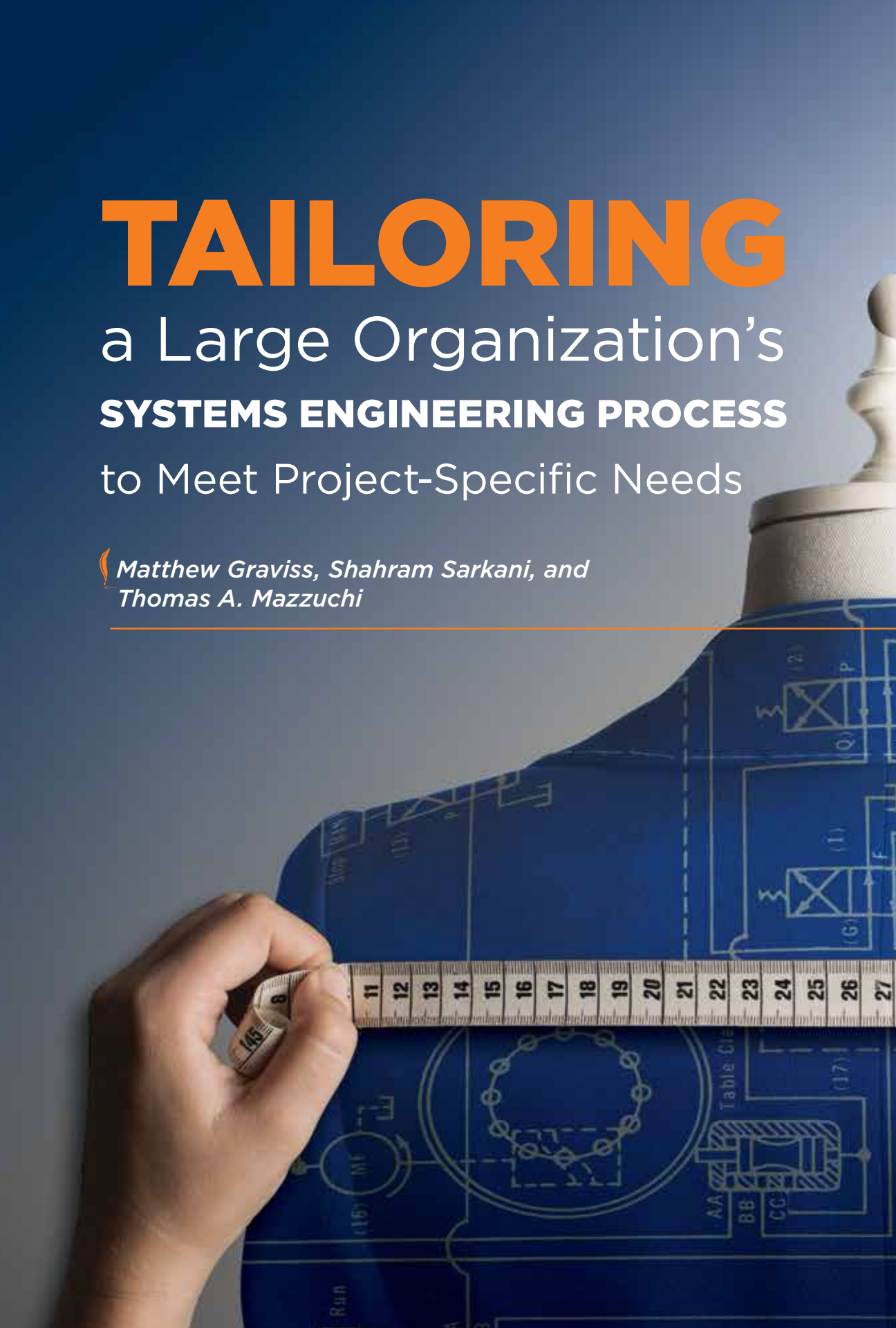
TAILORING a Large Organization's SYSTEMS ENGINEERING PROCESS to Meet Project-Specific Needs

Matthew Graviss, Shahram Sarkani, and Thomas A. Mazzuchi



Systems engineers are faced with the difficult challenge of adhering to broad systems engineering (SE) policies, while simultaneously tailoring SE processes to meet the unique challenges facing their projects. Tailoring is often performed in an ad hoc manner. Determining which stages, steps, and artifacts of the process are necessary can be time-consuming and challenging. SE guidebooks across industry and government organizations often stress the importance of tailoring, yet offer little practical guidance on how to perform the function. This article proposes a model for automating the SE tailoring process through the definition of an organizational rule set and a minimal set of project-specific inputs. The model is then analyzed through several case studies within the Department of Homeland Security to evaluate the proposed approach.

Keywords: *systems engineering, process tailoring, systems acquisition, method engineering, tailoring considerations*

lead image by Diane Fleischer

Most organizations develop their own systems engineering (SE) guidelines to specify the stages, reviews, and artifacts that should be executed on projects. Standard processes are important for today's projects, as they employ standardized terminology and may prevent project teams from reinventing the wheel; however, formal and informal processes must be balanced to yield the efficiencies gained through standardization and the effectiveness gained by adaptability (Laufer, 2001, p. 27). Since no two projects are identical, no two processes are either (Humphrey, 1989, p. 241). Organizations must develop tailorable guidelines that are broad enough to span the range of projects within their portfolio (Hwang & Park, 2006, p. 37). This places the burden of tailoring the organization's SE guideline on project systems engineers. This article proposes a rule-based approach to partially automating this activity, thereby reducing the manual burden yet still offering a significant number of custom SE process variations to best meet a project's needs.

