Risk Analysis for Software Assurance

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Module Topics

Risk Management Overview

Two Approaches for Analyzing Risk

Mission Risk Diagnostic (MRD)

Standard Driver Sets

Risk-Based Measurement and Analysis

Summary

Risk Management Overview

Exercise: *Project Risks*

See handout.

Software Assurance[1]

Application of technologies and processes to achieve a required level of ***confidence*** that software systems and services

- Function in the intended manner
- Are free from accidental or intentional vulnerabilities
- Provide security capabilities appropriate to the threat environment
- Recover from intrusions and failures

We will examine risk management in a software assurance context.

1. SEI Software Assurance Curriculum Project. *Software Assurance Curriculum Project Volume I: Master of Software Assurance Reference Curriculum* (CMU/SEI-2010-TR-005). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/reports/10tr005.pdf

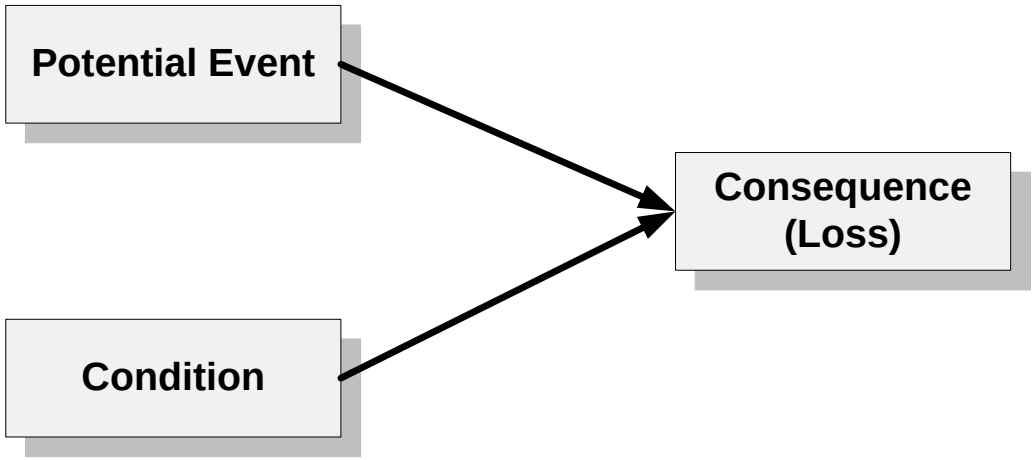
What Is Risk?

The probability of suffering harm or loss

A measure of the likelihood that an event will lead to a loss coupled with the magnitude of the loss

Risk requires the following conditions:[1]

- A potential loss
- Likelihood
- Choice



1. Charette, Robert N. *Application Strategies for Risk Analysis*. New York, NY: McGraw-Hill Book Company, 1990.

Risk Management Activities

Assess risk

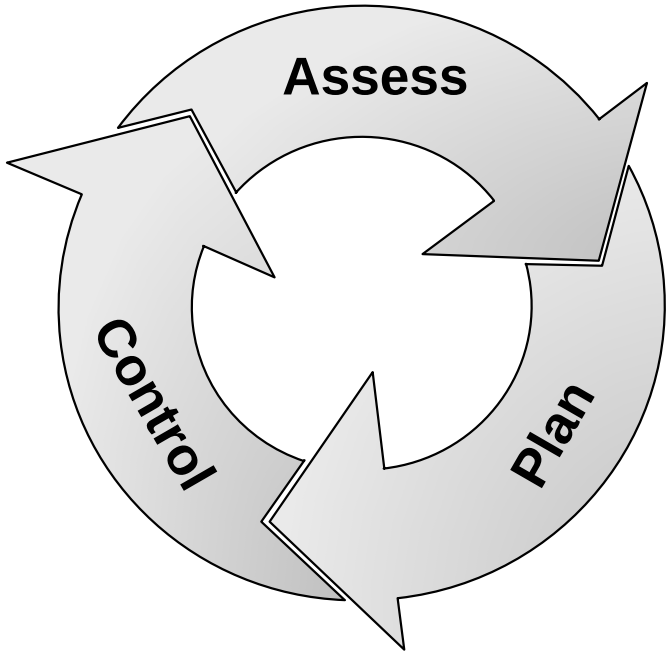
- Transform the concerns people have into distinct, tangible risks that are explicitly documented and analyzed

