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

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

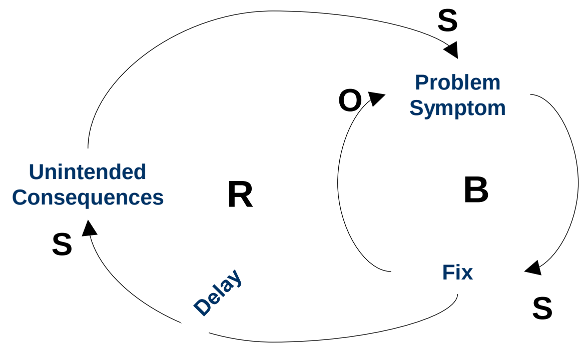
Misaligned incentives are a key driver of poor acquisition outcomes

- Misaligned incentives occur frequently in software-reliant acquisition programs

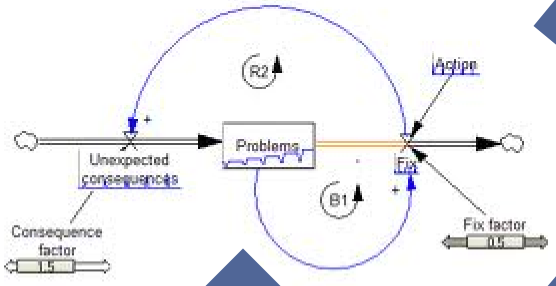
The Evolution of a Science Project

Solution Approaches 1

General Qualitative Model



Acquisition Problem Model



Deep Understanding of Dynamic Acquisition Behavior

Model-Based Simulation of Potential Solutions

Basis for Acquisition Instructional Simulations

Acquisition Qualitative Model

Independent Technical Assessment (ITA) Data

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

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.

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

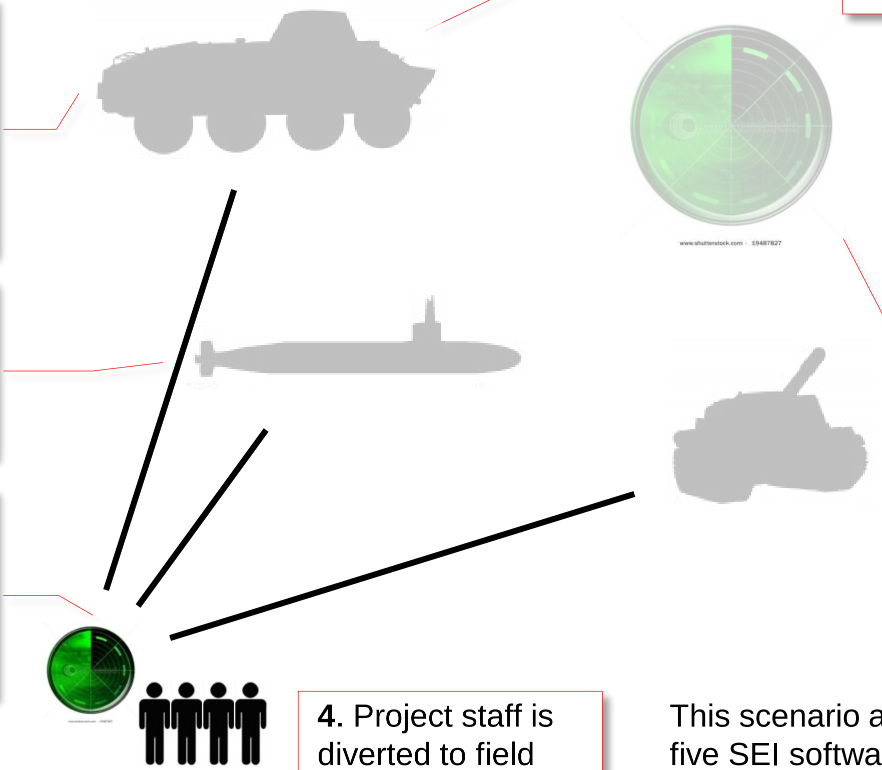
Evolution of a Science Project

"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 Pro

The Evolution of a "Science Project"



