Causal Factors of Software Intensive System Issues

Tri Service Assessment Initiative

Practical Software and Systems Measurement
July 24, 2002

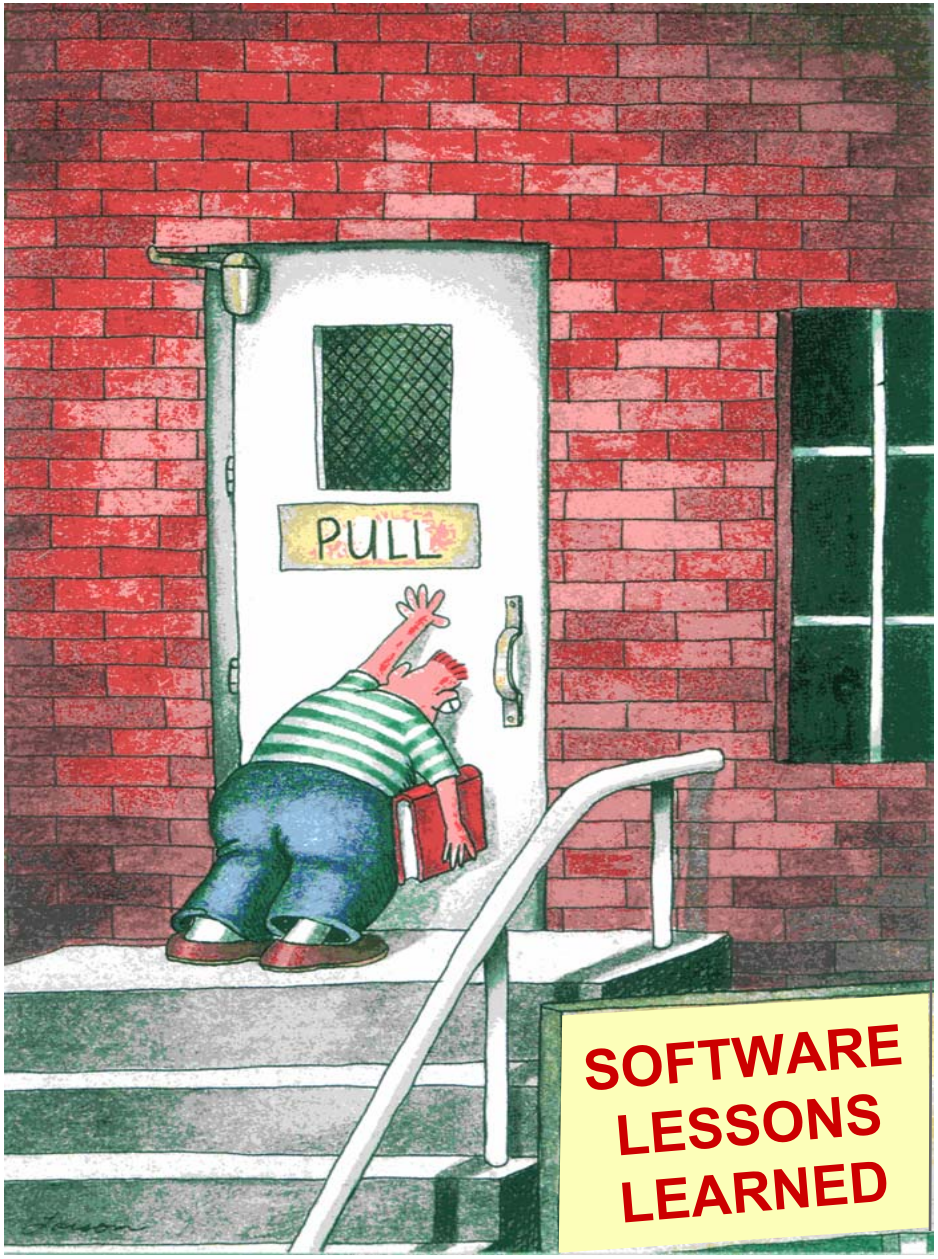
John McGarry, TACOM-ARDEC

Some Questions …

Why do we always seem to be trying to solve the same problems in our software intensive programs?

Are we focusing on the symptoms or the causes of our software issues?

How do we define program success?


PULL
SOFTWARE
LESSONS
LEARNED

Presentation Objectives

- Convey what we have learned through a systemic "Cross Program" analysis of multiple software intensive DOD programs
- Identify some of the recurring factors that materially impact software intensive acquisition and development efforts
- Provide some ideas on how we can improve based on the results from real program experiences

Presentation Outline

- TAI Initiative Overview
- Top-Level Analysis Results
- Systemic Analysis Approach
- Software Intensive Program Issues – Causes and Effects
- Next Steps

Independent Expert Program Reviews

- **IEPR Policy**
  - ACAT ID/IC programs shall conduct an IEPR after Milestone B and before CDR
  - IEPRs shall be considered for ACAT IA, II, and III programs
  - IEPR Implementation Plan provides guidance for implementing policy and will be staffed after DoD 5000 release

- **Tri-Service Assessment Initiative**
  - Primary implementation for conducting IEPRs
  - Sponsored by OSD - Software Intensive Systems Office
  - Three year history
  - Structured multiple-level assessment architecture

TAI Systemic Analysis Objectives

- Identify systemic issues that impact program success
- Understand their cause and effect relationships
- Develop recommendations to improve SIS acquisition:
  - policy and guidance
  - education and training
  - tactical and strategic decision making
- Provide DoD users with a source of objective - actionable - defensible information
  - Enterprise (OSD, Services, PEOs)
  - Program (PMs, staffs)
  - Technical Interface (DAU, SEI, IEPR WG, etc.)

