A Product Development Architecture with an Engineering Execution Representation of the Development Process

Gregory L. Neugebauer

Abstract. Successful development projects share common characteristics and attributes documented in project management curriculums and professional societies. This report gives tribute to those common elements as well as advancements in Systems Engineering to provide guidelines for system life cycle processes and activities.

Closely guarded by many corporations is the actual product development "process" which includes the steps, critical decisions, and roadmaps that enable a development team to efficiently evolve products from conceptual design to delivery and support. This paper addresses the question "can a single tailorable product development architecture be formulated and selectively optimized to address specific needs of a commercial or military application?" It will also guide the reader through a case study of creating a product development architecture and the benefits derived from the journey.

This approach will take a different perspective than the management, planning, and control viewpoints of product development and engage the topic with an engineering execution representation.

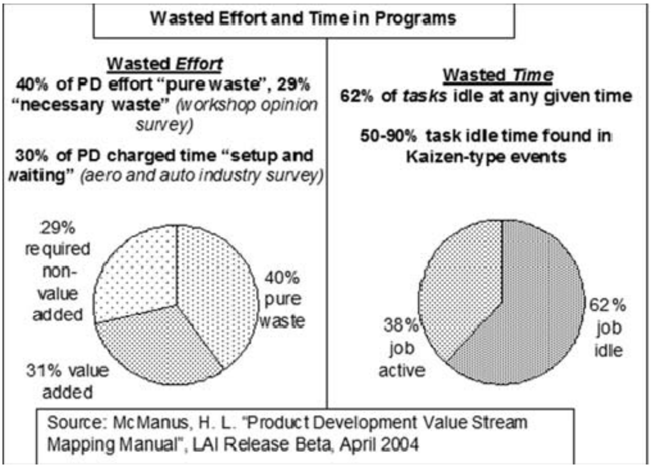
1 Introduction

The waste in traditional Product Development programs, results from a number of causes: craft mentality of engineers, poor planning, ad hoc execution, and poor coordination and communication cultures. (B. Oppenheim, 2008)

Gregory L. Neugebauer
Sandia National Laboratories,
Albuquerque, New Mexico, United States
e-mail: `glneuge@sandia.gov`

M. Aiguier et al. (eds.), *Complex Systems Design & Management 2013*,
DOI: 10.1007/978-3-319-02812-5_13,

Figure 1 below provides evidence from a study of Product Development (PD) efforts which establishes a challenge and opportunity for improved methods and tools.



Source: McManus, H. L. "Product Development Value Stream Mapping Manual", LAI Release Beta, April 2004

Fig. 1 A Product Development Opportunity

Acknowledgment must be given to the term *product development*. While it may be convienient to refer to these initiatives as products, the final deliverable is often a *system*, typically distinguished by the level of complexity.

While attempting to present a comprehensive view of a product development architecture, there are constituents that are not addressed in this research which must be inscribed within the product development architecture or as a component of portfolio management which include marketing analysis, product pricing, and end user testing.

Models and representations of the product development process may use the terms "architecture" and "framework" inter-changeably, although the use of "framework" is frequently associated with portrayals such as DoDAF, MoDAF, and Zachman. The term "product development architecture" used throughout this paper will refer to a substructure within the enterprise architecture.

2 Motivating Factors

The fact that most projects still fail in some manner, suggests that conventional project management does not meet current business needs. Although the conventional project management body of knowledge forms a good foundation for basic training and initial learning, it may not suffice for addressing development of complex systems. (Shenhar & Dvir, 2007)

Furthermore, to acknowledge the demand for a product development architecture, we must recognize the vast difference between the stable world of repetitive manufacturing and the high-variability world of product development. A product development process must thrive in the presence of variability. (Reinertsen, 2009)

From an engineering execution perspective, this will include:

- Learning from waste
- An antithesis of the earned value approach
- Critical early stage enablers, engagement of stakeholders, and building team confidence
- Application of Systems Engineering principles
- Recognition of an inefficient development process
- Maximization of agile concepts

In product development, a cross-functional team is brought together because its members have collective knowledge that cannot be held efficiently by any individual. This collective knowledge is not present by definition when the team is merely assembled, it is only potentially present. (Madhaven & Grover, October 1998) This condition begs for a product development architecture that catalyzes team members into a common goal and purpose.

3 Timeline

As a formal discipline, project management as we know it was born in the middle of the twentieth century. The Manhattan Project, which built the first atomic bomb during World War II, exhibited the principles of organization, planning, and direction that influenced the development of standard practices for managing projects. During the Cold War, large and complex projects demanded new approaches. In programs such as the U.S. Air Force intercontinental ballistic missile (ICBM) and the Navy's Polaris missiles, managers developed a new control procedure called program evaluation and review technique (PERT). This approach evolved simultaneously with the critical path method (CPM) which was invented by DuPont for construction projects. (Shenhar & Dvir, 2007)

The premier project management organization, the Project Management Institute (PMI) was founded in 1960 and has since done a remarkable job in building the guide to the Project Management Body of Knowledge (PMBoK), which has become the de facto standard of the discipline. (Shenhar & Dvir, 2007)

The modern discipline of Systems Engineering (SE) was developed in the ballistic missile program by Si Ramo and Dean Woldridge in 1954, with the first formal contract to perform "Systems Engineering and Technical Assistance

(SETA)." Under this contract, Ramo and Wooldridge developed some of the first principles for SE and applied them to the ballistic missile program—considered the most successful major technology development effort ever undertaken by the U.S. Government. (Jacobson)

"Total Quality Management" (TQM) which swept the industry by storm in the early 1980s was led by Deming [1982]. It was an attempt to adopt the successful Japanese industrial management methods to the U.S. industry. A strong message of TQM was that pursuit of higher quality is compatible with lower costs. (B. Oppenheim, 2008)

An important component of Concurrent Engineering (CE) was multifunctional design teams, sometimes called Integrated Product Teams or IPTs, which included representatives from the subsequent phases in the upfront engineering design. CE when effectively implemented with electronic design tools and workforce training led to dramatic reduction in design rework and, consequently, cost and schedule. (Hernandez, 1995)

Originating at Motorola several years later and relying on rigorous measurement and control, Six Sigma focused on systematic reduction of process variability from all sources of variation: machines, methods, materials, measurements, people, and the environment. (Murman, et al., 2002)

The term Lean as an industrial paradigm was introduced in the United States in the bestselling book, *The Machine That Changed the World, The Story of Lean Production* published by the MIT International Motor Vehicle Program [Womack, Jones, and Roos, 1990], and elegantly popularized in their second bestseller *Lean Thinking*, [Womack and Jones, 1996]. The authors identified a fundamentally new industrial paradigm based on the Toyota Production System. The paradigm is based on relentless elimination of waste from all enterprise operations, involving the continuous improvement cycle that turns all front-line workers into problem solvers to eliminate waste. (B. Oppenheim, 2008)

4 Transformation of Product Development Architectures

Research into the formulation of product development processes results in a deductive conclusion that many architectures or structures are the result of "accidental architectures" which evolved from sensible initiatives (TQM, Concurrent Engineering, Six Sigma…) or synthesized business practices. It is not uncommon for companies to treat internal processes as intellectual property, and when examined closely require non-disclosure agreements for release.

