Anticipating and Mitigating The Professional Challenge to Independent Verification & Validation

James B. Dabney
Intermetrics, Inc.
Houston, TX 77058

James D. Arthur
The Department of Computer Science
Virginia Tech
Blacksburg, VA 24060

Abstract: Independent Verification and Validation faces three classes of challenges: the Technical Challenge, The Management Challenge, and the Professional Challenge. In this paper we focus on the Professional Challenge, and in particular, the four phases that characterize it: Denial, Anger, Cooperation and Dependence. We believe that to implement an effective IV&V effort, one must understand the relationship among the phases and the critical issues underlying them. For each of the phases we (a) provide a characteristic description, (b) discuss how they affect the IV&V effort, (c) present representative issues and examples, and (d) describe steps to reduce the adverse impact of the three detrimental phases. The examples provided are those we have encountered while serving in an IV&V capacity; "lessons learned" guide our suggestions for addressing phase-specific issues.

Keywords:

Anticipating and Mitigating The Professional Challenge to Independent Verification & Validation

1.0 Introduction

Independent Verification and Validation (IV&V) has a unique status within the software engineering community because, at any given instance in time, its activities can be viewed as impeding or supporting the software development effort. Because of this perceived duality, attendant challenges faced by the IV&V agent are often multi-dimensional and require a thoughtful, proactive response rather than the conventional, often reactive one. In performing IV&V for a variety of programs at NASA, we have observed that challenges most often take on one of three forms: technical, management, or professional. To date, IV&V literature addressing such challenges has focused primarily on those issues stemming from the technical and management perspectives. In this paper, however, we concentrate on the professional challenge: what it is, and how to deal with it. The following three paragraphs provide an introduction to each challenge; the remaining sections expand on the professional challenge.

The Technical Challenge is to develop and execute a strategy that applies the proper techniques and tools of IV&V to derive the greatest benefit. While each IV&V project is different, both IV&V theorists and practitioners agree on much of what is important [1], [7], [8], [9], [11], [14], [15]. The principal objective in meeting the technical challenge is to locate errors as early as possible in the development process, and subsequently, minimize the effort and cost of removing them [2]. This objective places most of the IV&V emphasis on early lifecycle activities. While we have certainly made progress in meeting the technical challenge, there is still much room for improvement. In particular, we need a more automated process for and supporting tools to assist in the verification of requirements, design documents, code, and the testing process. Moreover, we also need to address ways to achieve V&V using the rigor of formal methods, but at reasonable costs. Research described in [3], [4], [5], [6], [12], [13] shows that progress is being made on this front and offers hope for better tools and methods.

The Management Challenge can be stated most succinctly as "How do I deal with the paradox that the people most able to perform IV&V are least likely to want to do it?" IV&V is a difficult technical activity that requires talented and experienced analysts. Unfortunately, people who possess the necessary talent usually prefer to apply their expertise to designing and building software systems, rather than to verifying someone else's work. Because this is a problem of perception and personal preference, it does not lend itself to a easy solution. In his work with organizational cultures, insights offered by Westrum are helpful [16], but also underscores the dire need for additional research in this area.

The Professional Challenge is often the most difficult to overcome. The basic issue is: "How do we deal with the natural tendency of the development community to view IV&V as the enemy?" Past experiences and observations leads us to the conclusion the that the professional challenge is composed of four distinct phases. Each phase represents a definitive change in the developer's attitude toward IV&V: (1) Denial; (2) Anger; (3) Cooperation; and (4) Dependence. Every IV&V project has the potential to exhibit characteristics inherent all four of these phases; to date, every project we've participated in has involved all four. In the following section we present each of the four phase and discuss how one can effectively deal with the problems they impose.

2.0 Phases of the Professional Challenge

2.1 The Denial Phase

Experience suggests that the denial phase is the most difficult one to mitigate. In the denial phase, the developer rejects the need to cooperate with IV&V, and refuses to provide them with the information needed to perform V&V tasks. Initial contacts with the developer often results in a litany of excuses justifying non-cooperation. Some of the most common claims we've heard are provided below.

- "My contract does not require me to cooperate with IV&V."

