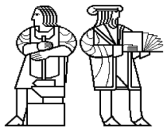

Massachusetts Institute of Technology
Engineering Systems Division

Working Paper Series

ESD-WP-2007-12

COMBATING SYSTEM-LEVEL QUALITY PROBLEMS IN COMPLEX PRODUCT DEVELOPMENT

Daniel E. Whitney

Massachusetts Institute of Technology
dwhitney@mit.edu

February 2007

COMBATTING SYSTEM-LEVEL QUALITY PROBLEMS IN COMPLEX PRODUCT DEVELOPMENT – DANIEL E WHITNEY, MIT JULY 2003 – MATERIAL SPECIFIC TO THE SUPPORTING COMPANIES HAS BEEN OMITTED

Summary

As products become more complex and their development involves more technologies, people, and companies, it is no longer sufficient to ensure that each part, component, or subsystem is designed and made correctly. Problems that involve many distinct elements can still arise, even if each is designed according to the specifications. Practitioners speak of Murphy's Law and "sneak paths," while academics refer to "emergent properties" and "undocumented interactions." The goal of this project is to look at this problem from the outside, focusing on Ford and two non-competing companies whose products are also complex: United Technologies and Boeing. Interviews were conducted with senior management and lower level supervisors, focusing on both official processes and anecdotal reports on what works and what does not. In addition, the academic literature on "system accidents" was consulted to see if there are insights that can be transferred. The focus of the study was on quality problems introduced during product development, not during manufacturing.

Interviewees at the companies agree that there is no sure fire remedy for system accidents or system level product quality problems. They also agree on some of the causes, in addition to growing product and process complexity. Among these are organizational instability, the threat to informal networks of experts posed by rapid personnel transfers, growing specialization in technical fields, the trend toward outsourcing, and a shortage of people who are sensitive to system problems. At Pratt&Whitney Aircraft Engines, more "design escapes" were caused by failure to follow existing procedures than to any other cause. At Ford, a habit of designing from scratch seems to provide opportunities for surprises, while at Toyota adherence to standardized designs or processes gradually eliminates surprises over time. At Boeing, the FAA certifies procedures and supervises adherence to them, encouraging learning over time. Nevertheless, at Boeing, Pratt, UTC Fuel Cells, and Otis Elevator, systems have so many possible internal states that some new ones are inevitably found during operation. Ford, unlike Boeing and Otis, has the opportunity to test many production-ready prototypes, but Boeing, Ford, and Otis all sell customized systems that represent new combinations of components or environments, providing new opportunities for emergent properties.

Social scientists have identified and studied system accidents for over 25 years, stressing the interactions between individuals and both their organizations and the technologies they have to manage. Events investigated include capsized offshore oil platforms and ferries, chemical plant explosions, ship collisions, and errors in hospitals. Most researchers agree that it is incorrect to blame an individual for an accident even if he or she made an obvious mistake. Many immediate and distant factors combine to cause these accidents, and most factors

are found to be deep-seated and of long duration. No deep study of product quality problems stemming from system interactions at the cultural/technological interface was found during this project, but the author suspects that system accident research can help product development avoid system quality problems.

Social scientists have developed a number of useful lenses through which to view these problems. "Creeping toward the edge" describes an organization that allows its procedures or standards to deteriorate because "we have not had any problems so far." Another lens is "problems waiting to happen." The particular problem that occurs is not in itself as important as the fact that the organization harbors a lot of them, and one was precipitated by an otherwise innocuous "trigger." Yet another lens is "a culture of diminished expectations," as in "no project in memory has made that milestone on time, so what's so bad about our project being late?"

These theorists are of two minds concerning what can be done. One school says that systems will inevitably become more complex and we have to expect problems to continue. Problems in some kinds of systems, like nuclear power plants, are potentially so bad that such systems should just be banned. Another school has identified "high reliability organizations" like air traffic control centers and nuclear powered aircraft carriers, and studied them to see how they avoid accidents.

Within the organizational and managerial community, there are also two schools of thought. One stresses process discipline and rigid adherence to standards, while the other moderates process discipline with openness and individual initiative, especially when something goes wrong. No one recommends abandoning procedures, but some stress a single approach in all circumstances while others observe how some organizations rearrange themselves depending on their situation, behaving in a standardized hierarchical way under normal conditions while improvising and substituting expertise for rank when emergencies arise.

The most sophisticated concept found during this project is "collective mindfulness." While this property has been identified in high reliability organizations, it is probably transferable to product development. It involves everyone being aware that problems are possible and being on the lookout for them, even if they occur in someone else's territory. Members of such organizations know that their lives are on the line and that no one will reject a life-saving gesture. Mutual trust and respect are basic to such behavior, as is plenty of training. Top-down organization, assigned roles, and frequent communication act to reduce the likelihood of problems, while group awareness and initiative at low levels act to suppress the ones that occur.

Project Motivation and Approach

This project came out of discussions between the author and Chris Magee, then the Executive Director of the Ford-MIT Research Alliance. The goal was to

identify work that would read directly on the mission of the Product Development Process Technology program within the Ford-MIT Research Alliance: how do we improve the quality of the product as well as the quality (i.e. efficiency and effectiveness) of the product development process? The author conducted an outsider's survey of Knowledge-Based Engineering (KBE) for Ford in 1998. This survey consisted of a literature search and interviews at Ford, Boeing, United Technologies, Kodak, and Xerox. The same approach was adopted for the current study.