Systemic Analysis Bottom Line

Use real program assessment data to:

- Identify
- Characterize
- Explain
- Correct

The problems that systematically impact Software Intensive Program success

TAI Systemic Analysis Team

- Kristen Baldwin, SIS
- Bob Charette, ITABHI
- Laura Dwinnell, Northrop Grumman
- Ken Smith, SEI
- Dave Zubrow, SEI

Top-Level Analysis Results

Top-Level Conclusions

- Assessment results show repeating trends:
  - across assessed programs
  - regardless of program characteristics
  - over a long time period

- Program failure is related to a combination of unrealistic enterprise constraints and expectations, and poor program execution:
  - can be poor program execution alone
  - can be a mismatch between expectations and program execution capability

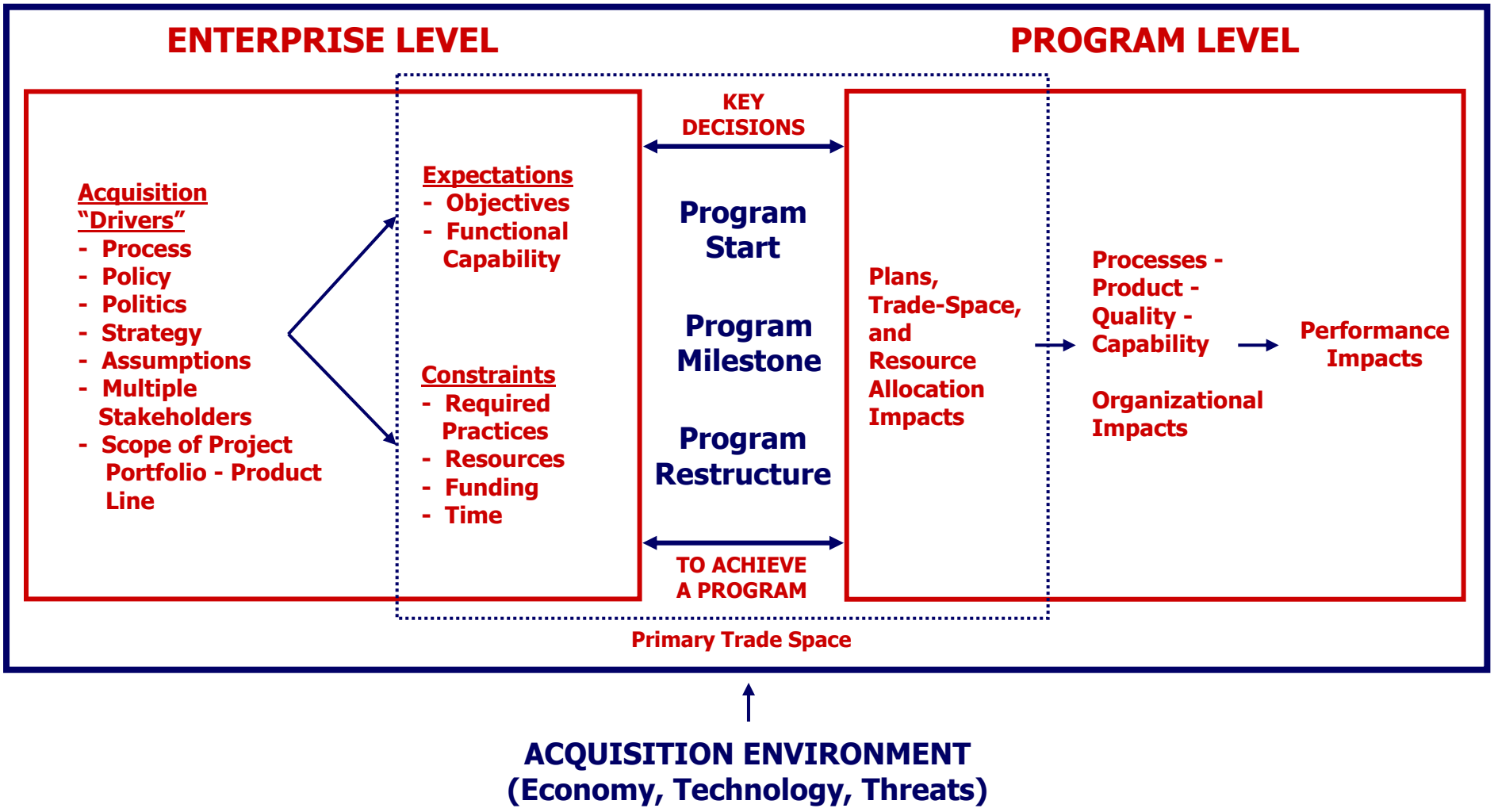
Top-Level Conclusions (cont.)

