Proceedings of ICAD2013
The Seventh International Conference on Axiomatic Design
Worcester – June 27-28, 2013
ICAD-2013-26

ENHANCEMENT OF THE SYSTEMS ENGINEERING PROCESS IN THE LIFE CYCLE WITH AXIOMATIC DESIGN

David S. Cochran
cochrand@ipfw.edu
IPFW Center of Excellence in Systems Engineering
Indiana University Purdue University Fort Wayne
2101 East Coliseum Boulevard
Fort Wayne, IN 46805-1499

Jason A. Barnes
barnesj@ipfw.edu
IPFW Center of Excellence in Systems Engineering
Indiana University Purdue University Fort Wayne
2101 East Coliseum Boulevard
Fort Wayne, IN 46805-1499

ABSTRACT

This paper is about the use of Axiomatic Design to enhance the Systems Engineering Process during the product life cycle. The Systems Engineering Process must be enhanced to include the design of the enterprise that develops products since the enterprise design affects the efficacy of the process.

Keywords: Systems Engineering, product development, Collective System Design, Enterprise Design, performance measurement.

1 INTRODUCTION

This paper examines the traditional product life cycle based Systems Engineering (SE) Process as described by Blanchard and summarized by Cochran [Blanchard, 2008; Cochran, 2013].

The paper has four research objectives:

1. Describe limitations of the traditional SE Process and its implementation.
2. Provide examples that illustrate why the design of the enterprise affects the traditional product life cycle within the SE Process.
3. Demonstrate the inherent lack of definition in the SE Process that results in design parameters being interpreted as functional requirements.
4. Propose a method called Collective System Design which uses Axiomatic Design to enhance the traditional product life cycle SE Process.

2 PROBLEM STATEMENT

This section defines the problem statement relative to the traditional SE Process and its implementation. There are eight key deficiencies identified for the purposes of this study:

1. The design of the enterprise affects the efficacy of the SE Process. An enterprise system is the arrangement of components, materials, information, and people to produce a product or service that achieves the Functional Requirements (FRs) that state intended purpose (of the system) to meet customer needs; the work in that system is arranged according to flow to provide value to the customer called the value stream [Rother and Shook, 1998]. The Value Stream defines the system boundary. An Enterprise Design is the design of the enterprise system through the selection of FRs and the Design Parameters (DPs) to choose the FRs of the enterprise.

In contrast, the definition of a system in systems engineering does not specifically address the design of the enterprise, which we may call "Enterprise Engineering; [Cochran, 2009] instead, a *system* in systems engineering refers to the process for developing the capabilities of a product.

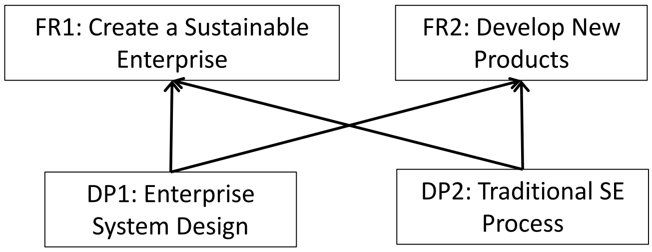

Figure 1. The Systems Engineering (SE) Process creates a coupled design.

For example, a coupled organization design can lead to a coupled product design. Sections 3.1 and 3.2 provide examples of this occurring in practice.

2. Requirements that are defined are often a mixture of functional requirements and design parameters [Cochran, 2009]. This means that the SE Process starts with requirements that may be actual solutions that are masked as requirements; the consequence may be to limits innovation and creativity. The opportunity is to add an up-front innovation process before defining FRs. In many cases, DoD contracts specify technical solutions as requirements that close the solution space before a contract is ever let.
3. Technical Performance Measures (TPMs) relate to requirements. Since requirements are a mixture of FRs and DPs (per item 2), TPMs are a mixture of performance characteristics placed on the FRs and attributes on the DPs. Also, TPMs apply at multiple blocks in the SE Process (see Figure 2).

Technical Performance Measures show how well a system is satisfying its requirements or meeting its goals. For the Joint Capabilities Integration and Development System (JCIDS), Key Performance Parameters (KPPs), "are attributes or characteristics of a system that are considered critical or essential to the development of an effective military capability" [Hagan, 2009]. This definition does not make a distinction between an FR or PS. The military contracting officer can consider a TPM to apply to a pre-conceived PS under the military procurement procedure.

