Faster, Better, Cheaper? A Critical View…..

Paul Pencikowski

Abstract

This paper will examine the genesis of the term "**Faster, Better, Cheaper**" (FBC) as it applies to Aerospace applications. Particular emphasis will be placed on the realization (or lack) of FBC qualities within the current aerospace environment. The paper concludes with a selection of proposed methodologies that, if implemented, would be expected to realize FBC expectations.

Introduction

Great care was taken to avoid the replacement of factual observation with opinion. This paper is presented from the perspective of the author, who was supported in the two-year research period by three individuals who fit the literal definition of "expert". The supporting cast and their specialties are Ed Hanna (Management Processes), Sam Metalis (Organizational/Motivational Psychology), and Mike Davis (Software Development)[1],. All have extensive and recent experience in both aerospace and best-practice commercial-sector environments, and have a proven ability to consumate "deliverables". The author remains actively employed within the aerospace industry as a business and product-development specialist and has been in this field for the past twenty-one years. This window of experience has seen a significant rise (the Reagan years) and fall (the Clinton years) of the aerospace industry as measured primarily by the employment levels of Southern California companies.[2]

Methodology

The research-area was limited to Aerospace Prime Contractors with annual revenues above $5B, and having a major division in California. Because not all contributors to this effort had access to all companies, the criterion for the identification of an area-of-concern was *commonality*. If a perceived problem was present in *all* companies, then it became a topic "worthy of study".

Management, and/or the administration of large corporations in support of highly complex projects is most decidedly *not* an exact science. The author and the contributing researchers agree to this view. Therefore no attempt has been made to reduce perceived deficiencies to a formula. There are *no* "quick fixes".

Readers will notice a departure from conventional writing-style. In place of formal reference and bibliography sections are detailed footnotes. This work is essentially original research. The reference section contains detailed information regarding the author and contributing researchers *including their email addresses.* Readers are encouraged to open an email thread on any pertinent topic with any of the contributors.[3]

Limitations

The author and the research team are well aware that the scope of an investigation into current aerospace management practices must be limited. To this end, only data which was *seen and verified* by the entire team was allowed into the paper. Inevitably, examples will be used that appear to be opinion. Not so. Everything that is described, *happened.* Our over-powering limitation is that the number of examples must be kept to a reasonable length as dictated by the page-restrictions of selected conferences.

Goal

The main intent of the paper is to encourage discussion of FBC implementation-methodologies. The secondary goal is to foster an awareness that

[1] All four contributors (with contact information) are also listed in the *References* section.

[2] Corrected for inflation, and comparing the Reagan-high budget to the Clinton lowest-year budget, defense spending has been cut by 46%. Aerospace cuts are proportionate.

[3] Don't be shy! If you ask a question that is out of the area of expertise of the contact person, we will promptly direct you to the appropriate subject matter expert. Please, no flames.

might, in some undoubtedly small way, lead to evolutionary change *away from* "business as usual".

Background

Traditional aerospace programs are haunted by two words that have become ubiquitous in the industry: *Cost Overruns.* That it would be difficult (if not statistically impossible) to "price" the design and development of a system designed to "boldly go where no one has gone before" seems to be lost on both the general population and the agencies tasked with acquiring the "products of tomorrow". Nonetheless, we shall continue with the evolution of the FBC term….

In the early 1980's, the way to the Department of Defense (DoD) heart was the application of the moniker "lightweight, low-cost" (LWLC)… as in "lightweight, low-cost fighter." This was an actual DoD program; it never got off the ground. Examining the basics of the program, how could it be that an airplane, designed to be lighter than current products, would be *lower cost* as well? Assuming that the airplane cannot be structurally weaker than its predecessors, and must perform at the same (or higher) levels of performance as the older airplanes it was to have replaced, where were the weight and cost-savings to be found? The selling point in this case was the application of "new technologies". Unfortunately, then, as now, "technology costs money, how much technology do you want? And how many units are you buying?"