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

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

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

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

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

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

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

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

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

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

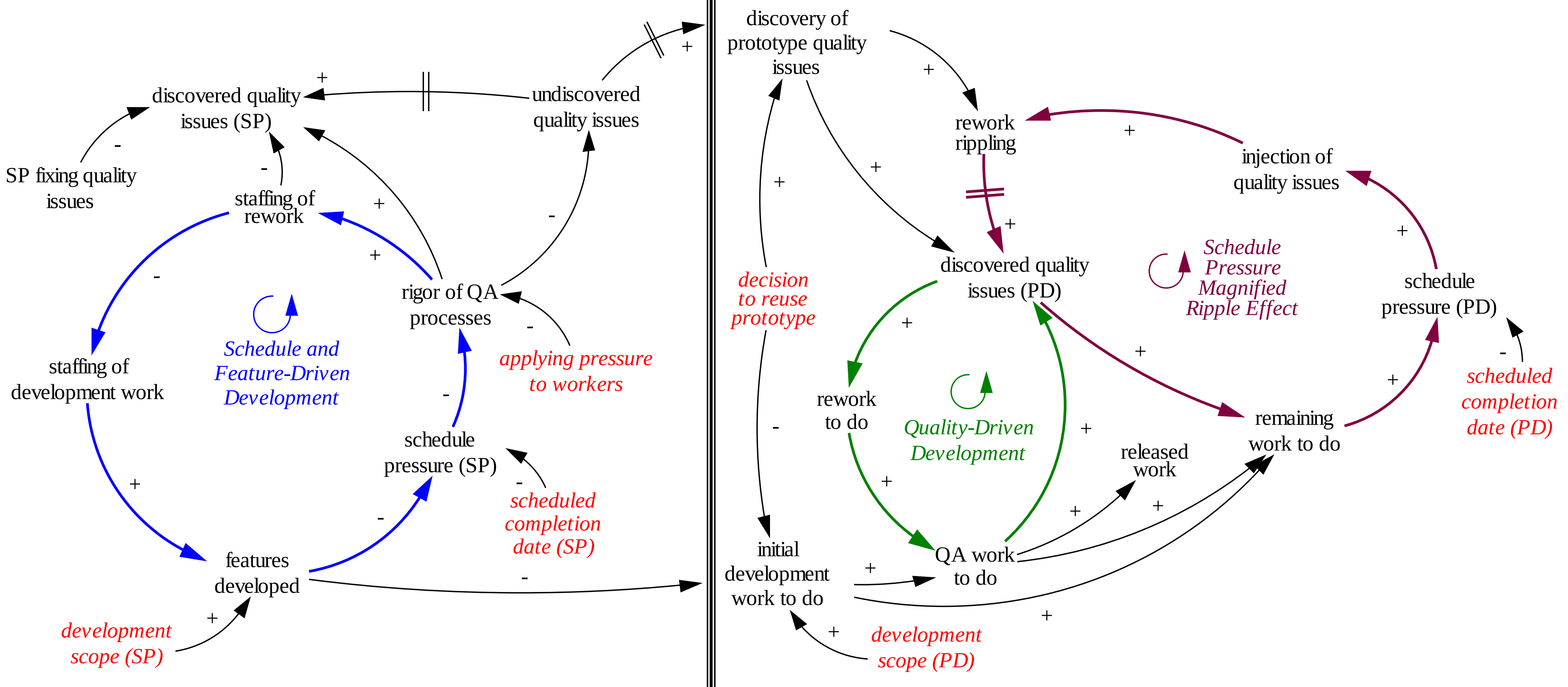
The Evolution of a “Science Project”

- 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*)

The Evolution of a “Science Project”

