Carnegie Mellon
Software Engineering Institute

A Taxonomy of Operational Risks

Brian Gallagher

Director, Acquisition Support

Carnegie Mellon
Software Engineering Institute

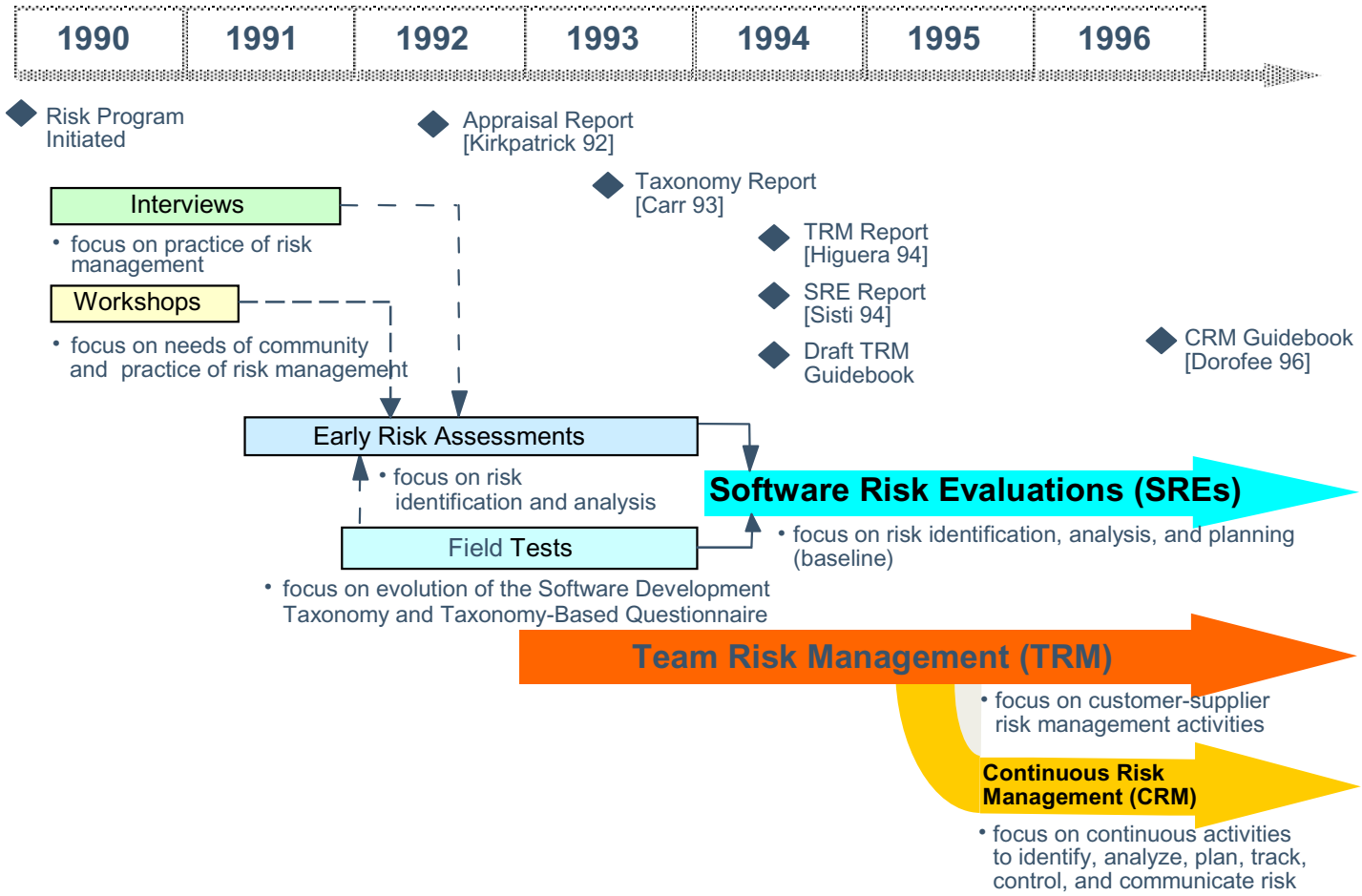
Operational Risk

"By its nature, the uncertainty of war invariably involves the acceptance of risk...Because risk is often related to gain, leaders weigh risks against the benefits to be gained from an operation."

NDP-1 (Naval Warfare)