Process Tailoring

Process tailoring refers to the modification of a standardized process to meet the unique needs of a project (Xu & Ramesh, 2007, p. 293). This can include determining which stages and reviews are necessary and should be executed multiple times, which artifacts should be produced, or even what content within an artifact is necessary. Successful tailoring results in modified processes that achieve the goals of the standard process model (International Organization for Standardization/International Electrotechnical Commission/Institute of Electrical and Electronics Engineers [ISO/IEC/IEEE] 15288, 2015, p. 86). Project teams often perform tailoring in an ad hoc manner without guidelines or rules (Pedreira, Piattini, Luaces, & Brisaboa, 2007, p. 1), and usually complete the activity based on declarative memory (Xu & Ramesh, 2009, p. 282). Process implementation directly affects project budget and schedule, as well as product quality. Choosing and adapting

the right design method is critical to successfully executing SE concepts and principles (Verma & Fabrycky, 1997, p. 592). Systems engineers are thus faced with the difficult challenges of adhering to broad policies.

While research related to process tailoring in software engineering is prevalent, similar research in SE is limited. Existing SE guidelines and handbooks stress the importance of tailoring SE processes to meet the unique needs of a project, but provide little guidance on process tailoring (Browning, Fricke, & Negele, 2006, p. 119; Pereira et al., 2007, p. 72). Fortune and Valerdi (2013) propose a framework for reusing SE products within an organization, and highlight key considerations for process reuse. Among them, Fortune and Valerdi (2013, p. 310) argue that a robust knowledge management system is critical to the success of process reuse. They also note the reuse of a product at too high a level may not apply to the specific environment of the new project. On the other hand, reusing products at too low a level can cause tailoring of enough significance that the effort saved by process reuse may be lost (Fortune & Valerdi, 2013, p. 305).

Process tailoring can be considered as a form of standard process reuse (Yoon, Min, & Bae, 2001, p. 202); however, tailoring involves process application to a unique project environment, as opposed to the flexibility of the standard process itself. Particularly in the field of software engineering, patterns have been explored as a viable option for process reuse; in fact, Cloutier and Verma (2007) extend the concept to a method for developing pattern forms in systems architecture. In addition, Hagge and Lappe (2005, p. 24) emphasize the need for capturing knowledge that can be reused in future projects. They propose four introductory patterns for requirements engineering and discuss a method for sharing observations across projects (Hagge & Lappe, 2005).

Method Engineering

Extending knowledge from the software process domain to the SE process domain has been ongoing for quite some time (Boehm, 2006, p. 5). Research in software process tailoring is well-documented and can be extended to SE. Henderson-Sellers, Ralyté, Ågerfalk, and Rossi (2014) provide an extensive literature review of the application of Method Engineering (ME) to specific situations in software development. Their review spans the various approaches to method tailoring and case studies in the application of method tailoring. Kuhrmann, Méndez Fernández, and Tiessler (2014) also conducted a thorough literature review of ME, which pertains to the process

of constructing methods for software development. They highlight 83 articles contributing to the research of software process tailoring, and conclude that "strong empirical evidence of the feasibility of ME" is still lacking.

Software Process Tailoring Approaches

A wide variety of approaches to software process tailoring have been documented in literature. The case retrieval-based method formulates a tailored software process based on retrieving a tailored process from a library of historical projects (Ahn, Ahn, & Park, 2003; Park & Bae, 2013). Retrieval is performed by comparing values of domain factors of a new project to those of past projects in the library. Two primary challenges exist with the case retrieval approach. First, the extensiveness of the library of previous methods, which requires a mature organization, limits the accuracy of the case retrieval approach. And second, the organization must have an established knowledge management system that not only archives previous projects, but also tags the projects with the values and domain factors required for retrieval.

In the preformed approach, an organization develops a standard set of tailored processes that apply to particular categories of projects (Plogert, 1996). The challenge with this approach is that it only includes the first level of tailoring, such as hardware or software, and does not address the significant variability that can exist from project to project within an organization. In addition, this approach does not easily allow for growth; to modify the approach based on lessons learned would require revisiting the pretailored processes altogether.

Hausen (1998) employs a rule-based approach to software process tailoring. The rule-based approach identifies values of factors that address the rationale for tailoring based on specific project characteristics. An engineer creates each rule such that for a specific value of a domain factor that exists for a new project, certain process elements are tailored (Kang, Song, Park, Bae, Kim, & Lee, 2008, p. 54). Kang et al. (2008) goes further by combining the rule-based approach with case retrieval where rules are used to retrieve the case most similar to the attributes of the new project.

Jaufman and Münch (2005, pp. 328–342) used both top-down and bottom-up approaches to software process tailoring, allowing for increased efficiency and process adherence, respectively. This approach leverages both the preformed and rule-based methodologies.

Hanssen, Westerheim, and Bjørnson (2005) use brainstorming to tailor a software process in an organization. Baldassarre, Caivano, Visaggio, and Visaggio (2002) propose the use of patterns for software process tailoring.

As previously stated, a significant amount of research exists in process tailoring for the software engineering field. While similar research exists in SE process tailoring, the opportunity exists to apply software engineering research to SE.

