NSi
THE SOLUTIONS COMPANY

Applying Simple Function Point Analysis to an 804 Rapid Acquisition Program Cost

GALORATH

Cost Estimating Team

- Led by Rainey Southworth, Cost Analyst PMA-281 (NSI)
- Supported by:
  - Bob Hunt, Software SME (NSI)
  - Chad Lucas, Senior Analyst (Galorath)
  - Esteban Sanchez, Senior Analyst (Galorath)

- Special thanks to our supporting team members:
  - Our government leads and counterparts
  - Patrick McGarrity, Program Manager, NSI

Outline

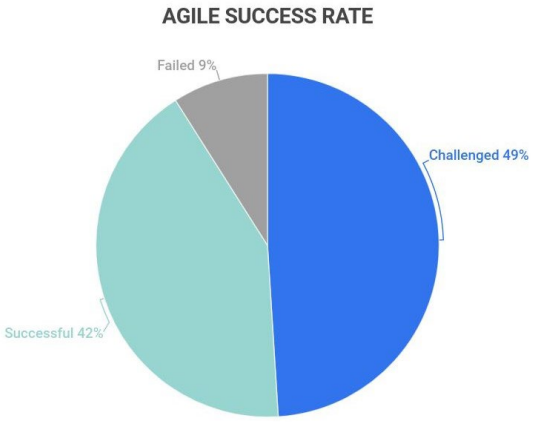
- Continuum Program Overview
- Problem Statement
- Section 804/Rapid Acquisition
- Agile Metrics
- Agile Cost Process
- Sizing Issues and Simple Function Points
- Summary Results/Lessons Learned

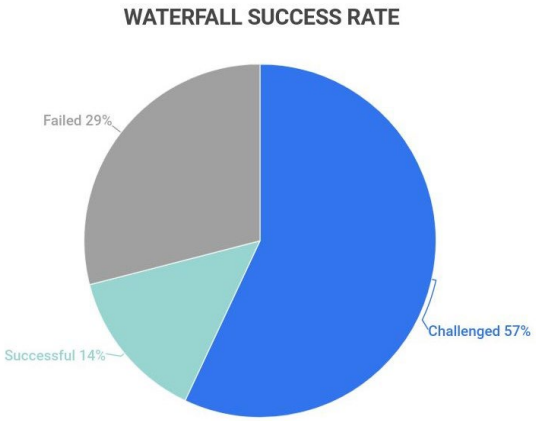
Vision Statement

- Original goal was to create a metric to size and estimate Continuum as a full Agile program
  - PEO and PMA gave specific detail to avoid using SLOC
  - Must account for flexibility of requirements and continual capability additions
  - Initial metrics were difficult to obtain, and resulted in immature team specific data that were inconsistent between teams
- The question became: How can a sizing metric be assigned and utilized, if the size of the program is undetermined?
  - Think of it like a bucket, the capacity cannot be determined if the size of the bucket is undefined
- Ultimately, a forward facing sizing metric that is adaptable enough to include capability additions would be a program management tool, not just a cost derivative
  - Proactive vs reactive estimating
  - Similar to Earned Value Management (EVM)

