

**The Learning Initiative at the
AutoCo Epsilon Program,
1991-1994**

George Roth and Art Kleiner

May 8, 1996

SWP# 3969

The Learning Initiative
at the
AutoCo
Epsilon Program,
1991-1994

George Roth
Art Kleiner
Center for Organizational Learning
MIT Sloan School of Management
30 Memorial Drive
Cambridge, MA 02139
May 8, 1996

Forward

The organizational learning history you are about to read emerges from a unique collaboration among practitioners, academicians and product development professionals. It is an important document because it lays out the dilemmas, paradoxes, and human emotions associated with teamwork in today's complex organizations. Our future will most certainly depend on how well we learn to manage conflicting needs in large systems.

This study is more about learning how to learn, than about the nuts and bolts associated with designing and building great automobiles. More often than not you'll conclude it's not about who is right or wrong, but about a world of perception and interpretation. For me, this is a human story because it reveals how different attitudes, beliefs, and assumptions rise to the surface, and may rule the day.

What's especially revealing is how the product development function is demystified as an exact science of equations, engineering procedures, and computer-driven technology. Instead, you'll find dedicated people at all levels relentlessly seeking alignment, recognition, and assurances that the day's effort will yield value-added results over chaos and self-interest. This dedication also requires a balanced perspective. We can be extremely efficient by way of quality, cost, timing, and flexibility. But, these objectives must be in service to the customer. Outstanding teams of the future will need to balance multiple initiatives more than ever before.

For me, this learning history is about a beginning, not an end. We are building on what we've learned with this first MIT effort by applying the methods and tools in two other vehicle programs. Additionally, there are many organizational learning projects going on in the Company outside of product development. Perhaps this will enable us to see the connections among all these efforts and move to yet another new level of understanding.

Senior Vice President - Product Development
AutoCo

Ywp50/pv11026

Executive Overview

This is the story of a group of 300 people, the Epsilon program team, charged with meeting the typical “tight” deadlines to get a vehicle out the door. They resolved to do it without the costly, hectic, last-minute “heroic” efforts that had dominated most automobile launches in the past. They discovered, along the way, that this meant not just developing new management techniques and a “systemic” understanding of their work, but recreating their relationships with one another and with the rest of the company.

It is also the story of a particular type of partnership, between managers in Epsilon and researchers from the MIT Center for Organizational Learning. The project from which this learning history is drawn began in the summer of 1991, when senior managers from Epsilon and MIT researchers agreed to explore how Epsilon might incorporate tools for systems thinking, improving mental models, and nurturing personal and shared vision into its product development process. The particular approach (written about by Peter Senge in The Fifth Discipline) used to deepen and accelerate team learning is a synthesis of individual approaches that evolved from more than twenty years of research at MIT, Harvard and elsewhere. Elements had been applied within consulting practice, yet it had never been “tested” in a multi-year, practical work setting. The researchers envisioned creating “managerial practice fields” to enable people working together to periodically step back, reflect, talk together, and thereby deepen their understanding of the systemic nature of pressing problems. They hoped to establish practice fields that would serve Epsilon's managers' desire to improve cost, timing, and quality and allow them to study how practicing managers developed new collective learning capabilities.

The AutoCo Learning History is one outcome of that study. It is a jointly-told tale by the participants and the researchers. In writing the learning history, we were asked by AutoCo to summarize what was learned. We suggest managers at AutoCo and elsewhere consider the following four lessons:

- The combined impact of the learning tools and committed managers willing to genuinely learn and grow can lead to significant enhancements in local business results. The Epsilon team met or exceeded many product development timing records at AutoCo and produced a vehicle with a top quality rating. In its first year of release it bettered quality ratings of the car it replaced, unusual for a new model year.
- New types of interrelationships, attitudes, and thinking can't just be decreed. They must be allowed to grow. There is no substitute for individual responsibility in learning. Senior managers can help most by giving people room to experiment within the constraints necessary to deliver the program, and by striving to serve as examples for the behavior they're trying to produce.
- With a modicum of specific attention to “building better conversations,” a flood of innovation poured forth. Managing the “softest” aspect of the team — the ways people thought and communicated together — gave the highest leverage for improving “hard” results.
- Having a great team is not enough, even when that team has as many as 1000 full-time equivalent people. The relationship between the team and the larger organization is crucial. The team learned and performed together, yet its ability to diffuse what it learned and impact the larger organization is an open question. Diffusing learning requires extraordinary efforts on the part of team leaders to help executives to

understand new workplace innovations. It requires extra attention (for example, the willingness to talk through potential misunderstandings, or to make themselves available for each other “beyond the call of duty”) because innovations can easily be misunderstood. Executives have managerial accountability for the team's business performance -- performance that they fear might be compromised when they don't understand the new practices.

However, merely stating such lessons sheds little if any light on *how* the team actually developed the capability to achieve these results and *how* they still managed to run into problems.

That's why even a busy manager or executive might find it useful to read the eighty-page historical account of another group's change effort. By seeing the struggles, doubts, misunderstandings, and varied points of view of the Epsilon project, *as told in participants' own words*, you might get a sense of what would be necessary to promote increased learning capabilities in your own projects. What worked for Epsilon cannot be readily applied to different groups and different contexts, but there are insights that we think can help other teams: an appreciation of how new learning tools and processes are actually used in practice, and what it means personally for people to develop systems thinking and mental models skills. Does the law of physics — that for every action there is an equal and opposite reaction — apply to social systems? Was it Epsilon's workplace innovations that released larger organizational forces?

What would another team need to repeat Epsilon's success — and avoid its failures — in the future? We believe the learning history shows that a significant success factor was this: Epsilon's leaders designed their work as a learning process. Reading the learning history, you will see a sense of humility and mutual engagement at work. Team members recognized that they had endemic problems which went beyond strict engineering issues, into issues of organizational communication. They also recognized that no one on the team had all the answers. Answers would have to emerge from the give-and-take between members of the team. They invested time and money in "learning labs," a "learning room," and "harmony bucks," as ways to bring people together regularly to work out critical design and team issues in an atmosphere of systemic understanding and dialogue. Through many such actions they established a sense that, “We're all in it together, because we are all connected together.” The learning history also shows, by contrast, their inability to create a comparable learning orientation in dealing with upper management, perhaps because they never had the partnership of upper management in the first place. How to build effective learning partnerships across hierarchical levels has now become a central question for managers and researchers alike.

Organizational learning is a process of collective sense-making. You don't just produce results; you produce a “theory of how you got there.” If Epsilon's participants seem uncommonly reflective for auto engineers, that is not merely a result of the learning history effort, which asked them to look backward. It also stemmed from what they learned as a team, and also, arguably, from the new atmosphere they created at Epsilon.

Themes of the Epsilon Learning History

The AutoCo learning history is organized around six themes. These themes present the important areas for describing and explaining the Epsilon program's achievements (see "Noticeable Results" on page 84) and include materials which provide a chronology of the three-year project (also see project timeline on page xiii).

- ① **Hard results, soft concerns.** When managers pay attention to human issues like openness and fostering trust, would teams be able to produce better business results? In Epsilon, the focus on how managers think and interact started with nine months of working sessions in a cross-functional leadership team composed of most of the senior functional managers in the Epsilon program. These sessions aimed to foster shared vision and shared understanding of one another's mental models in the context of addressing the program's practical problems. Thus, the senior team management began the learning process long before the rest of the team, which enabled them to jointly design the evolution of the process.
- ② **Setting an example of non-authoritarian leadership.** Many experts and consultants preach the need for a more non-authoritarian and participative approach to project leadership but can offer little help in how to develop and sustain such behavior. In Epsilon, this philosophy became reality as project leaders' behavior changed over time as a by-product of the tools and learning processes employed. For senior leaders, "walking the talk" is not a trivial matter. It requires concerted effort and mutual partnership. And it can make a huge difference.
- ③ **Introductory Learning Labs: Teaching techniques for thinking differently.** Eventually, a two-day "learning lab," taught by program managers and MIT staff, was created to introduce many members of the Epsilon team to the learning tools and methods with which the leadership team had been working. "Learning labs" may include a variety of techniques, but the key goal is inviting more in-depth conversation across functional boundaries, enabling people to focus on key business-related issues in a risk-free setting accessible to all.
- ④ **Combining engineering innovations with human relations: The Harmony Buck.** Combining new technical ideas with greater trust and new interpersonal skills (a "human relations" approach) can enable people to apply the technical ideas more effectively. The "Harmony Buck" speeded up prototyping by allowing people to come together and try out new engineering solutions. But it also built on the growing environment of involvement and openness and in turn contributed to that environment. The result was an increased flow of information among team members testing their ideas together.
- ⑤ **Partnerships.** Functionally based people were drawn together in ways that bridged differences and focused on collaborative learning and action. An atmosphere which encourages experimentation across traditional boundaries leads to benefits that the senior leaders can't necessarily predict or plan for.
- ⑥ **Process innovation in the context of a large organization.** Eventually, local process innovations are brought into larger management forums. The larger AutoCo organization responded to the Epsilon team in many ways, not always in ways Epsilon's members would have wished for. Innovative local line leaders often put their faith in proving that their innovations will lead to better business results, and that these results will bring credibility to their efforts. This assumption proved faulty for Epsilon's managers. Lacking senior management partners, they also lacked counsel on how to handle the larger system implications of their efforts.

The leaders of the Epsilon Program worked to change how they worked together first. Only then did they encourage other people to change their styles. Although use of learning tools and the learning labs created means to support new skills and new behaviors, we of the learning history team heard again and again that the changes in *senior managers' personal behavior* was the critical force. This change in behavior among the Epsilon team leaders allowed others on the team to change, be more open, and share their difficulties and mistakes rather than avoid embarrassing situations. That openness, in turn, gave Epsilon its capability for quality, its flexibility, and its inspiring atmosphere.

It is the goal of this learning history not only to document the experiences of an innovative team, but to provide a springboard other teams can use to learn from Epsilon's experience. Learning from others' experience is not easy. Simple lists of "lessons-learned" or technical tips might help, but they rarely provide insight into the process whereby innovation occurs or how it actually "feels" for the participants. On the other hand, if nothing is conceptualized and communicated, knowledge remains "tacit," below the surface, and can only be learned by direct contact with the innovators -- an inefficient process for diffusing innovative practices within an organization and an impossible one for those outside the organization. Learning from the Epsilon Team's efforts starts with appreciating the tools and methods they used, brief descriptions of which are included below (with references to more detailed descriptions). What this learning history adds is the *story* of what was done when, how it affected people, what those people accomplished, and the individual and collective challenges of deep learning. It is our hope that this learning history can stimulate inquiry and conversations in other teams, enabling them to discover their own commitments, questions, and tacit knowledge. To help, some initial questions to be considered and discussed after reading this history, are included in the "Appendix" on page 86.

The story which follows begins when executives gave final approval for Epsilon's development budget, in which styling, market positioning, and expenditures were set. The program was already many weeks behind schedule. At the same time, "process improvement" as an endeavor was beginning to gain importance and attention at AutoCo. Making a great car wasn't enough; you also had to improve AutoCo's process capabilities.

Seeking a better process for developing vehicles, the Epsilon researchers explored tools and techniques gleaned from the work of MIT's Center for Organizational Learning. They integrated program management and training, so that all work on the car could involve systems thinking and collaboration. Early efforts were focused on bridging the barriers between functions: creating a shared vision of the new vehicle, collocating engineers in one large multi-functional building, and bringing design engineers into the market research process at an early stage. As the team progressed, its vehicle development metrics went from low initial scores to setting new company records for prototype-build parts availability and quality.

The Epsilon program completed its assignment at the end of 1994. The vehicle launch, which took place a week earlier than scheduled, was truly a "non-event," without the crisis atmosphere that normally leads to legions of engineers camping out at the manufacturing plant. However, the Epsilon results also included controversy. The launch coincided with organizational changes at AutoCo, in which Epsilon team leaders did not

receive accolades for their accomplishments. As team members were assigned to new positions, some wondered if their efforts were valued or appreciated. Yes, they had broken performance records; but they had also broken some behavioral norms. For example, reports of problems had been deliberately brought to the surface earlier than usual. This had saved money and improved quality in the long run, but had also led to the appearance that the Epsilon program was “out of control.” We propose that as readers, you consider what lessons from Epsilon’s experience you can apply to create and sustain an atmosphere which encourages experimentation and learning while producing better business results.

Contents

<i>Forward</i>	i
<i>Executive Summary</i>	iii
<i>Contents</i>	vii
<i>Preface</i>	ix
<i>Chronology</i>	xi
Origins of Epsilon’s Learning Project	1
Theme 1: Hard results, soft concerns	7
<i>Learning in the core team (Jan. 1992 - Sept. 1992)</i>	7
<i>The system map (Learning Team Meeting, August, 1992)</i>	13
<i>The transition to openness</i>	15
<i>Creating the atmosphere of trust and cooperation</i>	17
• A mandate that “bearing bad tidings” would be safe.....	18
• Ongoing sharing of information and perspective.....	18
• A culture of greater inclusiveness.....	19
• Deliberate encouragement of informality and friendship.....	19
• A mindset that “no one has all the answers”.....	20
<i>Behavioral versus technical: a zero-sum game?</i>	21
<i>The dilemma of integrating process and engineering knowledge</i>	22
Theme 2: Setting an example of non-authoritarian leadership	25
<i>Leadership roles: modeling new behavior</i>	25
<i>What made it possible for leaders to change their behavior effectively?</i>	28
The core team meetings.....	28
Program management demonstrated commitment by taking part in every learning lab.....	29
Partnership: Having the two leaders learning in tandem provided a necessary support for one another.....	29
Purpose: the Program Manager and the Launch Manager never came into the project saying, “We want to become better people” for our own sake.....	29
Reflection: The leaders continued to increase their own capability to understand changes in themselves and others.....	29
<i>Modeling new behavior: “I don’t trust you”</i>	31
<i>Damage from backsliding</i>	34

Theme 3: Learning labs: Teaching techniques for thinking differently	35
<i>Designing the learning labs: from the beginning.....</i>	35
<i>Choosing learning lab participants.....</i>	37
<i>The systems archetypes and “systems thinking” skills</i>	38
The tragedy of the power supply.....	38
<i>The ladder of inference and “mental models” skills.....</i>	42
<i>The “management flight simulators”</i>	44
<i>Reinforcement: A learning room.....</i>	46
<i>Reactions to the learning labs</i>	46
<i>Learning Labs: How did they contribute to the change process?</i>	48
Theme 4: Combining engineering innovation with human relations: The Harmony Buck ..	51
<i>Getting approval: Building confidence to make a case for taking a risk.....</i>	52
<i>Implementation reveals resistance within the team.....</i>	54
<i>The harmony buck as a communication tool.....</i>	55
<i>Expanding the collaborators.....</i>	56
<i>The second harmony buck: Approval and decline.....</i>	59
Theme 5: Partnerships.....	61
<i>The market research clinic</i>	61
<i>Collocation: Opening a new realm of issues</i>	63
Theme 6: Process innovation in the context of a large organization.....	67
<i>Positioning the purpose of the team.....</i>	68
<i>Engaging senior management.....</i>	68
<i>Evaluating Epsilon: Miscommunications and misunderstandings</i>	71
<i>Implementing the new CR policy.....</i>	74
<i>Epsilon is “Out of Control!”.....</i>	76
<i>Freezing and reducing the change requests.....</i>	79
<i>The early retirement</i>	80
<i>In the end: Assessing the influence of innovation.....</i>	83
Epsilon’s Noticeable Results, 1991-1994.....	87
Appendix: Some initial questions for group discussions prior to moving forward with learning initiatives.....	89

Preface

This document is a learning history of collaborative efforts undertaken by AutoCo and the Massachusetts Institute of Technology Sloan School of Management Center for Organizational Learning. These efforts took place between 1991 and 1994 on the Epsilon vehicle development program. The project's objective was: To develop and study learning capabilities in a product development team of more than 300 people, while having a positive impact on business results. The purpose of this document is to report on those efforts and create materials from which other interested teams might learn.

The name of the company, AutoCo, is a pseudonym for an automobile company. Epsilon is also a fictional project name, as are all other formal names used in this document. People in the learning history are identified only by their titles. The company, program, and people are disguised to provide anonymity, protect AutoCo's need for confidentiality, minimize distraction and help the reader to focus his or her attention on the universal themes herein.

This document was written by a small "learning historian" team composed of people from AutoCo and from MIT. We developed the learning history in the hope that it could help other teams and individuals at AutoCo, and other companies, to benefit from this program's experience.

A learning history is a new format for presenting the story of a project. It is designed to portray the project as participants experienced it, and to invite readers to draw their own conclusions. In this history we make the "sense-making process" visible — we report not just what people *did*, but how they interpreted events around them and what reasoning led to their decisions. To gather this information, we interviewed over fifty individuals. The interviewees included engineers, process leaders, content leaders (see Sidebar: "Content leaders" and "process leaders" page 15), and managers at all levels and functions within the Epsilon team, along with suppliers, engineers from other functions, senior AutoCo management, and other key figures at AutoCo. We also reviewed transcripts of meetings, interviews, program documents, and speeches given by key participants during the program. People's perspectives and attitudes varied; we have made an explicit effort to include as wide a range of points of view as possible.¹

The value of this document depends on the conversation it generates: How can AutoCo's Epsilon experience provide a useful example for *your* team or project? We ask readers to suspend their assumptions — about automobile companies, management, engineering, and all other aspects of vehicle production — so that they can focus on what happened, how people described events, how they felt and what their attributions were.

The learning history report starts with an overview, and then there is a chapter describing the origins of the Epsilon learning effort. (The critical events and observable measures which provide data on how the Epsilon program progressed, "Noticeable Results" are listed at the

¹ The learning history team used a grounded theory, qualitative data analysis methodology to discern key concepts and patterns. See Strauss, 1987, *Qualitative Analysis for Social Scientists*; Corbin and Strauss, 1990, *Basics of Qualitative Research*; Miles and Huberman, 1994, *Qualitative Data Analysis*, and Glaser and Strauss, 1967, *The Discovery of Grounded Theory*. In analyzing of large quantities of qualitative data with a team of inside and outside learning historians, we have found it helpful to think in terms of meeting three "imperatives"— research (loyalty to the "data"), mythic (loyalty to the "story"), and pragmatic (loyalty to the audience's needs). Each of these imperatives represents a set of "pure" priorities — all important, and all in contention with each other. They can't be approached simultaneously. They are attended to in sequence, but there must be deliberate, balanced consideration of all three, in every phase of a learning history effort.

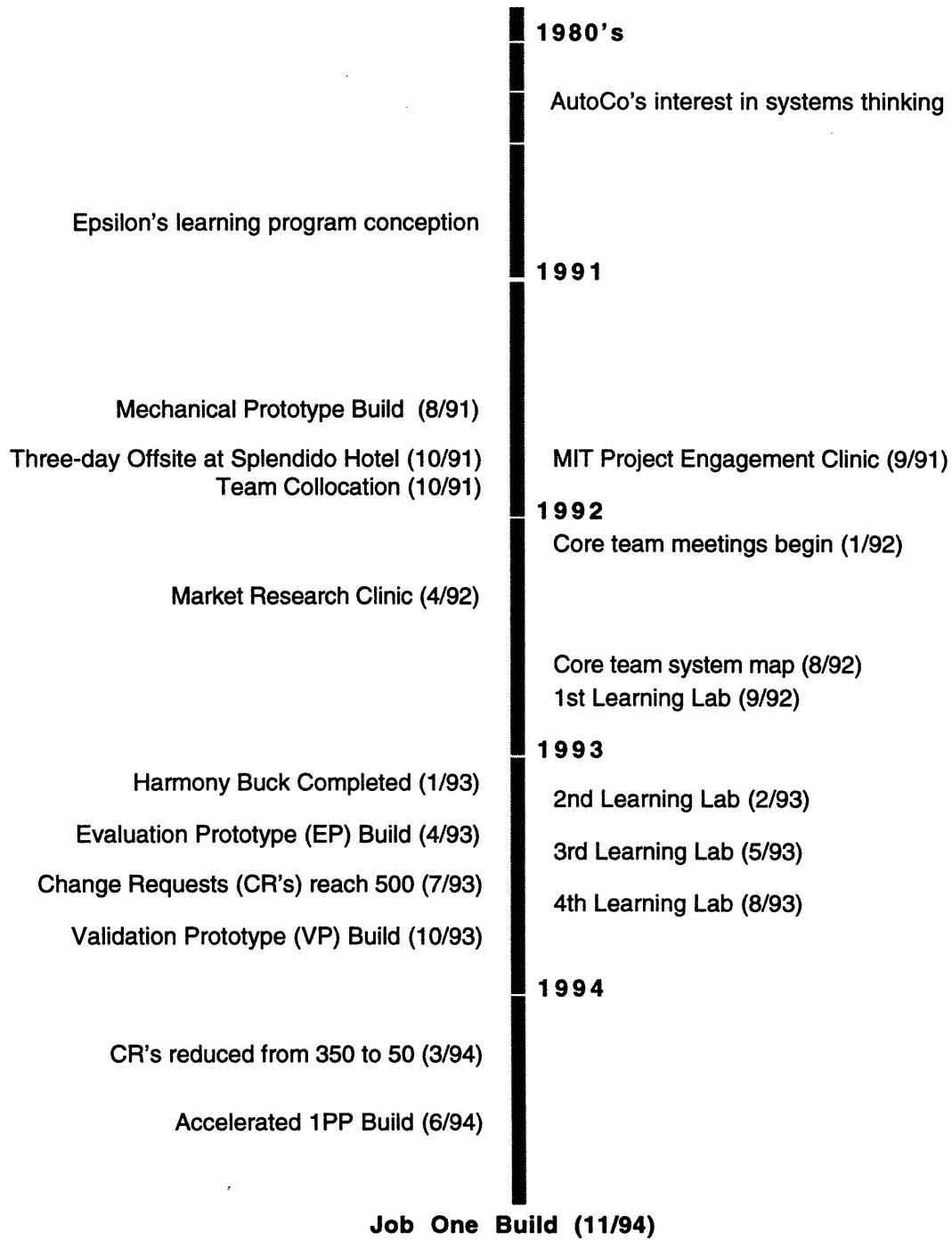
end of this document, on page 87.) The subsequent sections represent themes — key concepts which represent the underlying significance of this project, and which emerged from a close reading and examination of the materials collected in our research. We present each of these themes in the form of a “jointly-told tale,” separating the researchers’ comments from participants’ narrative. There are four different types of material in these “jointly-told tales:”

- The right-hand column of text, within each theme, tells the story in the words of key participants, taken directly from the “primary data” of interviews, speeches and meetings. (Each participant has seen, and approved, his or her quotations.)
- The left-hand column of text provides interpretive and synthesizing material: questions, analysis, generalizations, and implications, developed by the learning historians to help readers begin to apply the material to their own situation.
- There are also full-column passages which introduce topics, provide context, and set the stage.
- Finally, boxed “sidebars” provide background information on methods, tools and key topics, referred to in the text, which would otherwise distract you from the narrative.

