT Organizational Learning Center to put together a project. Daniel Kim from the Organizational Learning Center became our facilitator, teacher, and mentor, and with a cross functional group of managers on our team, we began our journey (see "Pilot Project Plan").

Importance of Changing Behavior

Keep in mind that many people at Ford are engineers. We were taught in the Cartesian/Newtonian paradigm—nonlinear systems thinking is not in our vocabulary. Our language is one of certainty, prediction, and results. Complexity is to be eliminated; the unknown is unacceptable.

In the 1980s going into 1990, I was responsible for several programs that were very successful. But we had huge armies of engineers and manufacturing people to deliver those programs, and every time we went into the implemen-

tion of a program, we were in trouble. The prototypes were late, we had too many engineering changes, there was confusion on the assumptions. We were missing our objectives, we were missing timing, and we had to allocate incredible resources to recover. We recognized that it just wasn't the way to manage a program, so we began to look for better ways.

Through our work with MIT, we're beginning to realize that certainty is not possible. The world is more complex than it used to be. Competition forces us to realize that some people do things

Organizational and Mental Barriers

Our organization is not any different from most product development organizations in North America. They tend to be very control-oriented, risk-averse, authoritarian, and hierarchical. There's a tendency for line management to walk into a situation and think they have the answer. They're very hesitant to admit that they don't know. To turn that around and to get people talking to each other and thinking together—developing shared mental models—is very difficult.

In most organizations there's a tendency to advocate individual positions rather than inquire into other people's thinking, which creates barriers to real communication. We wanted to eliminate the barriers we had created in

Pilot Project Plan

Phase I

1. Identify Key Themes and Interrelationships
2. Create Action Maps Around Themes Using Data Gathered
3. Construct Systems Maps from Action Maps and Data
4. Identify high leverage actions and Design interventions
5. Test/Validate through Systems Map and Computer Simulation
6. Pilot test intervention

Phase II

7. Extend process/success/learning
8. Refine process of reflection and ongoing learning

our own minds about how we should communicate and what our belief systems are about one another. To help, we used a tool called the Ladder of Inference (see "Ladder of Inference" diagram). It's very simple. When you have a conversation, or if you're articulating what you think is a problem or issue, the ladder gives you a way of questioning at what level of thinking are you discussing those issues. Is it at the level of beliefs, inferences, conclusions, cultural meaning or directly observable data? We found most of our

Continued on next page

Continued from previous page
discussions were somewhere up at the level of beliefs.

For example, we had some serious arguments with our finance office regarding what we wanted this car to be. The finance office wanted to achieve certain financial objectives, and we wanted to achieve certain product objectives. In the midst of an argument, one of the core team members said, "You want a Lexus at an Escort cost, that's what you want! That's an oxymoron." A heated debate ensued, but there was insight when we suddenly realized we were talking at a belief system: as a finance person, my job is to control you and get certain financial results; as an engineer, my job is to get a product that's competitive, and I need to get the kind of costs into that car to make it competitive!

In retrospect, this seems very simple. But at the time, we were trapped by our own vision of what our jobs were. Instead of thinking of what we wanted the car to be, we were thinking about our positions—my job is to be the

controller, my job is to be the engineer, my job is to build the car. And we couldn't communicate because we were operating at the level of beliefs.

In addition to the Ladder of Inference, we also used systems archetypes such as "Fixes that Fail," "Shifting the Burden," and "Tragedy of the Commons" to see our product development process more systemically (see "Tragedy of the Commons" for a systems archetype example). We struggled with them, but as a result of our work with archetypes, we were able to identify the leverage in managing change on our program. And because we're able to manage change more effectively (and earlier in the program), we will save millions of dollars in tooling.

The Learning Lab

We, the management group, needed to go through literally seven to eight months of working together to become a cohesive core group, before we could think about how to intervene in the rest of the organization. Throughout the project, the line managers were respon-

sible for learning and applying the tools to our own issues. The MIT people assisted us, providing the knowledge and the tools, but we were the ones who had to conduct our own interviews, analyze our own data, and learn to see and think differently. We changed as a result. We began to realize that the role of manager is not to boss and to direct, but to also become a teacher, a facilitator, and a coach.

We put together a two-day learning lab in order to share what we had learned with some of the other members of the Continental team (see "Learning Laboratories: Practicing Between Performances," October 1992 for a typical learning lab design). I was a co-facilitator with Dan Kim because I was being trained to ultimately train others.

We started with a short introduction to tell them what this was all about and how we got where we did. We then went through some exercises on the way we think and the way we create our current reality, but we were very careful not to abstract this too much. We talked a little bit about deconstructing problems; how problems are very frequently created by the way we look at them. We also introduced them to the five disciplines of the learning organization—shared vision, personal mastery, mental models, team learning, and systems thinking (from *The Fifth Discipline: The Art and Practice of the Learning Organization*, by Peter Senge).

Throughout the day, I brought up specific problems we were having on the program to help ground the issues in our current reality. On the second day, we used a management flight simulator for the product development process and challenged the participants to balance three objectives: timing, cost, and quality. It was a big hit. By practicing with the simulator they discovered how difficult it is to balance the three and how little changes in the beginning can dramatically affect the final outcome.

