RQ-4A Global Hawk

Photo courtesy of Northrop Grumman Corporation

THE PROMISE AND PERILS OF SPIRAL ACQUISITION: *A PRACTICAL APPROACH TO EVOLUTIONARY ACQUISITION*

Col Wayne M. Johnson, USAF (Ret) and Carl O. Johnson

Many in the Department of Defense consider the acquisition process broken. One means to address these problems involve spiral acquisition, an approach often misunderstood. It is evolutionary in the sense that the changes are incremental instead of one long, large acquisition. This gains flexibility in requirements definition and application. But few can actually state where they have seen spiral development in practice, or simply explain the difference between spiral development and block approaches or other approaches tried in the past. This paper offers a practical guide for spiral development, the major attributes in order to be successful, and one specific example. It is hoped this discussion is useful and provoke some additional thinking into how best to use spiral development.

Every five years or so, the Department of Defense evaluates the state of defense acquisition and comes to a not-so-startling conclusion. The acquisition process for defense in the United States is considered broken. To many, it seems ripe for repair. After all, tens of billions of dollars go into a system plagued by cost over-runs, late schedules, and unfulfilled expectations. Of course, there are many, many reasons for this.

Many causes are often cited. There are problems with changing or incomplete requirements or misunderstandings on what the ultimate need should be. Sometimes unrealistic or demanding expectations lead to high production costs. But so do changing budgets, inefficient production quantities, and occasional incompetence. Often the technical and schedule risks are understated by those “selling” the program, in and out of government, and wishful thinking by the end user also plays into the eventual problems that surface.

So when there are enough of these problems and a string of partial and full-blown failures occur in succession, good people in and out of government look for fixes to the underlying systemic causes. Sometimes there are causes and cures that cannot be addressed. Some would argue it is not that the problems cannot be solved. Some might suggest it is the systemic characteristics of defense acquisition that we must accept and realize in order to move forward.

Certainly there are fundamental facts that are a way of life in defense acquisition. But this is not to say that the experts are not trying to solve the problems. For example, low order quantities of fighter aircraft can be partially addressed by buying more through production partnerships with coalition allies (e.g., F-16 or Joint Strike Fighter). The downside is that partnerships can get complicated with varying requirements and the sharing of production costs. Other problems arise such as technology transfer. The fact is that an increased production base will solve only part of the problems. Other initiatives such as extensive increased use of commercial-off-the-shelf items and multiyear buys can help in some cases, but these initiatives, too, have their own baggage when it comes to solving the problems in defense acquisition. The community also needs to face the fact there is tremendous oversight to defense acquisition.

"The checks and balances put in place to ensure the acquisition office is doing it right often contribute to why it takes so long to do it at all."

The checks and balances put in place to ensure the acquisition office is doing it right often contribute to why it takes so long to do it at all. Most of the acquisition process improvements streamline around the edges and make marginal improvements in cycle time. Sometimes the acquisition community even tries new approaches. Now there is a new (some would argue, repackaged) approach to defense acquisition called *spiral development*.

As in most cases when the acquisition community desperately looks for answers that could save billions of dollars, the term spiral development has the potential of becoming little more than a buzz phrase. It is part of the overall plan to change how acquisition is done. The term has been in vogue for a little over two years and seemingly, one cannot turn around without running into a Pentagon staff briefing or business manager pitch that is not singing the praises of evolutionary, spiral development. It is evolutionary in the sense that the changes are incremental. The acquisition strategy is no longer to try to develop one big acquisition. Spiral developments are one way (some would agree the obvious way) to accomplish this. But few can actually state where they have seen spiral development in practice, or simply explain the difference between spiral development and block approaches or the old style P^{3}I (Preplanned Product Improvement) approach that has been used for decades.

The purpose here is to offer a practical guide for spiral development. The primary audience is program managers who are considering whether it should apply to their programs. Others in the acquisition and programming communities may find this helpful as well. This discussion will produce a simple (albeit nonregulatory) definition, some areas where it can be used, some attributes and must-dos in order for it to be successful, and one example in which it has a good chance of working. It is hoped this discussion will be useful and provoke some additional thinking about how best to use spiral development.

