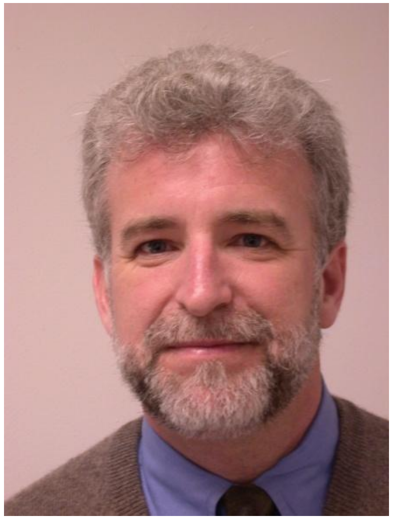
Solutions. Experts. Insights.

SEI TECHNOLOGIES FORUM

Software Engineering Institute | Carnegie Mellon



Software Acquisition Program Dynamics

William Novak
Acquisition Support Program

Novak is a senior member of the technical staff at the SEI. He is a researcher, consultant, and instructor in the acquisition and development of software-reliant systems. Novak has more than 25 years of experience with real-time embedded software product development, government acquisition, and business management.

.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
24 OCT 2011

2. REPORT TYPE

3. DATES COVERED
00-00-2011 to 00-00-2011

4. TITLE AND SUBTITLE
Software Acquisition Program Dynamics

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Carnegie Mellon University ,Software Engineering Institute,Pittsburgh,PA,15213

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**18**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Software Acquisition Program Dynamics
The Problem

Poor acquisition program performance inhibits military performance by depriving the warfighter of critical systems to achieve mission objectives

- Delayed systems withhold needed capabilities
- Wasted resources drain funding needed for new systems

Acquisitions fail for both technical and non-technical reasons; people issues drive adverse acquisition dynamics

- Human, organizational, and management issues drive cost and schedule overruns

Acquisition programs are complex systems with structural dynamics

- Feedback in acquisition produces non-linear interactions that add complexity
- Complex systems can produce seemingly unpredictable behaviors

Misaligned incentives are a key driver of poor acquisition outcomes

- Social dilemmas are a type of misaligned incentives that has received much study
- Social dilemmas occur frequently in software-reliant acquisition programs

Software Acquisition Program Dynamics

Rationale

The Challenge

- Acquisition programs regularly experience recurring cost, schedule, and quality issues
- Acquisition management personnel have limited experience to recognize that

1. Such problems are ubiquitous
2. Many programs fail for the same reasons
3. There are known corrective and preventive techniques to avoid these problems

