# APL System Development Process

*Elinor Fong, David Y. Kusnierkiewicz, Deborah P. Mendat, and Peter D. Tennyson*

*As part of the development of the APL Quality Management System, APL has developed the System Development Process. The purpose of the APL System Development Process was to establish a minimum set of requirements that complies with the International Organization for Standardization standard 9001 (ISO 9001), is consistent with the sometimes sensitive nature of our work, captures our existing best practices, and retains and enhances the existing efficiencies and values of APL. The process was developed so that it could be tailored to accommodate the broad range of system development efforts across APL. This article describes the activities associated with the APL System Development Process and also discusses its relationship to two preexisting system development efforts: the Modular Prosthetic Limb, one of several systems developed by the Revolutionizing Prosthetics program, and the Aegis Ballistic Missile Defense Test and Evaluation Target Instrumentation Project.*

## INTRODUCTION

The Laboratory has a well-earned reputation for excellence in systems engineering. Many staff members have authored or contributed to systems engineering textbooks[1–3] and teach in systems engineering programs. Our tasks range from small, quick-turnaround systems to very large, complex systems (e.g., Cooperative Engagement Capability, New Horizons) integrating components and assets from multiple sources in projects that take years to complete. However, until now, our systems engineering process has never been codified.

With the decision for APL to become certified to International Organization for Standardization (ISO) standard 9001 (ISO 9001) and develop the APL Quality Management System (QMS), it became necessary to formally define and document our systems engineering methods. Because many of our practices were already well developed, this was largely a task of documenting our best practices as they have evolved over the years. However, ISO 9001 also imposes certain minimum requirements. Although most of these requirements were already being executed, ISO certification also requires that we provide objective evidence, i.e., documentation that demonstrates our adherence to these principles. Our previous practices did not always provide that documentation at the level of detail required for ISO certification.

The goal of ISO 9001 is to ensure consistent quality in the work of the organization. However, the experience of organizations that have adopted ISO 9001 is that after the initial learning curve, more efficient operations also result. Increased efficiency comes from the critical examination that processes receive during their initial definition and subsequent continual improvement. This scrutiny identifies unnecessary activities and enables streamlining. Interfaces between QMS processes are as important as interfaces in any system and are subject to the same refinement. Most important, all of this must also preserve the APL culture that has been the basis of our success. Perfection is not expected, and is in fact unlikely, at initial rollout. Feedback from the staff executing the procedures and continual improvement allow processes and interfaces to be tuned over time.

The Laboratory management determined that we have three main products: systems, studies and analyses, and documents and presentations. Systems products cover an extremely broad spectrum of the work that we do. Systems may be funded from internal or external (sponsor-provided) funds; the system may be “delivered in place” or formally fielded. A system may be a prototype that is evaluated in the field by APL personnel; an operational system for which we also provide training, logistics support, and maintenance; or a spacecraft that is launched into space to a distant planet and operated from the APL campus. Examples of other systems products include simulators, systems developed internally for proof-of-principle, models and simulations, and testing facilities. Physically, a system may consist of only a single circuit board or software module, or it may be an entire space mission with flight and ground system segments.

The challenge was to define a system development process that encompasses the broad range of work that we do, establishes a set of minimum requirements that complies with ISO 9001, is consistent with the sometimes sensitive nature of our work, captures our existing best practices, and retains and enhances the existing efficiencies and value of APL.

This article will describe the APL System Development Process and will also discuss its relationship to two preexisting system development efforts: the Modular Prosthetic Limb (MPL), one of several systems developed by the Revolutionizing Prosthetics 2009 (RP2009) program, and the Aegis Ballistic Missile Defense (BMD) Test and Evaluation (T&E) Target Instrumentation Project (TIP). The RP2009 program began before establishment of the APL System Development Process. However, it followed systems engineering practices as required by the Food and Drug Administration for design of medical devices, and these practices align with the APL System Development Process. The goal of RP2009 was to develop an advanced upper-extremity prosthesis that will restore full motor and sensory capability to patients who are upper-extremity amputees. In 2009, the program completed a prototype of the MPL that is ready for system validation within clinical trials in the next phase of the program. The Aegis TIP develops ballistic missile target-based sensor instrumentation payloads and began system development in 2003. These target-based payloads are used during Aegis BMD Flight Test Missions to collect unique mission-related phenomena. The TIP is an ongoing system development effort; it also has been following good systems engineering practices based on the long heritage of APL projects developing launch vehicle-based instrumentation. As a result of the introduction of the Laboratory’s System Development Process, the TIP program has made some minor changes to its process documentation to better align with the APL QMS, primarily the extraction of standalone documents from its Performance Assurance Implementation Plan (PAIP). In each of the following sections, all of which describe a particular system development process activity, a description of how the activity was or is performed for the MPL or the TIP is included.

## PROCESS OVERVIEW

The System Development Process was defined to be in compliance with the following broad ISO 9001 requirements for product realization:

- Document and control changes to requirements
- Review requirements
- Document and control changes to design
- Review the design
- Verify and validate the system

ISO also requires the organization to be able to prove that system developments follow the defined processes. Therefore, the process also defines documentation requirements associated with these steps.