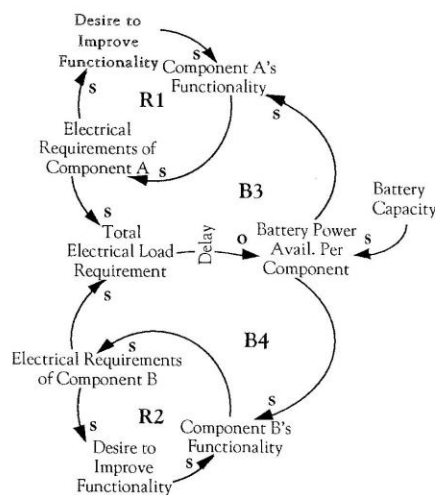
A Measure of Success

The success of that first learning lab was really surprising. I asked the participants to go back and start using these techniques and tell us about their experiences by keeping journals. When

Tragedy of the Commons

Early on in the project, we got stuck in a "Tragedy of the Commons" that lasted for several months. We had inadequate battery power in the vehicle because of the content we added to it, but we couldn't put in a bigger battery or an alternator because the package was set. Neither side would budge, because it was in each person's interest to look out for his or her own components. The team leader for the electrical components finally realized that neither side could solve the problem because it was a "Tragedy of the Commons." No matter what he did, each person would still look out for his or her own interest unless a) somebody discovered new technology, which wasn't going to happen in the next few months, or b) somebody from the outside came in and dictated.

What did we do? I came from the outside and dictated. It's not the best way to do it, but it worked, and they accepted it. Why? Because they understood that in a "Tragedy of the Commons" situation, the solution cannot be found at the individual level.



attending meetings with other learning lab participants, they've found that they understand each other when someone tends to respond in classic behavior. In other words, they're beginning to see the tools helping to surface their mental models.

For example, I was concerned about one of my younger engineers because he was working extremely long hours. One of the managers said the guy was going to burn himself out and I'd better go talk to him. So I walked up to him and started fishing for stuff, asking how he was doing. After a few minutes he said, "Nick! What are you trying to say? What's on the left-hand column of your mind? Why don't you just come right out and tell me?" I did, and the issue was resolved. I was impressed! I'm beginning to see that type of exchange happen more in meetings. Now someone will say, "John, where are you on the ladder of inference? We're not going to get anywhere if we're going to discuss this at a belief system level."

The real test is going to be taking these 20 engineers and cross-functional leaders and going through another couple of pilot programs, training them to be trainers. In the next pilot program, Dan is going to watch and I'm going to teach. One or two after that, I'm going to watch and they're going to teach, and then their team members are going to teach other team members. I want to be able to take this to the whole team of approximately 200 people in the next six months or so.

Learnings

There are four main lessons I've learned through this project. One is the role of the manager—it is critical. The best place to start with a project like this is to find some people who are willing to experiment, and work with them first. You have to have a champion, someone who's committed. It's best if this role is assumed by the line manager.

Next, we need to work on getting rid of this obsession with problem solving. It becomes a barrier to more effective learning. We need to start thinking about re-articulating issues—to get people to redefine what they think is the problem. Many people have heard

the story that we actually create most of the problems in our own mind. They're not *out there*; they're in our minds.

I've also found the systems archetypes are very useful, but I have this fear that they will become an obsession in themselves. I think many people believe the more complex they are, the better they are. That's wrong. In my opinion, the simpler we can make the archetype, the better. The archetype is a convenient tool to hang our thoughts on, and that's all it is. If you can do without them, fine, but I think they're very useful. When people start saying things like "that's a 'Shifting the Burden!'" Or, "those are 'Fixes that Fail!'" Or, "this is a 'Tragedy of the Commons!'" everyone instantly understands what you're talking about. They all have the same picture, and that is a basis for communicating.

The fourth insight is the importance of the individual and of personal transformation. One of my engineers said, "Okay, I'm enlightened. I know what to do, I know how to do it. But what about THEM? I have to go back to that place, and it's awful! I've learned so much, but then I have to go back there!" I told her not to worry about that; just worry about herself. It starts with personal transformation. Start using these tools yourself, and let others watch. They'll ask; they'll wonder.

I believe the most powerful point is that in the end it's really up to the individual. We have to stop trying to be advocates and fighting the system. We have to start realizing that if we fix ourselves, then even if we can't fix that other person, it'll be easier to deal with the situation. The point to remember as we proceed in the direction toward change and becoming enlightened, is that conventional wisdom is also present, and when situations get a little

The Ladder of Inference

Rung	
4	The theories we use to create the meanings on rung 3 (Beliefs)
3	Meanings imposed by us (Assumptions and Conclusions)
2	Culturally understood meanings
1	Relatively directly observable data, such as conversations

Adapted from Argyris, *Overcoming Organizational Defenses*

The ladder of inference gives you a way of asking, "At what level of thinking are you discussing those issues: the level of beliefs, assumptions and conclusions, cultural meaning or directly observable data?"

tough, there's a tendency to go back to conventional ways. We need to realize that this is our work. It's how we can become better and more effective at our behavior. ☐

Nick Zeniuk is a leader for organizational and behavioral change in Ford's product development process. He has extensive multi-functional experience in planning, finance, engineering, manufacturing, and marketing. Nick was the Planning Manager for several key products, including the Brazilian Del Ray (1981 Car of the Year Award), Mark VII and Lincoln Town Car (1990 Car of the Year Award).

*Further Reading: Chris Argyris, *Overcoming Organizational Defenses* (Needham Heights, MA: Allyn and Bacon, 1990). Available through Pegasus Communications.*

Mark Your Calendar!

The votes are in...
The 1993 Systems Thinking in Action Conference will be held at downtown Boston's
Marriott Copley Place
November 8-10, 1993

We are now accepting presenter and topic suggestions. To receive a presenter information packet, contact Anne Coyle at (617) 576-1231.

Watch for more details in upcoming issues of *The Systems Thinker*.