Copyright © 2013 by ICAD2013

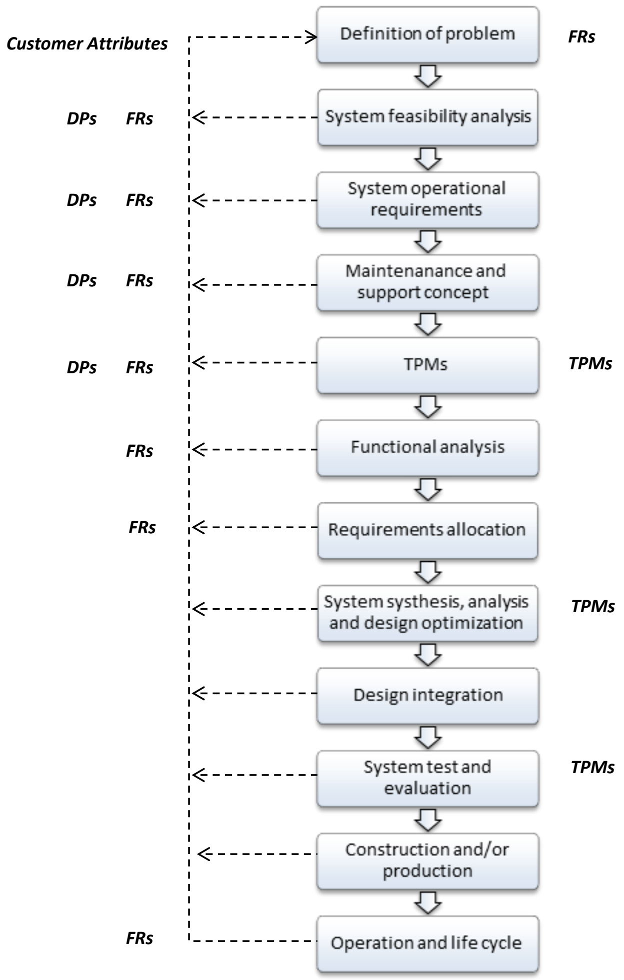

Figure 2. Technical Performance Measures (TPMs) apply to multiple blocks in the SE Process.

4. Requirements flow down and derived requirements eliminate an understanding of context of prior design decisions. When the design decisions of higher-level requirements are obscured, revisions to the design requirements only occur at the lowest level. Hence, engineers working to a set of requirements may not know the context of their work. Software tools like DOORS attempt to resolve this issue [IBM, 2008] but there frequently is no method to improve the design when there is no opportunity to change higher-level design decisions. An example of this occurring is provided in Section 3.2.

5. There is increased susceptibility to no-name agency requirements and "requirements soup." During the development of a system, requirements may be added late during system development and the requirements may not be traceable to the original source – thus, the identifier as a no-name requirement. Requirements soup occurs when every idea becomes a requirement, whether it is a solution or a functional requirement, doesn't have an identified level in functional decomposition or a priority in the implementation. For example, one result of the SE Process is the addition of "-ility" requirements at different times during the SE process, the impact on design level, sequence and implementation is unclear. An example of this occurring is provided in Section 3.2.

6. Testing is done at the end of the design process, ignoring the organization FR of not advancing a defect to the next operation (called Jidoka in Japanese). Often designs decisions are first tested when the first product is produced, leading to an expensive loop considered to be an integral component of the design process. An example of this occurring is provided in Section 3.2.

7. Operating scenarios on which requirements are based may not be well understood; therefore subsequent requirements definitions may be inadequate. For example, in an interview, a design engineer stated that he was working the design to a set of requirements that he had received in a requirements document. When asked if he understood the operating scenario for the product design that he was working on, he replied, "No, I don't." Furthermore, the product design itself had two other major product interfaces, both of which were also unknown to the designer. From this evaluation it is shown that the requirements documents and interface definitions in the SE Process assumed an understanding of the operating scenario and use of the product in the field. The SE documentation process does not ensure that "use-cases" are conveyed to the design engineers in their requirements documentation. To correct this deficiency, a front-end to enhanced systems engineering process using Axiomatic Design was developed using HP's use-case approach [Cochran and Wong, 2004].

8. Milestone checklists treat the SE Process as a recipe, not a design activity. Optimal or improved designs are often missed because the opportunity for innovation has been removed from the design process entirely. An example of the impact of this perception is provided in Section 3.2.

3 EXAMPLE CASES

This section presents two new case studies that illustrate the aforementioned deficiencies.

3.1 CASE STUDY OF ORGANIZATION A

The structure of the organization itself can determine whether a design is coupled or not (Deficiency 1). For example, the management program for a project is shown in Figure 3 as consisting of two FRs: Ensure successful development and Ensure successful integration. The organization at the highest level was split into a Development Branch and an Integration Branch.

Program management split the Development Branch by the type of contract issued; one contract was let for the vehicle program, while the other contract was let for affordable engine development. The impact of separating the development contracts showed up during program integration.

The Weight Management Office was responsible for ensuring that weight and thrust performance parameters were achieved. This office did not have direct contract responsibility for the vehicle and engine contractors. The two FRs of the Weight Management Office were to Ensure proper weight at launch and to Ensure proper center of gravity (CG) at launch (see Figure 4).

