A Different Perspective

Why Flight Test Is Distinctively Complex

By
Robert W Barham
Lockheed Martin Fellow - Retired

Starr J Hughes
Flight Test Operations & Planning

Table of Contents

List of Figures

List of Tables

Management and Training of Flight Testers – A Different Perspective

Starr Hughes
Bob Barham

Background

This article originated as a briefing to executives of a large aerospace contractor. These executives were visiting the company's flight test programs at Edwards Air Force Base but most had little or no firsthand knowledge of the flight test disciplines or its protocols. However, this visit was not the standard dog and pony show. Some of these visitors were responsible for engineering workforce recruiting, training, career management and program staffing that included both design engineering and flight test engineering. A briefing was constructed to explain what flight test does, how it fits into the larger program cycle, and the roles and responsibilities of the flight test people that the delegation would be meeting. The briefing also was intended to lay a frame of reference for places and operations the delegation would see during the day, to include an in progress flight test mission. Characteristics that differentiate flight test from the design engineering disciplines was also a topic of discussion. Finally the challenges facing the flight test discipline going forward were surfaced.

The intent of this article is to present a hypothesis, for peer consideration, that asserts that the traditional approach to training and management used with most design engineering disciplines and other program organizations is ineffective and perhaps dangerous when applied to the flight test discipline.

The Hypothesis

The hypothesis of this paper is that flight test presents an environment in which one operates best using patterns and principles vice analysis and rules. Effective training, management and decision making in a flight test organizational structure is fundamentally different than design engineering

Our observational universe is from flight test operations at a large aerospace contractor focused on military, fighter/attack aircraft. In addition to experience as flight test engineers, we both have held leadership positions in flight test organizations spanning from the 1990's until 2016 with Starr having the most recent experience. Going back a bit further, Bob Barham was an officer with the U. S. Air Force in the 70's and 80's involved with developmental and operational testing and tactics development before joining the contractor world.

What we do not know, and hope to gain insight into from the readers, is whether our observations and logic are valid for other flight test environments: bomber/tanker/cargo military programs, commercial transports or FAA governed general aviation programs for instance. Do flight testers in other venues have a different set of observations?

Introduction

In 1979, a flight of two F-4E Phantoms takeoff and climb away from a hot desert landscape. Departing the area around the air base, they light afterburners and climb to an initial working altitude of 22,000 feet. As the flight levels off, the flight lead calls for the wingman to move from the right wing to the left.

Taking a position off leads left wing, the wingman notices a small piece of sheet metal bent back into the airstream. It just a small covering at the base of the vertical tail where the horizontal stabilizer pivot shaft runs into the fuselage. Hardly even noticeable. The lead aircraft is flying just fine. The aircraft is experiencing normal flight control response and there are no caution lights or other cockpit indications of anything other than a perfectly healthy, normal flying Phantom. So what do you think the flight lead chose to do? If you've been around aviation for a while, the choice is obvious, is it not?

After a controllability check, a precautionary landing was made back at home base. It was a bit of an abnormal landing when the drag chute failed to deploy after landing (yes, drag chutes were pretty common for airplanes in those days) but we still were able to make the center taxiway turnout. Immediately upon turning onto the taxi way, the entire aft fuselage was engulfed in flames. There was no drag chute because it had burned to ashes. Later investigation showed that the bent piece of metal noticed by the wingman was cause by a fuel fed explosion and fire due to a cracked fuel vent line in the aft section of the fuselage where there were no fire detection loops. The investigators estimated that had we flown another 1 to 2 minutes, the stabilator actuator lines would have burned through. Loss of control would have been immediate and catastrophic. Better to be lucky than good, huh?

In considering this scenario, what process or thinking went into the decision to return to base? There was no procedure written to cover this scenario. There wasn't a flight rule that demanded a certain crew response. There was no "analysis" of data, at least not in the traditional sense of the word, that lead to selection of one course of action over another. Why then, were the decisions that were made, made?

Our goal is to try to explain, what drove the crew decision making in a particular direction. What governed the "thinking process?" Indeed, we want to show that this "thinking process" is something you do all the time, unawares, but is far different than the "thinking process" normally associated with program organizations, business school management practices and most design engineering workspaces. We will use a tool called the Cynefin Framework to help establish a perspective on problem solving that we believe you will find insightful and useful in the future as you contend with problems and deal with a management structure or organizational culture that doesn't speak the same "language" as the flight tester does.

The Cynefin Framework

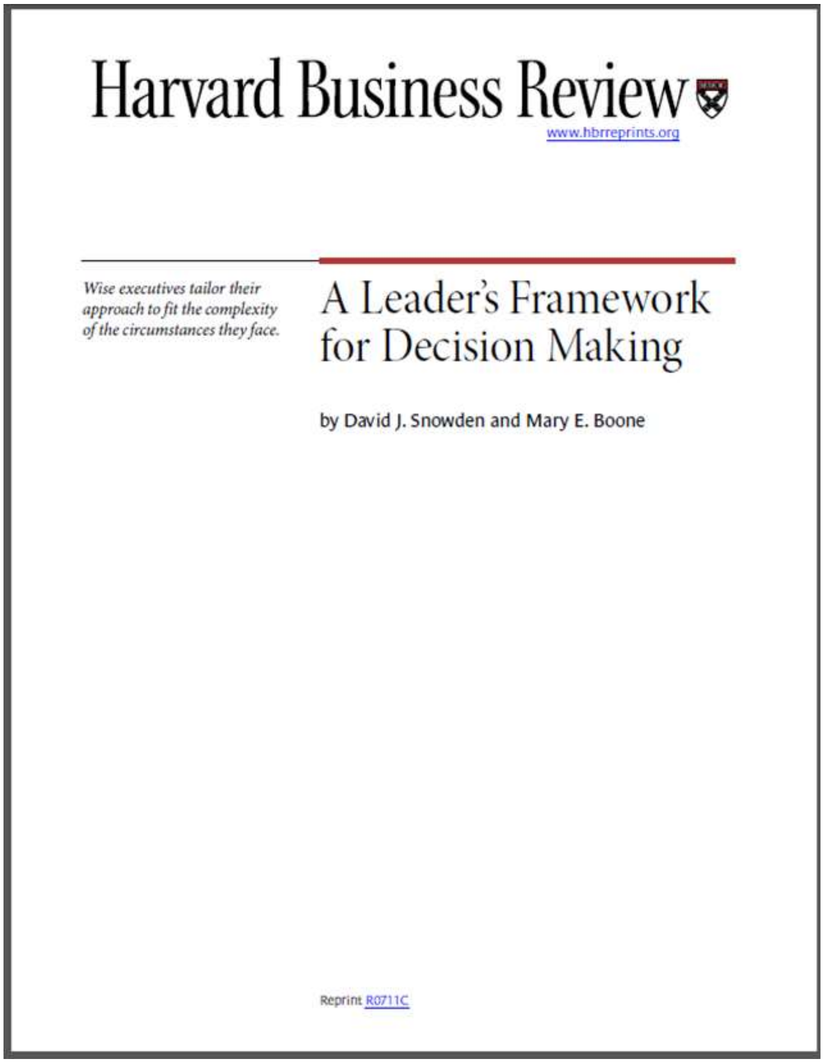
The Cynefin Framework was developed by David J. Snowden with collaboration from Cynthia Kurtz while at IBM, publishing a paper, "*The new dynamics of strategy: Sense-making in a complex and complicated world."* in the IBM Systems Journal[1]. After leaving IBM, a company called Cognitive Edge was formed where Mr. Snowden is Chief Scientific Officer.

The Cynefin Framework is an approach that helps one view problems from different viewpoints, assimilate problem scenario complexity, whether real or imagined, and choose an appropriate approach to resolution. Cynefin provides a framework for making sense of problem scenarios and optimizing solutions based on the problem context.

In 2007 "*A Leader's Framework for Decision Making*" by Mr. Snowden and Mary E. Boone was published in the November 2007 issue of Harvard Business Review[2]. In this paper, the Cynefin Framework is unveiled and explained in the context of decision making by organizational leaders. Mr. Snowden also explains the basic concepts of the framework in a You Tube video at this hyperlink **http://www.youtube.com/watch?v=N7oz366X0-8&feature=player_detailpage**.

We highly recommend the Harvard Business Review paper as well as the You Tube video.

Cognitive Edge also has a website where more information can be found. The IBM journal article (ref 1.) is a more detailed explanation which we also recommend for the interested reader.


Harvard Business Review

www.hbrreprints.org

Wise executives tailor their approach to fit the complexity of the circumstances they face.

A Leader's Framework for Decision Making

by David J. Snowden and Mary E. Boone

Reprint R0711C

Cynefin is a Welsh word, pronounced ku-nev-in, that signifies the numerous factors within our environment and realm of personal experience that influences us in ways that we can't understand or are not aware of. Take, for instance, the family vacation. Everyone takes the same trip, in the same car and sees the same sights. For all intents, everyone in the family has the same data inputs. However, after the vacation, each person will recount a different experience. Why is that? We all have sensory "filters", some that we are aware of and many that we aren't conscious of. We process sensory data through these "filters"; things like prior experiences, our culture, our philosophical presuppositions and those intangible personality traits, for instance. We are not consciously aware of most of the things we interact with. But as we go through life, the way we interact and the method/process by which those interactions take place is largely based on our memory of previous interactions. The Cynefin Framework provides assistance in helping us better identify the best way to interact and the best way to process interactions for a favorable result. It helps us to better understand and deal with complexity. I'm sure this doesn't sound like a flight test "thing" but hang in. We'll get there!

Now this treatise in not about Cynefin per se but it is important to have an understanding of the Cynefin context domains because they all play into how we can effectively train and better manage flight test people and organizations. We'll briefly describe each of the Cynefin context domains, then give some examples of each and that will begin the tie-into our flight test focused discussion.

Figure 1 depicts the Context Domains of the Cynefin Framework. There are five context domains; Simple, Complicated, Complex, Chaos and Disorder.