- Causative issues produce different performance symptoms in different programs
  - single issue can cause many symptoms
  - many unique issue combinations
  - relatively complex interactions

- Past DoD corrective actions and associated policy have largely focused on the symptoms and not the causative issues
  - usually on a symptom by symptom basis
  - symptoms are perceived as causes of failure

The sum of Enterprise constraints and expectations and poor Program execution limits the trade space on our programs - this reduces the opportunity to make changes to improve program performance

Systemic Analysis Model

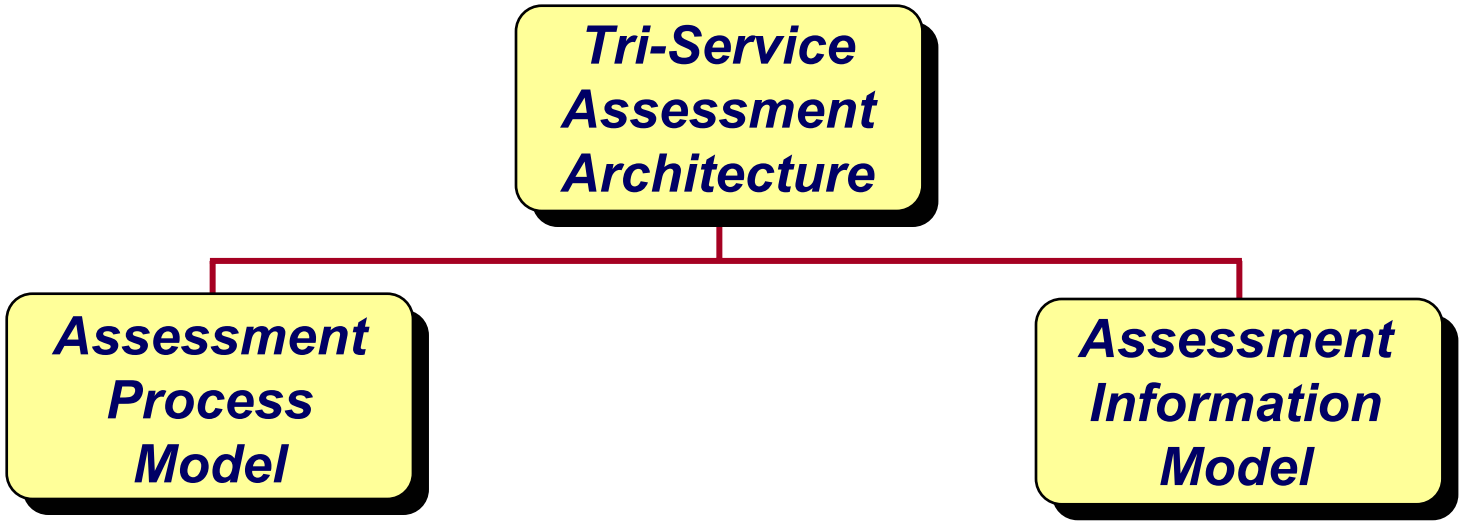


And In Addition …

- The number of enterprise mandated requirements is significantly large
- These requirements may be uncoordinated, unvalidated, and unfunded
- What we do is different from what we teach
- "Best Practices" require "Best Execution" to influence success
- Program success factors are not always the opposite of program failure factors
- Successful programs are more pro-active - they aggressively "manage the trade space"