In reading the two column format of the “jointly-told” tale sections, you will find yourself having to make a choice. Which column do you read first? Do you skip back and forth, and when do you do so? There are no “rules” for reading a learning history; different people read segments in different orders. As you make your way through the story, however, please pay attention to your own reactions. How credible do you find the story? How would you have dealt with the problems that faced the Epsilon team? How can their experience help inform the decisions that you (and your associates) have to make in the future? It is through the discussion and dialogue with colleagues, about the contents of this document, that we believe your own and your team’s learning will best be served.

— AutoCo Epsilon Learning History Team
Five internal AutoCo team members
George Roth & Art Kleiner

Chronology



Origins of Epsilon's Learning Project

AutoCo's interest in systems thinking (which later included organizational learning) began in 1989, when Peter Senge (who was developing a Center for Organizational Learning based at MIT) and Russell Ackoff (professor emeritus at the University of Pennsylvania's Wharton School) started giving monthly presentations in AutoCo's Executive Development seminars. A manager in AutoCo University [AU Manager], who sponsored the monthly training sessions, was interested in testing these systems thinking concepts in one or more live business settings at AutoCo.

Note how seeds were sown for some time before a project opportunity emerged at AutoCo. The company "pulled" the effort in, rather than being "sold" a bill of goods.

As at other large companies, AutoCo internal "change agents" had to consciously decide between a "bottom-up" or "top-down" approach in any given initiative. From the beginning, this effort took the "bottom-up" approach. This meant it would be easier to implement, but harder to expand to fit the larger AutoCo system.

Note how this very concrete learning effort begins with three abstract "governing ideas." It's not clear to outsiders whether a concept like "thinking differently" meant the same thing to different AutoCo managers and executives.

AutoCo called upon well known academics and consultants as part of an education program for senior managers. Each issue had its own leading "expert," with the exception of systems thinking, which had two strong voices: Peter Senge and Russell Ackoff. These two experts held, in common, the view that an organization's work could not be understood in fragments. AutoCo's managers responded to this message

AU Manager: We had just finished our first executive education program for the top 2000 people worldwide. It was a gathering from the four corners of the world, and it was quite a happening. The question was, what should be in the second round?

We decided on a "bubble-up" rather than "top-down" model. We went around the world and interviewed executives and asked them what was on their minds. What would be of the greatest interest to them? Three issues surfaced: globalization, thinking differently and leadership.

Underlying the first two issues was the pervasive issue of change. We then went about exploring what would be a senior executive program built around these three themes. There was controversy in presenting them to the top of the house because the top felt they might not be ready to get into all this subject matter. Nevertheless, they said, "press forward."

In the arena of "thinking differently," we came across two outstanding voices: Peter Senge and Russ Ackoff. The more we dug into the area, the more we found a very significant message coming out of Senge's and Ackoff's world views.

By late 1989, early 1990, we had both Senge and Ackoff doing a program at our center every other week. In our analysis of participants'

because it helped with the perennial problem of miscommunication and conflict between functions.

Intellectually thought provoking and personally compelling, but very abstract, the ideas in systems thinking were flavored by their academic origins. AutoCo managers were intrigued but they challenged the academic stance by asking, "How are these abstract concepts applied?"

The AU Manager took the response of senior executives as a challenge to find a way to operationalize systems thinking throughout AutoCo.

The comments people asked about application led to their being critically challenged.

Is it possible for executives who have spent their lives thinking in a particular way to change their thinking?

reactions, there was a large voice saying, "The ideas are intriguing, but I don't see how I would play them out in my ballpark – on Monday morning!"

In all fairness, we had asked both Senge and Ackoff to take us on a broad journey, and not to focus specifically on application. It's not surprising that participants were intrigued with it, and saw its depth, but they were right in feeling that there needed to be an ability to see further down that chain of, "What happens next?"

That's when I formulated the challenge for myself and AutoCo University to continue to pursue this subject area. I had the good fortune to be able to sit in on numerous sessions with Peter and Russ, and in the fifth or sixth session, the ball bearings started to rotate in unison.

I asked Ackoff publicly, "Russ, I've been sitting here for several sessions, it's an outstanding message, but I'm still having trouble digesting it and its implications."

Russ turned to me and said, "Well, that's because you'll never get it."

I turned beet red. Here I was standing in front of 50 executives, and the room was dead silent. Then Russ let me off the hook. He turned to the group, and said, "And you won't get it, either. We have built up over 400 years of methodology of 'reductionist' thinking. It is so powerful, so pervasive, that probably your children and their children will have a much easier time. For you folks, it's going to be tough grind."

Down to my socks, I understood that you can say you understand it, and still not understand it. The implications were absolutely profound. Organizational learning didn't mean letting go of analytical processes. It meant complementing and supplementing them with synthesis or

systems thinking. It's not such a clean thing - "Just throw out all the traditional tools, my past life - and switch into new formulas." It means learning something in addition: the "and," not the "or."

AutoCo's introduction to systems thinking thus represented a challenge: How could the organization make use of systems thinking in a business context?

In 1991 a diverse group of AutoCo managers attended a series of five two-day training sessions on the core competencies of a learning organization run by the founding staff from the Center for Organizational Learning. To the surprise of the AU manager, given the abstract nature of the materials, the first audience exposed to the concepts was receptive. This audience was composed, in part, of managers from product development. In particular, the Epsilon Program Launch Manager expressed keen interest. He was responsible for the vehicle development program to design and build the next model of the Epsilon luxury car.

Past experiences in developing cars were powerful influences in trying to create a better approach on the Epsilon program.

Launch manager: I had worked on the Delta program [another vehicle program] for several years as the Business Planner and Launch Manager. We had discovered, a year before Job One, that the program was 17 months behind schedule. So we quickly organized a 100-person launch team and we put the program back on schedule, with quality that was better than the first car. We met all of our program objectives except cost, which we knew from the beginning we could not meet.

Although good quality and timely delivery are held up as critical, achieving them does not make cost overruns acceptable.

I remember a meeting where a Vice President listened to us present the reasons for our success [on the Delta program]: team leadership, and the fact that everyone had the same goals and knew that they depended on each other. "That sounds really great," he said, "but what did you do?" He finally said it must have been a fluke, and that was the end of it.

This is not the only example where attributing good performance to team dynamics was not accepted as a convincing explanation by executives. It seemed that executives expected and listened for technical innovation, rather than management changes.

There was no learning from the experience. It bothered me a great deal. AutoCo is in love with managing by crisis; without a crisis, we don't know what to do. I resolved to learn

Do the ways in which success is achieved in large corporations have to have task oriented, replicable type answers to be accepted?

how to produce a car launch without a crisis.

In July 1991, the AU Manager, along with an internal consultant working with the Launch Manager, wrote a letter to Peter Senge, director of the MIT Center for Organizational Learning. They requested an active relationship between MIT researchers and the Epsilon Program.

Project Engagement Clinic

The formal partnership between the two organizations started with a project “engagement” clinic in September, 1991. The goal of this clinic was to engage one another in asking difficult questions about readiness for learning, and explore possibilities for a partnership which sought business improvements and research results. Attending the meeting were the key people in the project — the Program Manager, Launch Manager, and Body Engineering Manager for the Epsilon Program, two internal consultants on process improvement (who were to integrate systems thinking at AutoCo) and five researchers associated with the MIT Center for Organizational Learning. One researcher had visited AutoCo and conducted interviews in advance.

These interviews were summarized in a report that singled out several key issues:

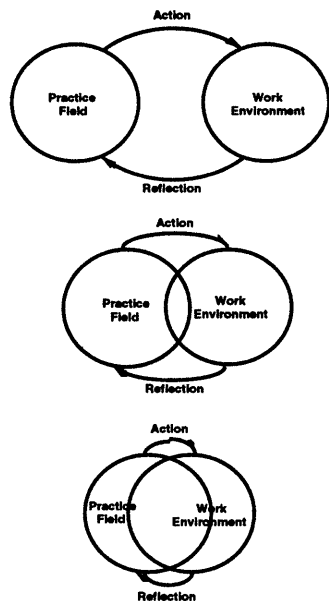
- The Launch Manager wanted the MIT project to focus on improvements in the Evaluation Prototype [EP]. If the EP could be made to “work” as it should, the car would be successful, he said. He wanted to create a climate within the team which reinforced more effective cross-functional communication, more responsibility for objectives, and less “games-playing.”
- The Epsilon Program Manager (the Launch Manager’s boss, with overall responsibility for the Epsilon project) believed that concentrating on one prototype, like the EP, was too narrow. He felt it was necessary to look at AutoCo's product development paradigm. Present product development management practices controlled resources ineffectively, treated suppliers with indifference or hostility, and had a history of not achieving quality, cost and time objectives. The Program Manager mentioned the inability of a program to reach a point where management could say, "Enough changes this time."
- AutoCo’s vehicle program management needed to see tangible improvement. The new product development program officially involved a 48-month time period, which was compressed to 42 months, and had very ambitious deadlines. Chronic dependency on "heroic" efforts took key people away from the planned product development process. Some people felt certain that it was just a question of time before a real crisis became obvious.

- MIT's researchers wanted the project to further the investigations being done in the Center for Organizational Learning. Supporting research would require more than a relationship where MIT played the role of expert consultant and AutoCo took the role of client. Instead, both organizations would have to collaborate in a systemic approach to improving the product development management practices and understanding the results that were achieved. Measurement would be important in determining if improvements occurred over time.

Following the project engagement clinic, several meetings were held to determine the project focus and how MIT researchers would work with the Epsilon team. In January of 1992, a core "learning team" of managers from the Epsilon program began meeting regularly, at one- or two-month intervals. The core "learning team" consisted of ten Epsilon managers (including the Program Manager, Launch Manager, Assembly Manager, Finance Manager, Body Engineering Manager, and Purchasing Manager, and two internal consultants). The lead MIT Researcher was present as a facilitator. These meetings provided opportunities to plan, learn techniques and tools that facilitated learning, and practice using those techniques between meetings.

Sidebar: What are “learning laboratories?”

“Learning laboratory” is a term used by the MIT Center for Organizational Learning for a workshop, often also called a “managerial practice field,” where people come to develop new skills, cycling back and forth between study and practice.



The learning lab is different from a training session in that over time its concepts and values are intended to become part of, to be integrated into, work issues and job settings.

In sports, the arts, and the military, teams are accustomed to practice sessions, simulating real events where members learn from mistakes and each others' examples. Similarly, learning labs simulate normal business settings, providing an opportunity for management teams to practice together and make mistakes without penalty or the pressures of performance. Participants learn new tools by applying them to the issues they face in their day-to-day jobs. The MIT-COL learning labs at AutoCo focused on skills of conversation, reflection, and systems thinking.

The two-day learning labs at AutoCo were designed and facilitated by Dan Kim from the Center for Organizational Learning and selected managers from the Epsilon program team. The Launch Manager was the first AutoCo facilitator, and he was directly involved in all the learning labs.

The designs were based on the experiences of the first core team, and modified after each session in light of the experience of new labs, the makeup of participants, and new issues faced by the program. Some participants in early labs became facilitators of later labs. Four learning labs were run during the program. Over 100 people — more than a third of the full-time, dedicated engineers on the program — attended.

The learning labs at AutoCo alternated between conceptual sessions for learning new tools (of conversation and systemic thinking) and exercises for practicing their use. These exercises were deliberately designed so that people could consider their own work issues with perspective that came from the deliberate telescoping of time and space. For example, a computer simulation allowed participants to spend an afternoon working together through a product development process that would normally have required three to four years.

For more about the learning labs at Epsilon, see **Theme 3: Learning labs: Teaching techniques for thinking differently**, page 35. For theoretical information on designing managerial practice fields and learning labs see Kim and Senge, “Putting Systems Thinking into Practice” in *System Dynamics Review*, 1994, Vol. 10, Nos. 2-3, pp. 277-290.

After meeting for eight months, the core learning team designed a series of learning laboratories, through which engineers on the program were to learn and apply techniques of systems thinking, working with mental models, and team learning. Those learning laboratories took place in September of 1992 and February, May and August of 1993. The efforts of the Epsilon program to apply organizational learning concepts spanned the following three years.

Theme 1: Hard results, soft concerns

When the Program Manager was appointed as manager of the Epsilon program his ambition was to develop an excellent vehicle *and* a process which valued and inspired people on the team. This section describes the challenge facing Epsilon: To achieve “hard” results (producing high-quality technical parts, sub-assemblies, and a complete vehicle efficiently) through an emphasis on concepts that many people termed “soft” (good communication, openness, honesty and trust.)

Learning in the core team (Jan. 1992 - Sept. 1992)

The core “learning” team began with an awareness that they faced “real internal conflicts.” The Epsilon program was unofficially expected to be a “Lexus-fighter at the price of an [American luxury car],” although as a front-wheel drive car with its planned variable costs, it would be difficult to compete with the Lexus.

The core team members all felt a high level of frustration, and a sense of urgency, since the program was already late. They recognized the dilemma of being a pilot learning project; they could already see that their effort might mean “taking on” AutoCo’s overarching culture. This culture included a strong reliance on hierarchy and functional authority — an expectation that the boss is on top of technical details and makes decisions. On top of that, the core team members had difficulty communicating with each other. “I couldn’t talk to [the Finance Manager] at first,” the Program Manager noted, “in less than a high-decibel range.”

At the beginning of 1992, the core team met every one or two months, working with a variety of communication and conceptualization techniques such as the ladder of inference, the left-hand column, role-playing, and system mapping (see sidebars

Sidebar: What are systems archetypes?

The systems thinking process involves building new collaborative understandings of the interplay of forces at work. To accomplish this, participants use “archetypes:” images of common systemic situations. Each of these patterns occurs in a wide variety of domains, from ecology to economics to manufacturing; each offers its own strategic insights, and gives people a better picture of how the forces of the system may trap them.

System dynamics researchers have published descriptions of about a dozen archetypes. They include “Limits to Growth,” in which a seemingly boundless growth pattern runs up against unexpected limiting forces. (Total quality campaigns, for example, run up against institutional disappointment after the “low-hanging fruit” is picked.) In another archetype, “Shifting the Burden,” a more immediately inviting, short-term solution to a problem weakens the system’s ability to develop a more fundamental, but slower, approach.

Another archetype, the “Tragedy of the Commons,” became the basis for the Epsilon team’s “Tragedy of the Power Supply” (described later).

Compared to computer models of systems, archetypes are simplistic; they have been compared to “training wheels.” But in well-designed group workshops, they can lead to very sophisticated collective understandings of common problems. Because archetypes imply counter-intuitive, but effective, alternatives for action, they are generally very useful strategic tools.

See Peter Senge, *The Fifth Discipline* (1990; Doubleday), p. 93-113 and 378-390; Daniel H. Kim, *Systems Archetypes: Diagnosing Systemic Issues and Designing High-Leverage Interventions* (1993, Pegasus Communications); and Senge et al, *The Fifth Discipline Fieldbook* (1994, Doubleday), pages 121-150.

on these pages). Their ability to communicate improved, and they began to focus in on their most serious problems.

Sidebar: What are the “ladder of inference” and “left-hand column”?

These “action science” devices are designed to build skills of “reflection and inquiry:” ways of holding conversation that lead to greater understanding of both process and content. The tools are simple exercises and metaphors that help people unlearn their own defensive, counter-productive conversational habits. For example, in many work situations it’s more effective to systematically inquire into why other people feel the way they do, instead of trying to hammer your own point home as dramatically as possible.

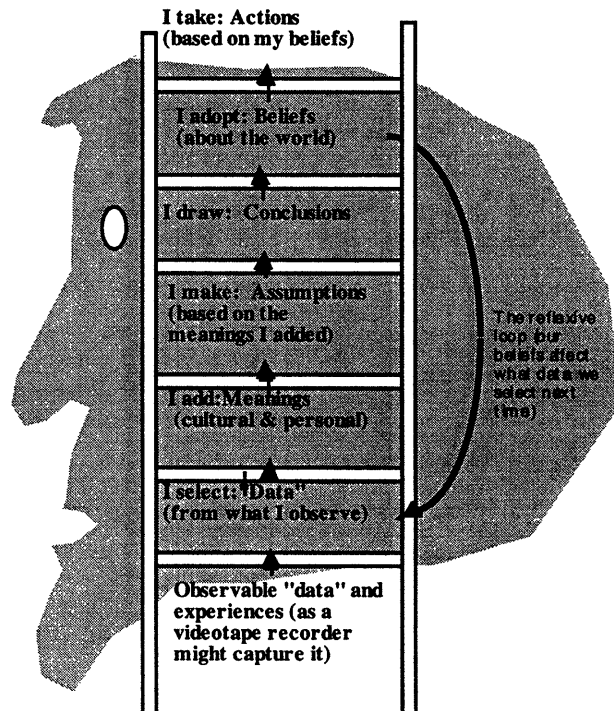
The “ladder of inference” — a term coined by Professor Chris

Argyris — is a metaphor that shows how rapidly we can leap to knee-jerk conclusions with little data and no intermediate thought process, as if rapidly climbing up a ladder in our minds. You start at the bottom with the observable data, which is so self-evident that it would show up on a videotape recorder (Larry has yawned at a meeting), and within the space of a few seconds, leap up to assumptions (Larry is bored), to more generic conclusions (Larry doesn’t care about this project). Since most of these conclusions are never discussed openly, there is no way to check them.

Thus, incorporating the “ladder”

into everyday conversation has proven to be a pivotal component of learning organization work. It gives people a safe way to raise and check their varied interpretations of events.

In the left-hand column exercise, people select a difficult situation and reconstruct a pivotal conversation. In the right-hand column, they write down what was said. In the left, they articulate what they were thinking and feeling, but not saying. The case becomes an artifact through which people can examine their own thinking, as well as the systemic problems which underlie the impasse.



The ladder of inference is described in Chris Argyris, *Overcoming Organizational Defenses*, (1990, Prentice Hall, p. 88-89); Argyris, Putnam, and Diana McLain Smith, *Action Science*, (1985, Jossey-Bass, pp. 57-58). Also see *The Fifth Discipline Fieldbook*, page 242.

The “left-hand column” exercise is based upon the two-column method developed by Chris Argyris and Donald A. Schön. The research method was first presented in their book *Theory in Practice* (1974, Jossey-Bass). Also see *The Fifth Discipline*, page 195, and *The Fifth Discipline Fieldbook*, page 246.

To give them a broader perspective, several members of the core team interviewed a dozen people from the rest of the Epsilon project, asking: What did people feel their greatest challenges and strengths to be? The team returned for a working session in

March of 1992, ready to pick two or three problems and start fixing them immediately.

The lead MIT Researcher reported his impressions of the team: bright people, burdened with frustration and puzzlement but not resignation. The interpersonal dynamics typical in a product development environment were evident in this program.

The comments of the researcher were typical of the ways managers from different functional areas talked about and reacted to one another in AutoCo.

Researcher: The participants had locked horns and couldn't get anywhere. [The Launch Manager] told me much later in the process that [the Program Manager] was ready to hang it up after a few months. But he saw enough value to keep him going. In one critical incident, early in the sessions, we used a ladder of inference and left-hand/right-hand column exercise to begin to look at the ways in which finance and program management couldn't communicate.

Using tools to understand one another's positions allowed people from different functional backgrounds to work together more productively.

The exercises opened things up so that they could really look at their assumptions. Program management revealed how they assumed that the finance people "want to hold us to [American automobile] costs." Essentially, program management thought that finance didn't have a clue (and didn't care) about what it took to build a car. "They just want to meet the numbers."

The Learning Organization work assumed that openness of communications, and the willingness to address assumptions, would lead to workplace efficiencies and effectiveness. How valid is this assumption?

The finance people showed their assumptions about the program managers: "These program guys don't care about costs. They don't want to work to get to the kind of car that we said we were going to do. They won't control costs."

If people could see issues from one another's perspectives would it allow them to work better together?

I said, "Look, we're here to do something different from the way you continually do things." Instead of having a normal "let's-solve-it" style consulting project, I suggested that we "try to understand our own assessments of the problem. We are blind to the limitations of our assessments, and thus we can never see what is really happening. Yet we design solutions based on our own individual assessments." The core team members understood that argument, and went along with another round of

A sound logical argument was necessary to convince the team to apply the tools in the service of improving their abilities

to work together.

Tools from Total Quality Management were used. A learning tool is one which people share with one another to work more effectively together. Learning tools are spread freely to create shared insights, not used by one person to gain an understanding which he or she uses to manipulate or change others.

See sidebar, next page. The "KJ diagram" combines the intellectual-rational thought process with intuitive-emotional feeling data. It is generally used to process a group's own thoughts and observations; here, they used it to make sense of their interviews.

The K.J. diagram incorporates multiple people and multiple perspectives and appeared to have brought to the surface an interrelated and interdependent set of problems.

The researcher focused on problem articulation rather than problem solution (see Sidebar: Problem solutions, versus problem articulation, page 14).

Are these three key issues important on any product launch? Are they endemic to the product development process? Do they stem from AutoCo culture? Are they worth taking time to deal with?

experimentation.

I had them break up into two groups (because there were so many of us) and each group did a "K.J." We used the technique to sort through the information gathered from interviewing Epsilon team members. "Listen to what you think people are trying to say," I said. "Get away from thinking only rationally about the comments. How do they feel to you?" We went through a "scrubbing" process [rewriting the statements to be clear], then grouped them intuitively. This is usually a very frustrating and bewildering process, especially for an engineering or action-oriented group, because everything is interconnected and diffuse.

Our theme question was, "What is the biggest weakness with our product development process?" And the overarching answer that emerged had to do with lack of trust and openness. This was unspoken before this point, and I think it wouldn't have been captured otherwise.

Now they couldn't accuse me of "making them" pay attention to this trust issue. They had seen firsthand, how it was at the core of their own problems.

This incident made a lasting impression on [the Program Manager], and changed the dynamics of the relationship with Finance, which was very important.

Program Manager: There were really only a few core issues. The rest of the problems all generated from those.

- Fear and consequences of being wrong led to people not sharing information;
- The boss' need to control came at the expense of drawing forth individual capabilities on the team;
- Other people weren't trusted to help you; they tended to one-up whatever you did.

By this point, we had been working together eight months. We had learned to generate trust in our own core team, so we could look at these issues and agree: "Yeah that's really what's going on."

Sidebar: What are "affinity diagrams" (K.J.'s)?

Affinity diagrams ("K.J.'s"), named for their inventor Jiro Kawakita, emerged from the quality movement in Japan. A team of 5-10 people considers a mass of issues, posts statements related to a main question on a wall, and repeatedly groups and rephrases them.

Over the course of several hours, a final pattern emerges: a

coherent set of themes that reveal key underlying issues.

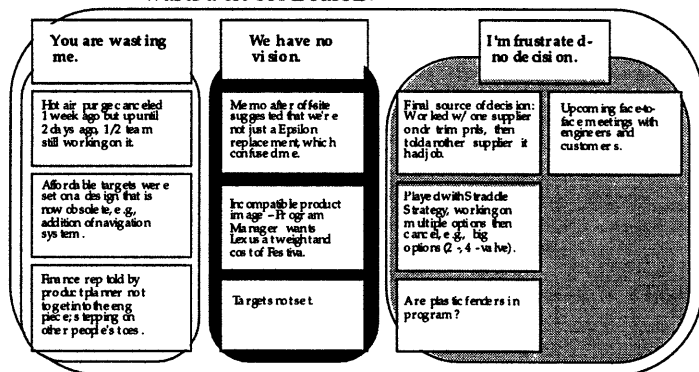
This illustration shows half of the Epsilon team's final K.J. arrangement. Each one of the groupings represents a key theme; the arrows show how (the team felt) one theme influenced another.