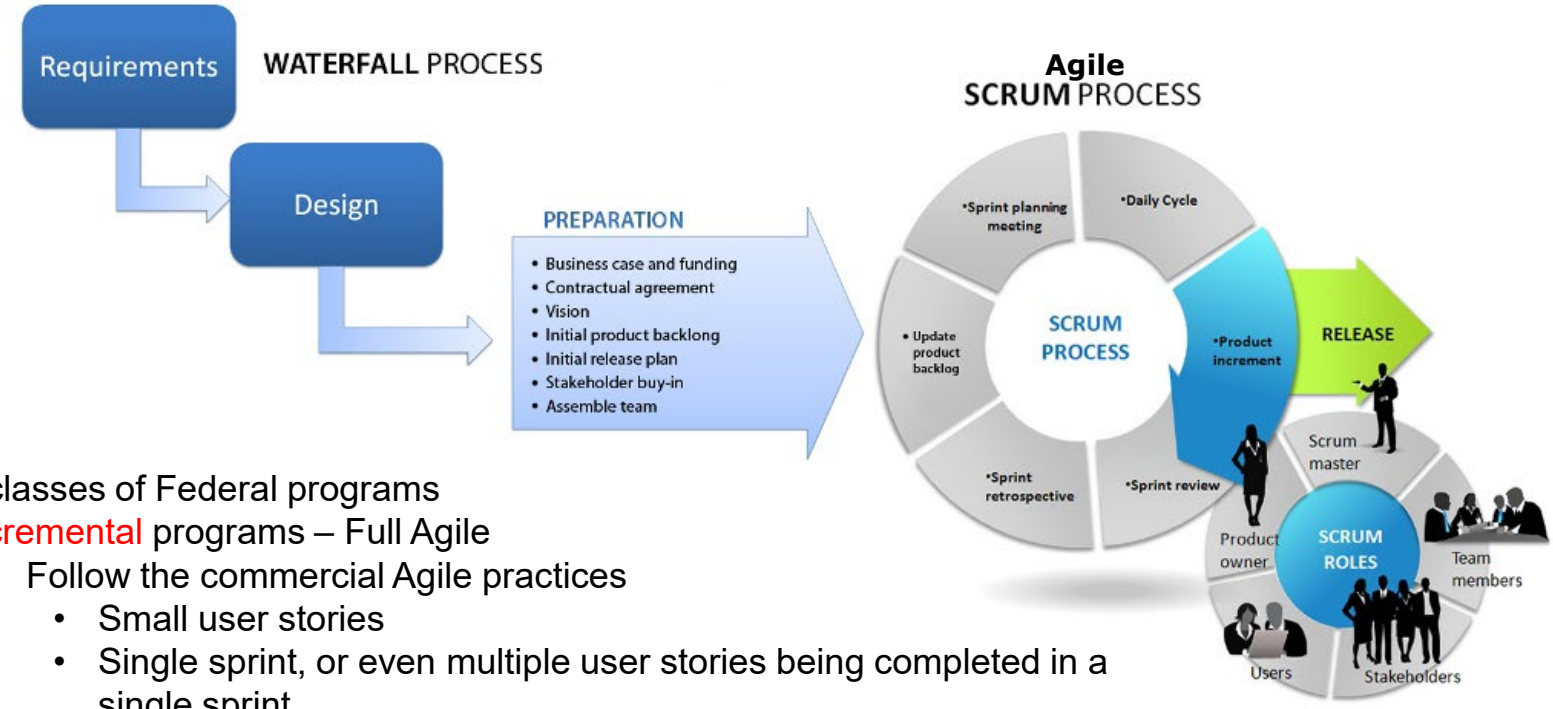
Agile Software Development Statistics

- Agile is an increasingly popular software building methodology
- At least **71% of U.S. companies are now using Agile.**
- Agile projects have a **64% success rate**, whereas projects under the competing methodology known as **waterfall only have a 49% success rate.**
- With that in mind, Agile projects are nearly **1.5X more successful** than waterfall projects.
- **Scrum is the most popular Agile framework**, with 61% of respondents from 76 countries reporting that they use it





Practical Applications of Agile
Full or Hybrid Agile Development



Two classes of Federal programs

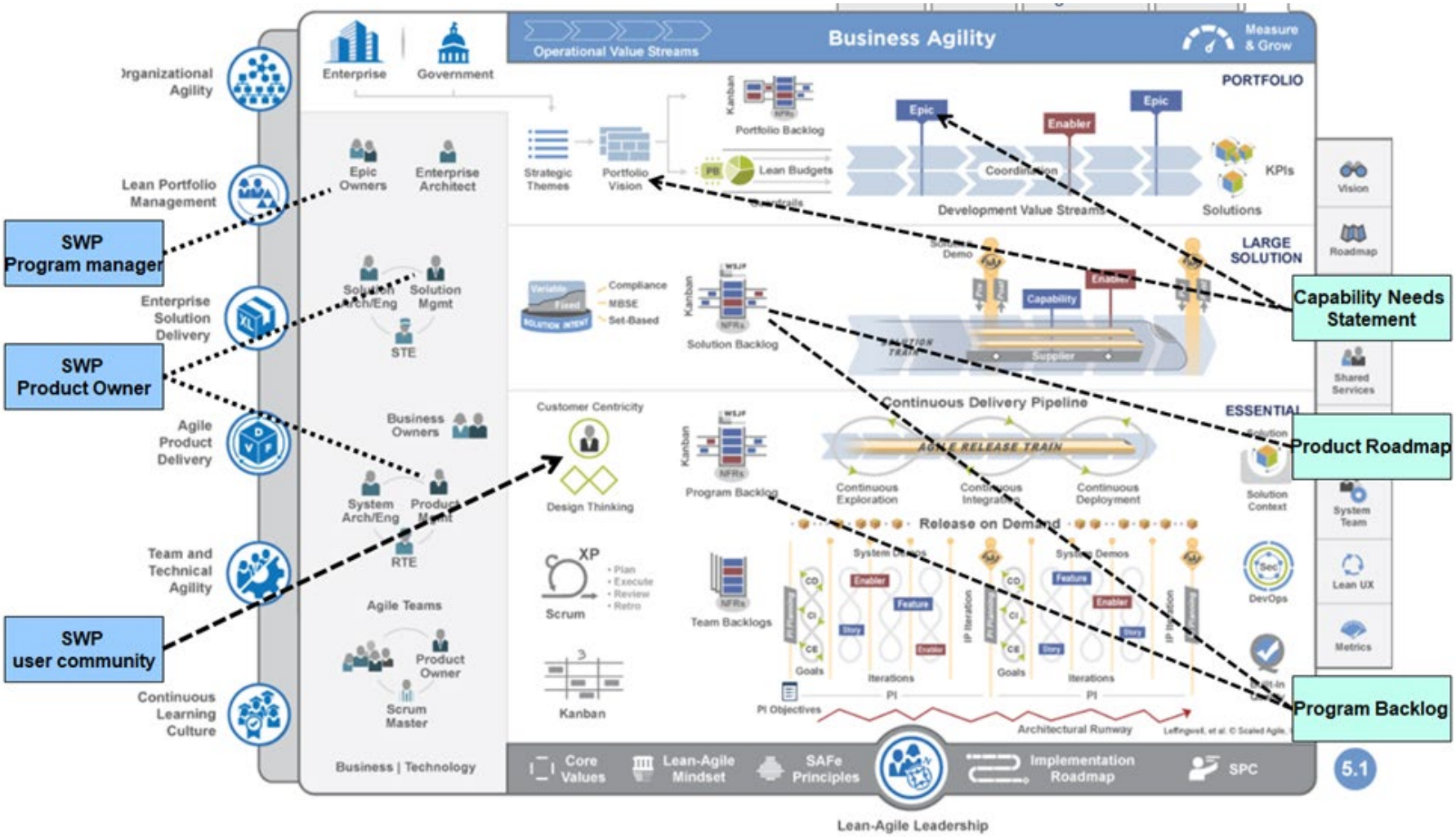
- Incremental programs – Full Agile
  - Follow the commercial Agile practices
    - Small user stories
    - Single sprint, or even multiple user stories being completed in a single sprint
    - **The program office's acquisition strategy led to selection of full agile for Continuum**
    - Generally, not applying a full EVM process
- Transformational programs – Hybrid Agile
  - Creating completely new capabilities
  - "Hybrid-Agile" approach applied
    - Longer sprints
    - Larger conceptual stories/features
    - Full EVM process.