The construction of an architecture offers a unique opportunity for many companies. As will be explored later in this paper, product development can be considered a form of knowledge management.

Within any organization there exists a reluctance to change. A carefully constructed product development architecture which takes on an engineering approach with value based principles will find an eager audience of supporters. The method and approach used to develop the architecture will likely uncover strengths, weaknesses, opportunities and threats (SWOT) that otherwise would remain unacknowledged. Leveraging these conditions will provide the organization a pathway to improved marketability and profitability.

Benefits realized by the shear process of creating a product development architecture may be enough to initiate and justify this endeavor.

A key outcome of the architecture may be the mental transformation from viewing the deliverable as a present day product to a future state system. The rewards of this transformation will be realized many times over as subsequent systems are delivered.

The presence of a product development architecture presents a compelling argument for improved and consistent supply chain management. All suppliers will understand the techniques, tools, and processes used by the integrating entity. Outsourcing portions of the product development will burden less risk as all parties are acutely aware of their responsibilities. Research and technology entities both within and outside the organization will have direct line-of-sight to the key goal of delivering a product to the customer.

Another benefit of adopting a product development architecture is the agility to move technical personnel between projects and organizations. It is difficult to overestimate the importance of having an agile workforce that can move between assignments without the encumbrance of a new product development style, and it will drive ideal behaviors. Using a common methodology, multiple interpretations of business practices are significantly reduced. A common product development architecture will also enable projects to ease staffing transitions as transfers, promotions and retirements take place.

Productivity will be enhanced by accelerating each project teams‘ transition phase by moving through “forming, norming, and storming“, allowing the team to reach the “performing“ stage (Psychologist Bruce Tuckman’s phrase) in a shorter time. Team members will understand their roles and feel secure about their presence on the project.

Monitoring of projects will improve and manifest decisions to increase funding of promising work, allowing this process to be data driven. A sense of urgency, early in the project will also be established, avoiding the late stage heroics often associated with product delivery.

The Diamond Approach to Project Management

To measure if a tailorable product development architecture can be developed, Shenhar and Dvir provide a useful examination of the many types of product development campaigns that may exist [*Reinventing Project Managment, The Diamond Approach to Successful Growth and Innovation*]. By categorizing the project, a unique footprint can be visualized. See Figure 2.

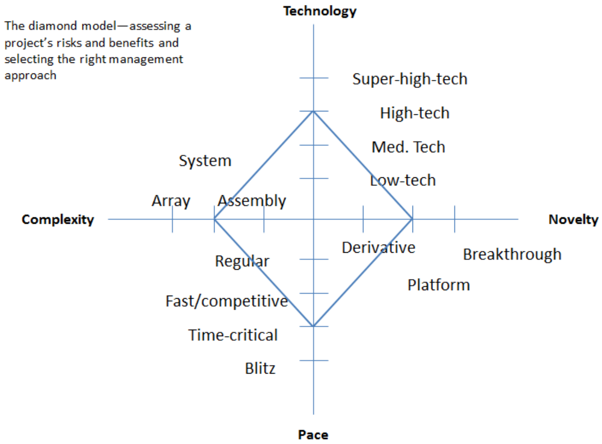


Fig. 2 NCTP Model (Shenhar & Dvir, 2007)

The diamond approach looks at the four dimensions of novelty, technology, complexity, and pace of a project. Novelty, as used by the authors, involves the uncertainty of the goal that is to be achieved and is an indicator of how clear or unclear the project goal and requirements are. Is the project about developing a new derivative, building upon an existing platform, or creating a breakthrough product or service? Technology describes the level of technological uncertainty involved and ranges from low to super high-technology. The complexity of a project, however, involves both the complexity of the product and of the organization involved. The measures of complexity range from an assembly of components to a complex system of systems. Finally, pace is the urgency required for the project, or its time frame for completion. (Mulenburg, 2007)

From Embedded to Embodied Knowledge

Using the notions of tacit and distributed cognition as a basis, Madhaven and Grover have proposed a model that links team members‘ and leaders‘ cognitive attributes and the team's process attributes to the efficiency and effectiveness with which the potential knowledge, resident within the team, is realized as a new product. (Madhaven & Grover, 1998). Their belief is that approaching product development as knowledge management will enhance the understanding of this critical process.

Participants in the Systems Architecture Forum identified Reference Architecture as a knowledge repository which facilitates knowledge transfer and communication. A Reference Architecture aids the understanding of the basic architectural and design principles. A Reference Architecture can also serve as a lexicon of terms and naming conventions as well as structural relationships within a company, industry or a domain. (Cloutier, Muller, Verma, Nilchiani, Hole, & Bone, 2010)

While the success of a system or product is ultimately determined by conformance to requirements and end use performance, failures may occur due to dysfunctional project teams. Thus, addressing the needs of project team members can be a critical success factor for a project.

Existence of a formalized product development architecture serves the development teams by building trust between team members. As described by Madhaven and Grover, the aggregate level of trust in team orientation will be related positively to the efficiency and effectiveness that embedded knowledge is converted to embodied knowledge. This principle applies equally to the team members‘ trust of one another.

The presence of an architecture also invites front loading of the project. This strategy has been found to be a key indicator of project success and allows program managers the visibility to determine if development funding should be continued or curtailed. The earned value approach in contrast, is better suited for serial type initiatives that have been previously executed. As will be discussed in the next section, incremental innovation is a possible candidate for applying earned value constraints to a project. Innovative or significant changes to the interfaces or use-application of a product must be managed differently than incremental improvements to a product with the similar interfaces.

Architectural Innovation

Henderson and Clark assert that innovation can be categorized as incremental or radical and that each has competitive consequences requiring different organizational capabilities.

Architectural innovation is characterized as a change in the way constituent components are linked together in the system. Radical innovation is identified when a new dominant design emerges for the organization and creates significant challenges since it extends the usefulness of existing capabilities. It needs new modes of learning and a unique skill set of developers. See Figure 3.

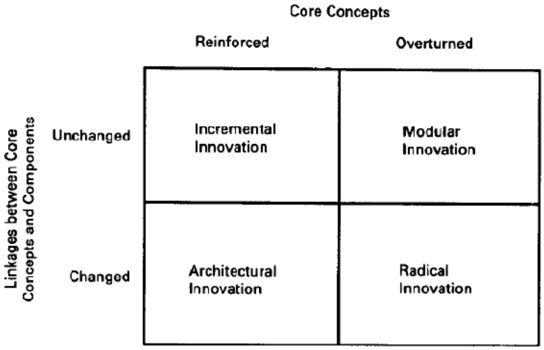


Fig. 3 A Framework for Defining Innovation (Henderson & Clark, March 1990)

Under this construct, tailorable product development architectures are needed to address the complexities and demands of various levels of innovation or radical challenges.

A product development architecture with focus on an engineering representation could prove to be an extremely useful tool to bridge the boundaries of innovation.