Copyright © 2013 by ICAD2013

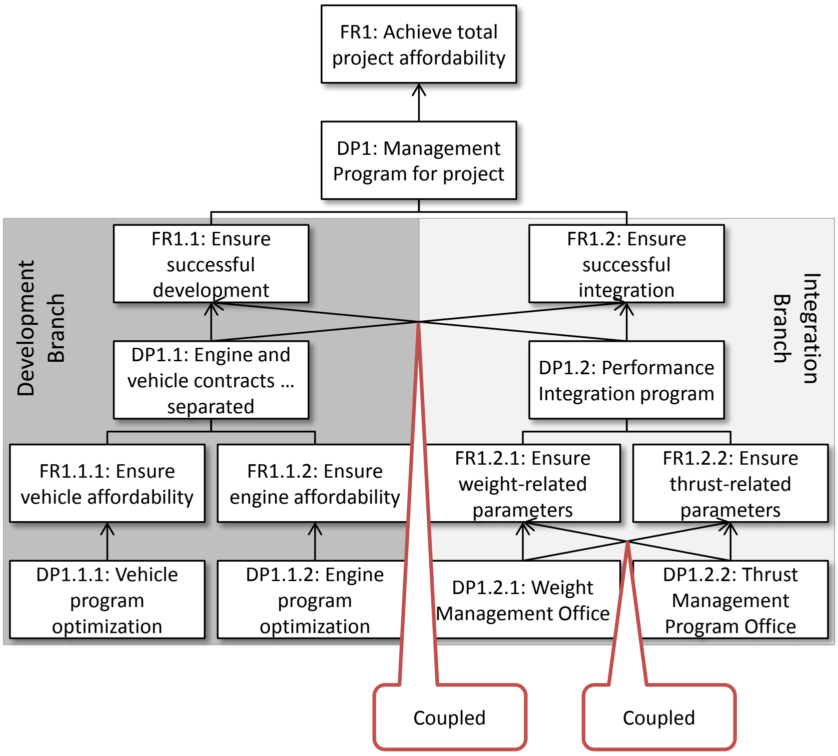

Figure 3. The Management Program of Organization A.

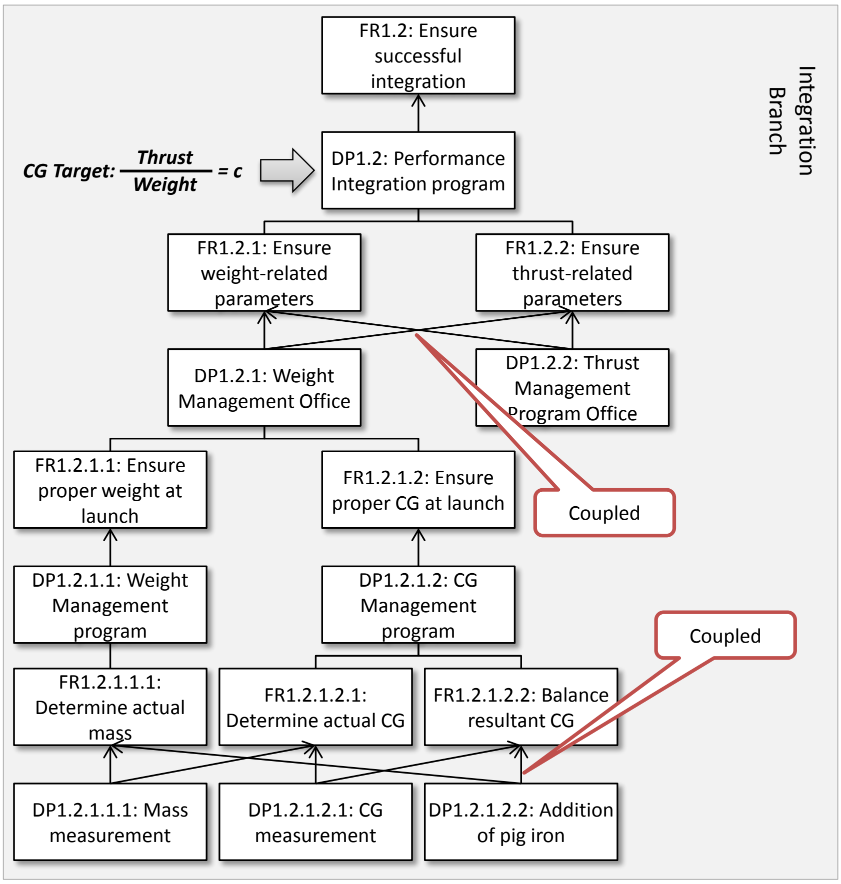

Figure 4. Expansion of the Integration Branch of Organization A.

Copyright © 2013 by ICAD2013

Figure 4 illustrates that the Weight Management Office did not realize that a problem with the organization design existed until there was an addition of pig iron to create the necessary CG for the vehicle. If Axiomatic Design had been used for the enterprise design, it would have been possible to realize that the selection of DP1.2.1.1 and DP1.2.1.2 resulted in a coupled organization design – one that would require a different approach to ensuring the achievement of weight and CG parameters at launch.

