

# The Evolution of a Science Project: A Preliminary System Dynamics Model of a Recurring Software-Intensive Acquisition Behavior

William E. Novak  
Andrew P. Moore  
Julie B. Cohen  
Jay D. Marchetti

October 25, 2012



## Copyright 2012 Carnegie Mellon University

This material is based upon work funded and supported by the 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.

### **NO WARRANTY**

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL 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.

This material has been approved for public release and unlimited distribution except as restricted below.

The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013 and 252.227-7013 Alternate I.

Internal use: \* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use: \* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

\* These restrictions do not apply to U.S. government entities.



# The Evolution of a Science Project Agenda

Misaligned Incentives

The Evolution of a Science Project

Findings

Interactive Learning Games

Lessons and Summary



# Misaligned Incentives

“Incentives are misaligned—PMs and contractors are not necessarily rewarded for decisions that lead to lower life cycle costs or provide a better balance between cost and performance”

—*Defense Acquisition Performance Assessment,*  
*GEN Ronald Kadish (Ret.)*



# The Evolution of a Science Project

## 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 are primary drivers of 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

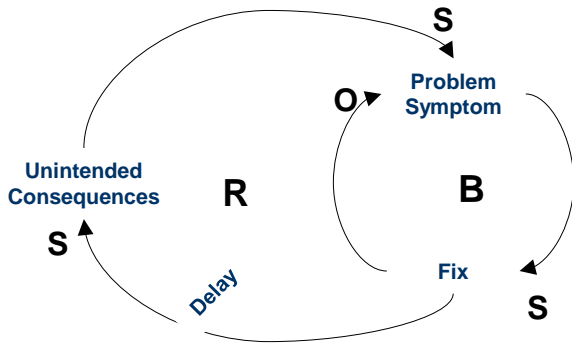
Misaligned incentives are a key driver of poor acquisition outcomes

- Misaligned incentives occur frequently in software-reliant acquisition programs



# The Evolution of a Science Project Solution Approaches <sub>1</sub>

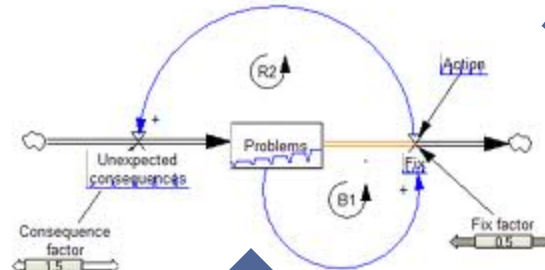
## General Qualitative Model



## Independent Technical Assessment (ITA) Data

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

## Acquisition Problem Model



## 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.



Deep Understanding of Dynamic Acquisition Behavior

Model-Based Simulation of Potential Solutions

Basis for Acquisition Instructional Simulations



## The Evolution of a Science Project

# 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 incentive *not* to report risks, placing personal interest ahead of program

### 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

### Take-away

Misaligned incentives are ubiquitous throughout acquisition



# Evolution of a Science Project

“What they did at first was a proof of concept, a quick and dirty prototype, and when they tried to scale it up, there were indications that it might not be possible...”