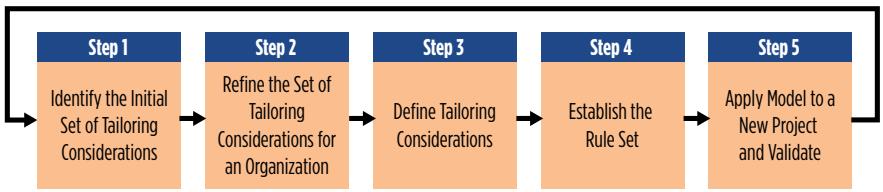
By proposing a model that simplifies SE process tailoring, this article seeks to bridge the gap in SE literature between SE process research that stresses the importance of tailoring yet offers little guidance on how to perform the function, and the practitioners who have to perform SE process tailoring to meet the specific needs of their projects. The proposed rule-based approach to SE process tailoring offers a significant number of custom SE process variations as compared to the preformed approach and without the reliance on a deep repository required of a case-retrieval approach. With this proposed approach, the number of tailoring decisions could be reduced from the total number of elements in a governing SE process to the number of rules established within an SE organization and applied to the project.

The next section of this article proposes the SE Process Tailoring Model (SEPTM), developed starting with tailoring considerations identified in the *INCOSE Systems Engineering Handbook*, published by the International Council on Systems Engineering (INCOSE, 2015), and that can be used to generate a project-specific SE process based on a project's unique characteristics and environment. The discussion on the SEPTM is then followed by a section that introduces the analysis of 24 case studies within the Department of Homeland Security (DHS), thereby evaluating the model against real world SE process tailoring instances. Finally, the article highlights conclusions of the research and areas of future work.

Systems Engineering Process Tailoring Model (SEPTM)

Figure 1 presents an overview of the proposed SEPTM. At a high level, the authors constructed the model based on an analysis of the generic SE process described in the *INCOSE Systems Engineering Handbook* that influences process tailoring. This analysis forms the basis of an initial set of Tailoring Considerations (TC). From the initial set of TCs, a subset is then selected that is applicable to a particular organization. A rule set is developed that links the TCs and the organizational SE process. A systems engineer can then apply the rule set to the unique attributes of a project to generate a customized SE process based on the conditions of a particular project. The following subsections further describe each of these activities.

FIGURE 1. OVERVIEW OF THE SYSTEMS ENGINEERING PROCESS TAILORING MODEL (SEPTM)



Identify the Initial Set of Tailoring Considerations

An important aspect of the model is determining the TCs used to identify critical aspects of a project. The TCs will broadly inform what SE activities are needed for a given project. The development of the initial organizational TC list is based on an analysis of the general SE process described in the *INCOSE Systems Engineering Handbook* and its discussion of considerations and enablers for process tailoring (INCOSE, 2015). The handbook states the appropriate approach will vary based on the unique needs of a project. It also describes various life-cycle approaches, including the waterfall, incremental, and agile approaches. In the waterfall approach, the project team performs SE stages serially; in the incremental approach, the project team performs SE stages serially multiple times; and in the agile approach, the project team iterates through short development cycles with frequent product releases (Aoyama, 1998). As part of the agile development cycles, project teams perform the SE stages of requirements definition, design, development, and testing in an iterative fashion. Moreover, project teams perform and update the associated reviews and documentation with

each release (Cao & Ramesh, 2008, p. 63). Each of these approaches varies in terms of stage and review frequency, as well as the amount of documentation required. As a result, a project's life-cycle approach must be a critical TC.

The *INCOSE Systems Engineering Handbook* also describes cross-cutting technical methods and includes a section on specialty engineering analyses that must be considered by a project's systems engineer; however, not all of the analyses will apply to every project (INCOSE, 2015). For example, interoperability should be considered for projects that interface with other projects. An organization may have specific artifacts that are required to address interoperability for certain projects or allow for deleting interoperability artifacts based on the project type.

The handbook's tailoring section includes a discussion of SE process tailoring and identifies considerations and enablers for tailoring: project scope, risk tolerance, complexity and precedence of the system, and organizational/enterprise policies and infrastructure (INCOSE, 2015). While the handbook identifies risk tolerance as a factor in process tailoring, it does advise of the importance of placing tailoring emphasis on the system and the project objectives, or project scope.

Analysis of the general SE process and tailoring discussion in the *INCOSE Systems Engineering Handbook* resulted in the following list of TCs:

TAILORING CONSIDERATIONS	
Life-cycle Approach	Logistics Engineering Analysis
Project Scope	Manufacturing and Producibility Analysis
Complexity and Precedence of the System	Mass Properties Engineering Analysis
Organizational/Enterprise Policies and Infrastructure	Reliability, Availability, and Maintainability
Modeling, Simulation, and Prototyping	Resilience Engineering
Affordability/Cost Effectiveness/ Life-Cycle Cost Analysis	System Safety Analysis
Electromagnetic Compatibility Analysis	Systems Security Engineering
Environmental Engineering/Impact Analysis	Training Needs Analysis
Interoperability Analysis	Usability Analysis/Human Systems Integration
	Value Engineering

Of the 19 TCs, the first four—Life-cycle Approach, Project Scope, Complexity and Precedence of the System, and Organizational/Enterprise Policies and Infrastructure—apply to all organizations, as the *INCOSE Systems Engineering Handbook* suggests. The remaining 15 vary in applicability, depending on the types of projects an organization executes. This requires an organization to determine the appropriate subset of TCs for their project portfolio.