Science Project (SP) Sector

Production Development (PD) Sector

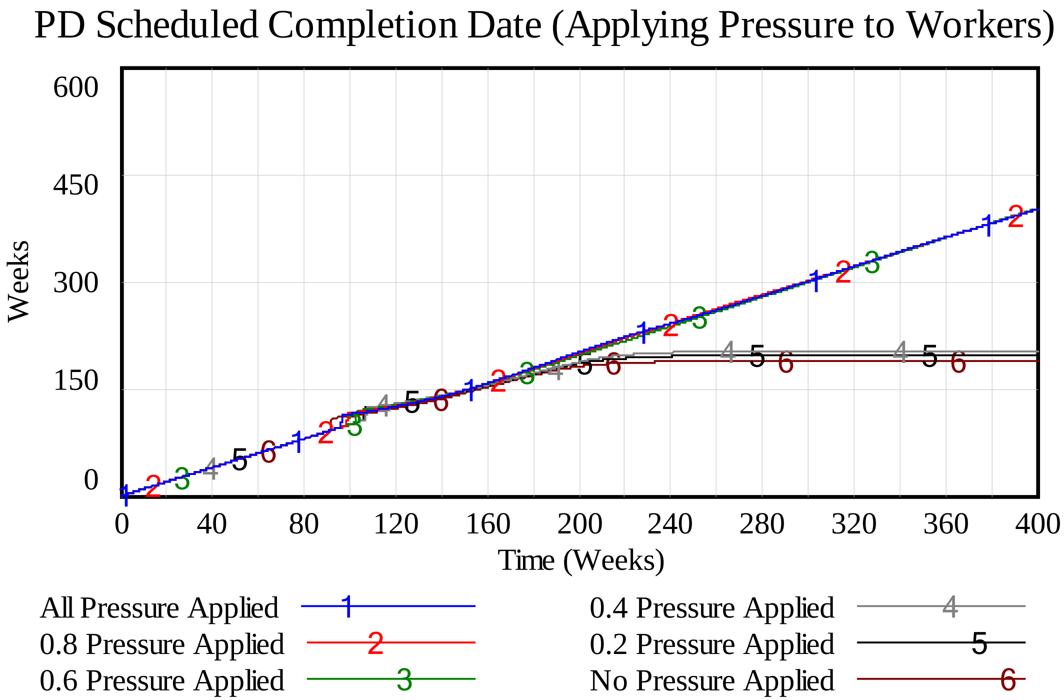

discovered quality issues (SP)
undiscovered quality issues
SP fixing quality issues
staffing of rework
staffing of development work
Schedule and Feature-Driven Development
rigor of QA processes
applying pressure to workers
schedule pressure (SP)
scheduled completion date (SP)
features developed
development scope (SP)
discovery of prototype quality issues
rework rippling
injection of quality issues
decision to reuse prototype
discovered quality issues (PD)
Schedule Pressure Magnified Ripple Effect
schedule pressure (PD)
rework to do
Quality-Driven Development
released work
remaining work to do
scheduled completion date (PD)
initial development work to do
QA work to do
development scope (PD)

Findings

The Evolution of a Science Project

Key Preliminary Findings -1

Undiscovered rework drives out the scheduled completion date



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”

