Considering Alternative Strategies for Value Sustainment in Systems-of-Systems

Nicola Ricci, **Adam M. Ross**, Donna H. Rhodes, Matthew E. Fitzgerald

IEEE SysCon 2013

April 17, 2013

Orlando, FL

Outline

- Motivation
- Value Sustainment Strategies
  - Self-recovery
  - Change
  - Evolve
- Era Analysis
- Maritime Security SoS Case Study
- Discussion

© 2013 Massachusetts Institute of Technology

MOTIVATION

© 2013 Massachusetts Institute of Technology

Continually Changing World

The world in which Systems Engineers practice has undergone a significant metamorphosis over the past twenty years

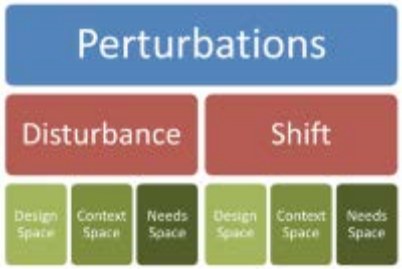
- *Advent of the internet* → great increase in amount of resources available
- *Information travels at the speed of light* → instantaneous communication
- *High-speed computation* → Performance of very complex analyses

Systems are subject to highly **dynamic operational environments**

- A multitude of exogenous uncertainties can impact a system
  - *Geo-political shifts* (e.g., policy/regulation changes)
  - *Disruptive technologies* (e.g., advent of GPS)
  - *Market variations* (e.g., price &demand variations)
- Unanticipated shifts in stakeholder needs
  - Change of *preferences*
  - Change of *mission objectives*

PERTURBATION

"Unintended state change in a system's design, context, or stakeholder needs that could jeopardize value delivery"
(Beesemyer, 2012)



RISK v. OPPORTUNITY

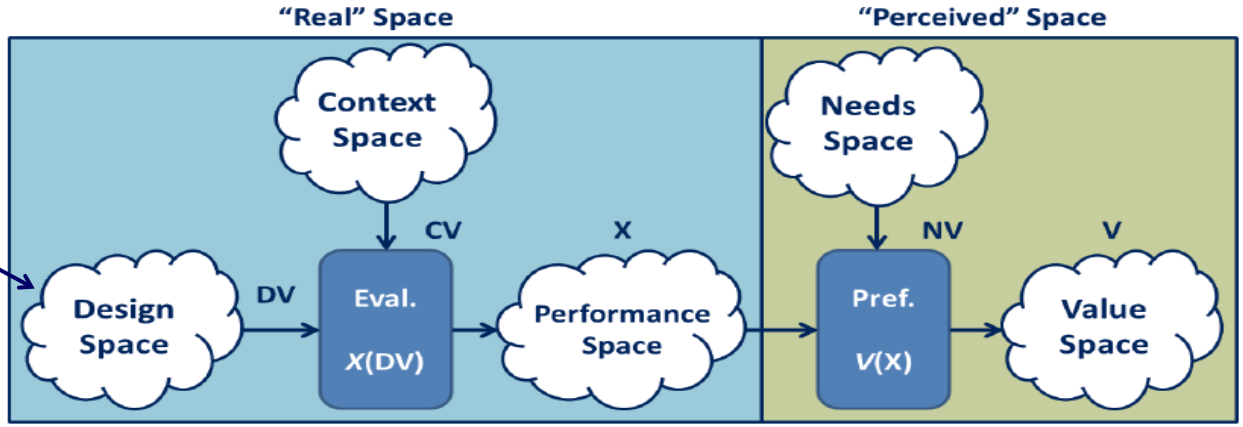
If focused solely on the present state of the world, engineers may encounter the problem of designing systems that may not be well positioned in the future, forced to operate in contexts for which they were not conceived, and delivering capabilities no longer of interest to stakeholders.

© 2013 Massachusetts Institute of Technology

Value Sustainment

- The purpose of any system is to provide some level of value to its stakeholders
  - A **fighter jet** may provide value to its user by exerting force on enemies or serving as deterrent
- The interest of the stakeholders is that system keep providing value throughout their expected lifecycle, despite the *occurrence of perturbations*
  - Context changes: perturbations like **jamming** or **thunderstorm** can affect the value delivery of the fighter jet
  - Needs changes: a fighter jet is suddenly expected to be **undetectable by radar**

- overdesign
- include options



(Beesemyer, 2012)

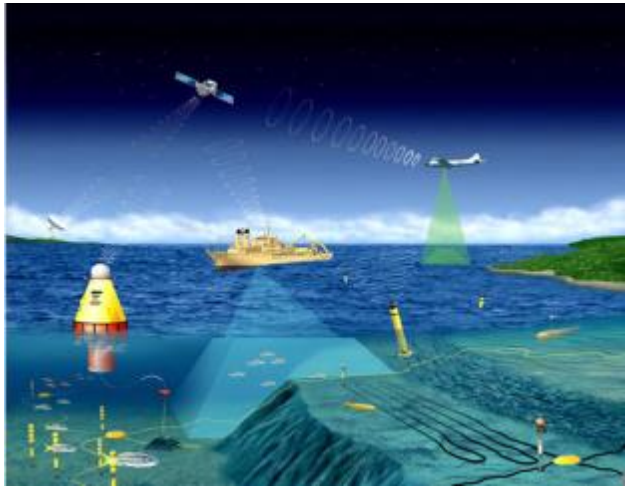
- System designers attempt to make systems that can provide value in spite of these perturbations → i.e., devise value sustainment strategies
  - They can design an **overly** stable aircraft for any thunderstorm or include the **option** of a thunderstorm-targeted control system the pilot can activate

System designers can devise value sustainment strategies to cope with external and needs-related change

© 2013 Massachusetts Institute of Technology

Systems of Systems

- Five characteristics that distinguish SoSs from traditional systems (Maier, 1998)
  - Operational Independence of Elements
  - Managerial Independence of Elements
  - Evolutionary Development
  - Emergent Behavior
  - Geographic Distribution
- Traditional SE v. SoSE
  - Activities performed vary – SoSE more dynamic
- Four different types of SoSs (Dahmann et al., 2008)
  - **Directed**: built and managed to fulfill specific purpose (*focus of this paper*)
  - **Acknowledged**: constituent systems retain independent ownership/objectives/funding
  - **Collaborative**: constituent system interact to fulfill agreed upon purpose
  - **Virtual**: lack of central authority and agreed upon purpose



Most Systems of Systems do undergo periodic re-architecting – to respond or anticipate changes in needs and/or context – but this does not necessarily occur at the "speed of need".

© 2013 Massachusetts Institute of Technology

The Need to Cope with Change

The question for architects in directed SoSs, who are making system-level decisions during SoS inception or during a periodic re-architecting activity, is:

What value sustainment strategies will be most appropriate?

→ What changes will prove to be unnecessary?
→ What necessary changes should occur with greater ease and at less cost?