There is one point that needs to be made up front. A spiral approach is not the way to proceed with every new acquisition. This is one of the perils of taking a spiral approach. As with any good idea, it does not apply to all situations. The intended spiral acquisition characteristics are large proportion of commercial technology or previously developed military technology; a desire to shorten technology insertion life cycles; schedule urgency; flexibility in requirements for later insertions and budgetary uncertainty. There are many tools in the acquisition toolbox, and the acquisition community needs to use them all. Remember the old saying, "If all you have is a hammer, then everything starts looking like a nail." There are times when a spiral development will not work, as will be shown by looking at the definition.

A Practical Definition

For Command and Control Systems (C^2) in the Air Force, there exists an AF Instruction 63-123 that discusses spiral development. It states:

> The spiral development process is an iterative set of sub-processes that may include: establishing performance objectives; design; code; fabricate, and integrate; experiment; test; assess operational utility; make tradeoffs; and deliver. (AF Instruction 63-123, Evolutionary Acquisition for C2 Systems)

That is a good, technical definition. But for practical purposes, a spiral acquisition could also be defined as a set of acquisition activities incrementally incorporated into an evolving baseline. Each increment or spiral increases capability and does so in a rapid pace, with each spiral building on the previous spiral and spreading risk and development costs over a longer period of time. Each spiral is made up of one or more projects developed independently to the maximum extent possible. When each of the developments is ready, it is dropped into the production baseline. Testing, both internal to the program (DT&E) and external (IOT&E) is done incrementally.

Okay; with that longer, but hopefully simple, definition come several significant subtle differences between this and other acquisition approaches, and points to consider.

First, we need to define the differences. A traditional block approach involves fielding a revamped, upgraded capability. If a program office is designing a fully integrated systems solution, requiring several pieces that must all be fielded at the same time, that may require a block. A software example might be the complete changing of operating systems and languages, which requires a complete rewrite of a million lines of code. These developments could take several years and are more revolutionary than evolutionary in nature. A new fighter or completely new avionics baseline also come to mind as examples. A P^{3}I approach is a case in which the developer knows up

"A spiral approach is not the way to proceed with every new acquisition."

front what the entire development is going to look like, hence the name "Preplanned" Product Improvement. But in a spiral, the program office has an idea of the end goal, but each spiral can be changed. Therefore, it is not completely preplanned. The succeeding spirals are based on the success of the previous spiral, changing requirements priorities, feedback from the field, or changing budgets (pluses as well as minuses) and speed of development. Now it is time to turn the discussion from the brief look at contrasts with other types of development and concentrate on the key points for spiral developments.

For a spiral development, these points concern the requirements definition, acquisition strategy, and employment concept. All of these are done with the goal of providing rapidly developed, smaller projects, fielded quicker to the user. These rapid developments must be as independent from each other as possible and yet provide a capability that is often synergistic with the user. That way the program office can separate the different parts and focus on where the program needs attention. This is a main tenet of controlling risks. If the developments do not depend on one another for success, then the risk of parallel developments impacting the overall program is mitigated.

"The user has to be involved up front and understand the desired end state solution will not come with the first delivery."

Now we continue with a look at what is required for the spiral approach to work.

REQUIREMENTS DEFINITION

First, the rapid developments fall into an "evolving" baseline. The user has to be involved up front and understand the desired end state solution will not come with the first delivery. This requires a great deal of communication between the user (most often considered the "warfighter"), the program office in charge of managing the product, Pentagon staff (OSD and service level) who support the program, and the contractor that ultimately builds the system. It is essential that this process have some formalized way of doing business. This group, led by the user, must agree on content of the spiral increment and then structure the required program documentation that supports the strategy.

It is difficult to do, but there must be near-continuous feedback among the stakeholders and an understanding by all that the ultimate course may vary with time. In modern warfare, few programs work in isolation. In this age of E-mail and voice messages, a face-to-face meeting is still the best, most effective method for reaching a rapid agreement. A formal regularly scheduled meeting with all stakeholders is necessary to agree on requirements for the next spiral. As different systems evolve, the contribution requirement of any one program may need to change. Spirals can accomplish changes if there is an opportunity for flexibility. Flexibility is one of the main strengths of spiral development. But the group must function as a team. The team's success requires communication and trust.