Refine the Set of Tailoring Considerations for an Organization

In the second analytical step, the overarching TC list is reduced to one that includes only the TCs relevant for that organization. This step is performed by comparing the TC list to organizational policies and the organizational SE process. For each TC, possible values are identified. The TCs can have values that are either yes/no (e.g., environmental impacts need to be considered or not) or multiple (e.g., life-cycle approach may be waterfall, incremental, or agile). The case study in the next section of this article offers a more detailed discussion of how this step is performed.

Define Tailoring Conditions

Before a tailoring rule set can be developed, some tailoring operations must be defined. For the purposes of this research, the following definitions apply:

- "Standard" is used when a stage, review, or artifact is to be developed consistent with the scope of the activity defined in the organizational SE policy.
- "Tailored" is used when a stage, review, or artifact is modified from the scope in the organizational SE policy. This could include combining two or more stages, reviews, or artifacts into one. This could also include repeating stages or reviews multiple times within a project. For example, the process tailoring model recommends that the Critical Design Review occur multiple times in a project that uses incremental development methodology.
- "Deleted" is used when the stage, review, or artifact is not needed and removed from the project's tailored SE process altogether. An example of this is the removal of a System Design Document for a Commercial Off-The-Shelf (COTS) acquisition.

Establishing the Rule Set

Once the TCs have been selected for an organization and tailoring conditions defined, a rule set must be generated to link the TCs and associated values to the elements of the organization's standard SE process. Process elements are those SE tasks and activities identified within the organization's SE process. Table 1 shows the matrix used to establish the rule set. In area 1 of the table, all of the elements (1 to M) of the standard SE process are listed vertically. For example, the DHS SE process has 107 process elements ("M"). In area 2, each TC (1 to N) and associated values (1 to O) are listed horizontally. "N" represents the number of TCs identified for the organization, and the values represent potential attributes of a project within the organization. As an example, if TC 1 is Interoperability Analysis, the potential values would be "yes" and "no" given that a system may be stand-alone or connected. For each intersection of a process element and a TC-value combination in area 3, a tailoring rule is established. This requires two levels of decisions. First, a determination is made as to whether a TC is relevant or not for a particular process element; and second, if so, a determination is made for each value of the TC whether the process element should be kept standard, tailored, or deleted. Thus, each relevant intersection in area 3 will contain either "S" for standard, "T" for tailored, or "D" for deleted. In summary, the resulting rule set is a list of TCs, for which each value drives a tailoring decision (standard, tailored, or deleted) for every relevant process element within the organizational SE process.

TABLE 1. ESTABLISHING THE RULE SET

Tailoring Considerations →	**TC 1**				**TC 2**				**...**	**TC N**			
2 Values →	Value 1	Value 2	...	Value O	Value 1	Value 2	...	Value O	...	Value 1	Value 2	...	Value O
Standard SE Process Elements													
Process Element #1													
1 ...							3						
Process Element #M													

Apply Model to a New Project and Validate

Once the model has been developed for an organization, project systems engineers can use it for tailoring SE processes to meet the needs of a specific project. To do so, a systems engineer must have a solid understanding of the acquisition strategy and the project environment. This is true whether applying a rule-based methodology as is the case here, or tailoring an SE process in an ad hoc fashion. To exercise the model, a systems engineer must align characteristics of the project to a value associated with each TC. Once this is complete, the model outputs the tailored SE process, identifying which of the full set of the organization's SE process elements should be executed as standard, which should be tailored, and which can be deleted. After the model has been applied to a real-world situation, the organization must incorporate lessons learned back into the model by reassessing the organization's list of TCs and the organization's rule set. Moreover, as SE research is constantly refining best practices and industry standards are updated, the organization must also revisit the initial TC set to assess changes from industry.

Case Study

The objective of this research is to use project attributes and a rule-based approach to reduce the amount of manual tailoring of the SE life cycle. Henderson-Sellers et al. (2014, p. 169) note that determining the best project methodology requires specific tailoring to a project's unique situation. Measuring the benefit of such an activity can be based on the amount of effort required to tailor standard processes (Xu & Ramesh, 2007). For purposes of this analysis, this article extends that definition to the number of manual decisions required to produce a project-specific SE process. By design, this model fixes the number of decisions on the number of TCs established for an organization. If an organization applies all of the initial 19 TCs, the number of manual decisions will be 19. The DHS, an organization with a large acquisition portfolio, has an SE process consisting of 107 process elements, resulting in 107 associated process tailoring decisions (DHS, 2010). Provided the model can reproduce the tailoring decisions made within the DHS case studies, the potential benefit of this proposed approach to SE process tailoring can be demonstrated by reducing manual decisions from 107 to 10, which as described in the following subsections, is the number of TCs determined for the DHS case study. As such, the following subsections focus on the accuracy of the model in comparison to case studies.

> ***The objective of this research is to use project attributes and a rule-based approach to reduce the amount of manual tailoring of the SE life cycle.***

Case Study Data

To perform the case study analysis, the authors required two key elements. First, an organization's SE policy was required to serve as the process baseline. The baseline was used to establish the linkages between the overarching TCs and the process elements contained within the baseline. The data source for this research was the *DHS Systems Engineering Lifecycle Process Guide* (DHS, 2010). The DHS SE process identifies nine SE stages, 11 technical reviews, and 87 artifacts to be executed through the life of a project, for a total of 107 SE process elements.