—*Acquisition Program Lead*





# The Evolution of a Science Project

## The Evolution of a “Science Project”

9. Warfighters wait years for a new system to be built from scratch.

8. New versions of the system can't be deployed with needed capability, robustness, and performance.

7. New program office unwilling to discard prototype code due to field deployment pressures.

6. Project infrastructure, processes, & staff not able to scale up to production development.

5. As system grows, poor architecture, documentation, & code quality cause poor reliability, performance, & usability.

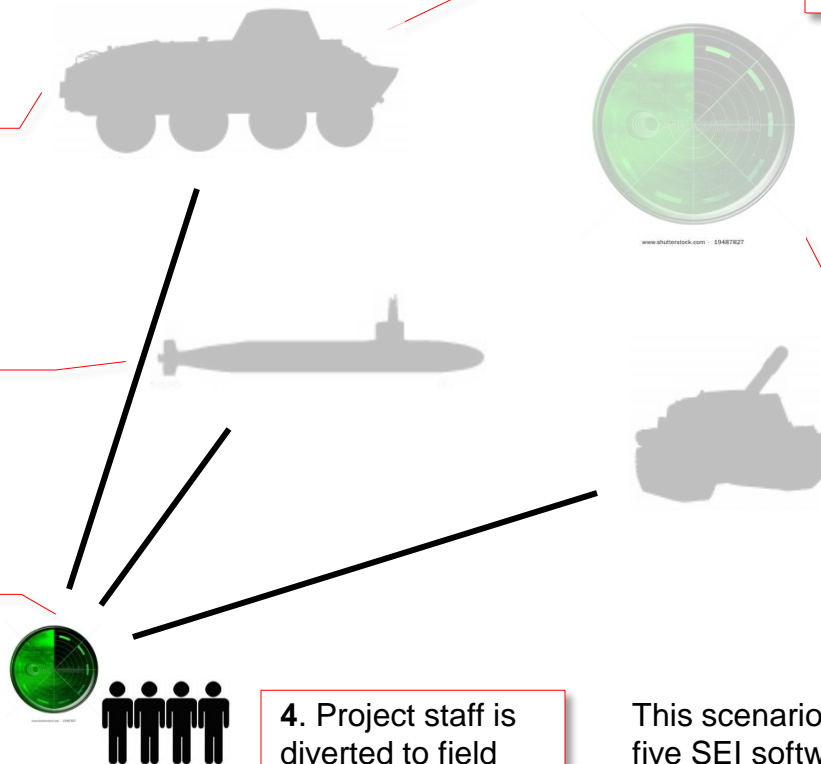
3. Warfighters and field commanders demand more capability, broader deployment, faster response.

2. Prototype is deployed on small scale, and is well received.

1. Project begins as small informal effort to build prototype & prove concept.

4. Project staff is diverted to field support, so development progress slows.

This scenario aggregates five SEI software-reliant system acquisition ITAs conducted in 2006-2009.



## The Evolution of a Science Project

# The Evolution of a “Science Project”

- Behavior has been recognized in many different programs
  - Acquisition executives have seen this dynamic play out in their portfolios
- Model was developed using VenSim system dynamics modeling package
- *Technical Report*: “The Evolution of a Science Project: A Preliminary System Dynamics Model of a Recurring Software-Reliant Acquisition Behavior”
- Planning interactive classroom exercise based on ‘Science Project’ model
  - “The hands-on learning model will be incredibly helpful to the DoD program offices” (*SEI Technology Forum attendee*)







# Findings

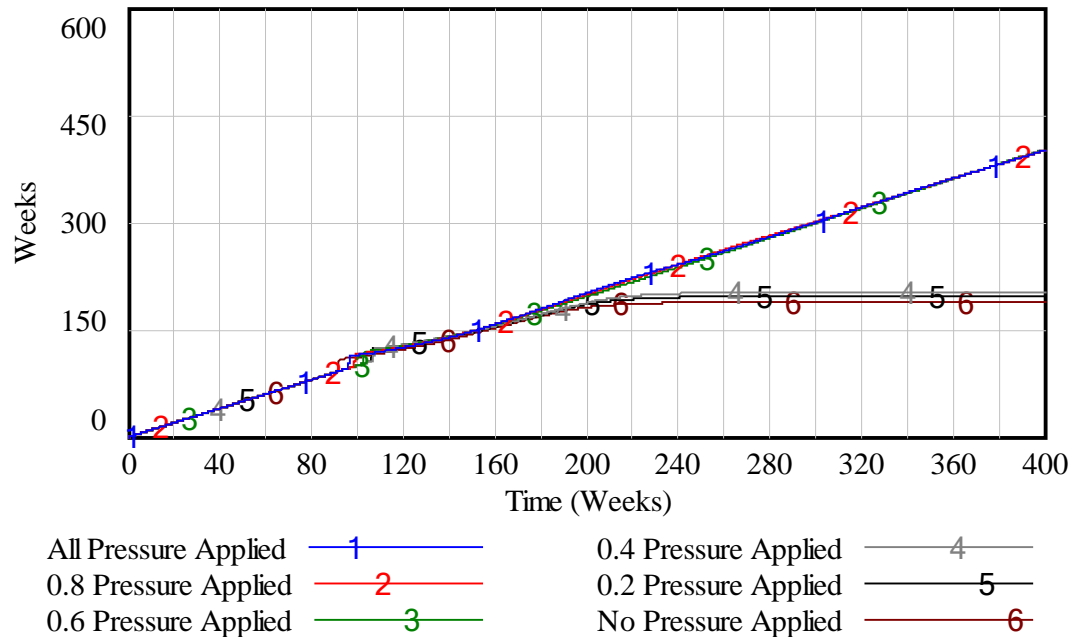


# The Evolution of a Science Project

## Key Preliminary Findings -1

Undiscovered rework drives out the scheduled completion date

PD Scheduled Completion Date (Applying Pressure to Workers)