Figure 1 illustrates the defined System Development Process. The process activities are presented sequentially for simplicity. However, not all system developments follow such a linear progression. Spiral or agile development methodologies may involve multiple iterations between several of the steps in the process or even iterations through the entire process. For instance, when a system is being developed for purposes of discovery, prototype evaluation may result in revisions to requirements, initiating another cycle through design, fabrication, and verification.

Some of our system developments may not have fully defined requirements from the sponsor; in addition,

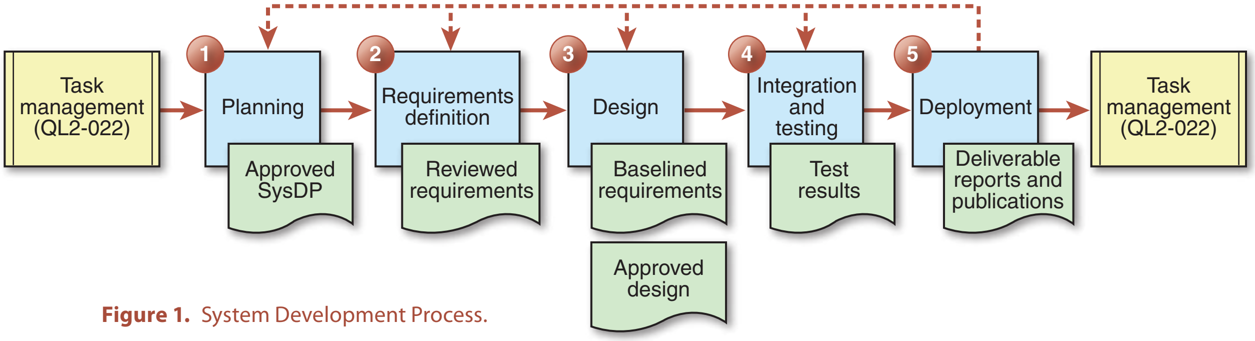


**Figure 1.** System Development Process.

a critical need may necessitate rapid turnaround. The System Development Process is intended to accommodate these realities. All of the review and verification activities do not need to be performed every time through the cycle. Rather, review and verification of only the final design is required, and may even occur after delivery. The intent is to minimize quality escapes and thereby enhance our performance and reputation.

In order to accommodate quick-turnaround projects or rapid prototyping, most of the system developments that are less than 3 months in duration and employ fewer than six full-time-equivalent staff are exempt from the System Development Process.

The process has two tiers, which differ mainly in the level of formality required of documentation and reviews. The basic ISO requirements must be satisfied for both low- and high-tier development efforts. For example, requirements must be documented for both tiers. For a low-tier task, requirements may be documented by using an Excel worksheet or a Word document maintained by the lead systems engineer; a high-tier task may necessitate use of a requirements database/management tool such as the Dynamic Object Oriented Requirements System (DOORS). The design review for a low-tier task may be conducted as an informal peer review, whereas a high-tier task would require a more formal Critical Design Review (CDR), with a published presentation package. Some high-tier tasks might require several successive design reviews as the development progresses from a preliminary to a detailed design change. The System Development Process is intended to be flexible in this regard, providing guidance and defining minimum requirements but leaving these decisions to the lead systems engineer, with the concurrence of the project or program manager (PM) and documentation in the System Development Plan (SysDP).

The required ISO activities are partitioned into the process activities shown in Table 1.

The System Development Process is coupled closely with the Software Development Process and the Hardware Development Process. Traditional systems engineering activities determine the functional and physical partitioning of the system, out of which system components (both hardware and software) are identified and defined. The individual components are developed (or fabricated) under the corresponding development processes. Components are delivered back to the system process for system-level integration after component-level verification is complete.

**Table 1.** Required System Development Process activities.

**Planning**

- Determine system components
- Determine tier
- Develop and maintain a SysDP

**Requirements definition**

- Define, document, and manage requirements
- Review system requirements

**Design**

- Establish requirements baseline and control changes
- Develop and review detailed design
- Establish design baseline and control design changes
- Plan for integration and testing

**Integration and testing**

- Review hardware and software verification results
- Integrate the system
- Conduct system testing
- Review system testing results
- Plan for deployment

**Deployment**

- Review system readiness for deployment
- Deliver the system

## PLANNING

The first major activity is the planning for the System Development Process. Typically, preliminary planning will occur during task proposal preparation and negotiation. This activity is the detailed planning that ensures that the appropriate steps are followed during the development of the system. This activity is usually initiated by the PM by a request for a SysDP. Before the

SysDP can be written, the system composition and the tier level must be determined. If the system is composed of both hardware and software, or the decision is made to proceed with the System Development Process for a hardware-only or a software-only system, the next step is to determine the tier level for the system.

The tier of the system defines the level of formality for the required activities of the system development. All internally funded system development efforts are low tier unless the Business Area Executive or PM directs otherwise. The tier determination for system developments funded by an external sponsor is based on the evaluation of the four attributes: Criticality (C), Use/Longevity (U), Effort/Schedule (E), and Size (S), also known as CUES. (The CUES criteria were adopted from our existing software engineering practices; this provides consistency among related development processes.) An externally funded system will be categorized as high tier if the system meets the criterion for criticality or if it meets the criteria for use, effort, and size. The CUES evaluation of the system and the resulting execution tier (low or high) is reflected in the SysDP. Based on these requirements, the MPL would have been considered a high-tier system, because C, U, E, and S would all have been "yes." The TIP is a low-tier system. Although TIP provides valuable data, it is not a mandatory asset and its performance does not impact the success or failure of the Flight Test Mission, so C is "no," as are U, E, and S.