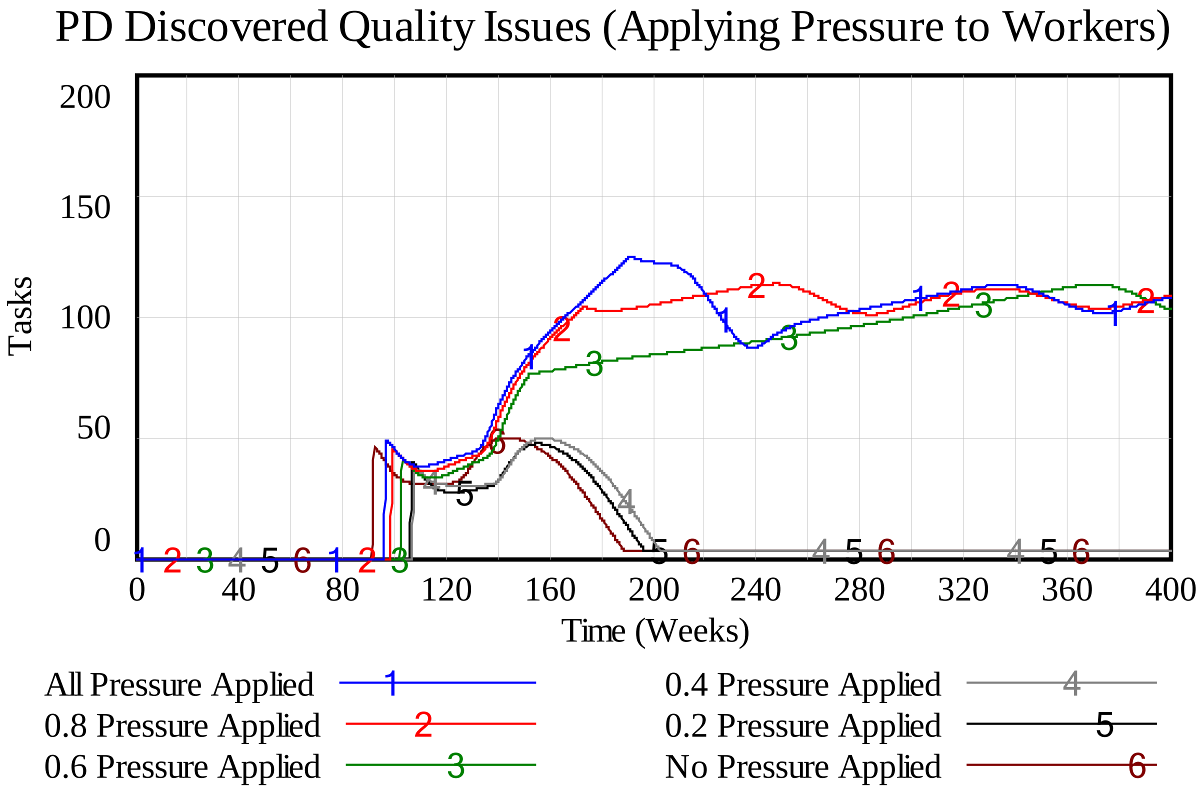
  c) the emphasis on schedule and features vs. quality