From LWLC came the 1990's acronym CAIV… "cost as an independent variable". This papers' author once took an *entire course* in grad school devoted to finding a real-world example of an "independent variable". The purpose of the course was to absolutely convince the students that "independent variables", <u>in real life</u>, do *not* exist, and therefore 1) there should be no credence given to anyone proposing that they do and 2) no research requiring the insertion of an Independent Variable into an equation as a requirement for success should be deemed valid. This having been said, The DoD came up with <u>*cost*</u> as an independent variable! Using DoD-logic in this case, one could draw the inference that cost can change independently of the product to which it is assigned, and therefore the "correct" cost to the DoD for any item, (assuming that the DoD likes "bargains"), would be *zero*. The readers are free to draw their own conclusions[4].

Following CAIV came literally dozens of platitudes-du-jour. F-B-C is but the latest. This catch-phrase was exactly that….. a platitude as opposed to a methodology. Nonetheless, the FBC logo has caught on in a big way, and now, from San Francisco to New York, there is no product-development proposed in a competition that cannot be had **Faster, Better, Cheaper!** from the offerer.

The FBC term, being a platitude, is difficult to disagree with. But as aerospace professionals, we ought to be asking ourselves if the tail isn't wagging the dog….. In other words, are we getting traditional products (airplanes, satellites etc.) built in a FBC-manner or is "FBC driving the products"? As we shall see later in the paper, the latter is the far more accurate description of the fallout from "FBC-Think".

The key elements of FBC as applied to the development of aerospace products are: (author's view in parenthesis)

1. Lean development team ("cheaper").
2. Minimal systems-redundancy in the product ("cheaper").
3. Comprehensive software-simulation in place of "test" ("cheaper").

What happened to "faster", and more importantly, "better"?

There are examples in the civilian-sector (i.e. commercial marketplace).

FBC as a requirement for success had its first manifestations in that American bastion of technology named Silicon Valley (the technology-development area located south of San Francisco). The emphasis there, driven by the investment community, is on "faster." If the product was new in the sense that it was unique, then it only had to be useful, and "better" was irrelevant. If the product

[4] Supporters of CAIV might point out that the true meaning of CAIV is that "specifications are not sacred", and that CAIV is only a way to prioritize the specifications so the customer will know where to "back off" in the event of budget problems. Perhaps.

was a replacement for another product, then it had to be "better," or "cheaper." Silicon Valley quickly learned that, of the two, "cheaper" had much greater impact on sales than "better." [5]

A second example from the Silicon Valley archives is the rapid evolution of product-development *away* from hardware and *toward* software. This occurred because in the attempt at FBC hardware it was found to be very difficult indeed to realize order-of-magnitude gains. Investors learned that "faster" was the key, because not only was the prospect of competition always looming, but that if development-times were excessive, "the market can evolve right out from under your proposed product"[6]. True, within the niche of memory chips and hard-drives, FBC success was partially[7] realized. But for the overwhelming number of "clean sheet" products (i.e. those products for which there is little or no precedent), FBC remains elusive.

Thus, Silicon Valley has spoken: its hardware-development requirement (again, for clean-sheet product development) does not lend itself to FBC-initiatives in any formal way. Software-products, on the other hand, *can* be FBC'd. Details regarding the management process, the workforce, and their interaction with the customer will follow in the Statement of the Problem and Proposed Solutions section of the paper.

Statement of the Problem

The #1 hypothesis of this paper is this: **The aerospace industry, in the primary areas in which it can exercise control (program/project management, staffing, facilities) is moving *away* from courses of action that would logically support the realization of FBC initiatives.**

Closely examining the primary difference between successful private-sector (non-defense) technology-development businesses and the aerospace industry, we find the primary difference is one of "high-performance vs. low-performance culture".