Out of all the possible problems that can occur during product development, probably the most elusive is the system problem. The author's discussion with Dr Magee occurred in the months following the emergence of the Ford Explorer-Firestone Tire problem, so the issue was on many people's minds. The author knows Ford very well and has heard Ford people say things like "we are part-centric." While focusing on parts is necessary, as is assigning responsibility for parts to individual people, one cannot lose sight of interactions between parts or fail to assign people to look out for them.

The author's field of expertise is mechanical assemblies. His research is both technical and managerial. Assemblies are inherently integrative, and studying how they are designed reveals many gaps in education and practice between the lavish attention given to individual parts and the relative neglect accorded to assemblies of those parts. A manager at another car company said "The customer looks at the gap between the door and the body, but that's an empty space. We assign people to manage metal, not empty space."

In addition to current events and company habits, a motivation to look at system problems arose from the formation at MIT of the Engineering Systems Division (ESD). The ESD focuses on systems and has attracted people who are interested in them, including the author and Dr Magee. Among the complex systems that ESD has chosen to focus on is the product development process.

Just as the project was getting under way, ESD held an internal symposium in which all ESD faculty gave short talks describing their research. Among them were two that seemed especially relevant. One was by Nancy Leveson [Leveson] on how to write specifications for software (such as for air traffic control) that would prevent accidents. Leveson put forth the concept of hazard analysis, a way of determining if a system is safe that is different from failure modes and effects analysis. John Carroll [Carroll et al] described industrial accidents and embedded them in the concept of two-loop learning. Single loop learning is sufficient to change the rules, while the second loop is necessary to embed new behavior in an organization. The latter is similar in many ways to collective mindfulness. The author benefited from exposure to these new ideas and decided to pursue them.

Tracing citations in papers by Leveson and Carroll, the author was led to a large lode of papers by organizational and social scientists who seem to have targeted and chewed over system accidents for almost three decades, well out of

sight of researchers in product development. Thus the goal of the project came to be to answer the questions:

Are system-level design problems in complex products analogous to system accidents, and can the system accident literature help us understand why product system design problems occur and how to prevent or mitigate them?

What do product development professionals think? Are system-level quality problems prevalent and important? What do they think the causes and remedies are? What is their opinion of the academic theories? What methods do their companies use now to avoid system level quality problems?

What The Academic Literature Says

The author looked at two academic literatures, system accidents within the larger literature of social and organizational science, and product development, especially in the auto industry. In addition, he consulted the system engineering literature.

System Accident Research

System accident research appears to have emerged from ergonomics and individual man-machine research. Psychologists look at the individual while social scientists look at systems of individuals. Social scientists assume that industrial activity occurs within an organization, which in turn forces a number of the behaviors that occur. The technologies that people deal with are also very influential, and there is plenty of debate about whether organizations shape technologies or vice versa. [Thomas]

This domain is relevant to product development for several reasons. Systems where accidents occur are complex and are operated by many people individually or in teams. They have different technical backgrounds and work in different functional departments or companies. Product development similarly requires defining and understanding complex systems and involves many people from many different departments and companies. The system complexities in both domains make it possible for unexpected interactions to occur. Managers in both domains therefore face similar problems, as do the individual engineers.

System accident research identifies at least four ways to look at an accident: [Reason, 2000] [Rasmussen]

- The *person* theory, which says that a person can be found who caused the accident. The response is to discipline that person and make his mistake known to others, so that it will not happen again. Individual parts are made more robust to prevent a repetition. This is also known as the "fatal flaw" theory.

- The *chain of events* theory, in which event A causes event B, which causes event C, etc., until event X causes the accident. The response is to give next design more redundancy. This is sometimes called the "want of a nail" or "domino" theory.

- The *"swiss cheese"* theory, which says that a "perfect storm" of things combine in just the wrong way, causing the accident (the holes in different cheese slices line up just right). People often hope that nothing like it will ever happen again.

- *The interacting events* theory, in which the causative factors do not create a simple domino effect, but rather bounce causes and effects back and forth among each other in complex ways.

- The *accident waiting to happen* theory, in which the causative events are the result of "inherent defects"[1] in the organization; they do not need to have any relation to each other, but all of them are necessary or else the event would not have occurred.[2]

Each of these views is increasingly sophisticated. It is likely that most real accidents are mixtures of some or all of these. But each is useful in its own right because it represents a possible mental model adopted by managers and members of organizations. If you think accidents are caused by individuals who make mistakes, then you will overlook system accidents and their causes. Discussed in more detail later, this insight says that preventing different kinds of accidents requires different kinds of management intervention and organizational policies.

System accidents are rare and for good reason: a lot of effort goes into preventing accidents of all kinds, and any accident with one or two contributing causes will be easier to anticipate and prevent than one that requires six or eight. So in reasonably well-designed systems, all the easy ones have been stopped already, leaving the hard ones.