The Objective

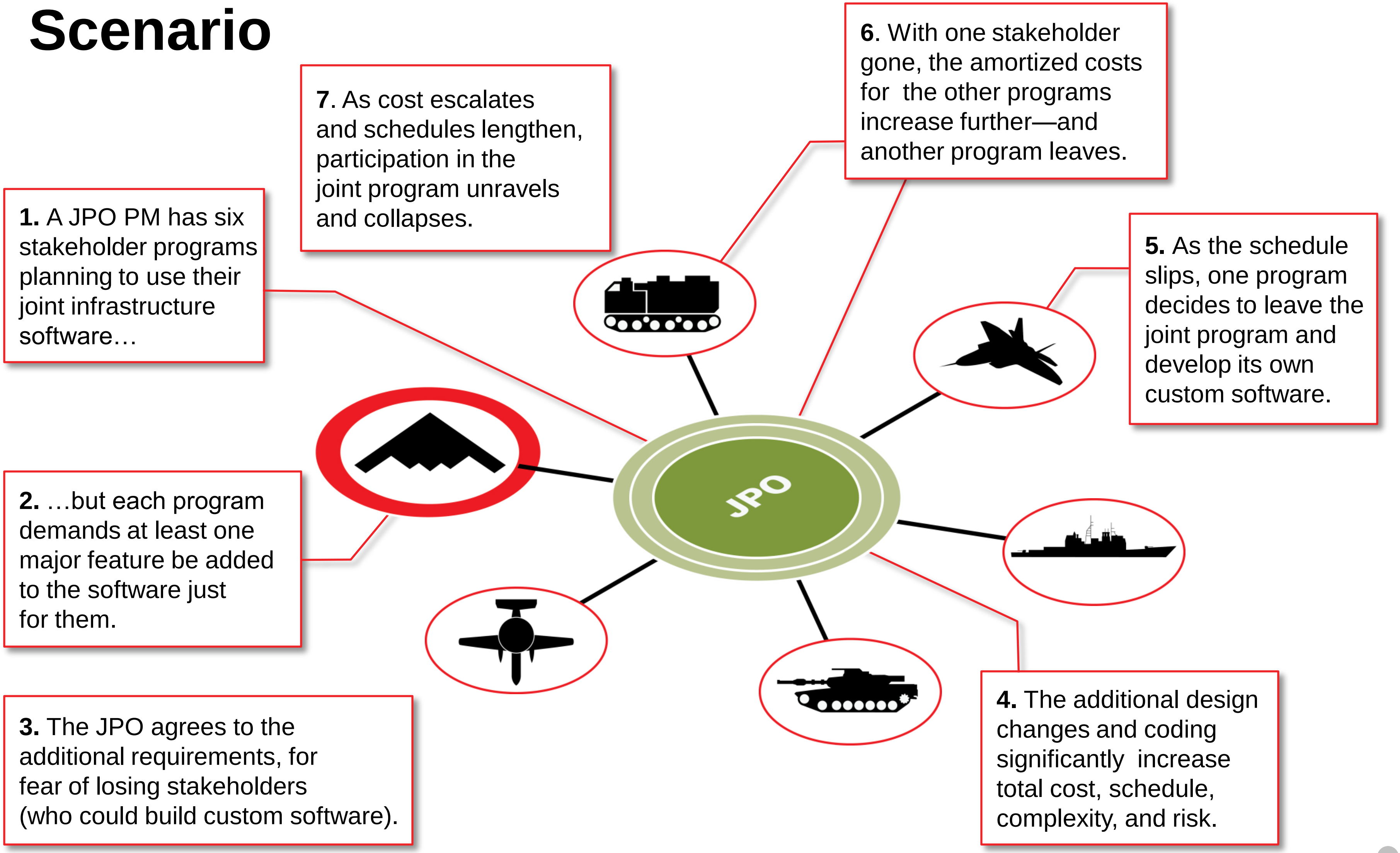
- Improve acquisition program staff decision-making, and thus improve acquisition program outcomes

What Will Change?

- Gain a deeper understanding of problematic acquisition dynamics to help anticipate issues
- Be able to develop and test solution approaches to evaluate their effectiveness
- Develop better decision-making for acquisition staff, to produce better program outcomes

Software Acquisition Program Dynamics

Scenario



1. A JPO PM has six stakeholder programs planning to use their joint infrastructure software…

2. …but each program demands at least one major feature be added to the software just for them.