## The Evolution of a Science Project

# Key Preliminary Findings -2

Placing modest pressure on developers for limited periods shortens schedule

- VenSim optimization shows that placing pressure at a low level is optimal with respect to reducing project duration
- Possibly by allowing periods of pressure, followed by periods of relaxation, the program might:
  - Limit worker burnout
  - Perform even better regarding schedule



## The Evolution of a Science Project

# Key Preliminary Findings -3

There is a critical tipping point in the “Evolution of a ‘Science Project’” dynamic

- “Firefighting” due to rework is a key underlying element
- Accumulating rework creates a dangerous feedback dynamic
- Key drivers in reaching the “tipping point” are:
  - a) pressure on developers
  - b) the degree of “ripple effect”
  - c) the emphasis on schedule and features vs. quality
  - d) the timing of the transition from science project to production development

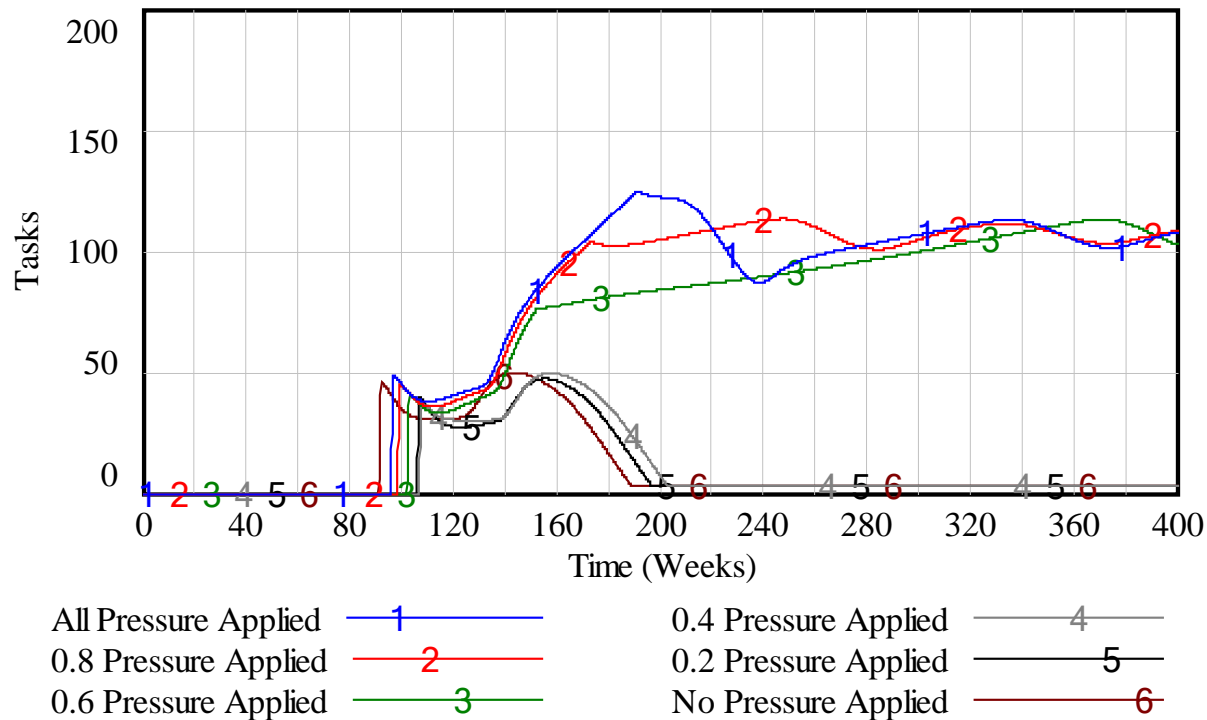


# The Evolution of a Science Project

## Key Preliminary Findings -4

High pressure, or moderate pressure for long periods, can reach the “tipping point”