This paper proposes a framework for evaluating and comparing different value sustainment strategies over the lifecycle of the SoS

→ During (re)design, architects must think of ways to appropriately respond to perturbations
→ **Mitigate risk** (downside uncertainty)
  - Ex.: Increase UAV altitude in face of enemy attack
→ **Exploit opportunities** (upside uncertainty)
  - Ex.: Adding new UAV type to leverage emerging technologies

"*In my opinion the two things most adverse to good counsel are **haste** and **passion**"* (Thucydides, ~400BC)

SoS Engineering is a dynamic process, where the identification of salient strategies for dealing with external changes can significantly benefit engineers and architects

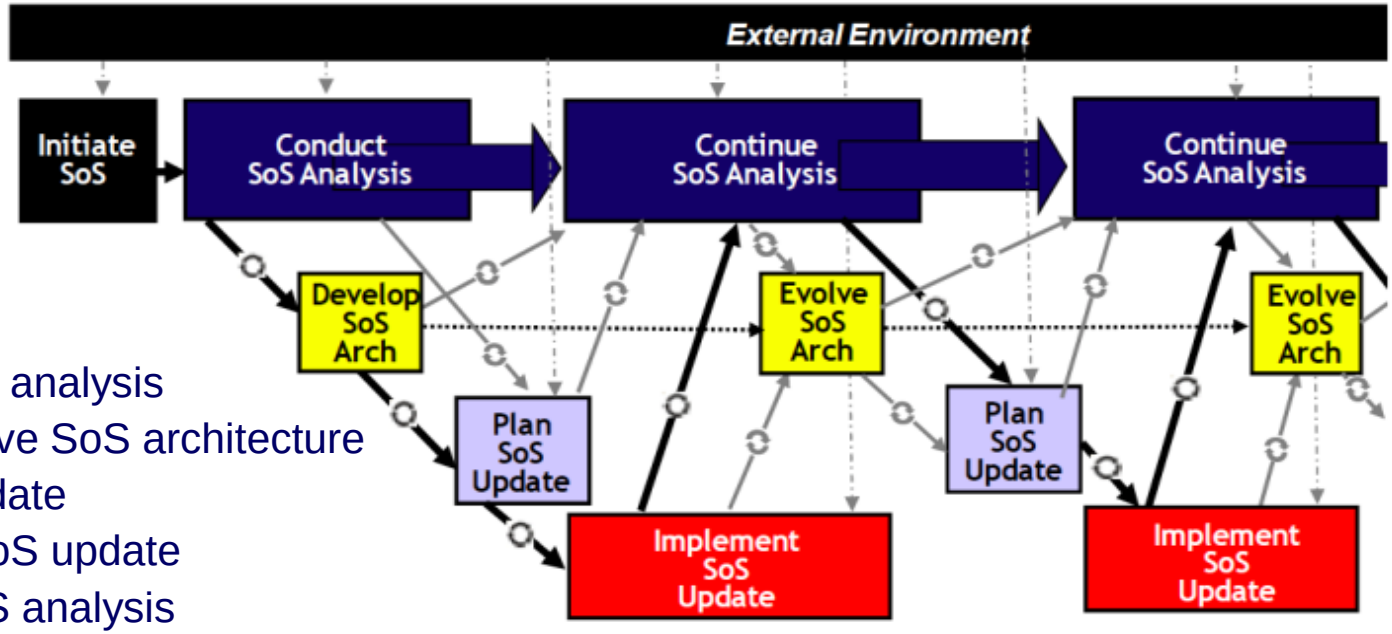
© 2013 Massachusetts Institute of Technology

VALUE SUSTAINMENT STRATEGIES

© 2013 Massachusetts Institute of Technology

Wave Model

- Time-sequenced implementation of the SoSE process
- Wave model* reflects SoS attributes
  - Multiple overlapping iterations of evolution (**incremental development**)
  - **Ongoing analysis** (unlike traditional systems)
  - Continuous input from **external environment**
  - Architecture **evolution**



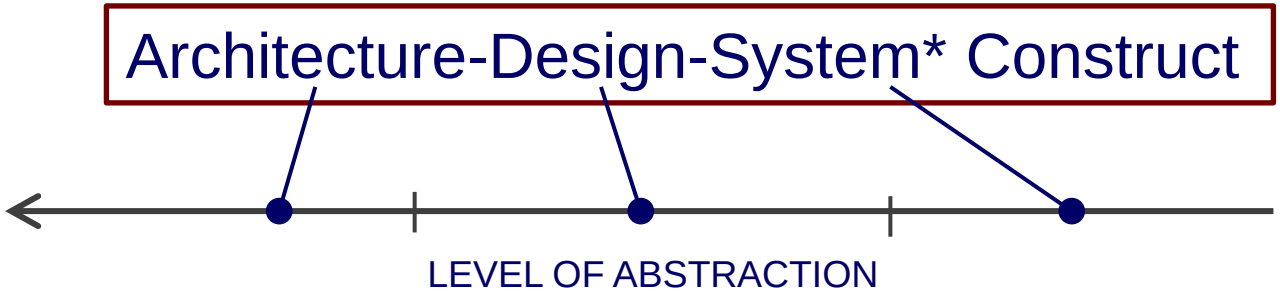
- Main elements
  - Initiate SoS
  - Conduct SoS analysis
  - Develop/evolve SoS architecture
  - Plan SoS update
  - Implement SoS update
  - Continue SoS analysis

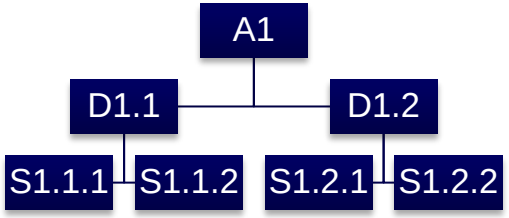
The wave model incorporates the dynamic nature in which SoS Engineering is performed

*Dahmann et al., 2011

© 2013 Massachusetts Institute of Technology

Design v. Architecture



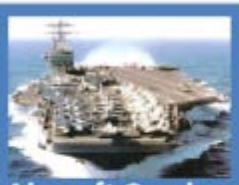


Architecture

- Defined by a set of characteristics that SoS architects deem fundamental to specifying allowed forms, functions and behaviors of the SoS

Design

- Specific instance of parent architecture
- Changing between design instances within an architecture is usually less challenging and time-consuming than changing from one architecture to another

Concept	iPhone	 Aircraft Carrier	Int. Space Station
Architecture	[iPhone 3G/ iPhone 4/...]	[Nimitz/Ford/ Forrestal/...]	The International Space Station
Design	[4/4S; Black/White; 16GB/32GB/64GB]	[USS Ronald Reagan/ USS Harry Truman/ USS George Bush/...]	
System	Clark's Black 32GB iPhone 4S		

*Beesemyer, 2012

Changes in design and changes in architecture correspond to different levels of effort and impact on value delivery