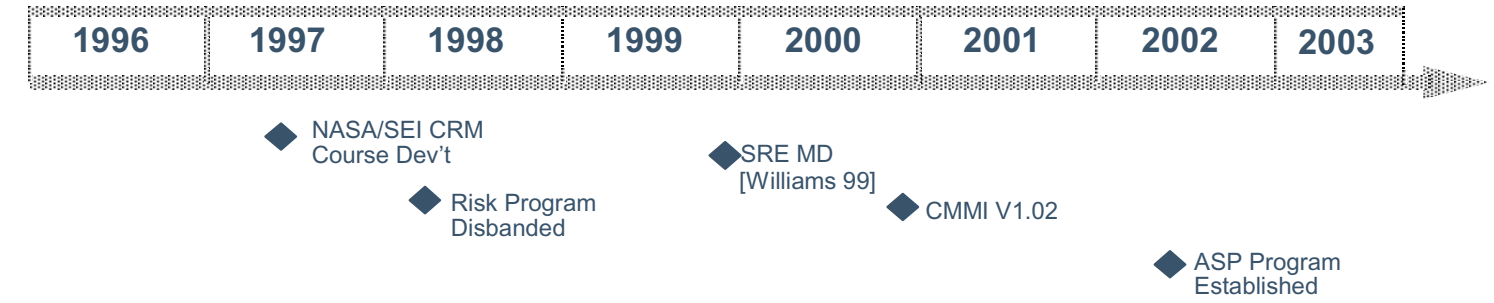
Carnegie Mellon
Software Engineering Institute

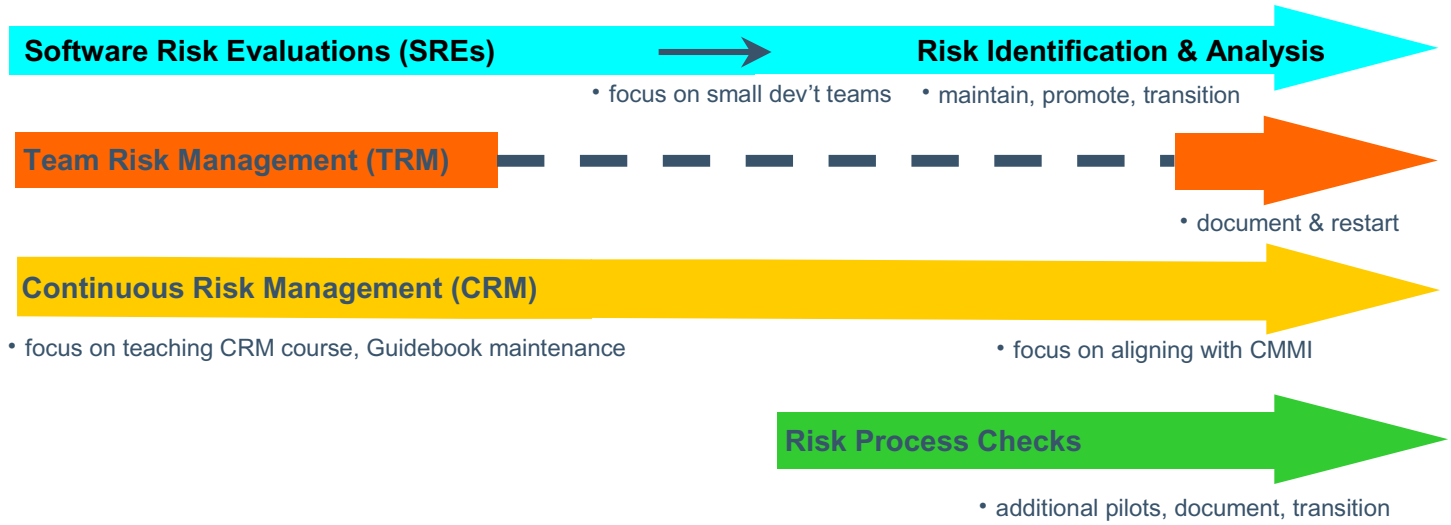
History of SEI Risk Management[1]



Carnegie Mellon
Software Engineering Institute

History of SEI Risk Management$_2$





Carnegie Mellon
Software Engineering Institute

Key Aspects of Continuous Risk Management

Identify – Continually asking, "what could go wrong?"

Analyze – Continually asking, "which risks are most critical to mitigate?"

