



35th Annual **INCOSE**
international symposium
hybrid event
Ottawa, Canada
July 26 - 31, 2025

Lifecycle Switching Costs

Yanheng Henry Zhu
USA

Copyright © 2024 by Yanheng Henry Zhu. Permission granted to INCOSE to publish and use.

Abstract. Switching costs, an important topic in adaptive systems, are the costs incurred by modifying a product design to another. Most previous works focused on development switching costs. However, design switching can trigger different production costs and operational costs too. Also, for development costs, only two methods, process-based and parametric are available. For open-source tools, only one tool is available, COSYSMO (parametric). However, parametric methods require large sets of historic cost data, and cannot support small or medium size products, which leaves the problem still unresolved. This paper addressed these gaps. Its contributions are: first, it analyzed the whole lifecycle switching cost and revealed that some mathematical properties from development switching costs no longer hold. Second, it identified and proved the new mathematical properties for lifecycle switching costs, and developed a new useful concept, vendor switching cost. Third, it presented a new tool for estimating these costs, which does not rely on product sizes, making it the first open-source tool that supports all systems. Finally, we demonstrated its effectiveness through an example use case of office area design, and made the tool available online, with a manual and this detailed example.

Keywords. adaptability, switching cost, systems engineering, software tool.

I. Introduction

Switching cost is an important concept in adaptability and adaptive systems (INCOSE and Steven Institute of Technology, 2023) (Adams, 2024). It represents the cost incurred when switching a system design or state to another system design or state. Therefore, it is useful in the design and optimization of an adaptive system. In cost estimation domain, there have been many tools and methods (Adams, 2024), for example, SEER Tools (Galorath Inc., 2024), PRICE Tools (PRICE Systems LLC., 2021), COSYSMO (Alstad, 2019) (Valerdi, 2007), NASA Cost Estimating Handbook (National Aeronautics and Space Administration, 2024), NASA Air Force Cost Model (NAFCOM) (McAfee, Culver, & Naderi, 2011), and Advanced Cost Estimating System (ACES) (4cost, 2024). In terms of research, on development switching costs for generic systems, rather than a specific type of hardware or software systems, so far there have been only two estimation methods (Zhu H., 2018) (Alstad, 2019). While the works on development switching costs are important, their scopes significantly limit their usages. Lifecycle switching cost must take into account production cost and operational cost which includes maintenance costs if any (Zhu H., 2024). As shown in this paper, some mathematical properties of development switching costs do not hold anymore. Therefore, it is important to study the theoretical properties of lifecycle switching costs. This is the first contribution in this paper.

As said above, so far there have been only two estimation methods for development switching costs for generic systems, process-based (Zhu H., 2018) (Zhu H., 2024) and parametric (Alstad, 2019). In terms of free estimation tools, there is only one tool COSYSMO, a tool that adopts parametric estimation (Alstad, 2019). This paper presents another estimation tool we developed, call “SWCost Tool” that was put online

for free use (Zhu Y. , 2024). This tool adopts the process-based estimation method (Zhu H. , 2018). Process-based method utilizes the commonly used WBS (Work Breakdown Structure) method, and has significant advantages over parametric methods: Parametric methods rely on statistical methods that rely on a large amount of data and large project sizes. In reality, this may not be always the case. Even when these are available, the type of projects may be vastly different thus requiring additional studies in data mining to achieve needed accuracy. Process-based method does not rely on big data and large project sizes, and thus can achieve higher accuracy for all types of projects, and therefore bridges the gap of needing a tool that covers all product sizes. This is our second contribution, which is useful to the systems engineering community and the industry. We will explain how SWCost Tool works and how to use it. An example case study on office cubicle modification switching cost estimation is provided to illustrate the effectiveness of this tool.

The remaining parts of this paper are as follows: Section II will describe the concept and mathematical properties of lifecycle switching costs; Section III will describe the theory, methods and the development behind the SWCost Tool; Section IV will describe an example use case for using the tool, and finally Section V concludes the study.

