How the Modeling and Simulation (M&S) Product Line Approach Supports Concept Exploration

David R. Pratt, PhD
Robert W. Franceschini, PhD
Robert B. Burch
Science Applications International Corporation (SAIC)
prattda@saic.com

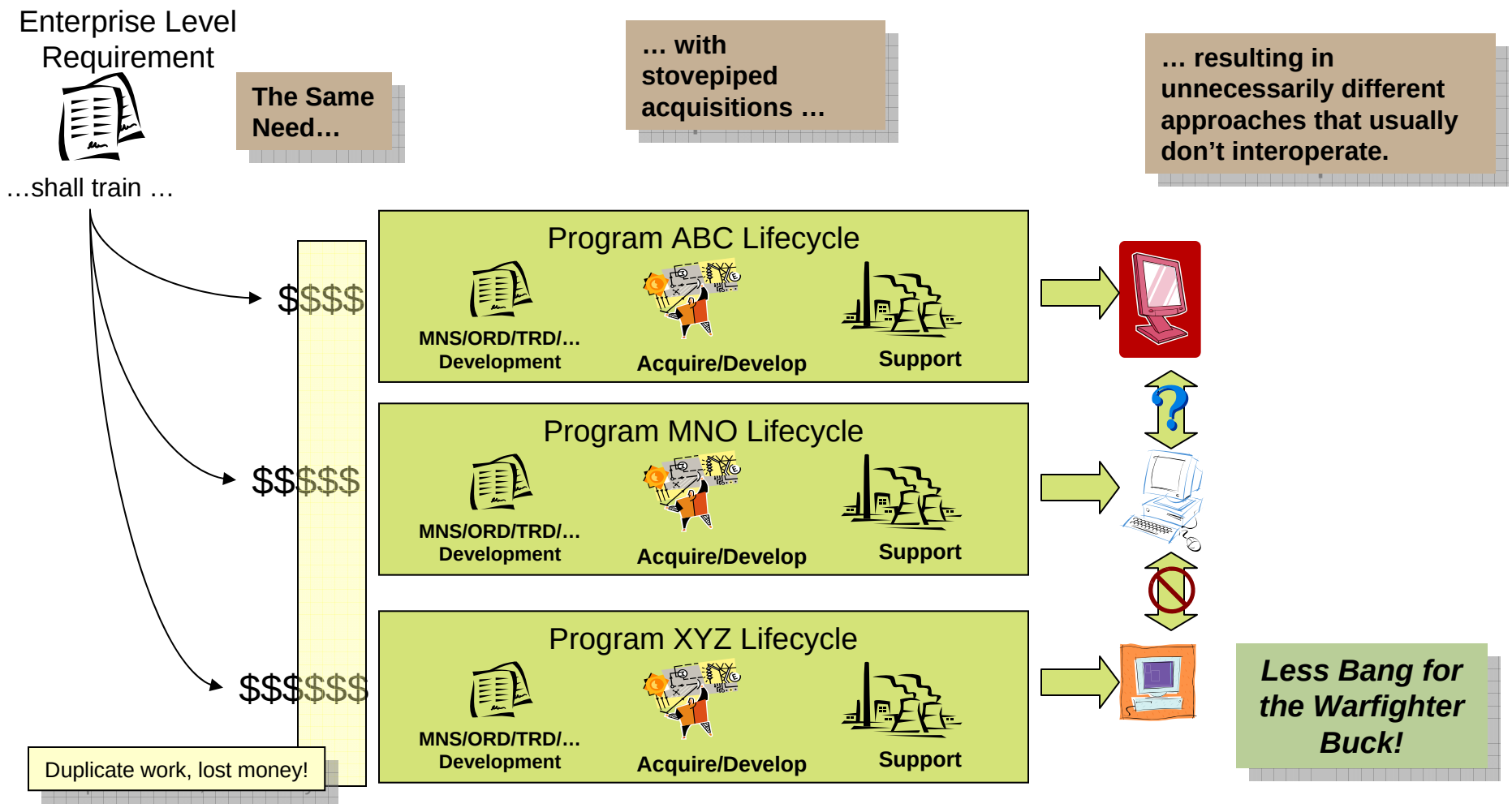
SAIC
From Science to Solutions℠

Agenda

- What is a Product Line
- How does a Product Line Work
- M&S as a System Engineering Tool
- Applications
- Caveats

