

Developing Requirements for Technology-Driven Products

Louis S. Wheatcraft
Requirements Experts
(281)486-9481

louw@reqexperts.com
<http://www.reqexperts.com>

Copyright © 2005 by Compliance Automation. Published and used by INCOSE with permission. Presented at INCOSE 2005.

Abstract: New technology development is often a mainstay in product development, yet projects responsible for developing products based on new technology frequently are over budget, delivered late, poor in quality, and fail to meet customer and user needs and expectations. Two things that contribute to these problems are: 1) failing to define the product requirements up-front and 2) failing to adequately manage the maturation of the enabling technology(s) needed. This paper introduces several key approaches and processes that address these issues and the many challenges of developing and managing requirements for technology-driven products.

Requirement Completeness and Technology Maturity Issues

In June 2003, the United States General Accounting Office (GAO) issued a report ^[GAO 2003] to Congress concerning common problems and their effects on the US Department of Defense (DoD) satellite and related acquisitions. They found that the majority of the DoD satellite programs cost more than expected and took longer to develop and launch than planned. From a requirement perspective, they found the following common problems:

- Requirements for what the satellite needed to do and how well it must perform were not adequately defined at the beginning of programs.
- Often software needs were poorly understood at the beginning of a program. Not having a good set of requirements from the beginning made it more difficult for programs to ensure that they could match their requirements to their resources (money, time, and technology).
- Requirements were changed significantly once the program had already begun. Unresolved conflicts among users on requirements resulted in competing/conflicting requirements. This, in turn, resulted in changes having to be made throughout product development. The more requirements that were added or changed, the more the project's cost and schedule increased.
- Projects frequently tried to attempt to satisfy all requirements in a single step, regardless of the design challenge or the maturity of technologies.
- Projects often took a schedule-driven instead of a knowledge-driven approach to the acquisition process. Planning was frequently too optimistic (especially when new technologies were being proposed) and costs were not adequately known at the start of the program. Not enough time was spent defining requirements and often test schedules were compressed to meet target launch dates or testing was done concurrent with production. This resulted in having less time to fix problems that were uncovered during testing. This, in turn, resulted in cost and schedule increases due to the rework needed to fix problems late in the product development lifecycle.

- Technologies were not mature enough to be included in product development. This often caused cost and schedule increases due to the need to fix problems later in development.

In reviewing this list, many readers will find that these problems are not unique to DoD space programs. Others may recognize one or more of these problems in their own projects. For organizations involved in developing technology-driven products, these problems are the norm rather than the exception. This is true whether the organization is NASA, DoD, or industry. It is also true for both hardware and software development projects.

Technology “Push” vs. Requirements “Pull”

Other literature and technical publications identify another problem unique to technology driven product development: developing products based on a technology “push” vs. developing products based on a customer/user requirements “pull”. With technology “push”, products are developed because the technology makes the product possible, with the hope that a market for that product can be found. With requirements “pull”, products are developed based on proven customer/user needs, with enabling technologies developed in response to meeting those needs. Products developed using the requirements “pull” approach are much more likely to be successful because they already have a market waiting for them.

Developing a product using a technology “push” approach actually results in many of the problems identified in the GAO report. How does the technology “push” approach get into product development processes? Charles Nuese ^[Nuese 1995] recognized this problem pointing out that many of the high-tech businesses are managed by former engineers and scientists resulting in product development often being “pushed” by technology, rather than being “pulled” by the needs of the customer and users. These businesses focus on the funding needed to develop new technologies and capitalizing on these new technologies.

As pointed out by Michael Schrage ^[Schrage 2004] these high-tech business managers often set high expectations. Sometimes these expectations are met, but often they are not. Right or wrong, in today’s culture of high-tech products, this is the way products are often developed. Schrage observes, *“Looking at old issues of Technology Review or Wired, you’ll swiftly realize that the past isn’t prologue: it’s a turbulent world of heady speculations and unfilled promises. Those promises are indispensable elements of the innovation ecosystem. When artfully calibrated against actual progress, they keep markets salivating and investment – both financial and human capital – flowing.”*

Researchers and technology developers, as a whole, have a bad track record when it comes to developing products using their newly developed technologies. Schrage ^[Schrage 2004] makes the point that while doing basic research there are many questions that cannot be answered until (and when) the technology matures. He also makes the point that while inventors have great expectations and often make promises based on those expectations, they are very poor at managing those expectations. Schrage says: *“Innovators are either overly optimistic or unduly pessimistic. They haven’t a clue what new costs their innovations will impose on potential users. They have no credible way to assess what needs will be most important to these users three years hence.”*

Another frequent problem is that in the quest to get to market before the competition, these high-tech companies often integrate their technology into a product before it is mature enough or before enabling technologies or infrastructure are mature enough. A good example of this is the

use of fuel cells in automobiles. Fuel cells need a source of hydrogen. Providing an extensive infrastructure similar to existing gas stations is a formidable task. Unless a practical source of hydrogen is readily available, who is going to buy a vehicle whose sole source of power is from fuel cells? Unless a product is developed using a requirements “pull” approach that starts at the top with clearly defined customer and user requirements, that product is doomed to fail.