Plan – Developing mitigation approaches for the most critical risks

Track – Tracking the mitigation plan and the risk

Control – Making decisions based on data

Communicate – Ensuring a free-flow of information throughout the project

Carnegie Mellon
Software Engineering Institute

SEI's Risk Taxonomy

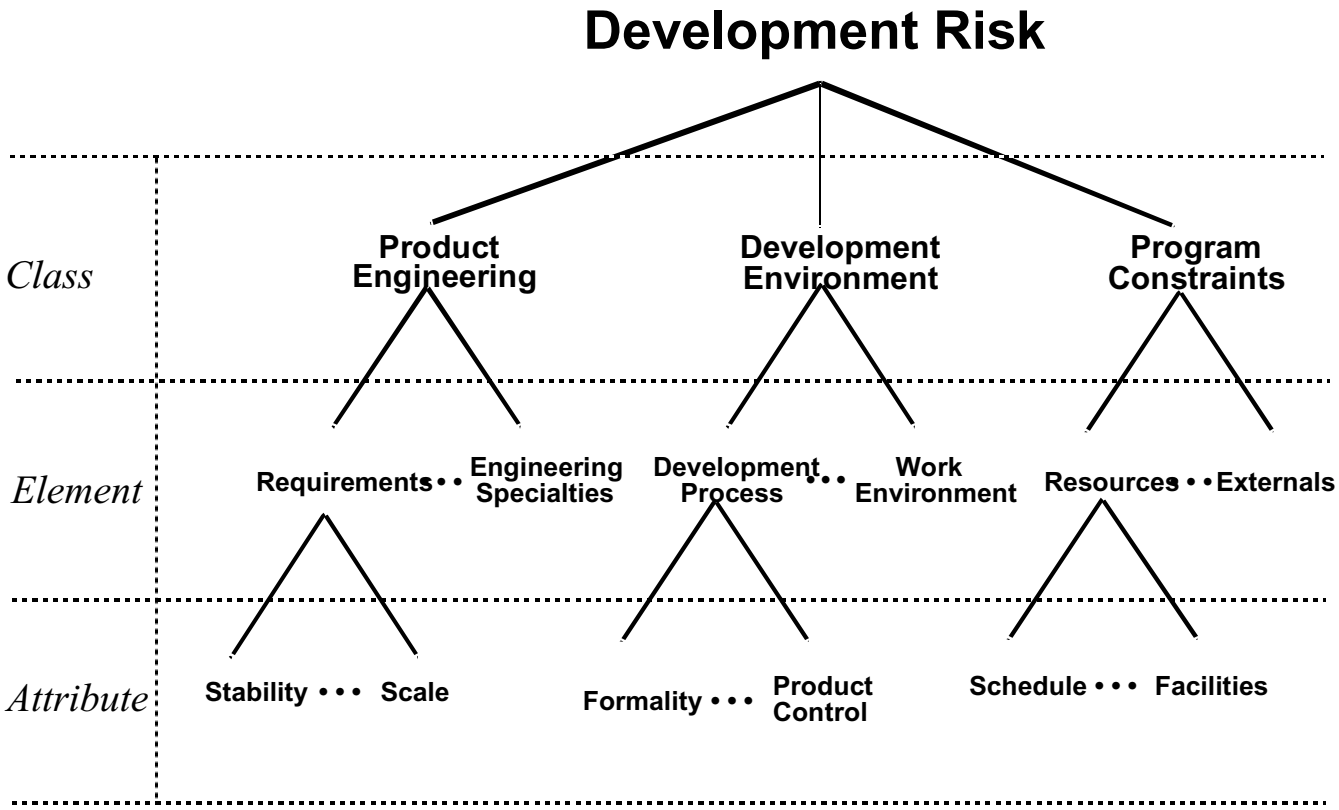
Developed in 1993 to help software-intensive system developers systematically identify risks

Used with the SEI's Software Risk Evaluation process or other risk identification techniques

Used as a "checklist" or expanded "radar screen" to ensure a greater number of potential risks are identified when doing on-going risk identification

Carnegie Mellon
Software Engineering Institute

Taxonomy Structure



Carnegie Mellon
Software Engineering Institute

Development Taxonomy

A. Product Engineering

1. Requirements
   a. Stability
   b. Completeness
   c. Clarity
   d. Validity
   e. Feasibility
   f. Precedent
   g. Scale
2. Design
   a. Functionality
   b. Difficulty
   c. Interfaces
   d. Performance
   e. Testability
   f. Hardware Constraints
   g. Non-Developmental Software
3. Code and Unit Test
   a. Feasibility
   b. Testing
   c. Coding/Implementation
4. Integration and Test
   a. Environment
   b. Product
   c. System
5. Engineering Specialties
   a. Maintainability
   b. Reliability
   c. Safety
   d. Security
   e. Human Factors
   f. Specifications

B. Development Environment

1. Development Process
   a. Formality
   b. Suitability
   c. Process Control
   d. Familiarity
   e. Product Control
2. Development System
   a. Capacity
   b. Suitability
   c. Usability
   d. Familiarity
   e. Reliability
   f. System Support
   g. Deliverability
3. Management Process
   a. Planning
   b. Project Organization
   c. Management Experience
   d. Program Interfaces
4. Management Methods
   a. Monitoring
   b. Personnel Management
   c. Quality Assurance
   d. Configuration Management
5. Work Environment
   a. Quality Attitude
   b. Cooperation
   c. Communication
   d. Morale

C. Program Constraints

1. Resources
   a. Schedule
   b. Staff
   c. Budget
   d. Facilities
2. Contract
   a. Type of Contract
   b. Restrictions
   c. Dependencies
3. Program Interfaces
   a. Customer
   b. Associate Contractors
   c. Subcontractors
   d. Prime Contractor
   e. Corporate Management
   f. Vendors
   g. Politics

Carnegie Mellon
Software Engineering Institute

Operational Organizations

An Operational organization is any group of individuals teamed together to carry out a mission.

Operational organizations consists of mission elements or teams that carry out mission requirements or subsets of requirements.

Requirements could come from external customers or from internal sources.

Carnegie Mellon
Software Engineering Institute

Examples

Examples of Operational organizations:

- military units
- educational institutions
- health care facilities
- fire and police units
- non-profit organizations

Carnegie Mellon
Software Engineering Institute

Task Defined

Operational organizations perform tasks to satisfy mission requirements.

Mission-essential tasks: A mission-essential task is any task that directly accomplishes mission requirements.

examples: flight operations, satellite control, mission management, etc.