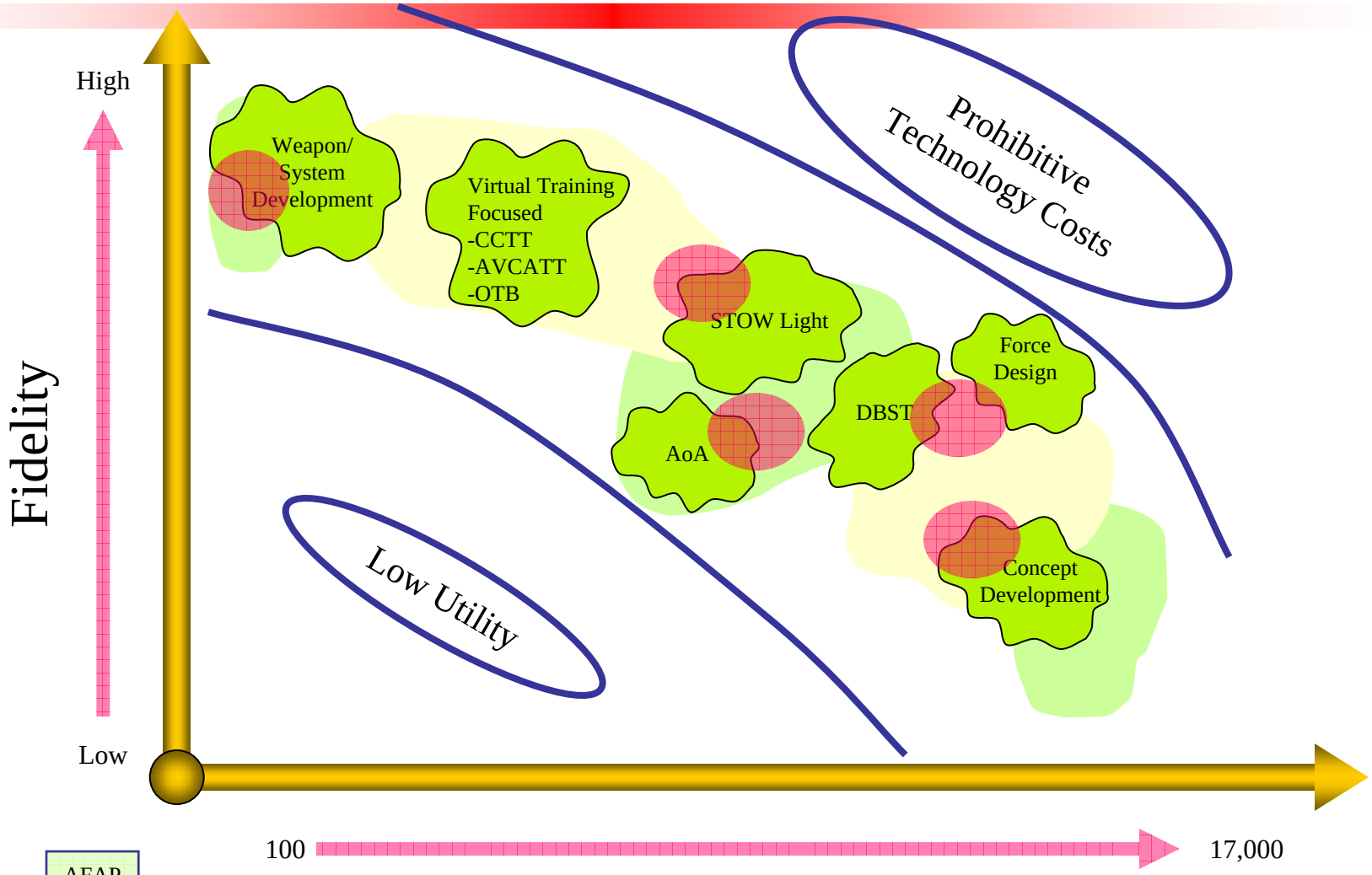
SAIC
From Science to Solutions℠

Classic Cross-Program Acquisition



No Enterprise/System Engineering Across Similar Acquisitions!

Typical Entity Count and Fidelity Characterization



AFAP

Slaved Real-Time

Repeatable

SAIC
From Science to Solutions™

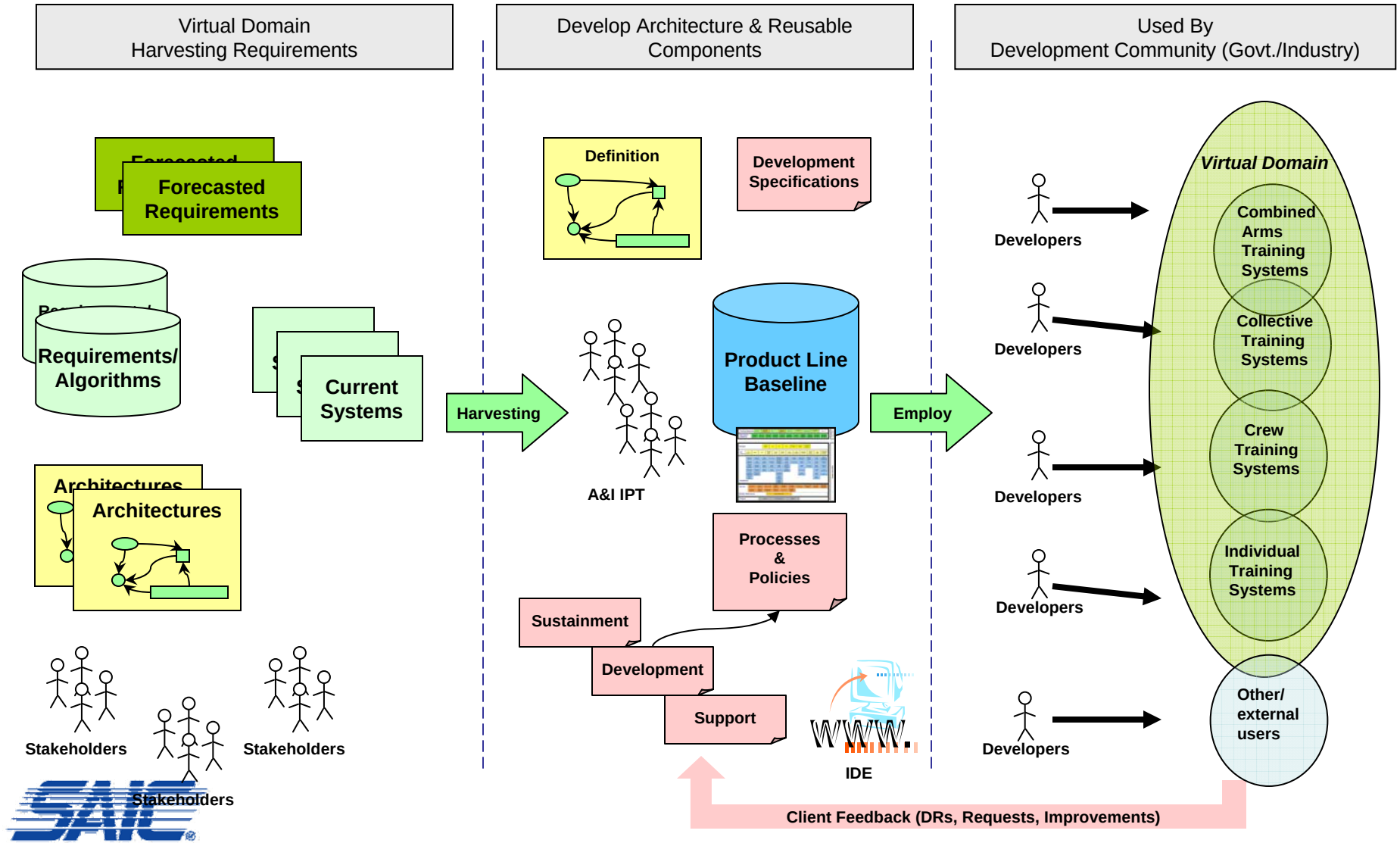
Product Line

- A software product line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.
  - Software Engineering Institute