But it is a common mistake to calculate that multiple cause accidents are rare because the laws of probability make them so. If the likelihood of a contributing cause is p and it takes n contributors to make the accident happen, then one might conclude that the probability of the accident is p^n. If p is a very modest 10% and n is 6, the likelihood of the accident is one in a million, right? Wrong. This calculation assumes that each contributor is independent of the others. If the person theory is correct (equivalent to being part-centric) then the causes might be independent. But if the accident waiting to happen theory is

[1] In the medical mistake literature, these are called "resident pathogens." Like the inherent defects, they are there long before the particular accident happens. [Reason, 1990]

[2] [Chassin and Becher] describe a hospital error in which 13 doctors and nurses made 17 separate and apparently unrelated mistakes resulting in a patient receiving another patient's procedure.

correct, then the individual causes, and the accident itself, are consequences of the culture in which they are embedded. They have a common cause and thus are not independent.

For example, the likelihood that an airline pilot will make a simple procedural error (like doing the checklist from memory instead of following the book) is fairly high, but it is unlikely to endanger the flight. The likelihood that the pilot actually skips a step in the checklist is much less likely to happen but is more likely to endanger the flight. Unfortunately, pilots who make the first kind of error are 40% more likely to commit the second and even more serious kinds of errors than pilots who do not. So these apparently different causes of airline accidents are correlated, and predicting their likelihood by multiplying their individual probabilities leads to greatly underestimating the likelihood of a serious accident due to those causes acting together. [Helmreich]

The airline industry is in fact relatively safe, and several reasons are cited. One is training. Another is a non-punitive error-reporting system, such as accident investigations. A third is government supervision of aircraft design, crew training and rest, and traffic control. Within these systems lies an important habit, namely to look very hard at "near misses." (Norman Augustine said, "Cherish your anomalies."[3]) Actual accidents occur too rarely to present enough learning experiences, and no one wants accidents just so that learning can happen. Covering up is therefore very dangerous, and this is why the industry pushes the concept of the non-punitive investigation.

An important conclusion from airlines, found in other industries, is that a variety of measures is needed, combining top-down directed, standardized, enforced rule-driven behavior, plus encouragement of observation, looking in unusual places, being ready for the unexpected, being open in communication, providing training in teamwork and cooperation, and encouraging reporting of problems. These are characteristics of what are called High Reliability Organizations, discussed below.

Rasmussen stresses several paradoxes encountered on the way to a more sophisticated view of system accidents. Each accident is a sample from a kind of snake pit of potential events, all of them similar in background cause but none of them identical. If you define the cause in enough detail, it will almost certainly never happen exactly that way again. Also, if you look for an abnormal event as the cause, you will overlook the frequent situation where an apparently *normal* event acts as the trigger. This is especially true if the "culture of diminished expectations" is in effect, because the prevailing culture is by definition normal to its participants even if it is in fact defective. Thus Rasmussen says that "the causal tree found by accident analysis is only a record of one past case, not a model of the involved relational structure." Very deep analysis is required to reveal this structure.

[3] "Simple Systems and Other Myths," Brunel Lecture, MIT December 7, 2001.

According to [Weick, et al], high reliability organizations (HROs) such as air traffic control centers and nuclear powered air craft carriers share several organizational and managerial characteristics:

1. *Preoccupation with failure.* HRO members know that problems are just below the surface, are rare, and give no warning. So members are constantly on the lookout and pay attention to the smallest clues that something might be wrong. "They act as if there is no such thing as a localized failure and suspect, instead, that causal chains that produced the failure are long and wind deep inside the system." Failure includes any dysfunctional response to success, such as complacency.

2. *Reluctance to simplify interpretations.* "Organizations are defined by what they ignore." HROs socialize people to notice more. They also cultivate "requisite variety," meaning that they make sure that people with different backgrounds, assumptions, and experience are mixed together, providing adversarial reviews and checks and balances.[4]

3. *Sensitivity to operations.* People [in military situations] work hard to create a "cognitive map that allows them to integrate such diverse inputs as status, information sensors and remote observation, and real time performance of equipment." "People who are sensitive to operations see more interconnections."

4. *Commitment to resilience.* "HROs acknowledge the reality of fallible humans, murky technology, and narrow specialties." They pay attention to both error prevention and error containment. When events get outside of normal operations, knowledgeable people self-organize into ad hoc networks to provide expert problem solving.

5. *Underspecification of structures.* "Effective HROs loosen the designation of who is the important decision maker in order to allow decision making to migrate along with problems." Authority passes to those who know the most.

In general, HROs are interesting because they have developed methods for dealing with unexpected events in complex systems. They know that planning ahead will not cover everything that could happen. They try to create a structure that combines discipline with improvisation so that surprises are not as surprising, so that sneak paths may get noticed before they cause trouble, so that the best minds address the problem, and so that response is fast and unencumbered by "efficient" but rigid decision processes. In an efficient organization, processes are rigid but attitudes vary, in the sense that people constantly seek workarounds when the rigid processes do not work. In an HRO,

[4] A Ford manager told the author "Good advice comes from unexpected places."

attitudes are constant but processes vary. The constant attitude is mindfulness, while the processes are allowed to restructure in response to the situation.