Mission-support tasks: A mission-support task is any task that supports the accomplishment of mission requirements.

examples: spares replenishment, mission planning, new employee orientation, etc.

Carnegie Mellon
Software Engineering Institute

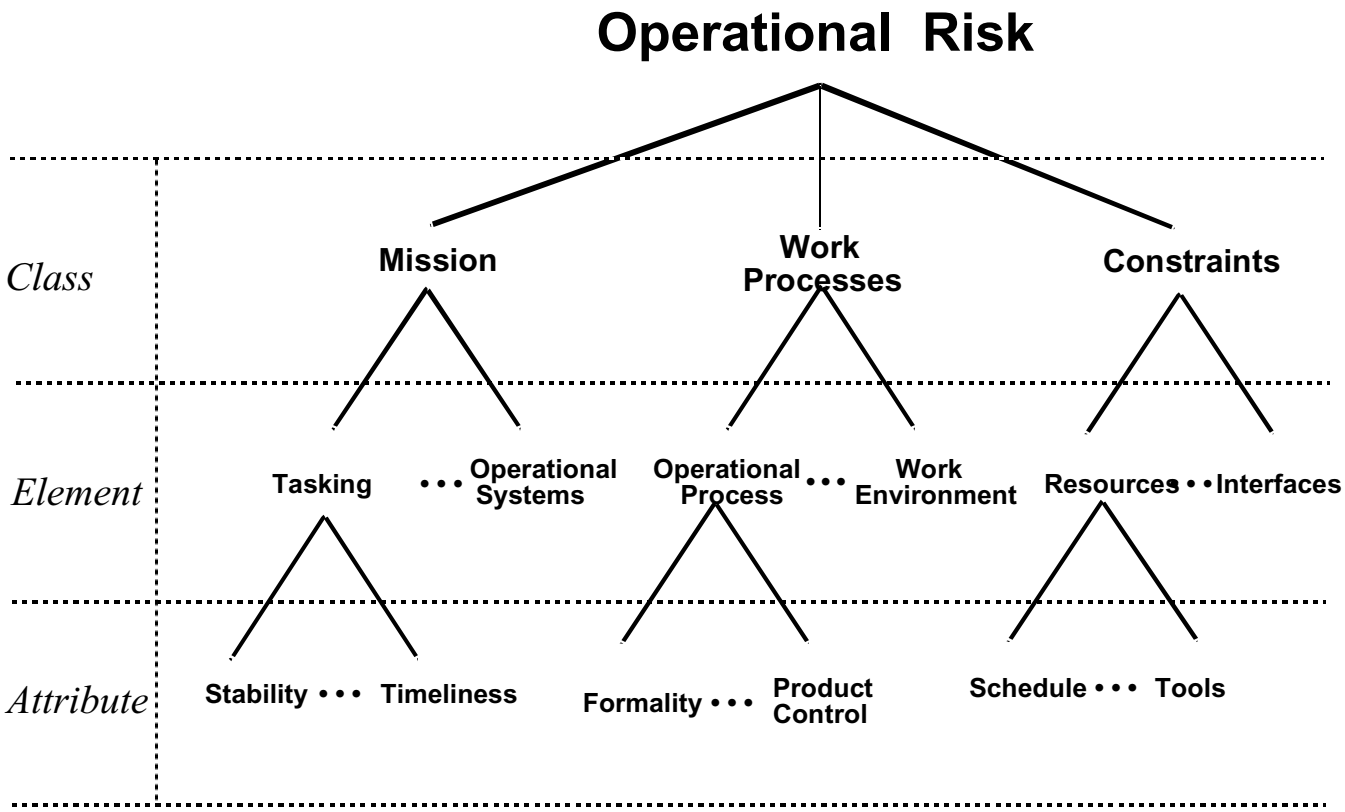
Identifying Operational Risks

When identifying risks in an operational environment, the Development Taxonomy doesn't work well

- Operational personnel don't do development per se
- Operational personnel don't feel comfortable with the definitions in the original Taxonomy
- Operational personnel need systematic tools to help identify mission-related risks

Carnegie Mellon
Software Engineering Institute

Constructing an Operational Taxonomy



Carnegie Mellon
Software Engineering Institute

Taxonomy of Operational Risks

A. Mission

1. Tasking, Orders and Plans
   a. Stability
   b. Completeness
   c. Clarity
   d. Validity
   e. Feasibility
   f. Precedent
   g. Timeliness
2. Mission Execution
   a. Efficiency
   b. Effectiveness
   c. Complexity
   d. Timeliness
   e. Safety
3. Product
   a. Usability
   b. Effectiveness
   c. Timeliness
   d. Accuracy
   e. Correctness
4. Operational Systems
   a. Throughput
   b. Suitability
   c. Usability
   d. Familiarity
   e. Reliability
   f. Security
   g. Inventory
   h. Installations
   i. System Support