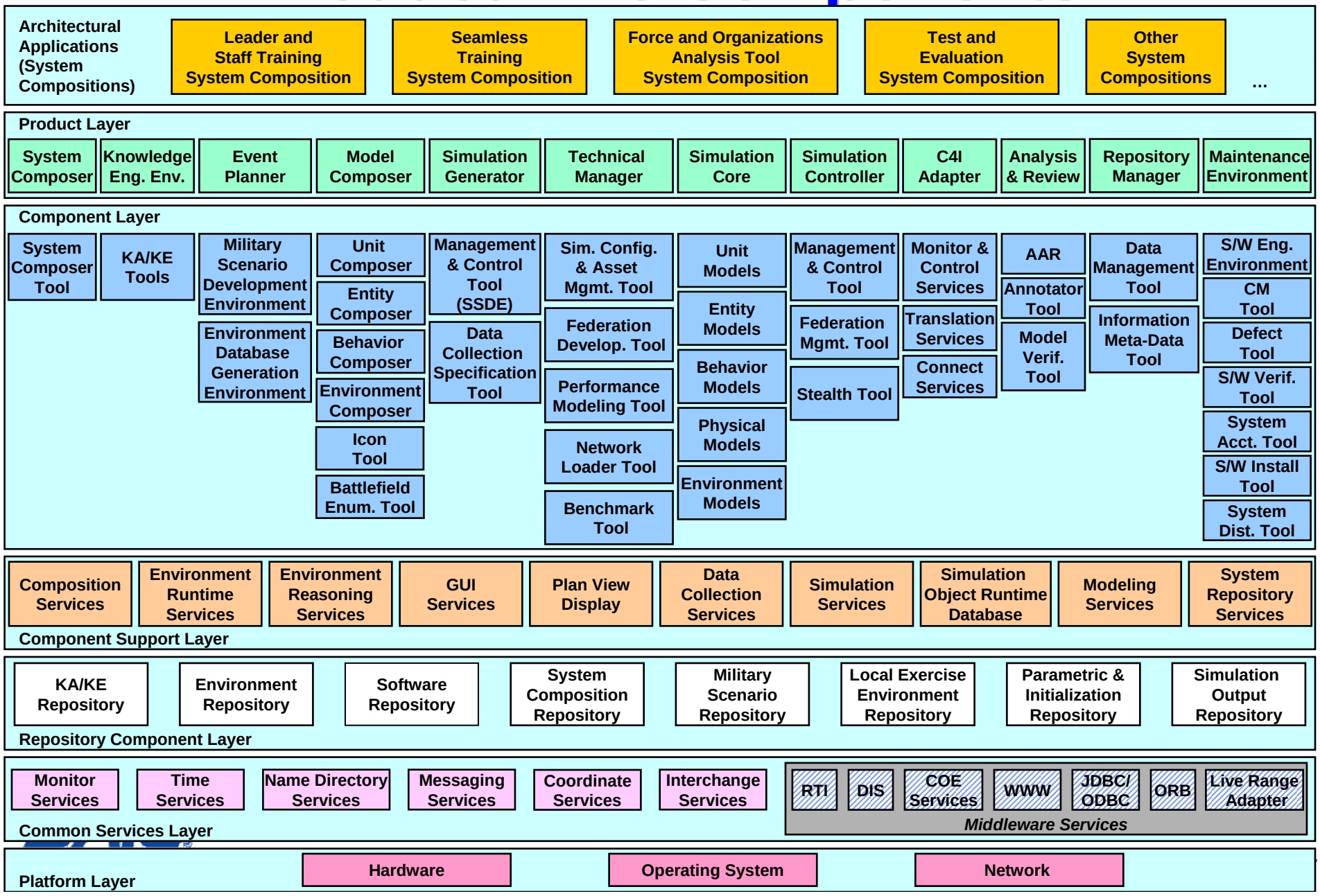
- A collection of interrelated, and possibly redundant, software components that can be brought together to create instances to suit different needs.
  - Me