Systemic Analysis Approach

TAI Assessment Architecture

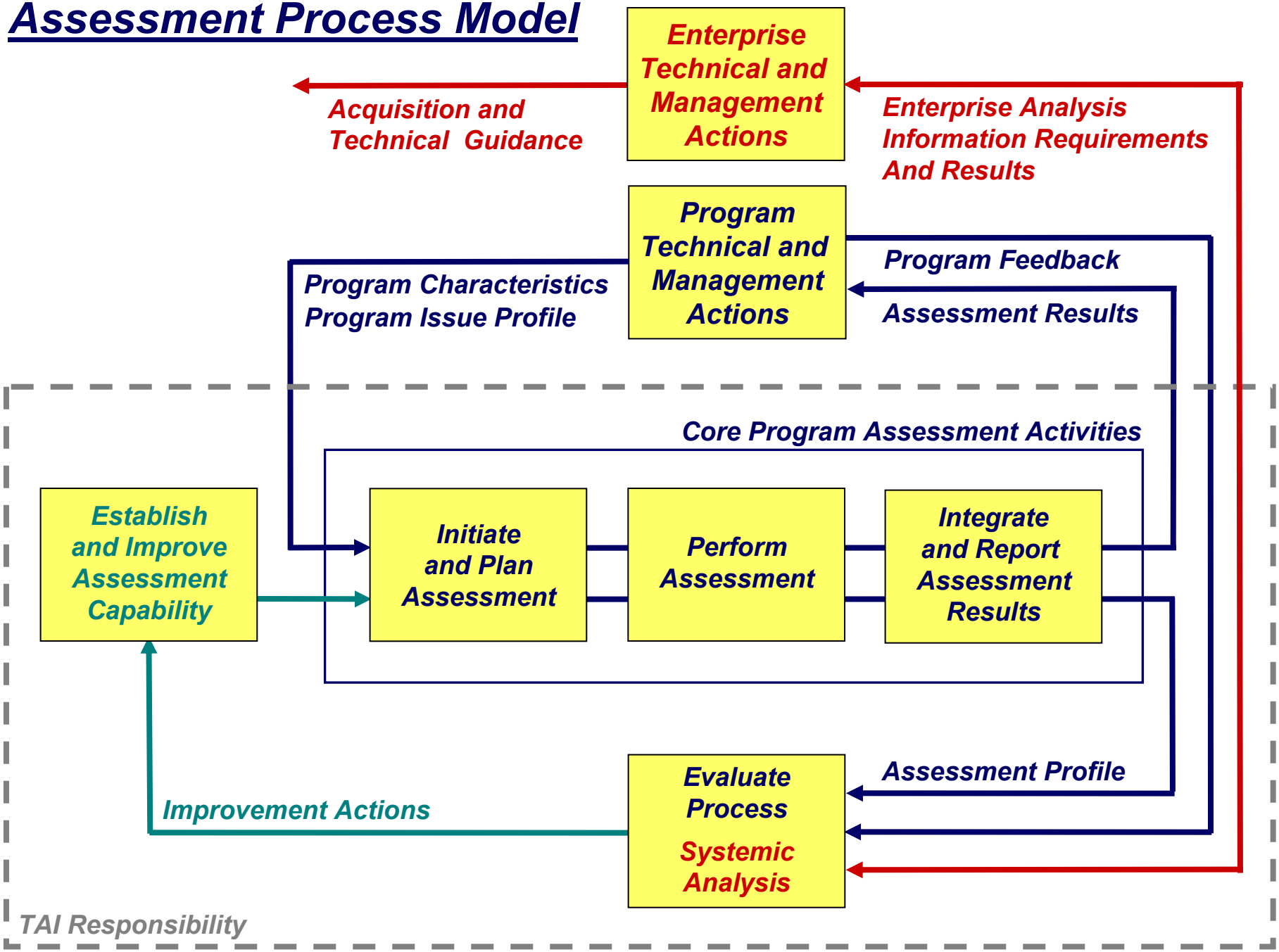


- Identify and prioritize program issues
- Develop value-added recommendations
- Generates consistent information sets

- Generic Program issue structure
- Defines assessment "scope"
- Flexible typology

Both Components are Required for Individual Program Assessment and Systemic Cross-Program Analysis