The lead systems engineer, in collaboration with the PM and the lead software and hardware engineers, prepares the SysDP. It is a reference throughout the development process for the PM and the development team and may change as the system matures. The SysDP describes how the system will be developed to meet its requirements. The SysDP contains an overview of the system: the how, what, why, and when. It states why the system is being built, its intended use, who are the intended users, and the sponsor's objectives. The development methodology is also described in the SysDP. It is recognized that the development methodology or framework will vary depending on the specific system and its objective. For instance, if the system requirements are well understood and will likely be stable or unchanging, a waterfall or linear methodology may be chosen. However, if the system is being developed for explorative purposes, a spiral or iterative process may be more appropriate.

Descriptions of how the requirements of the system will be documented and managed as well as how the requirements flow down into the system elements (i.e., subsystems or components) are also included in the SysDP. The configuration-management and change-tracking methods for the requirements and design are specified. The System Development Process does not require that the requirements or design be in a particular form or use a particular tool; such decisions are left to the discretion of the lead systems engineer. The intent is to allow flexibility in the detailed execution of the process. The reviews that will be held are identified in the SysDP. A requirements review, design review, and a review of the verification and validation results are required for every system. However, the formality with which these reviews are conducted depends on the tier. Other reviews, such as a conceptual and/or preliminary design review (PDR), may be conducted. The SysDP states which reviews will be held, their purpose, and when they will occur. Some of the items in the SysDP, such as verification and validation plans, may not be fully known during this planning phase. It is not expected that these plans be developed during this phase of the system development, but rather that they be identified and described in the SysDP.

## RP2009 Implementation

RP2009 created a Systems Engineering Management Plan that provided all of the content that would now be included in the SysDP. The development team for the MPL was geographically distributed, and it was imperative that this team have access to and could readily share information. Part of the planning process was to establish means to share information. Two elements were used extensively:

1. Microsoft Outlook Public Folders were used as an e-mail archive. By establishing direct e-mail addresses for many of these folders, communications could be archived by simply adding a "cc:" to the appropriate folder to e-mail communications relating to design decisions and issues.
2. A Microsoft SharePoint site was established in the portion of the APL network that is accessible to external partners (called the DMZ, or "DeMilitarized Zone"). RP2009 was one of the initial programs in the DMZ.

SharePoint served as the primary shared design and development repository for both APL and external team members. Cisco MeetingPlace was used for team meetings, for most peer reviews, and for various other meetings, including program-level presentations to the sponsor. Most meetings were recorded; the recordings were downloaded and posted to SharePoint for the benefit of those who may have been absent. These recordings are part of the MPL Design History File.

## TIP Implementation

The TIP program describes its development approach in a PAIP rather than in a SysDP. The PAIP also includes additional information, such as lessons learned. The PAIP is a configuration-controlled, living document that is used to communicate to the team members and management the expectations and standards for those working on the project.

## REQUIREMENTS

The next major activity in the development process is defining the system requirements. The requirements are defined in a hierarchical manner. The top-level requirements directly address the sponsor's needs and system constraints. These top-level requirements then flow down to the successive levels. Depending on the scope and complexity of the system, the requirements may be completely defined at this level. For large, complex systems with many components, requirements may need to be defined at the subsystem and component (box) levels. The lower-level requirements may not be known at this phase and will likely be defined during a later phase. Requirements also flow down to hardware and software to support their development in accordance with the Hardware Development Process and Software Development Process. Requirements must be verifiable; the expected method of verification is identified at the time of requirements definition. The four verification methods for this process are testing, demonstration, analysis, or inspection. One or more of these methods may be needed to verify a specific requirement.

An important step of the requirements definition is the review of the requirements to ensure that the system goals, objectives, and constraints have been decomposed into a set of requirements that are suitable for developing design and operational performance specifications. For certain system development methods, the requirements definition may be an iterative process; it is not expected that each of these iterations undergo a review, but final requirements must be reviewed. Whenever possible, inviting independent reviewers is recommended, because they bring a different perspective and broaden the experience base of the review. The level of formality of the review will depend on the tier of the system. Regardless of the level of formality, the requirements are reviewed to ensure that they are complete, unambiguous, and logically self-consistent, and that they flow from the system objectives and constraints.

The System Development Process allows flexibility in how the requirements are documented and managed. For small, simple systems, the requirements may be in an Excel spreadsheet and managed by version numbers assigned by the lead systems engineer. However, for larger, more complex systems, commercial database software, such as DOORS or GForge, may be used to document and manage the requirements.

### RP2009 Implementation

Some requirements for the MPL were generalized, such as "complete transmission and accurate interpretation of afferent (sensory) and efferent (motor control) signals between the nervous system (central and/or peripheral) and the prosthesis" and "perceived by the patient as natural with respect to function, weight, durability, and comfort." Other requirements were very specific, such as ranges of motion and strengths of the various components and joints, operational speeds, and touch discrimination. It was necessary to analyze and allocate these requirements among the various MPL components, which are illustrated in Fig. 2.