SAIC
From Science to Solutions℠

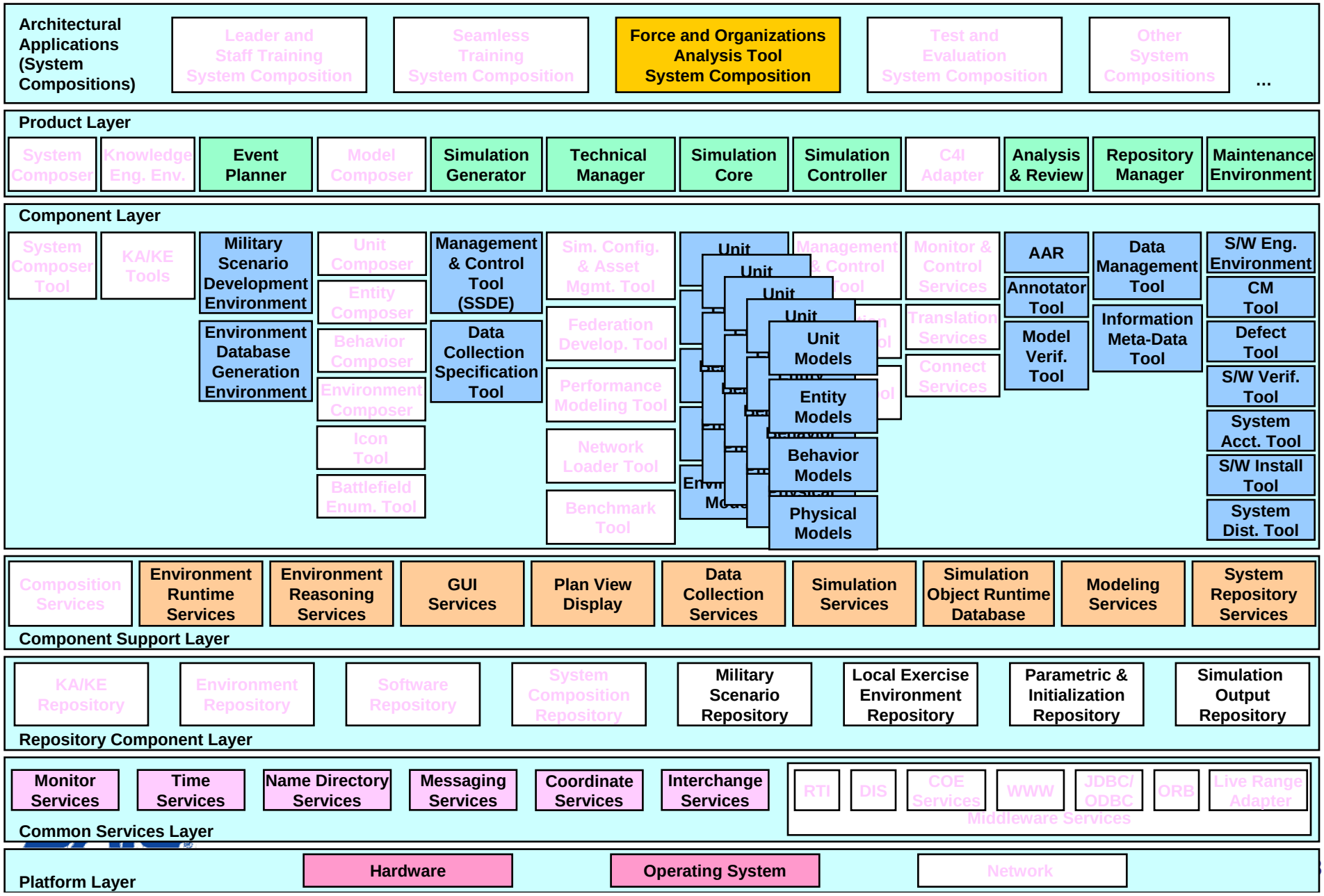
Product Line Architecture Development Process



