Acquisition Research Program:
Creating Synergy for Informed Change

DoD Software Intensive Systems Development: A Hit and Miss Process

Brad R. Naegle

Software-Intensive System Development

Hits

- Tomahawk Missile
- Aegis
- Link 16
- F-22 Raptor

Misses

- Future Combat System
- B1 Bomber
- P8 Poseidon MMA
- F-35 Joint Strike Fighter

The Problem

- The Defense Acquisition System produces both successful and challenged software-intensive systems

The Symptoms

- Development cost & schedule hyperinflation
- Systems fielded with less capability than desired
- Operational capability delayed years
- Costly and difficult software sustainment

The Underlying Causes

- The DoD Requirements Generation System
  - Requires interpretation between Capabilities-Based terms (JCIDS) and Performance-Based terms (Performance Spec), and again to Detailed Specification
  - ***Purposely vague to garner maximum innovation***
  - ***Dependent on the developer to correctly interpret and propose innovative solutions***
  - ***Provides only a glimpse at the operational environment through the Operational Mode Summary/Mission Profile***
  - Information Assurance/Cyber Security needs

Causes Continued

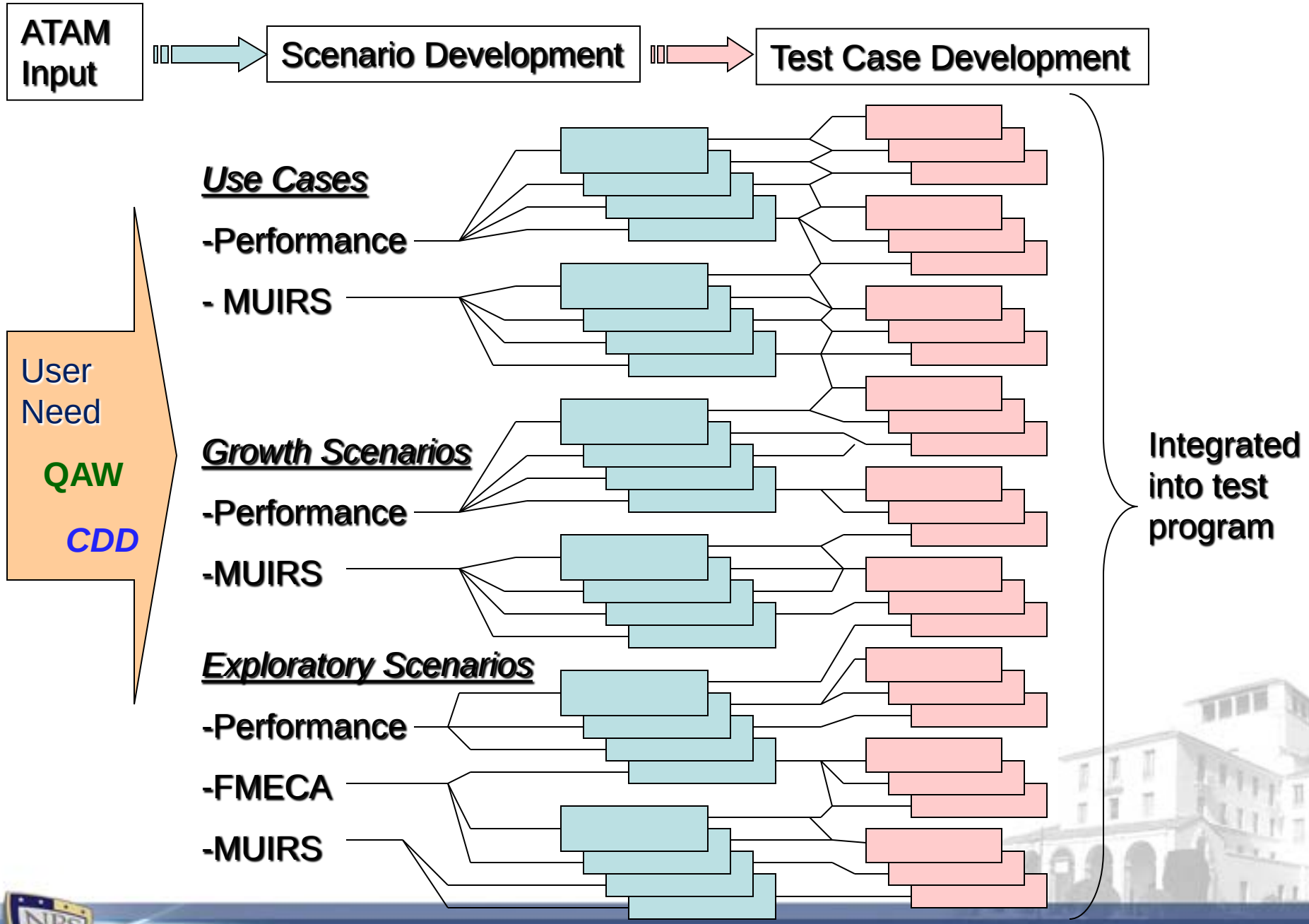
- Immature Software Engineering Environment
  - No industry-wide standards, protocols, formats, architectures, tools, or languages
  - No sustainability standards or architectures
  - Very limited capability for reuse
  - ***Totally dependent on clear, unambiguous, and complete requirements*** (Half or more of the software development effort occurs before PDR)
  - Requirements creep and late definition disastrous to the effort