To be honest, if the user does not trust the acquisition community, this will not work. The old rule of thumb has been, if users do not insist on getting everything needed up front, then they end up settling for the first capability delivered, and never get the end item envisioned when the requirements document was first penned. A spiral approach requires a spiral requirements document. Users must state up front that they are willing to accept less-than-perfect systems in the beginning. They will test it, field it, and use it knowing it does not meet all their needs, but it does have operational utility. This is a big commitment on several levels.

To begin with, Operational Requirement Documents (ORDs) are hard to write. They often are plagued with either vague or incomplete requirements, or the ORD goes the other way and directs a specific, point solution. Either end of the spectrum causes cost and schedule impacts. Either the program office is left wondering what is required, or worse, faces the "I don't know what I want, but I will know it when I see it" dilemma. This gives little direction to the program office and less direction to the testers. Chances are, the user will not be satisfied with the end product. The second problem of a point solution is just as bad. The user will state, "Get me what I saw at…" and the program office can fill in the blank with a specific solution. The influence for this point solution might have been in a marketing pitch given by a contractor or demonstrated at a trade show or a written magazine article. Experience has shown that the system is never quite like the view graph presentation that spawned it, and the end system is usually more difficult to modify and integrate to meet the end user's needs.

A spiral ORD requires a new way of thinking. The user must work in concert with the program office to state the requirements in such a way that a so-called 80 percent solution can first be fielded. Because the team goes into the acquisition knowing they will modify the system as they go, the architecture of the system must be built so it can be modified. The ORD must accept that the first few deliveries will not meet all the needs; however, the goals will be incrementally accomplished. The testers must be open to the idea that in testing a spiral ORD, the definition of "effective and suitable" is going to be widened to fit the strategy.

To help lay out the strategy, the ORD could state the objectives in such a way that the end goal is understood, but not spelled out with a hard and fast time and date as a block approach would. It might be as follows: Spiral 1, deliver an aircraft that carries at least 80 percent of the Spiral 4 payload requirements; Spiral 2, upgrade the infrastructure so it can integrate advanced navigation and communication avionics when they are later developed; Spiral 3, update the avionics and logistics; Spiral 4, allow for higher payload and reliability; and Spiral 5, to be determined. The requirements process needs to be flexible.

"Users must state up front that they are willing to accept less-than-perfect systems in the beginning."

The balance in this approach is simple to state, but harder to sell to a user community burned with failures in the past. The requirements must be sufficiently broad to give the acquisition community latitude to make trade-offs, yet give sufficient guidance so the acquisition team (government and contractor) knows where it is trying to head. In testing and fielding the early systems, expectation management is important in order for users to avoid getting discouraged ("It doesn't do everything I want, why should we build more?") and allow the lessons learned from early operations use to be incorporated in later spirals.

Somewhere about now, most experienced acquisition professionals may still be wondering how this differs from a block approach. To emphasize what was pointed out earlier, there are several key differences. First, in a spiral approach the program developer may make improvements that do not readily seem to support the end goal. An example might be communications open system architecture. In an earlier spiral, the program might put in the hooks for later improvements without actually making the improvements for another spiral. Some would argue for making all the known upgrades up front. Why not just incorporate the communications processor upgrades at the same time as the updates for the avionics boxes and electrical power to accommodate them? It might take a little longer, but so what? The result would be getting the final capability to the user sooner. That approach is the traditional block approach.

"The first challenge to acquisition is the development of a framework to place the spiral requirements document into action."

In a spiral acquisition, a program manager might know what the power requirements are and might preposition the wiring harnesses, but might not know what the commercial standard computer processor will be. Or, the processor might not be fielded yet and the program is now on the leading edge. Those architecture hooks may not cost much in budget or time up front, so putting them in gives the program flexibility. Trying to do it all immediately means the program manager must know everything he or she will eventually need. This occurs sometimes, and can often be done successfully, but it is a block approach. The second difference is that a block approach is usually considered a final end item. The aircraft fielded in block 20 are not always meant to go back through retrofit or field install to become a block 30 (the F-16 aircraft is an example). But in a modern spiral approach, the program can expect and plan to take the aircraft back through an upgrade to a later spiral. There are some unique challenges to this as well, and that "peril" will be addressed later.