PD Discovered Quality Issues (Applying Pressure to Workers)



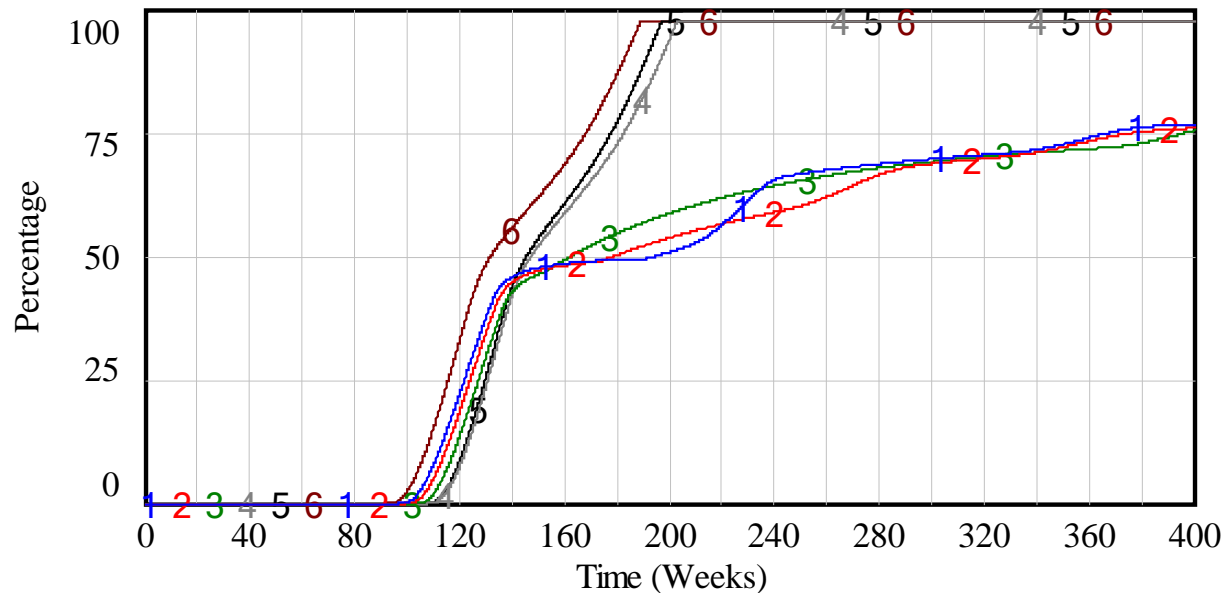


# The Evolution of a Science Project

## Key Preliminary Findings -5

The tipping point contributes to the “90% Done” Syndrome

Percentage Complete (Applying Pressure to Workers)



All Pressure Applied — 1 —  
0.8 Pressure Applied — 2 —  
0.6 Pressure Applied — 3 —

0.4 Pressure Applied — 4 —  
0.2 Pressure Applied — 5 —  
No Pressure Applied — 6 —



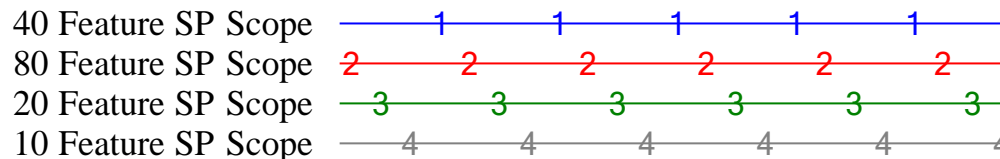
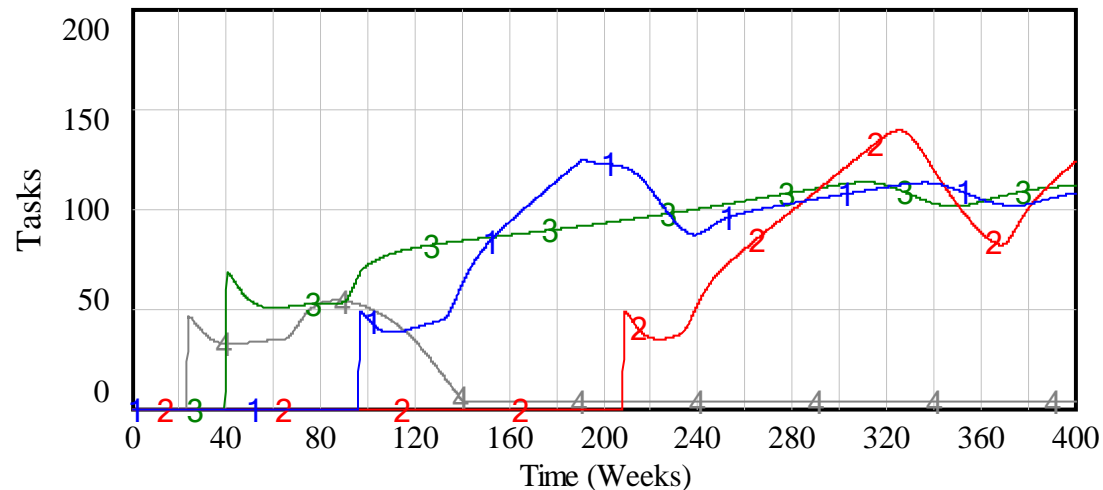
# The Evolution of a Science Project