One possible solution is to redefine the decomposition of DP1: Management Program for project to just two performance FRs, with cost as a constraint: to make the new FR1.1: Achieve the CG target at launch, and FR1.2 Achieve the thrust target.

However, upon reflection, the real issue lies within the definition of the highest level FR, FR1: Achieve total project affordability.

Achieving total project affordability is **not** the same as achieving mission success, which means that a vehicle is designed, built, and operated successfully. The new high-level FR could be stated as: FR1(new): Achieve successful mission operating parameters with DP1(new): Program to Identify successful mission and vehicle operating parameters.

Once a vehicle is developed that successfully meets the required mission and operational parameters, a second vehicle development program could be launched to refine the design of the successful vehicle designed under FR1(new) to reduce cost. The second program level FR could then be stated as, FR2: Reduce cost of the successful vehicle design with DP2: Program to reduce life-cycle cost. Life-cycle cost is brought into the picture because a development vehicle design would not necessarily consider maintainability cost factors.

It is important to consider that the best DP to achieve the original FR1: Achieve total project affordability is DP1: No Vehicle. By not developing a vehicle there is no cost; the FR is achieved for the least cost.

3.2 CASE STUDY OF ORGANIZATION B

Organization B is a systems contractor that developed the organizational structure shown in Figure 5 to implement the SE Process. This study examines the effectiveness of the implementation and the organizational transformation that occurred over a period of five years due to the process outcomes.

In this implementation, the SE Process was divided into phases: System Definition, System Design, Functional Design, System Integration, and Production. During the first phase, System Definition, a business development team interfaced directly with the customer to determine scope and project risk to develop a set of top-level requirements which would be integrated into a contract. These contract requirements were handed off to Systems Engineers at the start of the System Design phase to be broken into a conceptual implementation, allocating the requirements derived from the top-level, contract requirements into subsystems or subassemblies. Meanwhile, the business development teams of Organization B would typically end their involvement with a project once the contract was approved, moving on to the next development opportunity.

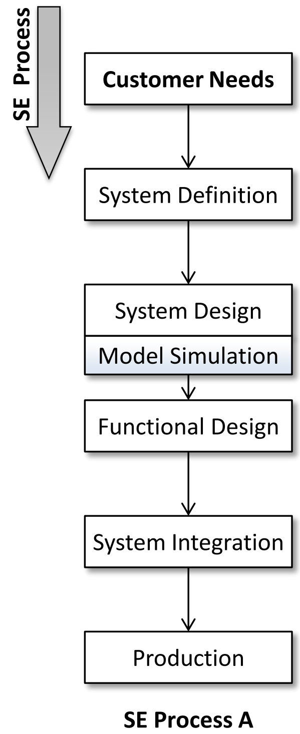

Figure 5. The implementation of Organization B's SE Process.

By design, Organization B removed the people with the best knowledge of customer need from the process as soon as the first phase was completed. This increased the difficulty of meeting the top-level FRs which are determined solely by customer need. In this regard, the implementation of the SE Process in Organization B reduced its effectiveness (Deficiency 1). Additionally, this immediately broke the flow of context in the system design. Systems Engineers in the System Design phase had no knowledge of customer need. Then when Systems Engineers supplied their derived requirements to designers in Functional Design, knowledge of the derivation process was similarly not communicated. Any changes to requirements in this phase were only done at this lowest level (Deficiency 4).

In the initial process design (SE Process A), Systems Engineers completed a Modeling & Simulation task as part of the System Design phase. There were four purposes of this task:

1. Assist with functional trade studies and design feasibility during System Design.
2. Provide reference artifacts for verification during Functional Design.
3. Troubleshoot test failures during System Integration and Production.
4. Provide baseline analyses for future applications and use-cases of the product family.

A typical product required a single Systems Engineer to spend 6-12 months developing the model. Because many products only allocated one or two Systems Engineers to the System Design phase, this development time showed up directly on the project budget and schedule. Deemed an unnecessary impact to the cost and delivery of its products by organization management, the SE Process was revised to

 Copyright © 2013 by ICAD2013

move Modeling & Simulation efforts in parallel with the Functional Design phase as in Figure 6.

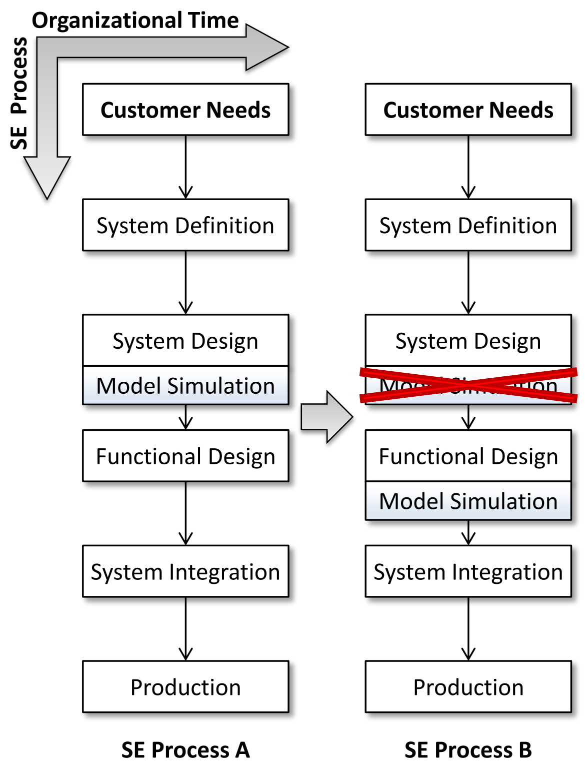

Figure 6. The first revision of Organization B's SE Process.