If the architecture were tailorable, the potential time-savings in understanding the variable focus areas that must be addressed is considerable. Incremental innovation would be streamlined and rely on the development efforts of previous systems where possible. Radical innovation would necessarily assume more risk, so additional validation and verification measures may be necessary.

Architectural innovation would benefit from a product development architecture by thorough evaluation of interfaces and modifications necessitated by the emergent new environments that the product or system would experience.

Mikkola provides an image to aid architects to evaluate the modularity of products that will be driven by the architecture. See Figure 4. The work outlines testable propositions and discusses the managerial and theoretical implications for the modularization function. (Mikkola, 2006)

	Customizable	Non-Customizable
Standard	n_{STD-C} • off-the-shelf components • detail-controlled components	n_{STD-NC} • carry-over components • supplier-proprietary components
New-to-the-Firm	n_{NTF-C} • new materials • new versions of upgradable components • modular innovations	n_{NTF-NC} • unique components • product-specific components

Fig. 4 Classification of Components

Meyer and Dalal observed that high levels of reuse generally indicate that a product family was developed with a platform discipline. Upon analysis, one product family showed substantial platform discipline, emphasizing a common architecture and processes across specific products within the product line. The other product family was developed with significantly less sharing and reuse of architecture, components, and processes. The conclusion offered that the platform-centric product family outperformed the latter along a number of performance dimensions over the course of the decade under examination. (Meyer, 2002)

Shingo Model

The Shingo Prize is awarded to organizations that demonstrate a culture where principles of operational excellence are deeply embedded into the thinking and behavior of all leaders, managers, and associates. (Huntsman Business School; Utah State, 2012)

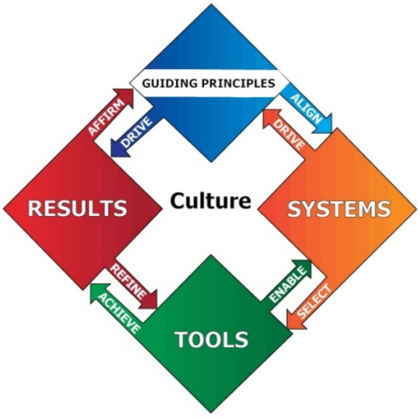


Fig. 5 The Shingo Transformational Process

The Shingo Prize for Operational Excellence is named after Japanese industrial engineer Shigeo Shingo. Dr. Shingo distinguished himself as one of the world's thought leaders in concepts, management systems and improvement techniques that have become known as the Toyota Business System. (Huntsman Business School; Utah State, 2012) A model for the transformational process is shown in Figure 5.

Principles associated with operational excellence can be summarized.

1. There is a clear and strong relationship between principles, systems, and tools.
2. Operational excellence requires focus on both behaviors and results.
3. Business and management systems drive behavior and must be aligned with correct principles

Although structured for operational excellence the Shingo model provides a strategic departure point for augmentation of new product development architectures.

5 Perspectives of Architecture

A comprehensive product development architecture initiative must recognize that an integrated effort with top-down and bottom-up construction, utility, and deployment will be needed.

In optimized cases the product development architecture will interface with the corporate reference architecture as shown in Figure 6. (Muller & Hole, 2007)

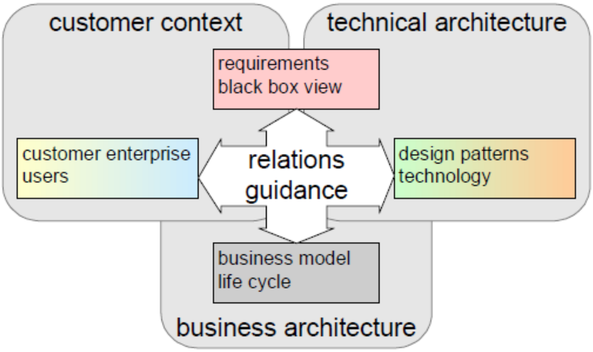


Fig. 6 Domains of the Reference Architecture

In practice, business architectures and customer context are often missing from the fully described reference architecture. (Rosen, 2002) Techniques using a SysML taxonomy or the Boardman Conceptagon potentially offer a systematic and critical thinking approach to address the complexities of integrating a technical architecture into a larger reference architecture. (Boardman & Sauser, 2008)

Conceptually, the technical architecture will scope the boundary, interior and exterior conditions. This paper will address these constituents, using a derivative of the Shingo model to address:

- Culture, Principles, Subsystems, Results (interior)
- Critical Tasks (interior)
- Ideal Behaviors (interior)
- Tools (interior)
- Links to critical resources and compliance with corporate business practices. (exterior)
- Communication Strategy (boundary)

Culture, Principles, Subsystems, Results

Culture is what we do when no one is looking and is behavioral based. Behavior is something that we can see and observe. Rich personal interaction, consisting of direct, frequent, and essential information exchange among team members, will influence the trust in the teaming commitment of other members positively. (Madhaven & Grover, October 1998)

NASA describes the Systems Engineering culture as a pervasive mental state and bias applied to problem solving across the development lifecycle and at all levels of enterprise processes. (NASA, 2007)

Principles are used to guide an organization throughout its life in all circumstances, irrespective of changes in its goals, strategies, type of work, or top management. Different companies will adhere to a different set of core principles and it is critical that the product development architecture reflect these principles.

Systems can be considered an organized collection of parts (or subsystems) that are highly integrated to accomplish an overall goal or defined objective. Companies new to systems engineering adoption will necessarily need to emphasize a systems thinking approach to product development.

Results are the observable and measurable outcomes of a system and evidence that the system met customer requirements. Best practices identified by NASA include visible metrics, effective measures and visible supporting data for better decisions at each organizational level. (NASA, 2007)

Critical Tasks

Critical tasks further refine the ideal behaviors into industry standards or corporate determined best practices. The tasks must support the ideal behaviors and delineate the actionable elements into manageable portions.

Ideal Behaviors

With behaviors now characterized as observable activities, a product development architecture will detail activities to a level that assures customers, stakeholders and participants that critical tasks are conducted with appropriate rigor, responsibility is properly delegated, and that the scope or breadth of work expected is appropriately communicated.

Tools

Tools are devices or process that aid in accomplishing a task. Tools can be a point solution, or a generic set of applications that can be applied to solving a problem or developing a solution.

Product Development tools, may be thought of as being formalized means for aiding product developers to carry out their jobs, and often summarized by terms such as Methods, Approaches, Diagrams, Guidelines, Models, Working Principles, Procedures, Representations, Standards, Steps, Techniques, and Methodologies. (Araujo & Duffy, 1996)

An important complexity in product development architectures is examination of the question "do tools drive architecture or does architecture drive tools?" At one abstraction, the product development architecture can be viewed as a tool itself. At another, the architecture may be represented by a comprehensive and integrated tool suite.

Links

Many organizations find themselves replete with business practices, quality assurance guides, and corporate policies and governance. Many of these regulations should flow-down into the product development architecture to insure that product launch is not encumbered by technical, supervisory or administrative delays.

Architects may consider these business procedures as exterior elements of the product development architecture with internal coupling.

Communication Strategy

Inherent in the implementation of any architecture, a training and communication strategy must be developed and supported by both key stakeholders and staff or user community. Pilot projects may be used for testing the architecture on a limited scale, but eventually the architecture will be broadcast to a larger user base.