## Key Preliminary Findings -6

The transition from science project to production effort must be made *early*

- A late transition increases the amount of undiscovered rework that is transferred

PD Discovered Quality Issues (Scoping the SP Effort)

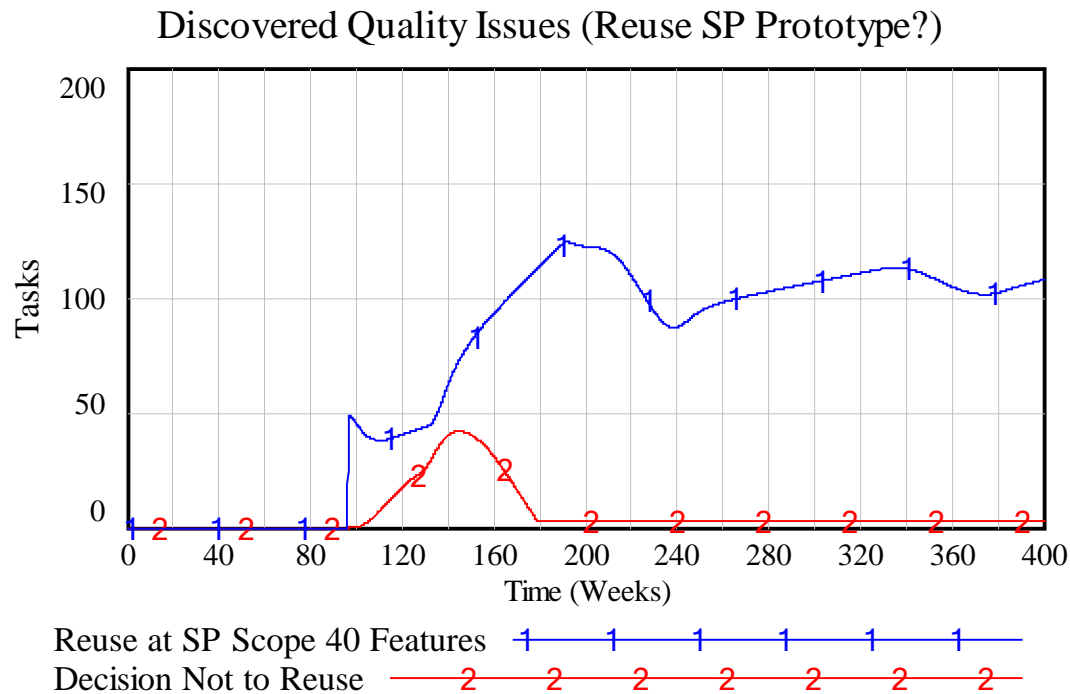


# The Evolution of a Science Project

## Key Preliminary Findings .7

Throwing away the prototype results in better program performance

- Evolutionary development might show better performance



# Interactive Learning Games

“Hear and forget;  
See and remember;  
Do and understand.”