II. CONCEPT AND MATHEMATICAL PROPERTIES

Lifecycle switching cost is useful for decision choices. To change a product design, often we need to go through a trade study where multiple design alternatives must be elaborated from which the best would be picked. The criteria for picking the best design may be based on many factors, such as functionality, performance, cost, etc. Within the cost factor, considering the switching cost from the current design to each of the design alternatives, if we only consider development cost, it may not be the best result. This is because less switching cost in development may not mean less cost in production and operations. This is why lifecycle switching cost must be considered.

Lifecycle Switching Cost

(Zhu H. , 2024) illustrated the major components for a product's lifecycle switching cost, which includes:

- Development Switching Cost
- Production Switching Cost
- Operational Switching Cost

Below we will show some of the mathematical properties for development switching costs (Zhu H. , 2018) no longer hold for lifecycle switching costs. First, we will define a few notations. Let us denote $SWCost_d(A, B)$, $SWCost_p(A,B)$, $SWCost_o(A,B)$ and $SWCost_t(A,B)$ as development switching cost, production switching cost, operation switching cost and lifecycle switching cost, respectively, when changing from System Design A to System Design B, where the subscript t means total. There may be other costs in the lifecycle, but we will factor them all into development, production and operations. Let us also define $C_d(A)$, $C_p(A)$, $C_o(A)$ and $C_t(A)$ as the raw development cost, where "raw" means developing System A from scratch without a prior product, and the raw production cost, raw operational cost and raw lifecycle cost associated with System A, respectively.

Properties of Development Switching Cost

Below we will review the mathematical properties of development switching costs (Zhu H. , 2018):

1. Non-Negative Property: Development switching cost is always non-negative.

This is because under the best case, no change is required from the prior product to build the current product.

2. Triangle Property:

$$SWCost_d(A, B) \leq C_d(A) + C_d(B) \quad (1)$$

3. Minimum Development Switching Cost

For practical purpose of assuming modification from A to B is implementable within finite time, there exists a lower bound of development switching cost from A to B.

Lifecycle Switching Cost

Let us examine the above properties and see if they still hold for lifecycle switching costs.

First, Property 1 (Non-Negative Property) is no longer true. For example, supposed modifying the design of System A's design to become System B costs \$500K in US\$, and A is not in production or service yet. But during modifying A's production facility to become B's production facility, many assets are no longer needed and are sold which lead to modification becomes a profitable action. Or, if some of A products were already produced but the new design uses much less materials which result in profits by recycling these. Supposed the resulted profit is \$700K, and the operational cost of A and B remain the same as \$100K over the next 5 years, after which the product will be decommissioned. Then $SWCost_t(A, B) = \$500K - \$700K = -\$200K$, which is a negative number. One can also see that if the resulted profit turn out to be \$500K, then $SWCost_t(A, B)$ would be 0. Or, if the resulted profit is $< \$500K$ or even negative, then the $SWCost_t(A, B)$ would be positive. Thus, we conclude the following property for lifecycle switching costs:

Polarity Property: *Lifecycle switching cost can be a negative, zero or a positive number.*

This differs from the Non-Negative Property in development switching costs. It should also be reminded that the production and operational switching costs contain recurring costs, while the development switching costs are generally non-recurring costs. Typically, production and operational costs are tied with unit costs and the number of productions, with some additional overhead.

Next, let us examine the Triangle Property for lifecycle case and see if the following holds:

$$SWCost_t(A, B) \leq C_t(A) + C_t(B) \quad (2)$$

Analysis is as follows:

First, $SWCost_d(A, B) \leq C_d(A) + C_d(B)$, per (Zhu H. , 2018).

Second, for practical cases, we can break down the production switching cost further:

$$SWCost_p(A, B) = SWCost_{p-f}(A, B) + C_m(A, B) \quad (3)$$

where $SWCost_{p-f}(A, B)$ is the cost of modifying A's manufacturing facility (called A's facility, for short, henceforth) to the facility for manufacturing B, and $C_m(A, B)$ is the cost of manufacturing B in such a facility. Here, the facility includes factories, supply chains, warehouses, and anything needed to support productions.