  d) the timing of the transition from science project to production development

The Evolution of a Science Project

Key Preliminary Findings -4

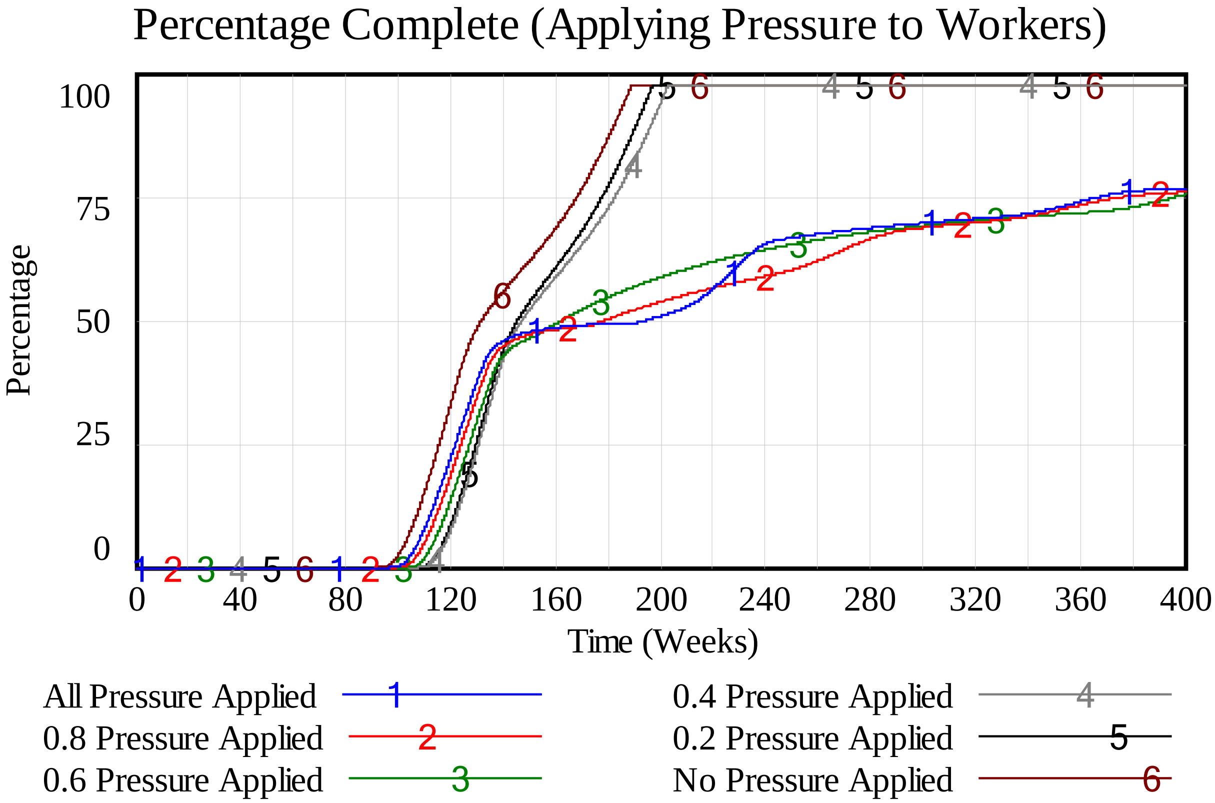
High pressure, or moderate pressure for long periods, can reach the “tipping point”



The Evolution of a Science Project

Key Preliminary Findings -5

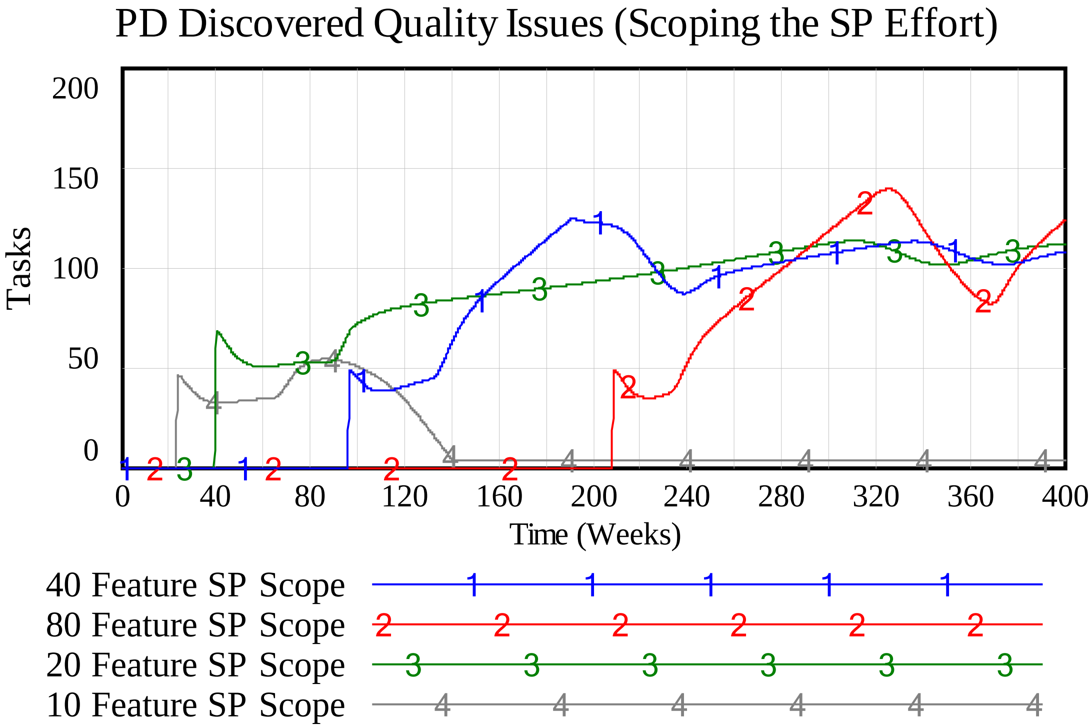
The tipping point contributes to the “90% Done” Syndrome