ACQUISITION STRATEGY

The first challenge to acquisition is the development of a framework to place the spiral requirements document into action. The acquisition community is not immune to communications failure and the program office must consistently work to keep the communications lines open with the user and test community. This is the formal, regularly scheduled meeting we discussed earlier.

We have resisted naming that group up to now, for fear of incorporating more Integrated Product Teams (IPTs) than necessary into the thought process. But in fact, the name used in AFI 63-123 is Spiral Development Integrated Product Team (SDIPT). We believe this is an essential part of the process and needs to be in place, tailored to meet the program, so that the program office stays in step with the user. That does not mean the user and testers run the day-to-day acquisition. It does mean they have insight into what the program office and contractor are doing.

Keeping the user informed is not always the same as asking permission. The team needs to realize the difference between insight and oversight and act accordingly. Spiral development is a different way of doing business on all sides. There is an element of trust that needs to develop. It will not happen overnight. All sides need to work on it. Flexibility in testing will be very important. The testing community cannot become rigidly fixed on an end requirement, or a spiral development will not work. One of the major tenets of spiral development is the management of risks. If too much capability and technology needs to all work at one time, then the risks go up. The burden of development will increase time, costs, and risk to success. In the words of an old engineer, "You can make the rock so big, no one can carry it." By cutting the development into smaller compartments, a spiral approach can manage that risk.

For example, the program manager might have several different development activities going on at the same time. If they are developed in such a way the interdependencies are minimized, if one development falls behind, it does not impact the rest. This approach will keep the program from depending on one miracle that needs to occur to keep the schedule on track. A more acquisition-friendly way of saying it is this: keep the critical path simple and singular. If too many risky projects need to occur before any project can have success, it is not a critical path, it is a train wreck.

The acquisition strategy must take a systems view: not just for the next spiral, but for long-term flexibility. Flexibility is a two-edged sword to some of the old war horses in acquisition. We see it as an imperative with tight budgets and dynamic requirements. An example scenario will illustrate.

"Flexibility is a two-edged sword to some of the old war horses in acquisition."

The program manager for a spiral development sets up a series of spirals, each fielding a capability update at the beginning of each production lot. The upgrades are small, but all of them are required by the ORD. The program office deputy gets a call from the Pentagon staff saying that, due to other priorities, the program has lost 15 percent of next year's development budget. They want to know how this will delay the next spiral. Answer? It will not. What it will do is require the program manager to contact the contractor and user and advise them the bottom priority item of Spiral 2 will have to slip to Spiral 3, but the rest of the items will stay on track. When the individual projects are developed and tested, they will be incorporated into the production line. So in other words, there will not be huge impacts to the overall program, but some spiral development content will slip. The program will have the flexibility to move on. The peril here is it makes it easier for the program to take hits to the budget and survive. That sounds like a contradiction. But those in Washington who are straightforward will say it puts the program at risk for cuts because it is a flexible program that can take cuts. It becomes easier for them to take money from the spiral program than from an old-style program that can show any cutbacks will delay fielding *any* capability for one year and thus will not meet their ORD or acquisition program baseline.

Some may be cynical and argue that kind of flexibility works against the spiral program, but do not believe it. If the United States is going to continue to field the best military in the world, flexibility is needed. Our leaders should demand all programs be as flexible as possible, not "bullet-proof" to cutbacks because of resulting dire consequences if cutbacks occur. Well, enough of the patriotic rhetoric.

"If the United States is going to continue to field the best military in the world, flexibility is needed."

There is a programmatic practical side as well. Not all cutbacks are from Washington, DC. When a program gets in trouble and an element costs more than budgeted, the program needs to be able to survive with as few consequences as possible. Remember, a five percent increase in the cost of a project impacts the program like a five percent decrease in the budget dictated from above. So the program can avoid risks with several parallel developments by making them as independent as possible.