Simply put, in a high-performance culture (e.g. Silicon Valley), people are generally *not* viewed as "interchangeable parts" (i.e. people <u>are</u> unique entities). In a low-performance culture (i.e. as today's aerospace is rapidly evolving), people are becoming "commodities to be purchased at the lowest possible price". This issue would fall under the *staffing* role as mentioned in the Problem Statement. For instance:

1. Human resources (HR) groups have a demonstrated propensity toward generating and supporting the type of environment that fosters high-turnover (e.g. time-in-grade requirements for promotions and benefits, institution of "non-entrepreneurial" environments). Strict work-rules serve to justify their existence, in fact, *support* the growth of the HR organization.

2. Management now has a handy excuse for project-failure ("HR cannot supply the caliber of people we need to do the job").

Management itself, through implementation of such organizational canards as Matrix Management (Figure 1), ensures the mediocrity of the workforce because the Matrix has the following limitations (as perceived by the workers and overwhelmingly supported by management scientists)[8]….

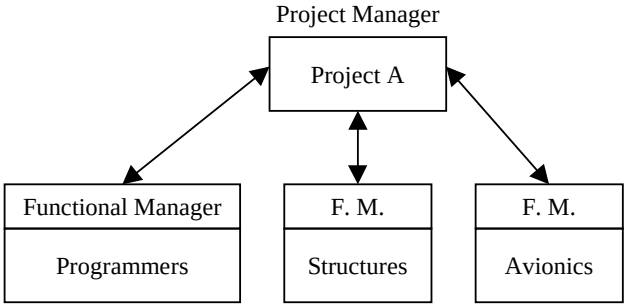

Figure 1 - Matrix Management.

1. No pride of authorship. Workers are shuffled in-and-out of various projects. Seldom does anyone do product-development "front-to-back."

2. No direct coupling of reward for performance. The mechanism to objectively determine the relevance and/or contribution of any single

[5] Effective in the commercial sector, this logic would be lost, for example, on an astronaut strapped into a space-capsule.
[6] Bill Gates
[7] "Partially" because the case can be made that few, if any, disk-drives or chips are truly "clean-sheet designs".
[8] In fact, matrix management acquired such a terrible reputation that management was forced to "reinvent it"; its new name is Integrated Product Team.

worker to any single project generally does not exist. Therefore, rewards tend to be arbitrary or "flat" as when each team member gets a similar reward [9].

3. Little job security. If lean times occur when a worker is "between assignments," then that employee is *out* regardless of past performance.

It must be strongly noted that the above statements generally elicit a howl of protest from aerospace management. From their point of view, this is understandable. However, one must only look at the general organization of aerospace companies (overwhelmingly matrix-managed), and the downsize-curve of the past 5 years (Figure 2) to see that a very significant segment of the worker-population went *somewhere*, and that the remaining population is *not* "the 99th-percentile performer". Workforce distillations should yield an improved capability. It has not[10].

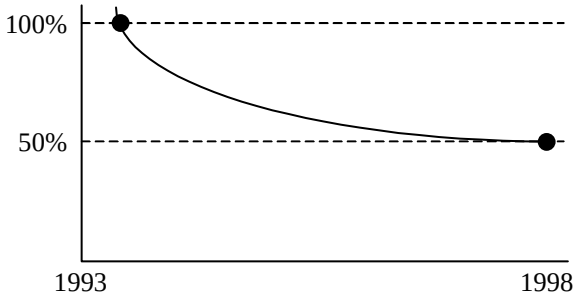

Figure 2. Aerospace Employment Approximation. (Technology-Development Sector)

In the management vein, further supporting-observations are in order. The age of the management team as it relates to overall performance is significant. Note the curve of Figure 3. First-level management is getting younger. By definition, this means the experience-base is weakening significantly because:

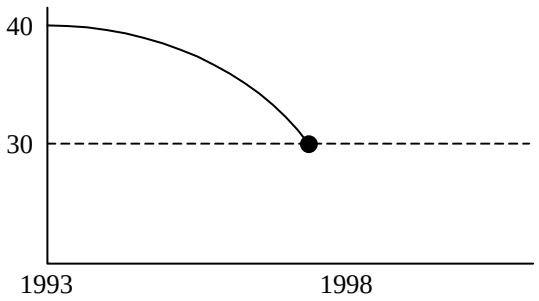

Figure 3. First-Level Manager Age (Approximate)

1. Reductions in experience-level is seldom a "positive feature" of a technology-development organization[11]. According to well-known industry consultant John Brooks, one can only "see a mistake coming" after a person has made that same mistake three times!