Testing and Sustainment

Finding the Right Requirements Document

- Continuum is using Agile management techniques implemented by an Integrated Product Team (IPT) with a dynamic requirement and no fixed requirements document
- Product backlog documentation was post decisional, where the focus needed to be on **planning**
- We worked with the IPT to identify the most representative "Roadmap".
  - The selected Enterprise Roadmap had over 1300 elements
  - Continuum should be able to roll these elements up to describe any scenario (e.g. by program, platform, milestone, etc)
  - Continual updates to the roadmap can be easily sized based on congruent work packages following engineering concurrence

SAFe/Agile Process



Assessed Estimating Methodologies

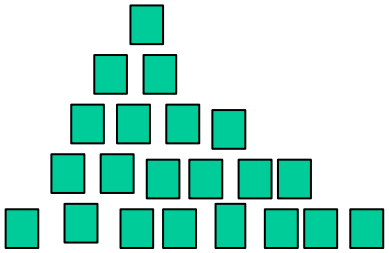
Methodology 1: Many Agile programs are level of effort (labor rates times quantity)

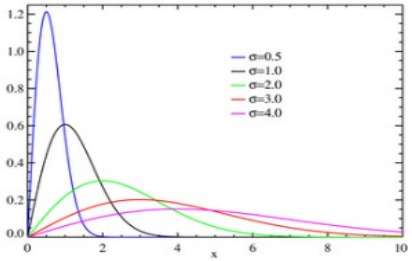
Methodology 2: Simple Build-up approach based on averages can be defined as:
Sprint Team Size (SS) x Sprint length (Sp time) x Number of Sprints (# Sprints)

Methodology 3: Structured approach based on established velocity – most often used internally by the developer since detailed/sensitive data are available to them – need several iterations

Methodology 4: Automated Models (NEMO, SEER, COCOMO, TruePlanning, SLM, …) approach based on a size metric (SLOC, FPs, Stories)

- **There is a fixed relationship between size and effort, e.g. Effort =*ai*(Size Metric)*(bi)*(EAF)**
- **Results are then modified by current trends and analyses**
- **Total effort can be distributed by a mathematical model; e.g. Weibull, Rayleigh**





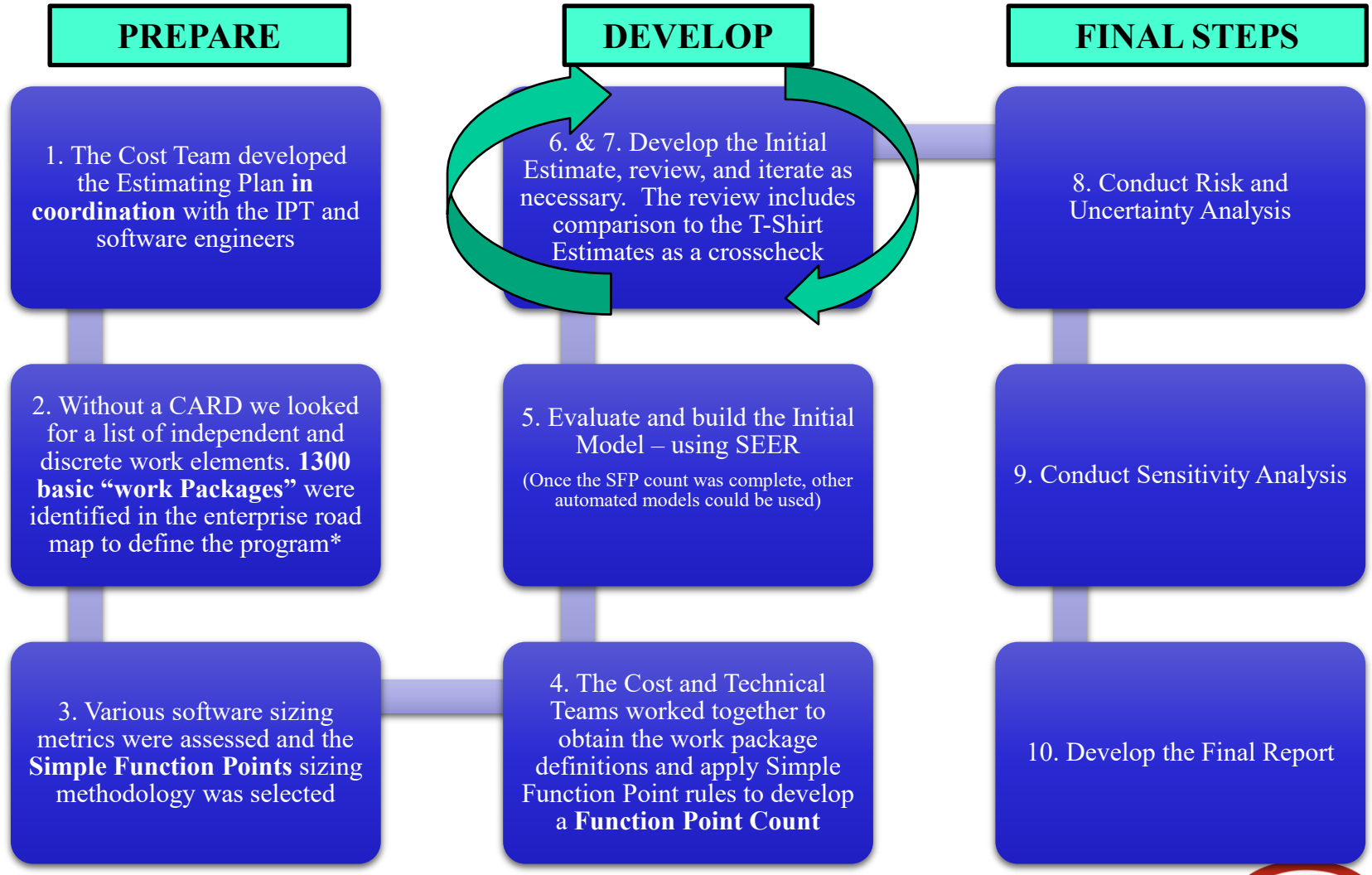
Methodology 5: Analogy/Factor/Complexity approach based on data generated in early *iterations, e.g. Tee-Shirt, Planning Poker, …*