Unfortunately, the technology “push” mindset seems to predominate today’s high-tech world. Researchers, inventors, and technology adopters often develop a technology based on possibilities, promising investors a big return on investment. Researchers from academia and industry alike are developing technologies that promise substantial benefits. They often glorify the possibilities in order to get the funding needed to develop these technologies. But often no one has identified a true customer or user need for the technology. They don’t understand that customers and users don’t care as much about the technology as the capabilities and performance a new technology provides them via a given product. The reason people buy products is to solve a problem and to achieve the associated benefits provided by the product to solve that problem. Technology only enables the product to meet the buyer’s needs. Take for example “Bluetooth” and “Blackberry”: they are great technologies, but what the consumers care about is the increased accessibility and communications capability these technologies enable.

It’s a Question of Managing Risk

The technology “push” approach is a high-risk approach. We constantly hear about new technologies like nanotechnology, biotechnology, quantum computers, data mining, image recognition, artificial intelligence, fuel cells, and many more. When will these technologies be mature enough to be included in a marketable product? *“Clearly, transmitting raw hope into net profits requires investors and entrepreneurs to place extremely prescient and/or damn lucky bets on those rare companies that will actually convert cutting-edge research into commercially successful products.”* - Joe Chung ^[Chung 2004]

Investors are often faced with choosing between several competing technologies or researchers developing a given technology. Which one has the least risk and most promise? Can the technology be developed within reasonable schedules and affordable costs? Will the resulting product be affordable? Which technology should I invest in? What will be my return on investment? When will I realize a return on my investment? Having a technology “push” approach is like taking the “if we build it, they will come” approach popularized in the movie *Field of Dreams*, starring Kevin Costner. If we build it, will there really be a market for our product? If we build it, will anyone buy it? If we build it, will anyone use it? Developers have many of the same questions when accessing enabling technologies for use in their products.

Converting cutting-edge research into commercially successful products is the ultimate goal of researchers, inventors, entrepreneurs, and investors. Yet, as pointed out in the above discussion, the technology “push” approach is a high risk approach. To be successful, technology driven products need to be developed using the requirements “pull” approach. Louis Fried ^[Fried 1995] made a similar point concerning how to determine which information technologies should be adopted within an organization. He says that users need to take charge of the information technology needs for business process redesign. In doing this they are changing their approach from where technology is “introduced” to the organization to an approach where the users set the requirements for new technology – that is, from technology “push” to demand “pull.”

The issue of managing risk for the development of new products was also addressed in another GAO report ^[GAO 2001] dealing with best practices. The report states that leading commercial firms have adopted practices that are knowledge based. *“The most important knowledge point occurs at launch—the point at which the product developer makes a decision to commit (or invest) the resources necessary to develop a new product that will meet customer needs. Successful programs are launched only when a product developer is confident that it has the resources— technology, engineering, and production knowledge, along with sufficient time, and money—to develop a product the customer wants.”* A key point is that there must be a match between the requirements and the technology needed to meet those requirements before launching product development. GAO found that projects had significant problems when launched without this match.

To address the problems outlined in the GAO reports and the uncertainty associated with technology “push” vs. requirements “pull”, projects need to follow a process that identifies, analyzes, and takes the appropriate action on the risks associated with developing technology-driven products.

A Better Approach

I propose that there are better, more disciplined approaches to developing technology driven products. Nuese ^[Nuese 1995] defines two major goals of the project team: to define the “right product” and to design and develop the “product right”. A key approach to do this is to ensure the team has a clear vision of what the end product is supposed to do and how well it needs to do it. He says what is needed is a customer/user “pull model” that is driven by customer and user needs. The challenge is to find the necessary enabling technologies to build a product that meets these needs. The project team must clearly understand customer and user needs to be successful.

In her book, *Customer-Centered Products – Creating successful products through smart requirements management*, Ivy Hooks ^[Hooks 2001] presents an overall model for defining and managing requirements that applies to any type of product or process development project. Hooks focuses on three fundamental areas: defining product scope before writing requirements; training requirement writers to understand what a good requirement is and how to organize the resulting requirements into a complete, cohesive set; and actively managing requirements throughout the lifecycle of the product. She maintains that all projects have a common goal: *“Deliver a winning product – one that is needed, developed within budget, on-time, and with the desired quality.”* As we can tell from her book, Hooks advocates the requirements “pull” model.

Hooks’ approach also addresses the problems identified in the GAO reports concerning technology driven products. The project must allow time at the beginning to define the product scope and baseline a complete, correct, and consistent set of requirements. To make this happen a concurrent engineering approach must be used where key stakeholders from each of the product’s lifecycle stages are involved in scope and requirement definition. The project needs to define the problem it is to solve so they can identify a true “need” for a product. They can then define a clear set of goals and objectives that the product must realize in order to meet the need. Defining the need, goals, and objectives up front, will drive all subsequent product development activities. Drivers, constraints, and external interfaces must be defined so the product boundaries are clearly understood. Operational concepts need to be developed which include the views of the key stakeholders, address all product lifecycles, and cover both nominal and off-nominal conditions. Allowing the time necessary to develop the product scope results in a clear vision to guide the project team, knowledge needed to write requirements, and a means to stay on course.