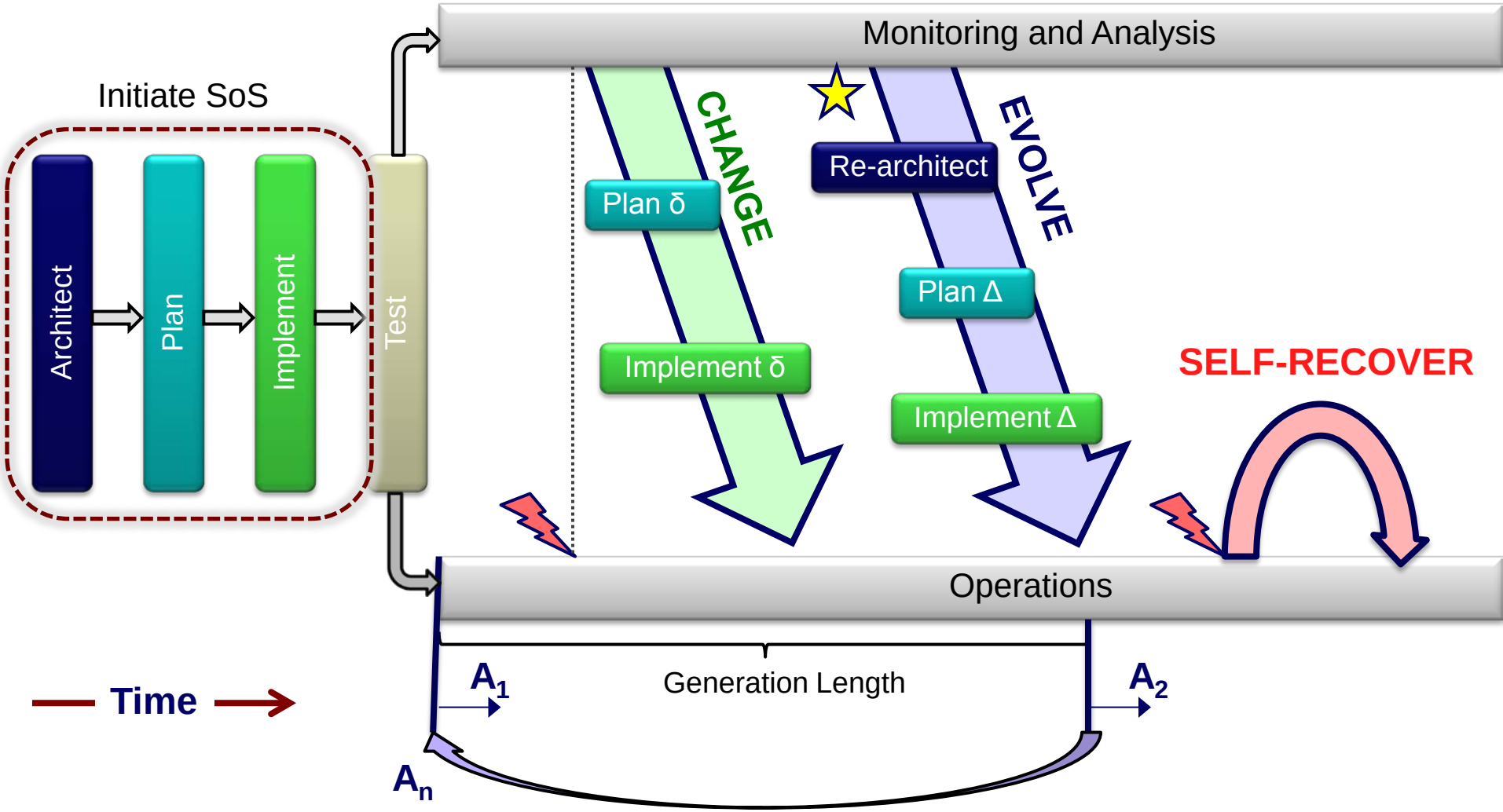
© 2013 Massachusetts Institute of Technology

= Perturbation

= Utility-enhancing opportunity

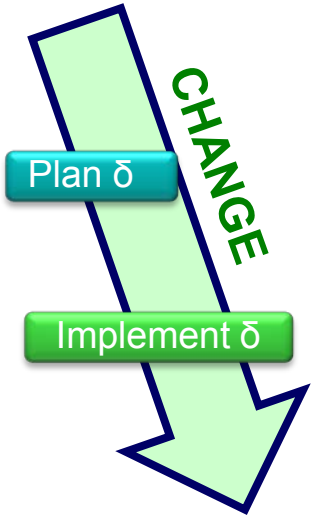
A_n = n^{th} Architecture

Change Type:
δ = Change in design (within pliable set)
Δ = Change in architecture

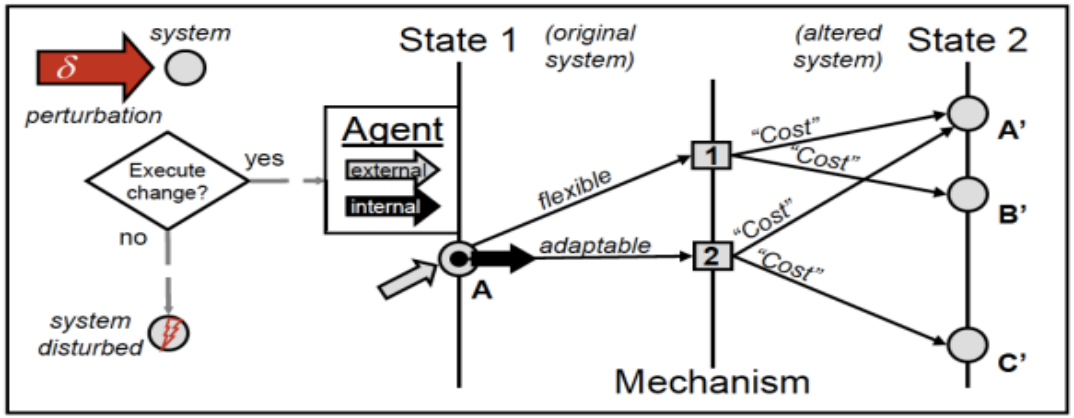


© 2013 Massachusetts Institute of Technology

Value Sustainment Strategy #1 Change



- Related ility: **CHANGEABILITY**
  - *ability of a system to alter its form—and consequently possibly its function—at an acceptable level of resource expenditure*
  - Different types: flexibility, adaptability, scalability, modifiability
  - Agent-Mechanism-Effect framework[Ross and Rhodes, 2008]
    - **Agent** → instigator
    - **Mechanism** → enabler
    - **Effect** → change of state
  - Enabled by inclusion of change options (path enabler + mechanism)