Useful in backlog stages, not in decision making stages

Simple Function Point Solution

- A basis was needed to determine the size of the roadmap
  - Reviewed different sizing options and compared the data requirements for each to the data available
- Simple Function Points (SFPs) were considered because they assign a consistent metric to each capability
  - SFP provided the best fit given the data available and the program's stage of development
  - Removal of the human work factor from the process creates a cost constrained roadmap that refines product deliverables
  - Found that flexibility and consistency could be applied to any Agile program
- Requirements data from development specific roadmaps were used to conduct SFP counts which were input into SEER-SEM to generate and estimate of developmental effort hours.

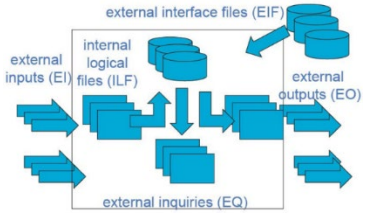
Agile Cost Methodology Overview



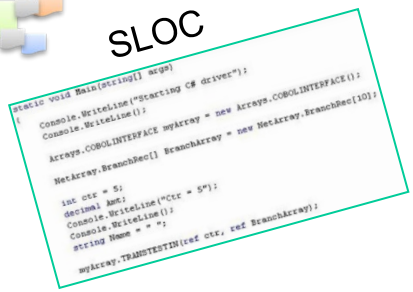
* The estimate can be updated when a new roadmap is provided

Size Continues to be the Main Driver

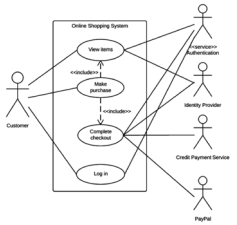
Functional Size
- IFPUG
- COSMIC
- Nesma
- **Simple**
- …



SLOC



Use Cases



Simple Function Points provide a consistent process for early development

User Stories

T-Shirt Sizes

Story Points*

Why Simple Function Points (SFP)

- **Simple Function Points are the best option for early-stage agile development because it is a consistent methodology to estimate size**
  - **Creates a proactive way to measure an early stage program with a high level of accuracy**
- Currently used by other large DoD agile programs e.g. IPPS-A
  - Simple Function Points are supported by the International Function Point User Group (IFPUG)
- Of the 1300 Work Packages (RoadMap elements) the SPF count was directly estimated for 1000 or 83%.
  - Other roadmap elements were estimated by analogy or by engineering judgement

Simple Function Point Analysis

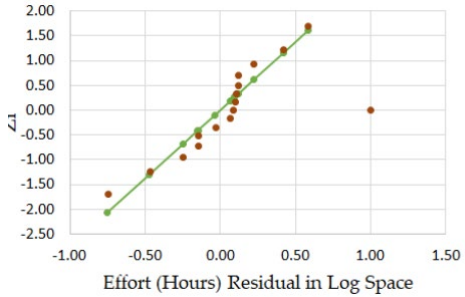
- Method developed by Italian researchers, acquired by IFPUG in 2019*
- Can be performed quickly and early in a program's lifecycle using existing documents
- Focuses on two elementary processes:
  - Transactions
  - Logical Data Groups

IFPUG Components	Low	Average	High
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6
Internal Logical Files	7	10	15
External Interface Files	5	7	10

SFPA Components	Weighting Factor
Transactions (Create, Update, Delete, Report, Read)	4.6
Logical Data Groups (Saves)	7

https://www.ifpug.org/ifpug-acquires-the-simple-function-points-method

Simple Function Point Analysis



Coefficient Statistics Summary

Term	Coef	T-Statistic	P-value
Intercept	5.56	10.67	0.00
SiFP	0.77	8.96	0.00
D1	0.48	2.14	0.05

Goodness-of-Fit Statistics

SE	R^2	$R^2_{(adj)}$	$R^2_{(pred)}$	MAD
0.35	92.00%	90.67%	86.25%	25.93%

Analysis of Variance

Source	DF	Sum of Sq.	Mean Sq.	F-stat
Regression	2	17.39	8.69	69.02
Residual	12	1.51	0.13	
Total	14	18.90		

ID	Model Equation	$R^2_{(adj)}$	$R^2_{(pred)}$	MAD	Rank
1	$E = 935.5xREQ^{0.882}$	88.8%	85.9%	31.2%	3
2	$E = 604.3xISSUES^{0.6879}$	69.3%	59.4%	51.5%	5
3	$E = 1365xSTORY^{0.6228}$	67.9%	59.04%	54.1%	6
4	$E = 206.5xSTY\_PTS^{0.6842}$	83.1%	78.2%	32.6%	4
5	$E = 189.5xUFP^{0.8747}$	89.3%	85.6%	31.6%	2
6	$E = 261.1xSiFP^{0.7708}x1.6^{D1}$	90.7%	86.3%	25.9%	1

- 2022 study of 15 DHS IT systems and 3 DoD IT systems
- "Based on the comparison of effort models, although all models passed the criteria for statistical significance, **simple function points, unadjusted function points, and functional requirements** are stronger predicters to development effort than stories, story points, or issues"
- Simple Function Points produced the highest adjusted R-squared value indicating a very strong predictive capability