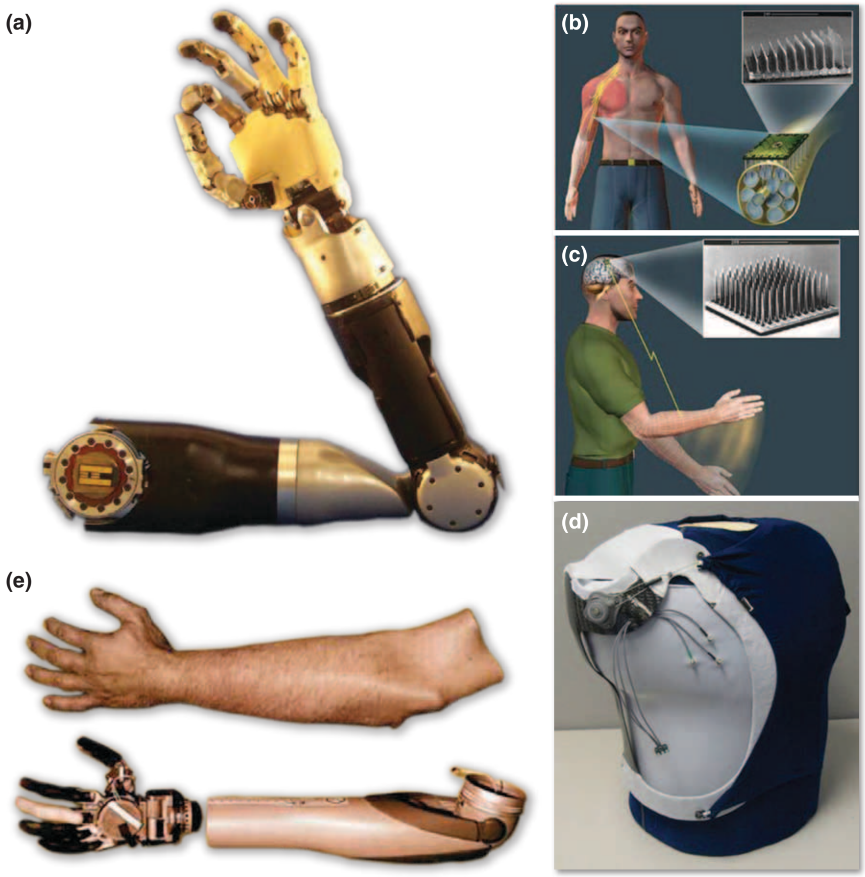
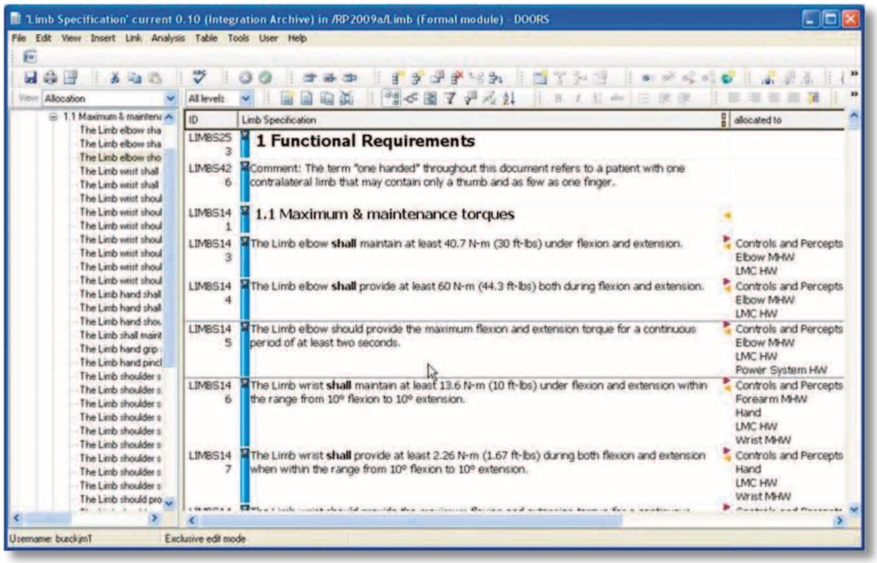


**Figure 2.** MPL components. (a) MPL upper-extremity prosthesis. (b) MPL peripheral nerve control/feedback using the Utah Slant Electrode Array. (c) MPL cortical control/feedback using the Utah Electrode Array. (d) Body attachment for shoulder disarticulation prosthesis. (e) Cosmesis for upper-extremity prostheses.



| ID | Limb Specification | allocated to |
|---|---|---|
| LIMBS25 3 | **1 Functional Requirements** | |
| LIMBS42 6 | Comment: The term "one handed" throughout this document refers to a patient with one contralateral limb that may contain only a thumb and as few as one finger. | |
| LIMBS14 1 | **1.1 Maximum & maintenance torques** | |
| LIMBS14 3 | The Limb elbow **shall** maintain at least 40.7 N-m (30 ft-lbs) under flexion and extension. | Controls and Percepts<br>Elbow MHW<br>LMC HW |
| LIMBS14 4 | The Limb elbow **shall** provide at least 60 N-m (44.3 ft-lbs) both during flexion and extension. | Controls and Percepts<br>Elbow MHW<br>LMC HW |
| LIMBS14 5 | The Limb elbow should provide the maximum flexion and extension torque for a continuous period of at least two seconds. | Controls and Percepts<br>Elbow MHW<br>LMC HW<br>Power System HW |
| LIMBS14 6 | The Limb wrist **shall** maintain at least 13.6 N-m (10 ft-lbs) under flexion and extension within the range from 10° flexion to 10° extension. | Controls and Percepts<br>Forearm MHW<br>Hand<br>LMC HW<br>Wrist MHW |
| LIMBS14 7 | The Limb wrist **shall** provide at least 2.26 N-m (1.67 ft-lbs) during both flexion and extension when within the range from 10° flexion to 10° extension. | Controls and Percepts<br>Hand<br>LMC HW<br>Wrist MHW |