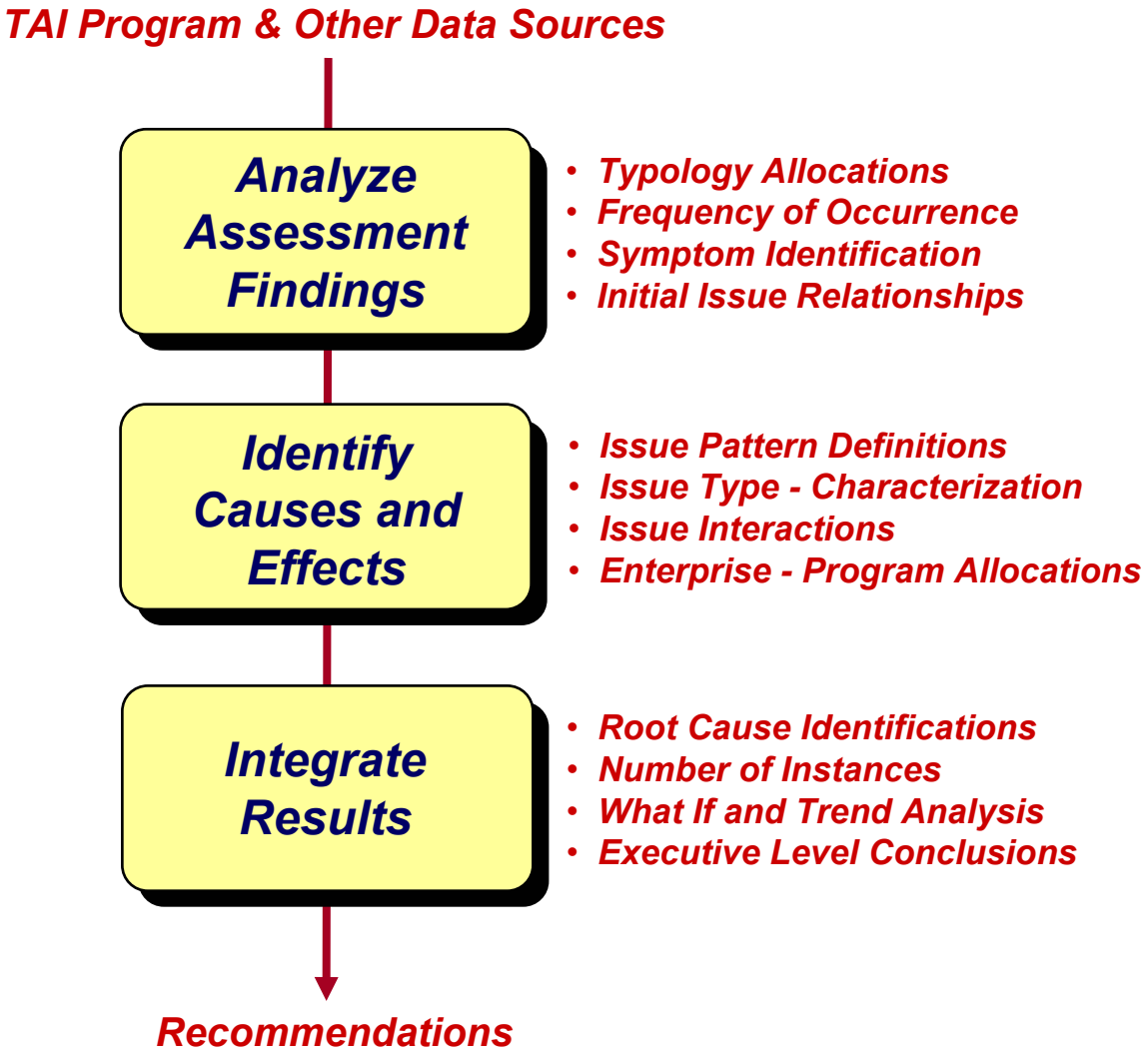
Assessment Process Model



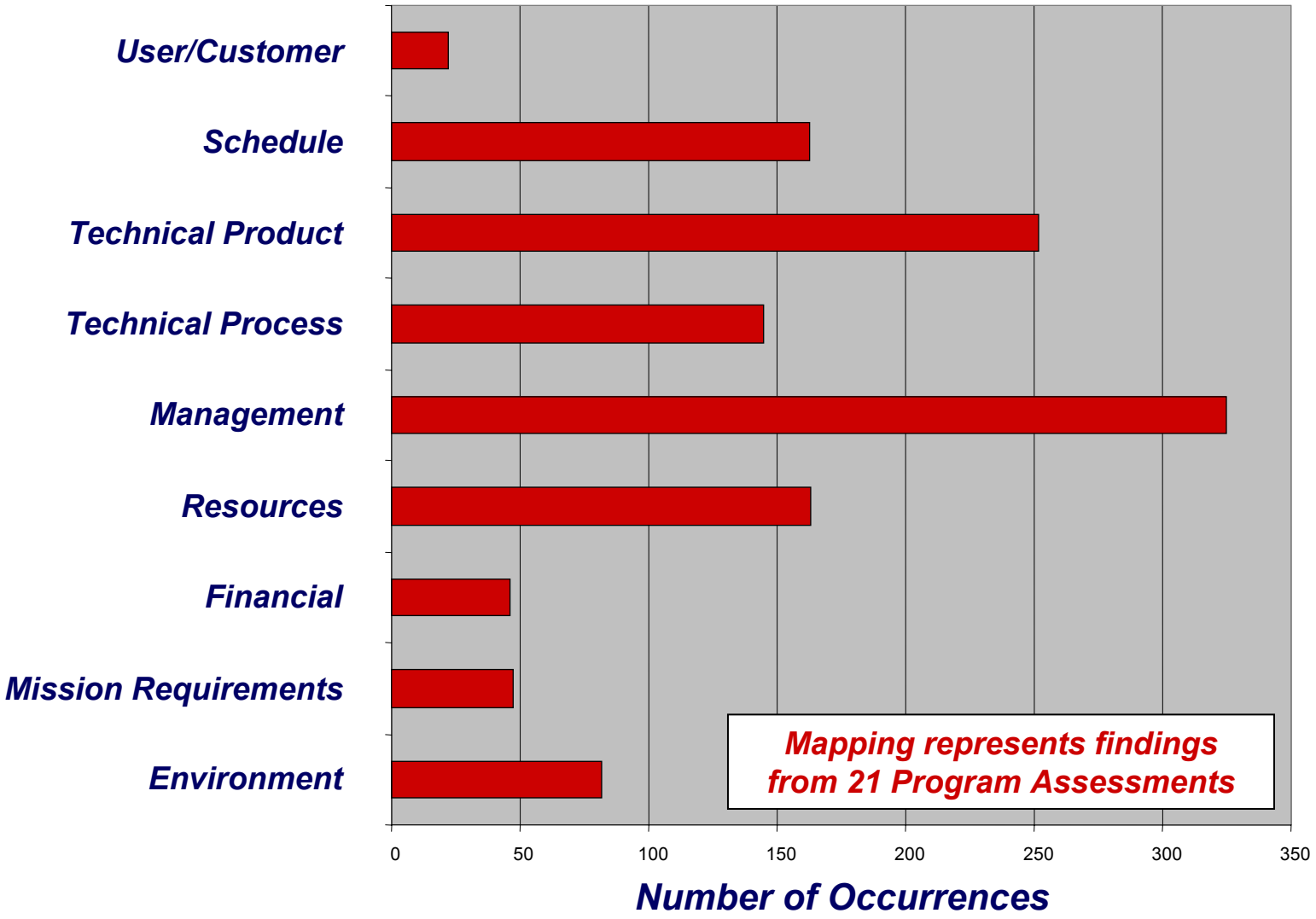
Assessment Information Model

- User / Customer
- Schedule
- Technical Product
- Technical Process
- Management
- Resources
- Financial
- Mission Requirements
- Environment
- Project Specific

Systemic Analysis Process



Findings Mapped To Information Model



Leading Second Level Issue Areas

- Program - Organizational Management
- Development Progress