  This is almost always claimed, even if the contract or statement of work explicitly requires support of IV&V. In this situation, the customer is the only one who can effectively address the problem and force resolution to the impasse. If the customer takes the steps to force the developer to cooperate with IV&V, mutually beneficial information exchange can be expected. Unfortunately, we have found that this does not always happen. We have worked on IV&V tasks in which the only information given to us was forwarded from the developer through the customer - no direct contact with the developer was permitted. Unfortunately again, this approach severely limits the effectiveness of IV&V, and in particular, adversely impacts IV&V activities in the early lifecycle phases where the more significant benefits can and should be realized.

  It is important in dealing with this "argument" that the IV&V agent realizes that he/she must have some form of cooperation from either the developer or customer to effectively address this problem. When faced with this situation the IV&V agent should attempt to explain to the developer exactly what the IV&V

activities entail. This often helps reduce the apprehension experienced by the developer and tends to establish a modest framework for interaction. Under no circumstances should the IV&V agent threaten the developer.

- "Cooperating with IV&V will consume too much development resources."

  This is generally a rationalization type of argument. The most important step the IV&V agent to take to counter this argument is to ensure that the IV&V effort is well-planned and structured so as to minimize any adverse impact on development schedule and resources. It is also helpful to point out that resources for IV&V activities have already been allocated and factored into the development plan.

- "The IV&V contractor doesn't have the expertise to understand our project, and we don't have the budget to train them."

  Again, this claim must be false. If it is true, the person(s) tasked in the IV&V role should not be there in the first place. However, we are also of the opinion that IV&V does not necessarily require an *in-depth* knowledge of the particular project from the outset. We have found that adequate project expertise develops rapidly. More important than project expertise is expertise in the practice of IV&V. In fact, the proper IV&V mind-set is not "how would I do this", or "could I do this better", but rather, "is the developer doing the right thing right."

- "Our processes (or code, or something else) are proprietary. We can't risk divulging this valuable information to a potential competitor."

This response is most often related to and offered as a supporting claim for the first argument. If such is true, in actuality or perception, the IV&V agent should offer to sign a non-disclosure agreement.

The above claims characterize a developer who refuses to engage in an form of dialogue with the IV&V organization. We must also be aware, however, that just because we can interact with the developer and exchange information, it does not necessarily follow that IV&V can or will proceed unimpeded. For example, we have also encountered developers who simply refuse to address any issues raised by IV&V. This is particularly true for requirements validation issues. That is, issues in which the IV&V agent claims that the software requirements are not the right requirements -- and in effect, that the developer is building the wrong system. Because requirements validation issues raise concerns as to whether the customer is actually getting what he/she expects, then it is the customer who must assume the position of arbiter and make the final judgment call. Experience has shown that if the customer takes the IV&V issues seriously, then the developer will begin to take them seriously, too.

In the above paragraphs we have outlined some of the most often encountered arguments stated during the denial phase. In the following paragraphs we offer several suggestions that can assist the IV&V agent in preempting contentious and obstructive arguments encountered during denial phase:

- *Don't get emotional* - expect direction and final adjudication to come from the customer. Also, consider the developer's perspective. Many people view IV&V as the "software police". They often believe that IV&V presents a "no win scenario" where an IV&V success is, by definition, a developer's failure. If the IV&V agent begins to exhibit strong emotional tendencies, the already contentious situation can (and often does) degrade into interpersonal conflicts that harm the effective interaction he/she wishes to develop.

- *Make absolutely sure that any issue raised is real.* Early in his/her involvement, the IV&V agent will identify issues that turn out to be misunderstandings on his/her part. The IV&V agent must constantly guard against raising errant issues because any such mistakes at this point will only add weight to denial arguments offered by the developer.

- *Don't assume that the customer necessarily supports the IV&V efforts.* We have been involved in long term IV&V projects in which the customer didn't really support the need for IV&V, but "higher headquarters" mandated it. The IV&V agent and organization must earn the customer's respect and support, and retain it. This is only achieved by providing the highest quality service possible, and by making every effort to interact with the customer and developer in a professional manner.