In this adapted process (SE Process B), the initial schedule seemed to show an improvement in delivery time by 6-12 months since Functional Design was starting earlier. However, because requirements developed during System Design might still be changed due to Modeling & Simulation outcomes, schedules for Functional Design often slipped by 6-12 months as Design Engineers incorporated or waited on varying requirements. In effect, the cost of one engineer became the cost of many engineers over that time with no difference in delivery time from the initial process. The constantly changing requirements during multiple phases of the SE Process also impacted the quality of the design (Deficiency 5).

Because of the obvious schedule slippages, Modeling & Simulation was viewed as a suboptimal verification tool for Functional Design and it was removed from the SE Process entirely as in Figure 7. As a result, all of the testing of the product design had been moved to the System Integration phase (SE Process C). It became typical for product designs to have errors that required passing results back to the System Design and Functional Design phases for iterated development.

By removing testing of the design during the System Design and Functional Design phases, the organization FR of not advancing a defect between phases was ignored (Deficiency 6). From an internal Six-Sigma Black Belt project, it was determined that 10% of the errors detected in System Integration or Production required a model to solve adequately. Without a model developed for the product earlier in the SE Process, Modeling & Simulation was done on a smaller scale when problems occurred, tailoring the model to the application in error. These models typically took 1-3 months to develop and halted workflow for all of the workers involved in the phase where the error was detected, either System Integration or Production. The Black Belt project estimated that having a pre-existing model would save about 6-12 months and $1 million in man-hours per project.

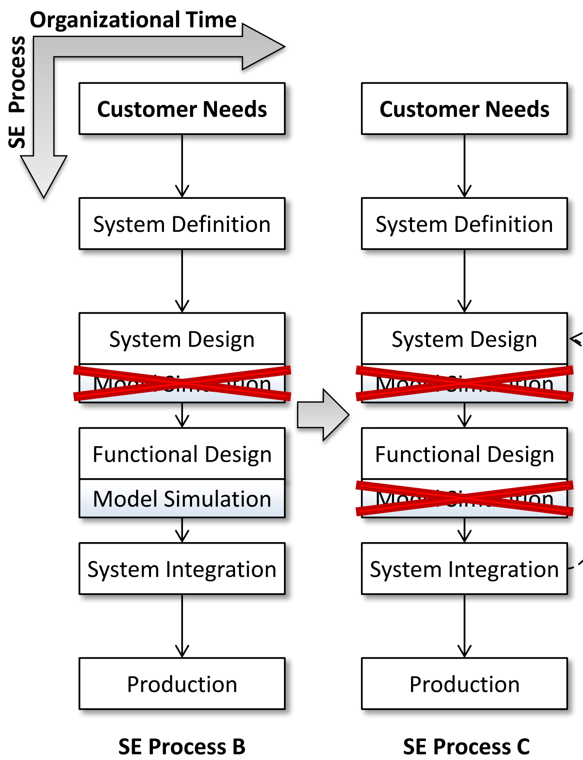

Figure 7. The second revision of Organization B's SE Process.

In addition to the added costs of not testing the design during each phase, the pressure of holding up System Integration or Production led to seemingly lower-quality models. These models also could only be applied to the very specific purpose of troubleshooting a single issue and could not be used for expanding the product family. This method of operation was allowed to continue because of the perception that the milestones present in the SE Process worked to vet the product design (Deficiency 8).

4 PROPOSED ENHANCEMENTS

This section describes how each limitation above can be resolved with a proposed method entitled Collective System Design, which is a combination of the SE Process, Enterprise Design, and Axiomatic Design. The section assumes a working knowledge of Axiomatic Design.

A key difficulty addressed by Collective System Design is lack of a shared purpose among the people involved in the development of a product or service and its delivery by a value stream. Management, engineering, production, finance, and other groups may have completely different viewpoints on how to meet customer needs. Thus, it is important to develop a shared mental model of the Enterprise Design *that*

Copyright © 2013 by ICAD2013

starts with the functional requirements of the enterprise. This can be accomplished using the Language for Collective System Design, a dialect of system relationships developed from Axiomatic Design (see Figure 8) [Cochran, 2010].