The diagram may look carefully considered and well-organized,

but this final snapshot does not show the hours of "messy," unfocused, frustrating deliberations that went into it. First by conducting interviews, and then by using the "K.J." process to sort and analyze that data, the core team discovered a critical issue: the extent to which engineers trusted managers and felt free to speak openly.

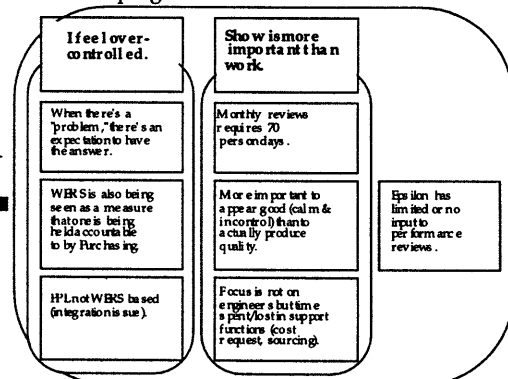
What are the weaknesses with the product development process in Epsilon?

We don't know what we're doing so we waste a lot of resources.

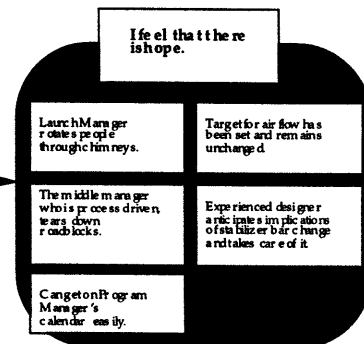
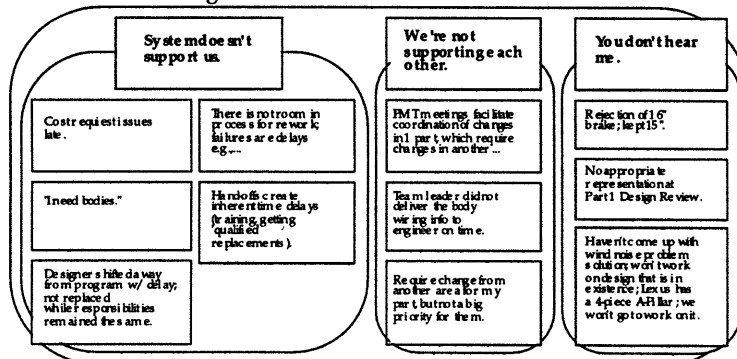


Management does not consider people or human resources an asset.

Management requires reports as a way of keeping control.



Management and engineers don't know how to work together.



March 1992

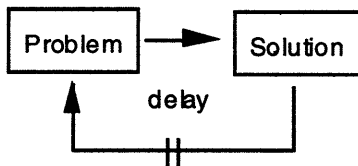
Sidebar: Problem solutions, versus problem articulation

How do most people solve problems in business settings? When a problem is identified — something is noticed as wrong — action is required. Do something, fix it! Under conditions which are often referred to as “fire, ready, aim,” how often are managers able to, or even expected to, think in detailed ways? Detailed thinking includes considering multiple cause and affect relationships as well as unintended consequences. But managers have neither time nor training to think about problems in this way. Thus, solutions are often developed and implemented before the problem is understood.

Nonetheless, problems and solutions are integrally interconnected. Decision-making researchers March and Olsen (“A Garbage Can Model of Organizational Choice” in *Administrative Sciences Quarterly*, 1972 and “Garbage Can Models of Decision Making in Organizations” in *Ambiguity and Command*, Pitman Publishing, Inc. 1986) have established the existence of this connection at individual and organizational levels.



For example, the identification of potential solutions often leads to an increased awareness of problems. Since there is a high likelihood of an inappropriate fit between problem and solution, new unintended and unanticipated consequences often erupt, in part based on the mismatch between solution and problem.

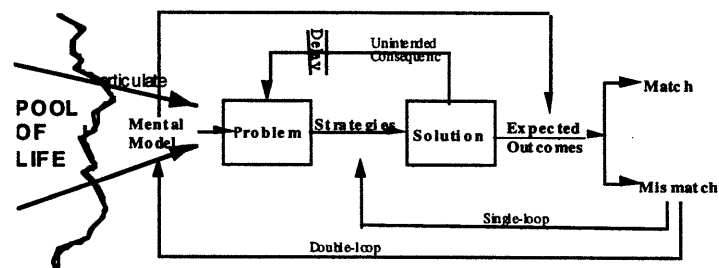


A “problem solving treadmill” is created by problem solving in this manner — solving one problem will lead, in time, to new problems. Technical types of problems, working with physical machines, for example, are situations where this type of rational problem solving works well. As systems get more complex, however, delays between problem and solution mean that it is increasingly difficult to link actions and their consequences.

The “problem of problem-solving” is exacerbated by the fact that a problem’s identification depends, at least in the short term, upon a person’s position and perspective. For example, when oil prices rose dramatically in the mid and late ‘70’s, this was a dramatic “problem” for consumers of oil and industries dependent upon fossil fuels. But it represented a windfall for others — Middle Eastern nations, oil producers, alternative energy producers, and even environmentalists (because it led to energy-efficiency research). Defining something as a problem is an interpretation of reality.

Thus, the way in which problems are framed limits solutions. Consider commonly heard statements: The problem is... “we need a better management information system,” “the reward system needs to be revamped,” “our costs are too high,” “our management philosophy is outdated,” or “we need a learning orientation.” These “problem statements” limit the solutions considered. More often than not, the known solutions are applied to the perceived problems. And, when new problems emerge related in part to previous solutions, people fail to make the connection. The problem solving treadmill leads to a condition which some define as “insanity” — doing the same thing over and over and expecting different results!

Systems thinking provides an alternative to the problem solving treadmill. By considering problems and solutions as linked, solutions are also evaluated on the basis of potential unintended consequences. Rather than moving to action, the systems thinking approach focuses on the assumptions, or mental models, from which the problems were articulated in the first place.



For more information in the “problem solving treadmill” concepts see Daniel Kim’s articles, “Using ‘Fixes that Fail’ to get off the problem-solving treadmill” and “Fixes that Fail: Oiling the Squeaky Wheel—Again and Again...” in the November 1990 and September 1992 editions of *The Systems Thinker*, Pegasus Communications, Cambridge, Ma.)

The system map (Learning Team Meeting, August, 1992)

Now the “core learning team” engaged in a “systems mapping” process, to connect the concerns of trust and openness that had surfaced in the K.J. diagram with other symptoms in product development, such as the chronic critical problem of late parts.

See the next page for the core team's view of “parts behind schedule.”

Focusing on one key problem - in this case lateness of parts - again showed the interrelatedness and interdependencies of all the problems facing the team.

The K.J. and causal loop diagrams appear to have not only influenced how people thought about the problems, but also their perception of problem symptoms.

How can a team effectively build a group understanding of a systemic issue, so all members “own” that understanding?

This story of the finance manager's role was often repeated to illustrate how the system dynamics work is valuable to a cross-functional team. Finance people typically don't get involved in engineering change processes; here, that involvement helped build a deeper understanding of systemic issues.

The researchers proposed that a shared image of the system would allow people to operate in a coordinated fashion.

Program Manager: We weren't sure what to do next. We didn't know how to “change” fear or mistrust. So we started to look at a key problem: Why we were always late. No matter what we did, no matter how we approached the problem, parts were chronically behind schedule. We began to share our views about why. Over the course of a day, we built up a diagram of the system.

When one engineer changes parts, that part usually affects somebody else. The “A” engineers can't start part of their work until the “B” engineers solve their problem. Parts get late. This leads to pressure to get back on schedule, so we compress the supplier time. But then something else would become late, because we would be putting all our resources on the part that was late. When parts get so late that we can't recover, we revise the build schedule. Then people feel they have more time, so more parts get late. Worse still, the next time you have a build, people will assume you're just going to revise the schedule again, so they don't even try to meet the dates.

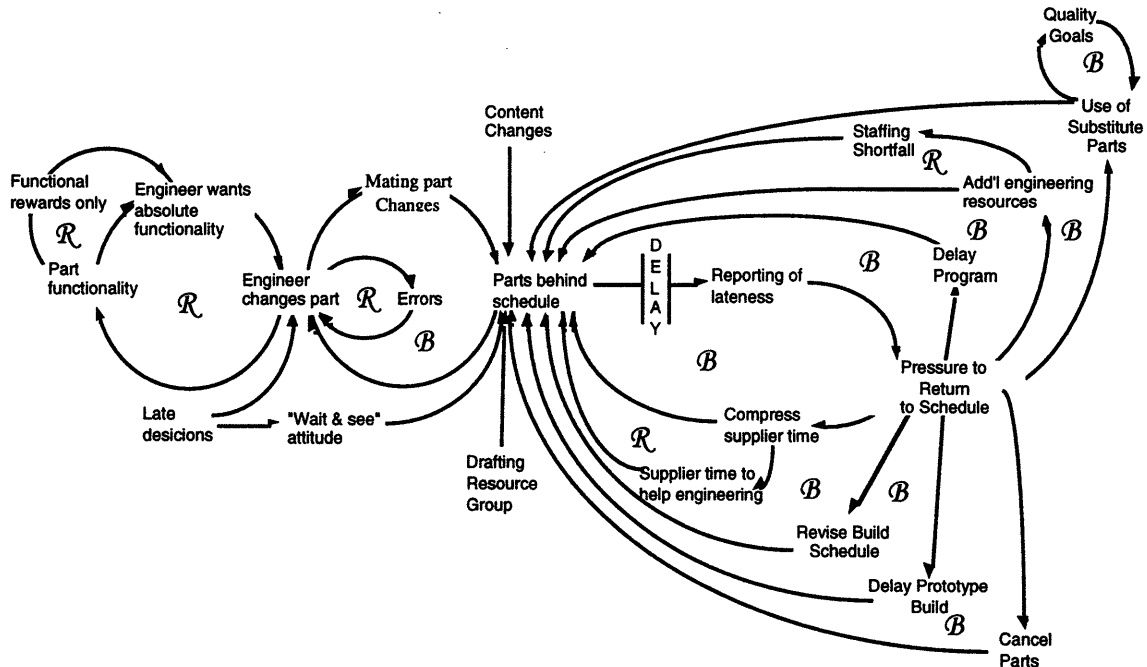
We eventually got all this into a complex chart. We all understood the whole system as it related to us, and we had all contributed to this map.

The map became critical. And the person who pointed the key leverage point wasn't an engineer, a development manager, or a planning manager. It was the finance manager. She pointed out that just before the “reporting of lateness,” there's usually a delay. The reason for the delay is: people are afraid to be criticized. There is a basic cultural commandment in engineering – don't tell someone you have a problem unless you have the solution. You're supposed

The hierarchical and autocratic management culture at AutoCo reinforced the engineering culture to limit discussions of problems until there were known solutions.

to solve it – and then tell them. But during that delay, nobody knows about the problem, and nobody can react. That delay automatically compounds delays in other loops going on through the system.

Sidebar: The core team's view of "parts behind schedule"



This picture is a simplified version of a causal loop diagram produced by the Epsilon core team. The key problem is the central concern, "parts behind schedule." At the left are a series of forces (many involving exponential growth) that contribute to increasing numbers of part changes and late decisions. At the right are many of the "fixes" that the conventional system uses to "solve" the parts behind schedule problem, and the unintended consequences of those fixes (staffing shortfalls, the taking up of supplier time to help engineering) that actually make the problem worse. Adapted from "A Framework and Methodology for Linking Individual and Organizational Learning Applications in TQM and Product Development," by Daniel H. Kim (1993, MIT Ph.D. dissertation, p. 282-296).

The systems map was described by most people as enormously useful. When a group "sees the same picture, and comes to the same conclusion," is it the visual (like the system map) that helps them focus on issues systematically? Or is it the conceptual link?

Program Manager: We all saw the same picture, and we all came to the same conclusion: This was a leverage point for us. We began to look at how we could structure the project to ease that delay, and we concluded that our real leverage was in improving the communication process, improving honesty, and improving trust.

We had begun to build these in the core team, but they didn't exist between the functional groups in the rest of the team. But we knew that building trust was possible, because, after all, if we hadn't shared our views of the system honestly, we would

never have gotten to this point.

Another factor accelerating the decision to promote open communication was the increasing complexity of automotive technology, particularly with fast-changing subsystems such as the electrical system. The better the team became at working together, the better it would deal with intricate components which cut across two or more functional roles.

Program Manager: Only a small [percentage] of the members of the team actually worked for me, where I could promote them and give them their performance reviews. Most of them worked for other functional organizations – finance, assembly, body engineering, climate control, plastics. If I just pulled the team together the normal way I would get what we always get. People would protect the objectives of their organization. They would work on the product, but their organization would come first. My greatest leverage was to make them view the other members of the team as people with whom they had a personal relationship. If I could make them feel that they depended on each other, and that they wanted to help each other as people, that would counterbalance the material reward from their functional organization.

Was the statement that “we don’t know how to make open communication work” a barrier which kept people from believing in the possibility of open communication?

We had been talking about open, honest communication around this company for as long as I’ve been here and I’ve been here 29 years now. We don’t know how to do it. This was the first time I thought it might really work.

Sidebar: “Content leaders” and “process leaders”?

On the Epsilon team, work groups were organized around subsystems of the vehicle — electronic, interior/trim, powertrain, etc. Many of these subsystem development teams were assigned two managers: a “process leader,” whose

attention focused on planning and coordination issues, and a “content leader,” whose attention focused on engineering and technical concerns.

In this learning history, when the transcript refers to a “content leader” or “process leader,” it means one of these subsystem team leaders. The learning history does not identify the particular subsystem in which a “content” or “process” leader worked.

The transition to openness

Converting to an atmosphere of honesty and trust was an ongoing process, with many setbacks, continuing throughout the forty-two month program.

A premise of the learning organization is that the boss isn’t the one with the

Launch Manager: In one of our earliest design reviews in February, 1991, folks were standing around saying,

answers. Bosses are teachers and coaches who help others learn and apply their learning.

What model of teachers are they using? Teachers of children? Or, teachers of adults?

Manager's prerogative is typically that of being the decision-maker. Making those decisions was thought of as creating a codependency between manager and employee — employees look to managers for decisions and managers expect they make the decisions.

To the Program Manager and the Launch Manager, "empowerment" meant the recognition that engineers (and others on the team) needed to make decisions and feel responsible for those decisions.

The launch manager saw the group's behavior as "dysfunctional." But how "functional" was it within the context of the larger AutoCo culture?

Involving people in making decisions implies that leaders make their reasoning process explicit or, as in this case, justify it as "reasonable."

The behaviors associated with letting people make their own decisions can appear messy to outsiders - not the calm where a boss calls the shots and all the action occurs below the surface.

Why does there have to be a level of

"What do you want us to do?"

I said, "Wait a second. You're the engineer. What do you think is the right thing to do for the customer and for the company? What is your recommendation?"

They couldn't answer that question. I realized then that this team needed a lot of help. They were accustomed to putting all the alternatives together and the planning manager or Program Manager would decide. I knew I had the prerogative, based on my experience and knowledge, to tell people what should be done. But I told them I would reserve that prerogative for when I was willing to challenge them. I didn't want to make decisions that they should be making.

I realized over a period of time, that AutoCo wanted to "empower" its teams, but on this team, people didn't feel empowered to do anything. That was our first realization that this was a dysfunctional group, probably very typical of most groups.

Content leader: Every once in a while we bosses have to grit our teeth and swallow a decision that the team made, even if we personally think, "Well I'd rather not go on that path." If their thinking is reasonable enough, and they haven't missed any obvious issues, then my objection is just a matter of: "I would go the other way."

Launch Manager: A new project manager, after sitting on a very typical discussion, ran up to one of my team leaders and said, "My God, your program is a disaster. You're not going to make it."

The team leader said, "No, this is the way we always talk. Everything that we know is thrown on the table. We thrash about, and issues sound a lot worse than they really are. This is our way of sharing what's going on."

The project manager said, "You've got to be kidding me. If anybody had

disaster before people will talk candidly?

said anything remotely like that on [my previous program], he would've been killed. Problems would have to reach a level of disaster before people would ever talk candidly."

Engineer: Every day someone comes in and says, "That was a really stupid thing you said in that meeting the other day. We all talked about it and we think this..." I may say, "Yeah, you guys are right," or maybe I don't agree. But at least nobody has any problem coming in and saying, "I don't agree with what you said," or "Other people think this," or "You didn't think about that." And it helps.

These comments are typical of the remarks made by engineers about trust and openness; they were gathered throughout the stages of the car project.

Engineer: When an engineer goes on vacation we cover for him. We all work as one, which I think would be pretty unusual in most other jobs."

Engineer: You don't hear dialog like, "People from body engineering are dragging their feet," or "Those hard noses from Assembly." We got that out of the way up front. Now we can get together and do the job.

Creating the atmosphere of trust and cooperation

Based on interview data, the following factors appear to be important in creating an atmosphere which leads to trust and cooperation. These factors related to behaviors which the core team sought to practice throughout the program. Each point is described and illustrated in a section which follows.

- A mandate that "bearing bad tidings" would be safe;
- Ongoing sharing of information and perspective;
- A culture of greater inclusiveness;
- Deliberate encouragement of informality and friendship;
- A mindset that "no one has all the answers."

Other factors described elsewhere in the learning history also seem important:

- Leaders who modeled the desired behavior (see page 24);
- The conversational tools of the learning lab (see page 35);

- Collocation* (see page 63).

- A mandate that "bearing bad tidings" would be safe

The Epsilon leaders had to start by mandating that there would be no punishment for candor, bringing in bad news, or raising uncomfortable questions.

How does a team effectively change its culture from one that expects everyone to "have the answers" to one that encourages people to experiment and raise issues before the answers are certain?

Content leader: If people know that they're not going to get punished (for taking risks), they'll try harder. We had all kinds of ideas coming out of the woodwork that people had been keeping in their hip pockets. The net we use for umbrella storage was an idea of one of the engineers.

This comment was one of many illustrating how engineers came up with and contributed new ideas because they felt they would be heard.

Engineers [on other projects] don't generate ideas like that; they wait for you to tell them what to do, so if it screws up they can say it wasn't their idea.

Does admitting that as the boss you don't know the answers help? If the boss doesn't know an answer, how would that evaluation affect the engineers?

Team leader: [The Program Manager] always gave us a strong feeling that he wouldn't treat you like you were an idiot if you didn't know the answer. Or if you came to him for help, he would try to help.

- Ongoing sharing of information and perspective

Taking the time to share information from different aspects of the development process with a broad range of people appears to have created greater awareness of program issues as a whole. Efforts were made for people to become aware of everyone's role in light of the overall program goals.

One of the questions the managers considered was whether taking the time to share information broadly could be made up by people working more effectively together?

Finance Manager: We had a better understanding of why each of us does what we do. This understanding was helpful in the long term, because it gave us a base of trust and understanding. I think it started to reduce people's sensitivities a little bit. You understand why something happened. If it wasn't necessarily what you liked to happen, at least you understood why.

Could managers sharing more openly

Internal consultant: When a supplier has a good relationship with the engineers, the supplier will say things like [for example], "You didn't hear it from me, but something is

* "Collocation" means that people from the diverse engineering functions developing the car (i.e. chassis design, air conditioning, suspension, alternator, sound systems, dashboard subsystems, etc.) work from offices in the same location. Team members are physically placed together, during the time it takes to design the car, instead of coming together only for formally scheduled meetings.

result in others sharing their information more openly, countering a larger cultural tendency to hide problems?

going to be late and somebody's lying to you. Don't tell anyone I told you."

- *A culture of greater inclusiveness*

Including people, not just on the basis of who needs to know to do their job, appears to have created an atmosphere where people were more aware of the whole of what they were trying to achieve.

Team work was a much talked about concept at AutoCo. However, there were no common standards or definitions of team work, and it was often managers promoting their management style that led to contests over who had created effective teams.

Content leader: I think if you talk to any [Epsilon team member], they'll say, "Oh yeah, we've got fantastic team work." Well, every program's got team work.

On this program, [the Program Manager] provided a strong network for team work. He allowed it to happen easily. It's more than just collocation or expertise.

For instance, take the EP prototype build; chassis engineering was given credit for having 100% of their parts there, and that was the first time something like that has been accomplished. And that says a lot about the strength of the engineering community as well as others who helped with that endeavor. It's like every day you're putting on your Epsilon suit and you're going in there and you're going to make a difference.

Are new channels of communication, such as including suppliers in internal meetings, sufficient in themselves?

[The Program Manager] made the program very, very inclusive. He was adamant about having suppliers in there for [internal prototype meetings]. The meetings up front were pretty boring but Epsilon seemed to achieve their success because of the meetings.

Could the time it took to broadly disseminate information to a large group of people be justified?

- *Deliberate encouragement of informality and friendship*

Some attributed the quality of the Epsilon team process (at least through first production prototype [1PP]) to collocation. Some attributed it to the people who had been chosen for the team, and the quality of the engineers. Others argued that the team appointment process was random; indeed, Epsilon probably didn't get the "best" engineers, who would have gone to a higher-visibility product like Theta² or

² "Theta" is a code name for a highly successful and best-selling vehicle which AutoCo produced in the mid 1980's. The success of Theta is credited with putting AutoCo in a financially solvent position and allowed investment in other vehicle development programs. In order to achieve its success, the Theta program "broke" a number of product development rules. Although not always judged as successful in its product development efforts (milestones, timing, quality, warranty and so forth), Theta was highly successful in the marketplace. The team that "went all out" to get Theta to market was not highly acknowledged within AutoCo for what was widely thought to be a heroic effort. A

Sigma. No matter the cause, there was an atmosphere of informality and friendliness which promoted good working relationships among team members. Managers invested in and sought to actively encourage the development of a climate which allowed people to work more informally with one another.

Off site meetings where people were in residence were typically frowned upon at AutoCo because of cost implications.

The time and effort the core team put into developing relationship and open communication was not valued by all team members.

How important is it to develop a focus on the personal as well as the professional relationship?

Was knowing people beyond their role something that would help team members through more challenging and ambiguous situations?

Engineer: There was a three-day off-site [Oct. 1991], where you basically had to walk away from your desk, designs and drawings. A lot of the team thought it was a waste of time; I think a lot of people walked away still not exactly sure what [the Launch Manager] and [the Program Manager] were trying to do. To me, the most critical thing that happened there was it really glued the team together. It was the first time everybody looked around and really realized they were all in this boat together, and it was powerful.

Some of the dynamics and camaraderie and friendship came from living together at the hotel for three days, eating together, working together, and going and having a beer together. There's something about friendship outside of the work place that I think really helps build trust. If you met new people and had a beer with them or whatever, then two months later if you had a problem with an issue on the job you felt a little bit more comfortable approaching them. And I think it's human nature that if you get an opportunity to know a little bit more about a person's family and their interests outside of work, you trust them more than a total stranger. At the off-site we started talking about visions, and what the car was going to be, and everybody's roles and I think it created the bond. It was a reality check: This group of people was going to build this car and either make it a success or not.