3. The JPO agrees to the additional requirements, for fear of losing stakeholders (who could build custom software).

4. The additional design changes and coding significantly increase total cost, schedule, complexity, and risk.

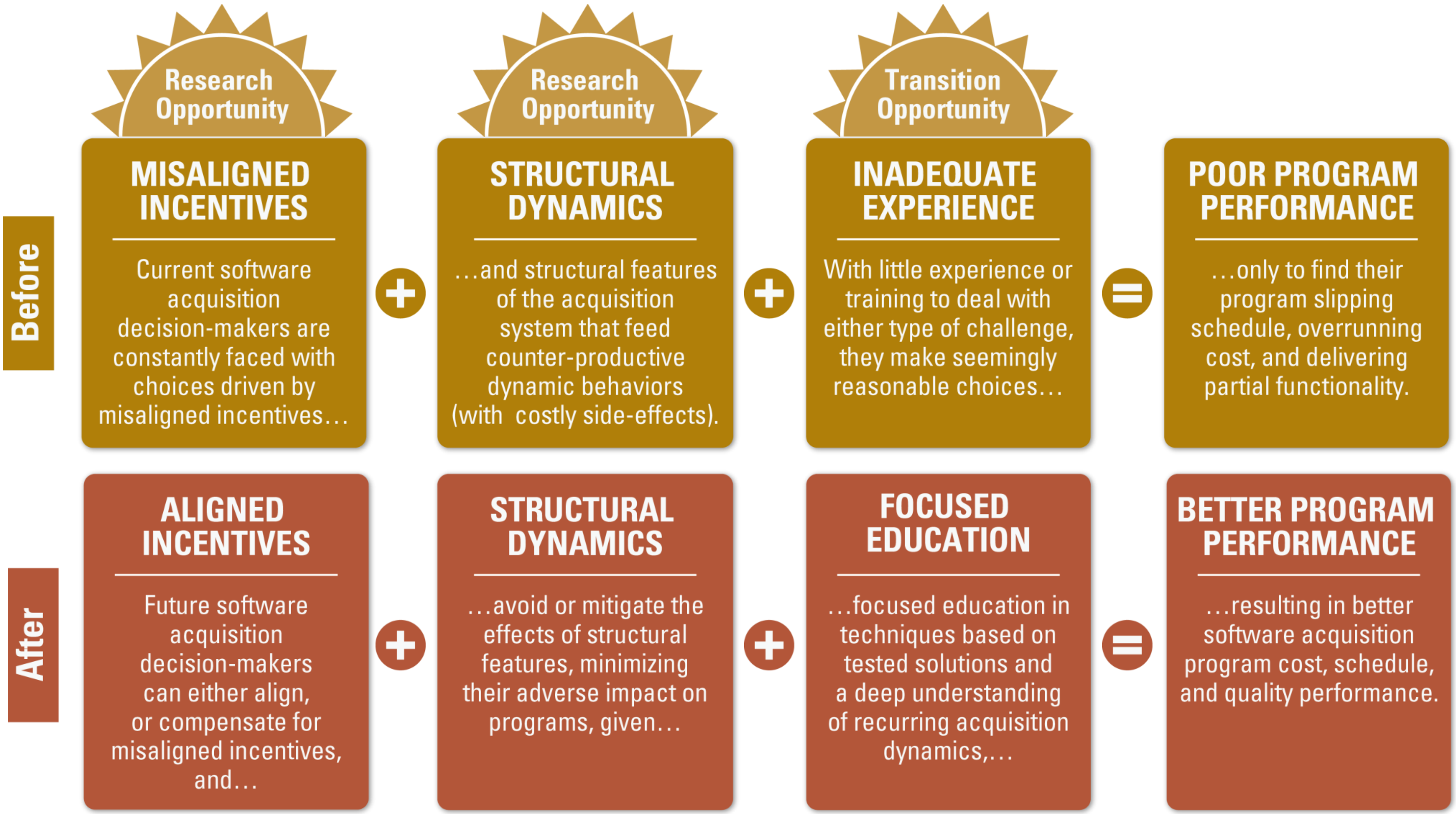
5. As the schedule slips, one program decides to leave the joint program and develop its own custom software.

6. With one stakeholder gone, the amortized costs for the other programs increase further—and another program leaves.

7. As cost escalates and schedules lengthen, participation in the joint program unravels and collapses.

Software Acquisition Program Dynamics

Misaligned Incentives and Structural Dynamics



Before

Research Opportunity

MISALIGNED INCENTIVES

Current software acquisition decision-makers are constantly faced with choices driven by misaligned incentives…

+

Research Opportunity

STRUCTURAL DYNAMICS

…and structural features of the acquisition system that feed counter-productive dynamic behaviors (with costly side-effects).

+

Transition Opportunity

INADEQUATE EXPERIENCE

With little experience or training to deal with either type of challenge, they make seemingly reasonable choices…

=

POOR PROGRAM PERFORMANCE

…only to find their program slipping schedule, overrunning cost, and delivering partial functionality.

After

ALIGNED INCENTIVES

Future software acquisition decision-makers can either align, or compensate for misaligned incentives, and…

+

STRUCTURAL DYNAMICS

…avoid or mitigate the effects of structural features, minimizing their adverse impact on programs, given…

+

FOCUSED EDUCATION

…focused education in techniques based on tested solutions and a deep understanding of recurring acquisition dynamics,…

=

BETTER PROGRAM PERFORMANCE

…resulting in better software acquisition program cost, schedule, and quality performance.

Misaligned Incentives in Software Acquisition

Immature Technology

- Government prefers providing greatest capability, which requires latest technologies
- Contractors prefer using latest technologies to boost staff competency for future bids

Risk Management Participation

- Management may not welcome bad news, viewing it as the reporter’s fault
- Developers have an incentive *not* to report risks, placing personal cost vs. program benefit

Shared Infrastructure Development

- Programs have an incentive to wait for another program to use the shared infrastructure first—better that they work out the bugs, than risk failure of your program

Joint Programs

- To meet conflicting requirements, cost, schedule, size, complexity, and risk all go up
- Users prefer custom solutions they control that are certain to meet their needs

Misaligned incentives are ubiquitous throughout acquisition

Software Acquisition Program Dynamics

Structural Issues in Software Acquisition

Dynamics in complex acquisition are also driven by structural aspects such as feedback and delays, which combine with incentives to increase complexity.