HRO researchers point out that one block to HRO performance is over-stretched people caused by downsizing and broken knowledge networks caused by outsourcing. Another factor is compartmentalization and organizational loyalty, which prevent people from noticing problems that cross these boundaries. These considerations apply to product development organizations as well.

Much of the HRO culture of collective mindfulness is captured by the following quote. It is the Captain of the US nuclear powered aircraft carrier Stennis ending his morning broadcast to the crew: "Take care of yourselves, take care of your ship, take care of your shipmates."[5]

To achieve HRO performance requires facing some paradoxes in management, "providing some hierarchy but being willing to relax, combining rigidity and flexibility, confidence and wariness, anticipation and resilience." "Reliability requires diversity, duplication, overlap, and a varied response repertoire, whereas efficiency requires homogeneity, specialization, non-redundancy, and standardization."

Product Development Research

The domain of product development research is too broad to cover in this report. Here the focus will be on two areas: One is specific field studies that look deeply into how Toyota manages product development, compared to two US car makers. The other is a process mapping tool that seems well-suited to documenting system and subsystem interactions, called the Design Structure Matrix.

Field Studies Comparing Toyota to US Car Makers by Sobek and Morgan

Durward Sobek and James Morgan did PhD research on Toyota's product development methods. Toyota seems to be especially aware of system issues and takes particular steps to develop system skills in its people and organization. Sobek learned Japanese and spent 6 months in Toyota City interviewing people. He also spent considerable time at Chrysler. At that time, (1996) Chrysler was wedded to its platform team concept, while Toyota was functionally oriented and reliant on the heavyweight program manager concept.[6] [Clark and Fujimoto]

Sobek emphasizes several aspects of Toyota's product development methods that are relevant to this study. First, Toyota cultivates deep technical

[5] From a CNN program about life aboard the Stennis during the Afgan conflict of 2001.

[6] At Toyota, the term is Chief Engineer, who is responsible for the vehicle concept as well as all the system-level design.

expertise in both its managers and its engineers. They are encouraged to learn about their own area plus how it relates to others. Toyota's engineers "are technical experts with a systems-level understanding of how they and their parts fit in the bigger picture." People who are targeted as Chief Engineer material must demonstrate, over two 10-year evolutions, that they can learn the connections between things as well as in-depth understanding of two technical areas. This creates people at the top of the product development organization who have and value "connection knowledge."

Second, Toyota uses a design method that academics call Set-Based Design. This involves developing to the level of prototypes several versions of certain subsystems, rather than setting one clear set of specifications early in the process. This method permits more debate about how to balance and trade off requirements among different subsystems. "Paradoxically," this delays decisions but can make better systems and cars. [Ward, et al] This practice acknowledges that one cannot write a complete specification and flow it down to subsystems, but instead must have some hardware or other essentially bottom-up way of seeing if misfits occur before a final decision is made.

Third, Toyota permits each design team some latitude in organizing itself and requires individual engineers to find the information they need (effectively a pull system for information). "The pull strategy is aided by maximizing written communication and by minimizing meetings."

Fourth, emphasized more by Morgan, Toyota standardizes parts and processes. This "maximizes learning and continuous improvement, speeds up the design process, and increases the reliability of designs." In fact, a lot of concurrent engineering is unnecessary because engineers and process designers already know what to expect.

Morgan completed his PhD thesis at the University of Michigan, in July, 2002. It extends the thesis by Sobek and shares the same faculty supervisors. Morgan compares body engineering from clay modeling to completion of stamping dies at Toyota and a disguised US firm called North American Competitor (NAC). Morgan pursues two main themes. One of these is a three-part framework consisting of "process," "people," and "tools." The other is "lean." Lean emerges in his emphasis on value stream mapping (VSM), derived from recent descriptive books by Smith and Reinertsen (1997) and Shook and Rother (1998). [Womack and Jones] also recommend VSM.

Morgan identifies the complexities of product development as

- Parallel activities and work streams
- Information interchange between the parallel streams (handoffs allow errors to enter the communication)
- Diversity of functions, language, and incentives across the participating people, organizations, and companies

- Long durations of activities and the process as a whole
- Queues, congestion, delays, errors, feedback, and rework

He says that these factors distinguish product development from manufacturing. In spite of this, the main theme of the thesis is that methods found successful in manufacturing can be successfully applied to product development, and that Toyota has done so. Some of the techniques Toyota uses in product development seem to transfer directly, such as JIT applied to both information and people, identification and elimination of excess inventory (queues of work waiting to be done), and standardization of processes. Not transferred is the idea of cross-trained people. In fact, specialization and deep technical knowledge of a limited range of tasks is emphasized in product development. People spend years doing the same job, and their deep technical knowledge, plus standardization, holds the system together.

A summary of Toyota's principles is as follows [Morgan, Chapter 1]

1. A holistic systems approach to product development, integrating and aligning people, processes, and tools. People are highly skilled and are rewarded for following the standard processes. Processes are aligned to minimize waste. Technology is "right-sized."[7]

2. Embedded customer-first attitude. This begins when new hires do a turn at a dealership. It supposedly helps align goals and incentives and resolves tradeoffs.