A more detailed discussion on developing product scope is presented in references [Hooks 2001], [Wheatcraft 2001], and [Wheatcraft 2002].

I often have students in my requirement definition and management classes whose project's purpose is to develop a new technology. Many of these students indicate they don't understand how requirements apply to their project. They say they aren't building a product; they are developing a technology to be used in a product. It is clear that they are from the technology "push" crowd – "I will develop a technology that provides certain capabilities and performance and users can then build a product based on what I have developed." When asked when their technology will be mature enough to be used in a production environment, technology developers often state that they don't know, there are too many unknowns. When asked what performance is required by the potential users of their technology, they say, "It hasn't been defined. First we have to conduct experiments and tests to determine what can be done and then more tests to allow us to deem the technology mature enough for use in a product."

This is not the road that will lead to a successful project – one that delivers a winning product. The project team must first go through the scope definition activities outlined above. As part of this scope definition effort, the team will identify the core or enabling technologies needed and then invest in those that show promise in supplying the needed capabilities and meet the functionality and performance objectives with reasonable schedules and affordable costs. The team will define key figures of merit or key performance parameters each technology must meet to be viable. Drivers and constraints that each technology must operate within will be identified. For example, NASA is currently involved in identifying key technologies in support of the Nation's Vision for Space Exploration. Key figures of merit include safety, reliability, sustainability, affordability, extensibility/evolvability, effectiveness, and innovation. Trade studies will guide which technologies are selected to best meet the project's objectives.

This is a road that is much more likely to lead to a successful project. This is the road that system engineers take. This is a requirements "pull" process where technologies are developed in response to needed capabilities and performance of the product under development.

The Doctrine of Successive Refinement

In reality not all the final product requirements can be defined upfront. There are many uncertainties. In practice it is not as simple as just technology "push" and requirements "pull" - it is an iterative process of push and pull. That is why technology driven products must invest the time it takes, in the early phases of the project, to clearly define the scope, develop a plan to mature the technologies needed, and define a workable concept that will result in meeting the need, goals, and objectives defined for the product. For technology driven products, the first attempts at defining product scope will be a little fuzzy. We may start with key performance statements containing threshold values as well as objective values. The threshold value is the minimum acceptable – if it can't be met we need to find another technology or wait until the needed technologies are mature enough. The objective value is what we would really like to have, if possible, within the schedule and cost bounds of our project. In the end, the performance will most likely be somewhere between the threshold and objective values. As the enabling technologies mature and we know what is possible, within those bounds, we will be able to write more specific requirements for the final product. It may take multiple iterations of scope definition before we have a feasible concept. This approach follows the doctrine of successive refinement defined in the NASA Systems Engineering Handbook ^[NASA 1995] shown in Figure 1.