The Simple Context Domain

In the simple domain, problem/interactions have a clear, one-to-one if you will, cause-effect relationship. All the information is known or readily available. The process/interactions are clear and stable. In other words, it's a domain of "known-knowns." The domain is one of <u>best practice</u> where there is a <u>single</u> best <u>solution</u>. Problems/interactions are resolved by **sensing**, **categorizing** and **responding**.

As an example, from an organizational perspective, bureaucratic organizations work in the simple context domain. They are marked by stability in interactions and stability in processing of those interactions. Bureaucracies

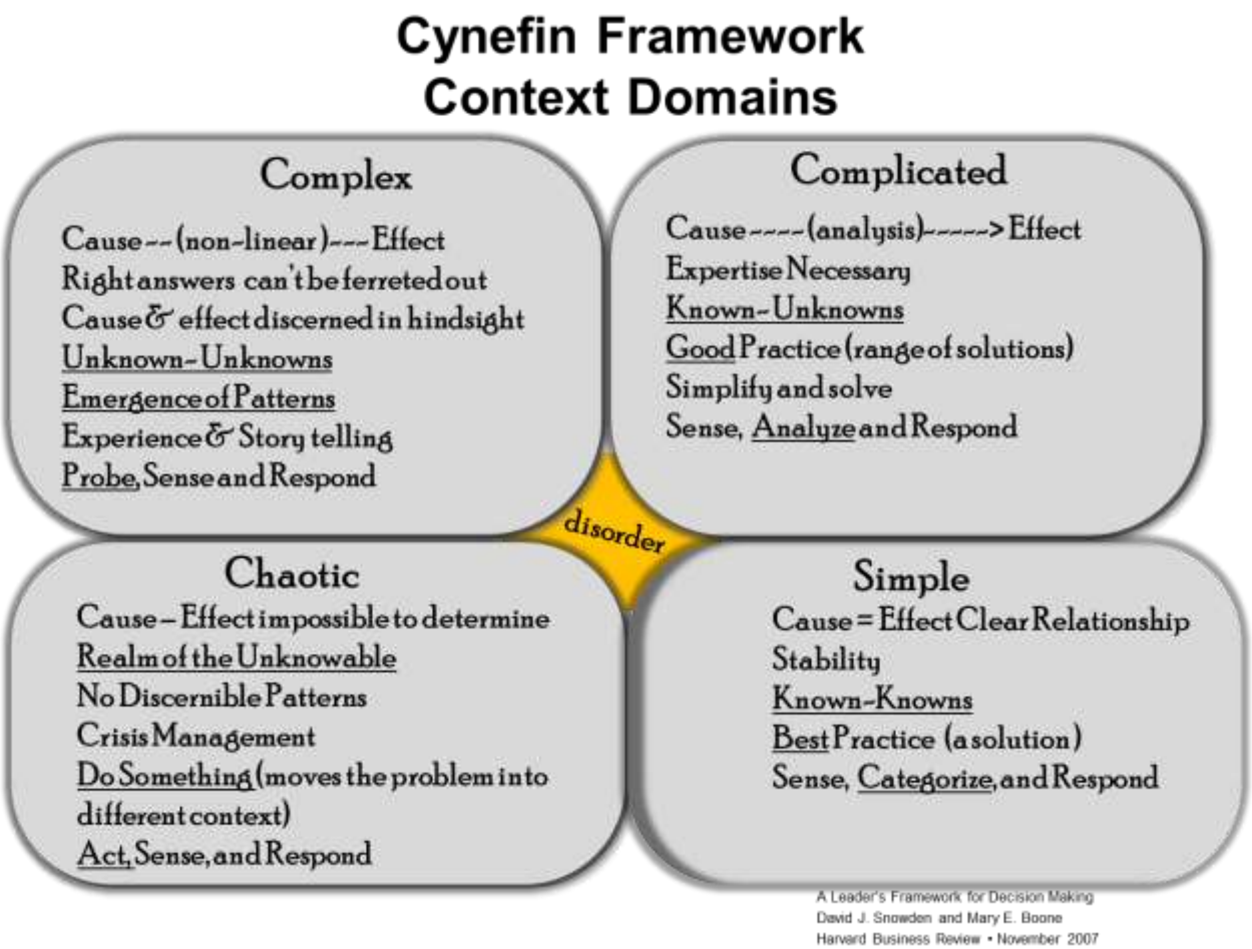


Figure 1 Cynefin Framework Context Domains

operate based on rules that govern the interactions. Think of a bank loan for instance.

Same paperwork is filled out by all loan applicants. All loan applications are processed by the same rule set. The information in the loan package is categorized and the loan approval decision and rate of interest to be paid are determined by established (best) practices that are codified in the criteria set up in the rules. Deviations from the normal interactions are usually minor, easily identifiable and a response to restore "normality" is readily available. Failures occur when the interactions are incorrectly categorized or there are errors in the rules. In the simple domain, problems that are not satisfactorily resolved are generally view as <u>process failures</u>, and a correction or modification of the process is the means of resolving the problem <u>in the future</u>.

The Complicated Context Domain

The complicated context domain has a discernable cause-effect relationship but expertise must be applied to determine that relationship. Analysis is a necessary element. One knows what he doesn't know, i.e. "known-unknowns". Good practices are successfully employed, not necessarily the "best" practice. This means that there are a <u>range of solutions</u>, not one unique solution. Problems/interactions can be broken down into simpler components and solved. Problems/interactions are resolved by **sensing, analyzing** and **responding**.

In this case, let's use the example of an aircraft structural design engineer. There is a direct relationship between the stress in a bulkhead and yielding or failure of that bulkhead. A given load applied at a given point on the structure will always result in the same stresses at a given point in the structure. However, specialized analysis is required to arrive at that stress number and expertise is required of the designer to know what stress levels at a given point are acceptable for a satisfactory design. This expertise is rooted in the application of mathematical rules and material properties that have proven to work <u>in the past</u>. Finite element models break the problem down into simpler forms, the simpler forms are solved and "added" together to arrive at the total design solution. Unlike the bureaucracy where there is one right solution, there are a range of acceptable solutions for design of the bulkhead that will be satisfactory.

There are at least two sources of "failure" for people working in the complicated context domain. One source is the tendency, over time, to become rooted or vested in a particular way of interacting/thinking. Past successes are an incentive to continue in the same vein in the future and there is a tendency to shun innovation, especially those that do not have a track record established. Change becomes hard. Perhaps a dated example is the resistance to introduction of composites, black aluminum, in aircraft structure, particularly in civil applications. An overgeneralization to be sure, but you get the idea.

Another source of breakdown can be "analysis paralysis". Generally this occurs when experts cannot agree on a path and more and more "analysis" is demanded. While that is a significant breakdown, to be sure, we believe the consequences are even more severe when it happens at the leadership level. In both cases, more and more analysis is demanded and in essence, sets up a scenario (or hope) where those demanding the analysis will continue to do so until the enough analytical evidence is forthcoming to make the conclusion or decision indisputable.

The Complex Context Domain

In the complex context domain, the cause-effect relationship is non-linear. Causes can have inordinately large or small effects. Right "answers" can't be ferreted out. Cause-effects are only discernable in hindsight. In this domain, one does not know what he doesn't know. "Unknown-unknowns." Patterns emerge and are discernable. <u>Experience</u> (i.e. pattern recognition) and <u>storytelling</u> are key elements in successful resolution. Problems/interactions are resolved by **probing, sensing** and **responding**.

That is to say that an interaction may or may not produce emergence of a favorable pattern. Interactions that produce favorable patterns are accentuated while interactions that produce unfavorable patterns are attenuated.

As an illustration, Reference 1 cites a case where West Point graduates were asked to manage the playtime of a group of kindergarten kids. The big "mistake?" They were given time to prepare and they, predictably we suppose, planned it like a military operation. Objectives, action plans and backup action plans were formed, all based on rational thinking. Now you parents (experienced) with kindergarten aged kids can predict now what happened. Of course, the kids immediately turned the whole, carefully planned affair into chaos; the plans were meaningless and objectives thrown out. It was a disaster. But experienced kindergarten teachers "manage" the same task quite differently. They allow some degree of freedom for the kid's play activities in the beginning and watch to see what interactions and patterns emerge. They act in ways, <u>based on their experience</u>, that stabilize and promote the favorable, desirable patterns of play and intervene to destabilize and discourage the unfavorable patterns.

We unconsciously use pattern recognition and stabilization/destabilization of favorable/unfavorable patterns all the time. You don't analyze your commute to work every day. Instead, it is largely a matter of recognizing the good patterns and acting to stabilize and reinforce them as well as recognizing the emerging bad patterns and taking action or mitigate or avoid those.

We contend that experienced flight testers and flight safety people, particularly accident investigators, are good at working in this complex, pattern based environment.

The Chaos Context Domain

A key characteristics of the chaos domain is that the cause-effect relationship is impossible to determine. This domain is the realm of the unknowable. There are no discernable patterns. In this domain, the name of the game is crisis management. Do something . . . anything to attempt to move the problem into a different context domain. Problems/interactions are focused on **acting, sensing** and **responding**.

Here, implementing certain interactions to produce a "right" answer is futile. The environment is confused and unstable. The early hours of the World War II Battle of the Bulge can be used as an illustration. Allied units were caught off guard and unprepared for the lighting thrust of the German armor. They were thrown into an utter state of mayhem and confusion. German units were not only to the fore, but were on the flanks and behind the allied units. Communications were cut off, individuals and companies and battalions were isolated and under attack from all sides. There no discernable patterns in the chaos at the unit level up to army corps levels. In the immediate aftermath of the surprise offensive, Allied commanders from platoon leaders to generals were simply trying to "stop the bleeding" as it were. There were <u>no rulebooks</u> for handling this scenario. As the hours passed, the <u>principals</u> of resistance, establishment of unit cohesion and denial of access were applied, after which patterns began to emerge; recognizable patterns which could then be managed in the complex context domain.

We chose this illustration because the kinds of people who are skilled, if not comfortable in this domain, are people such as skilled combat commanders and, perhaps not surprisingly, politicians.