"Lets Go Agile: Data-Driven Agile Software Costs and Schedule Models for DHS Projects", ICEAA 2022, Wilson Rosa, Sara Jardine, Kimberly Roye, Kyle Eaton, and Chad Lucas

Sample SFP Analysis

Work Package	Description	Transactions	Data Stores	SFPs	Notes
Registration and management of services	Provide utility to **manage registration and management of services**. This service will listen for registration information such as name, IP address. and TCP/IP ports from services as they come on line. It will also provide this data to services that request it.	4	1	25.4	The description implies the creation of a new process/utility. This would require the ability to Read, Write, Delete, and Change the underlying data for a total of 4 elementary processes. Also, the ability to save/store the data is required. 4 transactions and 1 Data store total. 4 Transactions at 4.6 SFP plus and additional 7 SFP for the Data store = 25.4 SFPs
Big Data AI	Develop machine learning routines to **analyze** post mission data to create post mission **brief**.	2	0	9.2	The machine learning rountines do not involve the user. However, the analysis and creation of a briefing do require user interaction. This would be an update and report transaction, respectively. 2 Transactions at 4.6 SFP per transaction = 9.2 SFPs
Route - C2_ISR	Add additional NOMS Route attributes and other features to reach full C2-ISR level of functionality	0	0	0	Changes/Updates to reach a standard. NFR

Some of the over 1300 “Road Map” elements (basic work packages) have a zero SFP count. Items such as documentation or meeting a certain developmental standard do not require end user interaction and, as such, are not functional. There is, however, effort associated with these requirements as they add complexity to the overall work effort. Most software cost estimating models account for these hours from the parametric estimating equations derived for their historical data base. SEER-SEM estimates are further characterized by the development standards used that tune estimate to capture the required documentation and QA efforts. Non-model users might utilize the Software Non-functional Assessment Process (SNAP) to size the non-functional effort.

Top-Level Estimate Summary

Summary:	SEER Full Agile	T-Shirt Estimating	% Delta
HOURS	2,363,461	1,876,031	20%

- The SEER model used the Functional Size (SFPs) to turn each work package into hours, then costs
  - There was only a 20% delta between the team's estimated hours and the SFP counted work package data
- Estimated labor rates were triangulated to account for the differences in contractor rates, and the contractor/government division of labor
  - Accounting for inflation from the estimates start date in FY19 to FY23, there was only a 17% delta in cost from the t-shirt sizing estimate and the SFP estimate

Summary/Lessons Learned

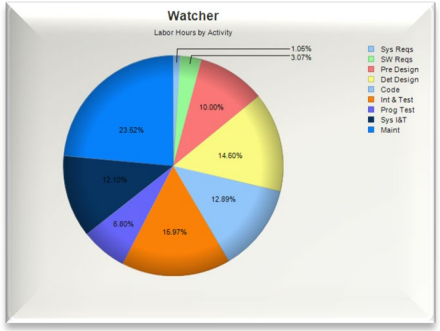
- Sizing remains a key cost driver
- Simple Function Point analysis works well for early agile programs, taking a proactive approach to measuring software mitigates human factors from team actuals
- Simple Function Point analysis: was consistently applied across all elements based on the element description
- The T-Shirt size methodology *may* be inconsistent
  - Dependents largely on team turn over rate and maturity
- At the top-level a 17% difference accounts for risk and provides a dependability to the estimate as a whole
- The functional size could be run through other models
  - Backfiring the hours through DoD standardized tools like NEMO/COCOMO provides a familiar sanity check, thus increasing data validation
- SFP allows for estimation earlier in the decision making process and allows for EVM-like integrated program management
- An analysis of selected elements could be used to begin to build a NAVAIR Function Point/Cost database that would be helpful in more DoD platforms

Backup

SEER Model Subcategories

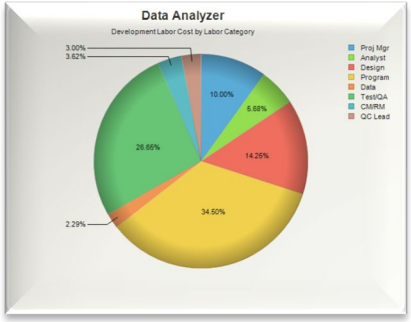
Software Development Activities

- System Requirements Design
  - Create initial system requirements
  - Decide which functions are allocated to software
- Software Requirements Analysis
  - Detailed software requirements analysis and synthesis
- Preliminary Design
  - Subdivide software into packages or functions
  - Define data flows between different program components
  - Map design to software requirements
- Detailed Design
  - Further define software is down to single decision points
- Coding and Unit Test
  - Software programming
  - Unit level testing performed by programmers
- Component Integration and Test
  - Integrate software units into cohesive software components
  - Component-level testing
  - Integrate components into a cohesive program
- Program Test
  - Formal testing to determine compliance with requirements
- System Integration and Test
  - Program-level software-to-software integration
  - Software-to-hardware integration (software labor only; estimate hardware portion in SEER HW)