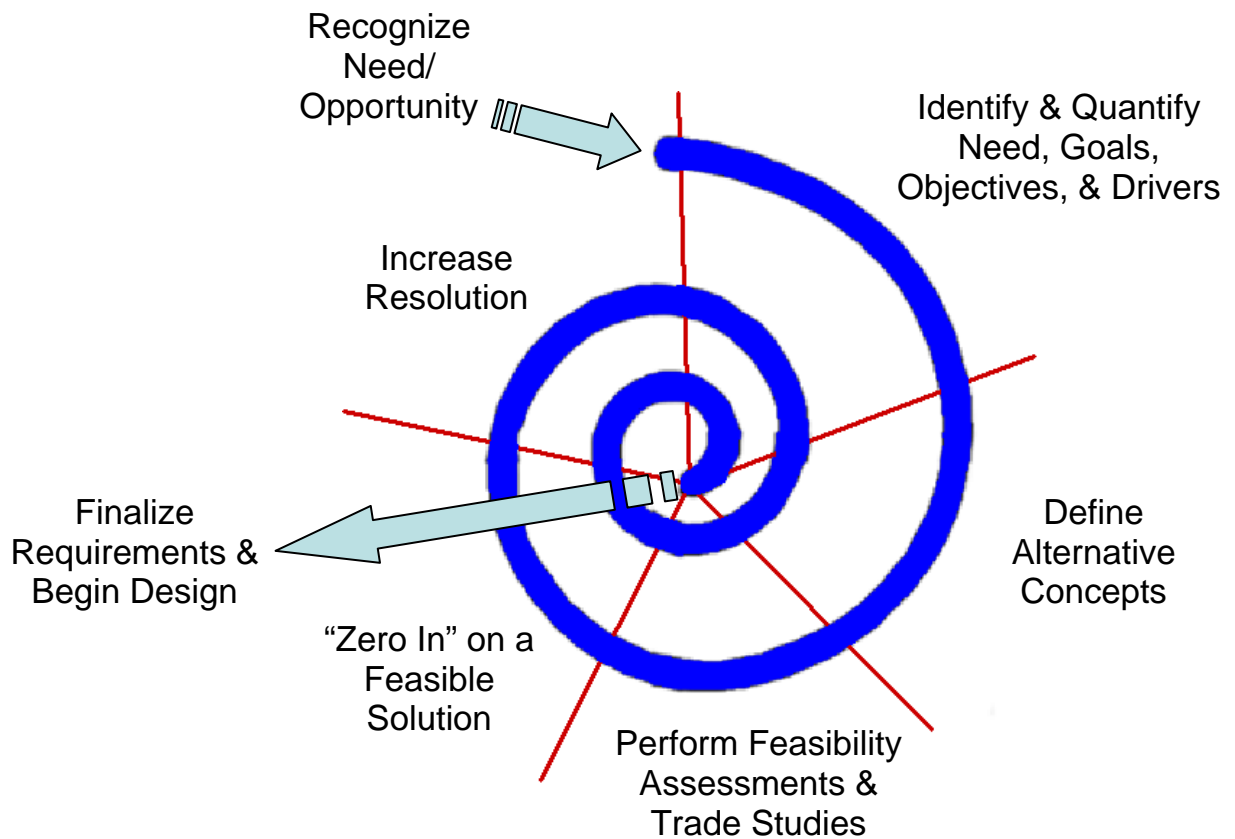


Figure 1: Doctrine of Successive Refinement

Using the Doctrine of Successive Refinement we start with a recognized need or opportunity; identify and quantify our need, goals, objectives, and drivers; define high-level alternative concepts; and perform feasibility assessments and trade studies. From the results of these assessments and studies we increase our knowledge, zeroing in on a feasible solution; increase the resolution by refining our goals, objectives, and drivers, then start the cycle over at the next level of detail each time zeroing in on the feasible alternatives and reducing the number of promising alternative concepts. We keep repeating this inward spiral until we have a single feasible concept that meets our need, goals, and objectives. The key point is that we start with an identified need which remains constant throughout the product development lifecycle. Our goals, objectives, and concepts start out at a somewhat ambiguous level, but are refined with increasing resolution and precision with each inward spiral.

The successive refinement of our concept begins in the scope definition phase of our product development lifecycle. Some lifecycle models refer to this as the formulation phase ^[NASA 2002], some the concept definition phase ^[ANSI/EIA 632, ISO IEC 15288], and others the pre-phase A/phase A stages ^[NASA 1995]. Once we have entered the preliminary design phase and have defined our system's architecture, the successive refinement approach is repeated for each of the subsystems.

The Doctrine of Successive Refinement directly addresses many of the problems identified in the GAO report ^[GAO 2003]: incomplete requirements, inconsistent and conflicting requirements, and not spending the time needed to define our requirements at the beginning of the project.

Spiral Development

Many of you may be thinking at this point that I have been describing Spiral Development – and you are probably correct. The Spiral Development model results in the same fundamental outcome. It is shown as starting in the middle and spiraling outward (refer to Figure 2). With each spiral our knowledge increases and so does the maturity of our concept and requirements to the point we are ready to build the final product. The Doctrine of Successive Refinement is shown as starting at the outside and then spiraling inward, zeroing in on the same end point.