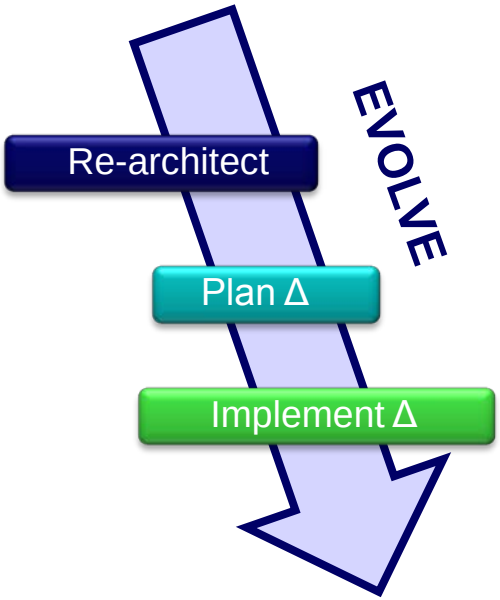


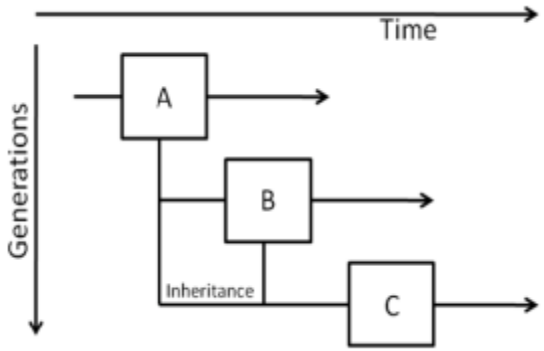
- Examples of changeability
  - Having a *modular payload bay enables swapping the payload*, and thus be changeable when facing different needs
  - High-altitude UAVs enables the possibility of changing to *avoiding physical attacks*

"CHANGE" strategy → changes in **design**

© 2013 Massachusetts Institute of Technology

Value Sustainment Strategy #2 Evolve





Evolution of SoSs over time

- Related ility: **EVOLVABILITY**
  - *The ability of an architecture to be inherited and changed across generations [over time]**
  - Design principles for evolvability

Design Principle	Implications for Evolvability
Targeted Modularity	Limits change propagation (Hansen 2003) (Holtta-Otto 2005)
Integrability	Compatibility and common interfaces (Fricke and Schulz 2005)
Scalability	Of a parameter or entire system to meet new needs (Fricke and Schulz 2005)
Decentralization	Distributed resources to limit effect of changes (Fricke and Schulz 2005)
Redundancy	Gives flexibility to designer to eliminate components (Fricke and Schulz 2005)
Reconfigurability	Self similar parts and maximizing information reconfiguration (Siddiqi and de Weck 2008)
Leverage Ancestry	Successful design choices from all prior generations (Kelly 2010)
Mimicry	Successful design choices from other systems/domains (Kelly 2010; Henderson and Clark 1990)
Disruptive Architectural Overhaul	Upgrading large aspects of architecture at a time (Kelly 2010; Henderson and Clark 1990)
Resourceful Exaptation	Repurposing successful design choices from other systems (Kelly 2010)

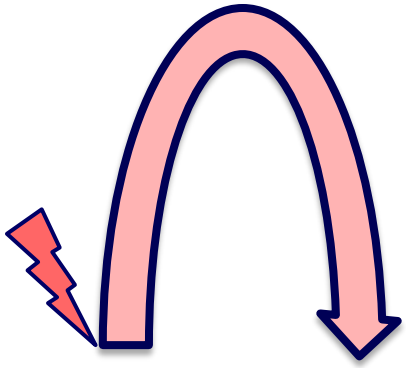
*(Beesemyer, 2012)

"EVOLVE" strategy → changes in **architecture**

© 2013 Massachusetts Institute of Technology

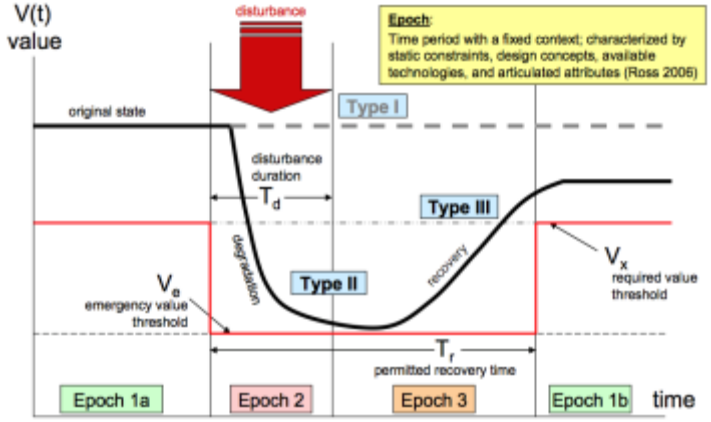
Value Sustainment Strategy #3 Self-Recover

SELF-RECOVER

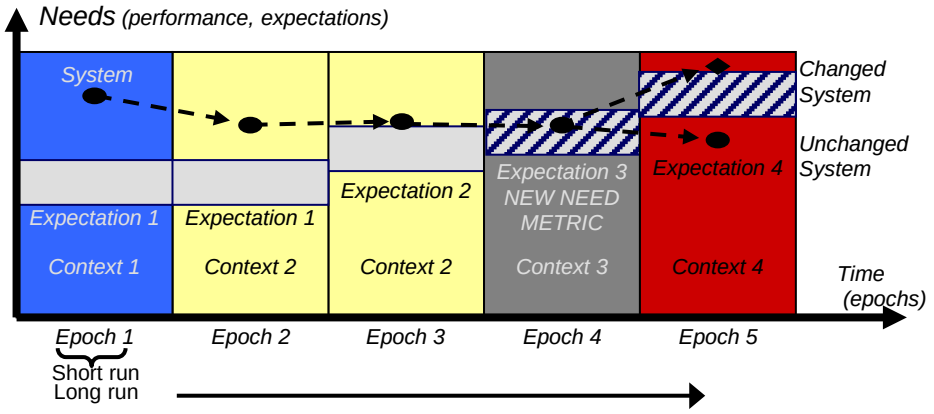


- Related ilities: **SURVIVABILITY (1)** & **ROBUSTNESS (2)**
  - **(1)** *ability of a system to minimize the impact of a finite duration **disturbance** on value delivery*
  - **(2)** *ability of a system to maintain value delivery in spite of long-term changes [**shifts**] in needs or context*
  - Enabled by inclusion of resistance options (path inhibitor + resistance mechanism) and change options (path enabler+ change mechanism)

X = bad state
A,B = acceptable state



An **epoch** is a discrete time period where *expectations* and *context* are fixed