There is a side benefit as well. Program offices and contractors only have so many experts to manage the difficult tasks. If a spiral has five parallel developments, but only one of them is required for the other four, then the contractor and government know where to focus their maximum attention to manage the risks and to watch for trouble. If all five are interdependent and all must work for the program to fly, then the risks are more difficult to watch. It is said that one of the reasons Charles Lindbergh decided on a single-engine aircraft for his historic crossing of the Atlantic instead of a two-engine plane was due to the risk of adding the second engine. At that time two-engine aircraft could not sustain flight on a single engine. By having two engines, both required for flight versus one, he would effectively double his risk of failure if he lost an engine. The lesson here is, keep the risks simple and singular.

EMPLOYMENT CONCEPT

For the user, the employment concept is complicated but pays dividends early on. A block approach may take years, and a traditional development could be structured with a three-year development and one year of tests before the user knows what he or she has. This is easier to manage, because if it does not work and fulfill the ORD, it is sent back. And the user has many years to prepare for its fielding. But of course, that means years without having had any capability as well, and the chances the user's requirement might have some changes over that four-year period are pretty high. So how should the program be set up?

The first thing is to get the user with the program office and testers and work out the priority list of capabilities they would like to see fielded. This gives the program office a means to make focused decisions. Of course, this requires the user to trust the program office to combine capabilities where efficiencies occur, sometimes taking the requirements a little out of order.

An example would be an engine upgrade (Priority 4) with an alternate fuel certification (Priority 6) in Spiral 1 and Spiral 2 containing the navigation avionics upgrades (Priority 5). Communication here is again essential. The program office would explain that by doing both the engine and fuel upgrade in the same spiral, the program could save resources in wind tunnel tests, thus lowering the sum of the development costs as opposed to placing these priorities in separate spirals, all other things being equal.

Another consideration is logistics for the fielded systems. Do not overlook the challenge that will face the logistics team. Under this concept of spiral development, a program could easily have three different configurations of the system out in the field at the same time. Before the roar of the nay-sayers gets too loud, allow us to point out that the situation exists already. With diminished manufacturing sources available, technology improvements, block approaches, and P^3I efforts, multiple configurations currently exist on many fielded programs. The difference is with a spiral approach, the program expects, plans, and condones different configurations, allowing capability to be fielded more quickly. The program does not expect everything to be "saucered and blown" before it reaches the field, which is rare anyway.