The Disorder Context Domain

This is a domain won't come into play for this discussion. By its very nature, it's difficult to tell if you are in this domain. If you can't tell which domain you are in, you're probably in this one. Somewhat like the chaos domain, moving the problem into one of the other domains is the goal and we'll leave it at that.

Table 1 The Ordered and Unordered Context Domain Plane

Complex & Chaotic ***Unordered Universe***	***Simple & Complicated*** ***Ordered Universe***
Cause-Effect relationships are not readily apparent	Cause-Effect relationships are perceptible
Same input can yield different results	Repeatable; same input yields same output
Solutions are determined based on emerging <u>patterns</u>	Solutions are determined based on <u>facts/analytics</u>
<u>Principles</u> guide	<u>Rules</u> guide
<u>Decisions during</u> the event	<u>Rulemaking after</u> the event
The unordered world is a world of <u>pattern based</u> management	The ordered world is a world of <u>fact based</u> management

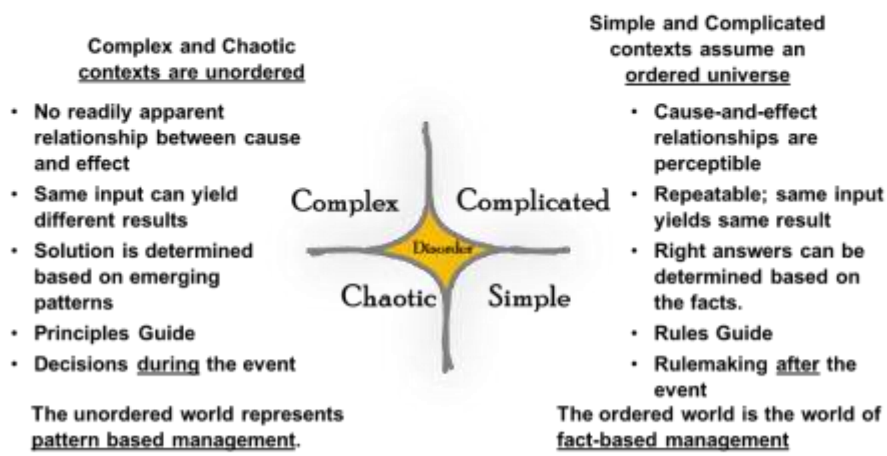
A Leader's Framework for Decision Making
David J. Snowden and Mary E. Boone
Harvard Business Review • November 2007

Ordered and Unordered Universes

Referring to Table 1 and Figure 2, the vertical border divides the domains between ordered and unordered universes. The reference works use the term "unordered" not as meaning a lack of order but an order of a different kind; one where there is no <u>apparent</u> / discernable cause-effect relationship linkage The Simple and Complicated context domains represent an ordered universe while the Complex and Chaotic domain contexts are unordered.

Also, we are careful to say that order and un-order are not exclusive of each other. In this work, that may seem to be the case. It helps to view each universe discretely for the purposes of understanding. But in the real, everyday world we live in, the two universes co-exist with each other.

Table 1 above compares and contrasts the characteristics of the ordered and unordered universes that are relevant to our discussion. Note the underlined words and phrases. Keys to our discussions will be the pattern based solutions of the unordered universe as opposed to the facts/analytics based solutions of the ordered universe.



A Leader's Framework for Decision Making
id J. Snowdon and Mary E. Boone
ard Business Review • November 2007

Figure 2 Ordered & Unordered Universes

Complex System Characteristics

Snowden and Boone make reference to characteristics of a complex system that we should pay particular attention to.

Complex systems have large number of interactions which are non-linear. Minor changes in a far removed element can cause disproportionately large consequences at the end of the tail. Complex systems have a history. The past is connected to the present and the elements of the system evolve with one another and with the surrounding environment. The system evolution is irreversible.

Complex System Interactions Illustration

You are "cruising" along in your exceedingly fast, single engine fighter at 650 KCAS about 1,000 feet AGL. Suddenly, there is a dull thud and slight shutter in the aft part of the airplane. Seconds later the Warning Caution panel lights up nearly every light, then goes off. Fire Warning lights illuminate then extinguish and the light test fails. But that light test wasn't really necessary since the cockpit is filling with smoke making visibility outside impossible. There's definitely a fire, a big fire. You know it's big because the cockpit panel at your right foot is melting. Home is almost 200nm away but there's another airfield 15nm to the southeast that's large enough to handle your fighter.

What kind of approach is suitable for handling situations like this that lead to a safe solution to the problem? What about the process (simple domain) approach? Is there a <u>rule book</u> that can adequately outline the steps necessary to resolve the situation; like a checklist response to a fire light or smoke in the cabin at 650 KCAS. What about rules for divert situations? Is there a <u>safe</u> Simple Domain response? As long as there are no other variables influencing the outcome, you probably could write a procedure/rule that governs the response to this unique scenario.

What of an <u>analytical</u> (complicated domain) approach? If we knew the temperature of the fire, its precise location and whether the source of the fire is internal to the engine or external to it, we could calculate the time left before critical systems or structural failures occur and calculate the relative probabilities for getting to the divert airfield versus the ejection option. Right?

Put on your best airmanship hat and take a few seconds to think about <u>what you would consider a successful outcome</u> and <u>what you would</u> do in this situation to achieve that successful outcome.

The problem is far more complex than the scenario we outlined, is it not? What other factors might influence a safe outcome? In fact, there are almost an unlimited number of factors that can change the answer and change it dramatically. Indeed, the definition of what constitutes "safe" is even influenced by a myriad of other, external interactions. What type fighter aircraft is this? Does that matter? Take something seemingly quite innocuous, like the latitude-longitude of the incident. Does that matter? Does it matter what the day of the month it is or even what year? Does it matter whether you are single ship or have wingmen? Over land or over water? How does your rulebook or analysis account for or not account for relevant interactions. Ok have your solution in mind? Have you decided what you would do? Let's see if we change your mind.

This isn't a made up story. It was an actual event. The pilot is Billy Sparks flying as Marlin Lead in an F-105 Thunderchief. The "Thud" had a reputation among its pilots of "dying gracefully"; it gave you everything it had and let you know when it was about to go out of control. True or not, the pilots believed that and that belief influenced decisions. The date is November 5th, 1967. What about that divert airfield? That airfield just happens to be the MiG base, Phuc Yen just a few miles north of Hanoi, North Vietnam. Marlin flight had just put 18, 750 lb bombs on that target. The fire is caused by three 57mm anti-aircraft gun hits and the airplane is literally melting around him. Safe resolution of the situation, is not defined as a successful landing at a divert airfield or even a bailout without injury. There's no rulebook that instructs one to blow the canopy off at 695 KCAS at 300 feet either . . . but it made sense at the time. The definition of "safe" is to try to stick with a burning aircraft for just a few more minutes until it could carry Marlin Lead west of the Red River. Sparky stuck with this dying airplane because he believed its reputation (an experience or story). Bailout 15nm northeast of Hanoi was not "safe."

The point is to try and illustrate the characteristics of the unordered universe and complex and chaotic Cynefin domains in Table 1. As the "pattern" of this story emerged, we suspect that your definition of a successful outcome and what you would do to reach a successful outcome changed as well. You reacted to emerging patterns. And in Sparky's case, so did he. He accentuated the emerging "good" patterns and tried to attenuate the "bad" ones. Principles not rules, guided the reactions.

And just to end the story, Sparky was successfully rescued. He was the second farthest north rescue by the HH-3 Jolly Greens out of North Vietnam in the entire war. His mantra "When I'm in the bar, the Jolly Greens never buy."

Restatement of the Hypothesis

It took forever to get here but now let us restate our hypothesis in light of the Cynefin Framework.

To restate, we are asserting that the flight test discipline, particularly the operations aspect of flight test, is rooted in the Complex Cynefin Context Domain and is best managed using the principles applicable to work in that domain.