B. Work Processes

1. Operational Processes
   a. Formality
   b. Suitability
   c. Process Control
   d. Familiarity
   e. Product Quality
2. Maintenance Processes
   a. Formality
   b. Suitability
   c. Process Control
   d. Familiarity
   e. Service Quality
3. Management Process
   a. Planning
   b. Organization
   c. Management Experience
   d. Program Interfaces
4. Management Methods
   a. Monitoring
   b. Personnel Management
   c. Quality Assurance
   d. Configuration Management
5. Work Environment
   a. Quality Attitude
   b. Cooperation
   c. Communication
   d. Morale

C. Constraints

1. Resources
   a. Schedule
   b. Staff
   c. Budget
   d. Facilities
   e. Tools
2. Policies
   a. Laws and Regulations
   b. Restrictions
   c. Contractual Constraints
3. Program Interfaces
   a. Customers/User Community
   b. Associate Agencies
   c. Contractors
   d. Senior Leadership
   e. Vendors
   f. Politics

http://www.sei.cmu.edu/publications/documents/05.reports/05tn036.html

Carnegie Mellon
Software Engineering Institute

Example Class/Element/Attribute: Mission

A. Mission
In an operational environment, a *mission* is considered to be the primary reason for the existence of the operational organization. The mission consists of a set of defined tasks that produce a product or service for a customer. The mission could be defense intelligence operations, banking, retail sales, manufacturing, or a variety of other missions, including those performed by civil agencies.

The elements of the Mission class of operational risks cover traditional aspects of the mission, including planning, execution, and the products and services provided. Mission elements include attributes of the operational systems and the organizations that operate those systems.

1. Tasking, Orders, and Plans
The Tasking, Orders, and Plans element contains attributes that are used to characterize aspects of the information contained in the tasks, orders, and plans of an operational organization. These attributes also describe the ability of an operational system and the organization that operates it to respond to requests. The following attributes characterize the Tasking, Orders, and Plans element.

a. Stability
The Stability attribute refers to the frequency with which tasks, orders, or plans change and the effect this has on the operational organization. It can also refer to the organizations that submit tasks or orders to an organization for execution. This attribute also addresses the flexibility of the operational entity in responding to changing tasks, orders, and plans and to handling multiple sources of tasks, orders, and plans.

Carnegie Mellon
Software Engineering Institute

A "Short" Taxonomy-based Questionnaire

A. Mission

Consider risks to the operation that can arise because of the nature of the mission that your organization is trying to accomplish.

- **Tasking, Orders, and Plans**

Question: Are there risks that could arise from the way the mission is tasked, orders are provided, or operational plans developed?
Examples:

a. Stability
b. Completeness
c. Clarity
d. Validity
e. Feasibility
f. Precedent
g. Timeliness

Carnegie Mellon
Software Engineering Institute

Using the Taxonomy of Operational Risks

The Taxonomy can be used:

- to establish a baseline set of operational risks
- to perform ongoing operational risk identification
- to help identify weaknesses in current operational capabilities and to help establish new statements of operational need
- when working with acquisition or development organizations to identify the operational risks associated with accepting new systems into operational use
- to participate with acquisition or development organizations using Team Risk Management techniques

Carnegie Mellon
Software Engineering Institute

Example: System Acceptance Risks

Context:

A military unit is responsible for operating satellite systems. An acquisition organization is acquiring a replacement system to consolidate operations at one location and upgrade the hardware and software to prepare for future acceptance of new satellite systems.

The program was late, and tension between the operators, the acquirers, and the developers was high.

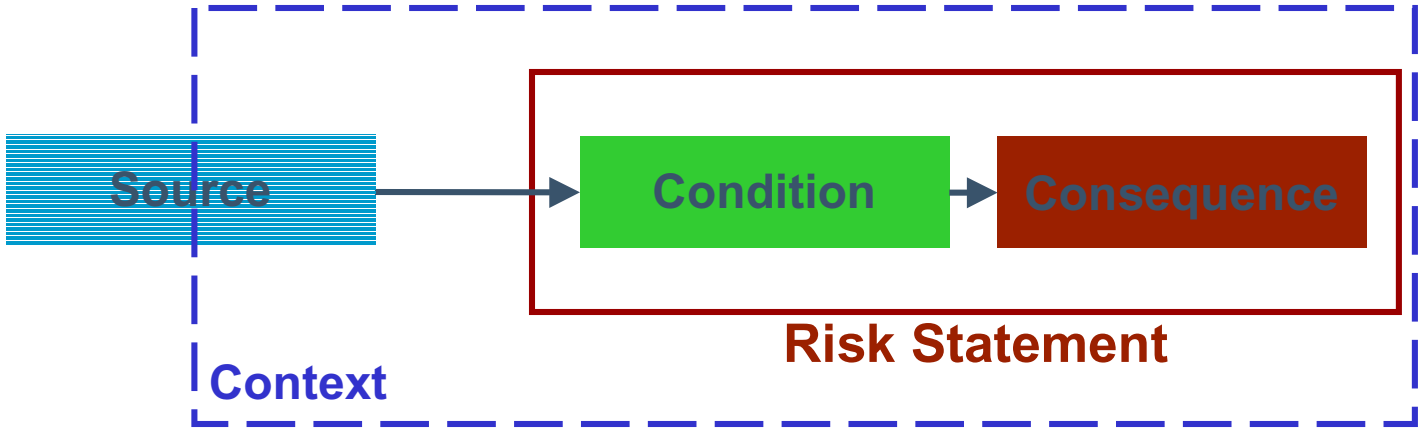
The SEI participated in a risk assessment using the SRE process and the Taxonomy of Operational Risks at the operational facility to help identify risks of accepting the new system and to uncover any root causes of the program delays.