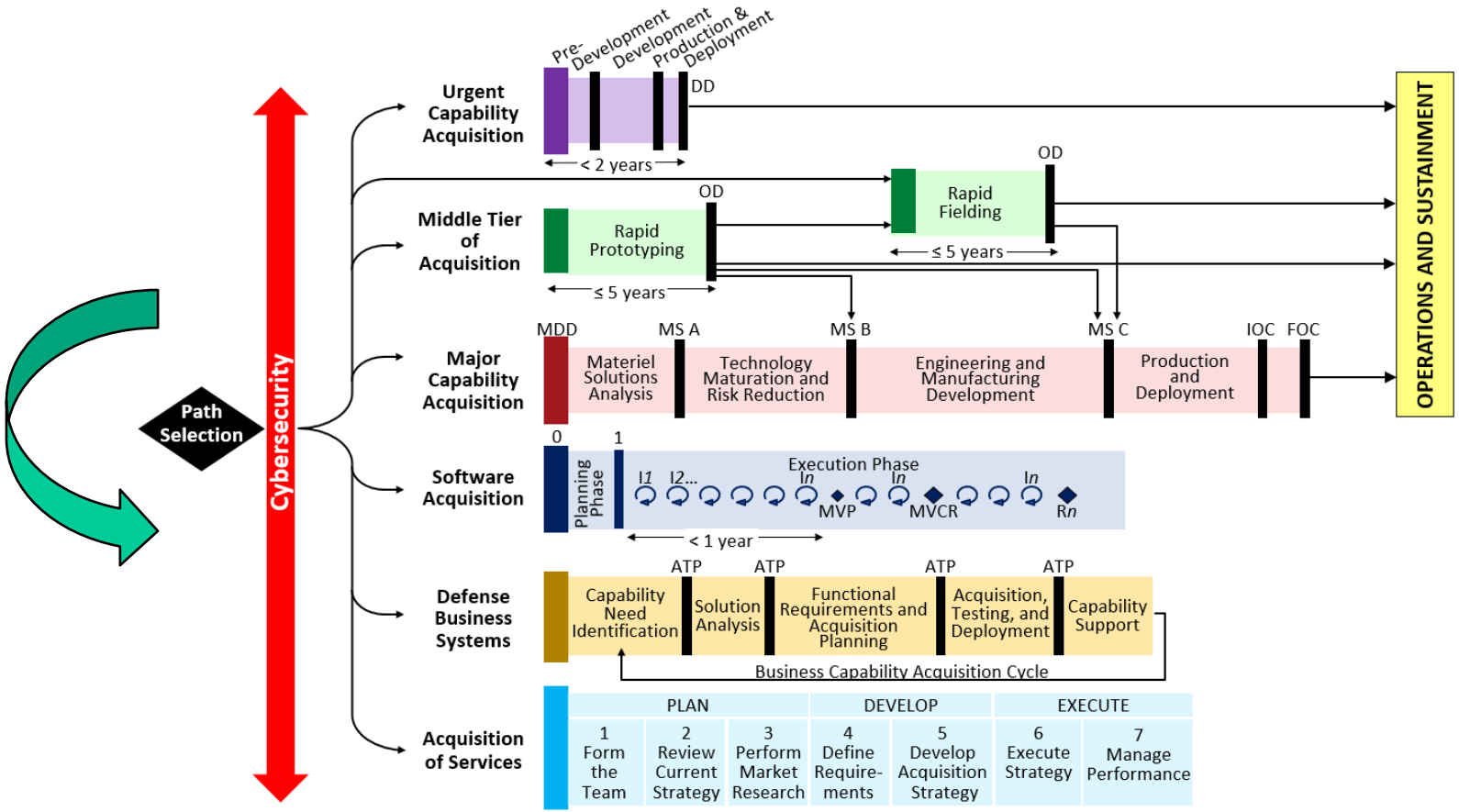
Software Labor Categories

- Management
  - Direct software project (technical) management only
- Business Analyst
  - Develop domain-specific or business requirements and document them
- Architect / Designer
  - Turn requirements into architecture & design documents, and create high level data models
- Data Analyst (Architect)
  - Layout of physical data structures. May validate requirements for correct implementation, perform throughput analysis
- Coders / Programmers
  - Coding, unit testing, maintaining appropriate unit documentation
- Quality Assurance & Test Personnel
  - Prepare test plans and procedures, run tests, and write test reports
- Configuration & Release Manager
  - Configuration identification, change control, status accounting, auditing, and release management
- Quality Control Lead
  - Quality engineering, inspections, audits, and standards certification