Here, let us define an assumption:

(Assumption A1) If modifying A's facility to be a facility that manufactures B is more costly than building B's facility from scratch, we will always and be able to build B's facility from scratch.

This assumption may not always be true due to non-engineering factors, such as policies, strategies or regulations. For example, the vendor may not be able to purchase new land for B's facility for the above reasons, and thus has to modify A's facility to B's. In the following, we also assume:

(Assumption A3) No prior A products were produced yet.

As seen earlier, $SWCost_p(A, B)$ can be >0 , $=0$ or <0 . There can be several scenarios:

Case 1: A's production facility does not exist yet, or it does exist and we need to modify it to be what facility you would build for B from scratch which we call B's *raw facility*, and then manufacture B from scratch.

In this case, $C_m(A, B) = C_m(B)$, the cost of manufacturing B using B's raw facility. Then, $SWCost_p(A, B) = SWCost_{p-f}(A, B) + C_m(B)$. On the other hand, $C_p(B) = C_f(B) + C_m(B)$ where $C_f(B)$ is the cost of building B's raw facility from scratch.

Because modifying A's facility to B's facility, sometimes can be more costly than building B's raw facility from scratch, in this case it may make sense to build B's raw facility from scratch when feasible, which means we will adopt Assumption A1. Then in Case 1, we can conclude $SWCost_{p-f}(A, B) \leq C_f(B)$ and then therefore $SWCost_{p-f}(A, B) + C_m(B) \leq C_f(B) + C_m(B)$, and thus $SWCost_p(A, B) \leq C_p(B)$.

Case 2: A's facility is not modified to be B's raw facility, instead modified to a facility where B can be produced by modifying A after A is produced fully or partially.

In another word, B's facility is not the same as B's raw facility. In this case, A's facility must still be reserved, or partially reserved, and additional facilities may be needed to support modifying A to B during production. Then, the final facility that includes part or all of A's facility, along with additional facility to turn A or part of A to B, together is called B's *non-raw facility*, another kind of B's facility. However, as use Assumption A1, if building B's non-raw facility from A's facility costs more than building B's raw facility from scratch, we will always have the option to build the smaller cost one. Thus, we can have facility switching cost $SWCost_{p-f}(A, B) \leq C_f(B)$, but this may not be always adopted as it depends on whether this will minimize the $SWCost_p(A, B)$ or not.

Next, let us look at the manufacturing cost of B in a B's non-raw facility, the 2nd item in the equation of $SWCost_p(A, B)$. In Case 2, it may not be $C_m(B)$ which is the manufacturing cost B's raw facility. That means manufacturing B in a B's non-raw facility may not cost less than manufacturing B in B's raw facility, i.e. $C_m(A, B)$ may be $> C_m(B)$.

Here, let us see what makes sense to do when designing a production process. It makes sense to consider both facility cost and manufacturing cost together and minimize the total, from the production planner's point of view. That is: if total production cost, which includes both facility and manufacturing costs, of Case 1 is less than Case 2, we will select Case 1 to proceed, otherwise we select Case 2 to proceed. We establish this as another assumption:

(Assumption A2) If production cost using modified A's facility to B's facility is higher than product cost using B's raw facility, we will always and be able to adopt B's raw facility.

Based on our assumptions, we can conclude:

$$SWCost_p(A, B) \leq C_p(B) \quad (4)$$

Finally, let us look at operational switching cost. Noted that there may be new equipment or modified equipment required for B's operations, after A is changed to B. Let us break it down to two parts: operational equipment switching cost and labor switching cost. One may use a similar strategy as Assumption A2 to ensure equipment change cost will be bounded by building B's operational equipment from scratch. But for labor switching cost, without Assumption A3, operating B by human may or may not cost the same under the cases when product A was previously in service or not. There is at least one element that can differ them: when switching from A to B, the human learning cost may be different from the cost of learning on operating B when A did not exist before. The difference can be positive, 0 or negative, depending on how different A and B are, etc. For example, sometimes knowledge with A may become a burden for learning B, and it may be less costly to learn B directly without knowing A, etc. This means, there is no guarantee that $SWCost_o(A,B) \leq C_o(B)$.