A Practical Example

It is time to look at what can be considered a practical example of spiral

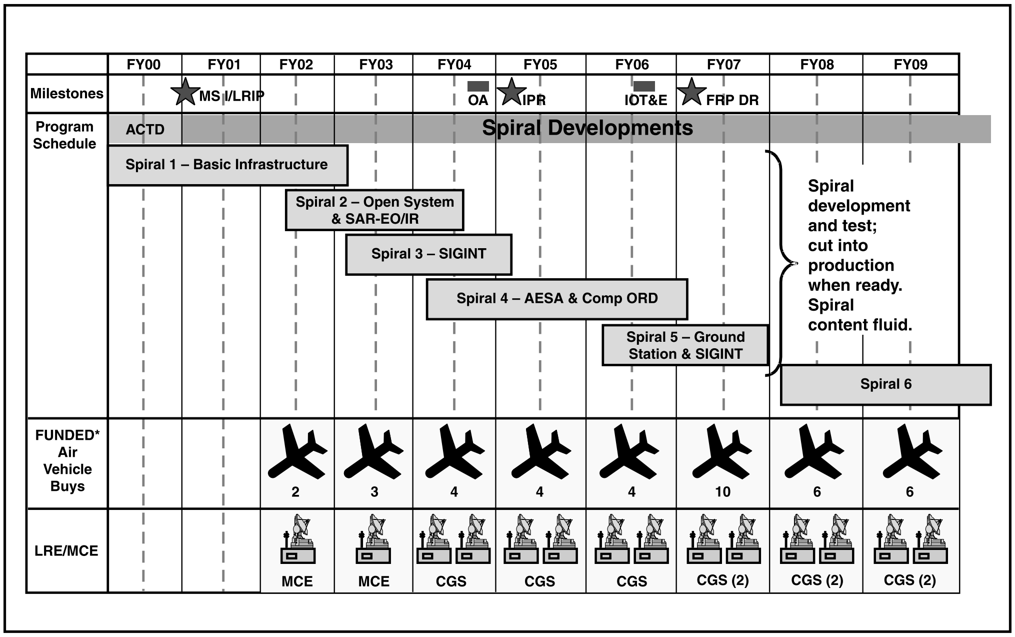

Figure 1. Draft Example of a Global Hawk Sprial Development

development, the Global Hawk unmanned air system. The Global Hawk started as an Advanced Concept Technology Demonstration (ACTD) program and entered Engineering and Manufacturing Development (EMD) in the winter of 2001. The problem was the ACTD was not what the user wanted to field and the development into what was wanted as a two-staged development was going to take seven years and two configurations before the user would gain the type of capability wanted. The initial capability was very basic, and the program was challenged to field additional capabilities quicker by the Commander of Air Combat Command. Although the program office touted the initial plan as a spiral development, it was not responsive to the user needs without a major shift in thinking. That shift became a transformation of the original program.

Given the task to develop a more rapid way to field capability, the program office and contractor took these two "spirals" and examined their characteristics. These spirals were three and four years long each, so for all intents and purposes they were blocks. Working with the contractor and user, the program office looked at how these individual projects could be logically broken out and allowed fielding priority upgrades more quickly. In the summer of 2001 the Global Hawk became a Transformation Program, and based on direction and funding, became a true spiral development with capability being dropped into

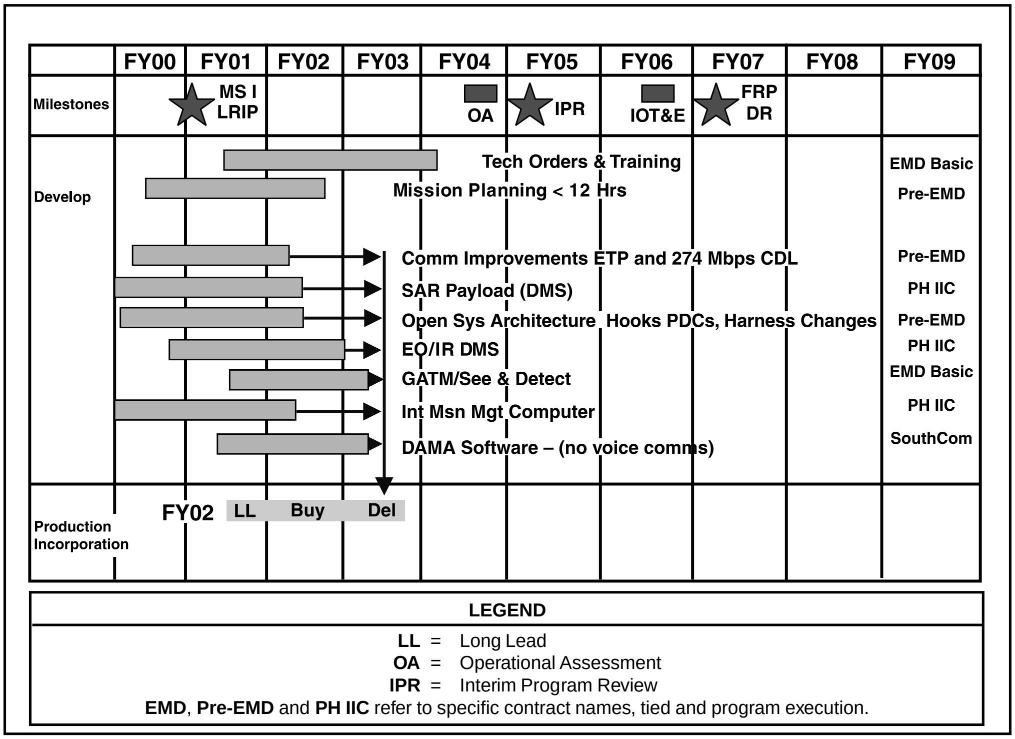

Figure 2. Example of an Initial Spiral Designed to Deliver Baseline Capability

the production line on a yearly basis (see Figure 1). The user conducted a requirements conference to get out in front of the formal requirements process, because the dynamic program was moving faster than the standard ORD process could keep up. The results are a flexible, dynamic program that is more able to keep up with changing requirements, better communication with the user, and more capability fielded in shorter time.