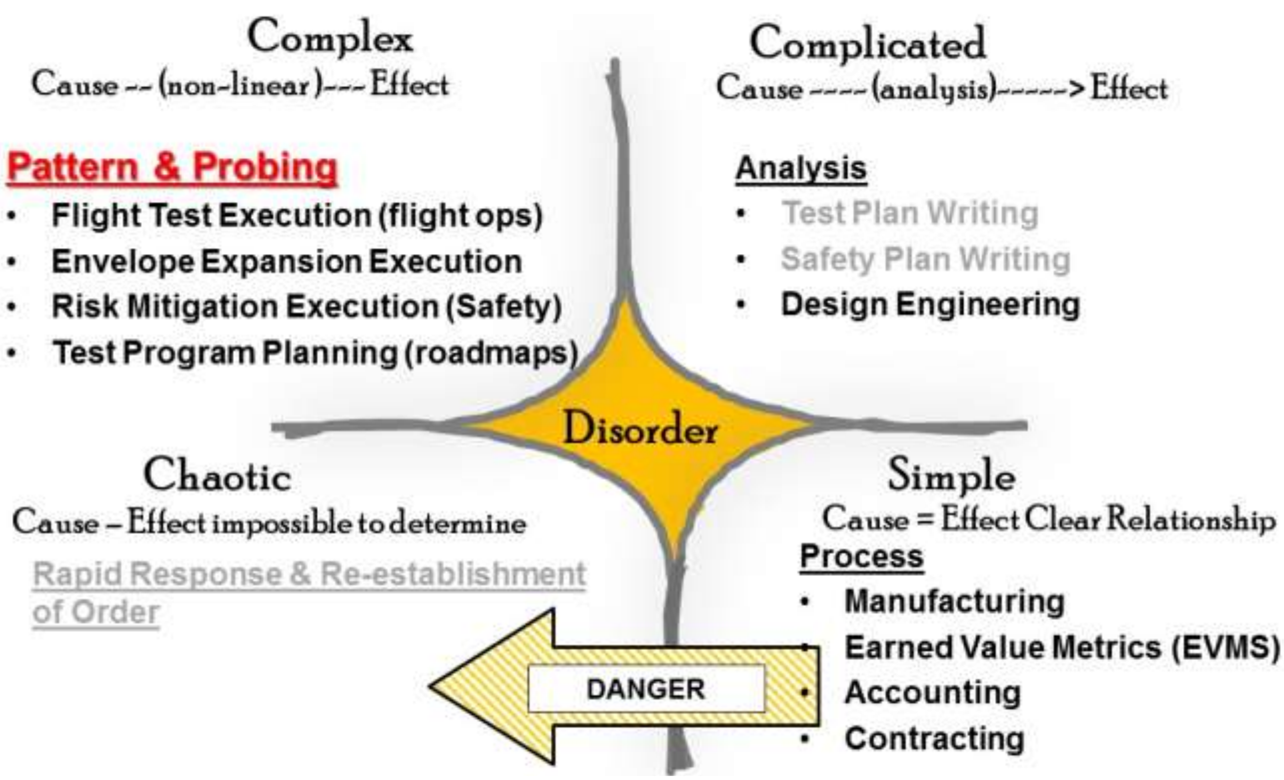


Figure 3 Cynefin Framework Context Domains Related to Flight Test

Flight Test and the Cynefin Domains

In an aerospace development environment, particularly a large military procurement program, there are a great number of organization involved. These organizations have numerous important interactions with each other. If we look at these organizations, their procedures and management protocols in light of the Cynefin Framework, one finds that, in a general sense, they fall in one of the Cynefin Domains discussed earlier. Figure 3 shows where some of the more familiar organizations/tasks/disciplines lie. Given the differences in the way problems and interactions are handled in each of the domains, one can see how communications and interaction between organizations in different domains can easily get garbled and breakdown. Crossing borders between the domains can be done, some easier than others. The border between the Simple and Chaotic domain is special, and we will spend a few sentences on that one. When crossing those borders, though, one must be cognizant that they venture into a land where people, process and interactions occur fundamentally differently and there is a danger of miscommunication, misunderstanding and confused interactions.

Common Simple Domain Organizations

We assert that rules & process (by no means in a bad sense) oriented organizations like Manufacturing, Accounting and Contracting and most program administrative functions, work in the Simple domain. All these organizations work in a clear cause-effect environment. Work utilizes best practices, is rule based and is heavily process oriented. Failures are treated as a failure in process. A new rule or change in process is the primary means of resolving problems.

The Earned Value Metric System (EVMS) is a practice rooted in the simple domain. EVMS is widely used for generating program progress reporting metrics and it works very well for those simple domain organizations where there are known-knowns and where cause-effect relationships are clear. EVMS gives accurate results in these cases and is an effective measure of the state of progress against a given task.

However when an interaction such as EVMS is imposed on organizations that are not in the simple domain, the value of the EVMS output suffers and the results are not likely to accurately reflect true progress.

Common Complicated Domain Organizations

The complicated domain is the analysis domain. Most, if not all, mechanical and electrical design disciplines falls into this domain. Expertise must be applied to the cause-effect relationship in order to effect a correct answer.

For an engineer, the schooling received in high school and the university is largely focused on teaching the math, materials, and skills necessary to develop the expertise required to analyze the cause-effect relationships in engineering design. The analysis "mindset" is engrained in us over time, becoming a natural and comfortable approach to daily workplace interactions.

For our flight test organizations, instrumentation design teams would generally work in this complicated space as well.

Software design mostly works in this space as well but, we would contend, it is positioned close to border between the complicated and complex domains. In our opinion, the best software designers are able move back and forth between the complicated and complex domains, depending on the design space they are in and the problems being encountered.

The Complex Domain

The authors contend that much, if not most, of flight test occurs in the complex domain, particularly flight test operations, envelope expansion, safety & operational risk mitigation and at least some elements of flight test program planning. Pattern recognition and probing to generate recognizable patterns is the key to being successful in this domain.

Interactions in flight test are non-linear and do not lend themselves to management by analysis or process, as in the simple or complicated domains. Unlike manufacturing or design engineering, flight testers think and conduct large portions of their daily business based on recognition of favorable and unfavorable emerging patterns. Since we "think" and "react" differently those management approaches common in design engineering and programmatic organizations is not nearly as effective with flight testers. Management, in the broad sense of the word, norms that are successful in the simple or complicated domains, do not work well in the complex domain where interactions are governed by principles, as opposed to rules and problems resolved by managing emerging patterns, as opposed to the more common analysis or process oriented management approaches.

Flight Testers and the Complex Context Domain

Pattern recognition enabled by memories of personal experience or from relevant stories relayed by others is fundamental to the flight test discipline. Take the attributes of a test pilot and flight test engineers. They are trained to:

- design (FTE) or fly (TP) test maneuvers with precision to extract engineering data
- observe flight characteristics and system performance in *comparison to desired behavior*
- translate *observations* into meaningful engineering language.

In the formal regimens like the military test pilot schools, the students are often intentionally put into an unfamiliar environments and situations in order to teach them how to apply their training in situations to which they have not previously been exposed. Placing students in these unfamiliar situations provides an opportunity for the student to gain confidence in their training, teaches them how to apply their training in unfamiliar environments (get out of the chaos domain and work in the complex domain) and lays a foundation of experience (stored patterns) which the student can draw upon in future situations. The student is learning to recognize patterns and is exposed to many different patterns, some good and some bad, with which they will become increasingly familiar as they progress through the training and continuing throughout their careers Recognizing patterns is the single most important, and perhaps crucial, element of the training. As more and more experience is gained and stories heard, the more patterns the flight tester has for reference.

Examples of Pattern Recognition

Let us give some rudimentary examples of what we mean by pattern recognition, as opposed to analysis or process oriented thought. Keep in mind that these examples are just illustrations, so don't carry the analogies too far.

Before you turn the pages, realize that these illustrations are really difficult to do in a written format, so hang in with us. It's far more enlightening when done in presentation form in front of an audience.

On the next two pages, we will ask you to look at two pictures and make a decision as to whether you are in trouble or if everything is ok. You have <u>two seconds</u> to make your decision. The time element is important, so no more than two seconds to decide. Note your decision and then turn the page and look at the second picture. Again, you have two seconds to make your decision.

Ready? Ok, turn the page. Two seconds.


Am I ok or
am I in trouble?


Am I ok or
am I in trouble?
280
290
300
310
OGS
OSS

So what did you decide about the first and second pictures. Did you decide that you were in trouble in either or both pictures or not?

Whatever you decided, in trouble or not in trouble isn't the point. The point of the illustration is to say that your reaction was based on your brain comparing this pattern (picture) with your own experience or with a story told to you by someone.

Was either of the patterns familiar to you? If the pattern you saw was not familiar, what happened? Brain freeze is a common reaction.

The first photograph is from an A-6 Intruder about to touchdown on the carrier deck. My reaction was "I'M OK". Now I've never landed on a carrier deck before, at least not from the front seat. But I've made a whole lot of good and bad landings on runways. My brain dredges up the closest pattern it can find in my memory and this picture looks pretty close to what a good landing on a conventional runway looks like. This "runway" looks awful short but the horizon, altitude and pitch angle look reasonable. I'm pretty comfortable that this is going to be a good landing and I'm ok.

Now had that picture been taken 5 seconds earlier when there was nothing but water under the airplane and the view is of the back of a big ship in my view with a tiny strip of deck apparent, the reaction may have been quite different. That's not a pattern I have in my head and comparison of the view out the A-6's cockpit windscreen from that position with my memories of landings on a runway would have not correlated nearly as well.

How about your decision on the second picture. In trouble or not? Comparing most of our experiences of good landing approaches with the view out the windscreen in this picture tells us that we've got a big problem. We are way too high to successfully complete a landing from this position. I'm in trouble! This is not a familiar pattern to us.

Turns out, however, that this picture is from a Space Shuttle approach to a runway at Kennedy Space Center. To an astronaut who has seen this picture many times in simulations and in practice approaches, there's nothing untoward in this picture and the shuttle pilot would say "I'M OK".

These two illustrations try to show how pilots and flight test engineers "think." We process data in terms of patterns, particularly sensory information. We are not "reading" or analyzing (using the terms a bit loosely). Instead, flight testers "think" by continually processing observed patterns and comparing them with familiar stored patterns. Frankly, it's much like the way you drive to work every day.

Pattern Recognition in Training for Flight Test

Let's do a third illustration. This one comes from a demonstration on National Geographic Channel's show "Brain Games". This really needs a pictorial demonstration to be most effective but let's try anyway. Consider the following and fill in the blank.

We've been having coffee with Nancy on a beautiful sunny, spring morning. In the course of conversation, she mentions that her mother has four daughters. The four daughters, oldest to youngest, are named April, May, June and ___________?

What is the fourth daughter's name? Did you answer July? Were you at least tempted to do so? Of course, Nancy is the fourth daughter. But our brain naturally looks for patterns. This illustration is a bit more intense if the same scenario is posed but we can reinforce the "familiar" by flashing up a picture of each daughter as their names are reveals. The urge to shout out "July" is almost irresistible, even if you suspect a trick question.

Aocddrnig to uinervtisy rseearch sduties, wdors can be grssoly msiplelsed but we can stlil raed them. Teh oredr of the ltteers si not ipmoreantt as lnog as teh frist and lsat ltteers are rghit.

I'm sure you had no trouble reading the gooned up paragraph above. Both of these illustrations help us to better understand how we can become complacent. Our brains are pattern oriented and will search hard to find a familiar pattern for comparison. It's an example of performing a routine, repetitive task over and over, like reading. The brain glosses over slight changes in the pattern in an effort to apply a familiar pattern. In the case of the paragraph above, your brain was easily able to substitute the familiar pattern for the unfamiliar one.

But, they say a picture is worth a thousand words.

Figure 4 A Flight Test Training Problem

If you look at this picture, there's nothing that particularly grabs your attention, right? Look again. Read the caption in the center of the picture. Still nothing? Try again. The word "YOU" is repeated. Your brain ignores it. It's redundant and unnecessary to understand the caption, right? We do this all the time in our daily lives. We filter out "extraneous" information from the familiar patterns we recognize. It is likely also that you glossed over the misspelled word "ovbservant" instead of "observant"

Think about it. Do you actually read the sign on a McDonalds when you pass by? It's quite recognizable by the color and the golden arches. There's really no need to read the name "McDonalds" on the sign to recognize the place, is there? Think someone could write "Burger King" in place of "McDonalds" and you would notice? Probably not. How many times have you reached into your pocket when someone has asked you if you " . . . have two dimes for nickle"?