On the other hand, if we attempt to optimize operational cost by doing this - if the learning cost and other operational costs due to modifying A to B is more than the cost of operating B without A's existence, we will pick B without A's existence. This is not possible because A's existence may be already a fact. Thus, Triangle Property does not hold for operational switching cost, and there are chances its impact can overwhelm the development and production switching cost, and disable $SWCost_t(A,B)$ from being smaller than $C_t(A)+C_t(B)$, depending on different scenarios. Thus:

Triangle Property does not hold for lifecycle switching costs.

However, we will introduce a concept of vendor switching cost $SWCost_v(A,B)$ which only includes development switching cost and production switching cost. This is because in many cases, the operational cost is not paid by the vendor, and thus does not need to be considered in the cost optimization by the vendor. For example, airline operational cost, or some government project cost may be paid by airlines or the government. This also means, to minimize lifecycle switching cost, service history and operational considerations must be given during the design stage.

For vendor switching cost, based on Eq.1 and Eq.4, Triangle Property does hold:

$$SWCost_v(A,B) \leq C_v(A)+C_v(B) \quad (5)$$

Next, we examine Property 3: Lower bound of lifecycle switching cost, for practical purposes (denoted as Assumption A4 here) where number of components, products, development time, learning time, operation time and service years are finite and bounded.

For development, there is a lower bound, per (Zhu H. , 2018). Due to Assumption A4, everything is finite, so there are lower bounds for both production and operational switching costs. Thus, lifecycle switching cost and vendor switching cost have lower bounds, which is practical in real-world.

Theoretical Results Summary

In summary, we conclude, based on our assumptions:

1. Non-Negative Property no longer holds for lifecycle switching costs.
2. Lifecycle switching cost has the Polarity Property where the cost number can be either negative, 0 or positive.
3. Without Assumption A3, Triangle Property no longer holds for lifecycle switching cost, but still holds for vendor switching cost.

4. Lower-Bound Property holds: Production switching costs, operational switching costs, vendor switching costs and lifecycle switching costs have lower bounds.

III. DESIGN OF SWCOST TOOL

As described previously, for free or open-source switching cost estimation tools, so far there is only one tool COSYSMO (Alstad, 2019). However, this tool is parametric-based and thus relies on a large amount of data and perhaps large products as well. Some other commercial tools also leverage some concepts from COSYSMO. The need for a generic tool that also supports small size products without wholly relying on big data or associated data mining effort is still not satisfied, at least to the best extent that it can be. This becomes an open problem and makes it an urgent need to fill this gap.

We developed the “SWCost” Tool and offer it as an open-source free tool downloadable online (Zhu Y. , 2024). The tool adopts process-based estimation method (Zhu H. , 2018) for development switching cost, and additional methods for production and operational switching costs as described in this paper. Similar to early versions of COSYSMO, it is in the form of an Excel spreadsheet, and is convenient to use. In this section, we will describe the architecture and functional design of SWCost Tool.

Estimation Methods

The development switching cost estimation uses the recursive algorithm developed in (Zhu H. , 2018), which is reviewed below.

First, hierarchical structures of System A and B are both available as the starting point, using a similar structure as WBS. They both can contain many layers, each of which is broken down to more detailed components until the components are available for acquisition or detailed design. Then a modification plan must be developed to indicate which components in A are to be modified to become which components in B, which components in B are new, and which components in A are removed. Connections are viewed as components too. In another word, each component modification activity falls into several options: add, remove or change, which will viewed as a system and be further decomposed to next layer, unless the cost of this modification activity already can be estimated, based on past cost model, acquisition price, historic data or EJ (Expert Judgement). The decomposition is performed recursively until all the decomposed component modifications can be estimated. All these component modifications will be added with additional systems engineering cost in that layer, using a tax model (Zhu H. , 2018), to produce the final switching cost from A to B.