2.2 The Anger Phase

If a denial phase has occurred, then its end is often accompanied by a feeling embarrassment by the developer - in effect, it is perceived as a loss of face. The inevitable consequence is anger, directed at IV&V. By anger, we mean shouting, invective fist pounding and emotional outbursts. We have experienced adult engineers shouting at us, calling us vulgar names, and pounding on tables -- everything short of physical contact. Our impulse in such a situation is probably normal - flee or fight. However, neither of these responses is appropriate. The key to dealing with this aspect of the anger phase is to understand that this phase really marks the beginning of a positive movement forward. More specifically, it implies that the developer now realizes that IV&V must be taken seriously. An additional fact that is encouraging and should help guide a positive reaction by the IV&V agent is the knowledge that the anger phase is usually short lived.

A second manifestation of the anger phase is an attempt on the part of the developer to get revenge. We have found this to be relatively rare, but it does happen. Revenge can manifest itself in several ways, e.g., sending incorrect information to the IV&V agent(s), directing the IV&V agent down the wrong path, or even (the developer) engaging in self-destructive behavior. An example of the extremes to which people will go in this regard was a case in which IV&V raised several issues with a requirements document. The development project manager, unwilling to submit to IV&V, retaliated by changing the scope of the document from binding requirements to "general guidelines." Such actions effectively nullified the document as a binding agreement between two parties.

2.3 The Cooperation Phase

The anger phase is always followed by the cooperation phase. Cooperation is the desired state of the relationship between IV&V and the developer. In the cooperation phase, the developer and IV&V team interact effectively, enabling IV&V to provide services that benefit all parties [2], [9], [14], [15]. The goal of the IV&V team should be to enter the cooperation phase as soon as possible, and to stay there for the remainder of the project.

The IV&V team must be aware of and recognize those indicators marking the beginning of the cooperation phase. The team must then work hard to establish good relationships with the developers, but just as important, maintain its own independence. We have found that several practices help to strengthen and maintain the cooperation phase.

- *Keep the IV&V/developer relationship relatively formal by following documented procedures for reporting and resolving issues.* This does not mean that IV&V should not discuss issues informally with the developer. However, failing to follow the formal processes can lead to a loss of accountability and a loss of archival documentation.

- *Always be cognizant of the fact that IV&V is a technical activity, not a political one.* Do not get involved in the inevitable political disputes within and between the developer and customer organizations.

- *IV&V must always perform its own analysis and provide recommendations reflecting an informed assessment of the results.* It is tempting to discuss a potential or suspected problem with an experienced developer, and to obtain advice on how to proceed. This practice is most unwise, as the developer's advice will always be colored by (a) loyalty to the development organization, (b) feelings of ownership of the product, and (c) a desire to avoid trouble.

- *Avoid training seminars sponsored by the developer.* The relationship between IV&V and the developer should be peer-to-peer. Engaging in training activities offered by the developer often leads to a perceived teacher-student relationship by some members of the development organization. This, in turn, can foster the undesirable perception of IV&V personnel being subordinate to development personnel.

- *Keep both the developer and customer well informed on the status of IV&V issues.* In particular, take time to ensure that the issue is valid, and then report it immediately thereafter.

- *Stay on schedule.* IV&V schedules must be written relative to trigger events, namely delivery of key project artifacts such as requirements documentation, source code, and test results. It is the joint responsibility of the customer and IV&V to ensure that developer project plans stipulate timely delivery of projects artifacts to IV&V as prerequisites for project milestone events, e.g., requirements and design reviews. If project milestones occur without completion of attendant IV&V, the IV&V team will rapidly become irrelevant, and the relationship can return to the denial phase.

2.4 Developer Dependence Phase

Dependence is the fourth stage of the professional challenge and arises when the developer begins to rely on IV&V for guidance or direction during the development process. Dependence is counter-productive because by definition it destroys the independence of IV&V. Dependence can arise from any of several causes, three of which are discussed next.

- The first scenario in which dependence can occur is when the technical expertise of the IV&V team is significantly greater than that of the developer. The developer recognizes this, and asks the "expert" what to do from time to time. An extreme example of this was one project where individual programmers were contacting an IV&V engineer to approve code before it was compiled: "Would you take a look at this and tell me if I'm doing the right thing?"

  Unfortunately, dependence caused by an imbalance in technical expertise cannot be entirely controlled by the IV&V team. The probability of occurrence can be minimized, however, by stipulating and following procedures that clearly define the relationship of IV&V and the developer.