• *A mindset that "no one has all the answers"*

The managers told people that they didn't know. They needed to learn too, and they needed to create the conditions where their teams could learn. This statement supporting "not knowing," ran counter to the expectations most managers and engineers held — of one another, their system, and themselves.

number of the managers on the Epsilon program, including the Program Manager, had previously worked on the Theta program.

How can business culture learn to acknowledge and accept mistakes in a constructive way?

Vehicle development team leader: This was foreign to a certain degree; it was uncomfortable that the boss was learning with us. He didn't have the answers. We were going to do this together. As uncomfortable as it was on one hand, it was exciting on the other, because all of a sudden we felt like we were paving new ground. We weren't just being fed "the right thing to do in a corporate environment today."

Some of the appeal may have to do with the word "failure." If you stumbled, it wasn't perceived by [the Program Manager] as a failure. Instead, the reaction was: "It was a good honest effort, I appreciate your energies, and what have we learned from this?"

Behavioral versus technical: a zero-sum game?

As the Epsilon program developed, the core team placed strong emphasis on what they called "process" issues—explicitly learning how to improve human communication skills, decision making methods, and understanding of each others' points of view. This emphasis on behavioral considerations—people and process—was neither familiar nor well understood in AutoCo, which has a heritage of strong technical and engineering achievement. In the larger company culture, most people felt that time was scarce and thus needed to be invested, as much as possible, in technology and engineering.

These thoughts about time were based on an implicit assumption that behavioral and technical skills were caught in a zero-sum game; one could only be improved at the expense of the other. In contrast, the Epsilon team leaders believed that a well-designed learning process would result in engineering and people skills reinforcing each other. Technological capabilities could be more easily applied, and have greater impact, as engineers' people skills improved and they developed the capacity to handle systemic and interrelated issues.

Does the AutoCo culture hold an assumption that developing process knowledge and understanding among people detracts from engineering knowledge and technical achievement?

Vice President: The first thing we have to do is make sure that people really understand the engineering processes – DVP&R, QOS, prototype process, change control, the timing discipline, and so forth. If you asked me, "Do I do that before I do any teamwork and team building?" – the answer would be an unequivocal yes. I think if you don't have a fundamentally strong engineering framework, then no amount of team building or team process will ever

The comments here were echoed by many people at AutoCo: "First handle the technical and then, if you have time left, you can work on the behavioral."

Do "process people" sometimes assume that their work is denigrated by the larger culture, when in fact it is not?

Process skills are difficult to measure, and their impacts are not easily assessed from the outside. How, then, can process work be evaluated? An implicit assumption, which came out clearly in other interviews, was that if you can't measure it, you can't manage it.

Do efforts in behavioral skills detract from "getting this car program implemented?"

Is there an assumption that resolving the tradeoff between technical and behavioral needs would extend a manager's work beyond the hours available in a day?

overcome that weak system. You'll fail. The team will thrash about with all kinds of good intentions.

Now, the converse, I guess, is – assuming you have a fundamentally solid engineering process – that as you encounter difficulties or fall behind in one area, team building and the human side of teamwork come into play. A team can rally, help each other overcome difficulties and move on.

As I understand organizational learning, it preaches an appreciation for the other person's point of view. If you, as a team person, understand that your action or non-action causes another person a problem, then you will try hard to make that not happen. That understanding comes from team building experiences. The team is a very large family that's working together. How you measure that understanding, I don't know.

Program Manager's Boss: I could tell that the Vice President was really interested in getting this car program implemented. I think there were times when he was concerned that the team was spending too much time on the soft stuff and that could get in the way of hard results.

I have been very supportive of this work, but at the same time continued to stress to [the Program Manager] that he had to get results.

The dilemma of integrating process and engineering knowledge

Despite the team leaders' intentions to have process knowledge and engineering knowledge reinforce each other, the synthesis was imperfect in practice. In at least one case, there were complaints that the engineering knowledge was short-changed. This had to do with the team's change of structure — bringing engineering and process authority together, assigned to one group of people — without looking closely at the assumptions that these people would hold about one another.

Content leader: But when it came to dealing with some of the particularly difficult engineering issues, we were left out in left field, and we had no one to go to. I don't think it was

Not all engineers felt included. The fact that Launch and Program Managers were not engineers meant that they didn't jump into difficult engineering problems and participate in resolving them. Instead, they put them back on the team. As the comment shows, some engineers did not feel free to raise this problem.

[the Program Manager's] problem to get involved with these issues, because he doesn't know about engineering. But we should have been able to discuss the problem [that engineers felt unsupported.] I think after a while we recognized it and we avoided [discussing it].

Bringing the process leaders and the content leaders together on a team, it was thought, would allow them to recognize one another's blind spots and develop an appreciation for how the other side saw things. This, however, did not always happen.

The contrast between process knowledge and engineering knowledge was prominent within the Epsilon team as well as in the larger organization. People seemed to be experts at one or the other, but not both. Communication was not strong between the two groups of experts and some issues may have been overlooked.

Content leader: There were some rough roads on this program. There used to be a conventional breakdown of car program management. You'd have separate managers for business planning, vehicle engineering, launch, and vehicle development.

On Epsilon, they tried to combine all those skills into one person. This was the first time they tried that management system and the way they implemented it didn't work very well. It worked in fits and starts.

The real engineers all ended up reporting to the Chief Engineer, and the planning types ended up reporting to [the Launch Manager]. [The Launch Manager] is not an engineer by background. The planning people worked with the Design Center effectively, and knew how to talk about the features of the car, but they didn't have the background to say, "OK, now we need A and B completed by this time and we always have problems with B, so let's concentrate our efforts there."

How can we design learning so that technical and process knowledge reinforce each other?

Could members of both groups be empowered simultaneously?

We found out after a three-month delay, for example, that some engineering issues affecting the audio system were not being elevated to the Chief Engineer's level of discussion. Some of the vehicle leaders didn't seem to know enough to bring up the right questions. They should have had much more deliberate communication between the groups from the beginning.

Theme 2: Setting an example of non-authoritarian leadership

The majority of Epsilon team members commented that the Launch Manager and the Program Manager not only said they believed in openness and honesty, but acted in ways that showed they believed in its value. Their behaviors were striking because they contrasted with the way “typical” program managers behaved, and the ways these managers themselves had behaved in the past.

Leadership roles: modeling new behavior

People throughout the Epsilon team, when asked what factors provided them with the greatest opportunity to learn and change, singled out the examples set by the Program Managers’ “walking their talk.”

Content leader: I’ve known [the Program Manager] for 25 years. He was a typical senior program director when he came aboard – very autocratic and power-based, and always had been. But I’ve seen [the Program Manager] do a 180-degree turn in the last two years. It wasn’t sudden; there were a lot of very subtle bends in the road over time. But he has gone through a complete change in management style. [The Launch Manager] deserves a lot of the credit for turning [the Program Manager] around, but as [the Program Manager] began to see the value in the learning labs, he became more supportive and he initiated a lot of the new approach.

Many people commented on the changes the Program Manager had undergone. Those who had known him for years, or even decades, often made remarks about his changed behavior. He had changed from having a punishing, frustrated, angry style to being relaxed, approachable, and willing to ask questions, listen, and learn. This made everyone more willing to raise questions and problems — not just directly with the Program Manager, but with the entire team

I enjoyed it because it was the first time in 30 years at AutoCo that I, as an individual, felt valued by management. I felt that they had an absolute trust in me and in the team – not initially, not midway through the program, but by the end of the program. They provided the vision and they gave us the guidance from a total car point of view, but they put absolute trust in us. They would ask, “What do you guys think is the best for the car? Here is your objective. How do you think you’ll do it? Tell me.”

I think we responded very

The managers created an atmosphere which helped people feel safe and valued. They worried, however, if that style was viable in the larger organization.

positively. Because I had trust from them, I put a lot of trust in my team. On other programs, I was constantly double-checking and telling people what to do – not asking them, “What do you think we should do?” I think we felt the weight of the responsibility we were given, and we tried to do a much better job because of that. I think we succeeded. It’s enthused a lot of people who had not been enthused at AutoCo for 20 years. And it depended on the program leader’s managing style.

To be honest with you, I’m spoiled. I’m worried. Who am I going to work for next, and can I adapt to that old AutoCo style again?

Comments by senior managers become part of a team’s collective “mythology.” Three people, for example, mentioned a meeting over a microphone package, in which the Launch Manager confronted an engineer in a way that people weren’t used to, especially in front of other AutoCo managers. After a pause, the Launch Manager explained the reasoning behind his disappointment and inquired into the engineer’s circumstances.

The Launch manager was honest and direct.

Launch Manager (at the microphone package meeting): You know, I am very disappointed with what you’ve done. I just want you to know that. I don’t understand why you did it. I need you to help me understand and I think we all need to understand what our roles are here.

Engineer, an attendee at the meeting: I had never seen that honesty and openness from a man at his level before.

Engineers knew they were trusted because they were not required to complete the paper trail that documented who did what and when. This approach saved time and freed people to concentrate on the tasks at hand.

Another Engineer: There’s so little paper work on this team compared to other teams. We didn’t have to prove everything to the last detail to [the Program Manager]. Other program managers spend so much time on studies.

Say, for example, that a program manager wanted you to put the speedometer in the A pillar. An engineer would say, “I don’t know how to tell you, but you can’t do it.”

"I don't care," the manager would say. "I want you to come back and tell me why." So you go out and do all these wiring studies. You pull a designer in to get the drawings done. You talk to your division manager. You get a nice document pulled together, showing every tic mark why you can't do this, and you call everyone together into his office and run through all the points. And he would say, "OK," as you've known all along he would, because the answer is implicit. But [the Program Manager] didn't have to do that.

We didn't make blind recommendations. The team did their homework every time. But we did not have to spend hours at the computer getting our reports to look just perfect.

The behaviors of program managers extended to people outside AutoCo. Suppliers, vital producers of the parts that make up the new cars, also commented on the treatment they noticed from Epsilon program managers.

Epsilon product manager at a key supplier: Here's the famous quote from [the Program Manager]. We were sitting in a meeting one time looking at this floor console and he was playing with some of the features on it. He looked at us and said: "You guys really do good work." Then he played with the floor console some more and said, "You guys are really expensive."

He didn't yell at the cost. He made a statement about value versus cost. He liked the floor console, but he thought it cost a lot. Certainly, cost is important; but others at AutoCo would have said, "You've got to cut the price," without regard to the value being delivered to the customer.

I was walking down the hall here with a young engineer from [my company], and [the Program Manager] saw us and stopped to chat. "You know," he said, "I really like it when I see you guys walking through here like this. It lets me know that you think this program is important to you." He didn't have to say that. Tactically, I don't know the degree to which he meant it. But it had a positive effect. The young engineer

Even a small amount of positive reinforcement seemed to carry a lot of

weight with suppliers. Those interviewed for this learning history sometimes commented that they were "used to being beat up" elsewhere in the auto industry, and thus appreciated their treatment here.

went back to our supply company office and said, "You ought to hear [the Program Manager]. He said he likes the way we support the program." That got translated into a whole host of things that were all positive for our work on the car. And that was just a side comment going down the hall.

What made it possible for leaders to change their behavior effectively?

We examined this question, directly and indirectly, in the interviews. The responses clustered into several ways in which new behaviors were demonstrated.

The core team meetings.

These meetings, limited to the senior-most managers on the Epsilon team, became a concerted effort to improve communication among the Epsilon senior leadership. That improvement, in turn, trickled down in ways that were noticed by engineers on the program.

Team leader: Some people are natural-born team leaders. I don't think [the Program Manager] was one, but I think he has become a dynamic leader now. I don't think he would have changed without the core team learning effort.

How do people begin to unlearn behaviors that have rewarded them up until now?

Launch Manager: It took the bosses literally eight months to learn how to quit being bosses. When we started Epsilon, we knew all the answers. That's why we were bosses. At least one of us always knew that our answer to the problem was the right answer; and boy did we defend our positions! We had conflict, mistrust, gamesmanship: all the dysfunctional stuff you hear about.

Teachers struggle with getting students to unlearn the old to teach them something new. That is a struggle, even in an explicit learning environment. What is possible in a business setting?

But, as we began to use the tools, and practice with them, [we started to listen to the rest of the Epsilon team members]. Can you imagine the bosses listening? We quit telling them what to do. We started to inquire. We started to challenge their perceptions.

Is the team environment a necessary factor in changing individual behaviors?

Program management demonstrated commitment by taking part in every learning lab.

Managers did more than talk about their behaviors. They participated in learning events, exhibiting their own learning process and support to the team members.

Launch Manager: My fundamental role [in the learning lab] was not so much to teach a new insight as it was to participate with them in creating a psychological safety net... [To show them] I was willing to make mistakes with them and share my own frustrations, and that it was okay to discuss these issues.

Partnership: Having the two leaders learning in tandem provided a necessary support for one another.

Several people (including the Program Manager and Launch Manager) explicitly pointed out how important the two men were to each others' success. They needed each other for perspective as "mirrors" to each other. They also needed each other's advice and encouragement.

If there had only been one leader involved in this learning effort, would the same events have happened?

Launch Manager: Without [the Program Manager's] acceptance of the process changes; without [the Program Manager's] encouragement for us to learn; without [the Program Manager's] coaching in the difficult situations, this would not have been possible.

Purpose: the Program Manager and the Launch Manager never came into the project saying, "We want to become better people" for our own sake.

Throughout all the interviews, the Program Manager never said, "I set out to change my personality - to become the kind of guy of whom people say, 'He really listens.'" He points out that his intent, from the beginning, was better communication to help develop a common sense of purpose across functional chimneys. In retrospect, modeling new behavior was a means to that end.

Epsilon leaders faced a dilemma when they tried to explain their success to other AutoCo managers. On one hand, they were reluctant to emphasize their own personal change because that carried an implicit message that, "you [other AutoCo managers] must change your behavior as well." On the other hand, if they left out their personal change, a key element was lost.

Reflection: The leaders continued to increase their own capability to understand changes in themselves and others.

People noticed the efforts the managers were making to reflect on their behavior, learn from their experiences, and improve their own managerial capabilities over time.

Researcher: I saw the changes in [the Program Manager]. Earlier in the program, I saw how intimidating he could be, sitting in that chair and grilling people. He could be very accusing: "I will not accept that." And I could see why he acted that way; in the AutoCo culture that is the only way to have credibility, to get heard, and to get things done.

What kind of personal development is necessary to come to the realization that some behaviors practiced in business are not compatible with what others might call "basic human values?"

[The Program Manager] lamented that he got screwed unless he acted autocratically. We had one meeting where he talked about a dilemma where he was unsatisfied with one person's priorities. He felt he had to go back and make that person's life more miserable than anybody else's. That was his standing strategy up to that point.

Can these kinds of changes be made when thinking in terms of "what is cost effective?"

I asked [the Program Manager] if he could share the dilemma with the person involved. Perhaps he could say something like, "Look, I'd rather not make your life difficult, but I have found that if I don't do this, I get screwed. Can we work together differently or is this the only option that I have?"

Here is an example of the type of conversation and reflection that resulted from new behaviors.

He said that he hadn't thought about doing that, and it couldn't hurt, because he always had the other option. I don't know if he ever did that...

Can this sort of conversation be a regular hedge against the danger of jumping to conclusions?

Program Manager: I was really upset and I said, "We're going to miss this [deadline]..." I really gave them a hard time and they sat there and listened.

Then one team leader looked at me and said, "I can see why you would feel that way from what you've heard. Let me give you a different perspective on what's going on." He just stopped me cold and told me what was happening from his perspective. Then we talked about what we might do differently.

When he left, I had an entirely different feeling about what he had

done, what he was trying to accomplish, and how he was doing it. He had a better understanding about what I was after. I didn't do that. He did that.

The changed behavior of the two top managers now began to filter down to influence other key leaders on the Epsilon team. Sometimes, this was the result of deliberate interventions by the top managers.

What conditions were created which allowed the Launch Manager to act as a coach to the engineer in this situation?

Launch Manager: I called in [a problem supervisor] and said, "Let me tell you what I'm hearing in the halls about you. I don't know whether it's true...." I repeated complaints I had heard and he defended himself. I said, "What do you think we should do about it?"

He said he didn't know.

I suggested, "Why don't you go and ask people what they think about you? Why don't you do an interview about yourself?"

What he heard wasn't pleasant.

He did that over the next two months. Then he came back and said, "My God, [Launch Manager], I couldn't believe what they were telling me, although I had to practically beat it out of them. I've really been a horrible person. I learned a lot about myself and I don't like what I've learned."

And I just left it alone. Four years ago, I would have said, "You've got to stop being a tyrant." I would have told him to go to the Carnegie School of Whatever-it-is. And he would have just defended himself against my criticism. But if he heard it from the people that worked with him, he couldn't ignore it. He has made a lot of progress since then.

Modeling new behavior: "I don't trust you"

The learning lab created an opportunity for people on the Epsilon team to begin to address deeper issues, with the team leaders involved.

Content leader "X": Another content leader ("Y") and I were down at the learning lab and we were going through

this struggle [with the Launch Manager] about managing change control. My biggest pet peeve is that we were wasting our time in sometimes four and five change control meetings a week. This is not unique to Epsilon; this was going on for years at AutoCo.

But [the Program Manager] and [the Launch Manager] would go after the little [change issues], rather than letting us manage them ourselves.

And managers, having promoted a climate of openness, heard directly about the impact they had on engineers.

Launch Manager: "Look [Launch Manager]!" they finally said, "You're making our lives miserable. You're making our jobs difficult because you're trying to control us. I can't get anything approved without coming to you and getting permission from you to get it approved. Why do we need a system that is so cumbersome?"

Lo and behold, I said: "Because I don't trust you."

Which required them to answer honestly...

Internal consultant: When [the Launch Manager] said that (and, actually, he shouted it), there was an uncomfortable silence in the room. What went through our minds was: We always suspected [the Launch Manager] didn't trust us, and now he's telling us as much. Then [the Launch Manager] proceeded to say, "And let me tell you why I don't trust you. If I did nothing to pressure you, you wouldn't meet your deadlines."

Once this honest statement was made, was there any course of action left but to do something about it?

Launch Manager: I would have had a difficult time saying that in the past. It would have cut the cord of communication and any hope for trust.

...and that, in turn, allowed people to speak much more truthfully about underlying issues.

But what happened next was amazing. I hadn't insulted them. They didn't get mad at me. They simply accepted that it was my position: I thought they would disrupt the system. And I accepted their position: that they were upset with the way I was acting with them. All of a sudden the truth came out. We finally got down to the nitty gritty – a meaningful discussion about how to dispel the problem.

How much self-esteem does it take, under these circumstances, to keep from feeling threatened or attacked?

But, what impacted the leaders didn't always have the same impact on team members.

Content leader "X": In that discussion, [the Launch Manager] said he would help us. He did write a nice strong letter. As it turned out, our solution didn't matter, because within a week, it was all turned around. We were forced to start daily "pink" meetings [named for the color of change request forms], and that was it. So it didn't work.

But it did appear to have an effect in another sense...

Another content leader [interviewed 8/93]: I'm one of the people that "X" and "Y" had fought with in the past. I've noticed already that they handle the issues differently than they did six months ago. They try to listen. They understand that it comes back around to them in the end a lot of times, because you can stonewall something only for so long.

...which others noticed some time later.

Recently "X" made a change that I was pretty upset about. I called him up and said, "You SOB," and started talking through with him the technical effects of that change on my part of the car.

And he actually apologized to me. He said, "You know, I'm sorry. We didn't know we were doing it to you."

I called "Y" up not long ago and said, "Your engineer rejected this change, but we really need it." I assumed that he was going to refuse to help me, and I would have had to force the issue with [the Program Manager].

Instead, he said, "We did that for a good reason. Why don't you and I meet and talk about it?"

We met and brainstormed together yesterday morning, and came up with a couple of ideas. That would have been unheard of in the past; he would have simply said, "I'm not helping you."

I realized that I've got to be retrained too, because I still don't trust them.

Damage from backsliding

The Epsilon leaders occasionally “fell off the wagon” — they slipped back to the authoritarian management style. Stories of lost temper, or accusations of “playing favorites,” seem to last for years, and some of those stories were told in our interviews — particularly about the damage done when a leader did not rein in his temper. This story is typical:

Do people expect more from managers who espouse openness and cooperation?

Team leader: In January 1992 we were having a lot of problems on our team. At an off-site meeting, which was supposed to be a group team-building session, [the Program Manager] blew up. He told us that everybody was finding excuses for why our division screwed up. That was the low point of our morale as a team.

He apologized, but it happened repeatedly. He would fixate on us as a scapegoat, and a week later he'd apologize one more time. After about three times, I made a joke in front of everybody. I said, “When am I going to start believing you, [Program Manager]?”

What shift in assumptions drive a leader to move from a command-and-control style to a more collaborative approach?

But in the last year and a half, after he started to espouse the learning methods, he's changed quite a bit. Now, when he asked for a last-minute change that we would consider an unfair request on his part - say, a minor change in nomenclature which nonetheless affects our software and other technical considerations - he called in the design supervisor and me and personally explained why the name change was so important to marketing the vehicle. He took the time to personally ask our support. That was a big difference from the way he used to dictate from above.

Was this approach really collaborative? Changes were still required that reflected the program manager's interests, but now time was spent listening and considering.

Theme 3: Learning labs: Teaching techniques for thinking differently

What exactly did people feel they experienced in the learning labs conducted by MIT? How much did the labs improve their capabilities, and how much did those improvements affect their work on the car? Finally, how much did they feel the learning lab techniques affected their attitudes about learning?

From our interviews, we tried to find answers to these questions. Some participants were indifferent to the labs, others found them surprisingly relevant.

Program Manager: The MIT stuff is not new. I don't think there's a single technique that hasn't been talked about for 20 or 30 years. But they have put it together in a powerful way that lets people understand it quickly, implement it and really do something with it.

Although the learning techniques and tools were not new individually (many simply represented common sense), their integration and use in a seminar seemed powerful.

Designing the learning labs: from the beginning

Research conducted on these computer simulations at MIT has shown the difficulty in transferring learning from one simulator to other real world environments.³

Would the group-based nature of learning lab instruction, along with teaching techniques for communication and inquiring into mental models, provide an effective bridge for transferring learning lab insights? What was the impact of how these techniques were taught, with managers teaching their subordinates?

Researcher: A group of us at MIT had created a learning laboratory at Hanover Insurance, using a computer-based management simulator. There, we had learned how effective it was to tell people about mental models and surfacing assumptions. I had that framework and flow in mind when we began to set up the learning labs at Epsilon.

The core team thought that our work with them, including the system dynamics and mental models tools, was valuable. They wanted to include more people in the conversation. We decided to develop the lab as a "practice field" - a safe place to talk about and work with the issues that participants would be wrestling with.

³ The dissertation research of Bent Bakken (Learning and Transfer of Understanding in Dynamic Decision Environments, unpublished doctoral dissertation, Sloan School of Management, Massachusetts Institute of Technology, 1993) reports the difficulty of subjects transferring their learning from one simulation environment to another, similar, simulation environment.