3. A front loaded process. In body engineering, this involves long discussions between teams of stylists, body engineers, and body manufacturing people to anticipate and eliminate manufacturing problems during clay model development.

4. Built-in learning and continuous improvement. This sounds similar to what is done in manufacturing but the methods are different. They include in-process and post mortem "reflection" events where problems are discussed by the product development team, and counter-measures are defined for next time.

5. Synchronization of parallel processes for simultaneous execution. This is where VSM comes in. It allows processes to be mapped, including information exchanges between parallel streams. Process standardization helps here too because some tasks can begin with partial information since the receivers know pretty much what the suppliers of info will do next.

[7] In the PI's own visits to Toyota in 1991 he noted small machine tools and simple computer analysis methods. At NAC both trends are reversed. Taka Fujimoto once told the author that Ford has more computer power than the entire Japanese car industry.

6. Rigorous standardization creates strategic flexibility. Standard processes for design and manufacture of fairly standardized sheet metal parts removes most of the surprises from their design and analysis, die design, and die tryout. When extra people are needed to handle surges in work, people from other body engineering groups or even affiliated companies can step in because the tasks are well known and practiced. This is the flexibility referred to above. Standardization also includes reuse, common architecture, standard die design methods, and so on.

7. "Go to the source" engineering. This means that engineers and others get information first hand and do not depend on information handoffs. They visit the dealerships and the plant, witness tests, and talk directly to other involved people.

One particularly interesting Toyota process practice is the designation of a "Simultaneous Engineer" (SE) for each sheet metal part and subassembly. This person is from manufacturing and is the mother hen of the part from clay modeling to production. The SE is involved with every process step that impinges on the part. He follows it from step to step and department to department. In some sense he is the system engineer or system integrator, analogous to the medical internist who musters the necessary specialists for the patient. It is significant that the SE is from the group that receives parts rather than from a group that generates part designs.

From other sources the author has learned that Boeing has assigned engineers end-to-end responsibility for parts on a recent 777 aircraft program [Cutcher-Gershenfield et al] and that GM has defined a role called integration engineer. When the author visited Toyota in 1991 they emphasized to him "continuing responsibility" on the part of engineers for their components later in the development and launch process.

In summary, it appears that Toyota is rich in procedures and rewards people for following them. We must be careful not to ignore the other aspects of Toyota that encourage connection thinking and cross-discipline or cross-time connections that link processes and organizations together.

System Engineering Theory

System Engineering Theory teaches designers to create a hierarchy of functions and physical objects. Boeing calls the highest level requirements "Key Characteristics." Ford calls them "attributes." In most cases, these are system behaviors that are visible to the customer. The requirements of upper levels in the hierarchy are decomposed and flowed down to the lower levels. This is intended to create separate manageable pieces that can be worked on independently. Carried to its extreme, this is called "reductionism." Major challenges include remembering all the requirements, keeping them consistent, and understanding the many interactions between branches of the hierarchy.

These interactions cause problems during integration at the end of product development and challenge the basic assumptions underlying reductionism.

System engineering theory works most smoothly when the product can be broken into modules that are relatively independent. Such products or systems are called modular. When products cannot be decomposed simply, or when their behaviors interact, they are called integral. [Ulrich and Eppinger] As shown in Figure 1, some kinds of products are easier to modularize than others. More integral products need to be designed at the highest level of the hierarchy, or their design requires a lot of coordination of the "modules" at lower levels.

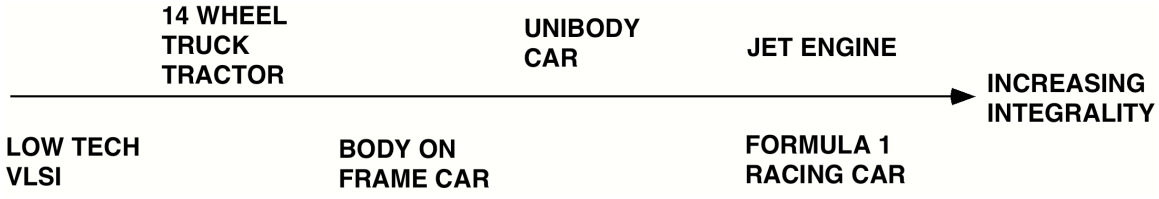

Figure 1. Examples of Products with Different Degrees of Integrality. As integrality increases, it is harder to divide the product into independent modules. (Illustration provided by Prof Jasper Steyn, University of Pretoria.)

Interviews or Literature About System Engineering

The top-down approach recommended by system engineering theory is a good guide to sound engineering practices. However, it must be applied with caution, as the following interviews indicate.

According to the theory of Axiomatic Design [Suh], the best design is one where each function is implemented in a way that is independent of implementation of any other function. This permits the maximum in independence and simplicity. It is impossible to define all of these relationships at once. Instead, one has to start with the top-level functions and define some top-level technological choices or implementations. These give rise to a second layer of functional requirements, which in turn are linked to more detailed or subordinate technical implementations. This process continues in a zig-zag fashion to the lowest level in the hierarchy. While there is general agreement that simpler designs with independent functions are desirable, there is less agreement about whether this is possible in complex systems.