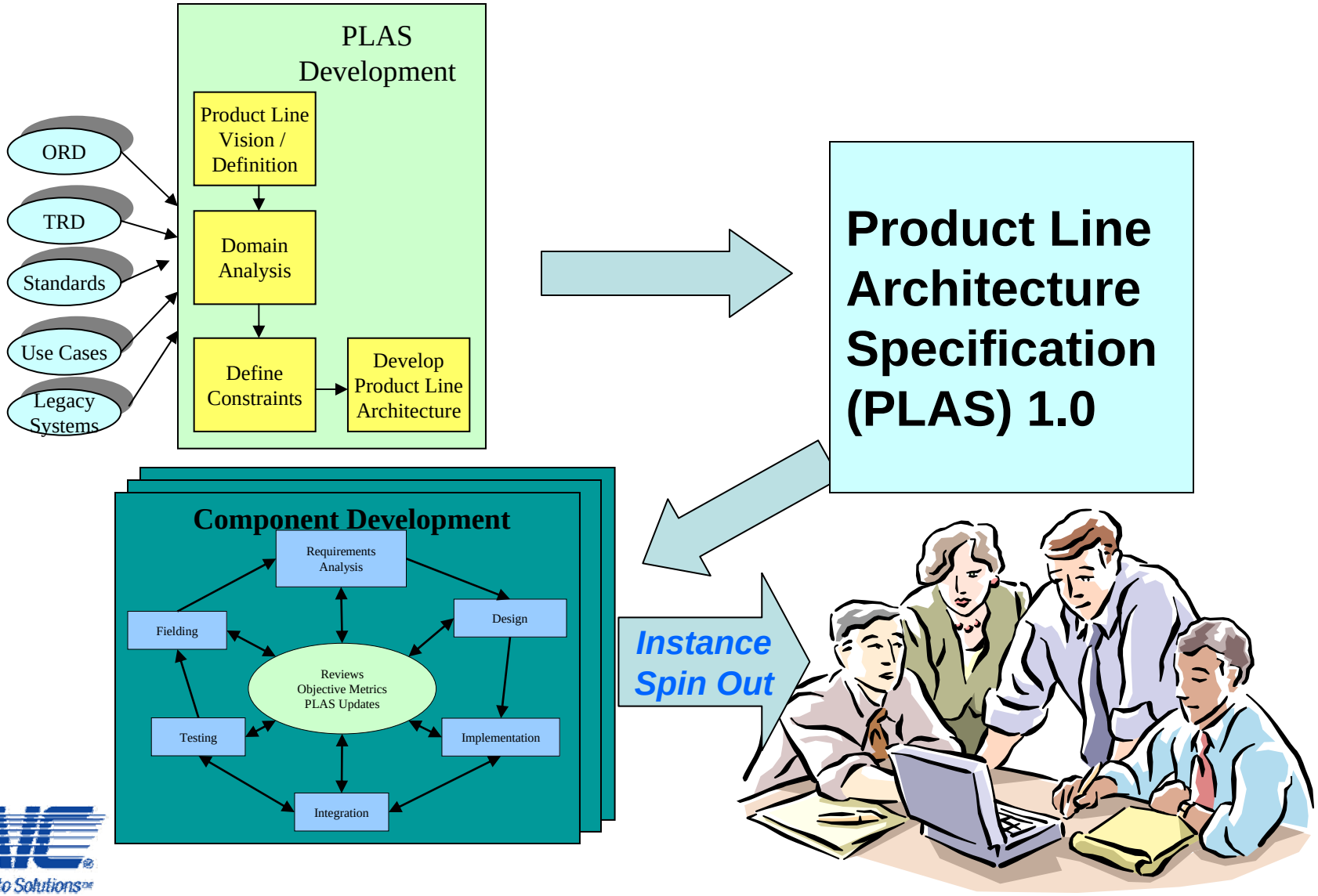
Product Line Components



An Instance of the Product Line

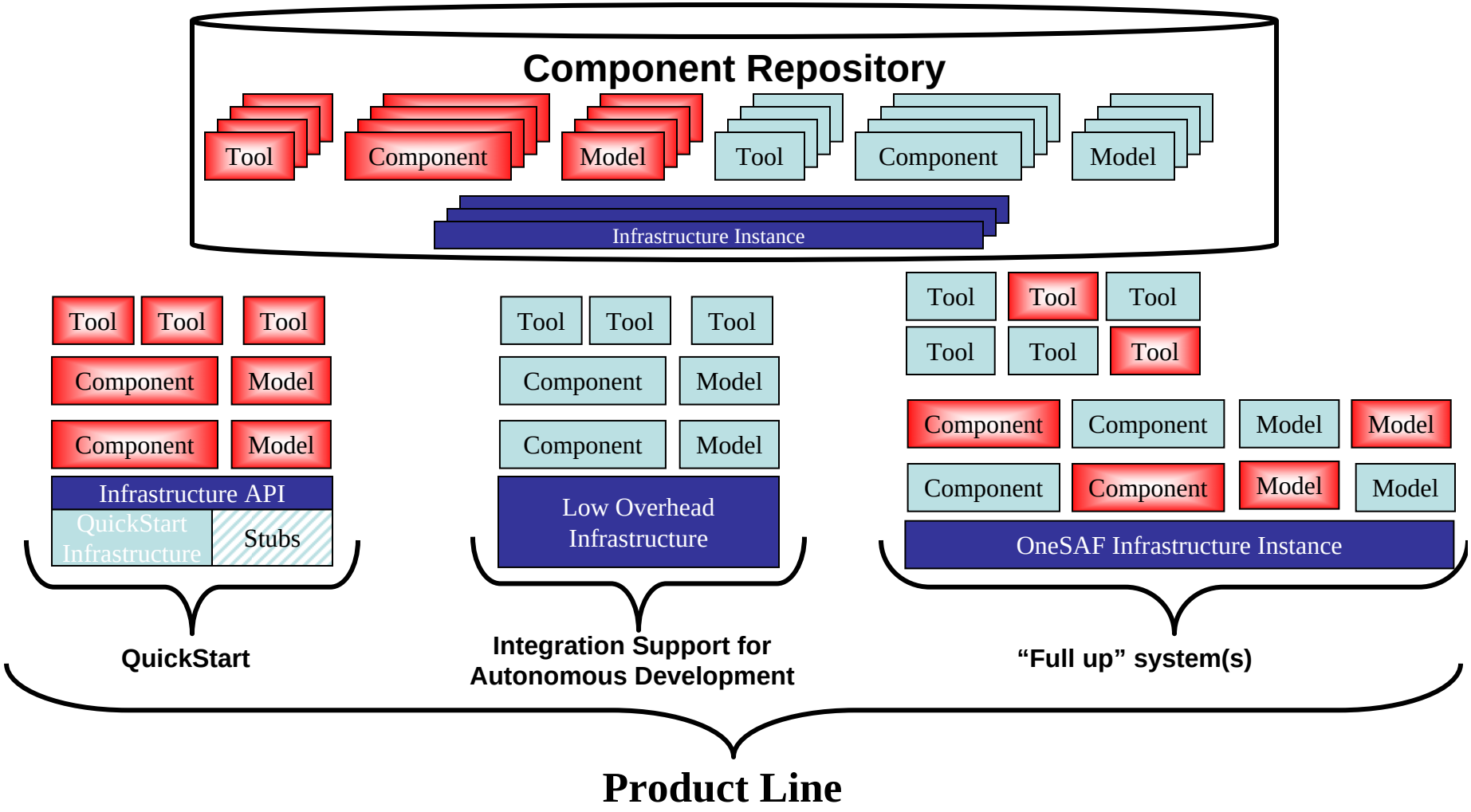


Evolution and Maintenance



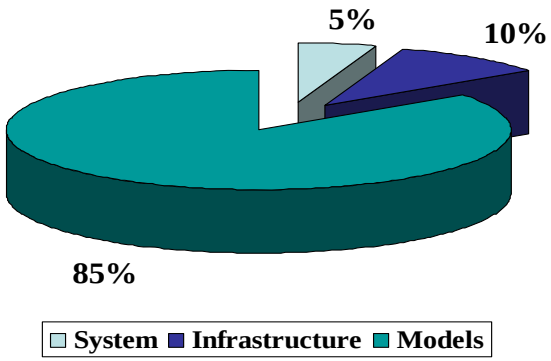
SAIC
From Science to Solutions℠

Multiple Instances from the Same Product Line

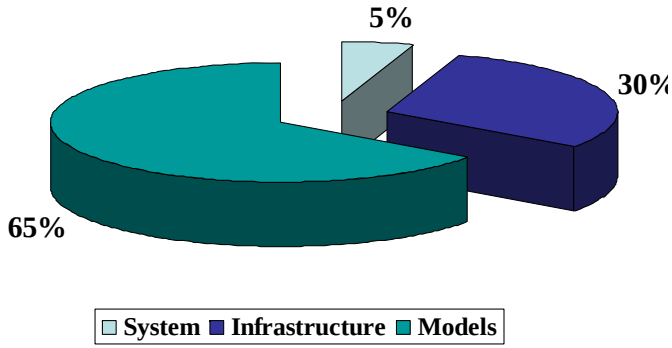


Components can be simulators, emulators, or real devices

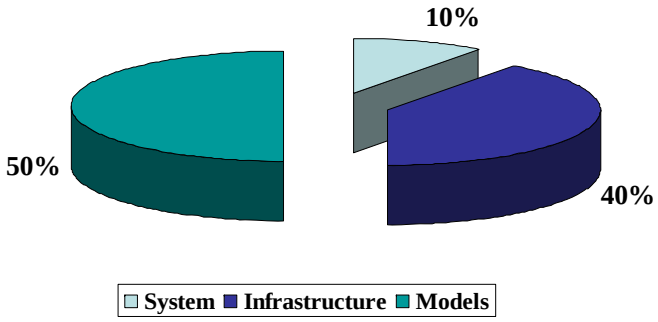
Computational Allocations Vary Per Instance



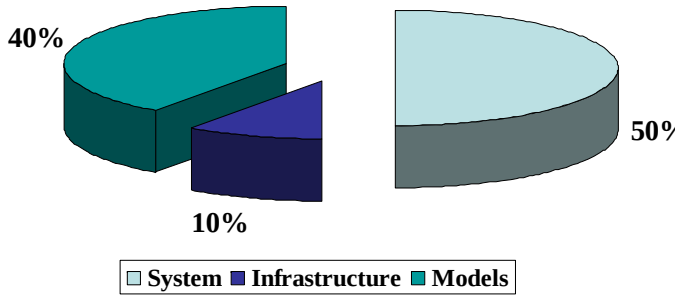
Training Stand Alone



Study Support



Distributed



Different Computational Platform

SAIC
From Science to Solutions℠

Notional Component Allocation Table

Models	% entity contain	# per entity	# models/ processor	# models/ system	Rate (Hz)	Period (sec)	% allocated (rel)	Normalized Allocated (msec)	Total Allocation (msec)	Agent Allocation (msec)	Normalized Allocation (msec)	Number of Events/ Second
Physical Agents												
DynamicsAgent	95%	1	95	1,425	7.00	0.142	41.95%	18.878	81.230	0.855	0.199	665
WeaponAgent	80%	1	80	1,200	5.00	0.200	15.00%	6.750	40.909	0.511	0.084	400
SensorAgent	95%	1	95	1,425	0.33	3.000	10.00%	4.500	409.090	4.306	0.047	31