Sidebar: Agenda for a learning lab

The Epsilon learning labs were two day sessions where participants were away from their work as they learned new tools and used them to reflect on their own issues. A typical learning lab had the following flow and content:

Day 1		Systems Perspective
Introductions and Objectives <ul style="list-style-type: none"> • teach techniques for thinking & learning • reinforce learning as a part of work • achieve commitment to use new tools in work 	Mental Models <ul style="list-style-type: none"> • how we perceive reality • as barrier to team learning • writing a left hand column case exercise • ladder of inference • balancing inquiry and advocacy 	<ul style="list-style-type: none"> • levels of explanation • new ways of thinking and perceiving • changing ourselves versus changing "them" • systems archetypes • causal loop diagrams • systems archetype exercises
Center for Organizational Learning and its Work	Practice versus Performance Environments	Management Flight Simulator Session
Overview of Learning Lab <ul style="list-style-type: none"> • distribute learning journals 		<ul style="list-style-type: none"> • overview of product development MicroWorld • how to use computer interface (simulator controls) • strategy sheets • simulation runs • debriefing
What is Learning? <ul style="list-style-type: none"> • exercise in personal learning • overview of five learning disciplines • problem solving vs. problem articulation 	Day 2	
	Creative Orientation <ul style="list-style-type: none"> • creating shared visions • stories of successful team experiences • creative tension structure • personal visions exercise • barriers and enablers to car vision exercise 	Learning Lab Debriefing <ul style="list-style-type: none"> • discussion groups and reporting out, or • group dialogue

Practice fields are meant to take people away from everyday performance pressures, and allow them to develop some new attitudes and approaches in a simulated environment. Just as a flight simulator prepares airline pilots for emergencies and pressures that would be shocking in real life, a practice field helps people learn to cope with emergencies and management pressures before they occur.

Launch Manager: In his speeches, Peter Senge asks: "How do great teams become great teams?" Well, they practice. Orchestras practice; baseball teams practice. Great teams tend to practice. But we don't have that capability in the business organization. So how do we become great teams? Well, we just kind of luck into it.

We wondered: "Is there a possibility for us to create an environment where we can create that kind of practice?" The way to do it is to start with learning labs: small groups in which we can begin to transform the way people think, behave

and communicate in teams, and then create models that they can use to practice the behavior in reality.

Can AutoCo managers incorporate some of these strategies into current meetings? And practice in that way? Or must it be done through formal learning labs?

Launch Manager: [The Program Manager] did not directly participate in the learning labs. We were afraid his presence would inhibit the free flow of information and discussion in the group. Whereas apparently I was in a position where I would not have that effect. We were going to test this by bringing [the Program Manager] into a learning lab, but we didn't have the opportunity.

The participation of management at AutoCo in these sessions, even as speakers for the dinners, was repeatedly mentioned by participants as one of the most important things they remembered about the sessions.

Nonetheless, [the Program Manager's] involvement was critical. He would always show up on the Thursday evening dinner. And we would have a guest speaker come in, generally from inside AutoCo: The Program Manager's boss, the segment director, or an operations person who has been a champion of this approach.

Choosing learning lab participants

The original strategy was to include everyone on the Epsilon team in the learning labs. Later, the team realized that a critical mass of practitioners could influence the whole team, without having to put the whole team through the labs.

Would this strategy work for other organizations?

Launch Manager: We had four Learning Labs for about 100 people in total. The first was in the Fall of 1992. We picked an interior team and an electrical team, deliberately choosing younger engineers whom we thought would be "early adopters": more open and receptive to new ideas. We conducted a two-day session. It had taken us, the Core Team, eight months to learn to "stop being bosses": to begin trusting and communicating with each other. Now, how could we accomplish the same result with a group of 20 people in two days?

Participants in the labs were selected, rather than invited to volunteer.

Well, to my surprise, the Learning Lab went extremely well. I was elated at how quickly they grasped the learning tools and began to use them - first, right in the Learning Lab, and then elsewhere on the team.

We went on to another learning lab with some very tough people: engineers

who had been at AutoCo for many years, whom we didn't think would ever open their minds up to new learning. These were crusty people, who were sitting there wondering what the matter was with us.

We discovered that in only two days these engineers were very quick to adopt these new tools. One of my roughest and meanest team leaders had been there 35 years; we had to practically drag him into the learning lab by force. Afterwards, he walked up to me and volunteered, "Gee, [the Launch Manager]. If I knew how good this was, I would have brought everybody that works for me."

The systems archetypes and "systems thinking" skills

At first glance, work on systems thinking is akin to "process mapping" and other quality movement methods. Participants uncover the hidden interrelationships that govern their own work. However, the mapping techniques are based on system dynamics theory (which is, in turn, based on servomechanism theory). This shows how situations evolve over time: how a "fix" may ultimately backfire, or how pressures may grow or diminish. As one internal consultant put it: "The systems thinking work shows how you can't just throw people and money at problems."

For more on archetypes, see Sidebar: What are systems archetypes?, page 7.

The tragedy of the power supply

One of the first understandings to emerge from the Epsilon systems work was the "tragedy of the power supply," a case in which two functional groups on the Epsilon Team were involved:

The multiple teams who were involved could see only the importance of their part of the power supply, and could not reach an agreement that accommodated everyone.

Launch Manager: The team spent five months trying to resolve an electrical issue which they couldn't resolve at their own levels. Each engineer had a component they wanted to maximize. Each engineer's organization said to them, "Don't give in." So whenever they came to meetings, they argued the same issues: "No, I can't give up the air conditioning unit design. We need it to be the best in the world." Or, "I'm not going to change the headlight designs because they have to be the best in the world."

But we had a battery system which couldn't accommodate all these best-in-the-world components. Usage was twelve amps above capacity. If we had

identified this earlier, we might have been able to put in a second battery, or a higher-capacity battery. But by then, the area available in the car couldn't be made any bigger. So they were trapped; they had to start compromising. And they couldn't resolve the disagreement, despite all their intentions.

Doesn't an engineer, or team of engineers, who couldn't resolve a problem consider himself, or themselves, as having failed?

Content leader: Embroiled in these arguments at a Battery Charging Team meeting, I suddenly thought back to the systems work from the first Learning Lab. "Do you think this sounds like a tragedy of the commons?" I asked. I talked about the Tragedy of the Commons systems archetype: The more a common resource is depleted, the more individual actors fight for a bigger share.

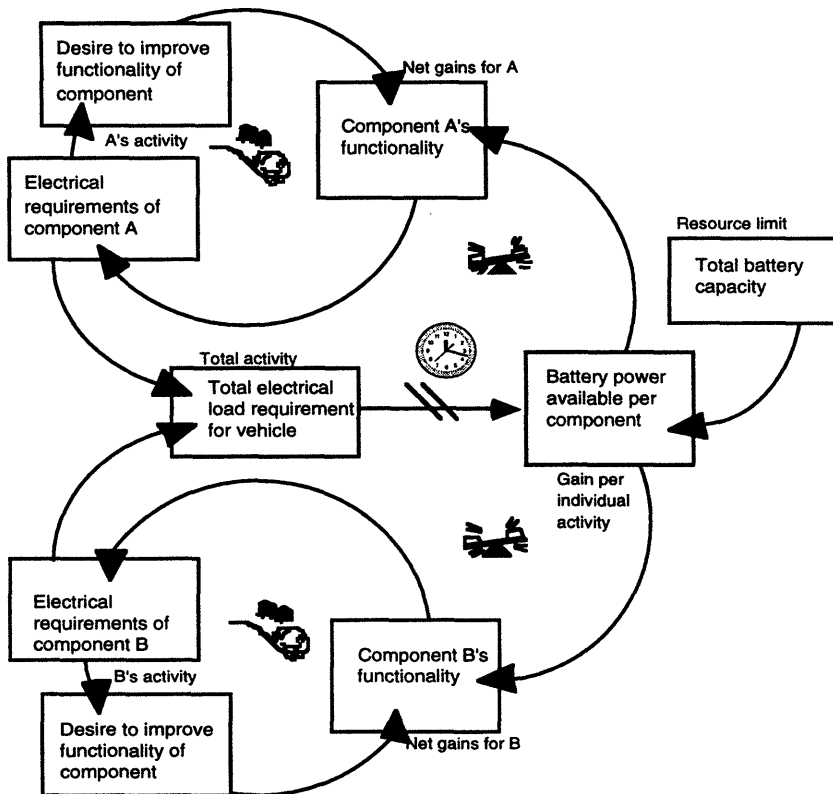
What if the problem that they sought to solve fell outside of their domain of technical and strategic decisions?

One of the process consultants, who knew the work, reminded us that it's very difficult to resolve a tragedy of the commons issue at the individual level. "How would you feel," he asked, "if we went to the Program Manager and the Launch Manager with this?"

An insight into the structure of the problem helped the team seek the outside help that it needed.

"I'd feel really bad," I said. "I'd feel like I failed." But we realized how the system had created a "no-win" situation, and there was no shame in going to the Launch Manager and the Program Manager with this systems issue.

Sidebar: Mapping the "Tragedy of the Power Supply"



In the two reinforcing loops (shown with snowballs) at left, each components group pursues actions which are individually beneficial — building electrical components with high functionality, which leads to high electrical requirements. In combination with the other components, this raises the total electrical load requirement to a point where the battery power available per component runs up against the limit of total battery capacity. Facing these limits, each design team, within its own group, adds even more functionality, as justification for being allotted as much battery power as possible from the common pool. The result is individual gain, but collective pain.

Participants agree that the value of this systems story, and others that surfaced in the learning labs, comes from their real-world relevance.

Program Manager: Finally, they mapped the story for themselves. They laid out all their individual problems and their incentives: Their organization's goals, how the results would be measured, and what the effects of a solution would be on that chimney and its performance measures. It was clear that they would all be individually right, but the car would fail. No one wanted to be criticized by his management for doing something foolish with his component to make somebody else successful. But they could all accept what they would have to do to make the car work - if the Program Manager told them to do it. They accepted my orders, even though the orders went against their chimney's objective, because they knew that's

what it would take to solve the tragedy of the commons.

Once they were worked out, there was a tendency to package the systems stories for wider distribution. Unfortunately, this gave the impression that the stories had primary value as prepackaged lessons, rather than as learning experiences.

"I'm still hearing the same stories about 'the tragedy of the power supply,'" said an internal consultant. 'It's almost become a joke with some of those folks: 'Why don't you whip out the old 'tragedy' chart?' So now, people tend to think that everything is a 'tragedy of the commons,' when they should be investigating their own situations." Does this type of pre-packaged learning need to be guarded against, as it appears to drive out experiential learning?

Did people use systems maps as a way to present problems to the Program Manager because they thought he would be better able to hear them?

Program Manager: I had heard of a few examples of system mapping. About most of them I said to myself, "That's rubbish. They would have come to that conclusion anyway." But in this case, if they hadn't mapped the problem, they would have fooled around for another couple of months, and it would have been too late. At the time of the EP prototype we would have discovered that the batteries went dead, and we would have had to do something drastic.

Another case was the battle between the chassis team and a development team that was struggling with noise, vibration, and harshness. Each side had a problem, and every time one side fixed it, it made the other side's problem worse. Both sides knew that if they had talked to each other earlier, they could have come up with a collaborative solution. But they hadn't. So both sides attributed motives to each other: "These guys are not willing to communicate. They don't care about us or our problems. So screw them." They stopped talking to each other.

Then we decided to map the problem together. It changed the way the two groups did business with each other. In fact, the chassis guys became very active with the development guys in design up front, instead of waiting back home for a transmittal from the development guys telling them what they had to do to fix the car.

There was nothing novel in the system diagram. Any of them could have

thought of it by himself. But because they did it together, and because they could draw something they all could see, they changed the way they worked together.

The ladder of inference and “mental models” skills

Outsiders find it difficult to grasp why these mental models skills seem so important to participants. One reason is that they offer a tangible, accessible way to quickly change non-productive conversations, and to build the ability in oneself to ask effective questions.

See Sidebar: What are the “ladder of inference” and “left-hand column”?, page 8.

Team leader: When you have the time to practice the “mental models” skills, they’re not needed. But when you get into a crisis mode, everybody starts scrambling and worrying about their own thing; the communications fall off. You’re facing off against body engineers not wanting to make a change, or another division, and people don’t want to work together. That’s where if you can retain a little bit of the skills, it helps you through a little bit.

Many of the simplest, almost common sense, techniques were what people remembered and continued using outside the learning lab environment.

One tool from MIT that I think has helped quite a bit is the phrase: “Let me repeat what I think I hear you saying. Is that right?” We use that in meetings, and it helps in communication quite a bit.

Was some of the influence that the learning labs achieved related to the fact that engineers were open to learning management, behavioral and communication skills when they came from a respected technical university?

Team leader: For an engineer, the material was very hard to grasp; it was so intangible. But the presence of MIT, as a respected university, gave it a little bit more credibility. The learning lab project wasn’t just [the Program Manager] and [the Launch Manager]; it wasn’t just an internal, whimsical, fly-by-night plan.

What does this influence say about the “mental models” of engineers?

Many people commented on the usefulness of the ladder of inference (also often called the ladder of abstraction).

Content leader: The ladder of inference was worth its weight in gold for me. If you wanted to make sure that the other person wasn’t just saying a conjecture or assumption, you could ask him to go down the ladder so the conversation got back “down” [to data]. All of a sudden, things would clear up.

Engineer: One of my suppliers – an

excellent supplier, by the way – mentioned under his breath at the end of a meeting that a part would be two weeks late. He didn't want anyone but me to hear him. I gathered my wits for a minute, because it was really a shock. We hadn't had any problems with that part. I asked why it would be late, and he said, "I'm not sure."

I could see everyone's eyes in the meeting hitting the roof. They were saying to themselves: "God, Bob is an idiot. What's wrong with him? We thought he was a good guy."

At that point I could see we needed to bring things down on the ladder. Bob was not going to tell us any more. Why would he? He had already risked enough, and made himself look foolish. So I consciously brought things down to observable data.

"Look, Bob," I said, "If you don't tell us where the problem is, we'll never get to the root of this issue." After a lot of prodding and pleading, it came out that he thought it was a problem with paperwork that we [at AutoCo] had lost, and he didn't want to be the one to tell us.

So I went back and checked. AutoCo's system had misplaced not only his paperwork, but a lot of paperwork. It was causing parts to come in a little later than anticipated, just enough to mess up the build. Bob's honesty allowed us to make sure everything coordinated on time.

Traditionally I would have thought, "I don't want to hear about Bob's problems. He is an idiot." I wouldn't have checked the reasoning which led me to that assumption. I would have said to him, "Just make sure the part's on time." There would be nothing he could do about it.

It's very difficult to use the new approach when you're the only person in the room who knows what it is. For instance, someone will make a statement like, "Oh, Bob's seat belt's going to be late." Everyone in the meeting says, "Bob's an idiot." You've

The ladder of inference concept created a possibility in the minds of engineers that the ways they "saw" things might be different than how others saw them. This led them to inquire before they jumped into action.

The ladder of inference was a conceptual tool which engineers, as this example illustrates, used to create different possibilities for how they communicated with suppliers and other engineering teams.

And by asking questions, instead of demanding action, the engineer in this example found out that there were problems in the way AutoCo had handled the supplier's paperwork.

The example illustrates how a hierarchical culture plays itself out. The engineers are given commands by their bosses to get parts out, and they in turn deal with suppliers in a command oriented fashion. There was no latitude to respond in any way except doing what you were told.

Moving conversations among groups of people from beliefs and assumptions to "observable data" was particularly helpful in an engineering world where people design and produce tangible, technical, automobile parts that need to work together.

Terms like "jumping up the rungs" or "where are you on the ladder?" were used to ask people to tell one another about the data they had to support their assertions.

jumped right up those rungs. No one even says, "Why are we saying this?" When I'm by myself, I'm more hesitant to step out of my comfort zone and ask that question in the group.

But if you have at least one other person who has been through the learning lab, you can bounce comments off each other without people in the meeting knowing what you're doing. It builds and builds and you see it taking effect. I'll be in a meeting with "X" [an engineer I work closely with], and he'll call someone else a moron. I might say, "X, you're at the top of your ladder." No one else knows what that means. But he'll think about it and say, "You know, you're right." Then we discuss why the "morons" might have done what they did, before we carelessly jump all over them.

Content leader: When the learning lab ended, I said, "Well, that was neat. Let's get back to work." Very little of it seemed to stick with me.

But after a couple of days, I started thinking about it. We gave it a try in our meetings. "Come on guys," I would say. "Let's start talking about observable data here instead of our opinions or our belief systems." Because you don't make any progress at all by shoving your opinion at the other guy. By now, we recognize right away now when that's happening. We used to sit in meetings for hours, and everyone walked out pissed off.

The ladder of inference was a technique which was taught in learning labs, and provided a foundation for other activities, experiences and conversations that took place in those "labs."

The "management flight simulators"

Some of the work of the learning lab centered around "simulation" exercises, programmed onto computers by a core team of Epsilon and MIT people, in which different strategies and approaches could be tested.

In general, the computer models weren't considered as crucial as the systems and mental model conversations.

Content leader: I thought one beneficial thing was the computer game we played at the end, where we could see the consequences of our decisions.

Sidebar: What is a "Management Flight Simulator"?

A management flight simulator (MFS) is an interactive decision-making computer simulation. The Epsilon learning labs used a management flight simulator based on the product development process in the automobile industry. Underlying the computerized "cockpit" (the simulator controls) is a system dynamics model, in which mathematical formulas represent the links between different variables in the system. These interactions have been tested and calibrated through extensive field work in a variety of organizations. Nonetheless, like those of any

model, they represent a simplification of the reality that the model is intended to describe.

A crucial feature of the management flight simulator is the environment in which "play" is conducted. Instead of simply "testing new strategies," as if they were playing a computer game, participants articulate their strategies ahead of time, and the reasons why they have chosen their course. Then, the model helps them see their own tacit, deeply held theories about their business environment and organization, theories which guide many of their decisions in real life.

Discussions with other team members help everyone see how those tacit theories affect the entire operation.

This illustration shows the "cockpit" (the on-screen command center) of the Product Development management flight simulator. The left-hand part of the screen shows indicators of activity in the project (for example, the percentage done). The right-hand part allows the participant to enter decisions (for example, about the number of process engineers to hire) and select reports or graphs for viewing. These reports and graphs show, in detail, current status and historical data.

Menu Bar. Click and hold mouse button on the menu choice and drag the mouse to the desired command. Start a new game by selecting the restart option under the explore menu.

The "Cockpit"

Menu Bar: File Edit Explore Options Model

Weekly Decisions

d_Process_Engin	0
e_Coordination_F	.00
f_Schd_PE_Finisl	25
g_SchdReleaseDa	40
h_Quality_Goal	1.00

Step: 1 time

Reports:

1. Project Overview
2. Product Eng Status
3. Process Eng Status
4. Quality Report
5. Time & Cost Forecast
6. Project Summary
7. Sales & Marketing

Graphs/Tables:

- A_Decisions2
- AIL_Staffing
- COP_Team_Costs
- Cost_of_Overtime
- Cost_of_Project
- Engineer_Experience_Mix

Decision Area: Select the decision you want to change and enter the new value.

Click on the step button to advance the simulation one week.

Reports show the current project status. Click on the desired report.

Graphs show history of the project. Click on desired graph.

Use up and down arrows to scroll through choices

Click on graph bar and drag to desired viewing option.

Main Viewing Area. Any graph or report selected in the "cockpit" will appear here. To close a graph or report, click anywhere in the view, then click in the box that appears in the left corner of the gray bar at the top of the viewing area.

The computer models were effective for engaging some AutoCo team members, but they ended up being seen as too time-consuming (and perhaps a distraction).

The models were abandoned after the

Many people said to me that they think that the thing that changed their outlook on things the most was the computer game. They had thought they had all the answers, but they plugged them into the computer and all of a sudden the red flag came up. We

second leaning laboratory because other exercises seemed more effective.

The comment here reflects one of the few remarks singling out the computer models as an important factor.

Perhaps different learning styles need to be taken into account in the laboratory designs?

It is interesting to ask the question whether, for some people, the data they get from a computer model has more impact on them than what people tell them?

didn't do as well as we thought we would.

It was comparable to the "beer game" that MIT uses in its courses.⁴ I had a fellow in powertrain sitting three chairs away from me in the "beer game." He was absolutely unmanageable, positive he was doing everything right. He made decisions and, in the end, he messed up pretty bad. And after that he was a totally different person. He wasn't so adamant about things.

Reinforcement: A learning room

To provide opportunities to continue and support the learning process, AutoCo program managers created a "learning room" where they held weekly breakfast meetings. This "learning room" was a conference room without a table, with flip charts from learning labs on the walls, and chairs set in a circle to create and encourage an atmosphere of peer-to-peer conversations. The breakfast meetings entailed managers buying doughnuts and inviting engineers to "drop in" to ask questions and for unstructured conversations.

How can reinforcement be designed in the AutoCo environment, so people don't lose the chance to practice learning skills as part of everyday work?

Researcher [in mid-1994]: We're trying to put support mechanisms in place: conversations to continue and reinforce the understandings which people gained in the original labs.

Launch Manager: Once a week, Wednesday morning, we asked members of the team to come to the learning room. There they could discuss anything they wanted, but they had to leave the baggage of the conventional way of operating behind. "Bring your problems and use the tools. Practice them."

You know what they say: "What makes great teams? Practice." But how many business teams practice? Well, we're learning how to do that.

Reactions to the learning labs

General reactions to the learning labs were mixed. Some people we interviewed weren't even sure they had been to one. Most remember being skeptical, yet being impressed that the boss was in front of the room talking about mistakes and learning from them.

⁴ For an explanation of playing "the beer distribution game" see Chapter 3, "Prisoners of the system, or prisoners of our own thinking?" in Senge, 1990, *The Fifth Discipline*, Doubleday/Currency, pgs. 27-54.

Assembly launch leader: When I learned about these "dog and pony shows," I thought, "Oh, they're flag waving." I laughed at it. "My God, what a waste of time and money."

But I saw it work. The labs got people talking to each other early on, picking each others' brains. They found out what things had gone right and wrong from prior launches, what we could do to make things better, and what pitfalls to watch out for. We had not only the engineering fraternity but the Mission Hill people, Assembly, and all divisions trying to talk to each other and air their dirty laundry up front. That built the team.

Later on, other groups, like the truck people, used to needle us: "You guys having another cake and party? Another off-site boondoggle?" But even though you didn't have to believe in it, it did work and it's showing here.

Would I do it again? No. I plan on retiring. But I'd advise another manager to go for it because it's proven out on the Epsilon.

Epsilon Program Manager at a key supplier: Companies like AutoCo are so authoritarian in their nature, that anything you could do to blast them off of authoritarianism and start to move them towards participation and team work is a good thing. I don't know about the MIT method, but I see it as having created a kind of level of credibility for team work, for the team effort. They might have done that using other methods, too, but the main point for me is that AutoCo is making the effort.