The extent of the shared models will be related to the efficiency with which embedded knowledge is converted to embodied knowledge. Shared models will also have a relationship with the effectiveness with which the embodied knowledge is converted. (Madhaven & Grover, October 1998)

6 An Approach

An examination of an initiative at Sandia will demonstrate the utility of an engineering perspective. Figure 7 depicts the Principles and Ideal behaviors as foundational elements, while the Phases distinguish the maturity of the product as it traverses the development cycle. Tasks, Behaviors, Tools and Links provide a second tier of detail discussed in later sections.

Creating the Architecture

The process of creating the case study architecture was launched with formal sponsorship from executive management. As with most projects, a vision evolved from a decomposition of the “as-is” state into the conception of the future state. This vision matured into a mission statement with objectives and strategies. The constraints and assumptions were discussed and acknowledged. Fundamentally, desire to change along with foundational knowledge and a team of motivated professionals were the primary ingredients.

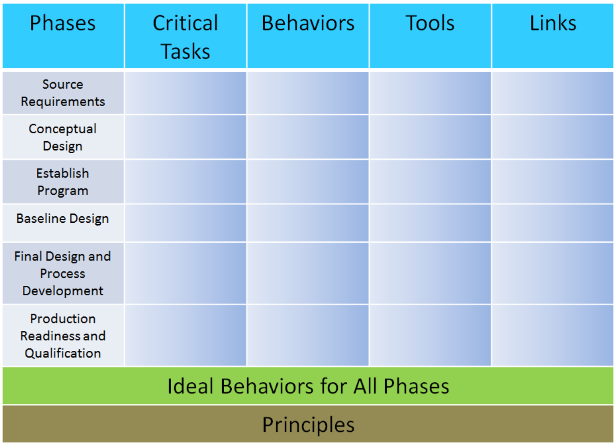
The purpose of many product development activities is to produce information that increases certainty about the ability of the design to meet requirements— i.e., these activities decrease performance uncertainty and assure a manufacturable product. Since risk is the product of uncertainty and consequences, reducing uncertainty equates to reducing risk in many cases. The objective of the product development activities is to drive this index to an acceptable level. Product development activities add value when they contribute towards this objective. (Browning, 2003)

Eventually the primary consideration of guiding principles and constructs of the architecture evolved. The recent enterprise adoption of integrated phase gates for all projects was apparent. In keeping with systems engineering principles outlined by INCOSE and corporate governance, these phase gates became the backbone of the architecture.

Further decomposition was approached using parallel information gathering and critical thinking approaches, in both focused forums and working meetings. Phases were broken down into tasks which were further resolved into behaviors. All through this process, management remained involved as active listeners or barrier-busters to furnish resources and offer perspective.

Checklists proved to be an important credential for the architecture, found in corporate quality, business, and technical procedures. Input and output criteria for each phase gate were examined and the critical activities to support these requirements identified and turned into actionable statements.

Only after the behaviors were narrowed did the architects begin to evaluate tools and links.



Phases	Critical Tasks	Behaviors	Tools	Links
Source Requirements				
Conceptual Design				
Establish Program				
Baseline Design				
Final Design and Process Development				
Production Readiness and Qualification				
Ideal Behaviors for All Phases				
Principles				

Fig. 7 An Enginnering Execution Perspective

Ultimately the architecture was "rolled-out." Several training sessions were offered to introduce the architecture. Deployment was supported by the presence of an internal web-site that provided a hierarchical representation of the architecture and active links to tools and infrastructure systems supporting the initiative.

Presently, the web-site provides a vital communication utility. It offers the potential to consolidate project data, while integrating various intra and inter organizational functions and processes. Integration challenges of web-based product development tools described by Sethi et al. and decisions on scaling the information technology investment for appropriate sophistication were examined. (Sethi, Pant, & Sethi, 2003)

Principles

Perfection

We build on our learning.

We consider designing products as a family (rather than single products) to obtain multi-product synergy.

Pull

We consider with our customer a broad set of options and trade spaces.

We seamlessly transition to production manufacturable designs that meet customer requirements (performance, schedule), are cost effective, and are sustainment-friendly.

Flow

We establish a project baseline early in order to understand the impact of changes and communicate those to the customer.

We understand, manage, react, and communicate risk effectively.

We recognize and act on critical decisions based on data.

Value Stream

We have a process that translates customer requirements to a manufacturable design based on project management and system engineering principles.

Value from Customer Perspective

We have active, continuous, and ongoing engagement with the customer.

We put ourselves in our customers' shoes to understand their environments and how they measure our success.

Ideal Behaviors for All Phases

A) Clearly define and negotiate roles and responsibilities throughout the project with high emphasis on interfaces with the customer and production.

B) Utilize a decision making process (i.e. who makes decision, identify points in time where decisions should be made, how decision is made, constraints, etc.) that will minimize/avoid rework and stick to it!

C) Identify a core set of peer reviewers (external to the Integrated Product Team) who will support the team throughout the process.

D) Integrate peer reviewing as early as possible to the extent that value is added and document results.

E) Consider risk impact in all planning and decisions.

F) Complete the check/act cycle continuously in order to build on learning and measure the project against customer expectations and requirements.

G) Collaborate with the appropriate people early in each phase to minimize rework from the reviews by addressing matters early in the process rather than later.

H) Go to where the work is done (internal, external, and customers) to see the environments first-hand.

I) Apply good project management skills by balancing and communicating the three tradeoffs of cost, schedule, and performance.

J) Consider next assemblies and subassemblies in the execution of this architecture.

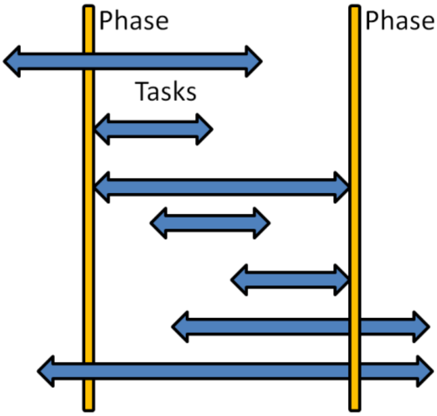


Fig. 8 Critical Tasks Muste Extend Beyond Phases

Phases, Critical Tasks, and Behaviors

Prior to further examination of the case study a review of task structure is paramount. As cautioned by Reinertsen, tasks must be allowed to start when appropriate. Some may be constrained to a particular phase, or require passage of a milestone to begin. It is vital that the majority of these tasks remain flexible to start and complete as needed. The following sections describe these tasks; however they are presented as suggested starting points. See Figure 8.

Throughout the following section, Phases are shown in bold, Tasks are numbered and Behaviors designated by letter.

Assess Source Requirements Phase

1. Identify mission and scope of work (charter)
A) Question what is in scope and what is out of scope
B) Request and/or receive authorization for work
C) Identify stakeholders and customers
D) Clearly define measurable stakeholder/customer expectations/requirements for budget, schedule, deliverables, and constraints

2. Identify cross-functional team
A) Identify resources for roles (e.g. Financial, Quality, Design, Production, Customer, Management, Modeling, Supply Chain, Tooling, S&T, ES&H, Statistics, or any role that may be used in the product lifecycle).