An SDM[8] graduate working for high performance workstation manufacturer told the author the following story about outsourcing, often cited as a cause of system-level problems: To combat falling margins, his company has outsourced more and more. Subject-matter experts were laid off on the assumption that their detailed knowledge would be filled by the suppliers. But customers want PC prices and high performance reliability, while the suppliers came from the PC culture and were unprepared for the company's requirements.

[8] SDM means System Design and Management, a graduate professional program at MIT that trains students to understand systems.

Things that look modular and low-tech, like cabinet doors, become integral when attached to a computer that stresses hardware the way his company's operating system does. Processors run very hot and lots of electro-magnetic interference can be expected. This affects cabinet design in many complex ways that the company understands (or understood) but suppliers do not.

A U S Air Force Systems Command SDM graduate told the author that requirements seem to be emergent. As the system is being designed, users begin to understand what its potential is and start to add requirements. Also, constraints emerge as implementations are explored, and these add new requirements. This is similar to Suh's zig-zag model of engineering design.

Carliss Baldwin and Kim Clark have studied modularity and conclude that there are three kinds: [Baldwin and Clark]

- modularity in design – each function is designed separately and placed in one physical object, or several functions and their objects are combined and designed together
- modularity in production – a group of functions or physical objects is built or bought as a package
- modularity in use – the customer can combine several functions or physical objects and use them, choose them when buying, or upgrade them together

The important thing to understand is that the three different kinds of modules may be different. As illustrated in Figure 2, a module that exists in production may include parts of many modules that exist in design, causing confusion, cutting systems in two, and creating disconnected pieces of requirements that must be satisfied by different suppliers who may not have the skills. The example in the figure was provided by Francois Fourcade, who was product line manager for a front-end module for a French car supplier. Its customer cancelled the program and the supplier, seeing no way to make a profit in view of the technical and business complexities, has not bid on this kind of item since. [Fourcade]

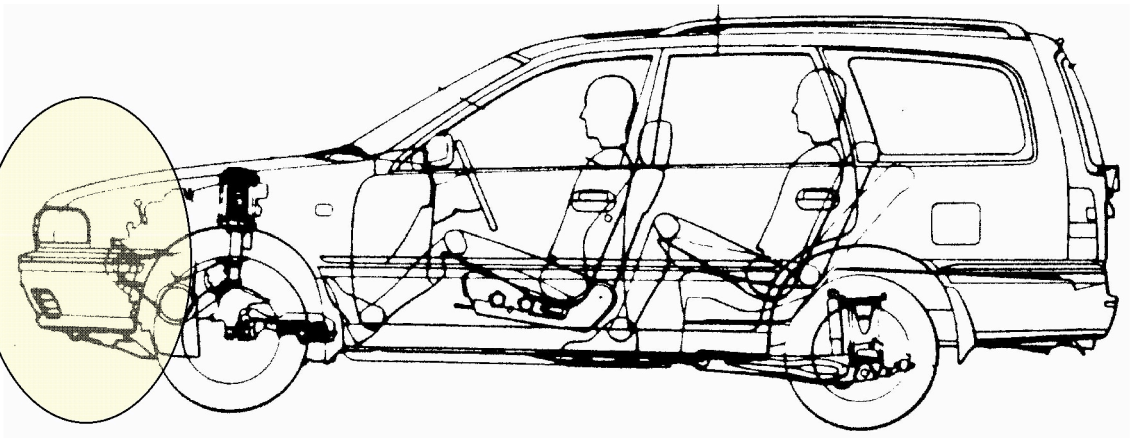

Figure 2. A "Module in Production" Consisting of Front Bumper, Bolster, Grill, and Lights Contains Portions of Several Systems but Not All of Any One. It is Easy to Install on the Car but Hard to Design and Test.

Norman Augustine, past CEO of Lockheed-Martin says "No change is a small change." Sneak paths are always possible, and a change can include or create a new sneak path. Constant awareness is the only answer. Question everything: make sure at least one non-yes man is on the team. Attention to detail increases your luck. Treasure your anomalies. Examine claims that the system is redundant: it may look that way on the print but not be in fact.

According to a Pratt&Whitney Aircraft Engines SDM graduate, component and system simulations can predict a jet engine's fuel consumption within about 2% but that is not good enough to know for sure that contract performance will be reached. Even separate tests on real components are not good enough due to strong coupling between them. Only a complete engine test will reveal actual performance. Engine manufacturers use results of engine tests to improve the accuracy of simulations for use on the next engine program. The best way to improve an engine after this test is to coordinate design activities on several components rather than trying to further improve individual components.

Design Structure Matrix Methods

The Design Structure Matrix (DSM) is a method for documenting interactions. The DSM can capture the sequence and relationships of design tasks or decisions, or the relationships between parts in a product, or relations between design parameters like diameter of a brake disk and pedal pressure. The DSM has been used successfully at Ford to streamline complex engineering processes as well as to align timing for exchange of information between many engineering teams.