Internal consultant: I talked to a couple of folks who were not part of the learning labs. They might have gone to an off-site or two. They said, "We don't do much of the MIT stuff, except when [the Launch Manager]'s in the room. [The Launch Manager] will say, "Where are you on the ladder?" And people start talking about that for a while. It's not that they don't think about it; it's just that they're

Even when people found they learned in the learning labs, it was seen with great skepticism by their peers in other groups. Was learning something that engineers could talk openly about in a culture that valued hard, scientific and technical types of conversations and data?

As managers and engineers, AutoCo people didn't know why it was effective. Some only professed that it was effective, which was their justification for continuing and promoting the work with MIT.

If people do not feel confident or capable of applying the tools, is that cause for concern? If it is cause for concern, does it suggest that changes should be made in a) the design of the AutoCo learning effort and learning labs? b) the MIT implementation of these tools? c) the

learning organization approach in general?

focused on their day-to-day jobs.

Are people more capable of using these tools than they immediately realize?

A couple of the hard-core engineers say things like, "I don't know why you credit this MIT stuff. This is a successful program because we set dates and held them." That's the engineering mindset.

Learning Labs: How did they contribute to the change process?

Some of the learning lab participants commented that the changes they saw in management (particularly the Program Manager, the Launch Manager, and the Chief Engineering Manager) were much more significant than their own changes from the learning labs. A few members of this group admitted that their own positions probably kept them from seeing the effect on themselves as clearly as they saw the effects on other people. Other participants talked at length about changes in themselves: an increase in openness, less tendency to blame others (or themselves), a better understanding of their own biases and mental blinders.

New techniques taught in learning labs that influenced thinking were not as obvious in anyone's own behaviors as in the behavior of others — particularly visible others, like their bosses.

Content leader: When people go to learning labs, there's no light bulb that goes off: "Aha!" The change stems from the subtleties that go along with it. People go back to work thinking about trying to be more open, more honest. Maybe they're just more open to the possibility. They see themselves interacting differently with each other. They don't say, "And now I'm going to use the ladder of inference," but the tools creep up in the backs of our minds.

What do these differences mean in trying to assess the effect of the learning labs? Is it worth trying to devise an "objective" way to assess the effects?

Researcher: This is the tough part of measuring learning. If you learn something that sort of changes the way you see the world, it may look like nothing has really happened. Yet everything has changed. Paradoxically, the bigger the change is, the less visible it may be. The change may take place in so many subtle ways, so diffusely, that you can't see it until, over time, you gradually see a bigger picture. It's much less visible than a problem-solving effort, where you can show visible progress on how you solved that sucker.

There was value expressed by participants in the experiential component of the learning process. Although managers and engineers often ask for materials to read, and don't take time to attend workshops, having the learning labs partially facilitated by managers and focused on current work issues made it feasible for people to find the time to participate.

Could AutoCo be expected to become a "learning organization" in the way it was described in books and articles?

Were the modifications people made to the textbook definitions helpful in adapting the learning materials for the AutoCo culture?

How might those changes have influenced the effectiveness of the tools?

This comment was typical. People appreciated the learning tools, but in the short exposure they had, they struggled

Launch Manager: You remember how in the Wizard of Oz film, the scarecrow got a diploma to legitimize his learning. Maybe we did something similar here. Maybe, by bringing in MIT, we legitimized our focus on process improvement much more than we could have done on an ad hoc basis. Without MIT, I would not know how to use these powerful tools. I could have read about them in a book, but without somebody showing me how we practice them over time, we would not have realized the tremendous power that is in those artifacts.

Process leader: What does it mean to be a learning organization? I think we fall short on several dimensions.

When you hear folks describe the left-hand column, for instance, it's not even close to a textbook description. They say things like, "If they can expose their emotional stuff, we can get to the real truth." As opposed to how I would describe the left-hand column: the things that people haven't said because the environment is not emotionally or politically safe.

If we don't understand what a tool really is, how can we use it very effectively? I may make too big a deal of the textbook, but I think that we need to do a better job of helping people learn what the tools are really about. MIT and AutoCo are learning this together; and the difficulty is, that there's not a lot of patience around here for struggle. That's not just true of AutoCo; it's an organizational reality. MIT needs to understand that organizations like AutoCo will fault MIT for not providing enough support, or for not teaching us well enough.

I rarely use any of the tools in a meeting. I wish I could. I think about them. If I tried to use them in a meeting, I'd go home with my head in my hand because I'm not good enough. I'm not facile enough with archetypes.

The tools you'll see used most are the communication tools: The ladder of inference and the left-hand column.

to use them on line.

The ladder of inference is helpful, but unfortunately, we generally use it during our spare time. If we're in a meeting and things don't go well, we don't use the tool then. We walk away from the meeting, and on the drive back to the office together in the car, I might say: "Well, wait a minute. You know, we're way up here on assumptions. Let's come back down to behaviors and try to understand why so-and-so did that."

Attitude, as well as aptitude, influenced the effective use of learning tools.

Or people may use the tools sarcastically: "Well, let's get into our left hand column and see if that helps, heh, heh, heh." And then they obviously don't.

Again, this comment is typical of how people made linkages between learning tools and business results. The learning tools helped, but were diffuse and subtle. No one, other than the program managers, attributed the Epsilon program's achievements to the learning tools. Yet, people often talked independently about the efficacy of the learning tools for product development management issues and achieving metrics which were significantly better than many other vehicle programs.

In other words, the tools are very powerful, but until we can use them accurately and on the spot, we haven't gotten there. We're still on the learning curve.

How could a definitive connection between learning and results be examined in this type of setting?

I think the results are better. I hear both sides. Some folks from the pilot plant, who built the EPs, say this is the best product they've had come down the line. It looks better than an EP ever has before. Other people say, "You guys are flat on your ass. There's holes where they don't belong [in the auto body] and so on." In terms of hard measures, the one you can look at is the percentage of parts that were ready the day we were supposed to start building. I've heard numbers that say we're at 80%. Typical is more like 40%. If that's accurate, then we are much better.

Now, is that a result of what we've done at MIT? It beats me.

Theme 4: Combining engineering innovation with human relations: The Harmony Buck

An innovation developed by the Epsilon team, the harmony buck, has become standard practice at AutoCo. Harmony bucks are a part of most new interior programs at AutoCo, beginning in 1998 and beyond.

The team's interviews suggest that large scale technical innovations like this one are possible because of two factors: a willingness to break established methods, and the opportunity to collaborate across functional lines.

Sidebar: What is a Harmony Buck process?

The Harmony Buck Process (HBP) is an engineering tool to assist in design efforts. It is a "full body" design aid encompassing the entire car, on which engineers and designers can test part prototypes and diagnose engineering issues generated *after* the first set of prototype parts have been delivered.

In the past there might have been many individual "bucks," representing different sub-assemblies of the car. Having a single "harmony buck" vehicle, accessible to all team members,

provides a central focal point for improved coordination among engineers.

With this early non-driveable "vehicle", an assessment of each component and system can be made before a prototype is built. Issues such as: basic design concept inadequacies ("My part doesn't actually work the way I assumed it would on paper..."), design incompatibilities ("My part clashes with another part..."), and system interactions ("When I get these parts all together, it just doesn't look right or

function right..."), can be evaluated and corrective actions taken early. In addition, the experience assembling and maintaining the harmony buck is invaluable at the build site.

On Epsilon, the HBP identified almost 300 issues, including about 30 "no-builds" — all corrected *before* the actual builds. At the end of the program, more than half of the \$90 million provision for changes at launch time was returned — unused. The HBP deserves a portion of the credit for this accomplishment.

The harmony buck story begins with the program managers' efforts at the beginning of the program to bring people together to set common goals and objectives, and to establish something intangible — a vision of what it would be like when they accomplished their program goals.

How do such visioning processes affect team innovation?

Program Manager: In the early stages of the launch, we asked people at every level of the team for their image of an ideal launch. What would make this launch successful, in their minds? Ultimately, the team settled on the image of a Maytag repairman, from the well-known Maytag commercials. When the car moved into the assembly plant, everything should be so well established that there wouldn't be anything left to do.

This simple image (the Maytag

Then we asked ourselves, "Ignoring the fact that we don't think it would be possible, and ignoring the problems

repairman) was used to guide many complex actions.

we anticipate - how would we make this "Maytag repairman" future come to pass? We said we would have to redefine the schedule. Engineering changes should stop at the point when prototypes are in progress. During the period of building prototypes, the suppliers should be learning how to make the parts. Then during the stages of pilot production, the assembly people should learn how to assemble the car.

By the time we got to Job One, we hoped, all that would be left would be training the assembly people to produce the car faster. Everything depended on setting things up so the engineers wouldn't make changes after the prototype phase. But how could we accomplish that?

Getting approval: Building confidence to make a case for taking a risk

Making the case to upper management for additional investment is not always easily done.

Top management approved the request for a concrete technical innovation, like the harmony buck, more easily than they approved a more abstract and general request for support of learning process initiatives.

This comment about the Program Manager's predecessor illustrates the importance of individual personalities and management styles. The systematic nature of the learning efforts with MIT were only possible, like the harmony buck, because of the Program Manager's willingness to support innovations. The importance of this is not to be missed — it appears that organizational learning is built upon individual learning and individual openness to new ideas.

Engineer: The harmony buck story started back in June 1989, at the very beginning of the Epsilon project. The question came through our department: What would it take for us to meet and exceed the standards for components of the past? This was the most complex car that we had ever done. It was pushing the edge of so many different component technologies that the electrical community was very nervous about it up front. We recommended a full harmony buck from engine to rear; the idea went up to the program manager at the time, [the Program Manager's] predecessor. He was not very interested, but some of the other team leaders were. They realized the impact the harmony buck would have on the quality numbers. Different engineers from different parts of the car began to push for this thing.

Later, after [the Program Manager] was there, there was a meeting of supervisors, managers, and executives at world headquarters. The senior managers asked again: How would we raise or exceed past standards? Once

again, the harmony buck answer came out.

Researcher: One of the engineers who initiated the harmony buck told me that he would never have had the audacity to propose it, except for the encouragement he had received during the previous months at Epsilon. [The Program Manager] had been saying: "Look we're not going to beat up on you. We really want to know what you are thinking."

Chief Engineering Manager: We presented the idea in a meeting with our Vice President. We said, "We think we can save money downstream by finding the problems earlier. But we can't tell you exactly how much it is going to save. You are going to have to trust us on that, and put up about one million dollars up front."

He turned to the controller and said, "I think I am projecting to under-run my budget by about \$2 million, so let's go do this."

If there had not been a cushion, would the harmony buck have been approved? Should it have been approved, if there were no cushion?

Approval required a personal commitment by the Program Manager to "take \$4 million out of the provision for changes."

Vice President: I am pleased that I approved it. The idea was to make sure that parts fit well before they ever showed up in a prototype, or later in Assembly operations. If you had a problem you could go back and, with a master grid, finesse your design until it fit. The buck was over budget but the team found a way to offset it. And it was a good idea. It was probably one of the very positive things that the team came up with. They committed to downstream savings in change count and PCR provisions which they wouldn't need. The first time through would be more expensive, but we thought it was right for quality and I think the cars turned out very well.

Assembly Launch Leader: The harmony buck let the engineer fit a wood mock-up or whatever of his part onto that buck before we ever had a physical unit going down the line. The engineer

could try it out and see: Would it really work? How would it package with other components?

We proved out so many parts on the harmony buck before we ever built our first unit. Usually, in other launches, by the time you get to 1PP the engineers are still going back and changing the entire part again. But the parts that were in our car in early 1993 were pretty much the same as in the package now [in September 1994]. Certainly, a lot of changes happened. Some of them you would see, some of them you would not. But the basic vehicle layout, including the wiring, did not change a whole lot.

Implementation reveals resistance within the team

Approval by top management did not mean that the innovation was supported by everyone involved.

The idea of yet another prototype was interpreted by some people as meaning more work, taking more time, and costing more money. It meant changing the way things were done, and there was resistance.

Engineer: At first, there was an internal struggle. The current buck program was a current Epsilon body with a new front end and front underbody. We were looking to expand that to a full new body and new interior. The people responsible for the front end resisted, I think, because we took the buck away from them for a month and rebuilt it from scratch. "Don't interfere with my job," they said. "I don't see any benefit in doing this anyway. I don't want to have anything to do with it." Every day somebody would tell us, "This is just not worth the money. I don't know how you're going to do it."

Over time people found that the harmony buck helped, and were more overtly supportive of the effort.

But those of us who had a clear vision of what it would do kept pushing until we got it through. We had to keep telling ourselves that it would work. We knew it would pay off in the end. We got into a few meetings about the buck. At one meeting, [the Chief Engineering Manager] said, "No, go ahead. Go on with your program. You'll get your body back soon enough and it won't lose you that much time."

I think that was a big turning point.

Does this comment suggest that AutoCo people need to push themselves beyond the comfort zone — beyond what they think may be realistic?

Content leader: If something hasn't been done before, we tend to be negative about a new idea. I heard some people say, "Why do we need another prototype? We already have enough prototypes." Well, we haven't implemented a bad idea on the Epsilon program yet.

If so, how can AutoCo's management systems and culture reinforce people for this? Could reward systems create incentives that lead to new behaviors?

Engineer: The harmony buck was a great tool. It was AutoCo's first harmony buck. It allowed a total car to be reviewed up front before we built the cars. Previously, the company had used partial bucks. By having a full car package buck, all the engineers could understand the interactions between their part and the surrounding parts. That allowed us to build VPs on the assembly line — which had never been done before. Typically all the VPs are built off line because the parts are not ready enough to go into a car and maintain line speed.

The harmony buck was linked by many with achieving better results on subsequent prototype builds.

The harmony buck as a communication tool

The technical innovation of the harmony buck served as a social innovation. People needed to meet around the physical buck, and these activities created opportunities for communication and collective problem solving.

The person hired to be in charge of the buck was able to bring in additional innovations.

Content leader: We hired a manager from outside the team to take charge of the harmony buck. He developed a lot of processes that we had never used before [involving computers and checking points]. He got a lot of cooperation from just about every group in the company, but it was largely a one-man effort. I hate to say this, but I don't think that we could have found anybody on the inside who would have been willing to take as many chances with as many new ideas and processes. He really showed us a lot of stuff.

How can companies like AutoCo cultivate and reward these in-house "pollinators" of ideas? Or is special cultivation of them necessary?

Engineer: We hired outside prototype shops, instead of the suppliers, to make many of the prototype components for the harmony buck, because they

The physical nature of a buck that was continually updated provided a place for people to meet and talk, as well as test out the compatibility of their parts and subassemblies.

Coordination became a critical factor in making the buck successful.

Did the white board in the harmony buck room help people in voicing their concerns formally as well? The record number of change requests (see Freezing and reducing the change requests, page 79), may have been exacerbated by this board.

While a diffuse and subtle learning effort might not be linked to business results, an innovation like a harmony buck, which was made possible in an environment created to be conducive to learning, is associated with financial results.

could do it more quickly. When we started getting those prototype parts in, it was a big turning point. You could see the interest ramp up. We had to fight to schedule time for people to get in there to see their parts.

Once we started the buck build, people were in and out of that room frequently. It was a good place to resolve a lot of problems because again, we made it accessible. [One content leader] stressed that to us: "Let the people get at it. If they want to call you up and come on down, don't resist." You could go down there anytime in the day and there'd be from 10 to 15 people around this thing, all doing something different. We made sure we set up a process where someone could easily change a part. "Just let us know about it," we said.

Program Manager: When we started the harmony buck process, we added a whiteboard to the Buck room. Whenever someone found a problem on the harmony buck, it was written on that board. And the rule was that you couldn't ever erase something from the board unless you had placed a concern number behind it. Once you make it a concern, everyone can think about it: "Now, how might that affect me?" Everybody can work on it together.

Content leader: After the harmony buck was set up, the engineers loved it. They were swarming over it. We found all kinds of problems, way in advance of even the first mechanical or EP. We paid that \$2 million back in a week or two, finding concerns early enough that we avoided major expenses in retooling. This was probably one of the big reasons we have underrun our investment target.

Expanding the collaborators

As the first harmony buck evolved into a more complete (EP) build, the coordinators deliberately set up the car's process to involve all of the constituents, including the Mission Hill assembly people and suppliers, in the design of the car from the beginning. Suppliers could see problems before they reached assembly.

Suppliers met with assembly people at the buck, and depended less on telephone calls to one another.

Assembly Launch Leader: In the past, assembly was the neck at the end of the funnel. Whatever transpired early on in design and engineering, eventually gets into the assembly plant and we have to assemble all these dreams. And they don't always fit. With the harmony buck, we moved the neck of the funnel up into the prototype stage. When we found problems, we called in the suppliers and told them where their mistakes were and what we required so that we could assemble the thing. We did this up front; we didn't wait six months to find these problems. To me, that was one of the biggest keys.

Are people more effectively involved if there is an upfront communication/ planning process?

Engineer, car programs management: Two guys who worked with us actually worked for the supplier that did the sheet metal body. They just stayed on site for a year, which is very uncommon, and they got very familiar with the car.

When we reached the EP phase, we wanted to develop the build sequence basically as if we were running it down the line. Those of us who were managing the harmony buck insisted that the engineers and suppliers come to Roxbury and install their parts. We told them: "Come and do your own part. Try everything out yourself."

So we scheduled times. We started with the wiring guys - engineers and suppliers. We had planned a half day for them, but they begged and pleaded for us to give them more time, because they started to find things that wouldn't have permitted us to build the car.

For example, a harness of 120 wires hanging out the door. There was no way to loop it around and attach it; we had to rework all the parts. If we hadn't noticed it on the harmony buck, it would have meant boxing up all the parts for the EP build, stopping the

build, and shipping the parts back to the supplier.

How much time would that have taken? How much money would we have lost? I don't know how to quantify it, but we found quite a few of these problems. We put together a list of the significant stoppages that we averted because of this program. We tried to quantify this list as well, but we could not come up with a dollar value.

Program Manager's boss: The whole process was built around having the individual engineer who was responsible for a component or a system go in and actually participate in assembling that component. It gave the engineers a lot more familiarization than they would have had otherwise with how their parts fit within the vehicle.

Vice President: It allowed the engineer to check the finish, or the robustness of the fastener. Could he jerk it? Would it rattle? It allowed them to look at wiring to determine routing robustness. It wasn't a prototype that was here for three days and then off on another test; it was there all the time. If you had a problem, you could go make another part, and put it back on again.

Chief Engineering Manager: It was at either the EP or the VP harmony buck that the President came out to review the status of the program. I remember that he talked with us while sitting in the back of the harmony buck. He thought it was great to see this level of parts that you could look at for fit and finish issues, and to get the hard stuff out of the way before you cut production tools.

Content leader: We used it as our golden car. We kept it up to date. We put all the VP components on the EP, and proved those out ahead of time. This ensured much more up front quality of the electrical system and

The buck was almost institutionalized by the team, as this comment about it being the "golden car" implied.

we're continuing to use the VP car right now [September 1994].

The success of the harmony buck process for Epsilon did not always attract admiration from other program managers. Some had to be told by the Vice President to go to take a look at what Epsilon was doing.

Could it be that professional jealousy or competition among program managers limited recognition for this innovation?

Engineer in Program Management: It was a phenomenal success. That was evident in just the amount of attention it got from other programs. People from the Omega program, Kappa, one of the truck programs – everyone was dying to get in to see this thing. It was like a miracle we'd pulled off. And it really wasn't that hard to do. It was just common sense when you really think about it.

The harmony buck was integrated with other practices that opened technical communication with outsiders. Before each of the three builds the team held a big suppliers' meeting. They also had a process called "Must-See-Before": engineers had to visit the suppliers' plants and see the parts before the build. Several engineers suggested that these two processes made an enormous difference.

The second harmony buck: Approval and decline

A subsequent harmony buck was not as widely successful in terms of producing new insights as the first buck had been. The lower level of success of the second buck raises issues about whether the first buck raised expectations, found the problems to be identified by this innovation, or whether the social innovation for meeting and talking around this second buck wasn't there the way it had been with the first buck. The second buck was less of a "big deal" — it was a way of "doing business" for the team, rather than an innovation.

Can the success of the harmony buck be attributed to the team's learning efforts? The quotes thus far show that the link is subtle: At every stage, the buck was successful to the extent that people felt free to raise questions with each other, suppliers, and assembly people.

Engineer: We had to have a second meeting with Vice President to ask for additional money to continue the program. We found we couldn't quantify exactly how much money the Buck had saved us in time, labor, changing tools, getting parts, etc.

So instead, we walked [the Vice President] through the program. We got stories from every engineer: "Here's what we started with. Here's how we built the car." And we had a board full of photos to show the process: "Here's the engineer putting his own part on the car three months ahead of schedule." The engineer wouldn't be standing back, watching an hourly guy hammering the part on the car because it didn't work; the engineer would be putting the part on the car.

The second buck was more for the transition to assembly, yet the concept of a buck was more appropriate and innovative for engineering design.

"The benefit is just phenomenal," we said. "We know the EP came off so smoothly in large part because of this. We'd like to continue it through Job One. Doesn't it make sense?" They couldn't argue with it. He agreed, and they gave us funding to let us keep going.

Engineer, Car Programs Management: I don't think they were able to generate the same amount of information in the second phase of the harmony buck. They were not able to generate the same interest and involvement that we had gotten before.

When communication broke down, the advantages from the harmony buck also seemed to diminish. According to some interviews, that communication breakdown happened in this final stage. There was less emphasis after the VP build on drawing engineers in to test their components on the buck, and making them feel as if it were "their buck." The first harmony buck identified many issues, and because it was so effective, there were fewer surprises on the second buck. However, the harmony buck process is one of the most generally agreed upon "noticeable results" from this program (as listed on page 87).

Theme 5: Partnerships

Several Epsilon efforts could be seen as attempts to develop infrastructure to support communication and partnership between functions. Two efforts stood out in participants' minds:

- The market research clinic expanded the team's opportunities to learn from customers and dealers, and to pursue cross-functional conversations.
- The collocation effort showed how physically changing the infrastructure is not sufficient in itself. It's also necessary to develop new habits and attitudes to help people get the most value from collocation.

(Another infrastructural event, which is not covered in detail in this Learning History, was the three-day off-site meeting in 1993, where 150 Epsilon team members gathered to create a shared vision for the program.)

The market research clinic

The engineers designing the Epsilon were exposed to customers early in the development process. The impact of this effort was reflected in the interviews:

Body engineering manager: Most of the engineers are young and have no plan to ever drive a [luxury vehicle like the] Epsilon. It's not their type of car. Their mindset is oriented to [lower cost passenger cars]. How can you feel a certain allegiance to making this the best car in the world if it's just another car to you?

Yet, none of the people designing the cars ever drive any of those cars. Only the senior management types drive them. I don't know if it went anywhere, but [the Program Manager] was going to get a fleet of luxury cars for people on the team to experience, to get a sense of what this car was all about.

Eight of the predecessor cars to the new Epsilon were obtained so that team members could drive them overnight and on weekends. The cars were continually in use for the next two years.