For Collective System Design, the term DP has been replaced by the term Physical Solution (PS) to convey the distinction between the functions of an organization and implementation in the form of physical solutions. To promote a mindset of learning, a PS is considered to be the best work method, known at the time, to achieve an FR. The result is that enterprise designers treat each PS as a hypothesis (H_0) to achieve each FR. The concept of work and physical implementation being a hypothesis was first proposed as part of the four rules of the Toyota Production System, in which it was stated that, "any improvement must be made in accordance with the scientific method, under the guidance of a teacher at the lowest possible level in the organization" [Spear and Bowen, 1999]. The design of an enterprise requires this same mindset that any proposed implementation requires the designers to realize that a physical solution is a proposed design choice that they think will achieve the FR. However, the proposed PS must be tied to achieving an enterprise FR that is both understood and agreed upon collectively by the people who are part of the design and do the work within the enterprise [Won *et al.*, 2001].

The Performance Measures of Collective System Design (M_{FR} and M_{PS}) implement metrics on the Enterprise Design to track how effectively the organization is achieving its functional requirements and its effectiveness of implementing its own physical design. While the SE Process uses TPMs that can be metrics on both requirements and design attributes, the separation of M_{FR} and M_{PS} reinforces the difference between FRs and PSs.

Collective System Design not only provides a language for obtaining agreement about enterprise requirements, but it also establishes an order of precedence in Enterprise Design. First the FRs must be defined by the group, then the PSs as proposed solutions for those FRs. Once that architecture is in place, measures on the design can be implemented.

For example of how to use this language, consider a Customer Need of traffic safety at a city intersection. The FR could be agreed upon as Safely regulate traffic. A suitable PS would then be a Traffic light, although that is not the only viable solution. Once that system is in place, the designers could agree on performance measures such as the number of accidents (a measure on the FR) or traffic light up-time (a measure on the PS.)

Figure 9 illustrates a learning loop to sustain an Enterprise Design. The enterprise system design is decomposed using the Axiomatic Design decomposition process, i.e. the language for Collective System Design. The result is the Enterprise Design (ED) Map, a hierarchy of FRs and PSs that determine the requirements of the enterprise and how the enterprise plans to achieve them.

Each Physical Solution (PS) is implemented to specify the content, sequence and timing of the work, also known as Standard Work. The Plan-Do-Check-Act (PDCA) learning loop is the method for implementing the Enterprise Design Map. A check of the physical work implementation leads to three options: (1) improving the Standard Work without modifying the PS; (2) creating a new PS and the new Standard Work; (3) deciding that the FR must be changed, which requires modifying the ED Map. In this way, the people in an organization practice the mindset that work is improvable and that the ED mapping can quantify enterprise purpose and actions necessary to achieve enterprise purpose.

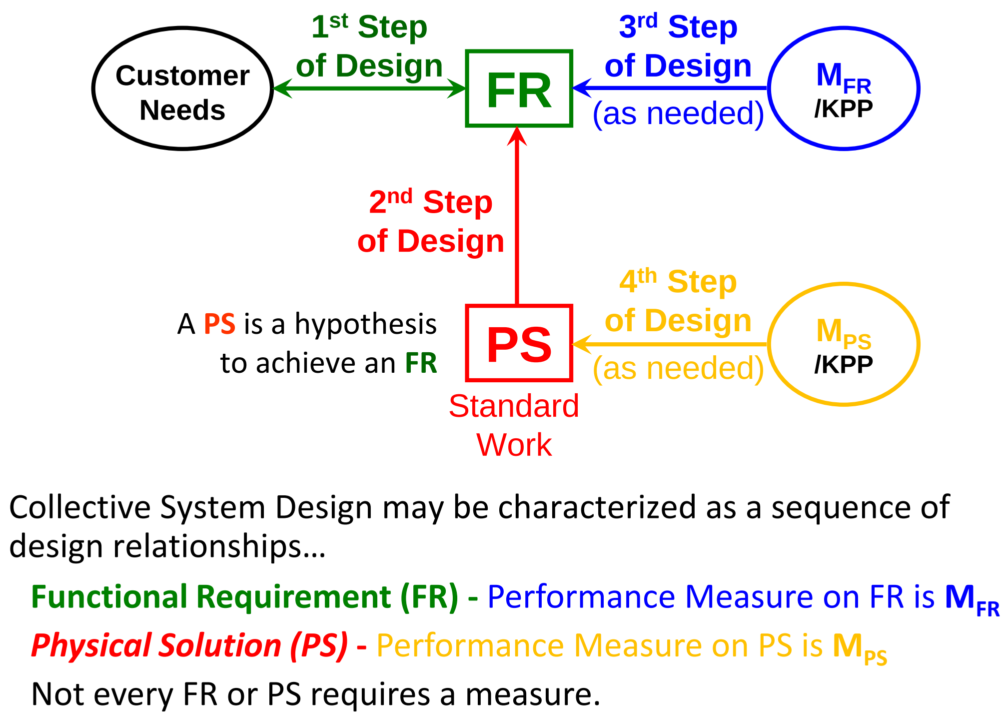

Figure 8. Language for Collective System Design.