The second key element required is a set of tailored SE processes implemented on projects within the organization. The authors selected 24 DHS projects for analysis based on data availability and variability across the TCs (Table 2). For example, some projects executed full research and development, while others were COTS acquisitions; some executed a waterfall development approach, and others, incremental or agile. The authors gathered and analyzed quantitative data regarding tailoring outcomes from each of the 24 projects and compared actual and modeled tailoring outcomes.

TABLE 2. CASE STUDY PROJECT VARIABILITY

Case Study Tailoring Considerations	Possible Values	# Projects
Security Impact	Impact/No Impact	17/7
Privacy Impact	Impact/No Impact	9/15
Intelligence Impact	Impact/No Impact	3/21
Interoperability Impact	Impact/No Impact	11/13
Accessibility Impact	Impact/No Impact	13/11
Technology Demonstration Planned	Planned/Not Planned	7/17
Environmental Impact	Impact/No Impact	7/17
Development Methodology	Waterfall/ Incremental/Agile	11/11/2
Development Type	Development/COTS	19/6
Project Size	< $300M, ≥ $300M	6/18

Selection of Organizational Tailoring Considerations

The first step in applying the tailoring model to an organization is to determine the subset of the 19 overarching TCs that are applicable to the organization. As stated previously, the first four TCs apply to all organizations and were adapted for the DHS TC list. Analysis of the DHS organizational/enterprise policies yielded additional considerations pertaining to privacy and intelligence. Additional analysis of the DHS SE process was performed to determine which of the remaining 15 are applicable. Of the 15, five were applicable to all projects per the DHS SE process, five were not discussed in the DHS SE process guide, and five may apply depending on the needs of the specific project. The five TCs that may be applicable were included as organizational TCs in the model. As a result, the 10 organizational TCs identified and applied for this research are as follows:

1. Security Impact
2. Privacy Impact (Organizational/Enterprise Policies and Infrastructure)
3. Intelligence Impact
4. Interoperability Impact
5. Accessibility Impact (Usability)
6. Technology Demonstration (Modeling, Simulation, and Prototyping)
7. Environmental Impact
8. Development Methodology (Life-cycle Approach)
9. Development Type (Complexity and Precedence of the System)
10. Project Size (Project Scope)

With the 10 organizational TCs in place, the rule set for DHS was established by linking the TCs to the standard DHS SE process as depicted in Table 3. Note that for any given TC, only a subset of process elements is relevant; that is, empty table entries reflect that there is no relevance between that TC and the intersecting standard SE process element. For example, in Table 3, four TCs (2, 7, 9, and 10) are relevant to process element No. 1.

TABLE 3. ESTABLISHING THE RULE SET FOR THE DHS CASE STUDY

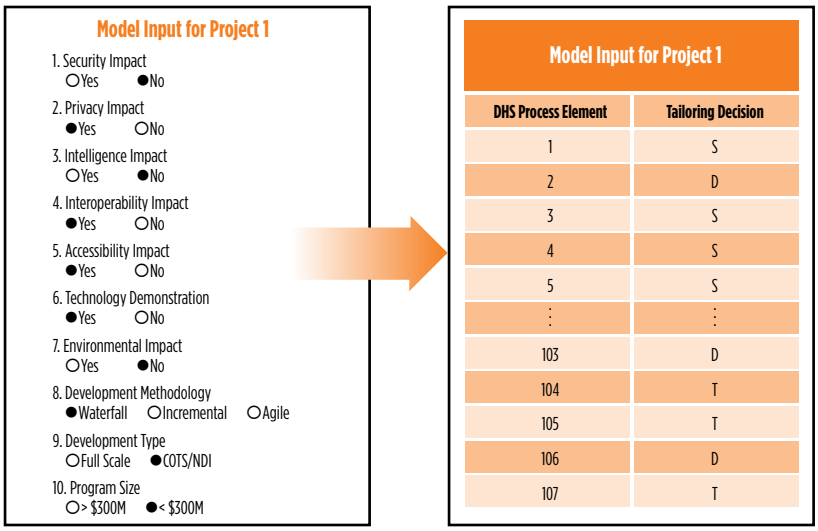
	TC 1		TC 2		TC 3		TC 4		TC 5		TC 6		TC 7		TC 8			TC 9		TC 10	
Tailoring Considerations →	Security Impact		Privacy Impact		Intelligence Impact		Interoperability Impact		Accessibility Consideration		Technology Demonstration		Environmental Impact		Development Methodology			Development Type		Project Size	
Values →	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	W	I	A	D	C	≥ $300M	< $300M
DHS Standard SE Process Elements																					
Process Element #1			S	D									S	D				S	T	S	T
⋮																					
Process Element #107			S	D			S	D	S	D			S	D						S	T

Note. Values: Y=Yes, N=No, W=Waterfall, I=Incremental, A=Agile, D=Developmental, C=COTS
Entries: S=Standard, T=Tailored, D=Deleted

Case Study Projects

With the rule set established for DHS as described previously, the next step was to input the attributes of each of the 24 projects into the model to generate a custom SE process for each project. Figure 2 depicts an example project's input and output, highlighting that the input is addressing the 10 TCs based on project characteristics, and the output is a tailoring decision for each of the 107 elements within the DHS SE process.