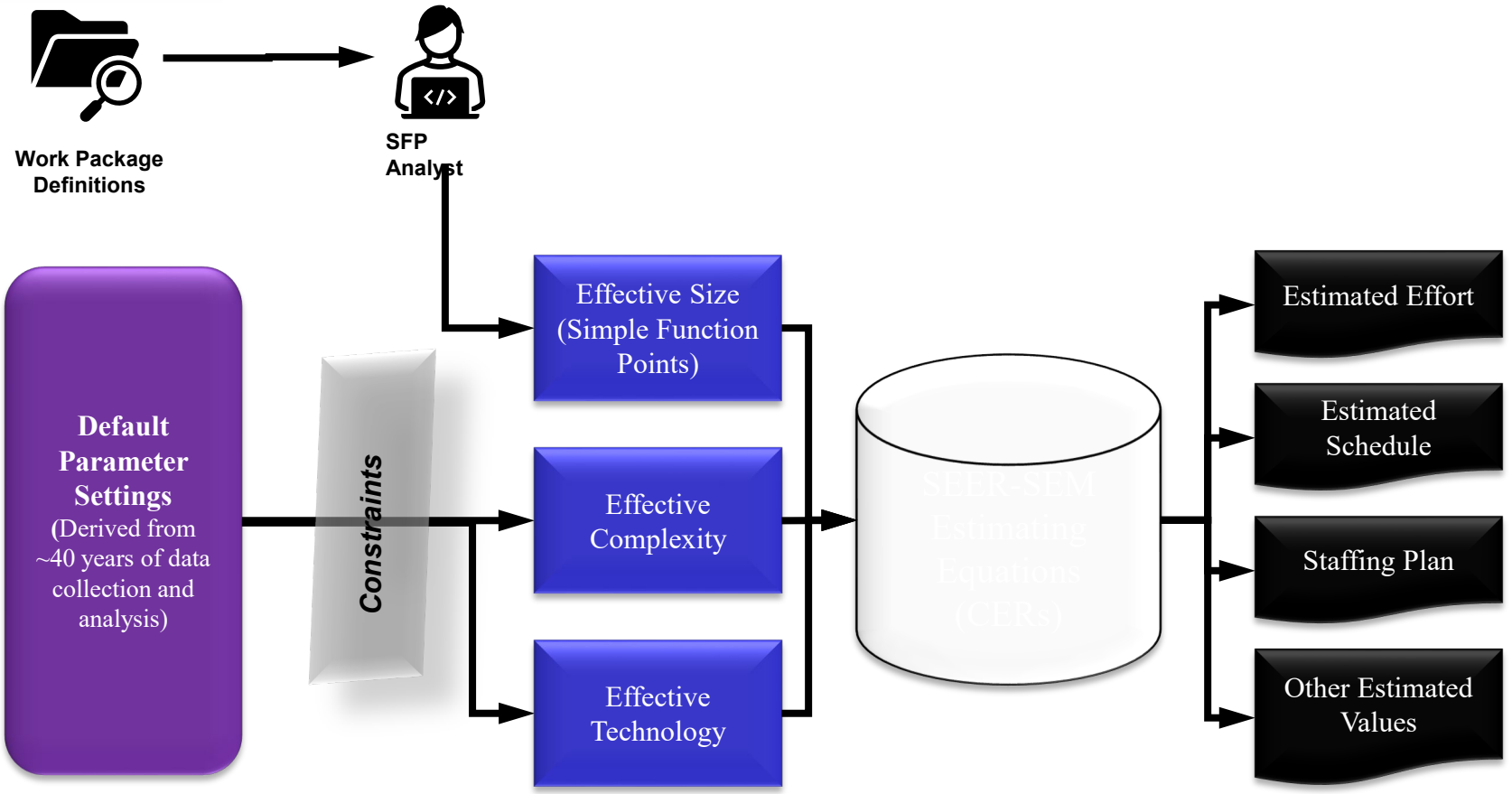


Transition to SW Acquisition Pathway

The purpose of the MTA pathway is intended to fill a gap in the Defense Acquisition System (DAS) for those capabilities that have a level of maturity to allow them to be rapidly prototyped within an acquisition program or fielded, within 5 years of MTA program start.



SEER SEM Overview



SEER-SEM Equations

Mathematical Underpinnings validated over time

Basic Definitions

Key Inputs

Input	Definition
S_e	Effective Size.
C_{te}	Effective Technology.
D	Staffing Complexity.
Pk	Staff Loading, which represents the place on the staffing curve where staff peaks.
F_s	Staff Loading Scale Factor, computed from Pk.

Key Outputs

Output	Definition
K	Life Cycle Effort, or the total area under the staffing curve.
t_d	Schedule to the peak of the staffing curve.

Basic Effort/Schedule Equations

Software Equation: $S_e = C_{te}\sqrt{K}t_d$

Staffing: $M = \frac{F_s K}{\left(Pk\% t_d\right)^2}$

Complexity: $D = \frac{K}{t_d^3}$

Minimum Time Solution occurs when the Complexity Equation and the Software Equation intersect. To do this, first take the software equation and solve for t_d.

Solving the Software Equation for t_d:

$$t_d = \frac{S_e}{C_{te}\sqrt{K}}$$

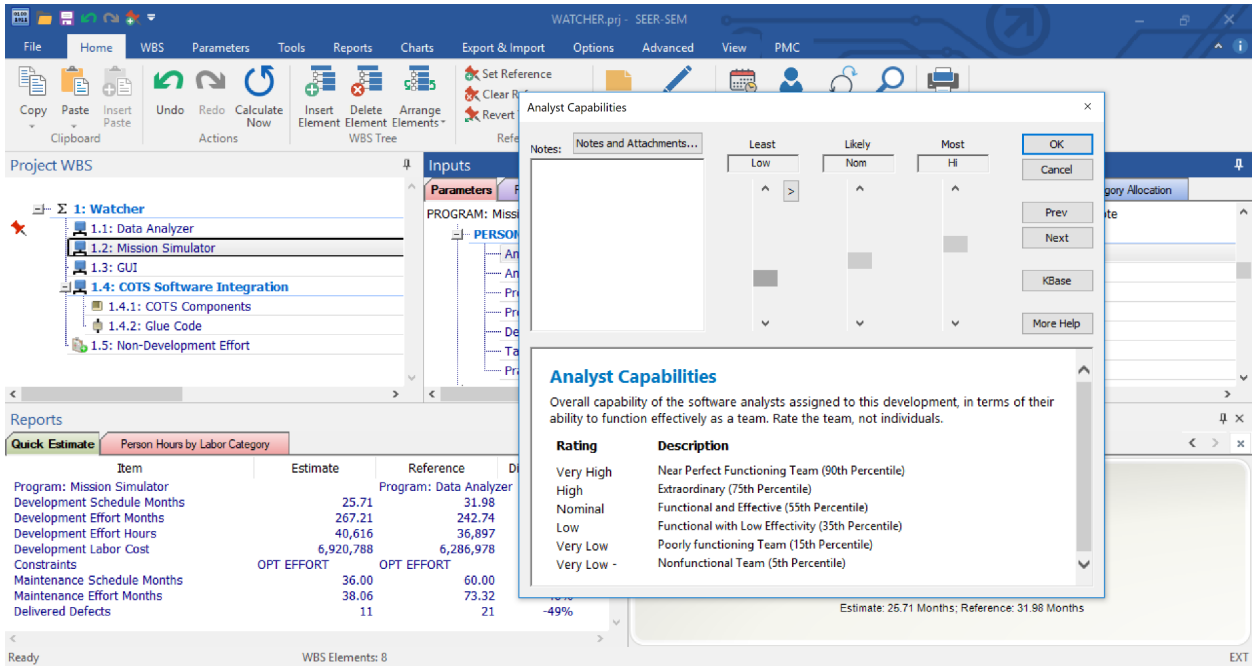
SEER-SEM mathematical foundations are discussed in more detail in the SEER-SEM User's Manual