Plan for risk control

- Determine an approach for addressing each risk; produce a plan for implementing the approach

Control risk

- Deal with each risk by implementing its defined control plan and tracking the plan to completion



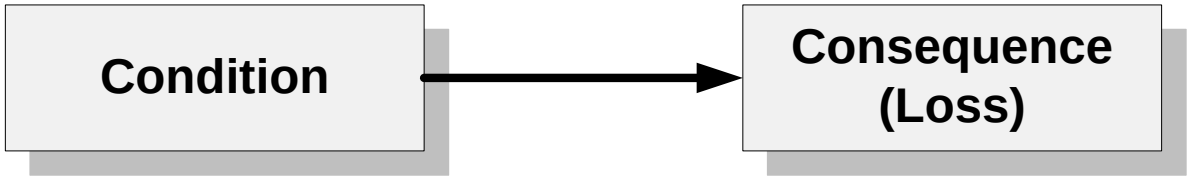
Issue/Problem

A condition that directly produces a loss or adverse consequence.

- No uncertainty exists.
- The condition exists and is having a negative effect on performance.

Issues can also lead to (or contribute to) other risks by

- Creating a circumstance that enables an event to trigger additional loss
- Making an existing event more likely to occur
- Aggravating the consequences of existing risks



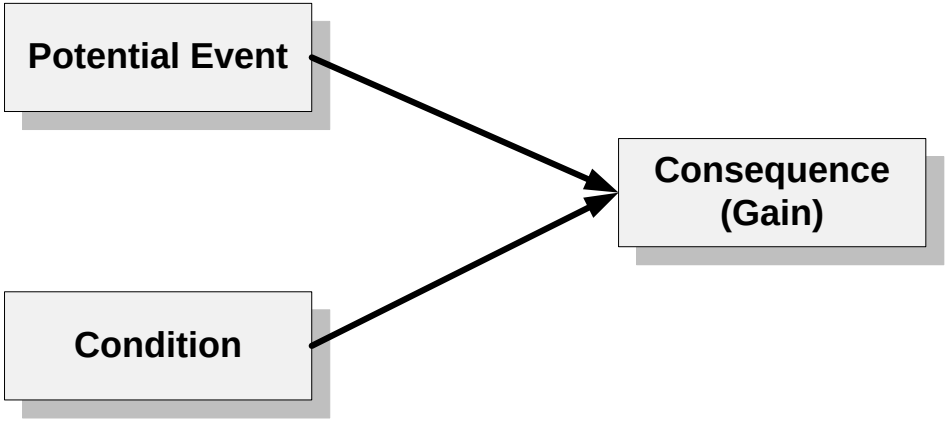
Opportunity

The probability of realizing a gain.

- Defines a set of circumstances that provides the potential for a desired gain
- Enables an entity to improve its current situation relative to the status quo
- Can require an investment or action to realize that gain (i.e., to take advantage of the opportunity)

Pursuit of an opportunity can

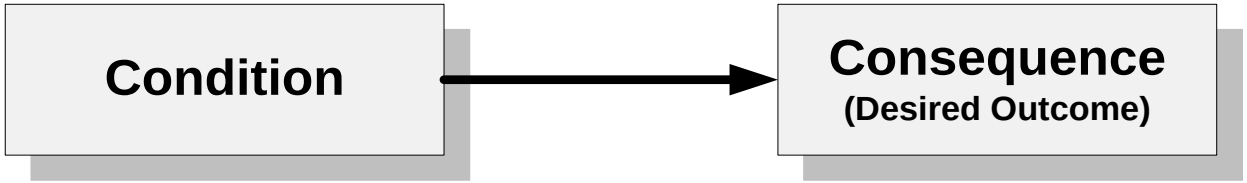
- Produce new risks or issues
- Change existing risks or issues