2. Younger managers, in keeping with HR compensation guidelines, place downward-pressure on the wage-scale of the workers[12].

This results in a negative-feedback loop as shown in Figure 4.

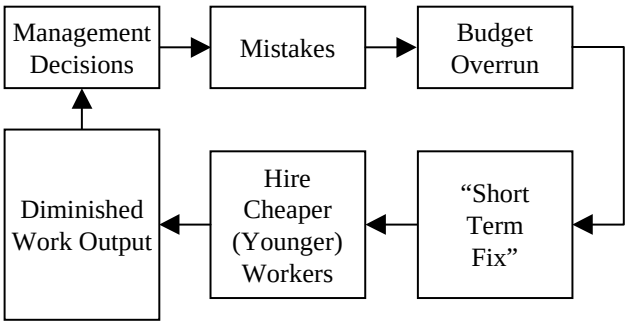

Figure 4. Negative Feedback Loop.

The negative feedback figure works like this: Less-experienced managers leading low-priced "commodity workers" combine to make an inordinate amount of mistakes, which cost an inordinate amount of money to fix, which puts pressure to "cheapen the workforce" (via attrition or "youth-izing"), which generally leads to more

[9] This is the exact type of problem that occurs in classic trade-unionism.

[10] Furthermore, if the statement *were* true and an improved capability was realized, did salaries rise to meet the productivity of the remaining "superior performers?" No!

[11] Famous Navy Saying: "Experience is that quality that allows you to recognize a mistake when you make it *again!"*

[12] Because these younger (and generally lower paid) managers tend to define the "de facto salary cap" for their department. Workers out-earning management are rare.

expensive mistakes. Some projects eventually recover; some projects just crash.[13]

John Brooks, in his seminal work *Mythical Man Month*, writes that it is generally the third design of a designer that makes the grade. His point being that the typical designer makes mistakes on the first design, was sure that he had *made no mistakes* on the second, and *"acquired some humility"* on the third (Figure 5 shows the designs/errors ratio). FBC-oriented development thrusts that include a sizeable portion of designers *working on their first design* would force the belief that these designers were relatively immune to error[14].

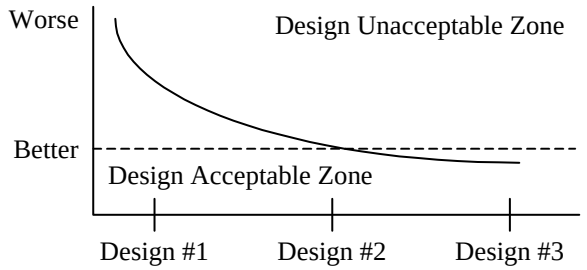

Figure 5 - Design-Acceptability as a Function of Experience.

The University of Southern California's School of Systems Management has long contended that management itself is "an organism". As such, it has the two prime qualities of an organism, those being *survival* and *growth.* With the downsizing of the aerospace industry, we have seen the elimination of "senior" technical positions followed by the hiring of multiples of junior people to take their place. In the case of managers who need a minimum number of people to keep the manager-title, or who need "more employees" to support a petition for promotion, this action solves both the survival and the growth issues. Given today's still-downsizing aerospace environment, that the average employee within aerospace is getting *younger* defies all logic (although the possibility exists that the addition of a small number of very young workers skews the data)[15].