The Objectives of Training for a Flight Test Team

The discussion in this section revolves mostly around the environment in a mission control room but the principles are applicable to a number of other situations.

There are two primary objectives to the training of a control room team. The first is to train the team to recognize favorable and unfavorable patterns. Equally important, we attempt to "train-out" our natural tendency to gloss over inconsistencies in those observed, emerging patterns i.e. training to avoid complacency. We do this in a number of ways, flight simulations, mission rehearsals, crew resource management exercises, emergency response simulations, table talks, etc. These are things most of us are familiar with.

However, when engineers from the design discipline or new flight test engineers are members of the control room team, it also becomes an important training objective to trainout the "analysis"thinking mindset and transfer to "pattern based" thinking.

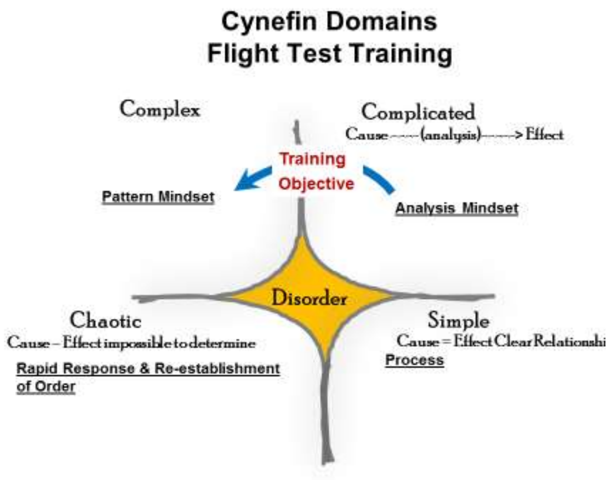


Figure 5 Control Room Training Objective

As technical people, we have spent the major portion of our education learning an expertise that is almost exclusively analysis based. For the most part, our success as engineers is largely coupled with our prowess at successfully analyzing problems, determining root cause and finding innovative solutions to the problem. We pride ourselves on tackling the tough problems, intensely focusing on them until we've reached a solution.

However, this approach is by no means appropriate for handling an emergency in a mission control room. The control room environment is largely a sensory environment. It is, in fact, a complex environment where there are enormous quantities of data from all kinds of sources, continually surfacing and moving through the environment. The data is not just wiggles on strip charts or scrolling numbers on a screen. If you are a test conductor, think about how much information you derive just from looking at the expression in someone's face or the tone of a phrase in someone's voice. How about body gestures? You don't need to look at plots or numbers to know if things are good or if they are going south, whether there is tension or business as usual. Even the pace, the tempo and the rhythm of working through the test cards tells you if the patterns you are sensing are favorable or unfavorable.

When an unanticipated emergency occurs the team is immediately placed in the Chaotic context domain. As a good test conductor, you act to stabilize the situation by applying principles (I have comm, I have warning indications, I'm prioritizing the warnings, I'm telling the pilot and control room team which checklist item we're going to work first, I've got the test director looking ahead at what could happen next, etc) and by doing so, attempt to drive the situation into a familiar pattern (from training/experience) in the complex domain from which I know (via checklist steps for instance) how to get to a safe condition.

There's a saying among us that I'm sure you've heard that says "Under pressure, you don't rise to the occasion, you sink to the level of your training." If you are an engineer, relatively new say, where is most, if not all your "training" centered? It's almost exclusively in the analysis rooted complicated domain. That's the way you have been trained ever since high school. Under pressure, in an emergency, the likely response is to revert to that analysis mindset; trying to sort through a mountain of data coming at you (bad) or the cope with realization that there is zero data (worse), separate the relevant from the irrelevant, prioritize the relevant, absorb the meaning of the data you can interpret, figure out what's going on, decide what to do to resolve the immediate situation, what to do next . . . . And to add to all that "data", you've got people in your headset screaming out orders, and queries in voice tones that don't sound right. Is that call for me or not?

Sensory overload or "analysis paralysis" is a likely outcome. If you've been in a control room setting long enough, we can just about guarantee that you have seen this occur. The individual just becomes unresponsive. Doesn't hear, doesn't see and can't think.

The training should recognize this potential and should "train-out" the analysis mindset and

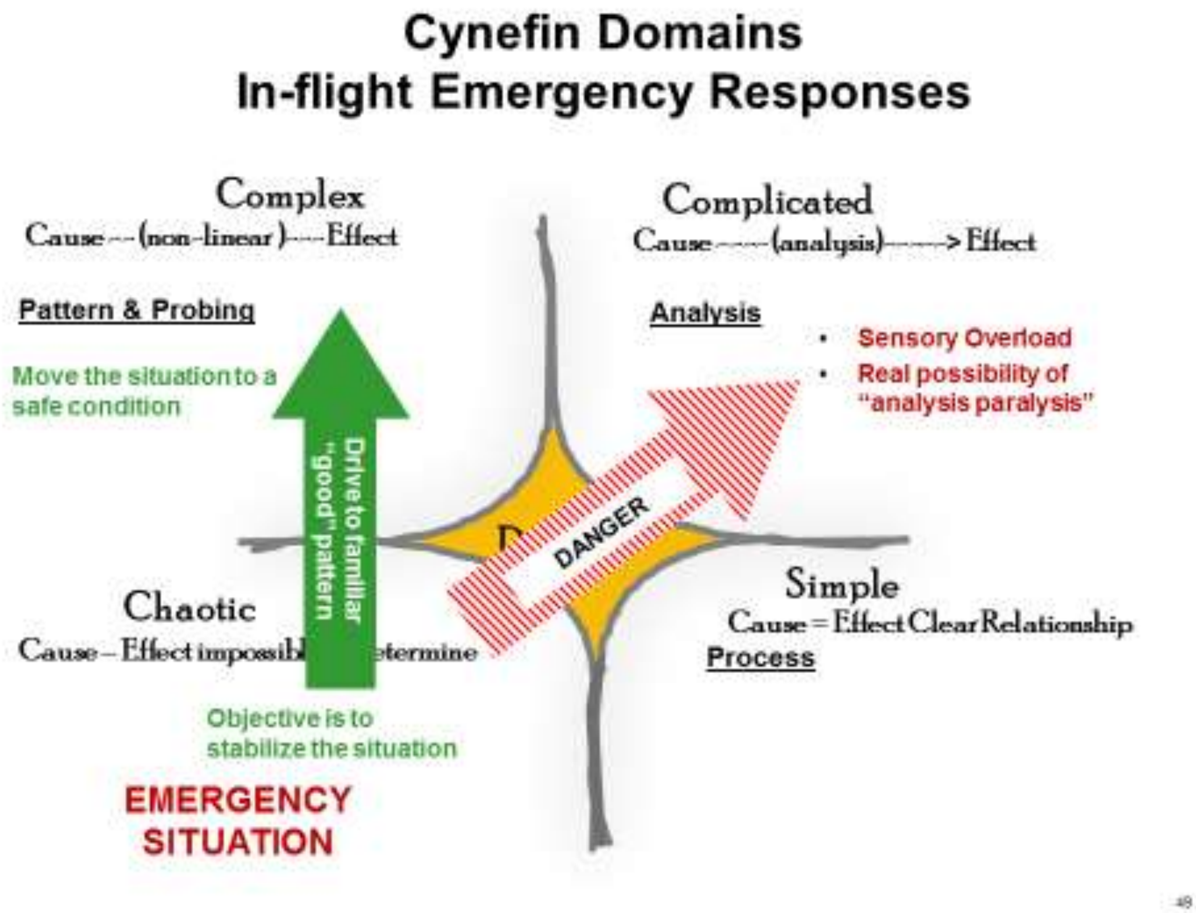


Figure 6 Perspective on Flight Test Training Goals.

move it to a pattern recognition mind set. When you all of a sudden have four of your six flight control surfaces fail, that isn't the time to analyze what happened to cause such a failure. The immediate concern it to stabilize the situation and get to a safe, or as safe as practical, state and pointed in a safe direction. There'll be plenty of time to do the analysis later.

The point is that flight test mission are conducted in a largely complex, sensory environment. The training program needs to insure that the "thinking" is moved from processing by analysis to processing by pattern association. The training program also must lay a foundation for recognition of favorable and unfavorable patterns. Application of principles is necessary for dealing with unfavorable patterns as opposed to making rules in an attempt to cover every possible situation; there are too many, maybe an infinite number of permutations.

Experience – Maturation and Replacement

For the professional flight tester and for those discipline engineers who may be at those control room consoles, pattern processing is the key to effective performance. Pattern processing/recognition allow one to operate satisfactorily, if not comfortably, in the non-linear, unordered complex context domain where most flight test operations are rooted. Experience and storytelling (i.e. others sharing their experiences) are the crucial elements to building a "library" of favorable and unfavorable patterns upon which one may draw when needed.

Unfortunately, building of that "library" takes a long time to mature. It's not something that can be gleaned from text books. It is a library built over a period of time, based on personal experiences and narrative from others. There have been a lot of airplanes saved and successes pulled from the jaws of defeat based on something learned from someone else's experience as related over a cold beer Friday afternoon after work. That "library" only comes with exposure over time.

Equally as unfortunate is that the library isn't easy to replace. When an experienced flight tester leaves the hangar for the last time, that library of experience leaves as well. For most engineering positions, the new hire has intrinsic value from the first day. They bring the ability to do the math with them from their formal education. Conventional training techniques are effective in getting the new hire "up to speed" in relatively short order; things that can be taught via lecture, study, and reading.

The new hire flight tester, however, doesn't bring a "library" that can be used immediately. Substantial exposure to favorable and unfavorable patterns, i.e. experience is necessary before the new hire flight tester can be fully utilized and is able to operate autonomously. This exposure can only occur over a period of time and it doesn't come quickly. Experiences can't be taught, at least not in a conventional teaching/training environment.