At Pratt&Whitney Aircraft Engines, two SDM students made a design parameter DSM of an engine to illustrate how parameter decisions propagate through the design. The goal was to help predict the impact of strategic design decisions, such as how many rows of blades should be in the compressor. Too few rows could cause compressor stall and backfires, while too many rows lengthen the engine. Often a chain of 6 to 8 parameters intervenes between the first decision and the ultimate problem. Only a few P&W employees have enough experience to be aware of the entire chain, but several of them can be found by inspecting this DSM. Thus the DSM can be thought of as a place to store system-level information about how things interact.

DSMs generally indicate that things are coupled. The best that one can do is use the DSM to identify clusters of things that must be considered together, say by assigning a team to them or focusing a meeting on them. It is unusual to find a design decoupled the way Prof Suh recommends.

Summary

The common theme running through both technically-oriented and non-technical research on product development, high reliability organizations, and system engineering is that unexpected or emergent behaviors are inevitable in complex systems. Among the causes are

- Action at a distance, or invisible actions
- Technological immaturity
- Domain crossing or linking (mechanical and electronics, for example)
- Unsophisticated view of interfaces
- Lack of oversight of system interactions
- Time delay, or gradual growth of interactions
- Human cognitive limits
- Human agency, ingenuity, and gaming
- Excessive reductionism, ignoring or trying to suppress interactions, or decomposing things that are integral or highly coupled

Reading down this list takes us from fairly straight-forward technical complexities to very sophisticated problems associated with quite abstract forces. A common issue is complex interactions, equivalently the refusal of separate elements of a system to behave separately. Design methods and design cultures must be attuned to these interactions and be prepared to counter them. Both social science and product development research indicate that, while procedures and process discipline are necessary, they are insufficient. Since systems do unscripted things, product developers and system operators must have some unscripted procedures available to them to help them respond.

Interviews[9]

MIT

The author spoke personally to several MIT faculty members who offered the following comments and advice:

Prof Nancy Leveson, Software Safety Expert: In software systems, most of the failures occur in spite of the fact that the software functioned as it was intended to. The cause of the failure is inadequate specifications. She advocates an approach called "hazard analysis."[10] This method differs from FMEA, which requires identifying ways things could fail, which she calls "scenario analysis."

[9] In general, interviewees are not identified in the interests of making the narrative flow. In some cases, the names are given in order to provide context for their comments or to give credit for their intellectual contribution. Some interviews were obtained in the course of general discussions about other topics and not specifically in connection with this project. Only the interviews listed under MIT or company names in this report were part of this project, and all interviewees understood that.

[10] This is also called "outcomes analysis."

In complex systems there are too many scenarios and you will never think of all of them. Instead it is more efficient and effective to identify the hazards and make sure that the system prevents them. In an air traffic control system, "two planes on a collision course" is a hazard. The software assigning paths to aircraft must check that no new path creates a collision course. This is different from trying to find all the ways that a collision course could come into being and then seeking to prevent each of these causes. She also believes that the person/part theory and the chain of events theory are believed by most managers. The response to the person theory is to fix the blame and try to design more robust parts. The response to the chain of events theory is to add redundancy in the hope of breaking the chain. But this can add complexity. When the cause is due to a network of interactions or due to culture and organization, managers who believe in the simpler explanations are baffled to find that the components worked as they were supposed to.

Prof John Carroll, Sloan School of Management, expert in organizational learning: Managers who believe the person/part theory respond to an accident by clamping down on the guilty individual and enforcing the rules more stringently.

Prof Paul Lagace, former co-Director of the SDM program and Prof of Aeronautics and Astronautics: He is an expert in composite materials and a consultant to Boeing. He says that Boeing Commercial Aircraft over-designs its structures more than Airbus does, but both are carefully supervised by the FAA and the European regulators. Aircraft design is very conservative and verified by full size prototypes tested to destruction. Design methods are standardized to "the Boeing way" and MIT graduates who work there sometimes get frustrated. Thanks in part to the influence of the LFM[11] program and Boeing's large cadre of LFM graduates, the company is strengthening its design-build team concept. Yet aircraft are extremely complex and there is no way to predict everything that will happen. His hope is that every LFM and SDM partner company will have a systems engineering department with real influence over the product development process.

[Company-specific sections omitted.]

Discussion and Conclusions

Complex products and systems are hard to design because a) they contain complex components, and b) because those components interact in complex and sometimes unpredictable ways. There is a limit to the ability of design and management processes to anticipate all of these emergent system behaviors. The ability of people to notice and discover these behaviors needs to be included in

[11] LFM means Leaders for Manufacturing, a professional program offered by MIT that leads to master's degrees in engineering and management.

the way product development is managed.[12] New attitudes and expectations are required, but new organizational structures may not be. (In fact, none of the known structures (functional, program, matrix, etc.) have proven totally satisfactory or durable.) Instead, the existing organizations need additional flexibility to accommodate some unscripted boundary crossing by people who know what to look for and are encouraged to do so. Such unscripted activities mirror the unanticipated emergent system behaviors and are so far the only proven response to them. This is the lesson from the High Reliability Organization research cited above. "Unvarying procedures can't handle what they didn't anticipate." [Weick, et al] Yet managers of complex product development activities seem to have an irresistible urge to add more checklists, health charts, and procedures in an attempt to rein in unpredictable problems.