Copyright © 2013 by ICAD2013

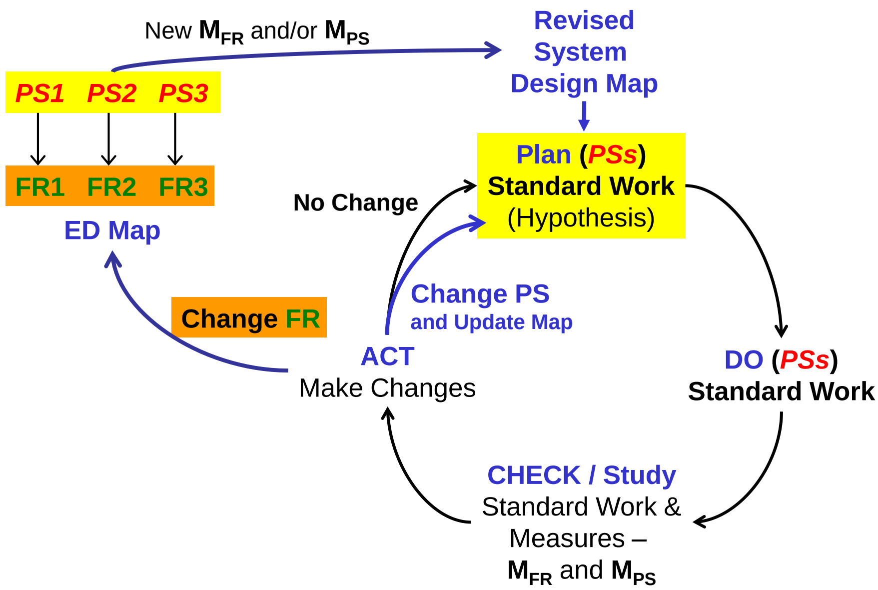

Figure 9. Learning loop to sustain Enterprise Design.

An example of this learning loop applied to the earlier traffic safety example would be that if the PS of the traffic light was deemed to not be effective, one of the following could be done: (1) change the timing on the light (changing the Standard Work); (2) replace the traffic light with stop signs (changing the PS); (3) change the FR of Safely regulate traffic to Prevent road intersections which may have its own PS such as a cloverleaf road design. As we can see, changing the FR changes the design of the enterprise system itself.

The seven FRs of the Manufacturing System Design for Stability (see Table 1) provide system design guidelines incorporating low cost, high quality, short lead time products with volume and mix flexibility [Won *et al.*, 2001; Cochran, 2012].

Table 1. The FRs of the Manufacturing System Design for Stability.

FR	Description
FR1	Provide a safe, clean, quiet, bright, ergonomically sound environment – **fundamental**
FR2	Produce the customer-consumed quantity every shift (time interval) – **from JIT**
FR3	Produce the customer-consumed mix every shift (time interval) – **from JIT**
FR4	Produce perfect-quality products to the customer every shift (time interval) – **from Jidoka**
FR5	Achieve FR2-FR4 in spite of operation variation – **robustness**
FR6	When a problem occurs in accomplishing FR2-FR4, rapidly identify the problem condition and respond in a pre-defined way – **controllability**
FR7	Produce product with the Least Time in System

Instead of applying the FRs to a manufacturing system, they can be modified to apply to Enterprise Design (see Table 2). In this context, the concept of customer is expanded to not just include the external consumer of the product but the internal entities that work together in the SE Process. For example, the Systems Engineers in the System Design phase of Organization B must treat Functional Design teams as a customer and produce design work that meets the seven FRs accordingly.

Table 2. The FRs of the Enterprise Design for Stability.

FR	Description
FR1	Provide a safe, clean, quiet, bright, ergonomically sound environment – **fundamental**
FR2	Produce the work as the customer needs it – **from JIT**
FR3	Produce what the customer wants – **from JIT**
FR4	Do not advance a defect to the customer of the work – **from Jidoka**
FR5	Achieve FR2-FR4 in spite of operation variation – **robustness**
FR6	When a problem occurs in accomplishing FR2-FR4, rapidly identify the problem condition and respond in a pre-defined way – **controllability**
FR7	Produce product with the Least Time in System

With these principles based in Axiomatic Design, there are proposed enhancements to deal with the identified deficiencies in the SE Process:

1. Most importantly, organizations need to be cognizant of their Enterprise Design and how it affects the SE Process. By incorporating the Language for Enterprise Design and a learning loop to sustain Enterprise Design with the FRs of the Enterprise Design for Stability, an organization can focus on implementing an SE Process that can serve its purpose. An effective Enterprise Design can eliminate no-name agency requirements, ensure that the operating scenario is defined effectively, eliminate milestone checklists and reviews that are

Copyright © 2013 by ICAD2013

done robotically and typically don't accomplish anything, and ensure the ability to improve the design process.