When estimating production switching cost, in this first version of the SWCost tool, we adopted a simplified framework of unit switching cost multiplied with number of products produced.

When estimating operational switching cost, in this first version of the tool, we adopted a simple framework of estimating unit operation switching cost multiplied with number of units, and an additional change in costs of coordinating all units, provided by the customer.

Future versions can be extended to include more complex methods, though this model already adopted some of the commonly used methods, such as tax models.

Architecture

The tool is coded in an Excel file that contains multiple worksheets. It is broken into several components as in Figure 1, where the Output SWCost module sums up the outputs from the Development SWCost,

Production SWCost and Operational SWCost modules. To calculate development switching cost from System A to B, it needs the information about the design of System A and B, the plan of modifying A to B (as illustrated in Figure 2) and the available data on cost models, historic data, acquisition prices and EJ data.

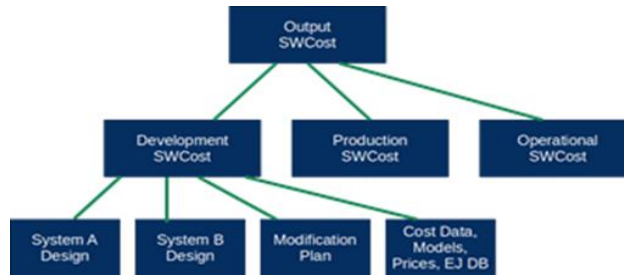


Figure 1. SWCost Tool Architecture

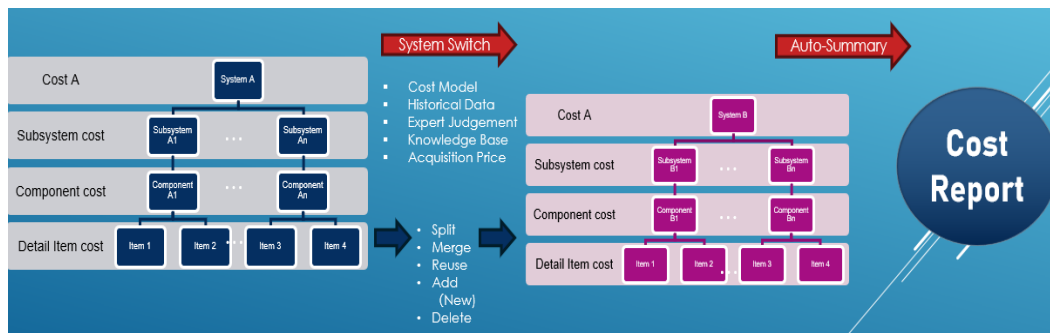


Figure 2. Modification from A to B

Usage

Apart from documentation information, the file contains several worksheets: System A Design, System B Design & Modification Plan and Development Switching Cost, Production Switching Cost, Operation Switching Cost and Summary.

The user needs to put in System A and B’s designs in hierarchical structures, and then a component-to-component modification plan where System A’s component items can be selected in a drop-down list to be the basis from which a System B’s component item is modified. It also includes the options of removing a component item, or adding a new item.

Upon collecting existing cost data, models, acquisition prices or EJ data, production information, operational information, the tool then automatically calculates the results in the 2nd, 3rd and 4th worksheets. The “Summary” worksheet gathers all such information and produces the development, production and operational switching cost estimation, from which the vendor and lifecycle switching costs are output.

IV. EXAMPLE USE CASE

In this section, we will show how to use this tool to estimate switching cost for a cubicle office modification project.

Office Design A and Office Design B

Supposed we have a cubicle office design A that we want to modify to design B. Design A contains a computer system, a set of furniture and some supplies, where the electricity outlets are not changed and thus excluded. The user can enter these details using System A worksheet template, as shown in Figure 3.