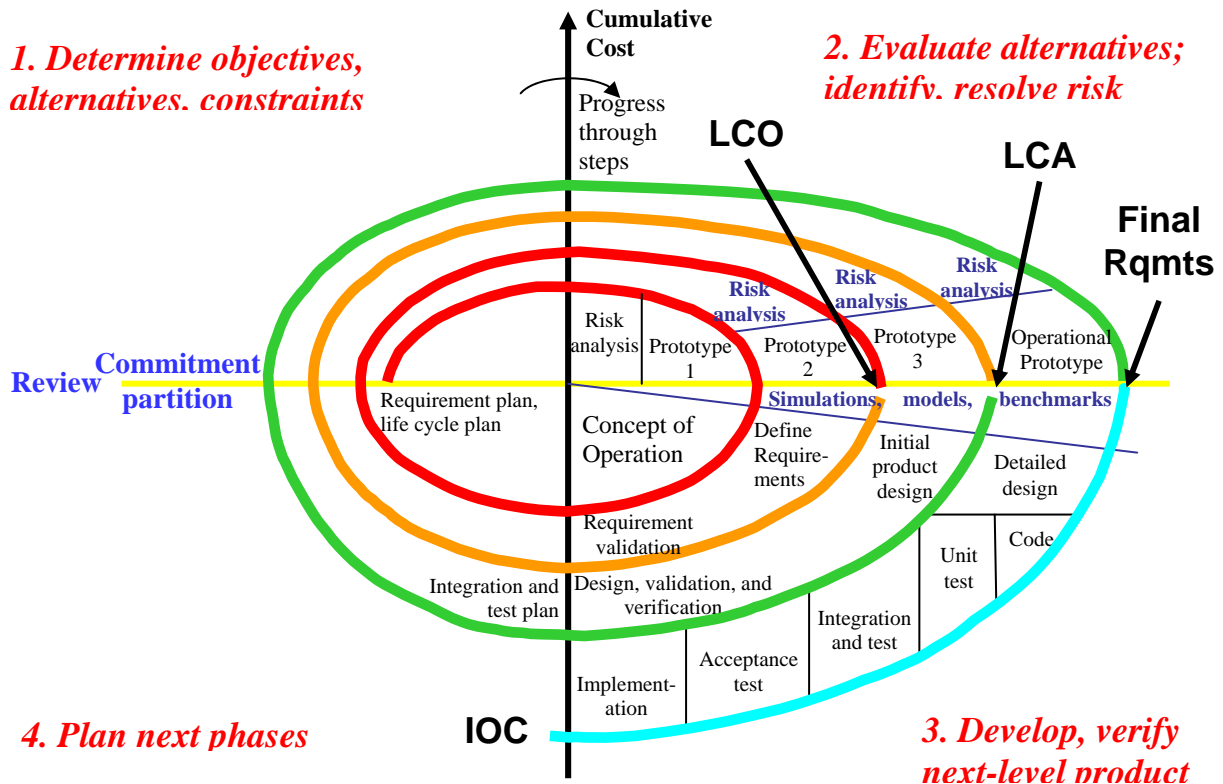


Figure 2: Spiral Development derived from [Boehm 2001]

Spiral Development was developed by Dr. Barry Boehm, primarily addressing software development projects. Spiral Development addresses projects where the requirements aren't clearly understood at the beginning, users often don't know what they want until they see a prototype, and based on what they learn, the user requirements often change. Dr. Boehm defines spiral development as^[Boehm 2001]: “The spiral development model is a **risk-driven process model generator** that is used to guide multi-stakeholder concurrent engineering of software-intensive systems. It has two main distinguishing features. One is a **cyclic** approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk. The other is a set of **anchor point milestones** [life-cycle objectives (LCO) – what should the system accomplish?, life-cycle architecture (LCA)- what is the structure of the system?, and initial operational capability (IOC)- first release] that are key decision points for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions.”

Prototypes are used to support the risk analysis process that results in a better understanding of the user's requirements and the challenges in meeting those requirements. Once a prototype

has been validated it will often become the simulation, model, or benchmark for the next step in the spiral but the prototype was not intended to be delivered to the customer as a product with usable operational capability.

Each trip around the spiral starts with determining scope as well as a set of requirements focused on the foregoing scope and ends with a prototype and finally the end product. With each spiral, the requirements mature from operational to development to design as the objectives, alternatives, and constraints are better known.

Each spiral considers these main elements: 1) critical-stakeholder objectives and constraints, 2) product and process alternatives, 3) risk identification and resolution 4) stakeholder review, and 5) commitment to proceed. These elements are very similar to those included with each spiral shown in Figure 1, the Doctrine of Successive Refinement. There is enough flexibility in this model so that you may go around the first part of the spiral several times before going on to the next level – it depends on the results of the risk analysis and the decisions made along the commitment partition shown in the diagram.

Boehm's inclusion of anchor point milestones is especially important for technology driven projects. Each milestone is a stakeholder commitment point. The purpose of the LCO review is to make sure that there is at least one feasible solution to meet the stated need, goals, and objectives of the project. The purpose of the LCA review is to align the stakeholders to a common vision and agree on the scope of the project. Boehm states: *"The LCA milestone is particularly important, as its pass/fail criteria enables stakeholders to hold up projects attempting to proceed into evolutionary or incremental development without a life cycle architecture."*^[Boehm 2001], Risk analysis is a key activity for each spiral and the project must have a risk management plan in place. At each milestone the project risks are reviewed and the resolution of each risk is agreed to by the stakeholders.

To better understand how a project can use the anchor point milestones, Boehm provides the following Stud Poker Analogy^[Boehm 2001]: *"A valuable aspect of the spiral model is its ability to support incremental commitment of corporate resources rather than requiring a large outlay of resources to the project before its success prospects are well understood. Funding a spiral development can thus be likened to the game of stud poker. In that game, you put a couple of chips in the pot and receive two cards, one hidden and one exposed. If your cards don't promise a winning outcome, you can drop out without a great loss. This corresponds to canceling a project at or before LCO. If your two cards are both aces, you will probably bet on your prospects aggressively (or less so if you see aces among other players' exposed cards). Dropping out of the second or third round of betting corresponds to canceling at or before LCA. In any case, based on information available, you can decide during each round whether it's worth putting more chips in the pot to buy more information or whether it's better not to pursue this particular deal or project."*

The Concept of Technology Readiness Levels

A key part of managing technology development is understanding the key challenges and associated risk to maturing the technology. How will the risk be mitigated and challenges met? Is there a plan for doing so? For each technology does the principle investigator have a clearly defined problem statement, need, goals, and objectives? Do they have a clearly defined strategy for maturing their technology? Do they have a project management plan? The answers to these questions will help us understand and manage the maturation of the enabling technologies and associated risk.