The specific values of the exponents and their mathematical development are proprietary (thus the "Black Box" issue)

24

Parameter Settings: User Entries

- You may choose to adjust any or all settings, or continue to use the default, knowledge base settings

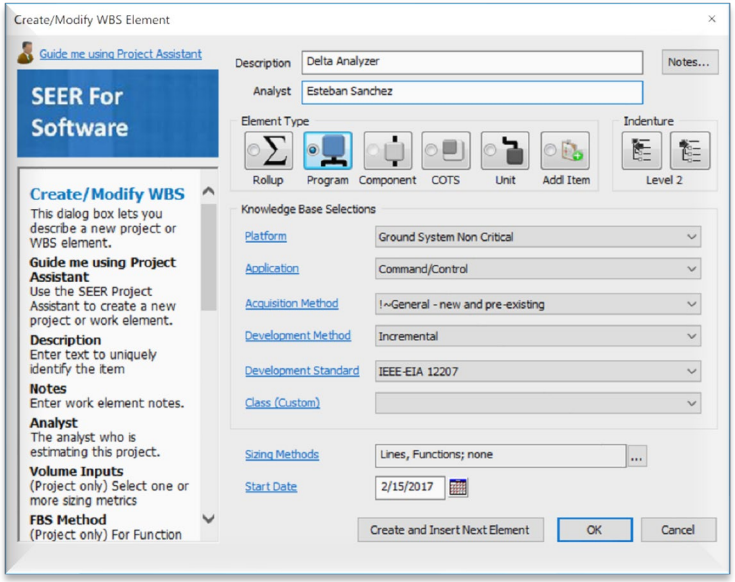


Adjusting parameter settings based on your knowledge of the program, people, processes, etc. will refine the estimate

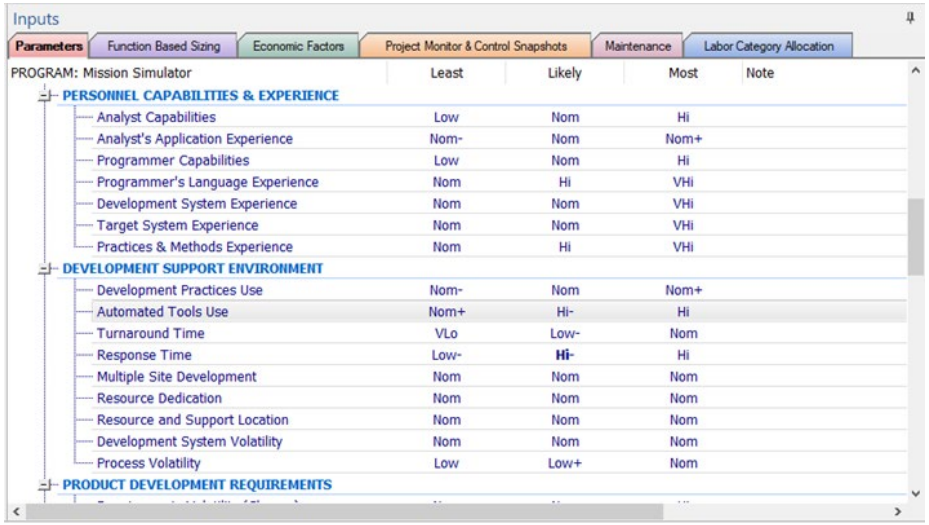
Parameter Settings: Knowledge Bases

- SEER-SEM uses knowledge bases to derive initial default parameter settings based on industry experience
- Choose knowledge bases for:
  - Platform (Ground-based Mission Critical, Client-Server): Characterizes the host environment
  - Application (Command & Control, Electronic Data Interchange): Characterizes a particular application or application technology type
  - Acquisition Method (new, modification, reengineering): Characterizes how the functionality will be delivered (new development, modified software, etc.). Establishes amount of rework estimated for reused software to function correctly
  - Development Method (Incremental, Agile): Characterize the particular SDLC method that will be used
  - Development Standard (IEEE 12207, DO-178C): Characterize a particular software development process standard that will be used

Size and knowledge bases will give you an initial estimate



Determining Complexity, Technology, and Constraints



PROGRAM: Mission Simulator	Least	Likely	Most	Note
PERSONNEL CAPABILITIES & EXPERIENCE				
Analyst Capabilities	Low	Nom	Hi	
Analyst's Application Experience	Nom-	Nom	Nom+	
Programmer Capabilities	Low	Nom	Hi	
Programmer's Language Experience	Nom	Hi	VHi	
Development System Experience	Nom	Nom	VHi	
Target System Experience	Nom	Nom	VHi	
Practices & Methods Experience	Nom	Hi	VHi	
DEVELOPMENT SUPPORT ENVIRONMENT				
Development Practices Use	Nom-	Nom	Nom+	
Automated Tools Use	Nom+	Hi-	Hi	
Turnaround Time	VLo	Low-	Nom	
Response Time	Low-	**Hi-**	Hi	
Multiple Site Development	Nom	Nom	Nom	
Resource Dedication	Nom	Nom	Nom	
Resource and Support Location	Nom	Nom	Nom	
Development System Volatility	Nom	Nom	Nom	
Process Volatility	Low	Low+	Nom	
PRODUCT DEVELOPMENT REQUIREMENTS				

- Complexity, Technology, and Constraint values are derived from parameter settings
- SEER-SEM includes over 50 parameters which describe:
  - Personnel Capabilities and Experience
  - Development Support Environment
  - Product Development Requirements
  - Development Environment Complexity
  - Target Environment
  - Schedule and Staffing
  - Requirements
  - System Integration
  - Software Maintenance