Diverting Staff from Development to Maintenance (*Firefighting*)

- Rework to fix defects in the current release diverts resources from the early design of the next release—injecting even more defects into it

Driving Staff to Work Harder (*Staff Burnout and Turnover*)

- Increasing pressure and long hours eventually lead to burnout and turnover—which reduce productivity and further increase schedule pressure

Testing to Get Positive Outcomes (*Happy Path Testing*)

- Schedule pressure drives the need to make up lost time, which can result in shortcutting quality processes such as rigorous testing

Adding Manpower Late Slows Progress (*Brooks' Law*)

- Adding new people to a late software project to speed development sounds attractive—but in reality adds training overhead, thus causing further delays

Long Program Duration Grows Schedule (*Longer Begets Bigger*)

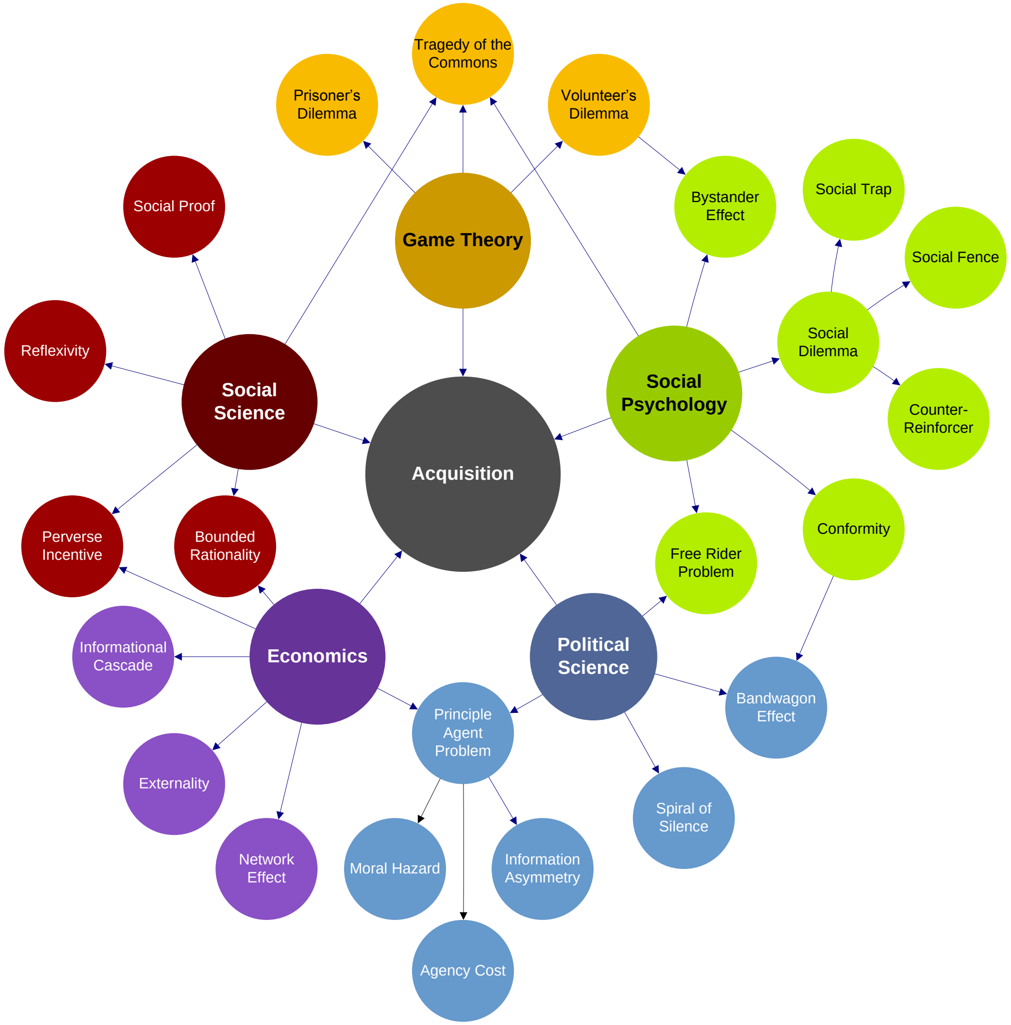
- Long duration allows greater capability to be built
- Long duration drives use of immature technology to avoid obsolescence
- Long duration drives scope creep due to changing threats and new technologies

Software Acquisition Program Dynamics

Related Research

The dynamics of acquisition apply concepts from several disciplines to describe acquisition problems:

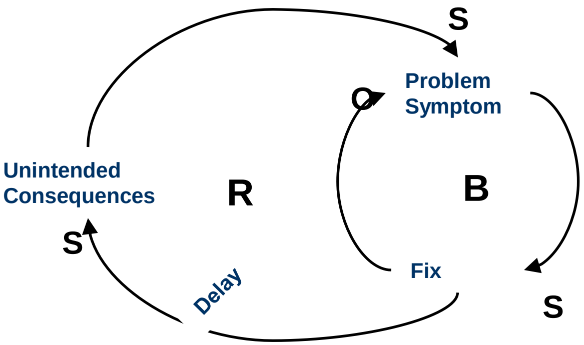
- ● Social Science
- ● Game Theory
- ● Social Psychology
- ● Political Science
- ● Economics



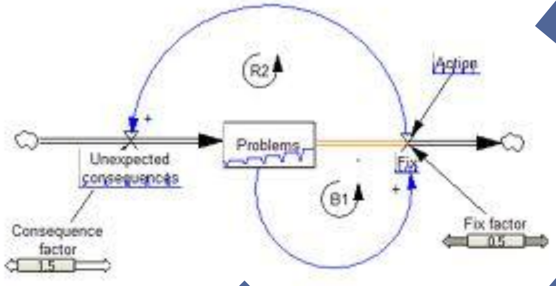
Software Acquisition Program Dynamics

Solution Approach -1

General Qualitative Model



Acquisition Problem Model



Deep Understanding of Dynamic Acquisition Behavior

Model-Based Simulation of Potential Solutions

Basis for Acquisition Instructional Simulations

Independent Technical Assessment (ITA) Data

Detailed examinations of challenged programs with interviews, document reviews, and code analysis

Acquisition Qualitative Model

Firefighting: If design problems are found in the current release, more resources must be used to fix them. This reduces problems, but now less work is done on the *next* release. This undermines its early development work, and increases design problems in the next release.

Acquisition Archetypes
Changing Counterproductive Behaviors in Real Acquisitions
Firefighting
"How can I break this vicious cycle of schedule slips, cost overruns, high defect rates, and deferred functionality?"

Software Acquisition Program Dynamics

Solution Approach $_{-2}$

Build models of previously identified acquisition archetypes to create executable simulations of significant adverse acquisition program behaviors

- Identify key dynamics that map to dilemmas in acquisition
- Build and validate system dynamics models of those dynamics

Extend the models to support simulation of promising solutions/mitigations to assess their efficacy

- Identify known applicable solutions to the modelled adverse acquisition dynamics
- Validate the ability of the models to represent the acquisition situations and resolutions

Recommend candidate solutions/mitigations based on their performance in the simulations at mitigating counter-productive dynamics

- Identify most promising solutions based on overall ability to minimize adverse dynamics
- Use recommended solutions/mitigations to inform educational materials for program staff

Social Dilemmas in Acquisition

Incentives can combine to create social dilemmas where everyone could be better off, but no one has incentive to change their behaviors:

Social Traps[1]: i.e., The Tragedy of the Commons

- An individual desires an immediate *benefit* that will *cost* everyone. If all give in to the same temptation, everyone is worse off.
- *Acquisition Example*: A set of joint stakeholders, each with mandatory custom requirements, can make it impossible for the program to deliver on-time and within budget unless they compromise for the common good.