B) Define responsibilities for roles defined in A
C) Consider all phases of the project when identifying roles
D) Negotiate roles& responsibilities with each team member and his/her manager (to meet both project needs and competency growth)
E) ID mentoring opportunities and resource/skills gaps (backups so no skill only one-deep), feeding those back to management for resolution

3. Draft project plan (resource plan, risk, budget, and schedule
A) Build a project plan collaboratively with the appropriate team members with a customer perspective
B) Determine project schedule strategy using a graded and phased approach (Don't over plan early!)
C) Plan for minimal fractionation of resources
D) Plan for team member co-location
E) Plan value-added tasks and decisions
F) Determine critical decisions that would preclude moving forward without substantial rework downstream
G) Integrate team building philosophy into project schedule (e.g. milestone celebrations)
H) Know the critical path
I) Request budget profile to start high and taper off to support planning activities before changes are expensive and time consuming
J) Strive to receive budget as planned/requested over the course of the program to minimize churn and rework
K) Consider Design, Qualification, Manufacturing, Sustainment, and Retirement aspects of lifecycle
L) Estimate budget for resources (people, facilities, equipment) for lifecycle
M) Understand the funding sources to avoid gaps in resource plan (i.e. who pays for what)
O) Work with management on make-buy decisions
Note: resource plan = budget (source and allocation), people, facilities, equipment, and procurement

4. Create risk management plan
A) Identify risks with corresponding plans to either Exploit, Mitigate, Avoid, Transfer, or Accept (Execute to EMATA)
B) Consider consequence, likelihood, and detectability (ability to detect risk occurrence) aspects in scoring each risk

5. Obtain & analyze source requirements
A) Distinguish requirements from background information/rationale/applicability
B) Question the requirement to understand the rationale and margin
C) Ensure the requirements are written using the SMART (specific, measurable, attainable, realistic, timely) guidelines

D) Consider all applications for the product as well as product capability and functioning with the next assembly
E) Obtain not only fit, form, and function requirements, but also consider cost and schedule
F) Consider the full life cycle of the product when developing requirements
G) Collaborate with stakeholders/customers to establish baseline and detailed requirements
H) Educate the customer about the options based on design, manufacturability, cost (e.g. costs of negotiable requirement verification) , risk, and schedule
I) Know what can and cannot be negotiated (musts vs nice-to-haves, know your tradeoffs)
J) Be open to any and all design options and associated risks
K) Consider how each requirement will be verified to enhance understanding of requirements
L) Understand and challenge assumptions that drive requirements

6. Negotiate requirements
A) Distinguish requirements from background information/rationale/applicability
B) Question the requirement to understand the rationale and margin
C) Ensure the requirements are written using the SMART (specific, measurable, attainable, realistic, timely) guidelines
D) Consider all applications for the product as well as product capability and functioning with the next assembly
E) Obtain not only fit, form, and function requirements, but also consider cost and schedule
F) Consider the full life cycle of the product when developing requirements
G) Collaborate with stakeholders/customers to establish baseline and detailed requirements
H) Educate the customer about the options based on design, manufacturability, cost (e.g. costs of negotiable requirement verification) , risk, and schedule
I) Know what can and cannot be negotiated (musts vs nice-to-haves, know your tradeoffs)
J) Be open to any and all design options and associated risks
K) Consider how each requirement will be verified to enhance understanding of requirements
L) Understand and challenge assumptions that drive requirements

7. Identify requirement option trades
A) Be open to any and all design options and associated risks
B) Challenge the team to keep thinking of potential requirement trades
C) Establish acceptable margin
D) Identify critical performance and quality parameters and their weight/priority per customer perspective

Complete Conceptual Design Phase

8. Create design concepts
A) Weigh design options against requirements (musts and nice-to-haves)
B) Diversify and challenge the team to keep thinking of options until multiple options exist
C) Draft physical and functional architecture considering interfaces (see system engineering terminology for further definition)
D) Review previous and existing designs and identify commonality and improvement opportunities
E) Create scalable designs based on requirement space (i.e. that vary across systems or vary overtime)
F) Balance variations in design attributes to meet your requirements
G) Identify options that can be modeled/simulated
H) Engage R&D subject matter experts to understand leading edge technology options

9. Analyze and begin down select design options and document
A) Understand design relationship with manufacturing options
B) Use modeling to analyze design options
C) Consider margin and maturity of technology when down selecting designs
D) Quantify design options against requirements to prioritize options
E) Establish and weigh down select criteria
F) Consider existing manufacturing and product capabilities
G) Analyze risk for lifecycle cost, schedule, and performance
H) Consider full lifecycle costs when down selecting
I) Carry forward as many options that passed down select criteria as feasible

10. Identify manufacturing options and capabilities
A) Collaborate with Design and Manufacturing personnel on manufacturing options corresponding to multiple design options
B) Compare existing manufacturing capabilities to manufacturing options to identify maturity gaps with a goal of commonality
C) Consider any and all manufacturing options and associated risks
D) Be open to design options that currently may lack manufacturing capability
E) Obtain or predict cost/benefit, maturity, process capability, facilities, and capacity for proposed manufacturing processes
F) Begin to identify critical manufacturing parameters
G) Consider the number of process steps, piece parts, and time
H) Consider make/buy and supplier trades and options
I) Consider lifecycle impacts (flow and future use) within manufacturer and supplier
J) Identify manufacturing constraints (e.g. maturity, floor space, etc.)

11. Draft system qualification plan
A) Define constraints and strategy for the qualification activities
B) Set a strategy such that rework is minimized to accomplish qualification (e.g. plan, requirement mapping, demonstration of each requirement, data collection system, and final report)
C) Determine how to demonstrate each requirement
D) Consider and plan all resource needs to execute qualification plan (e.g. assets, facilities, people, etc.)
E) Consider multi-system use in qualification planning
F) Consider both product and process in the Qualification Plan
G) Consider next assembly and subassemblies in your strategy
H) Glean lessons learned by reviewing legacy qualification documentation

12. Begin down selecting manufacturing options and document
A) Use modeling to analyze options
B) Consider margin and maturity of technology when down selecting
C) Quantify manufacturing options against design options to prioritize options
D) Establish and weigh down select criteria (e.g. manufacturability, Manufacturing Readiness Levels)
E) Carry forward as many options that passed down select criteria as feasible
F) Use data collected in previous step (i.e. analyze and begin down select design options and document) to down select manufacturing options

13. Conduct Conceptual Design Review
A) Capitalize on previous collaboration with stakeholders to conduct the Conceptual Design Review to minimize rework

Establish Product and Program Design Phase

14. Document conceptual manufacturing process flows
A) Decompose physical architecture into major process steps
B) Perform gap analysis to identify new process needs with goal of commonality across products
C) Develop the strategy and supporting schedule to address manufacturing needs concurrent with the design
D) Simulate flow and capacity which would include takt times, level loading, work cell configuration, process capability (goal of Cpk of 1.3), minimizing costs& risks, integration with other products, etc.
E) Incorporate quality into processes vs. inspection and testing at end of processes
F) Mistake proof processes
G) Model and minimize environmental impact and reduce waste streams, benefiting health and safety