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

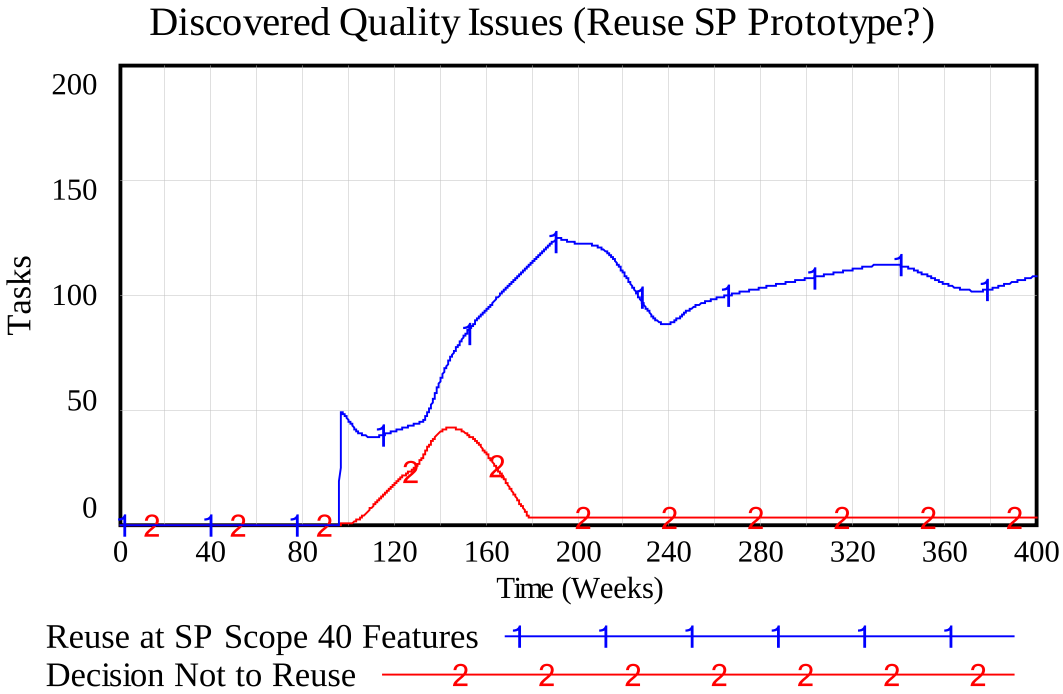


The Evolution of a Science Project

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*

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

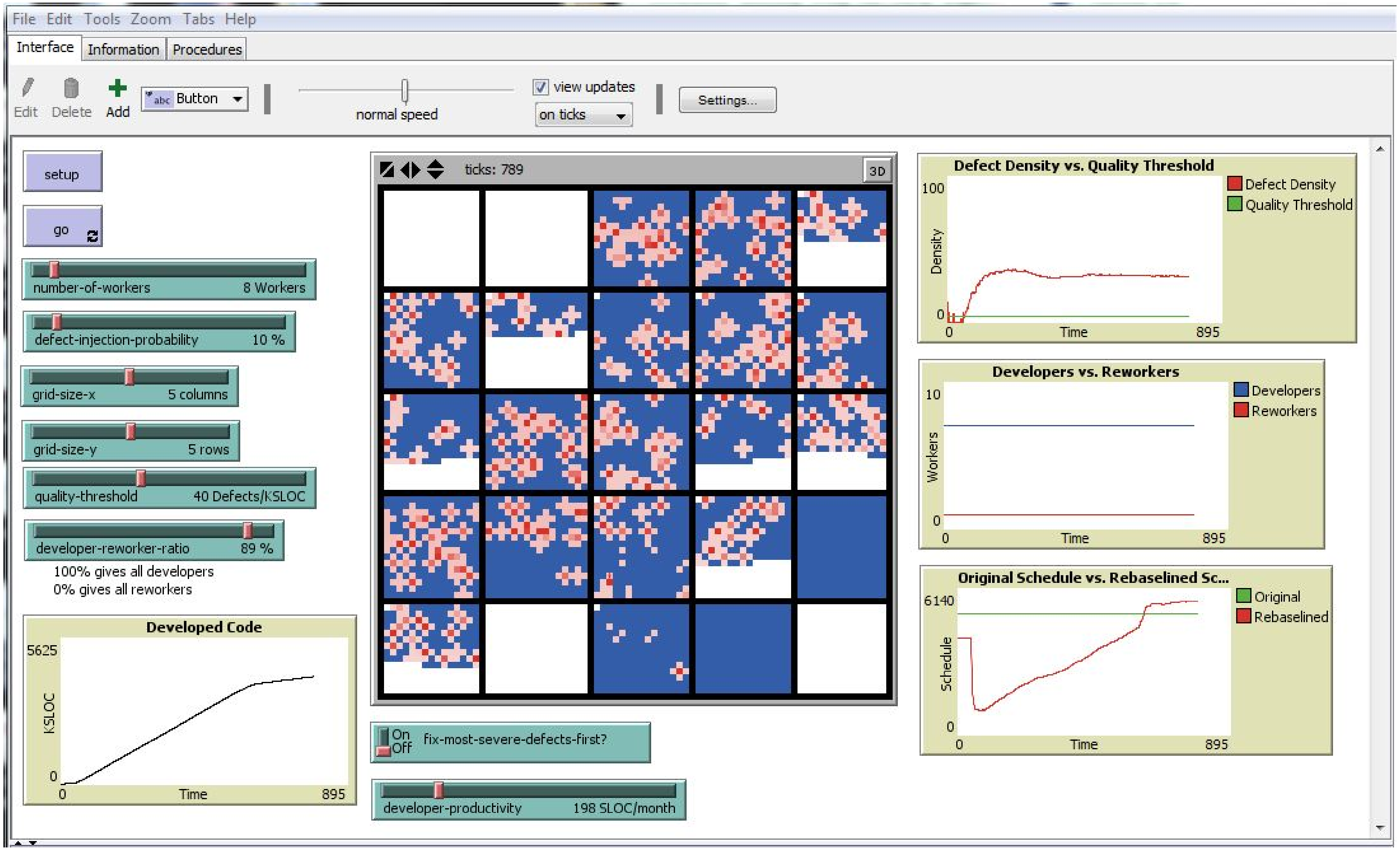
Games and Simulations Teach Better

- [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

“Firefighting” Interactive Simulation



☑ view updates

on ticks

number-of-workers 8 Workers

defect-injection-probability 10 %