As a separate and distinct matter relating to software development, it must be noted that aerospace software-developers ("programmers") are paid roughly 20% less than the (best-practice commercial) going-rate for their experience level, normalized for company size and location[16] (performance-benchmarks for programmers remains another can of worms). However, the pay-scale discrepancy still stands. The "revolving door" of programmer career-paths further complicates the matter (Figure 6). An interesting note is that programmers who stay at a single employer too long are left out of the buzzword-creep that infiltrates the resumes and interviews of job-jumpers; this is a serious disadvantage when attempting to move-up to "the next big project."[17] So why stay at one company?

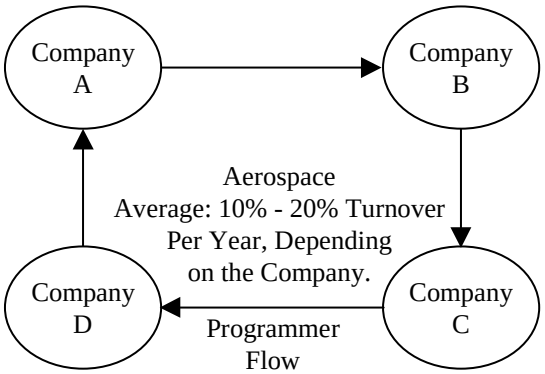

Figure 6 - "Revolving Door" of Programmer Employment.

There are literally thousands. of combinations and permutations of people, environments, and tasks which together make up a "program". The question is, therefore, are there any *common manifestations* that characterize most FBC-initiatives?

Absolutely.

[13] A few explode on prime-time TV. Does anyone care?
[14] Again, ask yourself this question: "Would I want to ride in a rocket built by first-timers"? Ex: Ariane V launch #1.
[15] Furthermore, given matrix-management, age-discrimination is apparently difficult to prove (data from the California Trial Lawyers Association).
[16] Interviews with nine Los Angeles temporary-agencies and twenty "equivalent talent" programmers.
[17] Industry typically counters that the pension/401(k) plans of "career employees" more than makes up for the benefits of job-jumping. Per the USC Business School, "pensions do not have the net present value to support this belief, and IRA's are an effective counter to the 401(k)".

Aerospace programs claiming to be FBC-endeavors[18] are uniformly guilty of the following "aerospace sins"….

1. Continual scaling back of the Functional Requirements (i.e. "what" the product was supposed to do)[19]. Program deliverables are tending to be simpler, not more complex. For example: man-on-moon ("trailer-truck"), Mars Viking ("pickup truck"), Mars Pathfinder ("toy truck")[20]. This replaces the result "meaningful mission completed" with "mission completed". Important note: This is <u>*no criticism*</u> of the Pathfinder developers (a very bright, creative, energetic group who "did the best they could with what they had". They absolutely *nailed* the "faster" and "cheaper" parts of the equation).

2. Operation on the "steep part of the diminishing-returns curve" (Figure 7). When the going gets tough, *reorganize*!

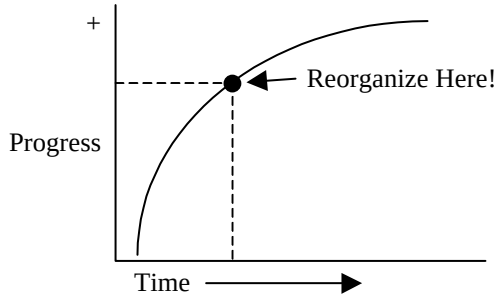

Figure 7. Diminishing Returns Curve; Early Going: High Progress/Short Time. Late Going: Low Progress/Long Time.

3. Overemphasis of the Technical Requirements ("how" to do the task[21]). This is the classic case of "doing what you know how to do, rather than *what needs to be done*," an all-pervasive fault of junior development teams.[22] Looking at this phenomena from another perspective, one could call it "confusing motion with progress". Senior-teams want to *minimize* motion. <u>Only progress counts.</u>

4. Commitment to suspect-processes (e.g. "reusable software", "matrix management"), blind reliance on purported "silver bullets" (tools designed to--Like Magic!--replace man-hours with automation; *see Appendix 1)*, and implementation of management theory (e.g. "integrated product teams"; a matrix-management re-hash[23]) in place of the proven management methods that produced such technical triumphs as Apollo and the B-2 bomber.