15. Baseline project plan
A) Detail the schedule collaboratively with the appropriate team members with a customer perspective
B) Plan for minimal fractionation of resources
C) Plan value-added tasks and decisions
D) Determine critical decisions that would preclude moving forward without substantial rework downstream
E) Integrate team building philosophy into project schedule (e.g. milestone celebrations)
F) Know the critical path
G) Request budget profile to support lifecycle costs
H) Strive to receive budget as planned/requested over the course of the program to minimize churn and rework
I) Consider Design, Process capability, Qualification (process, product and supplier), Manufacturing readiness, On-going production, Sustainment, and Retirement aspects of lifecycle
J) Refine and submit budget for resources (people, facilities, suppliers, equipment, risk mitigation)
K) Understand the funding sources to avoid gaps in resource plan (i.e. who pays for which activity)
L) Place project under formal change control
Note: resource plan = budget (source and allocation), people, facilities, equipment, and procurement

Create Baseline Design Phase

16. Perform detailed design
A) Capture interfaces and functional requirements into product definition
B) Calculate margins to understand tradeoffs between designs
C) Approach design characterization in a cost effective manner (minimize redundancy, rework, and by utilizing modeling capabilities)
D) Optimize your design via trade-off between product stability, cost, and design margin
E) Design once and apply in many different uses
F) Design for: sustainment, testing, assembly, acceptance, manufacturability, state of health monitoring, mistake proofing, etc

17. Develop & execute test plan
A) Iteratively execute Plan, Do, Check, Act
B) Use Design of Experiments to develop test plans
C) Correlate accelerated aging with requirements
D) Understand your risks associated with accelerated aging, HALT and HASS philosophies, and margin determination
E) Plan for test facilities in advance to ensure meeting schedules

F) Balance fidelity of testing against individual requirements with efficiency of enveloping
G) Test plan consists of mapping of requirements, pedigree of parts, risks of using different pedigrees, conformance vs acceptance tests, statistically significant sample sizes, assumptions, hypothesis, etc.
H) Understand the measurement uncertainties
I) Incorporate computational simulation to a level consistent with the fidelity of the models
J) Use historical data to augment testing
K) Balance margin with cost when down selecting to baseline design
L) Improve models based on test results
M) Update product definition based on test evaluation/results

18. Update and release qualification plan
A) Update qualification plan based on test evaluation/results
B) Update the requirement map to reference requirement demonstration
C) Update resource needs to execute qualification plan (assets, facilities, people, etc.)
D) Build and test to emulate any variability expected in production
E) Ensure the Qualification Plan is for product and process
F) Peer review the Qualification Plan and update as necessary

19. Conduct Baseline Design Review
A) Capitalize on previous collaboration with stakeholders to conduct Baseline Design Review to minimize rework

Finalize Design & Process Development Phase

20. Formalize product definition support drawings
A) Define the what's not the how's
B) Conduct tolerance analysis to allocate tolerance
C) Map requirements to product definition/critical features
D) Ensure that the product built matches the product qualified
E) Create product definition that can be exported directly to models (manufacturing and simulation)

21. Finalize manufacturing process flows
A) Detail and validate the capacity analysis which would include takt times, level loading, work cell configuration, process capability (goal of Cpk = 1.3), minimizing costs & risks, integration with other products, etc.
B) Create a risk-based assurance plan noting which data collection points are required before final development and after qualification.
C) Map design requirements to manufacturing processes
D) Execute schedule and monitor and build improvement plans including mistake-proofing methods

E) Define critical manufacturing parameters based on criticality to product performance
F) Design processes that allow for quick change out between processes (flexible and agile manufacturing)
G) Design for co-processing for multiple products
H) Use COTS equipment and tooling
I) Document process flows and environmental safety and health waste streams
J) Collaborate with Production to verify final development build readiness
K) Develop and implement a transition plan from development to production (i.e. transfer of staff, training, etc.)

22. Conduct Final Design Review
A) Capitalize on previous collaboration with stakeholders to conduct Final Design Review to minimize rework

Complete Production Readiness & Qualification Phase

23. Support process qualification and first production build activities
A) Partner with Production to prove in processes utilizing continuous communication
B) Conduct process walk through
C) Collect evidence for process qualification
D) Perform real-time monitoring of processes to ensure Cpk and yield goals are met
E) Complete process qualification reports based on final development lot data collected
F) Optimize processes as needed
G) Evaluate product qualification readiness (joint evaluation between Development and Production)
H) Complete the development to production transition plan

24. Verify requirements & issue complete & qualified engineering release
A) Verify and document that all requirements are met
B) Review the risk based assurance plan and modify data collection points as appropriate for full scale production

25. Support product submittal to customer
A) Continue to collaborate and support full scale production and field surveillance programs

Tools

Today's engineering departments typically standardize on particular modeling, analysis, configuration management and authorization tools. Additional tools for decision analysis, reliability estimations, requirements documentation and tracking can be ad-hoc or simply recommendations. The identification of these tools and the timeframe of when to use them can be a powerful addition to a product development architecture.

Standardization on tools is very important and will provide new team members with a list of approved applications, reference databases and general guidelines.

Tools may also include suggested or required training courses. Inclusion of this information will provide the product development architecture architects an opportunity to work with internal and external training providers to create a technical skills development program for engineering product teams

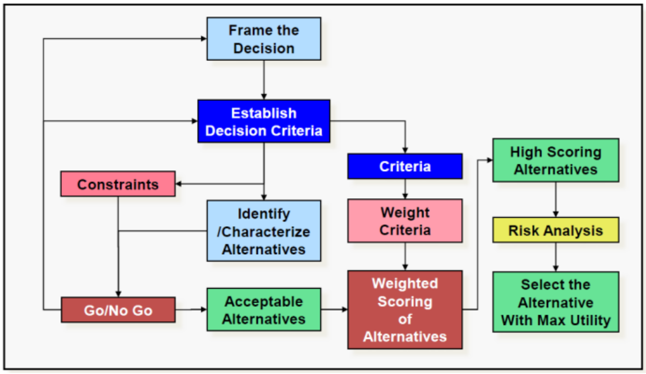


Fig. 9 Kepner Tregoe Decision Process

A time honored methodology for choosing tools can be found in the Kepner-Tregoe Decision Making process. The Kepner-Tregoe process provides a suitable method for dealing with complex, qualitative, somewhat subjective information and converting this into semi quantitative information more useful for decision making. See Figure 9.

The process provides a reasonable basis for keeping track of many factors simultaneously as must be done in comparing several complex alternatives. This approach is reasonably transparent in that it provides high visibility to the key elements leading to the final results.

Links

The utility of comprehensive and tailorable product development architecture will become evident when linking behaviors and tasks to business practices and programmatic requirements.

In particular, links to required checklists for mandatory reviews (conceptual, prototype, baseline, and final) along with the required or suggested content, list of reviewers, peers, stakeholders, et cetera, and the actionable tasks will be a powerful addition to the architecture.