**Figure 3.** RP2009 DOORS example.

The system requirements for the MPL were developed and maintained in DOORS (Fig. 3). The requirements were baselined and updated in DOORS, and periodic releases were made by exporting requirements from DOORS to a document format. A requirements review meeting was not held (as noted previously, this occurred before the release of the System Development Process); however, the review and sign-off of the MPL system requirements document constituted review and approval of the system requirements.

## TIP Implementation

As with many projects at APL, the TIP sponsor provides broad objectives and goals that must be turned into system requirements. The overall, high-level goals for a TIP payload generally fall into three categories: (*i*) provide confirmation of vehicle operation (fly-away Earth imagery; Fig. 4a), (*ii*) characterize the target vehicle (post-burnout debris characteristics; Fig. 4b), and (*iii*) characterize the intercept event phenomena (spatial, temporal, and spectral; Fig. 4c). Typically, when a new target sensor suite is needed, the system requirements are not impacted; however, a primary output of the requirements development for TIP is an Interface Control Document (ICD) that defines all the interfaces (mechanical, electrical, environmental, storage, contamination, etc.) between the sensor package and the flight vehicle. Interfaces between the payload's data handling unit and sensors are also controlled via the ICD.

## DESIGN

Given the wide variety of systems that are developed at APL, it is difficult to define what needs to be performed in the design activity such that it is general enough to be applicable across the Laboratory. Therefore, the required activities are relatively minimal, and much of this activity is given as guidance. Typically, the design activity begins after the requirements are defined. However, in some situations, such as with rapid prototypes, the system design may be concurrent with the requirements definition. If the system is large and complex, a conceptual design may be necessary. The conceptual design may consist of partitioning the system into major elements and functions. The functional partitions

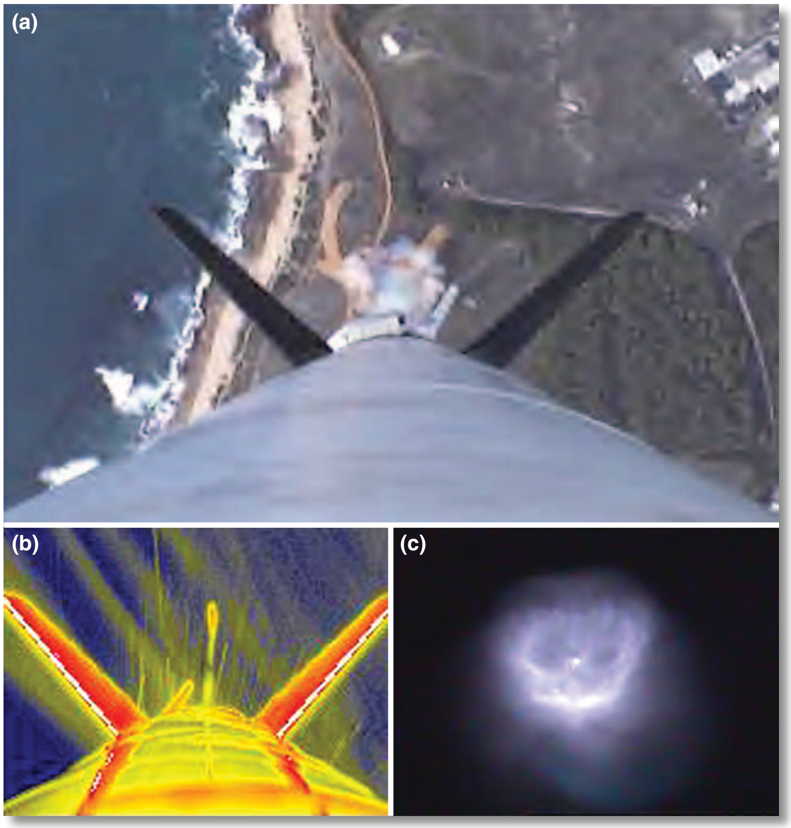


**Figure 4.** TIP goals. (a) Target ascent as viewed from aft-looking visible camera. (b) Booster post-burnout fuel debris as seen from aft-looking infrared camera. (c) Target intercept viewed from forward-looking target booster sensor payload.

between hardware and software are defined, and the interfaces between the system components may also be defined. If a conceptual design is developed, it is expected that a review of the conceptual design will be held. The system design may go directly to detailed schematics or separate phases, including a preliminary design phase. This is left to the lead systems engineer to determine, with concurrence from the PM. Regardless, a requirements baseline is established before the start of detailed design.