This problem has been addressed by both NASA and DoD. In April 1995, John Mankins of NASA's Advanced Concepts Office in the Office of Space Access and Technology wrote a white paper that discussed the concept of Technology Readiness Levels (TRLs) ^[Mankins 1995]. TRLs support assessment of the maturity and therefore the implementation readiness of a technology allowing a consistent comparison between different competing technologies as well as different types of technologies.

TRL 1: Basic principles observed and reported.	Lowest level of technology readiness. Scientific research begins to be translated into applied research and development. Examples include paper studies of a technology's basic properties.
TRL 2: Technology concept and/or application formulated.	Invention begins. Once basic principles are observed, practical applications can be invented. Applications are speculative and there may be no proof or detailed analysis to support assumptions. Examples are limited to analytic studies.
TRL 3: Analytical and experimental critical function and/or characteristic proof-of-concept.	Active research and development is initiated. This includes analytical studies and laboratory studies to physically validate analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.
TRL 4: Component and/or breadboard validation in laboratory environment.	Basic technological components are integrated to establish that the pieces will work together. This is relatively "low fidelity" compared to the eventual system. Examples include integration of "ad hoc" hardware in the laboratory.
TRL 5: Component and/or breadboard validation in relevant environment.	Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so that the technology can be tested in a simulated environment. Examples include "high fidelity" laboratory integration of components.
TRL 6: System/subsystem model or prototype demonstration in a relevant end-to-end environment.	Representative model or prototype system, which is well beyond that of TRL 5, is tested in a relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high fidelity laboratory environment or in simulated operational environment.
TRL 7: System prototype demonstration in an operational environment.	Prototype near, or at, planned operational system. Represents a major increase in maturity from TRL 6, requiring demonstration of an actual system prototype in an operational environment. Examples include testing the prototype in a test bed aircraft or in space.
TRL 8: Actual system completed and qualified through test and demonstration.	Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of system development. Examples include development test and evaluation of the system in its intended weapon system to determine if it meets design specifications.
TRL 9: Actual system proven through successful mission operations.	Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation. Examples include using the system under operational mission conditions.

Table 1: Technology Readiness Levels – DoD ^[Nolte 2003]

The NASA TRL model was developed for space systems. In June 2001, US DoD adopted the more generic TRL model ^[Nolte 2003] shown in Table 1 which is now mandated for all major acquisition programs. This model is now one of the cornerstones in the evaluation and selection of core technologies needed to realize the Nation's Vision for Space Exploration. The UK Ministry of Defense proposed the adoption of TRLs in February 2002 ^[UK MoD 2003]. Carnegie Mellon University has developed a software-oriented set of TRL descriptions for use in assessing practice-based technologies ^[Carnegie Mellon 2003]. DoD also has developed TRL definitions ^[Nolte 2003] for both hardware and software as well as have developed a TRL calculator ^[Nolte 2003] to determine the maturity level of a given technology.

The TRL process model includes 9 levels that address five stages of technology maturation:

1. ‘basic’ research in new technologies and concepts (targeting identified goals and objectives, but not necessarily specific system requirements),
2. focused technology development addressing specific technologies for one or more potential system applications,
3. technology development and demonstration for each specific application before the beginning of full system development of that application,
4. system development (through first unit fabrication), and
5. system ‘launch’ and operations.

TRLs 1-3 are considered the concept stage moving from basic to applied research where research to prove feasibility is conducted. TRLs 4-5 are the validation stage where the technologies are developed and demonstrated in the laboratory. TRLs 6-7 are the prototype demonstration phase where the technologies move out of the lab to relevant and operational environments ending with an actual system that is “flight qualified”. TRLs 8-9 are the testing, evaluation, and application phase where final system validation takes place with TRL 9 the final stage using the “flight proven” system.

In the TRL model, technology maturity increases while risk decreases as we progress through each level. While it may not always be necessary to develop a technology through every level, the risk associated with skipping levels should be assessed against the cost of progressing through each level. The TRL model is designed to be used with individual technologies, those enabling technologies that have been identified in the concept stage as showing promise in meeting our system’s need, goals, and objectives.

Technologies at TRL 1-3 often will fall into the technology “push” classification if basic research is being conducted for the purpose of gaining more complete knowledge without a specific application in mind. This is the nature of basic research. However, this paper addresses applied research in which the purpose is to develop a technology to enable a specific, recognized need. Applied research is directed at advancing the state-of-the-art so that a new, innovative technology is available – enabling future users to conduct a mission or provide a capability that is not possible given the current state-of-the-art. At this point the operational users and other stakeholders may not be aware of this research or the fact that this research will provide new capabilities and performance that enable them to conduct the new mission. In this situation, it may seem that the research is being conducted and the technologies defined without any requirements. This is not the case. For applied research, TRL 1-3 truly fall into the requirements “pull” classification. The researchers know the current state-of-the-art. They know what capabilities and performance are possible today. In order to justify their research and get funding, they need to have defined the problem they are trying to solve, established a clear and recognized need for why they are conducting the research, and established goals and objectives which include key-performance parameters (both a minimum threshold and a goal) that define where they intend the research to take them. By combining this information with the concept of TRLs, a clear set of requirements can be defined.