2. Use of Axiomatic Design distinguishes and separates the DP from the FR, ensuring that high-level requirements are functional requirements and are not pre-conceived solutions. Furthermore, by employing Collective System Design, the design team must gain agreement on the FRs before determining the DPs/PSs to achieve them. This practice also promotes innovation at the front end of product development by fleshing out the requirements for the entire system design while keeping them separate from design choices.

3. Use of Axiomatic Design identifies the FRs, allowing TPMs to be directly tied to FRs as measures, called FR_M. FRs to address each "-ility" can be placed visibly on a design board with Post-It notes for consideration at each level of the design decomposition.

4. The decomposition process in Axiomatic Design ensures that the DP is identified prior to moving to the next lower level of decomposition. To promote design clarity and improvability, a decomposition hierarchy should be tied in with a learning loop to enable and encourage a design improvement cycle.

5. By defining the system boundary of a development program, the agencies that affect the design are identified at the beginning of a design activity. Each agency must be brought in to the same room at the early stages of design and must identify the functional requirements and constraints they place on the design. Similarly, the enterprise must develop the hierarchy of FRs as the first step of the SE Process. This is the DP necessary for meeting FR3 of the Enterprise Design.

6. To achieve FR4 of the Enterprise Design, testing must be integrated into each stage of the SE Process. This testing should include checks at each layer of requirements decomposition, ensuring that the selected DPs are viable, uncoupled solutions.

7. By involving the designer in a physical model built to show how work will be done between the end-user and the product, operation scenarios can be demonstrated. The designer should also be allowed to discuss customer need and use scenarios with the customer.

8. Milestone reviews should monitor the TPMs of the design in reference to the FRs instead of being a predefined checklist. This would ensure that the system design is tracking the customer needs and would vary from product to product, reducing the likelihood that designs are created robotically.

5 CONCLUSION

The SE Process is necessary but not sufficient. Engineering is not checking the box on milestone checklists that were established by people who are external to a development enterprise. A milestone checklist does not convey the Enterprise Design FR, only "artifacts," proposed solutions of a design. When an agency or person mandates solutions without clarifying the FRs, the thinking leaves the room. People and organizations become robotic, checking a box for the sake of checking it (or to get paid). Leaders must get the FRs on the table within an organization before jumping to the implementation (the how-to's).

As engineers practice SE, we have the opportunity to get the FRs on the table and collaboratively agree on the best solutions understood by the designers at the time. The axiomatic decomposition framework enables requirements traceability and conveys an easy to understand visible model of the thinking process and design decisions that a designer makes when doing design.

The use of axiomatic design enables us to know the **why** (the functional requirements) before choosing the **how** (the design parameters or physical solution) of the design.

6 ACKNOWLEDGEMENTS

Thank you to Ma Oo, IPFW graduate student in the Spring 2013 semester of SE 53000 – Systems Engineering Management, for developing Figure 2 illustrating the FR/DP relationships to the SE Process.

7 REFERENCES

[1] Blanchard B., *System Engineering Management*, Hoboken, New Jersey: John Wiley & Sons, Inc., pp. 51-120, 2008.

[2] Cochran D.S., Wong D., "Integration of Collective System Design (CSD) and Hewlett Packard's Value Chain Requirements Management (VCRM)", Hewlett-Packard, 2004.

[3] Cochran D.S., "Enterprise Engineering, Creating Sustainable Systems with Collective System Design – Part I", *The Journal of RMS (Reliability, Maintainability, Supportability) in Systems Engineering*, Winter Journal, 2009.

[4] Cochran D.S., "Enterprise Engineering, Creating Sustainable Systems with Collective System Design – Part II", *The Journal of RMS (Reliability, Maintainability, Supportability) in Systems Engineering*, Spring Journal, 2010.

[5] Cochran D.S., "The Challenges of Advanced Manufacturing System Design and Sustainability", Blacksburg, Virgina, Nov. 13, 2012.

[6] Cochran D.S., Lecture Notes, *SE 53000 – Systems Engineering Management*, Indiana University-Purdue University Fort Wayne, Spring, 2013.

[7] Hagan G., *Glossary of Defense Acquisition Acronymns and Terms*, Fort Belvoir, Virginia: Defense Acquisition University Press, 2009.

[8] IBM Rational Software, *Telelogic DOORS: Getting Started with Telelogic DOORS*, IBM Corporation, 2008.

[9] Rother M., Shook J., *Learning to See: Value Stream Mapping to Add Value and Eliminate Muda*, Cambridge, Massachusetts: Lean Enterprise Institute, 1998.

[10] Spear S., Bowen H.K., "Decoding the DNA of the Toyota Production System", *Harvard Business Review*, Sept.-Oct., pp. 97-106, 1999.

[11] Won J., Cochran D.S., Johnson H.T., Bouzekouk S., Masha B., "Rationalizing the Design of the Toyota Production System: A Comparison of Two Approaches", *International CIRP Design Seminar Proceedings*, Stockholm, Sweden, June, 2001.

Copyright © 2013 by ICAD2013