Social Fences[1]: i.e., Producing a Public Good

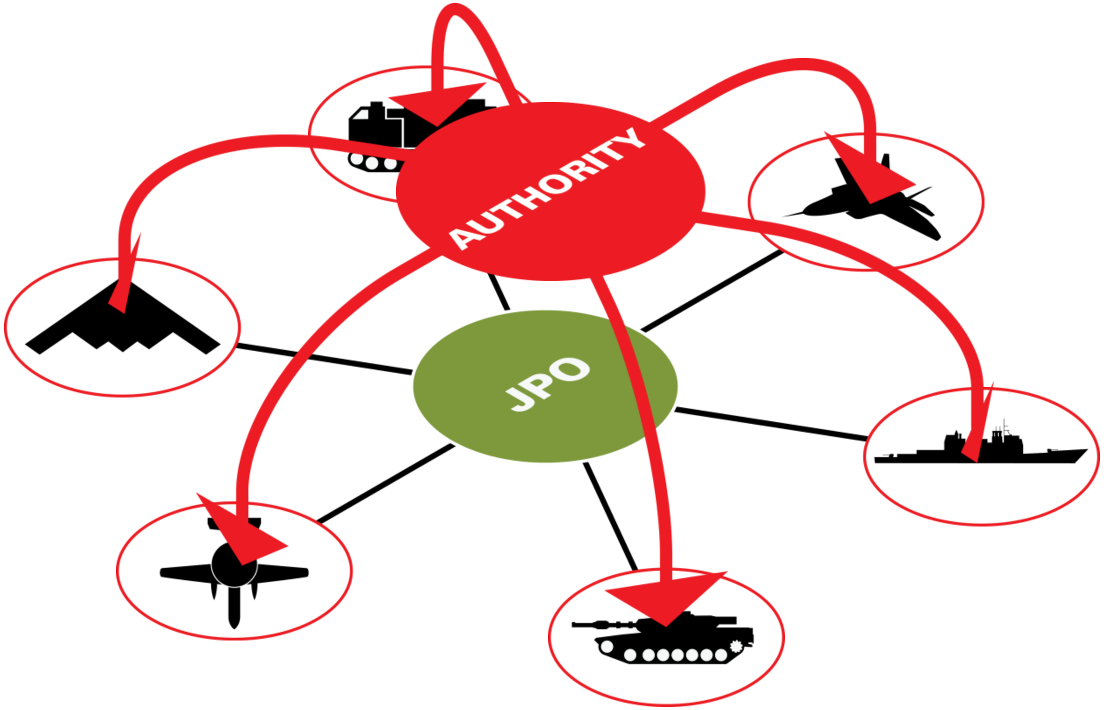
- An individual faces a near-term *cost* that will *benefit* everyone. Each prefers to avoid the cost, but if all do, everyone is worse off.
- *Acquisition Example*: Programs may choose *not* to use a new joint subsystem because of the risk it poses to their program—but if none do, it won't be built.

[1] from Cross and Guyer, *Social Traps,* University of Michigan Press, 1980.

Software Acquisition Program Dynamics

Solutions to the Tragedy of the Commons -1

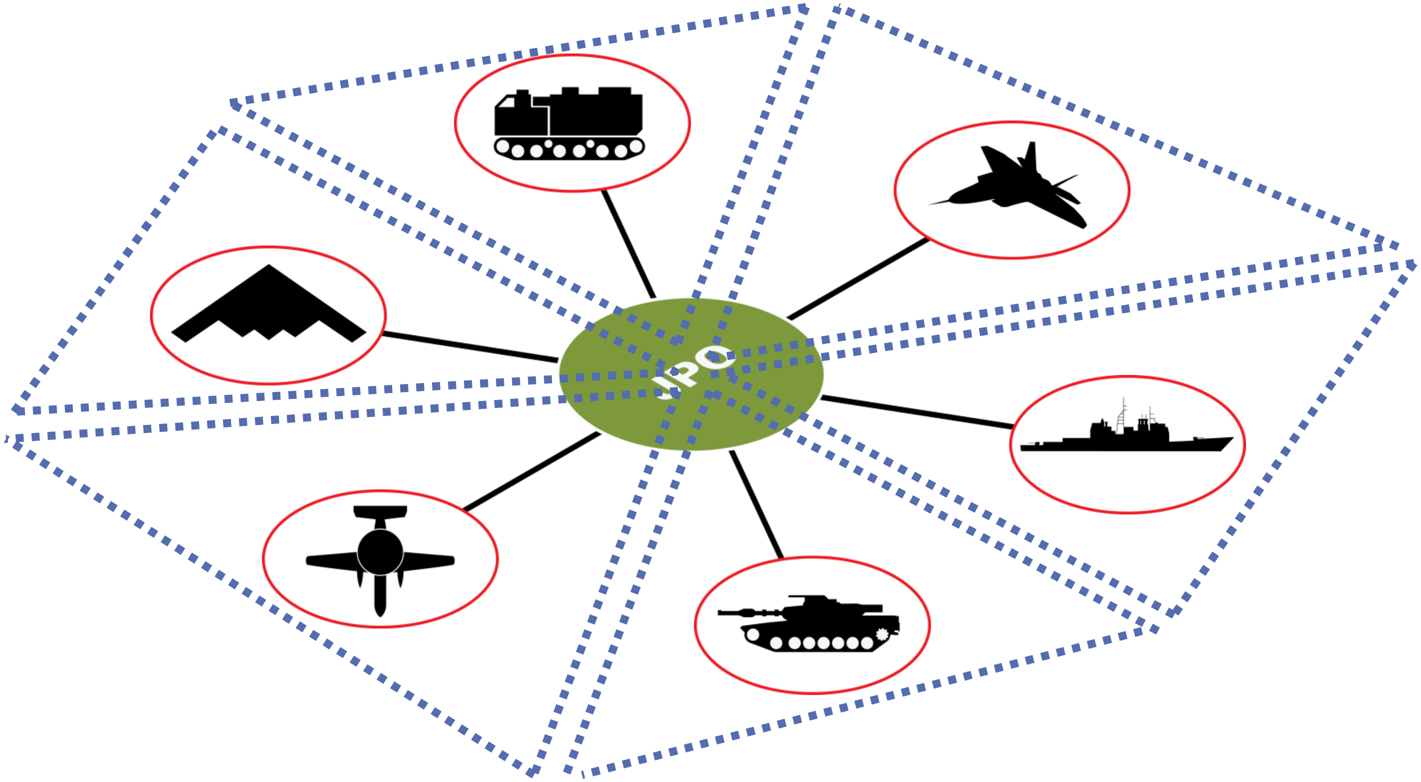
- ***Authority***: Designated authority regulates the good, restricts overusage
  - May be difficult and unpopular to enforce a mandate across organizations