- Product Quality
- Product Requirements
- Technical Process Capability
- Personnel Resources
- Acquisition Strategy
- Project Planning

All issue areas had both Enterprise and Program Level Allocations

Recurring Program Symptoms

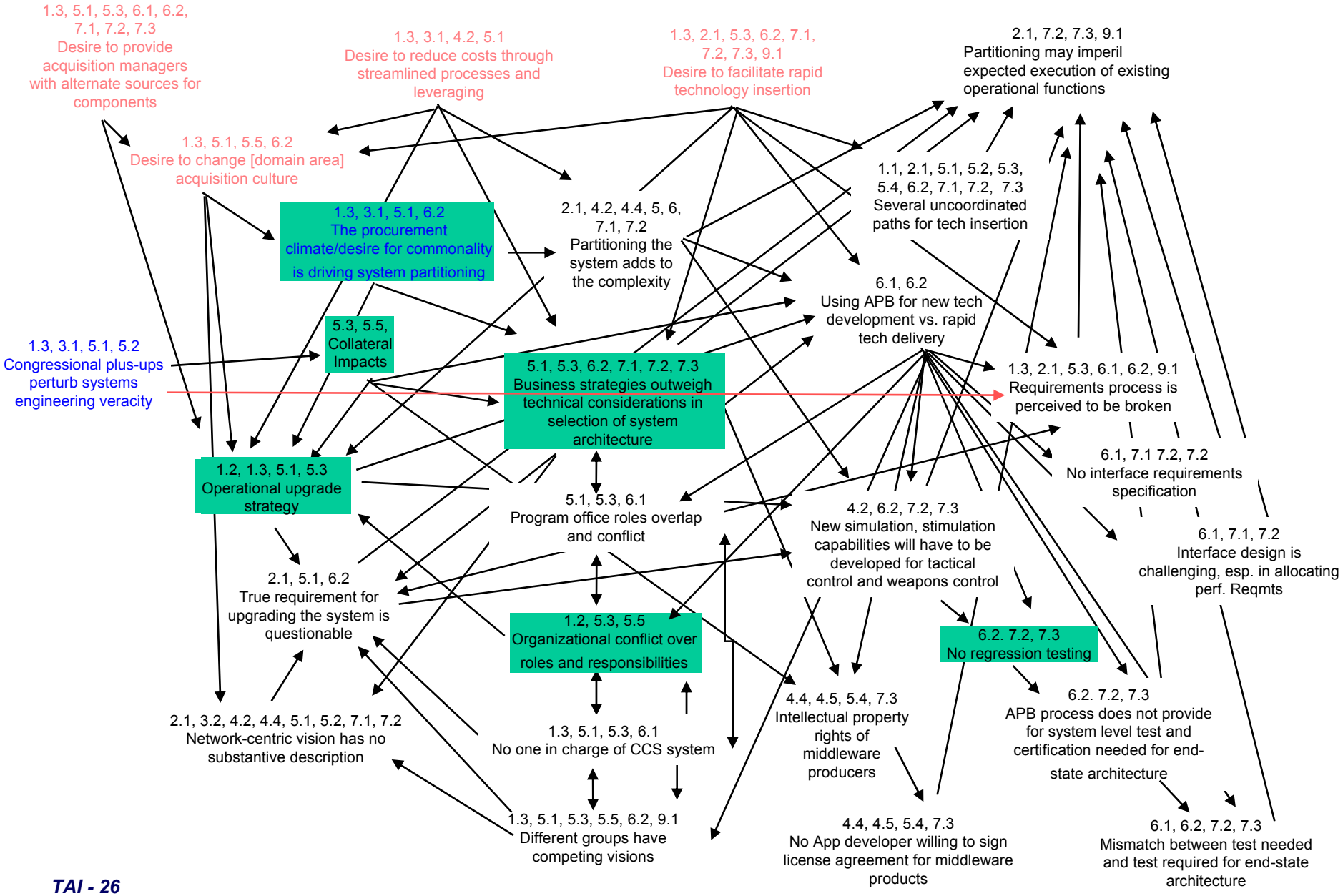
- Over Budget
- Late
- Poor Product Quality
- Poor Communication
- Inadequate / Late Decisions
- Costly Technology Refresh
- Poor Morale
- No Product Line Architecture
- Poor Interoperability
- Rework