FIGURE 2. EXAMPLE MODEL INPUT AND OUTPUT FOR A PROJECT



Model Input for Project 1

DHS Process Element	Tailoring Decision
1	S
2	D
3	S
4	S
5	S
⋮	⋮
103	D
104	T
105	T
106	D
107	T

Note. COTS/NDI = Commercial Off-The-Shelf/Nondevelopmental Item.

Three questions formed the analysis of each TC. Table 4 lists those questions and provides a corresponding example using the Security TC. The authors assessed each of the 10 TCs by comparing the model output for each project to the data from each project's SE process tailoring plan. A default process was established based on default project attributes within the rule set. Projects that had an attribute different from the default were compared to the model for that particular attribute. For example, the default project attribute for the Security TC is "No"; therefore, the 14 projects that do have security impacts were compared to the model to assess whether the projects performed the process elements relevant to that TC.

TABLE 4. CASE STUDY ANALYSIS QUESTIONS	
Analysis Question	**Security TC Example**
For a given pairing of process element and tailoring consideration, does the tailoring decision generated by the model for a project match the tailoring decision actually executed on the project?	For a project with security impacts, did the model and project data both include a plan to execute the disaster recovery plan process element?
For the given pairing of process element and tailoring consideration, what percentage of matches between the project data and their corresponding model outputs exists?	For the 14 projects that do have security impacts, what percentage of the projects included a plan to execute the disaster recovery plan process element as the model suggests?
What percentage of process elements have at least 75% matching between model output and actual project data?	What percentage of the 15 process elements relevant to the Security TC have a 75% match between model output and actual project data?

Table 5 summarizes the case study analysis approach using the Security TC as an example. Questions listed in Table 4 trace to Table 5 as "Q1", "Q2", and "Q3". Column 1 represents the process elements relevant to the Security TC. Column 2 shows the tailoring decision of the model for the Security TC and relevant process elements; the next set of columns shows the pertinent data from the 24 projects, and an assessment of whether or not each matches the model (Q1). The last column in the table calculates the percentage of projects that match the model for each process element (Q2). Consistent with validation studies of software engineering practices, a 75 percent acceptable matching level is used (Daneva & Ahituv, 2010, p. 282; Krishnan & Kellner, 1999, p. 806; Ramasubbu, Krishnan, & Kompalli, 2005, p. 83). Thus, for a TC to be considered valid, the model output for each relevant process element must match at least 75 percent of the relevant projects (last column in Table 5), and at least 75 percent of the process elements linked to that TC must meet that threshold (Q3). This process described for the Security TC was repeated for each of the 10 TCs.

TABLE 5. CASE STUDY ANALYSIS EXAMPLE

Process Elements Relevant to Security TC	Model	Project 1		...	Project 24		Q2. % Match (Must be ≥ 75%)
		Data	Q1. Match to Model?	...	Data	Q1. Match to Model?	
36. FIPS 199 Security Categorization	S	S	Y	...	D	N	89
53. Security Requirements Traceability Matrix	S	D	N	...	D	N	74
54. Plan of Actions and Milestones	S	D	N	...	D	Y	85
55. System Security Plan	S	S	Y	...	S	Y	100
56. Disaster Recovery Plan	S	S	Y	...	D	N	89
57. Security Risk Assessment (SRA)	S	D	N	...	D	N	74
59. Security Test and Evaluation Plan	S	D	N	...	D	Y	85
61. Contingency Plan	S	S	Y	...	S	Y	100
64. Interconnection Security Agreement	S	D	N	...	D	N	74
84. Security Assessment Report	S	D	N	...	D	Y	85
85. Security Accreditation Package	S	S	Y	...	S	Y	100
92. Authority to Operate Letter	S	S	Y	...	S	Y	100
100. FISMA Metrics Report	S	S	Y	...	D	N	89
101. Security Incident Reports	S	D	N	...	D	N	74
102. C&A Updates	S	D	N	...	D	Y	85
Q3. % Meeting Threshold (must be > 75%)							**95%**

Note. C&A = Certification and Accreditation; FIPS = Federal Information Processing Standards; FISMA = Federal Information Security Management Act.

Case Study Results

Before presenting the specific results of the case study analysis, the following key points must be noted:

- As stated in the previous section, the analysis focused on the project attributes that are not the default. As such, the table of results highlights the nondefault attributes and the associated comparative results to the model.
- Additionally, the Life-cycle Approach TC consisted of three options: waterfall, incremental, and agile. These three were chosen based on the common development methodologies used in DHS. With waterfall applied as the default attribute for the standard process, the analysis focused on incremental and agile.
- In the analysis of the 11 projects using the incremental development methodology, we discovered the tailoring plan for each was limited to a single increment of the project, with a corresponding note that the tailoring plan would be updated prior to initiation of each new increment. As such, the incremental development projects actually reflected a waterfall-based tailoring strategy.
- For the development methodology, the table of results then focuses on the projects that use the agile method, and assesses whether the model matches the tailoring approach of those projects for the process elements that correspond to the Life-cycle Approach TC.

Table 6 shows the results of the analysis comparing the relevant projects to the model outputs. The number of projects with each particular attribute is provided. Overall, the results show that nine of the 10 TCs can be used to reduce the amount of manual tailoring for a particular project. The one exception is the "Project Size" TC. As discussed previously, there is limited specific research regarding the application of SE based on project size, and more specifically, to small projects. Laporte, Alexandre, and Renault (2008, p. 98) discuss the need for developing international standards to address the needs of small development organizations; one objective is to "provide harmonized documentation" integrating standards, work products, and deliverables. This naturally extends to SE within small organizations and large organizations with small projects in their portfolio. Our analysis showed that low correlation exists between the model and the small project case studies. This occurred because of the following: while very few projects tailored the process elements recommended by the model, each project

team executed, on average, 90 percent of the relevant process elements. As a result of their small size, few projects leveraged the opportunity to tailor process elements.