Causes Continued

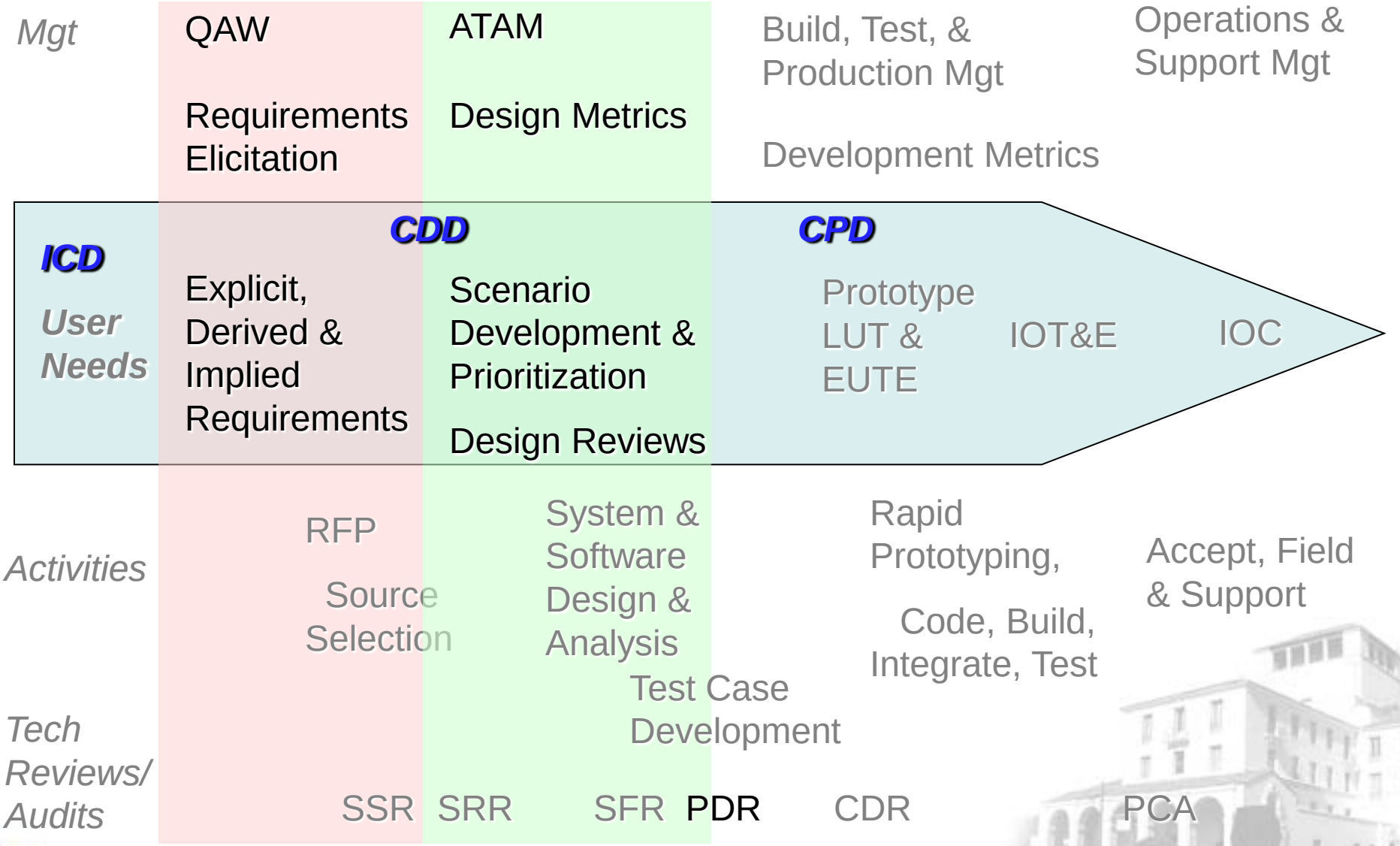
- The Defense Acquisition System
  - Pressure to reduce cycle time can impact front-end processes *(‘Get RFP on the street!’)*
  - No consistent methodologies for driving software architecture or sustainability design
  - Information Assurance/Cyber Security needs drives developers to typically build software from scratch
  - Software TRLs ineffective at reducing development risk
  - Contractor is assessed for risk (CMMI), but PM team has no ‘maturity’ requirements

Attacking the Causes

- Implementation of analyses, tools, and processes
  - SEI's Quality Attribute Workshop (QAW)
    - A more complete inventory of requirements
  - MUIRS Analysis
    - Analyses for sustainability and safety/security needs
  - SEI's Architectural Trade-off Analysis Methodology sm
    - Clarifies context and drives architectural design
    - Connects user needs to system design to test program
  - FMECA
    - Identifies critical and non-critical system attributes
  - SEI's Software Acquisition (SA)-CMM
    - Assesses the Government's PM team maturity


ATAM Input
Scenario Development
Test Case Development
User Need
QAW
CDD
Use Cases
-Performance
- MUIRS
Growth Scenarios
-Performance
-MUIRS
Exploratory Scenarios
-Performance
-FMECA
-MUIRS
Integrated into test program

QAW & ATAM Integration into SW Lifecycle Management



SA-CMM

Level	Focus	Key Process Areas
5	Optimizing	Continuous process improvement, Acquisition Innovation Management, Continuous Process Improvement
4	Quantitative	Quantitative management, Quantitative Acquisition Management, Quantitative Process Management
3	*Defined*	*Process standardization, Training Program Management Acquisition Risk Management, Contract Performance Management, Project Performance Management User Requirements, Process Definition and Maintenance*
2	Repeatable	Basic project management, Transition to Support Evaluation, Contract Tracking and Oversight, Project Management, Requirements Development and Management, Solicitation, Software Acquisition Planning
1	Initial	Competent people and heroics

Summary

- Using these tools, analyses, and processes will help address the ***causes*** of software development problems.
- The PM team must mature beyond ‘Competent People and Heroics’ to manage the complex software development challenge
- A mature PM team effectively implementing the tools, analyses, and Processes will result in more consistently successful software-intensive systems development