Software Acquisition Program Dynamics

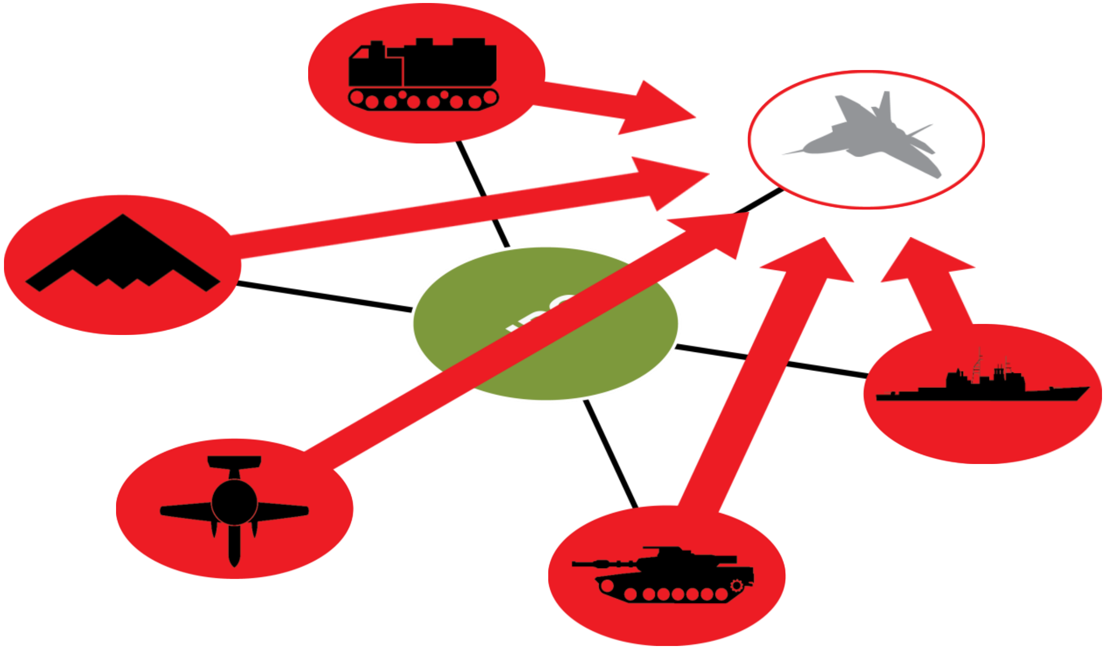
Solutions to the Tragedy of the Commons $_{-2}$

- ***Privatization***: Converts shared ownership to private ownership
  - Each participant has a strong incentive to care for what they own…
    …but privatization defeats the point of cooperation—causing siloed solutions



Solutions to the Tragedy of the Commons -3

- ***Altruistic Punishment***[2]: Participants can penalize uncooperative partners
  - Significantly increases cooperation when used
  - Cost of using penalty discourages overuse, making it self-correcting



[2] from Fehr and Gachter, "Altruistic Punishment in Humans," Nature, 2003

Software Acquisition Program Dynamics

Summary

Build on prior work in static models by developing interactive and executable models of acquisition dynamics

- Turn existing software acquisition domain expertise into a more usable form

Use acquisition models to analyze known adverse software acquisition dynamics, and test proposed mitigations/solutions