During the two-day risk identification and analysis activities, stakeholders from the operational squadron, operational test personnel, Contractor Logistics Support (CLS), and site management wrote seventy (70) risk statements over the course of four interview sessions.

The Risk Statement

A “standard” format for risk statements provides:

- clarity
- consistency
- a basis for future risk processing



A good Risk Statement is

- fact-based
- actionable
- brief

Carnegie Mellon
Software Engineering Institute

Example Risk Statements

ORD does not levy requirements at the level of capability of legacy systems; system will be less capable, loss is visible at general officer level

Loss of key technical experts (significant attribution); loss of continuity

Positive "spin" put on info going up the chain; expectation mismatch

Roles and responsibilities not defined under this implementation of TSPR. (Insight vs. Oversight); Confusion, delays, who's responsible, who's leading

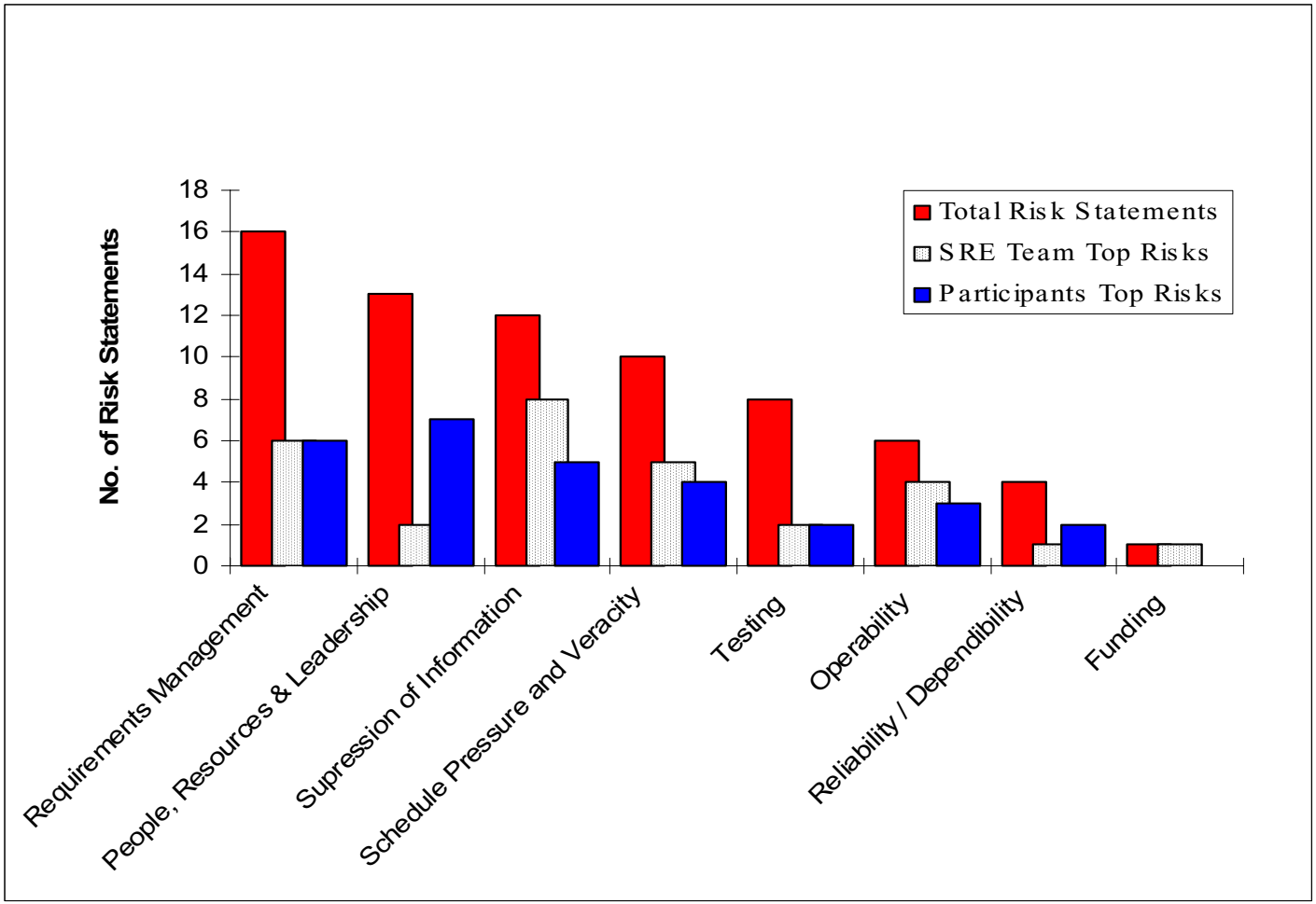
There is no official program schedule; Can't plan. Can't determine when to move personnel (out-year O&M and personnel costs)