—*Chinese proverb*



## Learning Games for Acquisition

# Why Learning Games?

### Inexperienced Acquisition Staff

- Acquisition staff often have inadequate experience in decision-making
- Well-intentioned decisions are undermined by adverse side-effects
- Poor acquisition management has major cost, schedule, and quality impacts

### Conventional Training is Limited

- Conventional training has been shown to be ineffective in preparing decision-makers for dynamically complex domains

### Learning by Doing

- Give acquisition staff a chance to learn how acquisition programs *really* behave, without risking an actual program

### Games and Simulations Teach Better

- [Cordova 1996, Ricci 1996] found that computer games and simulations enhance learning and understanding in complex domains
- “The hands-on learning model will be incredibly helpful to the DoD program offices” —*SEI Technology Forum attendee*
- [Mayo 2007] found learning doubled for classes with interactive learning vs. only lecture

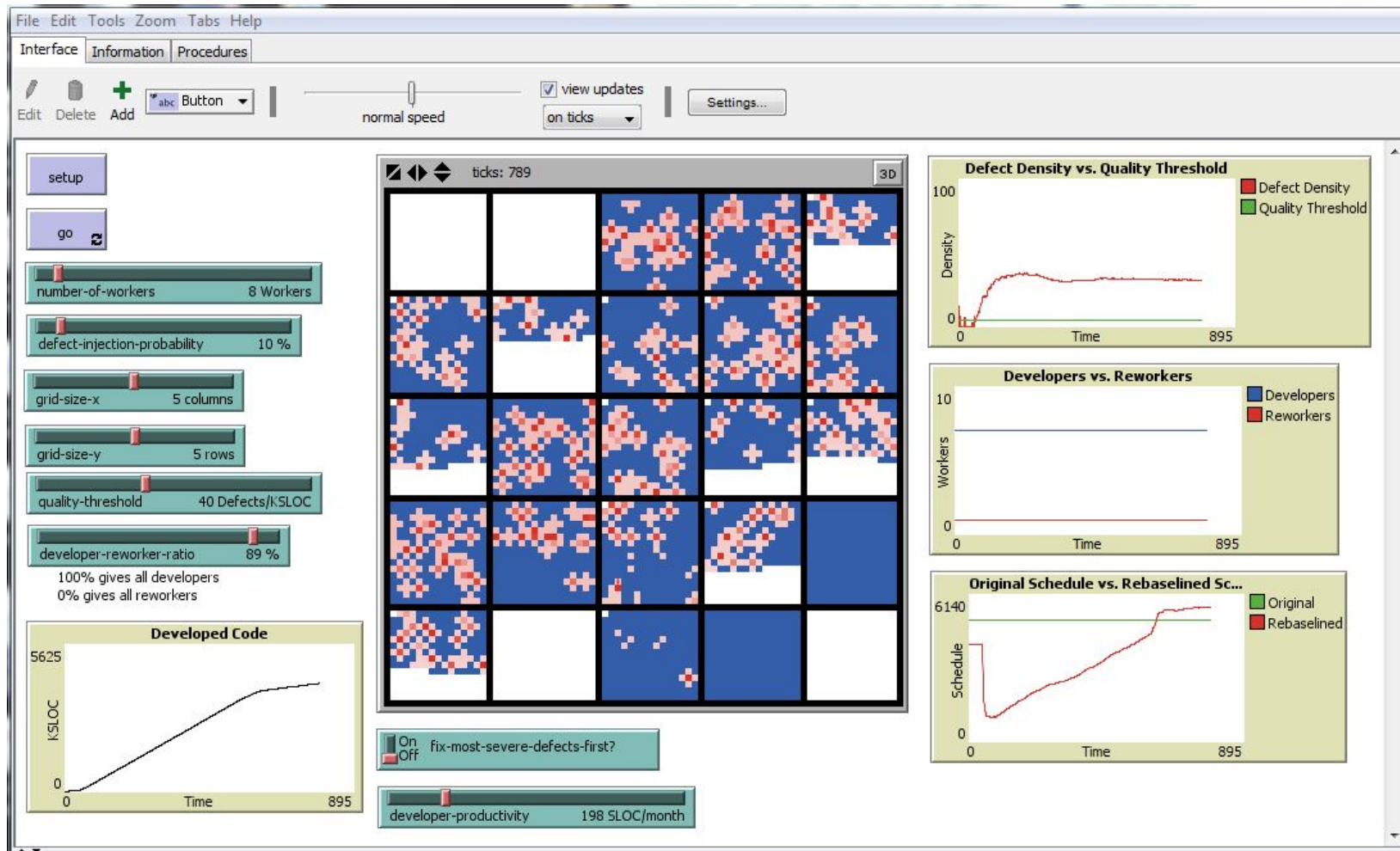
#### Take-away

Interactive learning games are a way to better leverage our investment in the acquisition workforce