grid-size-x 5 columns

grid-size-y 5 rows

quality-threshold 40 Defects/KSLOC

developer-reworker-ratio 89 %

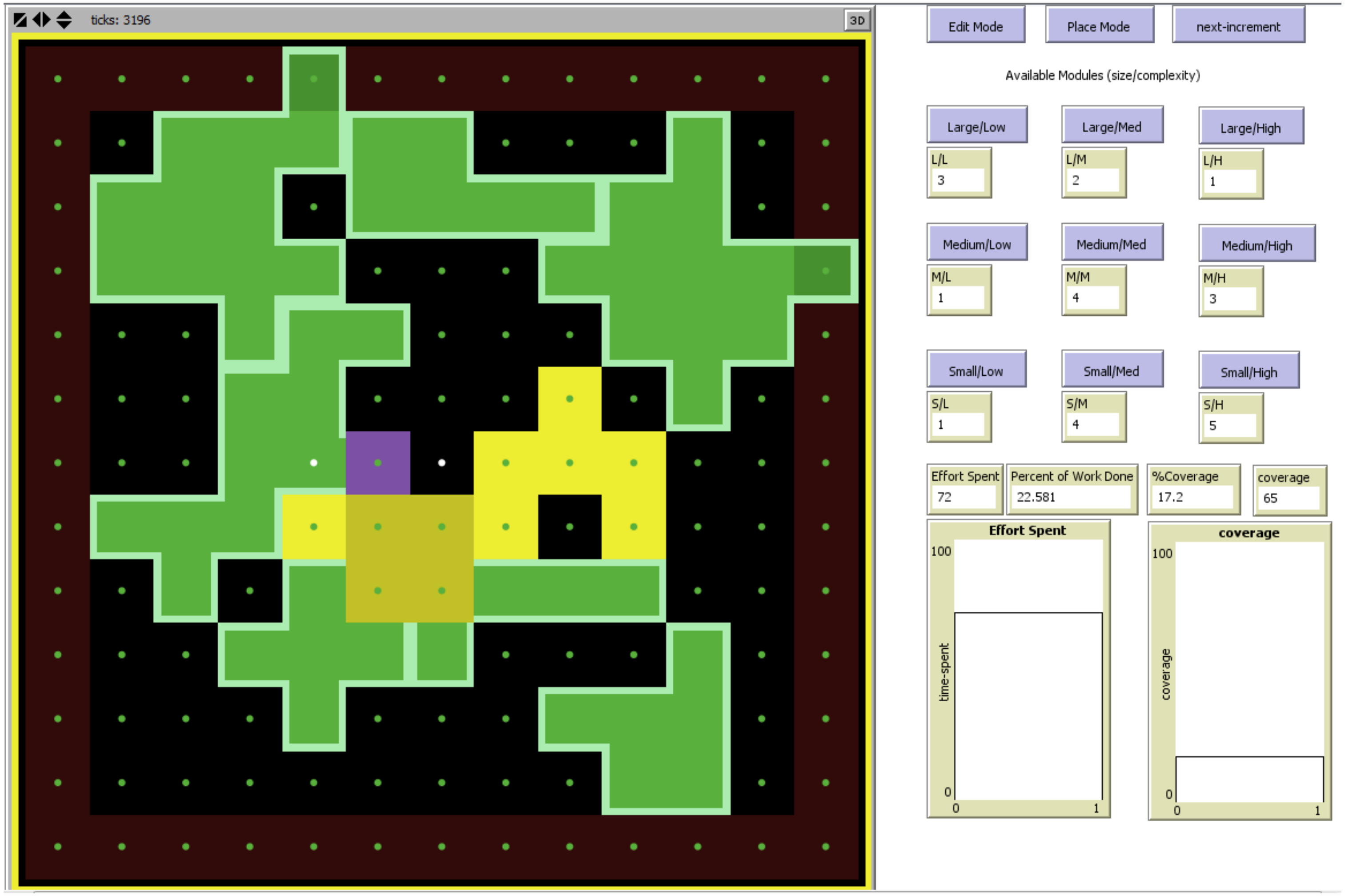
100% gives all developers
0% gives all reworkers

On / Off fix-most-severe-defects-first?

developer-productivity 198 SLOC/month

Learning Games for Acquisition

“Bow Wave Effect” Interactive Game



ticks: 3196

L/L 3

L/M 2

L/H 1

M/L 1

M/M 4

M/H 3

S/L 1

S/M 4

S/H 5

Effort Spent 72

Percent of Work Done 22.581

%Coverage 17.2

coverage 65

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

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

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

Acquisition Archetypes
Firefighting
Sacred Cow
Contractor Hostility
Bow Wave Effect
"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."
"We're trying to put 8 pounds of slop into a 5 pound bag."

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
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-3890

Software Engineering Institute | Carnegie Mellon