- A second cause of developer dependence is the natural desire of an engineer (the IV&V analyst) to solve problems, rather than merely identify them. For example, in one instance, an IV&V analyst suspected that an algorithm was susceptible to certain numerical singularities. Before reporting the issue, the analyst implemented and tested the algorithm, and verified that vulnerability to the singularities was indeed present. Now fascinated with the problem, the analyst developed a new algorithm that satisfied the original requirements, but was not susceptible to the singularities. In documenting the issue, the IV&V analyst provided the new algorithm, which was subsequently incorporated in the product. Since the IV&V analyst developed the algorithm, IV&V, rather than the developer, came to be viewed as the expert on (and in a sense, the owner of) this piece of the project.

  This second cause of dependence can be avoided through a careful peer review of all IV&V issue reports. In all cases IV&V should ensure that the reported problems are clearly identified, and proven if needed, but that solutions are not provided. It is sometimes possible to allude to possible solutions, but never provide the solution directly.

- A third cause of dependence is micro-management on the part of an IV&V analyst. In this situation, there is an analyst who can't be pleased. This analyst has the mistaken belief that the quality of IV&V can be measured by counting issue reports. Such a person will write the same number of issues for a development item that is nearly perfect as for a development item that has numerous serious flaws. This sort of behavior can rapidly demoralize the developer to the point that all decisions are made by the IV&V analyst.

It is often tempting to overlook the problem of micro-management by an IV&V analyst. Because the responsibility of identifying problems and reporting attendant issues is a primary responsibility of IV&V, it is easy to misconstrue issue production as IV&V productivity. Moreover, if IV&V management has had to constantly justify IV&V, a natural tendency exists to promote the analyst to "hero" status, and thereby, further exacerbate an already complicated problem. Nonetheless, prevention of micro-management is a key responsibility of IV&V management. The responsibility of IV&V is not to produce issues, but to ensure that the developer is building the right product right. Note that "right" does not necessarily mean best. IV&V is not supposed to make sure that the developer follows the optimal path, but rather, that the product will do what it's supposed to do, and nothing else. We have found that a vigorous IV&V peer review process helps avoid this third cause of developer dependence. It is also important to ensure that management measures of IV&V productivity do not unreasonably emphasize issue production. Reason provides some excellent insights on avoiding the micro-management problem [10].

The above paragraphs have outlined common causes of developer dependence and have discussed strategies for avoiding such dependence. Even with such precautions, however, some form of dependence can still evolve. In fact , almost every IV&V project we have worked on has, to some extent, entered this phase. Hence, it is still important to be able to recognize and to have a strategy for breaking that dependence. The following paragraphs outline such strategies.

- In the case of the developer taking excessive advantage of the IV&V analyst expertise, IV&V must meet with the developer and customer to discuss the problem. The discussion should address the importance of independence and lead to a definitive solution outlining how to break the dependence without sacrificing project goals.

- Developer dependence stemming from an overly-helpful IV&V is also best treated by open discussion of the problem with the customer and developer. The discussion should stress that it is the developer's role to assume ownership of all elements of the product. To help avoid the situation where the IV&V analyst provides solutions, we suggest that IV&V management consider changing analysts' assignments so IV&V becomes a group, rather than, individual effort.

- An overzealous IV&V analyst who tends to micro-manage often results in a breakdown of accountability and contributes to developer demoralization. When either IV&V or development personnel recognize that such behavior is occurring, steps should be taken immediately to end it. The problem should be openly brought to the attention of all parties concerned, i.e., IV&V management, the developer and the customer. If possible, the best solution is to assign another IV&V analyst to the task. IV&V should never try to pretend a mistake was not made -- the developer already knows, and the customer probably knows too.

3.0 Cyclic Nature of the Phases

In a long term IV&V project, we have observed that the sequence of phases in the developer's attitude happens once when the project begins, then is repeated on a small scale from time to time as new people are added to the development team. If a new task is added to the IV&V project, the new task will also go through the four phases. Understanding that these dynamics are in progress, and that they are natural, will help one deal with each phase most effectively.