# Learning Games for Acquisition

## “Firefighting” Interactive Simulation



# Learning Games for Acquisition

## “Bow Wave Effect” Interactive Game

ticks: 3196 3D

Edit Mode Place Mode next-increment

Available Modules (size/complexity)

Large/Low	Large/Med	Large/High
L/L 3	L/M 2	L/H 1

Medium/Low	Medium/Med	Medium/High
M/L 1	M/M 4	M/H 3

Small/Low	Small/Med	Small/High
S/L 1	S/M 4	S/H 5

Effort Spent	Percent of Work Done	%Coverage	coverage
72	22.581	17.2	65

Effort Spent

coverage



# Lessons and Summary

“At its very core, this acquisition business is not about contracts, testing, acquisition strategies, plans, technology, finance, oversight, or any of the other things one can learn about or make rules about. It's about people.”

—*Terry Little, Missile Defense Agency*





## The Evolution of a Science Project

# Guidance from the “Evolution” Model

- Focus on using the prototype to demonstrate feasibility and value
  - The prototype should generate interest and promote funding
- Transition early to production development once key concepts are proven
  - The sooner formal development starts, the sooner warfighters will get a system
- Prototype reuse should be conceptual—not code-level
  - Confine reuse to “lessons learned” and feedback for improvement
- All models are *wrong*—some models are *useful*
  - No model can accurately model all aspects of the real world
  - Good models produce key insights and raise important questions
- The “I Already Knew That” effect
  - Domain experts may say “I already knew that” about model results
  - It’s easier to point out something as obvious *after* it’s been explained



# The Evolution of a Science Project Summary

We can build on prior work in static models by developing interactive, executable models of key acquisition dynamics

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

We can use acquisition models to help us better understand known adverse software acquisition dynamics

- Model complex dynamic interactions that we can't fully comprehend otherwise

Key preliminary findings from “The Evolution of a Science Project”:

- Undiscovered rework drives out the scheduled completion date
- Placing modest pressure on developers for limited periods shortens schedule
- High pressure, or moderate pressure for long periods, can reach the “tipping point”
- The tipping point contributes to the “90% Done” syndrome
- The transition from ‘science project’ to production effort must be made early
- Throwing away the prototype results in better program performance

Interactive classroom games can improve acquisition staff decision-making

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



# The Evolution of a Science Project For Additional Information

SEI Report: *"The Evolution of a Science Project: A Preliminary System Dynamics Model of a Recurring Software-Reliant Acquisition Behavior"*

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

SEI Blog: *"Themes Across Acquisition Programs": Parts 1-4*

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

Download all twelve:

- PMO vs. Contractor Hostility
- Underbidding the Contract
- Everything for Everybody
- The Bow Wave Effect
- Brooks' Law
- Firefighting
- "Happy Path" Testing
- Longer Begets Bigger
- Shooting the Messenger
- Feeding the Sacred Cow
- Staff Burnout and Turnover
- Robbing Peter to Pay Paul



## The Evolution of a Science Project

# Joint Program Experience Needed!

We are analyzing the dynamic organizational behavior of joint and joint-interest programs as part of a new research effort.

We are conducting a few group modeling workshops to elicit key joint program behaviors, and are using the information to build a new system dynamics model.

If you'd be interested in participating in this work, please contact:

William E. Novak  
Senior Member of Engineering Staff  
Military Services Team  
Acquisition Support Program  
*Office:* 412.268.5519  
*Mobile:* 412.526.1885  
*Email:* [wen@sei.cmu.edu](mailto:wen@sei.cmu.edu)  
Software Engineering Institute  
Carnegie Mellon University  
4500 Fifth Avenue  
Pittsburgh, PA 15213-3890





**Software Engineering Institute**

**Carnegie Mellon**



**Software Engineering Institute**

**Carnegie Mellon**

The Evolution of a Science Project  
© 2012 Carnegie Mellon University