B	C	D	E
System A			
Computer System			
		PC	CPU- Intel i5 RAM- 8gb SSD- 256gb
		Monitor	Size- 20in
		Input Devices	Mouse Keyboard Headset
Furniture			
		Desk	40in x 16in
		Chair	Adjustable Height
		Shelf	3 Layers
Supplies			
		Printer	Canon Inkjet Ink Cartridges
		Small appliances	Coffee maker
		Stationery	Pens Staplers Rulers Copy paper

Figure 3. Office Design A

System B				
Sub-Systems	All Item SW Cost	Component SE Cost	Components	Items
Computer System	905	271.5	PC	CPU- Intel i7 RAM- 32gb SSD- 1tb GPU
	190	57	Monitor	Size-27in screen protector
	0	0	Input Devices	Mouse Keyboard Headset
Furniture	0	0	Desk	40in x 16in
	0	0	Chair	Adjustable Height
	50	15	Shelf	4 Layers
Supplies	0	0	Printer	Canon Inkjet Ink Cartridges
	35	10.5	Small Appliances	Coffee maker Water boiler Fans
	32	9.6	Stationary	More Pens Staples Rulers Scissors

Figure 4. Office Design B

In the 2nd worksheet, a user can enter Office B design, as shown in Figure 4. It contains a modification to the computer system, furniture and office supplies. In this figure, the “All Item SWCosts” and “Component SE Cost” are automatically generated once the modification plan is entered.

Modification Plan, Development Switching Cost

In the same 2nd worksheet, on the right side of each item of Office Design B, the user can select if this item is a new item that is added, an item that is the reuse of another item from Office A, or a modification of another item in Office A.

Earlier the user already entered the Office Design A items. The tool automatically picks up this design information to form a drop-down list. When reusing or modifying an item in Office A, the user can select items in Office A from this drop-down list, as shown in Figure 5. The cost models and data are also linked in this worksheet.

F	G	H	I	J	K	L
			From System A		SE Cost Tax Ra*0.3	
Components	Items					
PC				Modification Option	Costing Option	Item SWCost
	CPU- Intel i7	CPU- Intel i5	Modify	CostModel		225
	RAM- 32gb	RAM- 8gb	Modify	CostModel		100
	SSD- 1tb	SSD- 256gb	Modify	CostModel		80
	GPU		Add	AcquisitionPric		500
Monitor						
	Size-27in	Size- 20in	Modify	CostModel		180
	screen protector		Add	AcquisitionPric		10
Input Devices						
	Mouse	Mouse	Reuse	ExpertJudgeme		0
	Keyboard	Keyboard	Reuse	ExpertJudgeme		0
	Headset	Headset	Reuse	ExpertJudgeme		0
Desk						
	40in x 16in	40in x 16in	Reuse	ExpertJudgeme		0
		Adjustable Height				
		3 Layers				
		Canon Inkjet				
Chair						
	Adjustable Height	Ink Cartridges	Reuse	ExpertJudgeme		0
		Coffee maker				
		Pens				
		Staplers				
Shelf						
	4 Layers	Copy paper	Modify	CostModel		50

Figure 5. Modification Plan

Once Design B associated information is entered, the tool would generate the development switching cost in this worksheet.

Production and Operational Switching Costs

Production and operational switching costs follow the methods described in the last section. In this case, costs of installation etc. are estimated on a per-unit basis. Multiplied it with the number of units (i.e. cubicles), the production switching cost is estimated. Results are shown in Figure 6.

	A	B	C	D
1	Unit Production SWCost=	75	Note: cost of	installation etc.
2	Number of Production Units =	10		
3				
4	Production SWCost=	750		
5				

Figure 6. Production Switching Cost

Operational cost uses 5 years of operation cycle, mainly coming from additional electricity cost from added electric devices. Results are shown in Figure 7.

A	B	C
Unit Operation SWCost=	97.3	Note: additional electricity cost for 5 years of operations
Number of Operation Units =	10	
Fleet Management Overhead Change=	200	
Operation SWCost=	1173	

Figure 7. Operational Switching Cost

Vendor and Lifecycle Switching Cost

Finally, vendor and lifecycle switching costs are generated by the tool, as shown in Figure 8.