Much of the design activity in the System Development Process must be coordinated with the design activities within the Software Development Process and Hardware Development Process. Final hardware and software specifications are developed such that the detailed hardware and software design can be executed within the Software Development Process and Hardware Development Process. System design drawings and ICDs that specify interfaces between the hardware and software subsystems are also developed during this phase of activity.

The process requires that all systems undergo at least one design review to ensure that the design is complete and comprehensive and that it will satisfy the system requirements. As with the requirements review, the level of formality of this review will depend on the tier of the system. Representatives for the subsystems and functions under review are expected to attend the review so that details can be discussed and questions can be addressed. Depending on the development methodology, there may be requirements and design iterations. The System Development Process provides the flexibility to accommodate different design methodologies by not dictating when the reviews must occur in the development cycle. Each of these iterations does not necessarily need to go through a review, but the requirements and design of the final or "as built" system must undergo review.

During the design phase, the planning for implementation must also take place. The building of the system components will be done under the Hardware Development Process and Software Development Process. The lead systems engineer must remain cognizant of these efforts and ensure that the development and delivery of these components are coordinated and ready for implementation. Consideration is given to what must be done to receive the individual components and integrate them to perform together as a system. The team needs to consider system verification and validation, facilities for integration and testing, and system testing plans.

### RP2009 Implementation

The MPL held several formal design reviews that the sponsor attended: Design Approach Review, PDR, and CDR. For each electrical or mechanical hardware component, a Design Basis Document was developed. A Software Requirements Specification and a Software Design Document were developed for each software component. The initial versions of the Design Basis Documents and Software Requirements Specifications identified the known data, allocated the system requirements, and derived the implied design requirements. The information was supplemented and refined for the PDR with trade studies and analyses and then finalized before fabrication began after the CDR. In addition to the formal reviews, many peer reviews, which were the primary review mechanism, were held. A standard procedure for peer reviews was developed, and the meeting notice and issues worksheet for each review were maintained and available to the whole team.

### TIP Implementation

Because the TIP is an ongoing project, there is a hardware heritage of sensors and data-handling components for which to configure a specific payload. Once a sensor concept has been developed, the performance requirements are determined on the basis of the mission and target vehicle specifics. A formal design review is required by the PAIP. If the design of the instrumentation package is a minor modification of an existing design, a delta-design review is held. Before formal reviews, new instrumentation designs or updates are subject to peer review by experts who are not working on the project in "table-top reviews," as required by the PAIP. Minutes, action items, and dispositions of these reviews are captured on the project's configuration control site for wider distribution. TIP also participates in the vehicle provider CDRs and preship readiness reviews (PSRRs).

TIP performs planning for implementation during the design phase. This planning leverages previous payload build documentation, procedures, and procurements and is the responsibility of the lead systems engineer. The lead systems engineer ensures that the development, testing, and delivery of system components are coordinated and ready for implementation at the appropriate time to meet TIP milestones and deliveries to the larger flight mission program. Existing procedures and documentation are reviewed, and changes are identified and updated for each new build and component change. Fabrication and testing facility availabilities are reviewed, and alternatives are considered as part of the TIP schedule risk management.

## INTEGRATION AND TESTING

During this activity, individual components are integrated into a system, and system-level testing is executed. As mentioned earlier, the subsystems and components are built under the Hardware Development Process and Software Development Process; integration is the activity that brings these together to form the system.

Hardware and software components are delivered, along with the supporting verification and validation results of these components. For large systems, it is expected that the integration may occur incrementally. Typically, the hardware is delivered first. Initial software deliveries need sufficient functionality to support initial checkout of system interfaces. Successive software deliveries contain fixes to problems identified in previous versions and incorporate increased functionality.

An essential element of system development is a system test that consists of system verification and validation. Verification and validation may occur as one activity. However, certain portions of each may be done at different times during the System Development Process because they have different purposes. Verification is the process of determining that the system meets the requirements, i.e., that the system was "built right." Validation is the process of ensuring that the system built is the right system to satisfy the sponsor's objectives. Part of the validation process occurs during the requirements review, because the system requirements are driven by the sponsor's objectives. Portions of system validation may occur concurrently with the system verification. An example of concurrent validation and verification is when the system is tested in the expected operational environment with, possibly, the intended operators and users. Some systems, such as fielded prototypes, may not be validated until after delivery or deployment and may be included as part of sponsor acceptance.

Before system deployment or delivery, the verification and validation results are to be reviewed to ensure compliance with the system requirements. Also, any planning for the deployment activity occurs during the integration and test phase. Consideration is given to any needed plans such as those for shipping and transition.

### RP2009 Implementation

For the MPL, DOORS was used to create a verification compliance requirements matrix for each hardware and software component. A completed verification matrix contains the component requirements, an indication of whether the component is compliant, the verification methods, and any notes. After the component successfully completed verification testing, it proceeded to system integration. The system integration phase was planned and controlled via the Integration Test Plan. System validation of the MPL will be done in clinical trials and is part of a future phase of the program.

### TIP Implementation

The flight hardware for the TIP is subject to comprehensive functional and environmental testing according to configuration-controlled documented procedures before shipment to the launch vehicle provider. The testing process verifies the performance and environmental requirements identified in the ICD and design review. There is a formal verification of the testing results at a PSRR as required by the PAIP. The PSRR must be passed and all action items closed before the instrumentation package is allowed to proceed to integration with the target launch vehicle. The PSRR committee consists of experts from within the Laboratory, the launch vehicle provider and their government agent, and APL's Aegis BMD T&E sponsor. The PSRR presentations, action items, liens, and resolutions are posted on the Aegis T&E Program's configuration-control website for wider community review and comment.