TurretAgent	60%	1	60	900	5.00	0.200	5.00%	2.250	13.636	0.227	0.038	300
CSSAgent	5%	1	5	75	0.05	20.000	0.01%	0.005	2.727	0.545	0.001	-
CommAgent	95%	1	95	1,425	5.00	0.200	10.00%	4.500	27.272	0.287	0.047	475
VulnerabilityAgent	95%	1	95	1,425	0.05	20.000	0.04%	0.018	10.909	0.114	0.000	4
Subtotal		7	525	7,875			82.00%	36.900	585.773	6.845	0.416	1,875
Entity Behaviors												
Behavioral Agents	95%	6	570	8,550	1.00	1.000	10.00%	4.500	136.363	0.239	0.008	570
PrimaryAgent	100%	1	100	1,500	5.00	0.200	2.00%	0.900	5.454	0.054	0.009	500
Subtotal		7	670	10,050			12.00%	5.400	141.817	0.293	0.017	1,070
Unit Behaviors												
Behavioral Agents	100%	2	200	3,000	0.33	3.000	4.00%	1.800	163.636	0.818	0.009	66
PrimaryAgent	100%	1	100	1,500	0.33	3.000	2.00%	0.900	81.818	0.818	0.009	33
Subtotal		3	300	4,500			6.00%	2.700	245.454	1.636	0.018	99
Totals												
Blackboard	100%	1	100	1,500	-	-	0.00%	-	-	-	-	-
Totals		18	1,595	23,925			100.00%	45.000	973.044	8.774	0.451	3,044

- Ability of the system to meet the computational requirements in addition to the functional requirements
- Interrelationships between the components can result in cascading performance issue