Mentoring and On the Job Training

Unlike most engineering disciplines, attempting to foster training through traditional education means are not effective. Pouring over textbooks, lectures, self-study courses and so forth do little to advance the building of a library of patterns. Unfortunately, most companies focus on these traditional training methods, which are effective for most engineering disciplines. But the payoff of such training for flight testers is meager.

The most effective methods to build that library of patterns for flight testers is through on the job training and mentoring. Allowing a new tester to actively participate in test operations is a great way to learn to recognize and react to favorable and unfavorable patterns. With a mentor watching the trainee, i.e. a safety net in place, allowing the trainee freedom to venture forth and make mistakes enhances the training in terms of both speed and effectiveness.

Formal flight test courses via one of the military or civilian test pilot schools is prized. Short courses involving test planning, mission planning, in-flight data gathering and post-flight data processing are good regimens as well. But these courses are expensive and very few, outside the military, are able to participate in a short course, much less a full-fledged test pilot school curriculum.

Actual flight experience is an area that is not widespread among flight test engineers any longer, particularly among those engaged in testing of fighter aircraft. Airmanship, basic flying skills, understanding of airspace control & utilization, appreciation for the difficulties & constraints of test maneuvering, appreciation of energy management and a host of competencies are practically impossible to impart to someone without actually flying. In the '70's, most of the flight test engineers had hundreds of actual test flying hours, many of us had well over a 1,000 test flying hours and many more operational hours. That's not true today, at least for most flight testers in the work force, and those basic airmanship skills can no longer be assumed.

Hearing stories, first hand, from others is also a good way of learning. This informal knowledge transfer is not a prevalent today as in the past either, to the detriment of the new flight testers coming aboard today. The after work gathering in the squadron bar saw many lessons passed to young lieutenants new to the business.

Professional societies such as the Society of Flight Test Engineers, Society of Experimental Test Pilots and American Institute for

Aeronautics & Astronautics offer symposiums periodically. There is a wealth of "storytelling" that occurs at these gatherings, learning what others have experienced. Alas, industry training budgets, have severely curtailed the ability of many flight testers to attend these symposia unless they are presenting or are officers in the organization.

Journals from these organizations can help. Even magazines from AOPA (Aircraft Owners & Pilot's Association) or Aviation Week have some positive effects. But can you imagine asking for two labor hours a week to let your new hire read magazines?

Flight Test Team Management Challenges

Let us change gears now and discuss the challenges that managers in charge of flight test organizations now face. In particular, a manager who has not previously had first hand flight test experience.

Having been in aerospace for over four decades, we've seen a substantial number of management techniques come into and then fall out of favor. If you are old enough, perhaps you remember "Management by Objectives." Then, a little more recently, in the heyday of Silicon Valley was the "Management by Walking Around" that came out of Hewlett-Packard. However, through the reign of all of these methodologies, taught in the business schools, the fundamental interaction within engineering organizations and between engineering, manufacturing, contracting, accounting, etc organizations actually didn't seem to change much.

Flight Test organizations have always seemed, to the authors at least, to be set apart from other organizations. Not outcast, but sort of like living on one of the Florida Keys. Sort of left alone except for a single, long & rather narrow bridge over which all the interactions with other organizations occurred. In the context of the Cynefin Framework, decades ago when there were a lot of new programs coming each year, there were a lot of interactions between organizations in the complex, complicated and simple domains. That is to say that there were a lot of "border crossings" between domains. People who worked primarily in one domain learned how to work with people in other domains, mostly through trial and error (meaning they learned what favorable patterns looked like in other organizations and avoided unfavorable patterns whether they realized that was what they were doing or not).

For example, in the 70's into the 80's there really weren't System Engineering organizations per se. That's not to say system engineering was not occurring, just that there was no discipline organization call Systems Engineering. If you are old enough to remember the times before CATIA, there were huge rooms full of drafting desks butted one next to the other. You had to talk to the draftsmen next to you and around you because you couldn't complete your drawing without knowing how the next guys drawing tied into yours. At the rudimentary level, the systems engineering sort of happened by default.

But the management tone, at least with regard to flight testing began to change in the 90's as the number of new Department of Defense programs began to drop quickly. A lot of industry consolidation, both commercial and defense, occurred at this time as a result. Another result was that with fewer programs, there were fewer "border crossings" by fewer and fewer people. The trial and error interactions occurred less frequently and people operating in different organizations had fewer opportunities to observe and learn favorable patterns. That is to say communications began to erode and along with that, the level of trust between organizations.

Take a flight test engineer and a design engineer who joined an industry employer in, oh let's say, 1968. As each of these engineers rose in their respective organizations they might be involved five, ten or more programs. As these programs come and go, these two engineers interact more and more, over the course of multiple programs, and they become more and more adept at working “across the borders” with each other’s organization. Though they don’t “think” the same way, they learn how to communicate and interact to advance the overall program to completion. By the time these two individuals reach the leadership levels they have seen maybe 10 – 15 programs and through these frequent “cross-border” interactions, have

hard data. The same kind of establishment of trust holds true for the respective organizations.

Now let’s take the same flight test engineer and design engineer but they join an industry employer in 2001. These two engineers will likely see only one or perhaps two programs over their entire career before retiring. The frequency of “cross-border” interactions will be much, much less than their counterparts decades earlier. The trial and error opportunities to learn

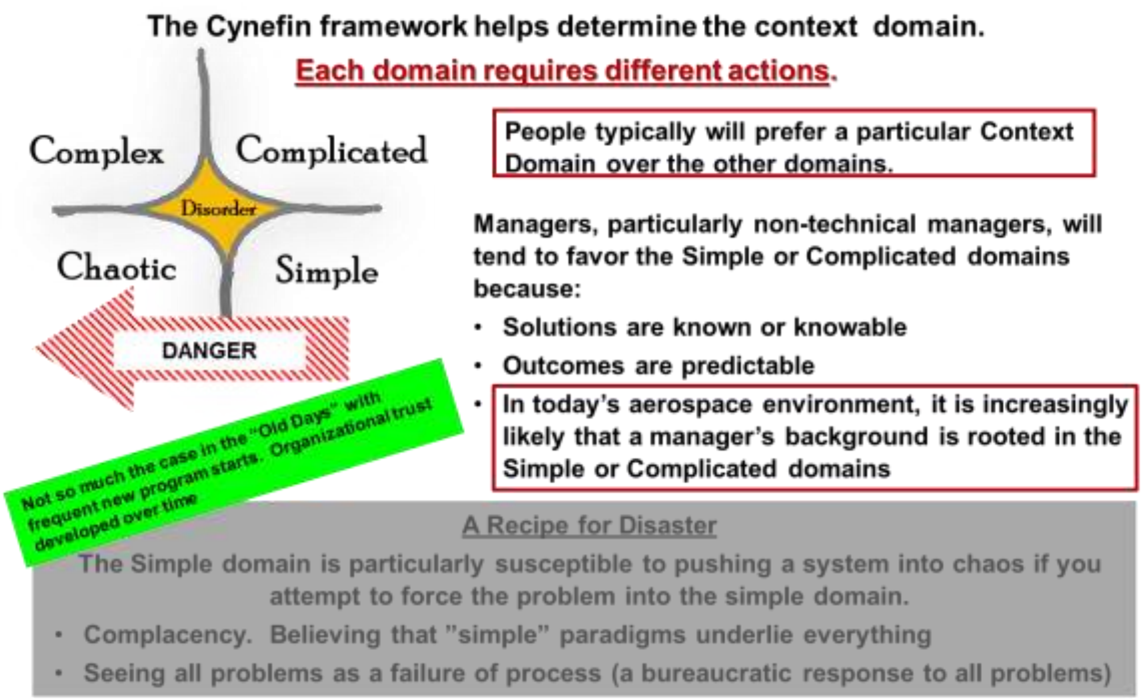


Figure 7 Management Preferences

established a level of trust in the other’s products, communications, observations, predictions and opinions. They can understand and correctly interpret the other’s “language.” Maybe more importantly the flight tester learns, sometimes the hard way, that when the design engineer is nervous, it’s time for the flight tester be nervous and cautious. The reverse also is true. If the flight tester says, “this won’t work” or “we need to change that”, the design engineer has a level of trust, established over time that allows him to give serious credence to the flight tester, with or without quantitative, unequivocal

how to interact with individuals and organization that “think” differently and communicate in different terms is greatly reduced. Indeed they may never actually realize that they do not “think” in the same way nor respond to interaction the same way. Like two witnesses who observe the same automobile accident, the both have the same data but may recount the events quite differently.

By the time these two individuals begin to enter management positions there is a real possibility that these two organizations have become, figuratively, islands within a sea of organizations. Each of the individuals on their island believes all the other islands are just like theirs. They presume that the same problems exist on all the surrounding islands and are managed in the same way, using the same metrics and same resolution methods. They have no reason to believe otherwise.

Referring to Figure 7, each of the Cynefin domains require different ways of interacting to resolve problems. Most people will prefer a certain context domain over another, because of educational background, cultural background, personality, or whatever, they just feel more comfortable operating in one of the context domains. Most engineers are more comfortable in the Complicated context domain. After all, nearly their entire formal education has occurred and taught them the expertise to comfortably work in the analysis based environment. On the other hand, seasoned pilots are comfortable in the Complex context domain. They operate in a largely sensory environment that favors pattern recognition "thinking."

It is the authors' assertion that managers today, particularly non-technical managers, will tend to favor the Simple or Complicated domains. They favor these domains because 1) the solutions are either known or are knowable by analysis, 2) the outcomes are predictable and repeatable and 3) it is likely that the manager's background is deeply rooted in the ordered universe of simple or complicated domains.

The difficulties arise when a leader views all problems as if they all reside the same context domain. Forcing all problems into one domain is often counterproductive. Imagine the product one might end up with if an aircraft bulkhead was designed using a pattern recognition approach, no analysis. You may get some sort of a product that works, at least for a little while, but it's apt to be grossly under designed or over designed if it works at all. The reverse situation, attempting to force complex problems into the complicated (analysis) domain is also not good, as we discussed in the training section with "analysis paralysis".