Launch Manager: After we go through a number of ideation sketches and start honing in on some favorite themes, we usually go out and do some market research. We invite people to sit behind a mock-up of the interior and the trunk. In the past, we would have had a market research expert create a qualitative report. The bosses would have read it and told the engineers

The early market research clinics, an innovation from the Epsilon team, and use of luxury cars by engineers represented ways that team members could learn directly about what they were to create.

what to change.

Instead, we had our background in vision work. We had asked ourselves: "What do we want this car to be and how do we want to engineer it?" The best way to learn, in that context, is to take our 40 engineering team leaders and have them spend a week talking to the customers. "Why do you drive what you drive? What would you like from your next luxury car? What do you think about the price? What do you think we should do?" The specific answers were important, but less important than making our engineers feel connected to the customer. So when they're engineering those cars, they remember John, Bill, Mary, or whomever they talked to. They have a face in their minds; not just an engineering drawing.

Interviewees often talked about the primary benefit of exposing engineers to customers. What about the secondary benefit of the forty engineers and marketing people traveling together and talking to one another about design attributes?

Market research liaison: They took the Epsilon to a market research meeting way earlier than I would have done. At the time, I fought them on this event. I wanted valid research and I knew the engineers weren't trained to do that. But in retrospect, it was brilliant. The engineers couldn't have gotten any of the feeling they got by looking at a video or reading a book. I thought that was extremely valuable. That was the first time I had seen that done.

They also wanted us to have dealers and media present at the research and that worried me a lot. It's really a risk, because the dealers are very influential. Had they disliked the car, it would have really hurt our launch. Had the media disliked it, that would have hurt us too.

It turned out that the dealers and media made suggestions. I found out at the launch meeting yesterday that the team had responded to each one of the suggestions. When we bring the car close to market at the end of this year, I can go back to the dealers and say: "When you told us this, this is how we responded." Suddenly the dealers and key members of the media

are a part of the team.

Engineer: I went on the market research clinic in April 1992. I thought it was great that they let engineers go, because we got a chance to talk to the customers. People were complaining about sluggish performance in the way the vehicle felt. From the powertrain area we already knew that it might be substandard, and we had proposed fixes and they were gonna cost X amount of dollars. It was hard to sell that point to the program people, but after the second clinic I think the point came across. We weren't just talking out of our hats; this was real. When a customer agrees it's a lot easier to say, "Hey, maybe he's right."

Market research liaison: The Epsilon team members always attend research, whether it's theirs or ours. You expect them to attend their research, but they attend our research. They want to be involved in the launch. They want to be involved in the delivery. They want to have their people learn for the next program.

Once the engineers had met and worked with marketing people, they maintained the contact and interest well into program development activities.

Collocation: Opening a new realm of issues

Collocation is not as simple as merely putting everyone under one roof. Because of timing, Epsilon had not been designated to be collocated; thus, collocation came late to the Epsilon project. It was still a relatively new practice at AutoCo. It might not have been introduced, people agreed, if the learning organization ideas had not reinforced the need people felt for intensive collaboration across functional lines.

Epsilon approached collocation a little differently than other teams. Instead of being seen as a cross-functional team because members were collocated, collocation was seen (from inside the team) as a first step. Once collocation began, people were responsible for coordinating old loyalties, and protecting new ones, in unprecedented ways. In retrospect, many people on the Epsilon team expressed a wish that the collocation process had taken place earlier, and had included representatives from more functional areas.

Content leader: Some of the value of collocation is [intangible]. But there is a real advantage to being in the

Epsilon was one of the first teams to be collocated. AutoCo has learned from these earlier experiences that collocation is effective and full collocation is valuable.

same room with other people. If you need to ask someone a question and you have to call and they're not there, and they call and you're not there, that can go on for days.

Engineer: The politics disappear. You don't have to go through another layer of management to resolve an issue. You just walk over and have a one-on-one discussion. It also helped to have the major suppliers on site during launch: Prince, the body shop people, Motorola, etc.

Team leader: When you see somebody every day, just by human nature, you build a bond because you're in each other's offices all the time. You just become friends. Trust-building is really encouraged by this collocation.

Content leader: However, a lot of our problem is not related to proximity; it's the chimneys. We started buying donuts and coffee on Wednesday morning and having the team leaders hang around for people to bring problems to. That was not really working. People came in for donuts and coffee, but they still didn't say what was on their mind.

Collocation provided an opportunity for improving how people worked together. However, as this quote illustrates, other factors were needed for people to benefit from their proximity with one another.

People talked in the interviews about the effect of collocation in much the same way that they talked about the benefits of the learning labs. Could collocation have been effective without the work on "trust" and "openness?" Could trust and openness have been developed without the reinforcement from collocation?

Chief Program Engineer: When someone is afraid to tell you that he's got a problem, it doesn't matter if he's sitting in the next cubbyhole or sitting on the other side of town, he's not going to tell you. If he is willing to tell you, there's an advantage to being together because a lot of time is spent in the halls talking together, instead of formal meetings. There's a great power in collocation, but collocation doesn't fix the lack of openness and honesty.

To what extent were team members affected by the ability and influence of the team managers to get top management to support their request for collocation, despite the fact that it wasn't planned or budgeted?

Chief Engineering Manager: It was a late collocation. There were a lot of people resisting it. The program is already off and running, they said. We can't collocate every team. Maybe this is one we don't collocate.

In fact, we never really did

assemble a fully collocated team. Our team was halfway in between. Those who felt strongly that it would be beneficial moved in. Those who didn't want to move resisted. But I noticed that those members who were collocated got things done faster and smoother.

Content leader: Collocation hasn't helped in any substantial way. When [the Program Manager] was trying to pull all this collocation together, we resisted until we were basically kicked out of our building. We came over here because we had no other place to sit.

Collocation pulls some people together, and still leaves some people out. Does the benefit from collocation depend upon the interdependence of the work of the people that are being put together?

If people didn't see the interdependence among particular teams, for example sheetmetal with trim with vibration teams, it was easy to miss the value that collocation provided.

The reason I resisted was this: 90% of my parts in body structures are made by Metal Stamping Operations. Those suppliers were right in our old building and sat next to my engineers. They were the tooling experts. When somebody asks us for a change, we have to ask an expert: "Is this feasible? Can you make it?" Now, after collocation, that tooling expert is a 15-minute drive away.

As with the other innovations in this program, there was initial resistance by both those whose approval was needed (top management) and by those who were affected (engineers).

Team leader: I was in the experimental vehicle garage when the team collocated. It was interesting because most of the vehicle development people did not want to collocate. They were set in their ways; they had the EV garage, and all their creature comforts. Who wanted to go to another building? From my perspective I saw a lot of resentment. This was not going to work.

And, as in the other cases, people only recognize the benefits afterwards.

But over the last couple of years I've seen a lot of positive changes. My manager asked vehicle development [in summer 1994] how many people, if they had to do it over again, would prefer to have stayed in the garage. I think about 2/3 of the people wanted to be here. That proved that in the end, the team cooperation that you get when you're all under one roof, along with the dedicated facilities that we had, really worked out well.

What is it that leads program managers to continue despite the resistance they encounter at all levels?

Theme 6: Process innovation in the context of a large organization

In creating a new approach to managing the Epsilon team, one deliberately different from traditional AutoCo management culture, the leaders came to feel isolated from the senior levels of the AutoCo system. The nature of competition at this level left the Epsilon program manager feeling isolated. The launch manager, while he received many inquiries from others as to what Epsilon was doing, felt the inquiries were superficial because most people appeared to be waiting to see how things would turn out.

From team leaders' point of view, senior managers sometimes applauded them, sometimes supported them, sometimes ignored them, and sometimes invalidated their efforts. Team managers tried to explain their approach, but they did not gain sustained interest or attention from senior managers. The Epsilon team coped with their position as innovators by assuming that their excellent results would make them popular, influential and acceptable, legitimizing their alternative methods. Unintentionally, this became a strategy of isolation. Yet Epsilon was, from the beginning, inherently dependent upon the larger AutoCo system.

This section of the learning history shows how an innovative team like Epsilon needs an advocacy from above that fulfills the spirit, not just the letter, of mentoring. Since their work, by definition, challenges the established rules of the game, the team needs help anticipating potential frustrations and roadblocks, in time to find strategies for managing them well. They need safe, open communication channels for raising difficult questions — and committed advisors who can help stop them from going off into organizational dead ends. In the end, the tension raised by an effective change effort represents opportunities — for improvement within the team *and* in the larger system. It is a challenging task to take advantage of these opportunities, but we hope future teams can build effectively upon Epsilon's experiences.

There is a temptation to view this story as a “David vs. Goliath” narrative — “the innovative team versus the rest of the organization.” However, that is just one perspective on a fairly complex and multi-faceted story. In reality, nearly every participant felt that Epsilon accomplished a great deal but could have achieved much more.

The Epsilon team leaders explicitly hoped that they might recreate their positive team experiences and results in other AutoCo settings, and were disappointed not to be given that opportunity. Others in the company felt that the difficulties perceived by Epsilon team leaders were just that — *perceived* difficulties. They felt that Epsilon was appropriately supported and nurtured, like all AutoCo teams.

Positioning the purpose of the team

Before his assignment as the Epsilon program manager, the Program Manager held leadership positions with other vehicle programs, including the Planning Manager for the original AutoCo Theta Program (see footnote on the Theta Program, page 19). These experiences strongly influenced his goals for the Epsilon team.

The priority of these goals were consistent with what was expected by top management at AutoCo.

Program Manager: I wanted to accomplish three items on this launch.

The first was to make the car as good as I could get it, given the total program constraints.

The second was to run an orderly, "no-surprises" program.

The Program Manager's goals did not explicitly include impacting the larger organization — but is it inevitable that anyone who succeeds at process innovation will want to see that success replicated on a larger scale?

The third was to take a better approach in managing people in the product development process.

What I really wanted to accomplish was to build a team like [the Theta car manager] built on the Theta in 1984. I was on that team. He didn't know any of the tools or theories that we used on Epsilon, but he loved us. That was important! He created an atmosphere such that no matter how he yelled at you and what he did, it didn't matter because he loved you. And I thought if I could ever build a team like that, that would be the crowning touch of my career. But I wanted to create that spirit through a reproducible process: one that we could spread to other teams without relying on personalities.

The management style of another process innovator influenced the Program Manager.

Those three main topics are exactly what I described to the team at our very first meeting. I never changed those goals.

People often talk about what its like to be on a "great" team; it seems to be a basic human experience which people then often seek to replicate.

Engaging senior management

In retrospect, some critics of the Epsilon project have suggested that senior management should have been more involved from the beginning. There should have been more attempts to help senior managers understand the theory, tools, and process associated with the five disciplines of a learning organization (see Senge, The Fifth Discipline, 1990).⁵

⁵ This learning history is focused on the Epsilon project, its learning initiative, and the issues Epsilon's efforts raises within AutoCo. In reading this document a number of people have surfaced questions regarding the relationship of the Epsilon team with senior level managers and the structure and

As early as the Project Engagement Clinic⁶, in September of 1991, when the Epsilon learning effort began, this problem was discussed. Chris Argyris, Professor at the Harvard Business School and an advisor with the MIT group, cautioned that the strategy of excluding senior management from direct involvement might not be effective in the long run. The Program Manager and the Launch Manager were aware of this problem. They said that AutoCo culture generally valued results over theory. They felt they could not effectively include senior managers in their effort, or even talk much about the MIT “learning” theory, until they had some tangible results to demonstrate. This put an extra burden on the team, because if results failed to measure up, it would call their approach with MIT into question.

The strategy that was broadly advocated at AutoCo was to take the risk by trying a new approach, and justify it retrospectively with the results it produced.

That early decision, which drew a “curtain of silence” around the Epsilon team, may have been more damaging in the end than team members expected. Of course, it will never be known what would have happened if the team had opted for a more visible approach, and had not moved forward until senior managers understood and accepted the new way of working. It might have limited the amount of innovation that was allowed.

When is the most appropriate time to engage senior management?

What dilemmas are set up for managers when in order to get support for trying something new, like process innovations, they have to point to hard results, yet in order to produce better hard results, they have to try something

Program Manager (at project definition clinic): There have been lots of studies in the company in the past that have highlighted the fact that there’s a lot of fear in the organization.... Nothing changes [peoples' minds] unless you have data [“noticeable results”].

Researcher: But this is not a problem of data. You will never be able to present hard evidence of a causal link [between your innovations and the positive results]. It’s epistemological impossible.

Launch Manager (later): We were asked: “How much success can you have without involving senior management?” We admitted we didn’t know. We were asked whether we would be prepared to involve them at some point in time. I think our answer was that when we thought we were ready we would be prepared.

Program Manager’s boss: I think one of the reasons that we didn’t spend more time trying to get management support is because this process was “soft”. As I look at this project and other similar projects within the Company, the people involved are clearly very,

support of the MIT research effort in engaging executives. The Epsilon managers were unable to get top management involved in learning project issues. The team requested, several times, that the OLC help them design a process of engaging the larger system. Involving busy executives, working with a research group in engaging executives, as well as the efforts of the learning history process itself in capturing and addressing these issue is the subject of a subsequent research document.

⁶ The project engagement clinic occurs as part of the process of starting a research project. A set of interviews were conducted with managers on the team to surface known issues. A document with a summary of these interviews was then read by the MIT researchers and Epsilon managers before attending the clinic. The clinic provided an opportunity to test the ability and willingness of managers to engage with challenging issues of the kind that surface through learning initiatives. An important issue raised in the clinic was the implications of conducting research while trying to accomplish business objectives. For more details and list of questions on clinic, see page 4.

new?

very large supporters of the process. Our approach was to let the results speak for themselves basically and not go out and try to preach the process because it could be viewed as soft.

Program Manager: I tried to be the buffer for the team. I said, "Wait until the results come in. When they see the results, they're going to start asking "How did you do that?" Then they'll be ready to listen".

But when results began to show up — as new records of achievement for the program — it was still difficult to talk about the learning effort. Achievements were not acknowledged in the ways that Epsilon team leaders had hoped for. And the link between process work and engineering results seemed to go unrecognized.

Program Manager: I brought the team leaders in [for a one-hour presentation to a group of senior managers, including Vice President, on the value of the learning effort]. We told them all the things we had accomplished, what we had done to accomplish them, and how much poorer we believed our results would be if we hadn't tried to do it a new way.

Senior managers who preside over an innovative pilot project need an effective way to educate themselves in the process that the team develops.

Two things seemed to make an impact on [the Vice President]. One of our engineers' stories dealt with immense reductions in development time because of what she had used from the learning lab.

Secondly, my perception was that he never got the chance to hear firsthand stories like this. They were always filtered through so many levels of management that this was a rare occasion to hear directly from the people who did something. I think that had an impact in itself. And we got him to agree to money for more training.

The Vice President's response was not seen by the Program Manager as an endorsement, but rather as a missed opportunity to see or accept the linkage between the learning approach and product development accomplishments.

Nonetheless, at the end of that meeting he told us we could keep working on this stuff. But we shouldn't let it get in the way of our real jobs. After that comment, I didn't think I had a prayer of convincing [the Vice President] other than with hard data and results.

Evaluating Epsilon: Miscommunications and misunderstandings

After the team leaders fell into a pattern of relative detachment from the larger system, there were recurring misunderstandings. In some cases, Epsilon team leaders thought the larger system was micromanaging them from above (as they did with other car programs): dictating requirements that didn't really apply to Epsilon (they thought) because the new management practices made those requirements obsolete. Other car programs also thought the requirements were obsolete, however, for lack of better measures, Epsilon and others continued to comply with conventional reporting requirements.

At the same time, some of the senior leaders perceived the Epsilon team as withdrawing into its own "true believer" approach — as if Epsilon leaders felt that they knew how to achieve results that the rest of the AutoCo organization did not. To senior leaders, the jury was still out on Epsilon's "unique successes." They did not know whether it would do better than other, more conventional teams.

These misunderstandings seemed like separate events, but over time they built upon each other. Epsilon leaders began to feel that the system would not let them communicate their ideas up the hierarchy. Their attempts to tell top managers did not meet with the enthusiastic reception they had hoped for. Instead, they were told not to let these efforts "get in the way" of their real jobs of producing a car. Some within product development worried that the Epsilon team might be falling prey to "group think."

Internal consultant: I got the impression that [the Program Manager] and [the Launch Manager] were saying to themselves, "We did everything right. Everybody at AutoCo should be pleased. When they see the results, they'll knock the doors down trying to learn how to do what we did." I'm sure I shared in some of that attitude myself.

But in retrospect, that was a naive approach to the world. If we expected results and teamwork, in themselves, to communicate our message to the rest of the company, then we were setting ourselves up for disappointment. And we have to be careful not to blame everyone else for not recognizing us in the way we hoped they would.

Change, by its nature, is painful. It means going against the flow from beginning to end. And results are always more ambiguous than we'd like. Perhaps we should have prepared for the ambivalence that outsiders would feel,

and adopted less of a missionary attitude about what we were doing. On the other hand, if we hadn't taken that attitude, we might never have begun the learning effort.

As the Epsilon managers and engineers learned to be more open and direct among themselves about the problems they faced, they began to practice some of that same behavior with their bosses and others outside that team.

Program Manager (in an interview in mid-1993): I have taken to discussing problems openly with my boss and the Vice Presidents. In one meeting, I told an Executive Vice President that there weren't enough resources on this program from body engineering. I showed him how that might jeopardize the program, how we were trying to recover, and what the risks were.

I felt the VP wanted to hear, instead, how we would make it with the head count we had. A year or two before, I would have told him what I thought was politically wise to tell him. This time, I was telling him what to realistically expect up front.

He seemed to think I was being uncooperative. But other people in the room – the Vice Presidents from chimneys I had to work with – responded more positively. Later, when I called them and said, "Hey, I need a hand," they helped in a way that they've never helped before. I think it's because they remembered me as having talked to them candidly.

As it happens, we did make up the problems with the existing head count – because of process things we did that had nothing to do with what the rest of the company was doing. In essence, we did what that Executive Vice President wanted us to do in a way he didn't expect.

Changing in a larger organizational system seems to require both an awareness that individuals are part of a system and that changes need to occur at individual and organizational levels.

What conditions are required for individuals to change and behave differently? What changes in organizational conditions are required for others to try new behaviors?

If I were 35 years old and worried about getting promoted, I couldn't have taken those risks. I guess I'm old enough and I've been around long enough, and some of the things I was doing with MIT were changing my mindset. The change wasn't with them. The change came from within me.

Why is it a risk to be open in a large firm like AutoCo? What are the implications for improvement if being

open is risky, and only a few people take those risks?

How might creating conditions of openness and risk-taking in one group, within in a large organization, be interpreted by others in that organization?

But can you find a way to talk about individual change without seeming like a "cult" follower?

Does the language that was used to describe the tools for learning become a barrier for others when it isn't understood?

When the orientation for managing the product development process shifts from managing and controlling to learning and coaching, how does using a new language to describe these activities affect other people's perceptions of them?

The learning approach was based on a philosophy of openness and acceptance of differences.

What happens when the patterns associated with traditional management process, that of seeing behaviors as either right or wrong, slips into how a learning approach is practiced?

Vice President: I felt a bit like "the outsider." It became almost cult-like to me. People would sit in meetings, look each other in the eye, and talk about the "ladder of inference." Meanwhile, I was trying to run the business. I had a lot of tough decisions, and I was very tight on my time.

It almost seemed that the tool became more important than the end result. The team became so process-driven, so mechanistic, so much like disciples of Peter Senge, that I think it got in the way of what they were trying to do. I know the team would disagree, but that was the view from the outside. There was critical time spent away from work in some cases.

When I was there as a senior person I got the impression that they were letting the process overwhelm the solution of their problems: "Make sure we follow this process, so we're aware of what we're doing." I think that's dangerous when that happens. That is my only personal experience with the whole thing. To an outside observer, if you weren't part of it, and if you didn't buy into all this, you were wrong and they were right.

I think that's where training can go bad. If you're going to expect performance changes or behavioral changes from the people you interface with, then you better make sure those people go through the interface, the process with you, at least so you understand it. Had I known it was going to be as broad and deeply spiritual as it seemed to turn out to be, then I should have been a part of it. We shouldn't have done it unless we all agreed to go through it together.

Did I ever call [the Program Manager] on this? I don't recall any

specific discussion.

Learning approaches are based on developing a level of skill in conversation and inquiry. What happens when people in other parts of the organization, those that are relied on for support and approval of resources, have not had exposure to learning ideas or an opportunity to learn those new skills?

Implementing the new CR policy

Traditionally, AutoCo programs tend to be judged by the numbers. Ultimately, the final measure of success is in the market place, but prior to launch, product development progress is closely monitored. Metrics are the major form of communications between program teams and senior management; the major way for senior managers to see if a program is meeting objectives and “under control.”

When a team innovates to improve the process, it changes the rules, as Epsilon did. This stymies the rest of the organization’s ability to measure the team’s progress. If the rules are truly changed, then traditional measurements lose their effectiveness. These traditional measurements were ways to check the progress of the program at intermediate points in time — a way of being able to predict how the final product would turn out. The Program Manager and the Launch Manager assumed that if they had the people process right, some intermediate metrics would be spectacular, while other metrics would become less relevant. In the end, they expected to produce a great car, whatever the intermediate measurements predicted. Unfortunately, these attitudes about metrics were never explicitly talked about.

If metrics like the Quality Operating Systems (QOS) are necessary to ensure that all the programs develop a disciplined level of quality performance and predictability, what is the best way to apply those metrics to programs like Epsilon, which are innovating new ways of articulating measurements?

If process innovations in technical areas initially involve behavioral changes, how would a system of quantitative metrics capture and reflect these innovations?

If the measurements don’t reflect the effect of the innovations, does that mean they are not effective?

Program Manager: We brought the Vice President in early in the program. In our very first meeting with him, we down-played all the normal predictor charts. We talked for two hours about all the processes that we were putting in. We told them that we were meeting all our indicators, and everything else was on time and under control.

But when we left the meeting he reportedly said to [the program manager’s boss’ predecessor] that he was worried about us. We were doing all this soft stuff and we were going to lose control of the hardware. We were not going to deliver the hardware.

When I heard about that from [the program manager’s boss’ predecessor], I didn’t feel threatened by it. I was disappointed. Isn’t it unfortunate, I thought, that he can’t understand what all this means. But we would show them; we wouldn’t lose track of anything.

This problem came to a head around “change requests” or “concerns.” (At AutoCo the two terms are synonymous.) Concerns are documentation of issues, problems and impending changes on a part. They include lists of associated parts that might be affected by changes in the original part.

Back in 1992, the core team had determined through systems analysis (see system map in Sidebar: The core team’s view of “parts behind schedule” on page 14) that engineers were slow in reporting concerns because of the unwritten expectation that they should resolve concerns shortly after they were logged. The Epsilon Program Managers decided that it was better to have engineers report concerns as soon as they knew there was a change in a part, and not punish engineers who reported concerns that they could not close out quickly. This would allow better coordination among changes in parts and lower overall costs because fewer late tooling changes would be required.

This account of how change requests were traditionally used and perceived by engineers and managers, and then the change in the ways in which the Epsilon team used the change requests and associated reporting system, illustrates the influence of reward systems on behaviors.