5. Team dissolution (i.e. layoffs) upon milestone completion. Interestingly, this "works"…. Once! After one product-cycle, the company is no longer viable[24]. Quoting management guru Tom Peters, "you cannot downsize your way to success."

6. Elimination of cross-check organizations (or, "the death of Systems Engineering"). The new management mantra is "Just build the subsystems and work out the interfaces as you go along, or just fix it later in the field"[25]. Elimination of Systems Engineering groups is the functional equivalent of "shooting the messenger"[26].

[18] Interestingly, <u>all</u> of the investigated aerospace technology-development programs, whether claiming to be FBC-initiatives or not, exhibited the same characteristics. The Aerospace Industry tends to be "notorious copycats".

[19] Again, Technical Requirements are "how" to do the task. Functional Requirements are "what" to do.

[20] The utility/value of each of these missions, as defined by media science-editors, unsupportive congressmen, (and this author) was roughly proportional to the size of the vehicle. The actual term used by Time editor Frank Pellegrini to describe the Pathfinder rover was "dinky cut-rate robot".

[21] Which increasingly means "using what I learned in school" which may or may not work "in the real world".

[22] Inexperienced teams also spend an inordinate amount of time "reinventing the wheel", albeit at a much faster pace than their predecessors. This is the main reason junior-teams seem to "quickly show so much progress".

[23] The most common occurrence is that one suspect-process is simply replaced with another (untested) process. Somehow, "cost savings" supposedly materialize.

[24] The MBA community refers to this slash-and-burn approach as "harvesting". Pathfinder and Kistler were particularly egregious examples.

[25] The Microsoft approach. Seriously, would YOU fly on a "Microsoft rocket"?

[26] Systems Engineers are typically the first to recognize system-deficiencies and "wave the red flag" to management. Since this invariably "costs money", the aerospace solution has been to *eliminate systems engineers* while, of course, keeping the System Engineering logo over the appropriate org-chart box…..

7. Implementation of Consensus Management as a "survival tool" whereby project managers unnecessarily share tasks among each other. (Why? Most likely to eliminate potential criticism from another department. Inter-departmental dependencies are an excellent way to avoid conflict).[27] See Figure 8.

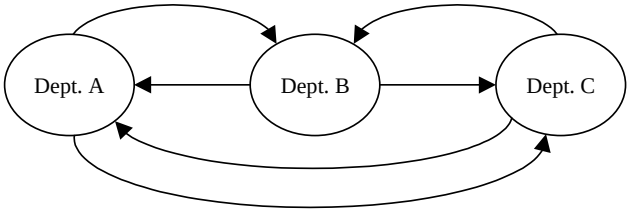

Figure 8 - Three Vertically-Integrated Departments. Similar Skill-Sets. Cross-Flow of *Discretionary* Budget. Why?

Without personal accountability, there is no personal responsibility in instances of failure. The corollary is, there are no hero-workers either.[28] In today's aerospace environment, most-often it is *management* that gets the credit and the reward, often to the detriment of the technical staff.

Proposed Solution

We can have FBC aerospace products. It won't be easy. A re-think at both the customer and contractor level will be required. Yet none of the recommendations that follow contain any magic. The following proposals are broken down into two sections: Customer, and Contractor.

Customer Responsibilities:

1. Functional requirements ("what" to do) are the customers absolute responsibility. In cases where the functional requirements are unclear, the customer may seek support from the contractor.[29] It is *not acceptable* for the Customer to off-load the definition of a specific functional requirement to a contractor[30]. Needless to say, the functional requirements must at some point become clearly defined. An example of Functional Requirements would be "Build an airplane that can fly 1000 miles at an average speed of 500 mph".