Considerations of Experience in Management Decision Making

One of the new management phrases floating around now days is "Data Based Decision Making." We're reminded of the witticism Spencer's Laws of Data.

1. Anyone can make a decision given enough facts
2. A good manager can make a decision based on only a few facts
3. A perfect manager can operate in total ignorance

It funny and poignant at the same time. You can live with manager 1, provided you are in an ordered universe, have access to Powerpoint, have or can create a lot of data, and if you have the patience of Job.

The second manager is the one you hope to have, particularly if you are in the complex domain.

The third guy . . . . well maybe Dilbert's boss.

When we hear the term "Data Based Decision Making" the picture formed for us is manager 1. The problem with <u>exclusive</u> use of this "Data Based Decision Making" approach is that it is almost meaningless (useless perhaps is a better adjective) in the Chaotic and Complex Context Domains. The approach assumes two key elements are present from the outset 1) that either data is available or data can be generated and 2) that the outcomes are repeatable. In the Complicated and Simple Domains, this isn't a big problem. Either the data exists so as to be categorized and processed or, through analysis, can be generated. In the extreme, the decision making stops until there is enough irrefutable data so as to make the decision self-evident. This is a form of "analysis paralysis"

In the Complex and Chaotic domains however, the data doesn't exist, at least not in a recognizable and manipulative form. This is the situation Flight Testers frequently find themselves in. In previous decades, the tester's experience was credited as being a form of data and, in the decision making, was weighed in along with whatever conventional, quantitative data existed. The tester's experience was valued because the decision maker, over a number of years of interaction and a number of programs, had learned to trust that there was real value in the experiences the tester brought to the decision table. The tester's experience represented a virtual library of patterns, some good and some bad, that could be retrieved and applied to the current problem, parallels drawn and conclusions made. In many, perhaps most situations, there were relevant patterns that could be compared to the current situation and a rational conclusion reached. Finding that the current situation did not have relevant, discernable parallels to situations of the past was in itself valuable in that it alerted one to the need to be more diligent or cautious in making the decision relevant to the current situation.

However, as an increasing number of people coming from the Simple or Complicated domains reach positions of leadership, and having no opportunity to develop a level of trust in the value of experience, the weight given to information in the form of experience (learned past patterns) has decreased and, perhaps reached nearly zero.

Consider an Example

Set a scenario where you have been given the task of laying out the detailed flight test plan for a medium or large program. In addition, you are responsible for defining the required instrumentation list of each test aircraft. The flight test involves multiple test aircraft, each will be instrumented for a targeted series of tasks. The flight testing is, in general, 10 to 20% of the total development program cost so we are talking about a large chunk of money allocated to flight testing that most managers would sorely like to use elsewhere.

You have one aircraft that will be your primary structural loads aircraft. Strain gauges must be installed/imbedded in structure which is likely to be inaccessible after assembly. That means spare gauges will need to be laid as well. After instrumenting, the aircraft will need to be installed in a ground test fixture to apply known loads at key locations on the structure in order to calibrate the output of the strain gauge instrumentation. As one may well understand, the strain gauge instrumentation design and data acquisition system is non-trivial. Meticulous, specialized skills are required to install perhaps several hundred strain gauges on structure in different stages of assembly. This is also time consuming and therefore, costly. The strain gauge calibration also requires special ground test fixtures and is a costly test in terms of time and in labor costs.

Having experienced a number of envelope expansion flight test programs, you know that there is a pretty good chance that you will lose the services of that one loads aircraft for an extended period during the flight test program, due to some unforeseen or unfortunate event. Stuff happens. Perhaps unexpected structural overload that causes cracked or yielded primary structure for instance. The aircraft could be damaged during hazardous high energy braking tests or arresting hook testing. A mechanic could run a tug into the side of the airplane. Your experience tells you that since the strain gauge instrumentation must be installed during assembly and it must be calibrated on the ground in a specialized fixture, it would be worth the incremental cost to install the gauges and calibrate a second aircraft that could take the place of the primary should the primary aircraft be lost to service for some period of time. Make sense?

For illustration sake, let's say the flight test effort is costing the program $2 million per month to run. Let's say the incremental cost of instrumented and calibrating a "spare" loads airplane is $5 million.

What chance to does the flight tester have of convincing the “data based decision making” manager that the $5 million insurance policy premium for a spare loads aircraft is a good investment for the program? Probably very low now days. The only data you may be able to provide will come from other programs, which may or may not have lost their primary loads aircraft. You are certainly unlikely to be able to show statistically significant data that can assign a probability number to the loss of the primary aircraft for some period of time. There are no analytics that will prove or disprove the wisdom of investing $5 million now and averting a $20 million cost down the road. That’s a very tough argument to advance today. Spending $5 million today on what some flight test person thinks “might” happen based on their experiences on other programs, with no conclusive analytics with which to even assign probabilities . . . . well you are unlikely to get much traction.

But it hasn’t been so long ago when “sparing” critical task test aircraft was rather common in the business. Just about everyone in the leadership ranks had personally experienced or knew of just such an event in one or more of their programs. When the flight tester came with such a proposal, the discussion centered more around what the minimum set of instrumentation could be and how can the labor cost and down time to instrument and calibrate the spare be mitigated. They recognized a “pattern” and gave weight to the pattern in their decision making just as they weighed the value of analytical data from design engineers or the accounting office.

And sure enough, it happens 20 months into the flight test program. The gunfire tests on the loads instrumented airplane show that the recoil loads are twice the predicted loads and, as an added surprise, the gun gas build-up in the gun bay is not being removed as rapidly as anticipated. So not only does the aircraft structure require beef-up to take the gun loads but modifications to the gun ventilation scheme are also required and your loads airplane is down for 5 months for structural modification.

If you made the data based decision, you just added at least $10 million to your program cost and perhaps much more if you have other airplanes whose testing depends on the loads work that the downed aircraft was doing. If you made a decision based on the “patterns” advanced by the flight tester, you will have avoided at least $5 million in additional costs to the program

But what if nothing did happen? Would the manager have been right and the flight tester wrong? We think not. The $5 million was the equivalent of an insurance policy that insured against a real, but non-quantifiable risk. Was the risk premium commensurate with the cost should the risk be realized? We like to look at the cost of the risk (insurance) premium as compared to the monthly cost of the test program. In the scenario presented here, the decision would have been marginal. Paying $5 million now to avert an unknown problem would have to add 2 ½ months to the program end date before it is worth the cost. However, if the monthly cost to run the program was say $10, $50 or $100 million a month, it’s a no brainer as far as the authors are concerned. Of course the whole problem, unlike an insurance company with large statistical samples and policy language that limits coverage to explicit risks, is that you can’t define what the problem will be nor can you quantify the statistical probability of the unknown problem occurring. What we all know for sure is that airplanes are grounded all the time for stuff like this. We know something is very likely to happen. We’ve seen it over and over again. But unless the people and program organizations that absorb the cost have some experience with flight test, and THEY know it happens too, good luck. It’s increasingly likely this program is the only program they’ve ever worked on and THEY don’t know it happens.

A second example

Take the same opening scenario as the previous example. Let's say that you plan a flight test program that utilizes 4 test aircraft and you've calculated a calendar span of 42 months to complete the test program. You've used your experience and past program histories as your "pattern" and you've assigned 9 flights/month to loads flights and 14 flights/month to flying qualities flights and 8 flights/month for flutter, and 11 flights/month for high angle of attack, etc. Your overall average flights per month per tail is 11 flights/month when the aircraft is flying and your overall average test points/sortie is 9. Based on your experience with similar past programs, you've set aside 15% of the work days for unplanned maintenance downtime per tail per year. This flight rate is also predicated on two maintenance shifts working 6-days per week, equivalent to a 20% overtime rate.

If you are a flight tester and you have a program or two under your belt, we would bet that you are thinking these numbers are well within the range of the reasonable. Present this plan to a leader rooted solely in the Simple or Complicated (Analysis) domains and you are likely to get a barrage of questions and demands for analytical data that conclusively proves that the program can't be completed faster and cheaper. This leader is looking for just what he is used to seeing and has successfully used in the past to make decisions in the engineering design space. He expects data and math centric analysis that spits out an unassailable, mathematically defendable number.

Rationally, you will be asked to quantify:

- Why can't you fly 20 times per month?
- 9 test point per flight is crazy. Why didn't you use 20 or 30 test points sortie? What are you doing during a 1 ½ hour flight that only allows you to net 9 pts/flight?
- Why should I pay 20% overtime? Why should I pay for two maintenance shifts?
- Tell me why this program can't completed in under two years?

These are all valid, rational questions that, unfortunately, don't have neatly quantifiable answers that can be substantiated by mathematical analysis, like one might calculate stress through a bulkhead. But what you have to offer, in defense of your carefully constructed plans, are repeatable patterns, which have appeared in similar programs for decades. The challenge one faces is in helping the leader to understand and to weigh, as legitimate data, these patterns and the application of these past patterns to form rational conclusions about current programs.

But there's no easy formula to help that leader reach that understanding and appreciation of the true value of experience (and storytelling) in recognizing favorable and unfavorable patterns. While you may see unfavorable patterns developing in a program, it is unnatural and uncomfortable for a leader who has not been exposed to complexity (in the Cynefin sense of the word) to lend weight to experience as a factor in decision making. Indeed the leader's own experience in the analytical world may argue against giving any weight to anything other than analytics. The only solution we've found is, unfortunately, a long term education, usually "by fire".

If a leader has a budget / schedule challenge to meet and can make the problem disappear by changing your flight rate assumption from 11 to 15 and by changing your test point per sortie factor from 9 to 20, you can be sure that there will be an overwhelming temptation to do so. There's no analytical device or conclusion at your disposal to dispute the leader's "desired" factors. You can show the statics (patterns) from multiple similar programs that support your 11 flights/month/tail and 9 points/flight factor basis, but you cannot prove analytically, that these numbers will be the most likely outcome for the current program. As we discussed earlier, in the complex context domain, ultimately, the problem is non-linear and the cause-effect relationship is only discernable in hindsight. Until that hindsight becomes available to the leader, and the value of applying observed past patterns to reach conclusions about current situations becomes apparent, your ability to influence decisions by "data based decision making" leaders is somewhat limited.