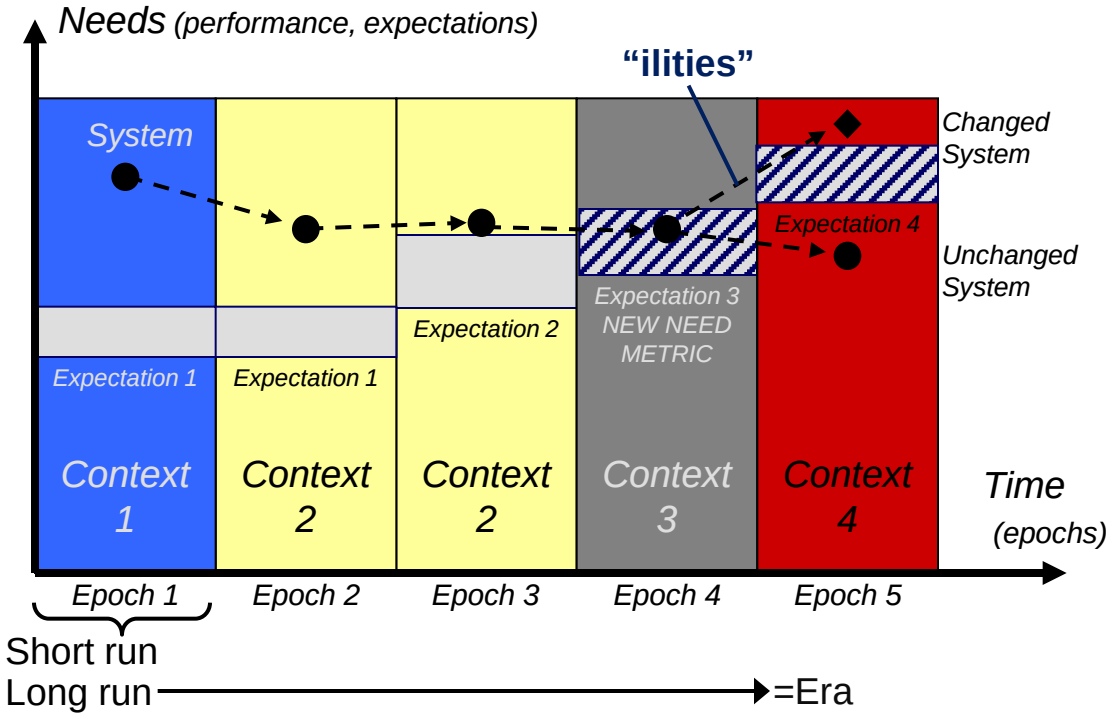
© 2013 Massachusetts Institute of Technology

ERA ANALYSIS

© 2013 Massachusetts Institute of Technology

Using Epochs for Proposing Time-Based Strategies across Eras

Discretization of change timeline into short run and long run enables analysis



Epoch:

Time period with a fixed context; characterized by static constraints, design concepts, available technologies, and articulated needs

Era:

Time-ordered sequence of epochs

Legend

- System
- System Trajectory
- Expectations

(Ross and Rhodes 2008)

In order to pursue dynamic strategies, a system must have temporal properties, e.g., "-ilities" such as flexibility, evolvability, or survivability

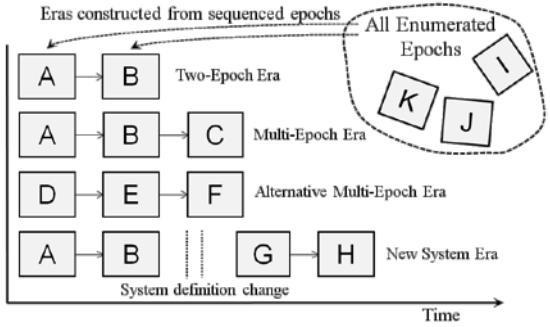
© 2013 Massachusetts Institute of Technology

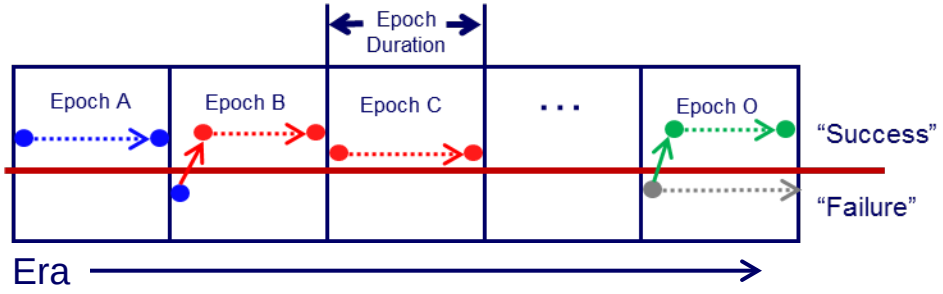
Era Simulation and Analysis

Era Analysis uncovers additional information, emergent only when considering time-ordered effects of uncertainty across the system's lifetime

Conceptually: "how does this system perform, using various value sustainment strategies, when uncertainty evolves over time?"

- Simulation of sample eras (constructed by sequencing epochs according to some model) allows collection of more data
  - Change mechanism usage
  - Cost/benefit "going rates" for adding/removing changeability
  - Lifetime cost/utility/revenue/efficiency statistics





© 2013 Massachusetts Institute of Technology

CASE STUDY

© 2013 Massachusetts Institute of Technology

Maritime Security SoS

MarSec SoS *Directed SoS*

High-level Operational Needs Statement:

Provide maritime security for a particular littoral Area of Interest (AOI)

Stakeholders want a *system (SoS)* that:

- **Detects**, **identifies** and **boards** boats entering AOI
- Is capable of carrying out **search and rescue** missions upon request

Architecture & Design Variables

Form

- Hermes
- Shadow
- Prop Plane
- Helicopter
- Manned Patrol Boat
- Satellite Relay
- Land Sensors

ConOps

- Tech Level Upgrade
- Info Sharing Use
- Task Assignment
- Geographic Segmenting
- Operators Per UAV
- Workforce Buffer
- Authority

Perturbations

Disturbances

- Serious Attack Occurrence
- Asset Unavailable
- Information Attack
- Storm
- Tsunami

Shifts

- Technology Level
- Workforce Availability
- Info Sharing Availability
- Boat Arrival Rate
- Pirate Percentage
- Smuggler Percentage
- Search and Rescue
- Jamming (Bad Com)

© 2013 Massachusetts Institute of Technology

Baseline Designs

Eight designs considered for evaluation in Era Analysis

- Each design is associated with a specific *initial architecture*
- Previously selected among the entire design space using robustness and survivability *screening metrics* (Fitzgerald, 2012)

Design ID	Rationale for selection	Design Characteristics									
		Hermes	Shadow	Prop	Helo	Boats	Task Assign.	Zones	Operators	Authority	Workforce Buffer
A	Robust under preference set 1	2	2	0	0	4	Dedi	2	2:1	Central	0%
B	Robust under preference set 1	2	2	0	0	4	Multi	1	2:1	Central	0%
C	Robust under preference set 2	2	2	0	1	4	Dedi	1	2:1	Central	0%
D	Robust under preference set 2	2	4	0	0	4	Dedi	1	2:1	Central	0%
E	Survivable to some of the perturbations considered	2	4	2	3	8	Multi	1	2:1	Distr.	0%
F	Very robust within 1% of the Pareto frontier	2	6	0	1	4	Dedi	1	2:1	Central	0%
G	Robust across all preference sets and cost types, in specific context of interest	6	6	0	3	12	Multi	2	2:1	Central	0%
H	Expert opinion (exploration into large designs with workforce buffers)	6	6	2	2	12	Dedi	1	2:1	Central	33%