The contrasting approaches described above could be called "top-down" and "bottom-up." In various academic literatures, top-down is also called mechanistic, reductionist, and having the process perspective. Correspondingly, the bottom-up approach is also called organic, wholistic, and taking the practice perspective. The message from this study is that neither of these approaches can be counted on to suffice alone, but that top-down has been given more play and confidence, while bottom-up has not been given enough, except in special circumstances.

It takes a long time for someone to learn all the interactions that are known, much less learn how to find the hidden ones. Data from several lines of research indicate that individual attributes are supported by 6 to 12 underlying and interacting items, and sometimes more. The interactions comprise long chains that can snake unrecognized through assemblies, tooling, internal organizations, and supply chains. Only the most senior employees, given the chance, are likely to understand the whole chain in any given situation. Management incentives and career path planning are needed to ensure that a critical mass, perhaps 10% of all technical employees, have a chance to develop this kind of knowledge and put it into use.

In some cases, the required knowledge is so broad that one person will never have it all. Instead it will be found shared among several people who have been allowed to work together for many years. Often these people rotate the responsibilities among themselves so that each has some taste of every aspect of the system. [Whitney] This is in sharp contrast to the notion that deep knowledge is always associated with individuals with experience in one domain.

The challenge for traditional management, especially people brought up on success formulas like Lean or Re-engineering, is to trust their staff with more opportunities to find out for themselves what is going on. Process discipline

[12] The author had a friend at IBM, now deceased, who said his job was "corporate gadfly." He had no direct reports but instead reported directly to the CEO for many years. He traveled around this large corporation, noticing things and making connections between people who did not know each other but who could benefit by making contact.

promotes standardization, repetition, and doing things the same way every time. At low levels in the product system, or on a production line, this is essential. But emergent behavior cannot be commanded to appear. It must be discovered by continuous effort of everyone. This effort should be considered successful even if some discoveries happen by accident. Such lucky accidents have to be encouraged to appear.

References

[Baldwin and Clark] Baldwin, C., and Clark, K. B., *Design Rules*, Boston: Harvard Business School Press, 2001.

[Carroll, et al] Carroll, J. S., Rudolph, J. W., and Hatakenaka, S., "Learning from Organizational Experience," available at http://esd.mit.edu/wps/wps.html as ESD-WP-2003-01.11

[Chassin and Becher] Chassin, M. R., and Becher, E. C., "The Wrong Patient," Annals of Internal Medicine, vol 136, no 11, 4 June, 2002, pp 826-833.

[Clark and Fujimoto] Clark, K. B. and Fujimoto. T., *Product Development Performance: Strategy, Organization, and Management in the World Auto Industry*, Boston: Harvard Business School Press, 1991

[Fourcade] Fourcade, F., "Vehicle Modularization: Challenges Facing Level 1 Suppliers amd Prerequisites for its Implementation," draft paper for GERPISA, Feb 7, 2003.

[Helmreich] Helmreich, R. L., "On error Management: Lessons from Aviation," British Medical Journal, vol 320, 18 March 2000, pp 781-785

[Leveson] Leveson, N., "A New Accident Model for Engineering Safer Systems," MIT Engineering Systems Div Internal Symposium, May 29-30, 2002. Available at http://esd.mit.edu/wps/wps.html as ESD-WP-2003-01.19

[Morgan] Morgan, J. M., "High Performance Product Development: A Systems Approach To A Lean Product Development Process," PhD Thesis, University of Michigan, 2002

[Perrow] Perrow, Charles, *Normal Accidents*, (revised) Princeton University Press, 1999.

[Rasmussen] Rasmussen, J., "Human Error and the Problem of Causality in Analysis of Accidents," Phil Trans R Soc of London B Biol Sci 1990; 327: 449-60

[Reason] Reason, J., "Human Error: Models and Management," British Medical Journal, vol 320, 18 March, 2000, pp 768-771

[Sobek] Sobek, Durward K, II, "Principles that Shape Product Development Systems: A Toyota-Chrysler Comparison," PhD Thesis, University of Michigan, 1997

[Suh] Suh, N. P. *Principles of Design*, Oxford University Press, 2000

[Thomas] Thomas, R., *What Machines Can't Do*, Berkeley: University of California Press, 1994.

[Ulrich and Eppinger] Ulrich, K., and Eppinger, S. D., *Product Design and Development*, New York: McGraw-Hill-Irwin, 2nd edition, 2002.

[Ward, et al] Ward, A, Liker, J. K., Cristiano, J. J., and Sobek, D. K., II, "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster," Sloan Management Review, Spring 1995, pp 43-61.

[Weick, et al] Weick, Weick, K. E., Sutcliffe, K. M., and Obstfeld, D., "Organizing for High Reliability: Processes of Collective Mindfulness," Res Org Beh, v 21, pp 81-123, 1999

[Whitney] Whitney, D. E., "Visit to Nikon Nishi-Ohi Works," downloadable at http://esd.mit.edu/esd_books/whitney/whitney_1991.html

[Womack and Jones] Womack, J., and Jones, D., *Lean Thinking*, New York: Simon & Schuster, 1996