Test resources at the factory are currently insufficient; Late discovery of problems

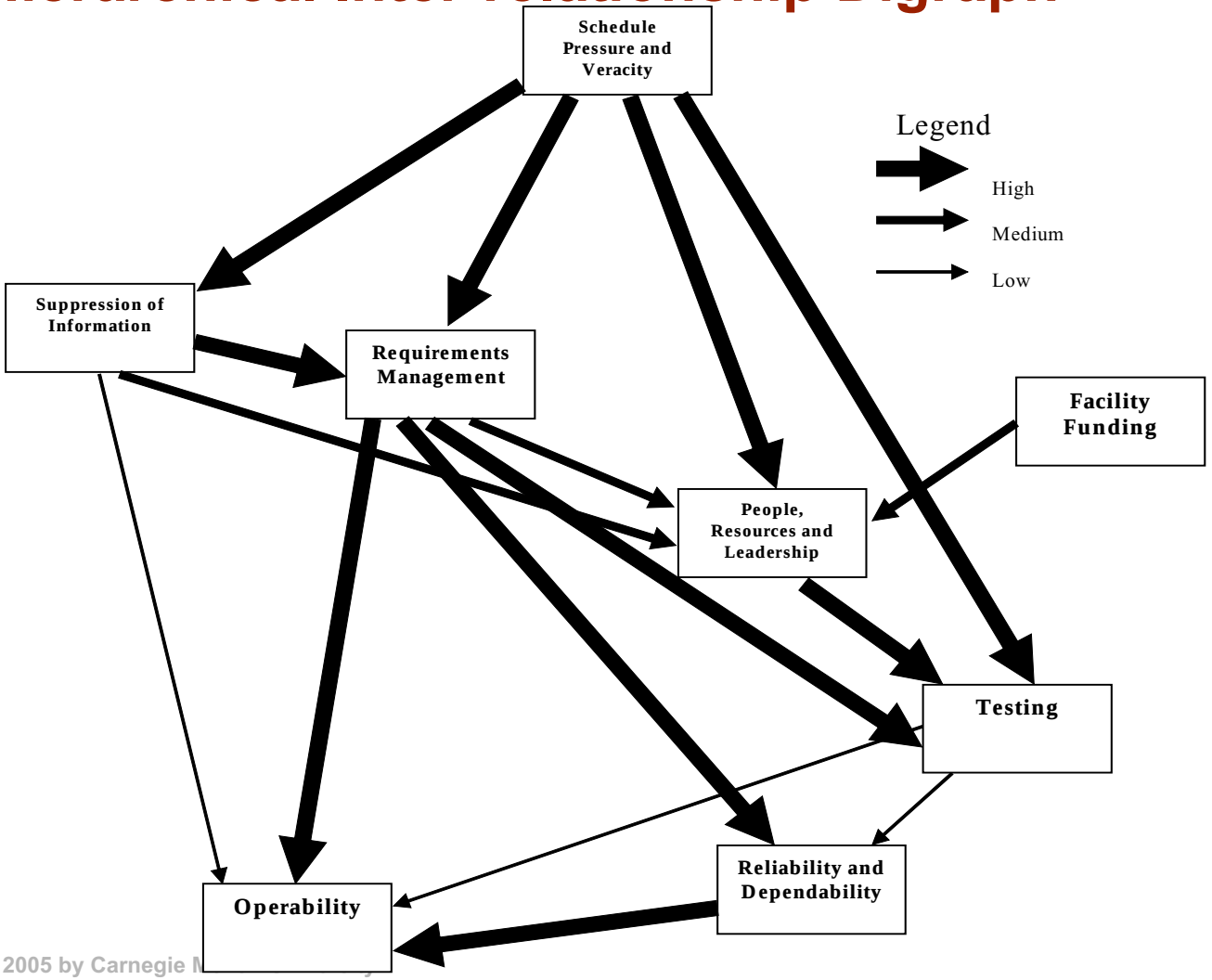
Training suite is sub-optimal, does not meet expectations or requirements, cannot perform integrated crew training; Will force training and evaluation on OPS floor

Carnegie Mellon
Software Engineering Institute

Risk Areas Identified



Hierarchical Inter-relationship Digraph



Carnegie Mellon
Software Engineering Institute

Outcome

Risk assessments were also done at the developer's location using the development taxonomy and at acquirer's location using the SA-CMM as a "taxonomy" to get their unique perspectives

With all three perspectives, the team was able to make informed recommendations back to the PEO