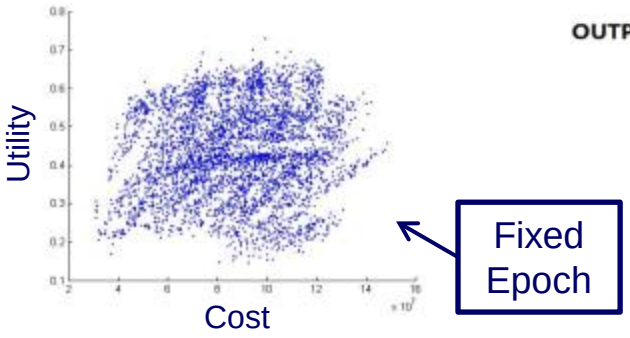
Initial designs are then allowed to change to any other accessible design point in the design space throughout the era (*depending on strategy adopted*)

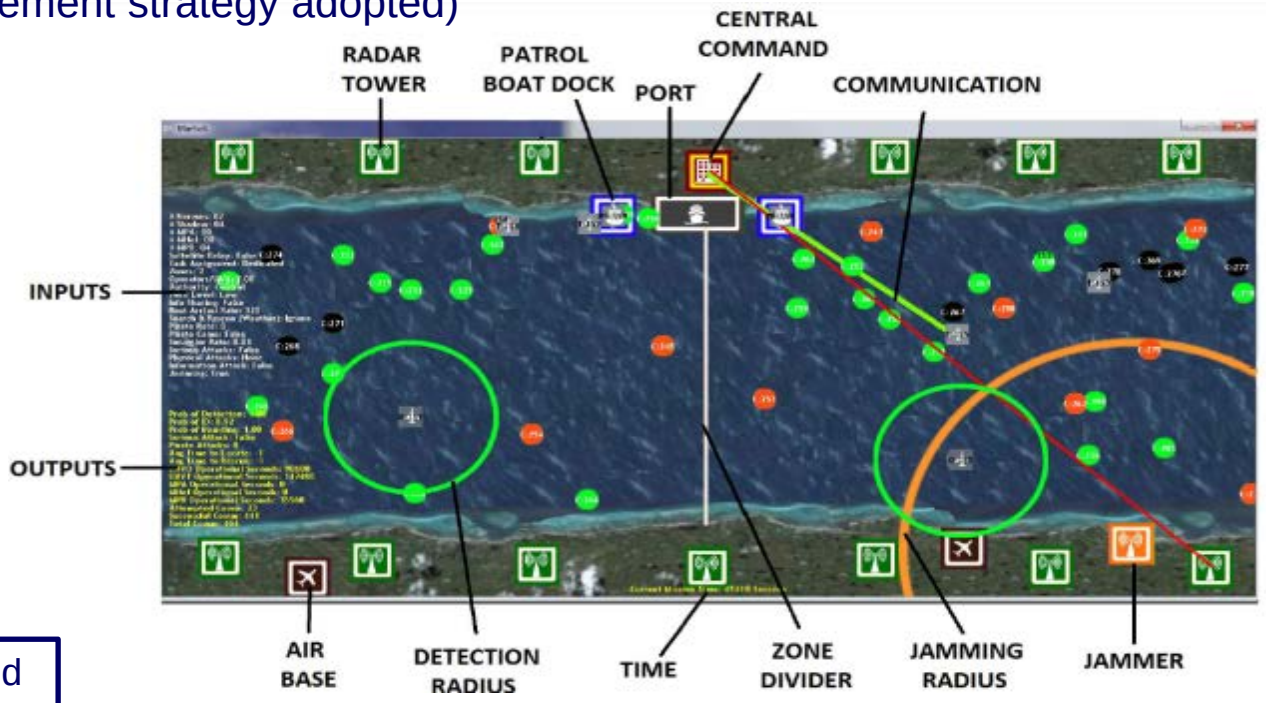
© 2013 Massachusetts Institute of Technology

MarSoS Discrete Event Simulation

DISCRETE-EVENT SIMULATION(Mekdeci, 2013)

- Used to *evaluate the performance* of different MarSec SoS design instances in given epoch
- Allowed inclusion of various **stochastic processes** (e.g., boat arrival rate)
- Allowed analysis of SoS **emergent behaviors** arising from the interaction of constituent systems and operational choices (e.g., how the presence or number of particular UAVs interact with the task management strategy adopted)
- Evaluated **10,368** alternative designs and **128** alternative contexts
- Was *verified* and *validated*
- Evaluated performance was used to generate SoSs' **utility** scores





MarSoS DES is used to evaluate SoS performance in given epoch

© 2013 Massachusetts Institute of Technology

Era Used

ILLUSTRATIVE ERA (8 years)

- For the purpose of performing Era Analysis over the long run for the SoS
- Favorable and unfavorable *epoch shifts* alternate as the SoS **operates through time**

Epoch #	1	2	3	4	5	6	7
Epoch ID	11	27	19	20	55	111	110
Epoch Duration (months)	12	6	24	6	24	18	6
Tech level	Low	Low	Low	Low	High	High	High
Workforce	100%	100%	100%	100%	100%	67%	67%
Info Sharing	Off	On	On	On	On	Off	Off
Boat Arrival	1/640sec	1/640sec	1/320sec	1/320sec	1/320sec	1/640sec	1/640sec
Smuggler Percentage	5%	5%	5%	5%	5%	5%	1%
S&R	No	No	No	No	Yes	Yes	Yes
Jamming	No	No	No	Yes	No	No	Yes

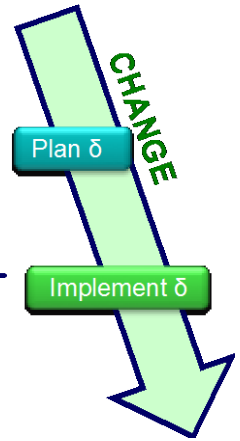
Era Analysis allows for the evaluation of different strategies over the course of the given era

© 2013 Massachusetts Institute of Technology

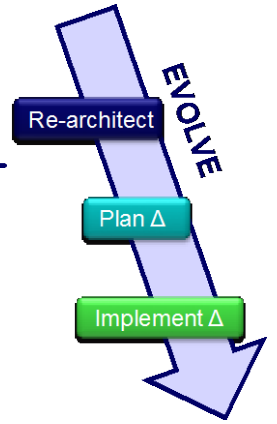
Change v. Evolve

Transition rules

1. Reduce to pre-validated vehicle set (design-level change), i.e. expert-picked stable designs, which can be implemented with current operating SoS constituent systems.
2. Changing short-term (design-level change) ConOps (i.e., Task Assignment, Operators per UAV).