   Corollary: The Customer must *not* in any way, shape or form, interfere with the development of the technical requirements ("how" to do the job). An example of technical requirements would be "To satisfy the functional requirements of 1000-mile range and 500-mph speed, *we must build the airplane out of (what kind of?) metal and use (what kind of?) engines*".

2. Oversight groups must work to quantifiable performance-metrics and defined-deliverables. Rather than "looking over the Contractors shoulders" (i.e. "intrusive micro-management"), the Customer should only "accept and evaluate the deliverables.[31]"

3. Minimize, as much as possible, classified programs which historically operate at a 200% to 400% cost-premium to "white world" acquisitions. Alter-natively, section programs so as to minimize classification requirements. Use the cost-savings to either expand the functional requirements of the proposed product or initiate a "new start" program in an area of need.[32]

Contractor Responsibilities

1. Expand the role of Systems Engineering (SE) organizations from the traditional interface-definition role to a hands-on *mediation* role (i.e. empowered to resolve disputes between subsystem designers, and make technical decisions that prevent expensive, time-consuming subsystem-optimization). Place SE

[27] This is not a 1990's phenomenon. It was the management technique-of-choice in Medieval times

[28] In today's aerospace environment, John Wayne would never make it past cow-poke; George Patton, sergeant.

[29] This strategy, however, can be very dangerous in a "competitive shootout" where (mis)interpretation of a functional requirement may well alter the outcome.

[30] For example, it is entirely common for the USAF to come to industry-analysts with the question: "What threats will we face in the 21st century and what products do we need to defend America?"

[31] Supposedly at the core of the USAF acquisition process, significant progress in this area is yet to be seen.

[32] Adaptation of customer-responsibilities #2 and #3 are forecast to save over 50% of the projected secret-program development costs! (Consensus estimate)

members *on-site* with the subsystem designers. Make the SE group the *corporate memory*.

2. Eliminate consensus management. Both the Contractor and the customer *must know exactly who is responsible for what.*

3. Resist all efforts made by the customer to involve himself in the technical design or fabrication of the product. If this cannot be avoided, then strict measures must be taken to document the customer's directive and fix the responsibility for the action to a specific individual.

4. Eliminate expensive, redundant organizations such as project engineering.[33]

5. Return to conventional program management (where each program manager "owns" a multi-disciplined team).

6. Charge all first-level managers with the responsibility for recruiting their own employees. Minimize the HR role.[34]

7. Software development teams must be kept small, utilizing the best talent and tools that money can buy.[35]

8. Involve the vendors in the preliminary design. This will serve to emphasize "producebility."

CONCLUSION

Would the adoption of the proposed-solutions outlined in this paper guarantee success? *FBC success?* The author firmly believes that implementing the above changes in customer and contractor responsibilities would achieve, to some quantifiable degree, FBC initiatives.

Appendix 1. Productivity Tools ("Silver Bullets")

The author thanks Mr. Michael C. Davis[36] for this input. All of Mr. Davis' inputs are quoted. The authors comments are in brackets [ ].

"There are, in fact, two Productivity Tools that meet the criteria for being labeled Silver Bullets (a Silver Bullet is quantifiably defined as a process capable of achieving productivity gains of 100X-1000X, or 10,000%-100,000%). These software development tools are IBM's Visual Age for Smalltalk, and Apple's OPENSTEP."

"In addition to the standard object-oriented litany-of- advantages, both of these tools allow for the two additional actions that are the Holy Grail of software development. These are (a) the allowance of software development in the absence of complete requirements, and (b) extensive code re-use."

Mr. Davis goes on to say….

"Each of these tools requires an experienced, '99th percentile' programmer. Programmers of this caliber, if you can recognize them, currently command salaries in the range of $100K-$180K/year. Actually, unless the programmer has some intrinsic reason to work for you (family proximity, 'hip' location etc), he's a consultant being paid more than this.[37]"

[Aerospace, as a rule, has no mechanism by which to *identify* these people, never mind actually hire them.]