Program was restructured

Carnegie Mellon
Software Engineering Institute

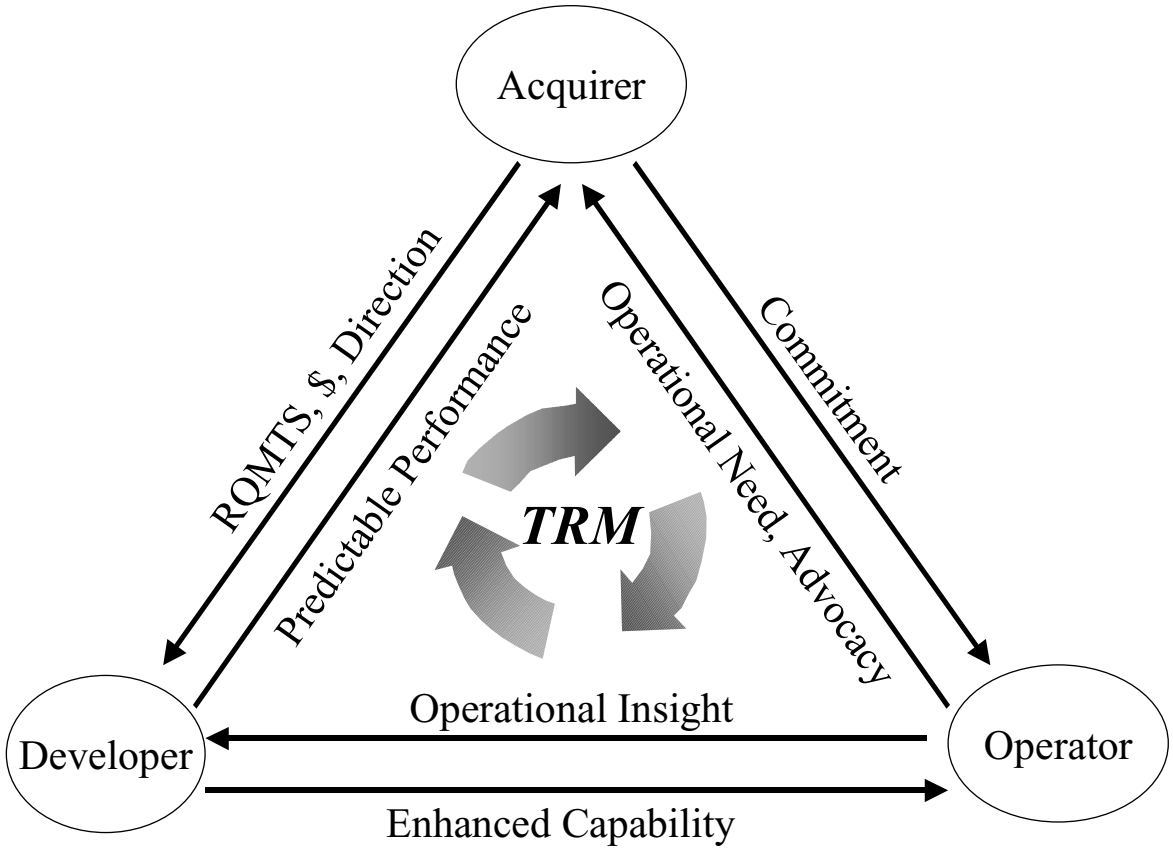
Team Risk Management

Team Risk Management (TRM) builds on healthy and active risk practices within diverse organizations, or organizational entities, teamed together for a common purpose.

TRM works to aid decision making in supplier-acquirer relationships.

Adding the end-user, or operator, TRM is the ideal method of managing risk during new system development.

TRM "Vision"



Carnegie Mellon
Software Engineering Institute

Conclusions

New systems or capabilities delivered to operational forces should mitigate operational risk.

Using a structured Taxonomy to help identify operational risk increases the likelihood of delivering usable systems or capabilities into operational use.

Carnegie Mellon
Software Engineering Institute

Contact Information

Brian Gallagher
Director, Acquisition Support Program
412-268-7157
bg@sei.cmu.edu

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

Carnegie Mellon
Software Engineering Institute

References

A Taxonomy of Operational Risks (CMU/SEI-2005-TN-36). Gallagher, Brian P.; Case, Pamela J.; Creel, Rita C.; Kushner, Susan; Williams, Ray C. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005.
http://www.sei.cmu.edu/publications/documents/05.reports/05tn036.html

Taxonomy-Based Risk Identification (CMU/SEI-93-TR-006, ADA266992). Carr, Marvin J.; Konda, Surish L.; Monarch, Ira; Ulrich, F. Carol; & Walker, Clay F. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.
http://www.sei.cmu.edu/publications/documents/93.reports/93.tr.006.html

Continuous Risk Management Guidebook. Murphy, Richard L.; Alberts, Christopher J.; Williams, Ray C.;Higuera, Ronald P.; Dorofee, Audrey J.; Walker Julie A. Pittsburgh, PA: Carnegie Mellon University, 1996.
http://www.sei.cmu.edu/publications/books/other-books/crm.guidebk.html

Software Risk Evaluation (SRE) Method Description (Version 2.0) (CMU/SEI-99-TR-029, ADA001008). Williams, Ray C.; Pandelios, George J.; & Behrens, Sandra G. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999.
http://www.sei.cmu.edu/publications/documents/99.reports/99tr029/99tr029abstract.html