One key management function is assessing the progression of the technologies of interest through each of the TRLs. The key question asked is, “When will the technology be mature enough to consider?” In the acquisition world, technologies are not normally considered viable candidates until they reach TRL 3. Technology candidates that are at TRL 3 then go through the

assessment stage in TRLs 4 and 5 and a functional prototype demonstration in a relevant environment at TRL 6. It is highly recommended that a given technology be at TRL 6 before being chosen for inclusion in the final product's pre-production development phase. The technology level should be at TRL 7 by the preliminary design review and TRL 8 at the critical design review of the final product.

A principal finding in a 1999 GAO report ^[GAO 1999] on managing technology was that maturing technologies at program start is an important determinant of success. They stated *“Programs with key technologies at readiness levels 6 to 8 at the time of program launch met or were meeting cost, schedule, and performance requirements.”* The report also addressed the importance of recognizing when a gap occurs between a technology's maturity and the intended product's requirements. They found: *“For technologies that were successfully incorporated into a product, the gap was recognized and closed before product development began, improving the chances for successful cost and schedule outcomes. The closing of the gap was a managed result. It is a rare program that can proceed with a gap between product requirements and the maturity of key technologies and still be delivered on time and within costs.”*

The purpose of some projects is to develop an interim product that can be used to demonstrate technologies in a relevant environment. Even with these types of projects, projects can use TRLs to manage risk by ensuring the technologies are mature enough to meet the objectives of the technology demonstration activity.

Combining the Doctrine of Successive Refinement and Spiral Development with the Concept of Technology Readiness Levels

Combining the Doctrine of Successive Refinement and Spiral Development with TRLs provides product developers with a very powerful tool to manage the maturation of enabling technologies and develop requirements for their product. Taking a product from one TRL to the next forms the basis of a one or more paths around the spiral. The original need doesn't change, but as you progress from one spiral to another you will have more refined goals and objectives including the goal to get to the next level. For example, if your technology is at TRL 5 and your goal is to progress to TRL 6, then one of your goals will be to develop a new model, prototype, or flight demonstration unit and another goal will be to successfully test your new system in a relevant environment. Based on the progress made getting to TRL 5, you will have a better idea of the capabilities and performance that are feasible and can refine your overall objectives for reaching the key performance parameters. You will also have a better understanding of the drivers, constraints, and risk associated with getting to the next level. You will be able to refine your concepts for a given technology and for the overall product under development. Once your revised system has been developed, you can perform the testing needed to get to the next level. With each spiral you will be better able to define the requirements for the final product.

In order to reduce risk, projects should not “put all their eggs in one basket”, i.e., if practical, they should provide themselves with alternative technologies in case the leading candidate runs into development problems. This means projects need to consider funding more than one candidate technology for a given area. This leads to a set of candidate technologies being developed in parallel which may be at the same or different TRLs. Product developers will then closely monitor the progress of each technology under development, assessing how well it is progressing and its ability to achieve the key performance parameters needed for the final product. If a given technology isn't able to meet your objectives and requirements, or is not

progressing to the next TRL needed to support your product development schedule, that alternative can be dropped from consideration and you can switch to your alternate technology.

Combining Spiral Development with TRLs addresses the technology maturation problems and the problem of requirements not being completely defined at the beginning of the project identified by the GAO as well as the technology “push” vs. the requirement “pull” issue. Combining these methods also provides a way to manage risk for investors and product developers. This approach provides a project with the option to invest in several competing technologies or companies involved in developing a technology and then continue investment based on how well each is maturing using the TRLs as a measure.

Some last thoughts on Spiral Development and TRLs. Originally, Boehm assumed that the technology being used was mature ^[Boehm 2000] before entering Spiral Development (at least TRL 6) and that the prototypes were not intended to provide an operational capability. However, when applying Spiral Development to maturing technologies, the nature of the prototypes being developed is different. Prototypes are developed in the form of breadboards, brassboards, and operational demonstration units. As the technology progresses through the TRL levels, more and more operational capability is being added. For TRLs 4-5, the breadboards and brassboards are built to demonstrate key capabilities and performance. For TRL 6 the demonstration unit is a higher fidelity prototype with all the capabilities and with the performance close to what it needs to be for the final product. TRL 7 results in the development of a mature operational prototype – with the capabilities, functionality, performance, and quality required for the final product.

Another significant attribute of the spiral model is its focus on risk reduction. Boehm ^[Boehm 2001] defines risk as: “*situations or possible events that can cause a project to fail to meet its goals.*” As shown earlier, the cyclic nature of the spiral model is similar to the Doctrine of Successive Refinement. Rather than develop the completed product in one step, multiple cycles are performed with the project taking steps needed to resolve risk within each cycle before proceeding to the next cycle. This focus on risk reduction is the same outcome as envisioned for adopting the concept of TRLs.

When system engineers fully understand the underlying goals of the Doctrine of Successive Refinement, Spiral Development, and TRLs, their reactions are generally, “That is how we have always developed our products – that is just good system engineering.” And they are correct.