Engineer: The change request (CR) account reached 500 because we were all encouraged to bring our issues out and to stop keeping them on the hidden log that every engineer has. In the past, engineers would keep a hidden log of their problems until they knew the answer. Then they’d put them on the CR with an answer at the same time. To say we were not rewarded for revealing CRs would be an under-statement. Typically more than one person would be trying to solve the same problem. And a lot of people would do a lot of different things, not knowing what each other was doing, because there was no common document out in the system that tracked the problem.

I might be working to solve something and it might involve sheet metal. The sheet metal people wouldn’t know because I didn’t have it on a CR out to the world. I might not even have known it effected sheet metal. When I wrote the CR, they might say: “Wait a minute. We can’t do this. It effects us.” If I had known that a month ago it would have changed my solution.

A change in the use of a metric, like change requests, by one group affects the other groups that use the metric.

Thus, with this process we were encouraged to get CRs out in the open sooner. This meant everyone else understood that you knew what your problem was and what you and other people were doing to follow it.

Other programs may require every CR

Is it required that other groups share the perception of what a metric means and how it is used? What does this require of groups who seek to innovate processes?

to go through the Program Manager. [The Program Manager] empowered us to handle our own problems; if we couldn't handle them, we brought them to him. But he didn't need to see them all.

Not only were [the Program Manager] and [the Launch Manager] strong proponents of getting the problems out there on "pink" [the color of the change request form], but they went to bat for us. My organization, body engineering, is very meticulous about tracking CRs and how long they've been on pink. We have daily meetings on this. It can become very punitive when you have a problem out there for a long time.

[The Program Manager] and [the Launch Manager] went to my organization and said, "Look, we're telling our group to get the problems out there right away. That means they're probably gonna be on pink a little longer." That circumvented a lot of problems.

Epsilon is "Out of Control!"

At Epsilon, the dilemma about how to use intermediate metrics escalated. This gradually led to a judgment, by people outside the team, that the program was "out of control." This was one of several circumstances in which Epsilon managers and senior managers elsewhere in the corporation held different interpretations of events.

This comment comes from an interview conducted at the same time that the "out of control" perception was brewing, by a manager outside the Epsilon team.

The information in metrics is used by many different managers to interpret how the program is evolving. Given the changes in how the team used the metric, what are the implications for how others use that information?

Job One was not moved back six months. The car eventually came out on time, and without the "heroic effort" that is typically required at the end.

Assembly Launch manager, interviewed July 1993: In vehicle operations we have a metric that starts with green and goes to yellow and goes to red. Well, I called [a top manager] the other day and said the Epsilon program was "purple." "That's the other side of red [worse]. The other side of red! You can make all the processes you want, but there's zero substitute for experience!"

The Epsilon isn't ready. I've had a unique ability for the last 10 or 15 years to say what's ready and what isn't ready and be right 95% of the time. The patient is terminal. My recommendation is to move Job One back six months. If that's not an option,

then move Job One back six months after you crash.

What made it difficult for Epsilon managers to explain their strategy and behaviors in promoting early reporting of concerns? Why, after those explanations, did the top managers continue to be troubled by the large number of concerns they saw on management reports?

What sort of agreement or "buy-in" is appropriate to ask for from other organizations and non-team members ahead of time in unproven experiments like this?

If managers from other functions argue convincingly that a program is in trouble, is there anything that can be done, outside of "heroic" efforts to bring the program back on track?

Through his approach to change control, the Program Manager let engineers make their own decisions about what changes should go into the system. He trusted them to put in only what was important, and interpreted the numbers as an accurate reading of the state of the vehicle.

It was difficult for the Program Manager to communicate the significance of the way that Epsilon used the CR system. At a program review meeting in March 1994, the Program Manager described his approach.

The way the Epsilon team used the change request metric was the opposite of the way it had been used by others. The more change requests on the system, the earlier they were logged, the better.

The results from early prototype builds, and logic behind the way they were using change requests, seemed to indicate that the innovation was sound and successful in achieving results.

Content leader: Initially, I felt really good about our CR count. We were using the Engineering Release computer system, which made it much easier to write a change request.

Since these were pre-release CRs, management wasn't following them. There was no threat to engineers. It was fantastic to find out about all these things and have them documented, and then have them sent over to car management people and designers. We knew that they were being worked on, and we could manage them daily right there in the Engineering Release system.

Program Manager: In the normal course of events there are anywhere from fifty to a hundred and fifty concerns in the system during a program. We went into that meeting saying, "We have five hundred concerns, and that's good. We've encouraged engineers to tell us when they have a problem, as soon as they have a problem. Based on the measurements we were using, we've had the best quality evaluation prototype that we've ever had in our history, so we were able to do concerns that involved fit and finish earlier than in the past. We're closing concerns early enough so that they're still in design without affecting hard tools. They're costing thousands of dollars to close instead of millions."

I went through our accomplishments at the meeting. We had an exceptional MP [mechanical prototype build] and EP

[evaluation prototype build]. We met our quality goals, and we had the highest number to date for MRD [delivery at Material Requirement Date]. "By encouraging engineers to write concerns," I said, "We're actually getting work done earlier and we'll have a better quality product. This is a change in our system and we want to keep it that way. We want it to be not punitive for an engineer to write a concern early."

[The Vice President] nodded and listened. But after the meeting he still said the program was out of control.

Vice President's perspective on the same incident is at right. To him, the Epsilon project was not "out of control": it was simply going through the normal expansion and retraction of changes.

Vice President: I wouldn't have found the change request situation to be that unusual. It wasn't too different from Kappa [another program]. The Epsilon program was not out of control in my view. Nor did I find the experience of working them back down to near zero again any different from what the Kappa team wanted to do.

The ethic that everybody was trying to follow was: There is a right time for change and there is a wrong time. You ought to be following a curve where changes get less numerous as you get closer to Job One. Before every milestone, such as a prototype build, you're bound to have many changes where people rush to get things in to make a prototype date. One of the biggest problems that any Program Manager has is to get hidden changes and problems out of engineers' desks onto the top of the table, so you can get them resolved and closed. That pushes the number up.

This description of how the change request system was used illustrates the expected behavior of program managers — to perform managerial tasks so that the change request system follows a predictable pattern.

Then you just have to hammer them back down again. This would have been eight months before Job One. At that time, from a manufacturing standpoint, the engineer's job is done. Now let us do our job in manufacturing, and you keep the product stable — which means no changes. Let us optimize the product and process so we have a quality launch. That's the way it's supposed to work. It never does work that way. The

engineers keep changing things, most of the time for good and relevant reasons. Manufacturing drives some changes. And so you have this constant battle of late failures, problems, fits, and finishes. There is huge pressure from Assembly to drive the change count down.

When would the Epsilon program eventually have to succumb to pressures to get the number of CRs lower? With what consequence?

Freezing and reducing the change requests

In March, 1994, under pressure over the high number of change requests, the Program Manager and the Launch Manager instituted a change in procedures. Engineers were told to stop everything else and resolve changes. During an intensive weekend, the engineers reduced the number of open CRs from 350 to 50. At the time, this enhanced the program's reputation. For example, at least one senior Assembly manager gave the team a lot of kudos for driving the changes down to a lower level. In his eyes, a program that had been a disaster now inspired confidence — the metrics were what they were expected to be and the Program Manager had demonstrated that he was in firm control.

Ironically, however, the appearance of solving problems early may have contributed to an outbreak of late-breaking problems when changes that had been "pushed underground" resurfaced later in the game.

In physics there is a law which specifies that any action is met by an equal and opposite reaction. Does this law from the physical sciences also apply to the behavioral arena of managerial action? What is the reaction to pressures to reduce CRs?

Content leader: But when management takes that approach you drive your engineers underground. Nobody will write a CR that they don't have a solution for if they know that their supervisor has been told to come to them three times a week to ask them about their open CRs. The engineer won't tell me about it. Thus, after we got through the VP build it reverted back to the old "hidden" system.

Program Manager: Instead of calling them concerns from then on, we called them investigation issues or some other name so we could identify what the concerns were. That's better than nothing. But that's not what you really want.

What you really want is for everyone to know that there was an issue. Once the whole company system knows a concern exists and it's a problem, they can all think about, "Now, how might

that affect me?" Everybody can work on it together.

I even went back to [the Vice President] and said, "The magic of this system is we capture everything, I mean everybody knows about it from the day we capture it." He thought that sounded terrific, but he still didn't like open concerns!

The early retirement

AutoCo announced major organizational structure changes in April of 1994. These changes were part of efforts to become better at managing in a global marketplace. One effect of these changes, as they began to be implemented in the summer of 1994, was that many (approximately one third) middle managers no longer had positions. In August, the Program Manager was informed he would not be promoted, and he was given an option of taking another assignment at the same pay, or early retirement. He chose to take retirement. The Launch Manager faced a similar situation and chose the same option several months later.

When executives were asked, they said that these events had nothing to do with reactions to Epsilon's achievements, and nothing to do with the perceived effectiveness (or ineffectiveness) of the MIT learning efforts. However, the early retirement options occurred around the same time that the Epsilon project was garnering notice in the press. The Program Manager and the Launch Manager had begun to give speeches about the process, both inside AutoCo (as part of the activities promoting the car to dealers) and outside AutoCo (often at MIT-sponsored events).

The early retirement was taken by many on the team as a signal, albeit unintentional, that the Epsilon effort, and its process innovations, had not been valued as much as team members hoped. Epsilon team members struggled to understand what it would mean to their future. Other people in AutoCo began to wonder if the Epsilon team performance had been as high as the hype suggested. And there was general confusion about how much AutoCo could learn from the Epsilon experience. In hindsight, the timing of the reorganization sent signals which senior managers wish had been interpreted differently.

Program Manager: I never expected to be offered early retirement. I still expected that, after seeing the hardware results we got, they'd see that we did something special. When [Vice President] called me over for a one-on-one meeting, I thought I was going to get promoted. Instead, he gave me a brief presentation, told me I would be demoted [sic], and handed me a

Is it possible to go through a significant change effort, in which you come to believe you are special because of your achievements, and still keep the

perspective of the traditional organization? In this case, does success simply mean, "You've done your job?"

retirement package for consideration.

I honestly don't think there were any ulterior motives. They simply didn't regard what we did as so special. They had to reduce the total number of people at my level. They wanted engineers in the program manager roles. I'm not an engineer. I don't think they thought about it as a terrible thing, or even as hurting my ego. They created a formula and then carried it out. They just didn't think we did anything much different from what any other team does.

Once given the opportunity to speak and act freely, will people assume that a continued atmosphere of trust and openness will protect them? Will they grow accustomed to speaking and acting freely even in other environments and circumstances at AutoCo? Will they be frustrated if they feel they cannot speak and act freely?

Content leader (interviewed September, 1994): I'm not sure that we are coping with it very well. There are a lot of morale problems right now. We were so proud of what we had been able to accomplish, and then to get slapped in the face like this. It was like every one of us got fired. We feel unappreciated and totally demoralized. Yeah, we were extremely successful. We made a lot of breakthroughs. We had tremendous success in just about every measurable you want to throw up. Then to be told that "You really didn't do anything special at all. Oh, and incidentally, we're getting rid of your Program Manager," was a terrible experience.

Senior executives confirmed, in interviews, that they supported the "learning organization" efforts. Why, then, did their actions present another picture to Epsilon team members?

Vice President: I think the team spirit witnessed itself in a negative fashion once it was clear that [the Program Manager] was going to retire. The team reacted not as individuals, but as a unified group. Individuals were not incensed; the whole team was very upset.

One hypothesis suggests that the lack of time spent by top managers and Epsilon Program Managers in trying to understand each other was a factor. The misperceptions built on each other and rippled out to other participants in the AutoCo system, both inside and outside the team.

A lot of people associate some of the disadvantages that the people on the Epsilon team went through with the MIT organizational learning experience. That is absolutely wrong. The perception is not correct, and we ought to correct it.

Program Managers' boss: But we haven't done anything to correct it. All we did was take the top guy and hammer him.

That's what people see, and they believe what they want to believe.

I think we have taken a major step backwards from showing the people out there that we support improving our processes. They are all scared to death. When the reorganization started I argued that we ought to take some overt action to support organizational learning. Because we have a lot of people who are unsure whether they ought to be involved in any of this kind of stuff.

In the end: Assessing the influence of innovation

Vice President: If the MIT course is something that everyone should be following, then I think not only I, but each of the vehicle center people ought to go through it. And we at [the top management staff level] ought to adopt it as the process that we want to follow for human interaction and team building.

With all the different consultants and approaches used in an organization, how do top managers most effectively learn to evaluate them?

But if it's equivalent to other approaches, and there are 40 different ways to achieve the same results, then it's not so important that we choose that specific course. I don't know which is the right one. That's the confusion factor.

Does the anecdotal data from the Epsilon team mean that organizational learning will be effective in other circumstances?

Program Manager (interviewed October, 1993): We'd like to get the kinds of things we're learning to spread in the company. We're showing some of the new Program Managers what we've done and the results that come from it. We're saying, "I know this works. It has worked for us. It has changed our teams. I have lots of hard data and war stories from different people." I don't know really what to do to spread this any further, but I know if I could it would be an extremely valuable tool for the AutoCo.

Launch Manager (interviewed in April, 1993): Maybe if the Epsilon were successful, there would be more open minds about the approach. But my great fear is: Others may not have the patience, inclinations and discipline to go through what we did. They'll expect a two-day learning lab to produce a miracle. They'll see no miracles, so they'll say, "That stuff doesn't work," and go back to the old ways.

Others on the team weren't convinced that the "learning organization" approach made a critical difference. It was experimental, not that clearly explained, and there were concerns that it could be

Process leader: We haven't made a strong enough case of the benefits to cause anybody to say, "Let's do this on every team." Worse still, they don't understand that "doing this" means developing your own systemic

reproduced in other teams.

As they reflected on the program, corporate executives recognized the conditions that Epsilon was saddled with at the beginning. In the complex business of developing vehicles, it is difficult to separate the effects of initial conditions from the benefits of an innovative process.

It is difficult for top managers to compare qualitative results from one program to another. They see the overt behaviors of the team. But the way that Program Managers are asked to report their progress and results makes it difficult to assess if an innovative process had any impact.

understanding, commitment, and in-depth reflection. They see "doing this" as: "Okay, you guys have tested it. Now we can go train everybody." So, we have not done a credible job of presenting the learning process.

Program Manager's boss: I spent quite a bit of time with the Epsilon program team. I was able to watch their progress versus the progress of the other programs for which I was responsible. I saw a measurable difference in the way the Epsilon team went about their business, and the way they reached closure on issues. The interaction of all the cross-functional participants on the team was much more supportive, much less confrontational, and much more focused than it was on the other programs.

I guess I didn't see the same kind of chemistry relationship on [other successful programs such as] the Kappa. Kappa had excitement and cohesiveness, because everyone rallied around the product, and it is a great product. It was more difficult for [Program Manager] to rally people around their product. Yet I didn't see the same type of positive relationships within the Kappa team. As far as I'm concerned, it is important for us to have those kinds of relationships on every launch team.

Vice President: If you look back at the history of the team, they went through some periods where they went over cost. As a team they pulled it back to objective and in the end they've even beaten their objectives. [the Program Manager] went through some very positive sessions with suppliers to make sure that each supplier knew what he had to do, specifically when he had to deliver it, and at what quality level. He literally took 500 people in massive meetings and made sure each and every person understood it. That represents involvement of the people you depend most on.

Whether all this is Senge, or whether it sprang from the MIT

experience, I don't know. But I think you could only do that if you operate as a very strong team.

The final situation of the Epsilon vehicle development effort and associated learning program leaves open many questions. Can a company, through a learning process, ask people to change their attitudes? What does real learning require? If it involves personal and internal changes, what is the role of a company and the work environment in promoting these changes? Can they be legislated? How are conditions created where people can examine their attitudes and make choices for themselves and others about their attitudes? Beyond the realm of individual change, what is needed for an innovative team in terms of support and understanding by the larger organization? If teams proceed, led by innovative managers, without establishing an organizational context for their efforts, what consequences might be anticipated?

One of the organizational consequences for the Epsilon team was that their accomplishments were not recognized by AutoCo in the way they had anticipated. What are the implications of organizational consequences for individuals who have been engaged in a learning process, perhaps undergoing some personal changes, and then finding themselves, as the project ends and they move into new teams, back in the more traditional environment?

As the vehicle went on to manufacturing, it experienced what were anticipated issues in reconciling design versus manufacturing concerns. These issues, although described at first by some long-standing AutoCo observers as taking place in an atmosphere of crisis and heroics, were ultimately seen as much less traumatic than those of most other programs. The initial response from manufacturing was explained by design engineers as the typical “posturing by Assembly.” In the end, the plant manager commented that Epsilon was “the smoothest launch he has ever seen,” said shortly before he too left AutoCo to retire.

The tangible evidence for a sound design and development process came when the production began *one week earlier* than planned. The factory had all the parts at hand, they were of the expected quality standard, and fit and finish concerns had all been addressed. The quality (from the syndicated independent research on competitive new vehicle quality (CNVQ) showed a 30% improvement in quality, a 9% improvement in satisfaction and a 50% improvement in surprise and delight features. The warranty data was not as good news, as it initially got worse than the previous car.

Also significant, perhaps, is the fact that organizational learning continues to be discussed in depth at AutoCo. This report, in fact, was commissioned as an effort to capture the Epsilon experience — so that the triumphs, along with the questions and difficulties, would not be lost.

Epsilon's Noticeable Results, 1991-1994

This list of noticeable results was collected and amended through the learning history interviewing process. People were asked to comment on items, their significance, and if it was familiar, describe how it was accomplished and what if any role they and others they knew had in it. The items in this list are observable events or objective measures which provide data on Epsilon program progression.

- **Mechanical Prototype (MP) build (8/91):** The Mechanical Prototype is a production level prototype for the underbody and front end of the car. The Epsilon MP design represented a considerable stretch from the previous AutoCo vehicle; it incorporated multiplex wiring, all new suspension and accommodation for electronic navigation systems.

In part as a result of earlier delays the MP drawings were *sixteen weeks behind*, but the first MP build was completed *only four weeks behind* the original schedule. The quality of the MP prototype build and maturity of its design allowed extensive testing to be done much earlier than is normally possible.
- **Team collocation⁷ (10/91):** Although the Epsilon team had not been designated to be collocated, program management pushed for it. The Epsilon team collocated 37 months before Job One (the date when production manufacturing was set to begin).
- **Stage 8 Market research clinic (4/92):** Forty engineers from the development teams participated in a market research clinic in California. This was said to be the first time engineers formally talked directly to customers this early in a vehicle program.
- **Harmony buck complete (1/93):** The harmony buck is a mechanism to review early designs and design changes prior to the periodic prototype builds. The harmony buck was an idea proposed by engineers on the Epsilon team. However, the \$2 million cost to build a harmony buck was not covered in the Epsilon program's budget. Program management supported the concept and lobbied Senior Management (Vice Presidents) to gain funding support. AutoCo now uses the harmony buck in other programs.
- **Evaluation Prototype (EP) build (4/93):** The EP brings all vehicle systems together, so that integrated testing can occur. The program team completed the first EP on April 1, 1993, making up for earlier delays and meeting the original program timing plan. Eighty-five percent of parts were available for the EP build (setting a company record; other car programs have had between forty and sixty percent of parts available at this point in their programs).
- **Change Requests (CRs) reach 500 (7/93).** Change Requests (CRs) are documents which engineers write to indicate the need for alterations in parts or technical specifications. CRs indicate that rework is needed; thus, senior management uses the count of CRs to evaluate program performance at any moment in time. Following the EP build, the Epsilon had 524 outstanding CRs, ordinarily a sign of very poor performance. (A more typical number would be 200.) Product development and manufacturing management said that they had never seen a program recover from such a high level of CRs.

⁷ "Collocation" means that people from the diverse engineering functions developing the car (i.e. chassis design, air conditioning, suspension, alternator, sound systems, dashboard subsystems, etc.) work from offices in the same location. Team members are physically placed together, during the time it takes to design the car, instead of coming together only for formally scheduled meetings.

- **Validation Prototype (VP) build (10/93):** Validation Prototype vehicles are built to test changes made after the EP build. The VP design was frozen in July of 1993 — three weeks ahead of plan. Ninety-three percent of the VP parts were on time to the material requirement date (MRD). According to manufacturing management, the quality of the VP prototypes was the best any vehicle program had ever accomplished. The subsequent engineering release (ER) was completed in August of 1993, four weeks ahead of plan. Ninety-eight percent of the ER parts were delivered one month ahead of plan, with the other two percent known and accounted for. Four VP prototype vehicles were built on the regular assembly line at the Mission Hill manufacturing plant.

The new owner vehicle assessment (NOVA) scores for VP were 96, compared to an average of 108 for other vehicle programs (lower scores mean higher quality ratings). The NOVA scores for the earlier build had been substantially *worse* than average; they were 145, compared to an AutoCo average of 105. Top AutoCo managers made what were described as uncharacteristic acknowledgments that the Epsilon program was performing well.

- **Accelerated 1PP build (6/94):** The 1PP (final) prototype build began one week early. The team had 70 percent “production status” parts (normally 50 percent). The new owner vehicle assessment (NOVA) scores for 1PP were 28, a company record. The previous best NOVA score was 35, and the average score was 55 for other vehicle programs.
- **Job One accelerated by one week (11/94):** Production builds began one week ahead of the scheduled date. Starting production early was previously unheard of, and thought not feasible given the normal chaos that surrounds a vehicle launch. The program was able to return an estimated \$65 million of the \$90 million budgeted for late changes to parts. Based on the “18 panel” reports submitted at the end of product development, the Epsilon met or exceeded all forecasted goals (quality, weight, fuel economy, performance, functional image, customer satisfaction, variable costs, investment, and vehicle profitability).
- **Final quality results:** The final Nova C quality assessment for the Epsilon was 5.8 — significantly lower (better) than the average Nova C assessment for the last six recent launches (which averaged a score of 9). Subsequent quality rating by an independent market research organizations (Competitive New Vehicle Quality) showed a 30% improvement in quality as measured by things gone wrong, rating AutoCo’s Epsilon in second place for automobiles in initial customer quality.

The noticeable results are measurable and provide firm indication of the Epsilon program’s achievements. These noticeable results were used to focus description and evaluation in interviews. What transpired so that these results were accomplished, how they were interpreted, and what they mean has been the subject of this learning history.

Appendix: Some initial questions for group discussions prior to moving forward with learning initiatives

This learning history presents a story of what happened in the Epsilon Program. It also provides a context for considering important issues which surface as business organizations undertake explicit learning efforts. Provided below are four questions applicable to what happened with Epsilon, which can also be taken as a starting point for management teams considering either leading or supporting process innovation efforts:

- How have the approaches taken by the Epsilon team added value to the traditional product development process? Can whatever value-added there was be recognized and accounted for by existing vehicle program metrics?
- Which methods and techniques used by this team can be transferred and used by other program teams? How do these tools get used to provide early improvement results (“quick hits”)?
- Which methods and techniques require longer term investments to produce improved results? How does the value-added of “quick hits” compare with those produced by longer term investments?
- What action steps and resource commitments are necessary to achieve visible improvement results on both a quick and a long-term basis in other parts of large organizations?