1. Changing long-term (architecture-level change) ConOps (i.e., number of Zones, Authority).
2. Adding Constituent Systems (architecture-level change). The cost of adding is the cost of the new vehicles, and the delay of adding is vehicle-dependent.



Path enablers for changeability can affect the cost and schedule associated with transitions

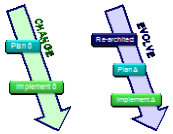
1. Stockpiles of UAVs for quicker transitions
2. Up front training to lower cost and delay of changing CONOPS

© 2013 Massachusetts Institute of Technology

Strategies Used

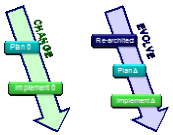
No changes allowed

No re-architecting



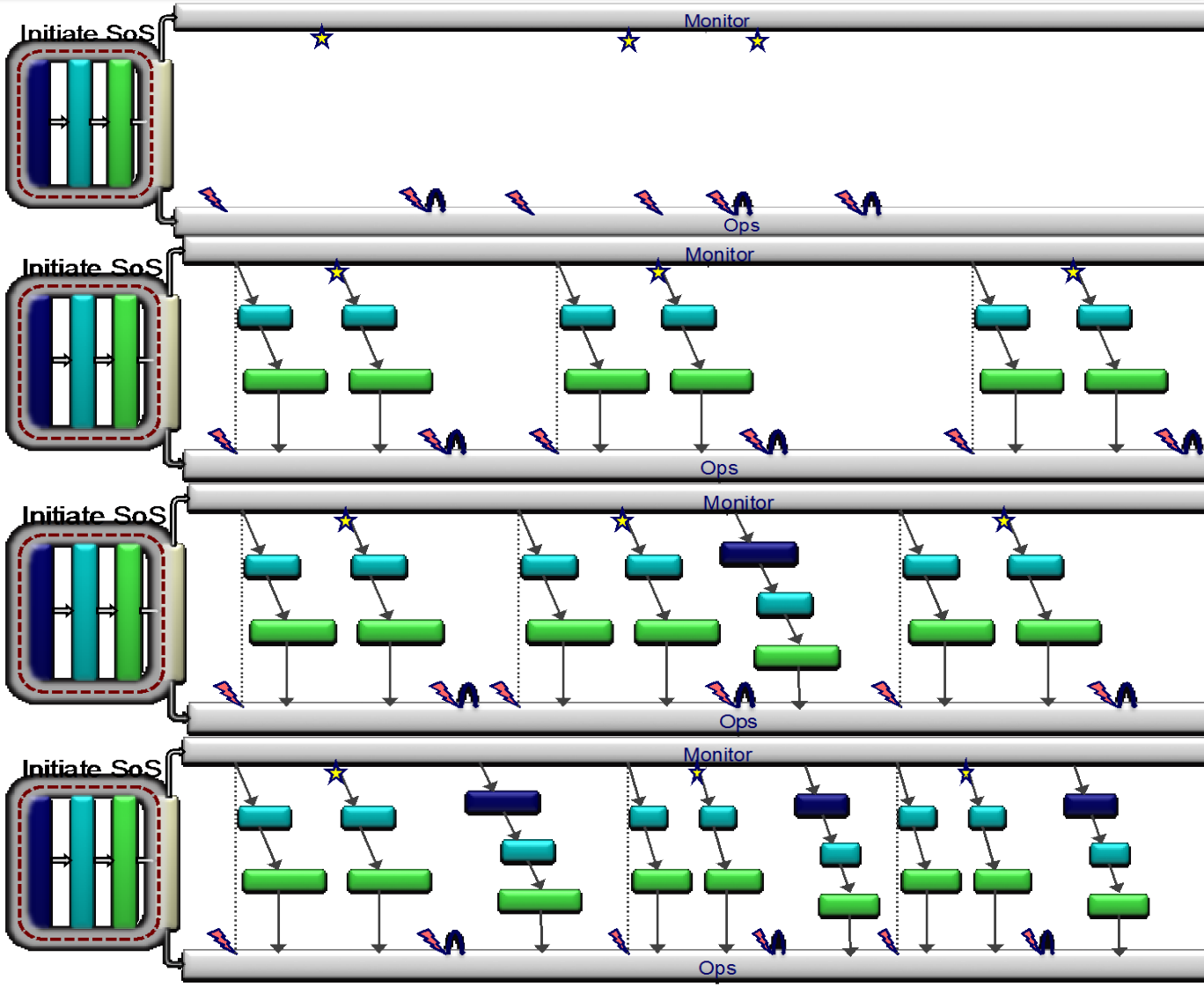
1 re-architecting

(4 years in)



3 re-architecting

(2, 4 & 6 years in)



© 2013 Massachusetts Institute of Technology

Lifecycle Results for "Change" Strategy

"Change"

No re-architecting

Design	Total Utility - Security (utile-months)	Total Discounted Cost ($100M)	Total Down-Time (months)
A	44.8	2.33	0
B	44.7	2.33	0
C	44.7	2.35	0
D	44.9	2.33	0
E	27.2	2.51	3
F	44.8	2.26	0
G	46.4	2.82	0
H	58.0	3.05	0

Similar utility performance amongst designs of interest, except for E (poor) and H (excellent, but at ~25% cost increase)

Maximize efficiency approach is used for design changes between epochs

© 2013 Massachusetts Institute of Technology

Lifecycle Results for "Self-Recovery" Strategy

"Self-recover"

No changes allowed

"Change"

No re-architecting

Design	Total Utility – Security (utile-months)		Total Discounted Cost ($100M)		Total Down-Time (months)	
A	+0	**44.8**	+0	**2.33**	0	**0**
B	+13.5	**44.7**	+0.22	**2.33**	0	**0**
C	+14.6	**44.7**	+0.33	**2.35**	0	**0**
D	+14.7	**44.9**	+0.38	**2.33**	0	**0**
E	-25.3	**27.2**	+1.74	**2.51**	+87	**3**
F	+17.1	**44.8**	+0.86	**2.26**	0	**0**
G	+7.5	**46.4**	+4.21	**2.82**	+24	**0**
H	+10	**58.0**	+4.70	**3.05**	0	**0**

- General increase in utility/cost implies that max efficiency strategy is reducing the size of the designs
- Tradeoff appears likely to be "worth it" for already-small designs (+25% utility, +10% cost) but less so for larger designs (+15% utility, +100% cost)
- Not changing puts you at risk of more downtime
- Possibly an alternative strategy would be best for the large designs

© 2013 Massachusetts Institute of Technology

Lifecycle Results for "Evolve" (Once) Strategy

"Self-recover"	"Change"	"Evolve"
No changes allowed	**No re-architecting**	1 re-architecting