A secondary and perhaps even greater value for the organization will be the ability to link these critical tasks and behaviors with work package agreements or a work breakdown structure. Handoffs and interfaces with internal and external entities will be apparent as the roles and responsibilities are associated with the behaviors.

Communication Strategy

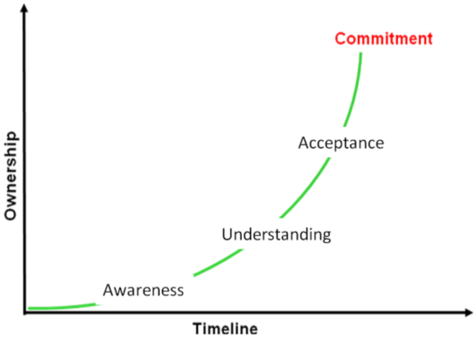


Fig. 10 Awareness to Commitment Curve

As presented by Madhaven and Grover, the path to embodied knowledge requires a significant effort to improve communications. The acceptance of a product development architecture by the organization or distinct product development teams will be significantly influenced by the degree to which this architecture is absorbed and retained by the participants.

Project teams can be considered a special class of stakeholders for emerging product development architectures. There exists a life cycle to transitions: people involved in the transition must move through the Awareness-Understanding-Acceptance-Commitment curve shown in Figure 10. (Hahn, 2013)

A singular model that quickly communicates the fundamental elements of the product development architecture is vital. Visibility of this model in conference rooms and passage ways will provide further reinforcement of the strategic product development architecture.

Traditional top level architectures are useful for managers and program leaders but strand key staff on an engineering project thirsty for greater detail and guidance. This gap is further compounded when new team leaders are assigned who have not benefitted from the mentoring of an experienced leader. Effective product development architecture must address this gap if it is to be deployed and utilized.

Details can be absent; however the image of the architecture and the specifics of where to drill down for further information is one of the distinguishing characteristics of an engineering representation of the product development architecture.

Along with principles, and the earlier illustration of communication strategy, Figure 11 and the accompanying notes provide a useful example of how a strategy can be developed and communicated.

The backbone of the Product Development System is the Integrated Phase Gates (A thru F) which are based on system engineering principles.

The Product Realization Team (PRT) is a cross-functional team which enables seamless transition to manufacturable & sustainment-friendly designs that meet customer requirements.

Having customers at true north of the objectives will encourage active, continuous, and ongoing engagement. This enables putting the development team in the customers' shoes to understand their environments and how they measure our success.

The model promotes the team to recognize and act on critical decisions based on data through iterative Plan, Do, Check, Act cycles as prototypes and final product as *systems* advance through stages of technical and manufacturing readiness.

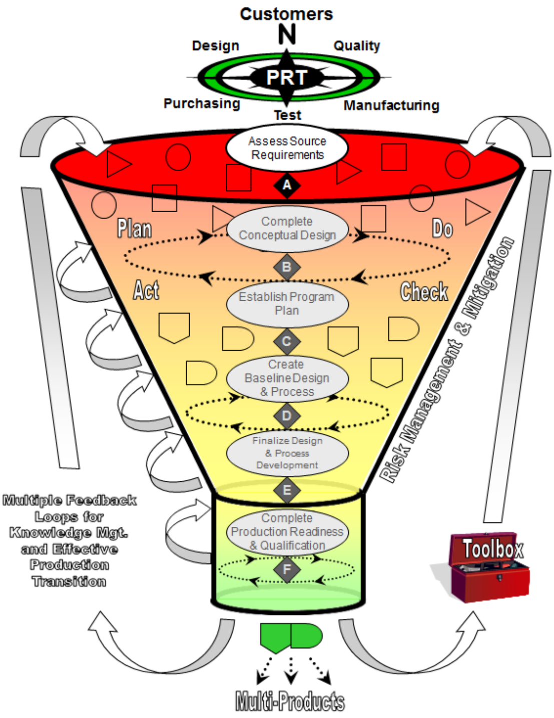


Fig. 11 Communication Strategy

The funnel shape in the system image illustrates how we carry many options as long as possible to avoid discounting options too early. Our designs are set-based to allow us to extract an optimum solution from multiple options.

Feedback loops are not only employed to build on our learning, they are emphasized to illustrate the importance of an agile system. Sequential or "waterfall" styles of product development architectures may miss this agile opportunity at all stages of the program.

Naturally, the questions of architecture "enforcement" will arise. While there are several options ranging from management authority to voluntary adoption, persuasion is a model where the architect builds relationships and a position of authority. Persuasion requires clear communication. However, even when the architecture is clearly captured in diagrams and the architects have fully applied themselves; the outcome may not be as intended. In other words, initiating architecture is insufficient to enforce it. Monitoring and adaptation of architecture is also necessary.

The construction of a product development architecture will reap many benefits. The development of teams and their dynamics is vital to project success. This is enabled by a consistent reference to the adopted product development architecture.

Figure 12 illustrates the hierarchy and integration of the product development architecture relative to the enterprise model and distinguishes the parts and relationships to the whole.

This approach will lead to rich personal interaction, consisting of direct, frequent and informal interaction among team members, and will influence the team commitment positively. In today's competitive environment, every project is schedule-driven to an extreme, while attempting to address all aspects of product development with precious resources.

Expertise is converted appreciably to embodied knowledge. (Madhaven & Grover, October 1998)

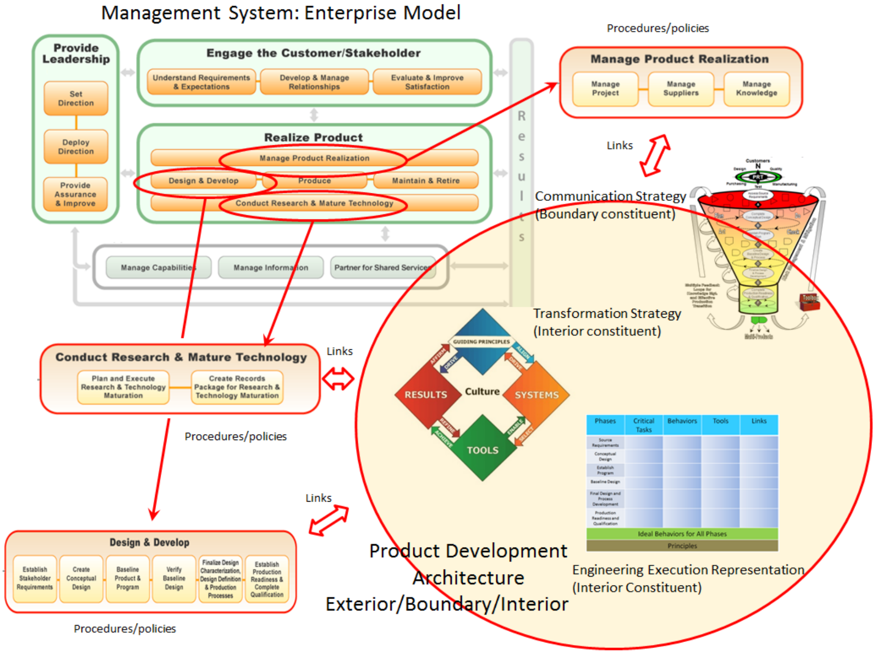


Fig. 12 Architecture Heirarchy

Future Work

Product development architectures encourage upfront communication, planning while setting expectations in accordance with budget, availability of resources, and the importance of maintaining open rapport. Architecture disciplines us to identify options early to prevent acute problem-solving dilemmas typical of schedule-driven projects.