Software Intensive Program Issues Causes and Effects

Initial Findings Analysis Indicated:

- That identified issues were connected in recurring patterns or sequences
- That issue sequences were of different types:
  - Triggering Sequences
  - Failure Sequences
  - Success Sequences
  - Symptoms / Impacts
- That the issue interdependencies - the causes and effects - were extremely complex

Tri-Service Assessment Initiative



Phased Approach to Causal Analysis

- Identify End-to-End Issue Sequences (Top-Down)
- Quantify Recurring Issue Patterns
- Define Terminology
- Identify and Count Issue Sequence Components (Bottom-Up)
- Quantify and Characterize
- Integrate Sequences into Causal Threads
- Numerical and Scenario Analysis
- Assessment Process Feedback

Recurring Issue Patterns

- Unintended Policy Impacts
- Impacts of Congressional Mandates
- Inadequate Family of Systems Management
- Interoperability Clashes
- Premature System Deployment
- Poor Technology Refresh Management
- Ineffective Systems Engineering
- Compliant but Inadequate Process Capability
- Overly Aggressive Program Concurrency

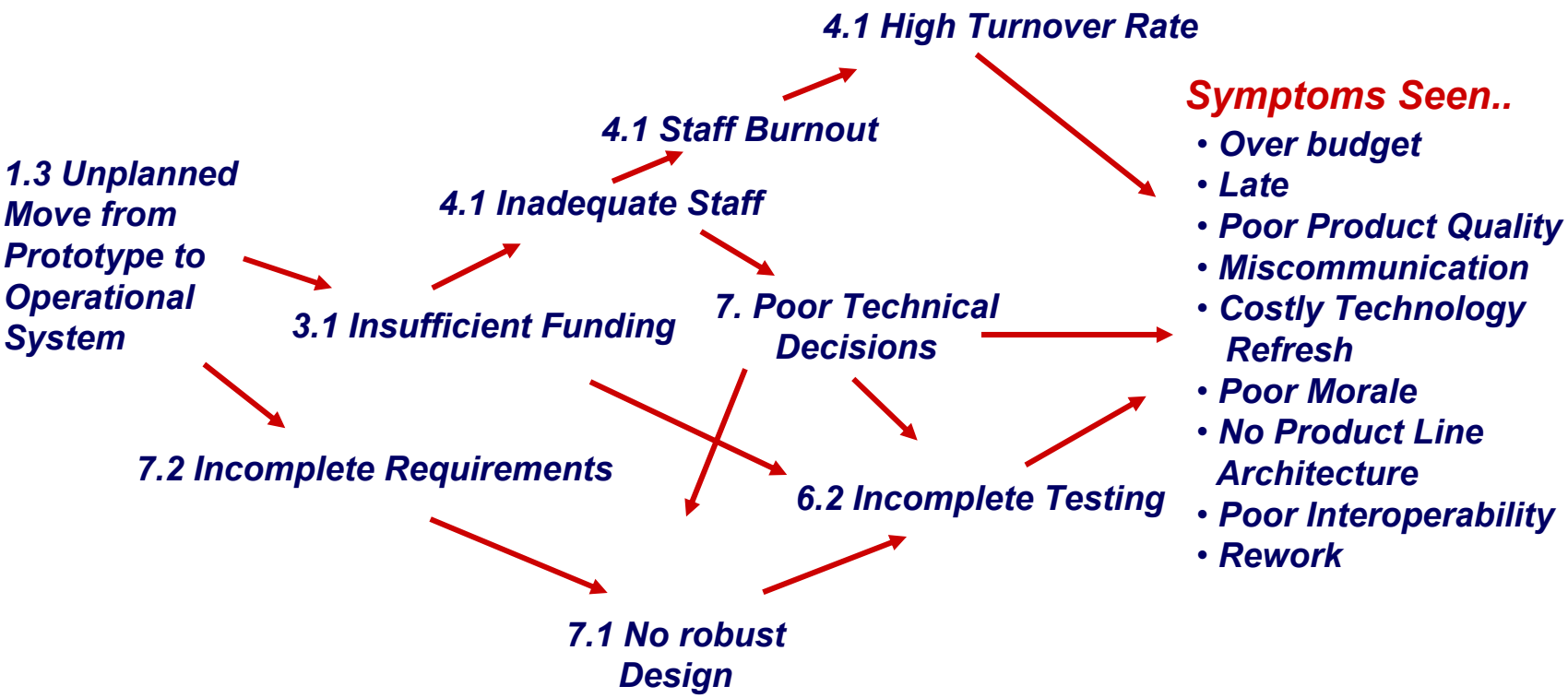
Recurring Issue Patterns (cont.)