TABLE 6. CASE STUDY ANALYSIS RESULTS

ID	Project Attribute	# Relevant Projects	# Process Elements	# Process Elements With ≥ 75% Match to the Model	% Process Elements With ≥ 75% Match to the Model
1	Security Impact	17	15	15	100%
2	Privacy Impact	9	4	4	100%
3	Intelligence Impact	3	1	1	100%
4	Interoperability Impact	11	8	6	75%
5	Accessibility Impact	13	2	2	100%
6	Technology Demonstration Planned	7	1	1	100%
7	Environmental Impact	7	1	1	100%
8	Agile Development	2	19	15	79%
9	COTS	6	9	7	78%
10	Small Project	6	20	2	10%

Conclusions and Future Work

Process tailoring is critical to effectively and efficiently executing SE based on the unique characteristics of a project. Many SE processes recommend tailoring, but are not supported with tailoring guidelines. This article examines the previous research of process tailoring in software literature and SE standards for certain types of projects, and proposes a model for SE process tailoring. Our comparison of the SEPTM model to several case studies within the DHS shows rule-based process tailoring, coupled with project attributes, can be a viable approach to reducing the amount of manual tailoring required for a given project.

While our research objectives were accomplished for the development of the initial model, additional research should be performed in the area of SE process tailoring. Key elements requiring further research include:

- Very little research exists regarding SE process execution on small projects or in small project teams. The INCOSE Very Small and Micro Entities (VSME) Working Group is developing work packages that recommend various levels of SE process execution for small projects. The SEPTM described herein could be informed by the findings of their research.
- The model's "Complexity and Precedence of the System" TC is based on a selection of either development or COTS. The model should be further refined based on an investigation of COTS integration, which has become more common in recent years.
- The initial model is based primarily on the *INCOSE Systems Engineering Handbook*. Certainly any number of standard SE processes can be used as a starting point for organizational process tailoring; additional research should be performed to identify common tailoring considerations across a broader set of SE standards and literature.
- Investigate applications of data mining to the process of selecting the TCs that form the basis of the model.

References

Ahn, Y. W., Ahn, H. J., & Park, S. J. (2003). Knowledge and case-based reasoning for customization of software processes—A hybrid approach. *International Journal of Software Engineering and Knowledge Engineering, 13*(3), 293–312.

Aoyama, M. (1998). Web-based agile software development. *IEEE Software, 15*(6), 56–65.

Baldassarre, M. T., Caivano, D., Visaggio, C. A., & Visaggio, G. (2002, October 3–4). ProMisE: A framework for process models customization to the operative context. *Proceedings of the International Symposium on Empirical Software Engineering* (ISESE'02) (pp. 103–110), Nara, Japan, IEEE Computer Society. doi: 10.1109/ISESE.2002.1166930

Boehm, B. (2006). Some future e trends and implications for systems and software engineering processes. *Systems Engineering, 9*(1), 1–19.

Browning, T. R., Fricke, E., & Negele, H. (2006). Key concepts in modeling product development processes. *Systems Engineering, 9*(2), 104–128.

Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. *IEEE Software, 25*(1), 60–67.

Cloutier, R. J., & Verma, D. (2007). Applying the concept of patterns to systems architecture. *Systems Engineering, 10*(2), 138–154.

Daneva, M., & Ahituv, N. (2010, May 19–21). *Evaluating cross-organizational ERP requirements engineering practices: A focus group study*. Proceedings of 2010 Fourth International Conference on Research Challenges in Information Science (RCIS) (pp. 279–286), Nice, France, IEEE. doi:10.1109/RCIS.2010.5507387

Department of Homeland Security. (2010). *DHS systems engineering lifecycle/process guide* (Version 2.0). Retrieved from https://acc.dau.mil/adl/en-US/245894/file/53585/Appendix_B_Systems_Engineering_Life_Cycle__SELC__VER_2_0_Release_9-21-10.pdf

Fortune, J., & Valerdi, R. (2013). A framework for reusing systems engineering products. *Systems Engineering, 16*(3), 304–312.

Hagge, L., & Lappe, K. (2005). Sharing requirements engineering experience using patterns. *IEEE Software, 22*(1), 24–31.

Hanssen, G. K., Westerheim, H., & Bjørnson, F. O. (2005). Tailoring RUP to a defined project type: A case study. *Product Focused Software Process Improvement* (Vol. 3547, Lecture Notes in Computer Science) (pp. 314–327), Springer-Verlag Berlin, Heidelberg.

Hausen, H. (1998). A rule-based process model for cooperative software projects: Tailoring and testing a software process to be used on the Web. *Knowledge-Based Systems, 11*(2), 105–113.

Head, S., & Virostko, B. (2008, September 8-11). Improving systems engineering execution and knowledge management. *Proceedings of the 2008 Automobile Testing Conference (Autotestcon)* (pp. 122–127), Salt Lake City, Utah, IEEE.

Henderson-Sellers, B., Ralyté, J., Ågerfalk, P. J., & Rossi, M. (2014). Tailoring a constructed method. *Situational Method Engineering* (pp. 169–194), Springer-Verlag Berlin, Heidelberg.