	A	B
1	Development SWCost =	1575.6
2	Production SWCost =	750
3	Operation SWCost =	1173
4		
5	Vendor SWCost =	2325.6
6	Lifecycle SWCost =	3498.6
7		

Figure 8. Final Results

This provides a complete estimation of needed switching costs in the product lifecycle.

V. CONCLUSION

Switching costs (or reuse cost) are the costs incurred by modifying a product design to another. Most of the previous work focused on development switching cost estimations. However, switching designs can also cause differences in production cost and operation/maintenance cost. This paper addressed this gap in theory, methods and also tools. First, it analyzed the concept and the theoretical properties of a lifecycle switching cost and showed that the Non-Negative Property and Triangle Property that were valid in development switching costs no longer hold in lifecycle switching costs. Second, it proved that the Triangle Property still holds for vendor switching costs, and also showed the Polarity Property that is suitable for lifecycle switching costs. Third, it presented a new tool for estimating these costs. So far, there have been only two methods in estimating development switching costs for generic systems engineering, process-based and parametric, where parametric methods have an open-source tool COSYSMO. However, parametric methods require large sets of historic cost data, and thus are difficult to support small and medium size products. Our tool does not rely on product sizes thus provided solution to all systems. Finally, we showed the ease of use and effectiveness of this tool through an example use case of office area design.

References

4cost. (2024). Retrieved from Advanced Cost Estimating System (ACES): <http://www.4cost.de>

- Adams, J. (2024). *ENGR 531: Engineering Risk Analysis*. Retrieved from https://www.engr.colostate.edu/se/wp-content/uploads/2020/11/ENGR_531_Course_Info_Sheet.pdf.
- Adams, J. (2024). *SYSE 573: Cost Optimization for Systems Engineers*. Retrieved from <https://www.engr.colostate.edu/se/wp-content/uploads/2020/11/SYSE573.pdf>.
- Alstad, J. (2019). Development of COSYSMO 3.0: An Extended, Unified Cost Estimating Model for Systems Engineering. *INCOSE International Symposium*.
- Galorath Inc. (2024). Retrieved from SEER Tools: <http://www.galorath.com>
- INCOSE and Steven Institute of Technology, I. (2023). "System Adaptability Chapter" in *Systems Engineering Body of Knowledge 9SEBoK) Version 2.8*. IEEE.
- McAfee, J., Culver, G., & Naderi, M. (2011). *NASA AIR FORCE Cost Model (NAFCOM)*. National Aeronautics and Space Administration. Retrieved from <https://ntrs.nasa.gov/citations/20120002891>
- National Aeronautics and Space Administration. (2024). *National Aeronautics and Space Administration*. Retrieved from NASA Cost Estimating Handbook: <https://www.nasa.gov/ocfo/ppc-corner/nasa-cost-estimating-handbook-ceh/>
- PRICE Systems LLC. (2021). Retrieved from PRICE H.: <https://www.pricystems.com/>
- Valerdi, R. (2007). COSYSMO Reuse Extension. *22nd International Forum on COCOMO and Systems/Software Cost Modeling*.
- Zhu, H. (2018). Developing Case-Based Costs Estimation: A Recursive Approach and Case Study. *INCOSE International Symposium*.
- Zhu, H. (2024). Model-Based Switching Costs. *INCOSE International Symposium*.
- Zhu, Y. (2024). *SWCost Tool*. Retrieved from <https://buzlist.com/SWCostTool/>

Biography



Yanheng Henry Zhu. Mr. Zhu is an enthusiast and active developer in systems engineering, robotics/virtual reality and computer science. He is a member of INCOSE New England chapter, and has 5 years of experience in software development. He developed the SWCost tool, an open-source free tool that estimates development switching cost, production switching cost, operational switching cost and lifecycle switching cost. He also developed stochastic models for switching costs for lifecycle cost optimizations. In his free time, he enjoys games and arts.