The first spiral would line up to deliver a baseline capability. Additional spirals would follow rapidly, allowing the user to interject or remove forecast requirements. As long as the interdependencies are kept to a minimum, then maximum flexibility can be achieved. In each case a systems review and risk analysis would need to occur to ensure the program content can function (see Figure 2).

Additional spirals would follow the same format and logic but would drop capability into the production line when ready (see Figure 3).

THE PROMISE OF SPIRAL DEVELOPMENT

To sum up, spiral acquisition can offer some unique advantages over traditional acquisition if done properly:

- Incremental capabilities can be fielded quickly, giving the warfighter more capability sooner.

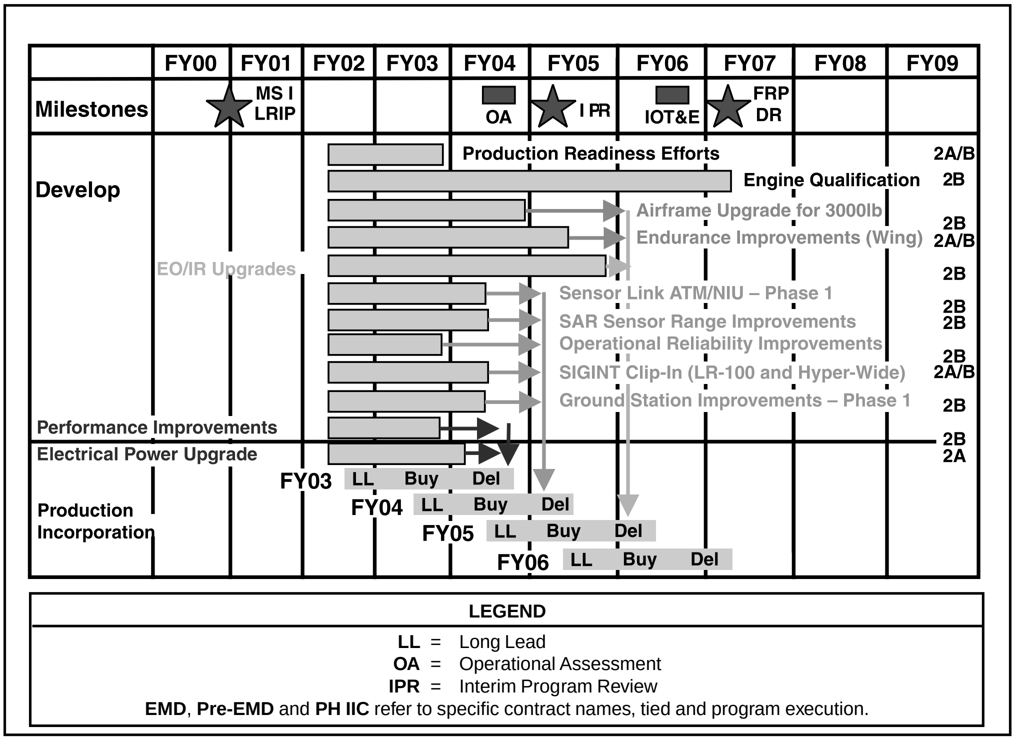
FY00
FY01
FY02
FY03
FY04
FY05
FY06
FY07
FY08
FY09
Milestones
MS I
LRIP
OA
I PR
IOT&E
FRP
DR
Develop
Production Readiness Efforts
2A/B
Engine Qualification
2B
Airframe Upgrade for 3000lb
2B
Endurance Improvements (Wing)
2A/B
EO/IR Upgrades
2B
Sensor Link ATM/NIU – Phase 1
2B
SAR Sensor Range Improvements
2B
Operational Reliability Improvements
2B
SIGINT Clip-In (LR-100 and Hyper-Wide)
2A/B
Ground Station Improvements – Phase 1
2B
Performance Improvements
2B
Electrical Power Upgrade
2A
Production
Incorporation
FY03 LL Buy Del
FY04 LL Buy Del
FY05 LL Buy Del
FY06 LL Buy Del
LEGEND
LL = Long Lead
OA = Operational Assessment
IPR = Interim Program Review
EMD, Pre-EMD and PH IIC refer to specific contract names, tied and program execution.