- Inadequate / Inappropriate Development Approach Chosen or Mandated
- Intellectual Property Rights and Proprietary Issues
- Production Capability Inadequately Addressed
- Incomplete Risk Management / Measurement
- Inadequate Resource Infrastructure
- Inadequate Change Management
- Disconnected Education & Training

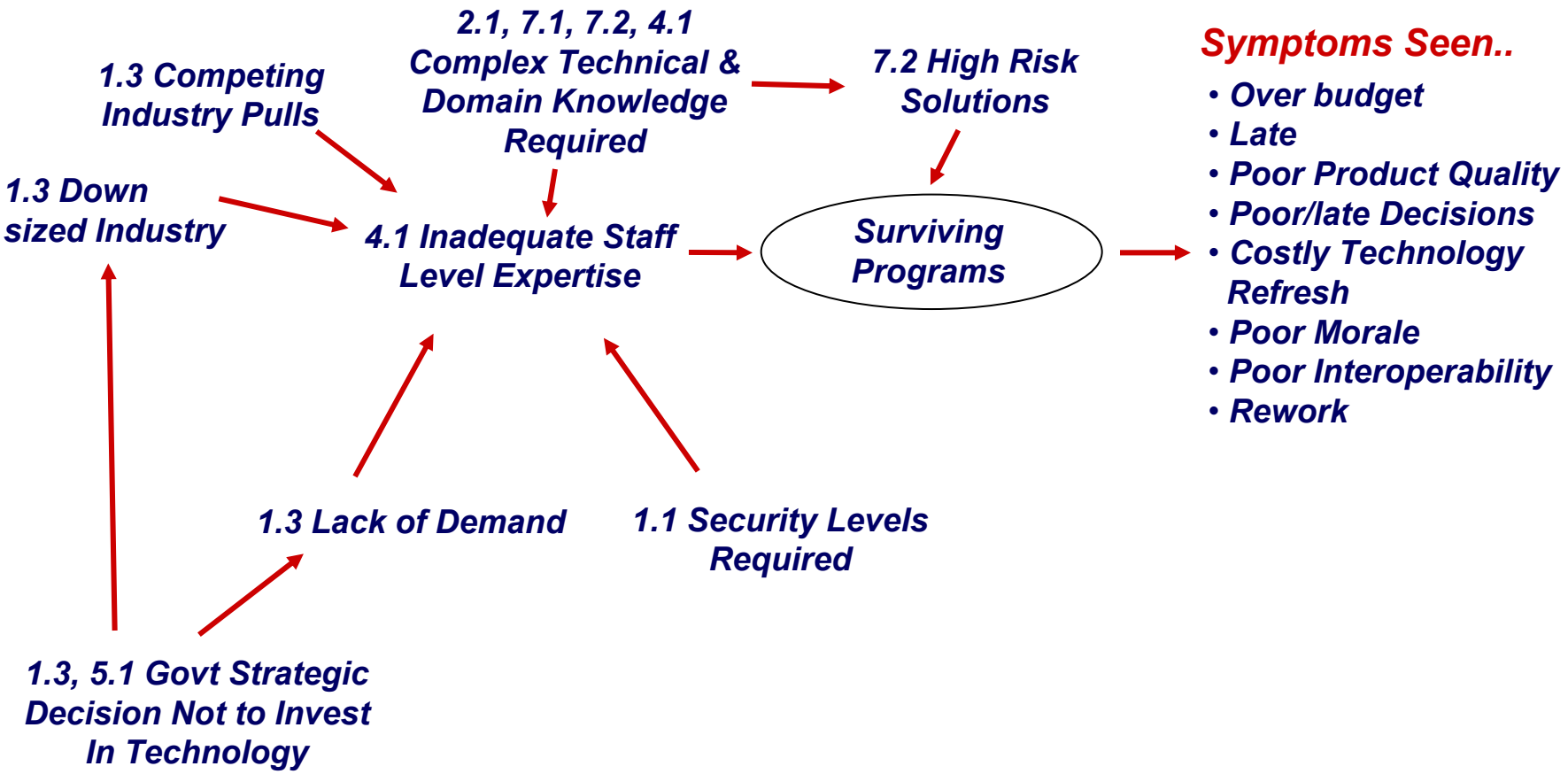
Unintended Policy Impacts

COTS vs. Supportability

Premature Systems Deployment



Inadequate Resource Infrastructure



The Way Forward - Recommendations

- *Improve and integrate IEPR results - Refine Systemic Analysis Techniques*
- *Identify and Quantify both Issue and Success related Causal Patterns*
- *View Enterprise and Program Level Corrective Actions as Part of an Integrated Solution*
- *Recognize and Address Issue and Performance Interdependencies*
- *Use a "Corrective Action Team" Approach to Address Prioritized Causative Issues*

Summary

- *Systemic analysis based on real program assessment results provides a unique opportunity to use actual data to make a difference*

- *The causes of program performance shortfalls are extremely complex - improvement strategies and associated action plans must address this complexity*

John McGarry
Executive for Performance Management

U.S. Army TACOM - ARDEC
AMSTA-AR-QAT
Picatinny Arsenal, NJ 07806-5000

(973) 724-7007
jmcgarry@pica.army.mil