Parting Thoughts

The requirements, schedule, and technology maturity issues presented by the GAO as well as the technology “push” vs. requirement “pull” issue are common to many projects, especially technology driven products. Projects that adopt the requirement “pull” approach are requirement driven projects. By being a requirement driven project and adopting good system engineering approaches outlined in this paper, they are better able to successfully develop and manage their system requirements. Combining the Doctrine of Successive Refinement and Spiral Development with the concept of Technology Readiness Levels, both product developers and investors will be better equipped to manage the risk and challenges associated with developing technology driven products.

References:

- Acquisition Management System, United Kingdom, MoD, paper *AMS Guidance on Technology Readiness Levels (TRLs)*, FGB/36/10, 4 Feb 2003.
<http://www.ams.mod.uk/ams/content/docs/trlguide.doc>
- ANSI/EIA 632, Standard, *Process for Engineering a System*, January 1999
- Boehm, B., *Spiral Development: Experience, Principles, and Refinements*, CMU/SEI-00-SR-008, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.
- Boehm, B. and Hansen, W., *The Spiral Model as a Tool for Evolutionary Acquisition*, CrossTalk - The Journal of Defense Software Engineering, May 2001: 4-11
- Carnegie Mellon University, *TRL Corollaries for Practice-Based Technologies*, 2003.
<http://www.sei.cmu.edu/ttp/presentations/trl-corollaries/trl-corollaries.pdf>
- Chung, Joe; "Panning Out – An angel investor's view of emerging companies"; *Technology Review*, page 37; October 2004
- Fried, Louis, *Managing Information Technology in Turbulent Times*, John Wiley, & Sons, 1995
- GAO Report GAO/NSIAD-99-162 *Best Practices, Better Management of Technology Development Can Improve Weapon System Outcomes*, United States General Accounting Office, July 1999 <http://www.gao.gov/archive/1999/ns991620.pdf>
- GAO Report GAO-01-288 *Best Practices, Better Matching of Needs and Resources Will Lead to Better Weapon System Outcomes*, United States General Accounting Office, March 8, 2001
<http://www.gao.gov/new.items/d01288.pdf>
- GAO Report GAO-03-825R *Satellite Acquisition Programs, Military Space Operations: Common Problems and Their Effects on Satellite and Related Acquisitions*, United States General Accounting Office, June 2, 2003 <http://www.gao.gov/new.items/d03825r.pdf>
- Hooks, Ivy F. & Farry, Kristin A., *Customer-Centered Products: Creating Successful Products Through Smart Requirements Management*; AMACOM Books, NY, NY, 2001.
- ISO/IEC 15288 *System Engineering-System Life Cycle Processes*, October 2002
- Mankins, John C., *Technology Readiness Levels*, Advanced Concept Office, Office of Space Access and Technology, NASA, April 6, 1995.
<http://etdo.msfc.nasa.gov/proposal/docs/TRL.doc>
- NASA *Program and Project Management Processes and Requirements*, NPG 7120.5B, November 2002
- NASA *System Engineering Handbook*, SP6105, June 1995.
http://ldcm.nasa.gov/library/Systems_Engineering_Handbook.pdf
- Nolte, William; Kennedy, Brian; & Dziegiel, Jr., Roger; presentation "Technology Readiness Level Calculator", NDIA Systems Engineering Conference, October 20, 2003
<http://www.dtic.mil/ndia/2003systems/nolte.ppt#1>
- Nuese, Charles J.; *Building the Right Things Right*, AMACOM Books, NY, NY, 1995
- Schrage, Michael; "Great Expectations – Making Good Ideas Matter"; *Technology Review*, page 21; October 2004.
- Wheatcraft, Louis S. & Hooks, Ivy F., *Scope Magic*, 2001.
<http://www.complianceautomation.com>
- Wheatcraft, Louis S., *The Importance Of Scope Definition Prior to Developing Space System Requirements*. INCOSE *INSIGHT*, Vol. 4 Issue 4, January 2002.
<http://www.complianceautomation.com>

About the Author

Lou Wheatcraft has over 40 years experience in the aerospace industry, including 22 years in the United States Air Force. Over the last 10 years, Lou has worked for Compliance Automation (dba Requirements Experts), where he has conducted over 140 seminars on requirement development and management for NASA, Department of Defense (DoD), and industry. Lou has had articles published in the International Council of Systems Engineering (INCOSE) INSIGHT magazine and in DoD's magazine, CrossTalk.

Lou has made presentations at NASA's PM Challenge, INCOSE's International Symposium, and at the local Project Management Institute (PMI) and INCOSE Chapter Meetings. Lou has a BS degree in Electrical Engineering, an MA degree in Computer Information Systems, an MS degree in Environmental Management, and has completed the course work for an MS degree in Studies of the Future.

Lou is a member of INCOSE, co-chair of the INCOSE Requirements Working Group, a member of PMI, the Software Engineering Institute, the World Futures Society, and the National Honor Society of Pi Alpha Alpha. Lou is the recipient of NASA's Silver Snoopy Award and Public Service Medal and was nominated for the Rotary Stellar Award for his significant contributions to the Nation's Space Program.