## DEPLOYMENT

The final activity is deployment, which will vary tremendously depending on the specific system. The system may or may not be directly or formally delivered to the sponsor. The system may have been built to be used and supported by APL staff as either part of a large T&E effort or concept exploration. Whether or not there is a formal delivery, it is expected that approval or acknowledgment of completion will be received from the sponsor. Before delivery of the system, it is expected that the system will be reviewed for readiness for deployment. If the system is delivered externally or is to be used by others, operations support and training may be necessary and would be part of the deployment activity. Other activities such as logistics support and maintenance may be required. Regardless of the specific activities that will be performed, most will likely have been negotiated in advance with the sponsor or user of the system. Many, if not most, of the activities will be supporting deliverables that are part of the terms of the contract or an agreement with the sponsor. These will have been known early in the System Development Process and would have been accounted for in either the SysDP or the project plan.

### TIP Implementation

After the PSRR for the TIP, the system is shipped to the launch vehicle provider for the next level of vehicle integration. Figure 5 shows an instrumentation package integrated with a target launch vehicle. After the vehicle integration, the target is shipped to the test range for launch preparations and launch (Fig. 6). At each step, configuration-controlled documented procedures are used to verify that the instrumentation package is ready for flight. After the flight mission, the data from the instrumentation package are analyzed and deliverable data products are prepared. Each of the deliverable data products undergoes technical, editorial, and management review to ensure the quality of the products. The data products are provided to the sponsor's data archive

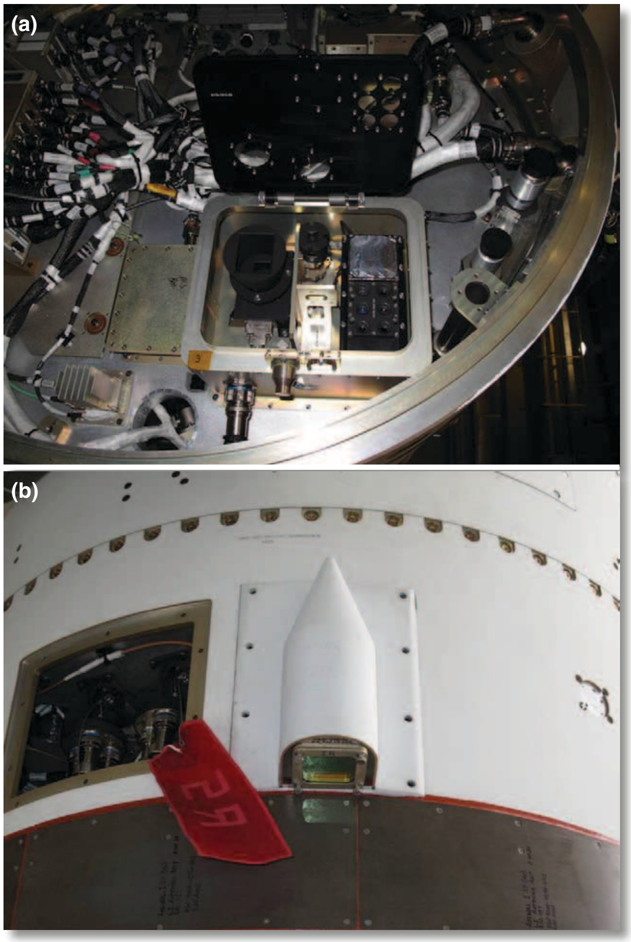


**Figure 5.** Instrumentation package integrated with a target launch vehicle. (a) Forward payload section. (b) Aft-looking pod housing infrared camera.

in accordance with a Data Management and Analysis Plan and are also posted on the program's configuration-control website. Data from the payload system support overall postmission analysis activities.

## CONCLUSION

As part of the APL QMS, the System Development Process was developed. The process captures the best practices within the Laboratory for system development and complies with ISO 9001. The process is configurable to meet the needs of the wide diversity of system developments across the Laboratory. The MPL is an example of a system development effort that followed good systems engineering practices and, if the quality process had been established at the time, would have met the requirements of the APL System Development Process. The Aegis TIP is another example of a system development effort following good systems engineering practices, and the fundamental principles of the APL's QMS Process are present in this preexisting project. The QMS processes have been tailored to match the needs of the TIP and the sponsor. Although TIP is a low-tier system, when it is beneficial, TIP leverages ongoing QMS activities to improve quality through exploitation of documentation and processes developed for high-tier system development. This "quality sharing" has paid off for the TIP by identifying Laboratory resources that might be of value to the project and have already been developed by other programs. The TIP has benefited from a formal process by having engineering drawings, documents, and procedures in an accessible format when a new build or activity is started, such as environmental testing. This has reduced the staff effort to produce a sensor package for a new vehicle and allowed the transition of new staff into the project with minimal disruption. As part of the continual improvement process, the team identifies additional cross-enterprise resources, including processes and procedures, that are of value to increase quality and producibility and to reduce cost.