Going back to the kindergarten story, if you are the army general in charge of the kindergarten classes' playground plan, who are you going to believe, the West Pointer or the kindergarten teacher? (Flight test is the kindergarten teacher).

A Management Recipe for Disaster

The border between the Simple and Chaotic context domains is of particular concern. Attempting to manage all problems from the Simple Context domain is a recipe for disaster; and it's something that often happens, particularly in large organizations or programs. The Simple Domain is particularly susceptible to pushing a system/problem into chaos if you attempt to force a solution into the Simple Domain. A characteristic of the Simple Domain is that of "known-knowns." Management in this domain presumes that simple architypes underlie everything and failures occur because of process deficiencies or not "following the process." The response tends to be the establishment of more rules to cover the failure. In short order, the rule book becomes so large that the ability to function without breaking one rule or another drives the system to a halt (static) or into a chaotic state (dynamic).

As a very simplistic illustration going back to the picture of the A-6 carrier landing, pretend that there are a number of night carrier landing accidents. A perfectly acceptable response, from the stand point of the Simple Context domain is, don't fly at night. Simple paradigms underlie the problem. If the night environment is removed, we won't have any more accidents with A-6s landing on carriers at night. Now this is a rather crude and ridiculous illustration but we hope you get the idea

Bureaucracies are almost always involved with program organizations to some degree or another. This is not, in itself, a bad thing. Interactions, problems and tasks that can be categorized and processed in a standardized manner are most efficiently handled and resolved in bureaucracies. Bureaucracies are good at handling large quantities of "stuff" expediently, with minimum labor expenditure. By definition, bureaucracies are founded and operate on the presumption of order. In the context of the Cynefin Framework, when faced with a complex situation, the bureaucracy attempts to categorize and "restore" order to a system that is inherently unordered. Unfettered "rulemaking" is likely to occur in an effort to "restore" order and to break the system into simpler and simpler "components" in attempt to categorize the components and "process" them. As we discussed earlier, in the complex domain, small changes in the initial conditions can have inordinately large effects on the results and the results are not predictable beforehand. The bureaucracy's attempt to simplify into components, in and of itself, changes the very nature of the problem. One can see why attempts to handle complexity in the simple domain often results in chaos and nonsensical, absurd outcomes. Hence the way to resolve A-6 night carrier landing accidents is not to fly at night.

For us in the flight test community, we can see some familiar examples might be:

- New rule making as the normal response after any "failure" or incident. A rule to cover every possible situation/scenario.
- Inability or unwillingness to manage situations that have defined or undefined risk. Tendency toward zero risk tolerance (which sends cost skyrocketing to the stars) or the shifting risk to another organization.

We are not saying these things are necessarily bad. What we want to emphasize again however, is that bureaucracies see "failures" as a deficiency in the process or not adhering to the process and the response focuses on process.

The Organizational Gap

Because flight test operates largely in the Complex Cynefin domain and most other organizations operate and are managed in the rule based, ordered universe of the Complicated or Simple Cynefin domains, an organizational gap can, and in our opinion, does exist.

First of all, communications (or interactions) between organizations can get confused. The program organizations ask us very simple questions, in their mind, and they get non-answers, again in their mind, from us that are by no means simple. When will you complete the loads program? They expect a simple answer. 200 points to go ÷ 10 points per flight = 20 flights. 20 flights ÷ 10 flights per month = 2 months. Simple answer, right? The answer they get back from us may go something like "well, these last 200 points are a lot harder to get than the others, we need to perform some mandated inspections prior to flying next, the control room needs a new software configuration and there are flight control laws coming soon which will cause us to have to refly the points if we do them now so we want to wait for the new software . . . . ." And we give them a projected finish date of 4 ½ months with all these caveats.

They look at us like we can't plan our way out of a wet paper bag and the natural reaction is to assume we need better planning and more detailed and disciplined planning. The flight test program plan of record shows we should be finished with test XX by now. Instead, we are completing test YY ahead of plan but haven't started test XX yet. What are these guys doing? Are they incapable of following a plan?

We, on the other hand, conclude they have no idea of all the things that can impact a plan of execution, many of which are unforeseeable, and why all test points are not created equal. . . . And it's true! They don't have any idea. They do not appreciate how small changes can have big impacts; a newly discovered quirk in flight control software on the flying qualities airplane that shuts down the loads flight test program on a different airplane until it's corrected, for instance.

Coming from an ordered universe, these sorts of interactions are viewed breakdowns in process or failure to properly analyze the situation. We view them as out of touch with the real world and just want them to go away and leave us alone to do our job as we know it needs to be done. This organizational management "mismatch" can easily introduce friction. Imposition of rule based metric systems based on the assumption of an ordered universe are resented and resisted by those of us working in the principles based, unordered universe who know that the "data" feeding these systems is temporal at best, that it is not possible to capture all interactions driving the metric system and therefore, the projections coming out of these systems is suspect at best and is certainly not certain.

As friction between organizations grows, there is a tendency for the flight testers to become "culturally isolated." The bridge to the Keys is cut. We behave differently. We have different organizational norms. We don't "play by the same rules." We often physically reside in a location away from the mainstream organization of the program. It is very easy for the mainstream program to isolate the flight test organization. For the most part, we flight testers welcome the isolation.

The bad news is that over time, that "cultural isolation" results in mistrust and/or dismissal of flight test experience (patterns) as "data" to be weighed in management decisions making. "Cultural isolation", as much as we would like to be left alone to do our job, is a very bad evolution in the long run and to be avoided tenaciously.

Concluding Remarks

Summary

To restate our hypothesis, we believe that flight testing, particularly the operations portion, rests predominantly in the Complex Cynefin Domain. As such, it should be recognized that flight test interactions and problems are best viewed from a pattern recognition perspective. Experiences and storytelling are extremely important to the building of a personal library of favorable and unfavorable patterns with which to compare with current patterns and make decision that result in development of favorable patterns going forward.

Training for flight testers should be designed to expose the trainee to a variety of favorable and unfavorable patterns in order to build a mental library from which to draw. The training should recognize the strong bias toward analysis-based thought processes and should, instead, move the trainee into a pattern recognition response to problems and interactions in flight test operations. The tendency to revert back to the analysis-based thought processing during chaotic situations (like the initial stages of inflight emergencies) can result in freezing under pressure. The training program needs to instill confidence, and to some degree comfort, functioning in chaotic situations and to resist this impulse to revert to analysis-based reactions. This is done by employing pattern recognition built from mental library of experience and stories, guided by principles (not rules) to choose appropriate actions to remedy the situation.

Because flight testers work largely in the complex domain, and most other engineering disciplines, manufacturing and programmatic organizations are managed in the ordered universe of the Simple and Complicated Cynefin Domains, communications between organizations can easily become confused. Managers need to acknowledge the realm of the unordered universe where there is no readily apparent relationship between cause-effect and problem solutions are best achieved when decisions are made based on emerging patterns. The analysis and process based management techniques of the ordered universe (the complicated and simple domains) do not work well because of the difficulty of establishing a cause-effect relationship. Effective management should take into consideration past patterns (experience) of favorable and unfavorable interactions/results and not make decisions solely on rules and analysis. Taken to the extreme, "data based decision making" will result in either non-sensical outcomes or no decision being made at all because of the lack of facts and figures.

Challenges Ahead of Us

There are a number of challenges facing the flight test community in the days ahead. While these questions have always been present, in the past, there were enough new program starts to allow evolution of trust between individuals and organizations over time. By the time an individual reached leadership positions a level of trust had been established due to the number of interactions between Flight Test and other organizations. The authors contend that a gap exists between Flight Test and other engineering disciplines and program organizations. As that gap widens, there is an increasing likelihood that Flight Test will be viewed as an isolated community of "cowboys", resistant to management norms and largely irrelevant to the overall program. We believe it is vitally important not to let that occur and the burden is largely on us to bridge that gap. Our challenges going forward:

- How do we avoid "cultural isolation" from other engineering disciplines and program organizations?
- How do we staff programs with an appropriate level of experience distribution with very few program starts?
- How does one replace retirees with whom the most experience resides?
- What can we do to better "match" management tools that are almost exclusively

applicable to an ordered universe to flight test which is in an unordered universe? Tools such as Earned Value Metrics System and Integrated Master Scheduling.

- How do we deal with a "Data Based Decision Making" management mindset when no data exists, aside from historical patterns and experience?

Epilog

As we opened this discussion, these are the author's observations and experiences coming largely from a fighter aircraft based background in both the military and industry. It is not at all clear that our opinions and conclusions are shared universally throughout the flight test community. We are especially interested in whether there is any correlation between our limited universe and that which is occurring in the commercial transport and business aviation flight test communities.

Experiences and storytelling are important. We are hopeful that those who read this will share theirs with as many other flight testers as possible. We welcome your feedback to this article and any observations you might have.

We also highly recommend the referenced sources for more information and a much better explanation of the Cynefin Framework and its applicability to management and leadership. If you don't view anything else, watch the video *How to Organise a Children's Party*. It's 3 minutes that are worth your time. It's hilarious and at the same time illustrative of the Cynefin Framework and complexity.

References

1. Kurtz, C. F. & Snowden, D. J. *The New Dynamics of Strategy: Sense-Making in a Complex and Complicated World*, IBM SYSTEMS JOURNAL, VOL 42, NO 3, 2003
2. Snowden, D.J. & Boone, M. E., *A Leader's Framework for Decision Making*, Harvard Business Review, November 2007
3. Snowden, D.J., *How to Organise a Children's Party,* https://www.youtube.com/watch?v=Miwb92eZaJg, *October, 2009.*
4. Snowden, D.J., *The Cynefin Framework,* http://www.youtube.com/watch?v=N7oz366X0-8&feature., July, 2010.