Humphrey, W. S. (1989). *Managing the software process*. Boston, MA: Addison-Wesley Professional.

Hwang, Y. H., & Park, J. (2006). Approaches and requirements to develop and improve the standard processes for a research and development organization. *Systems Engineering, 9*(1), 35–44.

International Council on Systems Engineering. (2015). *INCOSE systems engineering handbook: A guide for system life cycle processes and activities* (4th ed.). Hoboken, NJ: Wiley.

ISO/IEC/IEEE. (2008). *Systems and software engineering – system life cycle processes* (ISO/IEC/IEEE 15288: 2008). Piscataway, NJ: Author. doi:10.1109/IEEESTD.2008.4475828

Jaufman, O., & Münch, J. (2005, June 13–15). Acquisition of a project-specific process. *Proceedings of the 6th International Conference on Product Focused Software Process Improvement (PROFES '05)* (pp. 328–342), Oulu, Finland, Springer-Verlag Berlin, Heidelberg. doi: 10.1007/11497455_27

Kang, D., Song, I., Park, S., Bae, D., Kim, H., & Lee, N. (2008, December 2–5). A case retrieval method for knowledge-based software process tailoring using structural similarity. *Proceedings of the 15th Asia-Pacific Software Engineering Conference (APSEC 2008)* (pp. 51–58), Beijing, China, IEEE Computer Society.

Krishnan, M. S., & Kellner, M. I. (1999, November). Measuring process consistency: Implications for reducing software. *IEEE Transactions on Software Engineering, 25*(6), 800–815.

Kuhrmann, M., Méndez Fernández, D., & Tiessler, M. (2014). A mapping study on the feasibility of method engineering. *Journal of Software: Evolution and Process, 26*(12), 1053–1073.

Laporte, C. Y., Alexandre, S., & Renault, A. (2008). Developing international standards for very small enterprises. *Computer, 41*(3), 98–101.

Laufer, A. (2001). *Tailoring project processes* [Powerpoint presentation]. Retrieved from http://www.nasa.gov/pdf/293181main_56639main_laufer_ east_ west_0301.pdf

Park, S., & Bae, D. (2013). Tailoring a large-sized software process using process slicing and case-based reasoning technique. *IET Software, 7*(1), 47–55.

Pedreira, O., Piattini, M., Luaces, M. R., & Brisaboa, N. R. (2007). A systematic review of software process tailoring. *ACM SIGSOFT Software Engineering Notes, 32*(3), 1–6.

Pereira, E. B., Bastos, R. M., & Oliveira, T. C. (2007, March 20–23). A systematic approach to process tailoring. *Proceedings of the 2007 International Conference on Systems Engineering and Modeling (ICSEM'07)* (pp. 71–78), Herzeliya and Haifa, Israel, INCOSE.

Perry, D. E. (1996, June 17–19). Practical issues in process reuse. *Proceedings of the 10th International, Software Process Workshop: Process Support of Software Product Lines* (pp. 12–14), Dijon, France, IEEE Computer Society.

Plögert, K. (1996). The tailoring process in the German v-model. *Journal of Systems Architecture, 42*(8), 601–609.

Ramasubbu, N., Krishnan, M. S., & Kompalli, P. (2005). Leveraging global resources: A process maturity framework for managing distributed development. *IEEE Software, 22*(3), 80–86. doi:10.1109/MS.2005.69

Verma, D., & Fabrycky, W. J. (1997). Systematically identifying system engineering practices and methods. *Aerospace and Electronic Systems Transactions, 33*(2), 587–595.

Xu, P., & Ramesh, B. (2007). Software process tailoring: An empirical investigation. *Journal of Management Information Systems, 24*(2), 293–328.

Xu, P., & Ramesh, B. (2009). Impact of knowledge support on the performance of software process tailoring. *Journal of Management Information Systems, 25*(3), 277–314.

Yoon, I., Min, S., & Bae, D. (2001, December 4–7). Tailoring and verifying software process. *Proceedings of Eighth Asia-Pacific Software Engineering Conference (APSEC 2001)* (pp. 202–209), Macau, China, IEEE Computer Society.

Biographies

Mr. Matthew Graviss is the director of the Systems Analysis and Evaluation Division within the U.S. Customs and Border Protection's Office of Technology Innovation and Acquisition. He holds a BS in Mechanical Engineering from Auburn University and an MS in Mechanical Engineering from Texas A&M University. He is currently pursuing a PhD in Systems Engineering at The George Washington University.

(E-mail address: mgraviss@gmail.com)

Dr. Shahram Sarkani is professor of Engineering Management and Systems Engineering (EMSE), and director of EMSE Off-Campus Programs at The George Washington University. He designs and administers graduate programs that enroll over 1,000 students across the United States and abroad. Dr. Sarkani holds a BS and MS in Civil Engineering from Louisiana State University, a PhD in Civil Engineering from Rice University, and is a licensed Professional Engineer.

(E-mail address: sarkani@gwu.edu)

Dr. Thomas A. Mazzuchi is professor of Engineering Management and Systems Engineering at The George Washington University. His research interests include reliability, life testing design and inference, maintenance inspection policy analysis, and expert judgment in risk analysis. He holds a BA in Mathematics from Gettysburg College, and an MS and DSc in Operations Research from The George Washington University.

(E-mail address: mazzu@gwu.edu)