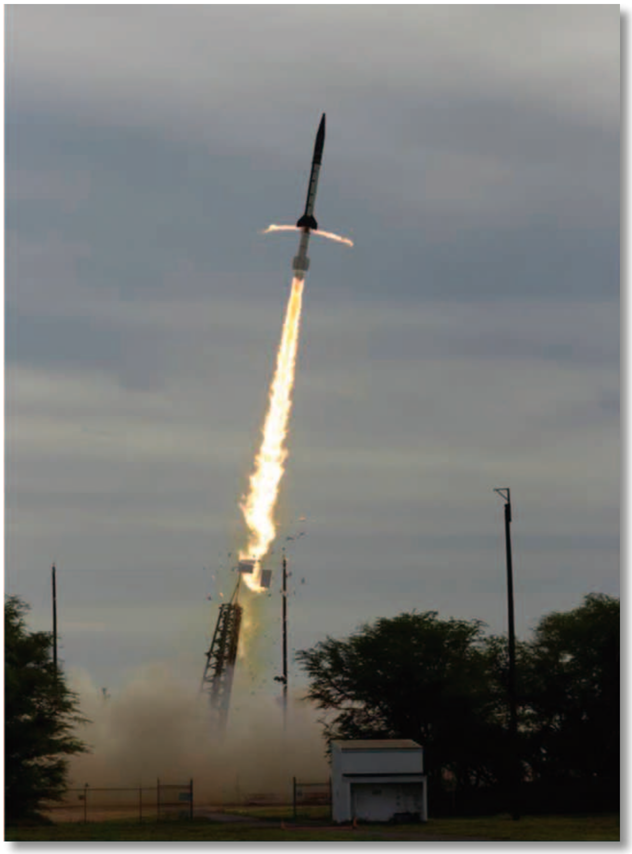


**Figure 6.** TIP sensor package in flight.

### REFERENCES

[1] Kossiakoff, A., and Sweet, W. N., *Systems Engineering Principles and Practice: Wiley Series in Systems Engineering and Management*, A. P. Sage (ed.), John Wiley & Sons, Hoboken, NJ (2003).

[2] Pisacane, V. L., and Moore, R. C. (eds.), *Fundamentals of Space Systems*, Oxford University Press, New York (1994).

[3] Griffin, M. D., and French, J. R., *Space Vehicle Design: AIAA Education Series*, 2nd Ed., J. A. Schetz (ed.), American Institute of Aeronautics and Astronautics, Inc., Reston, VA (2004).

# The Authors

**Elinor Fong**, co-owner of the APL QMS System Development Process, is the Branch Supervisor of the Systems Integration Branch of the Air and Missile Defense Department (AMDD). Ms. Fong has worked on radar systems and sensor netting systems, initially as an analyst and algorithm developer and subsequently as a systems engineer. Currently, she is supporting the Missile Defense Agency, contributing to systems integration tasks by co-leading the Sensor Registration Integrated Product Team and airborne IR sensor integration into the Ballistic Missile Defense System. **David Y. Kusnierkiewicz**, also co-owner of the APL QMS System Development Process, is the Chief Engineer of the Space Department. He has held the position of Mission Systems Engineer for the New Horizons Pluto–Kuiper belt mission and is the Mission and Spacecraft Systems Engineer for the Thermosphere, Ionosphere, Mesosphere Energetics and Dynamics (TIMED) program. Ms. Fong and Mr. Kusnierkiewicz led a team of systems engineers from most of the departments across APL to develop the APL QMS System Development Process. The first version of the process was released in March 2008. **Deborah P. Mendat** was a member of the team representing the National Security Technology Department (NSTD), and the program on which she worked, RP2009, was used to help define the APL QMS System Development Process. Ms. Mendat is currently the NSTD Department Quality Manager and has more than 20 years of experience as a systems engineer and project manager in the fields of military communications cryptography, electromagnetics, acoustics, and chemical–biological detection technologies. As the Performance Assurance Engineer for RP2009 Phases I and II, she defined, implemented, and improved management and engineering processes and procedures to comply with Food and Drug Administration Design Controls for APL and 30 subcontracting organizations. **Peter D. Tennyson** is the Assistant Group Supervisor of the Test and Evaluation Group in AMDD and is the Project Scientist of the TIP program. TIP was one of the first system development projects to begin to adapt the APL QMS System Development Process. Dr. Tennyson has an extensive background in end-to-end sensor systems development, including concept development, specification, design, fabrication, testing, data acquisition, data reduction, signal and data processing, result presentation, and upgrade design. He has supported numerous DoD programs, such as Aegis, Terminal High Altitude Area Defense, Exo-atmospheric Kill Vehicle, Standard Missile 3, and Space Surveillance and Tracking System. For further information on the System Development Process reported here, contact Elinor Fong or David Kusnierkiewicz. Their e-mail addresses are elinor.fong@jhuapl.edu and david.kusnierkiewicz@jhuapl.edu, respectively. For information on RP2009, contact Deborah Mendat. Her e-mail address is deborah.mendat@jhuapl.edu. For information on TIP, contact Peter Tennyson. His e-mail address is peter.tennyson@jhuapl.edu.

Elinor Fong

David Y. Kusnierkiewicz

Deborah P. Mendat

Peter D. Tennyson

The *Johns Hopkins APL Technical Digest* can be accessed electronically at **www.jhuapl.edu/techdigest**.