This leads to a culture that is intended to reflect the activities we should be doing AND builds on each other's learning. Being grounded in an architecture gives the confidence to say "no" when "no" needs to be said…and that "no" is NOT a reflection of inadequacy.

7 Summary

Constructing a product development architecture will enable an organization to excel in the practice of engineering.

Architectures will strengthen management and engineering execution of programs by adopting or developing techniques and practices that enable a disciplined approach for the engineering of products and systems with assured quality by utilization of common product development practices.

In meeting this objective, organizations will provide customers and themselves with increased confidence that product or system development will reliably and consistently deliver on commitments, and will advance the state-of-the-art in the practice of engineering.

Product development architecture will provide a consistent set of tools, terms and technologies that will accelerate the transition from development to manufacturing. Manufacturing personnel will understand critical design parameters while design personnel will appreciate and consider alternative manufacturing proposals.

An architecture that requires participation from production partners will ease the difficult transition from design to manufacturing. This architecture can also be leveraged to promote a rotational program within product development to broaden and sharpen the skills of team members, abbreviate the learning curve and provide organizations with strong leaders for future projects. Opportunities may be afforded to other members to "follow" the product as it transitions from phase to phase or into full scale production.

8 Conclusion

Project management is not a linear, predictable process or a universal activity with one set of rules and processes for all projects. (Reinertsen, 2009)

A single tailorable architecture may not be possible to bridge the many types of product development that exist. A review of published architectures and execution of a sequence of steps – described in this paper, will lead the architects to a strategic product development architecture tailored and optimized for a general class of product or system development initiatives.

Product development architectures must not be viewed as static or perfect models. An effective architecture that can be tailored to the needs of the organization or the specific product under development will provide systematic agile execution of the product development cycle. An organization may choose to have a "reference" architecture, which is customized as needed to provide a point of departure for a product development activity. Additions or omissions should be defendable and provide feedback to the responsible architects improving the system.

As shown earlier, surveys indicate that 42% of product development effort is categorized as waste and 62% of tasks are idle at any given time. Given this possibility, architects should address the questions:

What types of “waste” are present in the current product development process?

What obstacles can be eliminated while developing new products that must be eliminated?

What embedded or embodied knowledge could reduce waste via the product development architecture?

Assessment should be an integral element of evaluating the product development architecture. Architecting is primarily a forward oriented activity, where by pro-active analysis and synthesis, a solution is shaped. Assessments are reactive, architecture assessments tend to detect issues that have not been covered sufficiently by the architecting effort. All processes, tools, and checklists that are proposed for architecture assessments are presumably also beneficial for the forward architecting activity. (Muller & Hole, 2009)

Finally, stimulating the conversation on new product development architectures and identification of the fundamental characteristics of both unsuccessful and proven implementations will undoubtedly contribute to richer and more useful models. The scalable nature of this work is allied to the dilating scope of architectures, and best practices of systems engineering.

Architecting is recognized as a discipline that connects customer needs and constraints with feasible technology-based solutions. The exchange of experience between practitioners from different domains may provide an overview of the status of systems architecting. (Muller & Hole, 2005) The strength of a product development architecture is its ability to be adaptable and provide an engineering execution model of the process!

Acknowledgements. The author would like to thank Eirik Hole, faculty advisor Stevens Institute of Technology for topical guidance, challenge and defense; Carla Busick, Dee Dee Griffin, Tina Hernandez, and Gary Pressly for their collective contribution to the Product Development Initiative; Cliff Renschler and Neil Lapetina for management support, review, and encouragement.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories

References

Araujo, C.S., Duffy, A.: Product Development Tools. In: III International Congress of Project Engineering, Barcelona (1996)

Oppenheim, B.W., Murman, E.M.: Lean Enablers for Systems Engineering. Wiley InterScience (2008)

Boardman, J., Sauser, B.: Systems Thinking, Coping with 21st Century Problems. CRC Press, Boca Raton (2008)

Browning, T.: The Journal of Systems Engineering 12(1), 69–90 (2009)

Huntsman School of Business, Utah State University: The Shingo Model. The Shingo Prize: `http://www.shingoprize.org/model-guidelines.html` (retrieved February 8, 2013)

Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., Bone, M.: The Concept of Refernce Architectures. Systems Engineering 13(1) (2010)

Hahn, H.A.: Managing the Project Team as a Special Class of Stakeholder for Enterprise Transformation Projects. Presented at INCOSE Enchantment Chapter Meeting, Los Alamos, New Mexico, USA (March 2013)

Henderson, R.M., Clark, K.B.: Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. Administrative Science Quarterly 9 (March 1990)

Hernandez, C.: Challenges and Benefits to the implementation of IPTs on large military procurements. SM Thesis. MIT Sloan School, Cambridge (June 1995)

Jacobson, C.: JDAM. TRW, Cleveland

Madhaven, R., Grover, R.: From Embedded Knowledge to Embodied Knowledge: New Product Development as Knowledge Management. Journal of Marketing 62, 1–12 (1998)

Meyer, M.H.: Managing platform architectures and manufacturing processes for nonassembled products. Journal of Product Innovation Management 19(4), 277–293 (2002)

Mikkola, J.H.: Capturing the Degree of Modularity Embedded in Product Architectures. Journal of Product Innovation Management 23(2), 132 (2006)

Mulenburg, G.: Standish Group International, Inc. (2007)

Muller, G., Hole, E.: The State-of-Practice of Systems Architecting: Where are we heading? Architecting Forum, Helsinki, Finland, October 4-5 (2005)

Muller, G., Hole, E.: Reference Architecture; Why, What and How. Architecting Forum, Hoboken, NJ, USA, March 12-13 (2007)

Muller, G., Hole, E.: Architecture Assessments; Needs and Experiences. Architecting Forum, Washington, DC, USA, April 14-15 (2009)

Murman, E.M., Allen, T., Bozdogen, K., Cutcher-Gershenfeld, J., McManus, H., Nightingale, D., et al.: Lean enterprise value: Insights from MIT's Lean Aerospace Initiative. JDAM, Paggrave (2002)

NASA, Key Enablers for Successful Programs in Aerospace. NASA Pilot Benchmarking Initiative, Washington, DC (2007)

Reinertsen, D.G.: The Principles of Product Development FLOW. Celeritas Publishing, Redondo Beach (2009)

Rosen, M.: Enterprise Architecture Trends 2007: The year ahead. Cutter Executive Report (September 2002)

Sethi, R., Pant, S., Sethi, A.: Web-Based Product Development Systems Integration and New Product Outcomes. Journal of Product Innovation Management 20, 37–56 (2003)

Shenhar, A.J., Dvir, D.: Reinventing Project Management. Harvard Business School Press, Boston (2007)