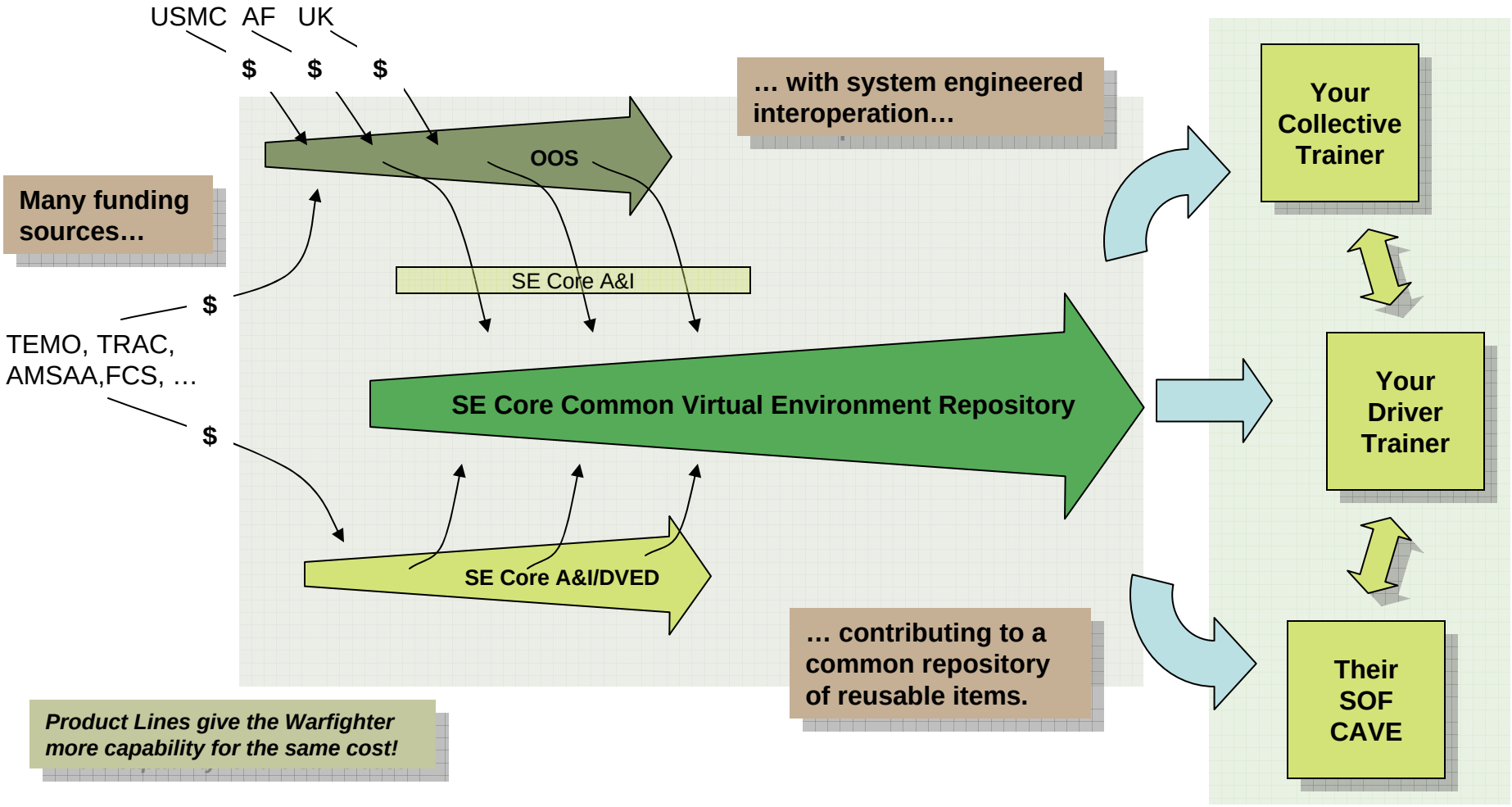
SAIC
From Science to Solutions℠

Updates to the Model

- Network
  - Network throughput, rates, and utilization
  - Interaction verses object update characterizations
- Disk
  - More accurate quantification taking XML into account
  - Sizes of repositories
- Timings
  - Times for specific capabilities
  - An event oriented model
  - Better approach for multi-node
- Structure
  - Must adapt the model to any architectural changes in the system
- Verification
  - Sample real times and compare them to the allocation model