- Apply new and known solutions to solving recurring dilemmas in acquisition

Use experiential learning from hands-on simulations to give DoD acquisition staff a deeper understanding of acquisition dynamics to help make better decisions

- Understand common side-effects of decisions that lead to poor performance
- Let acquisition staff gain experience through *education*—not costly mistakes

Build foundation to test future mitigation/solution approaches to assess value

- Qualitatively validate new approaches before applying them to programs

For Additional Information

SEI Report: "*Success in Acquisition: Using Archetypes to Beat the Odds*"

SEI Blog: "*Themes Across Acquisition Programs*": *Parts 1-5*

Website: http://www.sei.cmu.edu/acquisition/research/archetypes.cfm

Acquisition Archetypes analyze recurring patterns in actual programs, and recommend interventions and preventive actions

Acquisition Archetypes
Changing Counterproductive Behaviors in Real Acquisitions
Firefighting
"How can I break this vicious cycle of schedule slips, cost overruns, high defect rates, and deferred functionality?"
"The contractor is taking resources off the next release to put them on current release firefighting, so there's a rippling effect."
"Six months after 'go live' became a disaster, the contractor still denied there were significant technical problems."
Cow
Contractor Hostility
Effect

Download all twelve:

- Firefighting
- The Bow Wave Effect
- Everything for Everybody
- Underbidding the Contract
- Robbing Peter to Pay Paul
- PMO vs. Contractor Hostility
- *…and other titles*

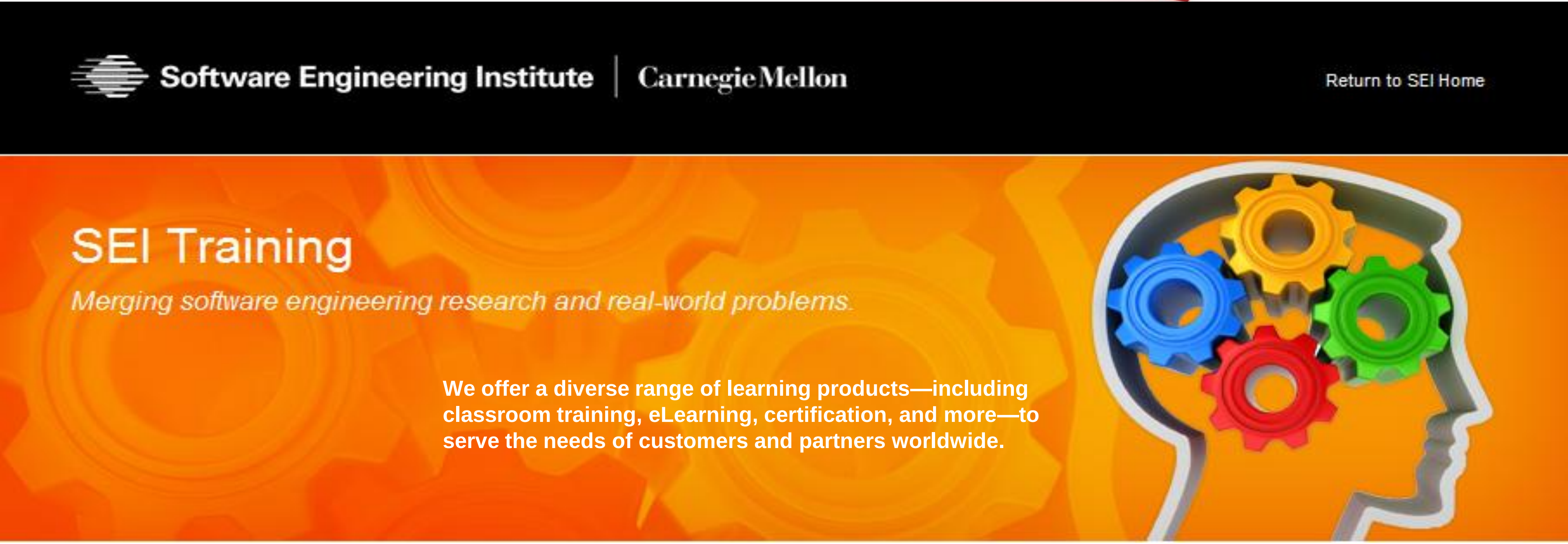
This material is based upon work supported by the U.S. Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.



Software Engineering Institute | Carnegie Mellon

SEI Training

Merging software engineering research and real-world problems.

We offer a diverse range of learning products—including classroom training, eLearning, certification, and more—to serve the needs of customers and partners worldwide.