Figure 3. Example of a Secondary Spiral Designed to Move Upgrades into Production

- Risks can be spread across a series of spirals, allowing demonstrated capability to the user.

- Lessons learned in earlier fielded spirals can be added to later spirals, making the acquisition community more responsive to user needs.

- Lessons learned from operations, such as Enduring Freedom, can be interjected more quickly.

- Technology can be incorporated faster — lean, agile acquisition by its very nature.

THE PERILS OF SPIRAL DEVELOPMENT

The preceding is an example of how spiral development can work; however, we would be remiss if we did not add a review of the perils touched on throughout this article. The difficulties and problems with a spiral approach follow:

- A spiral approach does not work if the user cannot accept fielding an 80 percent solution in the beginning (an example might be a nuclear power station).
- Spiral acquisition is inherently flexible and could lead to budget cutbacks in difficult times because the program can weather the impacts without catastrophic failure (the peril is it could be viewed as a "cash cow" for less flexible acquisitions).

- The test community must be on board to negate an automatic failure (planned partial long-term capability must be seen as a success).

- The requirements must be flexible with possible updates in the middle of acquisition.

- Another peril is the false comparison. Some will compare the first spiral of a new system with the legacy system it is destined to replace. Worse yet, they will compare the new system's first spiral with the next planned block upgrade capability of the legacy system. Then the question will be, "Why fund the new system that does not greatly perform over the older system?"

- The logistics community must buy into having multiple configurations in the field.

- Communication must be continuous and trust must be built among the team, but each must know what is in each "job jar."

- The financial community and leadership must accept that content in later spirals is subject to change based on technology and user needs. They must accept placeholders in some cases and budget for that.

CHECKLIST FOR THE FUTURE

If the program manager believes the proposed project is a good candidate for spiral development, he or she needs to ask these questions:

- Does the program have a good foundation program on which to build? If there is no basic capability, that is where the program must start. In the case of Global Hawk, the ACTD provided a basic capability on which to build.

- Are the forecast upgrades severable? Spirals need a series of discrete, smaller upgrades; otherwise, it is essentially a block program.
- Can the user accept incremental changes and multiple configurations?
- Can the user accept lower performance initially, at a lower price and shorter schedule? Most would say they can accept the lower cost and quicker schedule, but there are no free lunches. An initial quick capability comes with a cost in the fact it will not be 100 percent of what the user wants. A spiral ORD is needed.
- Does the program have the support of headquarters for this type of development? It is difficult to get the staff to buy into the concept, but if they see the benefits, it can be successful.

SUMMARY

This has been a very quick overview of spiral developments. In a practical sense, spiral development must be a partnership among the user, program office, contractor, testers, and headquarters in order to work. It is hoped that program managers and others have found this discussion helpful. Spiral development is not a panacea for every activity in defense acquisition. But it can shorten technology cycle time and cut down on risks. The payoff is quicker capability upgrades, manageable risks, and flexibility in development and fielding the systems necessary for the warfighter to fight and win the next war.

Colonel Wayne Johnson, USAF (Ret), former Director of the Global Hawk program, WPAFB, OH, was commissioned through ROTC in April 1977. He is a Command Pilot with various flying assignments including the B-52G, B-1B, T-38 and T-37. Previous acquisition assignments include being a program manager in the B-1 and F-16 program offices, and as director of Joint Airborne Signals Intelligence Programs. As Global Hawk program manager, Col Johnson was responsible for transitioning the Global Hawk Unmanned Aerial Vehicle for a highly successful ACTD program to a formal Engineering and Manufacturing Development program and into production. Col Johnson is a 1998 graduate of Air War College and led the Global Hawk team that won the 2000 Collier Trophy.

(E-mail address: wayne.johnson@daytonaero.com)

Carl Johnson is currently the vice president, Global Hawk Programs. He is responsible for directing the current Air Force-contracted program, managing Northrop Grumman's support of the system's use by the Air Force in Operation Enduring Freedom, and overseeing the program's new business development efforts for the U.S. Navy and for international markets. Prior to his current position, Carl was the B-2 Deputy Program Manager within Northrop Grumman-Air Combat Systems.

(E-mail address: cjohnson@ryanaero.com)