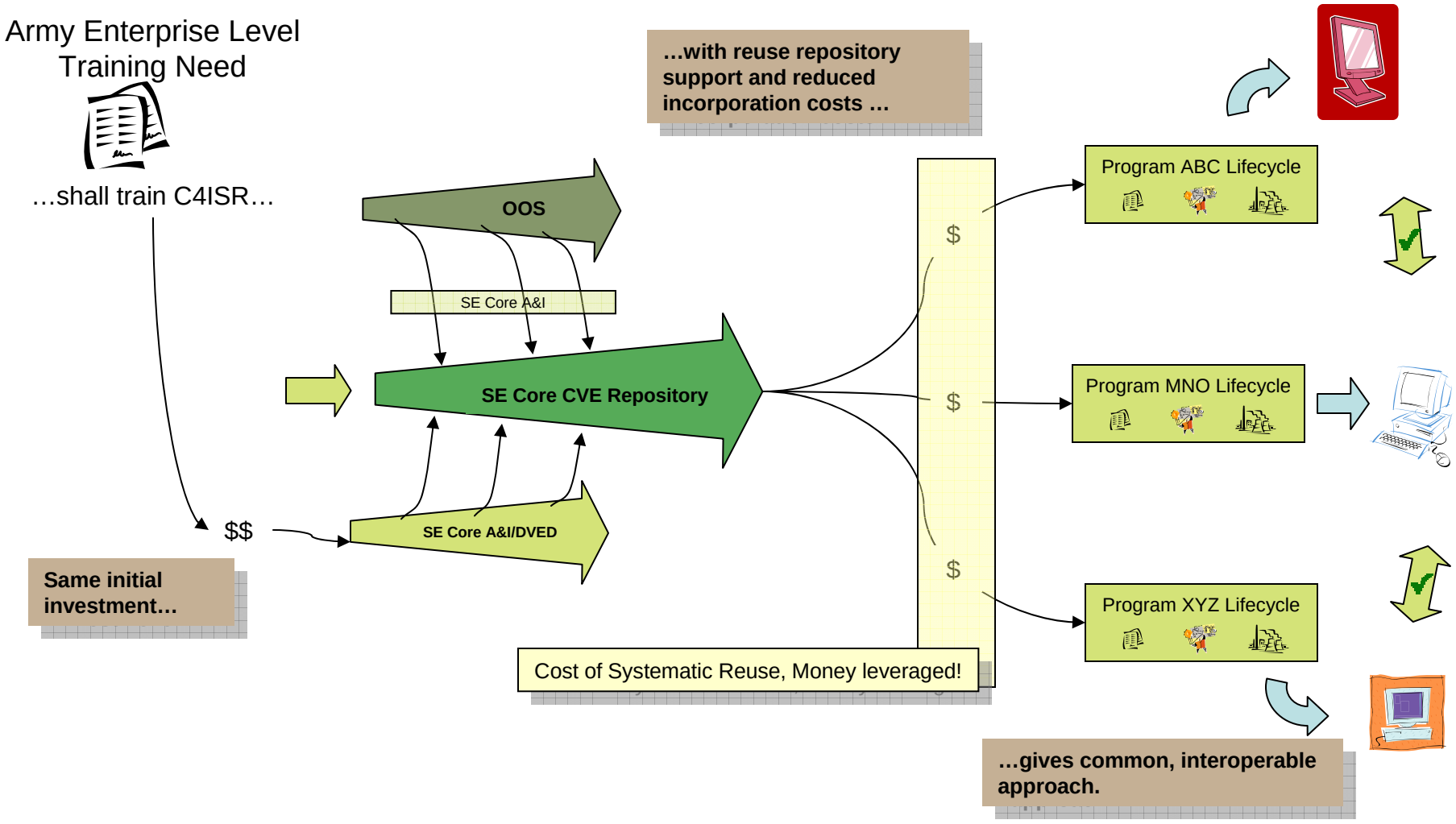
SAIC
From Science to Solutions℠

VSA - Product Line Application



Because Other People's Money Becomes Your Money Too!

C4ISR Example Using SE Core

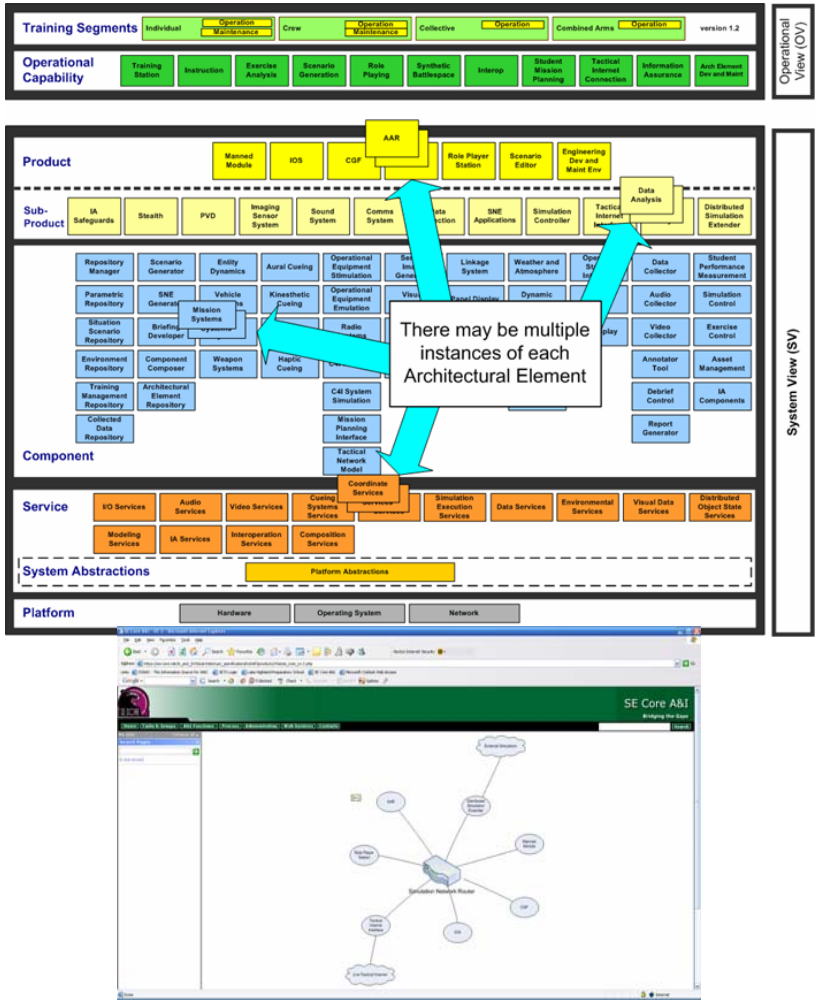


Less Overall Cost, Common System Engineering, Common Support!

From Science to Solutions℠

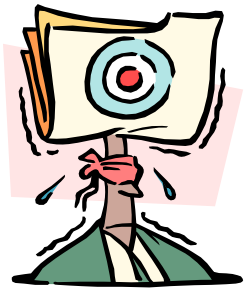
VSA Products

- **Product Line Architecture Framework (PLAF)**
- **Product Line Architecture Specification (PLAS)**
  - **Defines the VSA including components**
  - **Organizes architectural artifacts and documentation**
- **Classical Analysis/Design Artifacts**
  - **DoDAF Product Set**
    - AV-1, AV-2, OV-1, OV-2, OV-5, SV-1, SV-4, TV-1, OV-3, OV-4, SV-2, SV-6, SV-8, SV-9, TV-2
  - **Domain System/Subsystem Specification (SSS)**
  - **Technical Use Cases**
    - High-level trainer operational
    - Identify key system level to drive VSA/CVCs
- **Evolution Plan**
  - **How/when to migrate current programs to VSA**



The Ugly Truth

- The ***Product Line*** is only as good as its inputs
  - The architecture depends on the structure of the system and how thread of control is portioned out
  - The model depends on subjective allocations of relative times (percents) based on legacy systems
  - The longevity is only as good as the configuration control board
- The ***Product Line*** will change!
  - The components allocated today to a capability may be different in the future
  - It won't change that much, more like +/- 10% not +/- 50%
  - Usually it changes because of structure changes or misallocation of relative time and new requirements
    - Usually because of something the user doesn't know
- The ***Product Line*** is only viable when it is accessible
  - The updated component listing should be maintained on the repository web page
  - The updated performance model should be maintained on the performance web page
- The ***Product Line*** will need to manage the change
  - Repository updates
  - Performance Model updates
  - Structure changes or additions
  - Rate changes
  - Data rates and sizes



SAIC
From Science to Solutions℠