Design	Total Utility – Security (utile-months)			Total Discounted Cost ($100M)			Total Down-Time (months)		
A	+0	**44.8**	+0	+0	**2.33**	+0	0	**0**	+0
B	+13.5	**44.7**	+0	+0.22	**2.33**	+0	0	**0**	+0
C	+14.6	**44.7**	+0	+0.33	**2.35**	+0	0	**0**	+0
D	+14.7	**44.9**	+0	+0.38	**2.33**	+0	0	**0**	+0
E	-25.3	**27.2**	+5.4	+1.74	**2.51**	+0.17	+87	**3**	+0
F	+17.1	**44.8**	+0	+0.86	**2.26**	+0	0	**0**	+0
G	+7.5	**46.4**	+0	+4.21	**2.82**	+0	+24	**0**	+0
H	+10	**58.0**	+0	+4.70	**3.05**	+0	0	**0**	+0

- Only E benefits from single rearchitecting over none
- Decisions about rearchitecting plans to be made weighing the pros and cons of additional/fewer architecting efforts

© 2013 Massachusetts Institute of Technology

Lifecycle Results for "Evolve" (3 Times) Strategy

"Self-recover"	"Change"	"Evolve"	"Evolve"
No changes allowed	**No re-architecting**	1 re-architecting	3 re-architecting

Design	Total Utility – Security (utile-months)				Total Discounted Cost ($100M)				Total Down-Time (months)			
A	+0	**44.8**	+0	+0	+0	**2.33**	+0	+0	0	**0**	+0	+0
B	+13.5	**44.7**	+0	+2.1	+0.22	**2.33**	+0	-0.18	0	**0**	+0	+0
C	+14.6	**44.7**	+0	+15.9	+0.33	**2.35**	+0	-0.09	0	**0**	+0	+0
D	+14.7	**44.9**	+0	+14.8	+0.38	**2.33**	+0	-0.07	0	**0**	+0	+0
E	-25.3	**27.2**	+5.4	+11.9	+1.74	**2.51**	+0.17	-0.39	+87	**3**	+0	+0
F	+17.1	**44.8**	+0	+15.9	+0.86	**2.26**	+0	-0.08	0	**0**	+0	+0
G	+7.5	**46.4**	+0	+10	+4.21	**2.82**	+0	+0.09	+24	**0**	+0	+0
H	+10	**58.0**	+0	+1.6	+4.70	**3.05**	+0	+0.18	0	**0**	+0	+0

- All except Design A benefit from the triple re-architecting, even to the point of gaining utility AND decreasing cost

© 2013 Massachusetts Institute of Technology

DISCUSSION

© 2013 Massachusetts Institute of Technology

Discussion

- Analysis **demonstrative** in nature
- General understanding of the *contextual and needs-related uncertainties* affecting the SoS, as well as the *possible strategies* one can employ to respond to how uncertainty unfolds, are **prerequisites** for the application of the analysis presented
- Addresses some of the **important issues** that characterize modern-day SoSE
  - Considering (and possibly highlighting options to facilitate the implementation of) possible timed and/or contingent SoS changes **early in the design phase**
    - Ex.: Include multi-role asset to respond to loss of UAV via re-tasking
    - Design E performs badly when it can't rapidly and effectively change
  - The value of a given strategy (self-recover, change or re-architect) is dependent on the **initial design selected**
    - Ex.: "change" strategy "addition of back-up assets" to the operating SoS is more effective for smaller (in number of assets) initial SoS designs
  - Consideration of the **timing of execution of architectural changes**
    - re-architecting once was hardly beneficial for any design; re-architecting three times nearly benefitted all designs (improved performance)

© 2013 Massachusetts Institute of Technology

Wrapping up

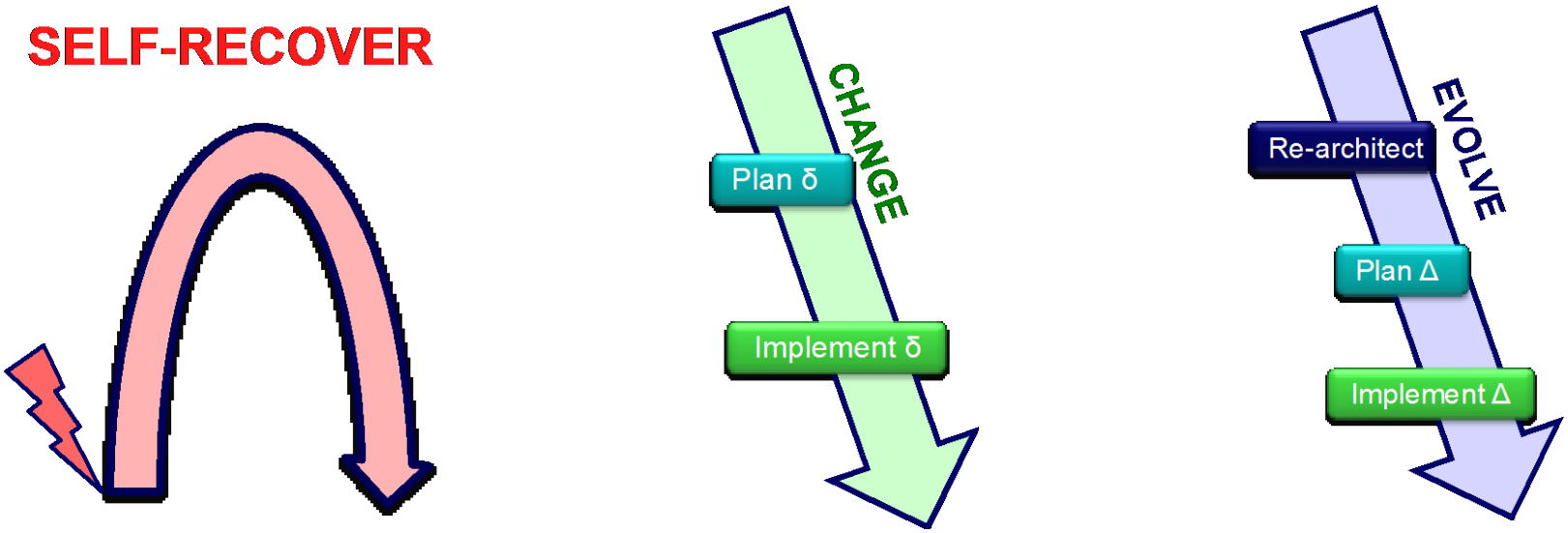
- **Different strategies** can be considered, as well as **alternative eras**
- **Validation** of the approach
- Strategy selection for a particular design is dependent on the era considered
- A **"multi-era" analysis** could be the basis for future research in terms of helping architects to select design-strategy pairs that are most robust across different likely era progressions

Overall contributions of the analysis:

- Era Analysis enables the **exploration** and **comparison** of results associated with the adoption of **alternative "change strategies"**
- Helps systems architects **choose** an appropriate value sustainment strategy for a given envisioned era, as well as **highlight salient time-scales** for SoS intervention
- Helps identify changes that will turn out to be *unnecessary*, or others that will prove to be very important and need to occur with *greater ease*
- Invites systems architects to *think about the adoption of different strategies before the inception of the operational SoS*

© 2013 Massachusetts Institute of Technology

QUESTIONS?



© 2013 Massachusetts Institute of Technology