4.0 Conclusions

The four phases of the Professional Challenge described above are a natural part of the culture of a software project, and should be anticipated. A major goal of an IV&V organization must be to provide a supportive IV&V operation that, to the largest extent possible, maximize touted benefits. Crucial to achieving such a goal is (a) recognizing that some aspect of the Professional Challenge is always present, and (b) understanding of how to respond when faced with a challenge that is potentially detrimental. The issues and recommendations presented in this paper reflect of our experiences in performing IV&V and "lessons learned." We are firmly convinced that to implement a successful IV&V program, one must necessarily confront and successfully mitigate those issues underlying the Professional Challenge. Success is defined by an IV&V project remaining primarily in the cooperation phase and producing results having a positive impact on the quality of the project.

References

1. AFSC AFLC Pamphlet 800-5, Software Independent Verification and Validation, Andrews Air Force Base, DC, Department of the Air Force, 1988.
2. Arthur, J.D., Gupta, S., Groener, M.K., Cannon, M., and Khan, Z., "Reducing the Mean Time to Remove Faults Through Early Fault Detection: An Experiment in Independent Verification and Validation," *Eighteenth Minnowbrook Workshop on Software Engineering*, Blue Mountain Lake, NY, July 1996, pp. 28-46.
3. Cheng, Betty H. C., and Auernheimer, Brent, "Applying Formal Methods and Object-Oriented Analysis to Existing Flight Software", *Proceedings of the 18th Annual NASA Software Engineering Workshop*, 1993, pp 274-282.
4. Easterbrook, Steve, and Callahan, Jack, "Formal Methods for V&V of Partial Specifications: An Experience Report", *Proceedings, Third IEEE Symposium on Requirements Engineering (RE'97)*, Annapolis, MD, 1997.
5. Easterbrook, Steve, and Callahan, Jack, "SCR as an IV&V Tool", *Proceedings, Fifth International Software Cost Reduction Workshop*, Ottawa, Canada, 1996.
6. Gerhart, Susan, Craigen, Dan, and Ralston, Ted, "Experience with Formal Methods in Critical Systems", *IEEE Software*, January 1994, pp 21-28.
7. Leveson, Nancy G., "Safety", in Anderson, Christine, and Dorfman, Merlin, Eds., *Aerospace Software Engineering, A Collection of Concepts*, Washington, DC, American Institute of Aeronautics and Astronautics, Inc., 1991, pp 319-335.
8. Leveson, Nancy G., "Software Safety Research", *Second NASA Formal Methods Workshop*, 1992, pp 175-198.
9. Neal, Ralph. D., McCaugherty, Dan, Tulasi, Joshi, and Callahan, John, "A Case Study of IV&V Cost Effectiveness", *Software Engineering Process Group Conference '97*, San Jose, CA, 1997.
10. Reason, James, "The Identification of Latent Organizational Failures in Complex Systems", in Wise, John A., Hopkin, V. David, and Stager, Paul, Eds., *Verification and Validation of Complex Systems: Human Factors Issues*, Berlin, Germany, Springer-Verlag, 1993, pp 223-237.
11. Rogers, Richard A., "Planning for Independent Verification and Validation", *Third Computers in Aerospace Conference*, San Diego, CA, 1981, pp 15-23.
12. Rushby, John, *Formal Methods and Digital Systems Validation for Airborne Systems*, NASA Contractor Report 4551, Hampton, VA, National Aeronautics and Space Administration, 1993.
13. Rushby, John, *Formal Methods and Their Role in Digital Systems Validation for Airborne Systems*, NASA Contractor Report 4673, Hampton, VA, National Aeronautics and Space Administration, 1995.
14. Wallace, Delores R., and Fujii, Roger U., *Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards*, Gaithersburg, MD, National Institute of Standards and Technology, 1989.

15. Wallace, Delores R., Fujii, Roger U., "Software Verification and Validation: An Overview", *IEEE Software*, May, 1989, pp 10-17.
16. Westrum, Robert, "Cultures with Requisite Imagination", in Wise, John A., Hopkin, V. David, and Stager, Paul, Eds., *Verification and Validation of Complex Systems: Human Factors Issues*, Berlin, Germany, Springer-Verlag, 1993, pp 401-416.