Mr. Davis elaborates……

"Given the turf upon which these tools/programmers operate (brokerage houses, large banks, major insurance companies), their seemingly-stratospheric salaries are not a significant barrier-to-hire (such employers routinely pay $250-$400/hr for Big-6 consulting assistance)."

[33] An expensive organization that does, as far as can be determined, "make very pretty charts out of second-hand information." Where is the value-added?

[34] Scott Adams (the "Dilbert" creator), portrays HR-types as cats, because they "play cat and mouse games" with both the employees and the job-candidates; furthermore, the HR-types filter the job-seekers to reflect "HR's own image". HR is the fastest-growing bureaucratic sector.

[35] For a detailed explanation, see appendix 1.

[36] See Reference section.

[37] OPENSTEP wizard Eric Buck executed a $300K software deliverable for an aerospace prime-contractor; our estimate for the product (C-code, traditional management), was four man-years over one calendar year. Eric delivered the executable "to spec", *in three months, working part-time, BY HIMSELF.* The customer is known to be "delighted with the product".

"Each of these tools, coupled with a capable user, virtually eliminate whole departments of conventional programmers".

[Neither tool has caught on, and likely will not. The problems are with the management organism and its' previously mentioned requirements for survival and growth, as well as the problems of the programmer population in general. Super-expensive people? Elimination of whole departments? NIMBY![38]]

Mr. Davis concludes.…

"Finally, there is no rush by rank-and-file programmers to learn the effective use of these tools. Why? Two reasons… The first is that these tools *require* the 99th percentile programmer. Anybody else will fail and (most generally) end up visibly counter-productive (an excellent way to get laid-off). The second reason is that management can quickly determine (perhaps in 2 or 3 days) the effectiveness of these programmers. There is no on-the-job-training at stratospheric salaries. Those with the capabilities (and usually, egos) required to achieve success in this scenario are *already* successful, and are most-likely working in their environment-of-choice."

"Those companies (IBM, Apple excepted) that have tried to teach these tools to their in-house staff have ultimately failed (even if the target-project succeeded) because the newly-minted 'expert' is now being recruited heavily by consulting houses, or perhaps outright competitors."

[Is there an answer? Perhaps. The standard (non-aerospace) industry approach in this case is to aggressively (i.e. quickly) raise salaries in large steps as proficiency is demonstrated, but this has its own problems, and is not likely in the aerospace industry. All in all, *a very nasty dilemma.]*

References

The author gratefully acknowledges the research and contributions of the following industry experts…

Mr. Michael C. Davis
(Executive Advisor, Software Initiatives, for Anthem, Inc.--Blue Cross/Blue Shield). Contact Mike at **mdavis@one.net**

Dr. Sam Metalis
(Motivational/Organizational Psychology consultant). Email inquiries to **smetalis@hotmail.com**

Mr. Ed Hanna
(Management Practices consultant and Managing Partner for the company FasterBetterCheaper.com). Ed will be receiving his Doctorate (in Management) from Pepperdine University within the year. Contact Ed through **www.FasterBetterCheaper.com**

The Author

Mr. Paul Pencikowski is an Executive Advisor and Advanced Projects Manager for an aerospace Prime Contractor. He was the System Architect and Program Manager for the *Display Formats and Operational Logic Subsystem*, B-2 bomber, and has 20+ years of aerospace technology-development experience. Mr. Pencikowski is also an internationally-recognized authority in the field of Process Automation.

Mr. Pencikowski holds a B.A. in Business Economics from St. Michael's College, and a M.S. in Systems Management from the University of Southern California. He is a combat-experienced Naval Aviator with over 3000 hours in F-4's and F-14's, and is a graduate of the Navy Fighter Weapons School (Top Gun). Contact him at **pencipa@yahoo.com**

[38] Not In My Back Yard! (common management cry)