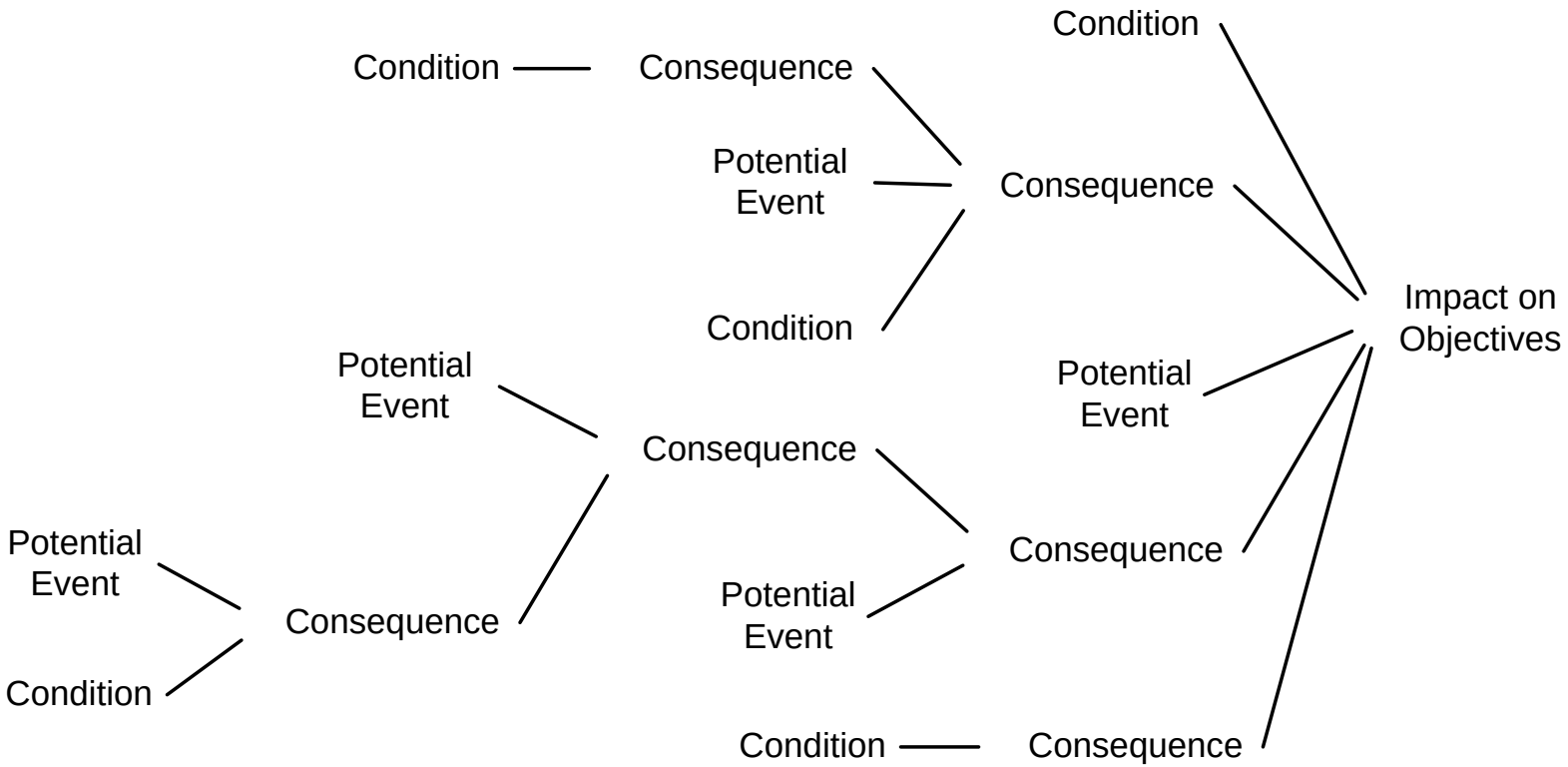
Strength

A condition that is driving an entity (e.g., project, system) toward a desired outcome.

- No uncertainty exists
- The condition exists and is having a positive effect on performance (i.e., driving an entity toward a desired outcome)



Casual Chain of Conditions and Events



Risks, issues/problems**,** opportunities, and strengths are part of an interrelated causal chain of conditions and events that must be managed.

Analyzing Risk in Interactively Complex, Software-Reliant Systems

For software assurance, you must be able to analyze risk in interactively complex, software-reliant systems across the life cycle and supply chain.

- Projects and programs
- Business processes and mission threads
- IT processes

Two Approaches for Analyzing Risk

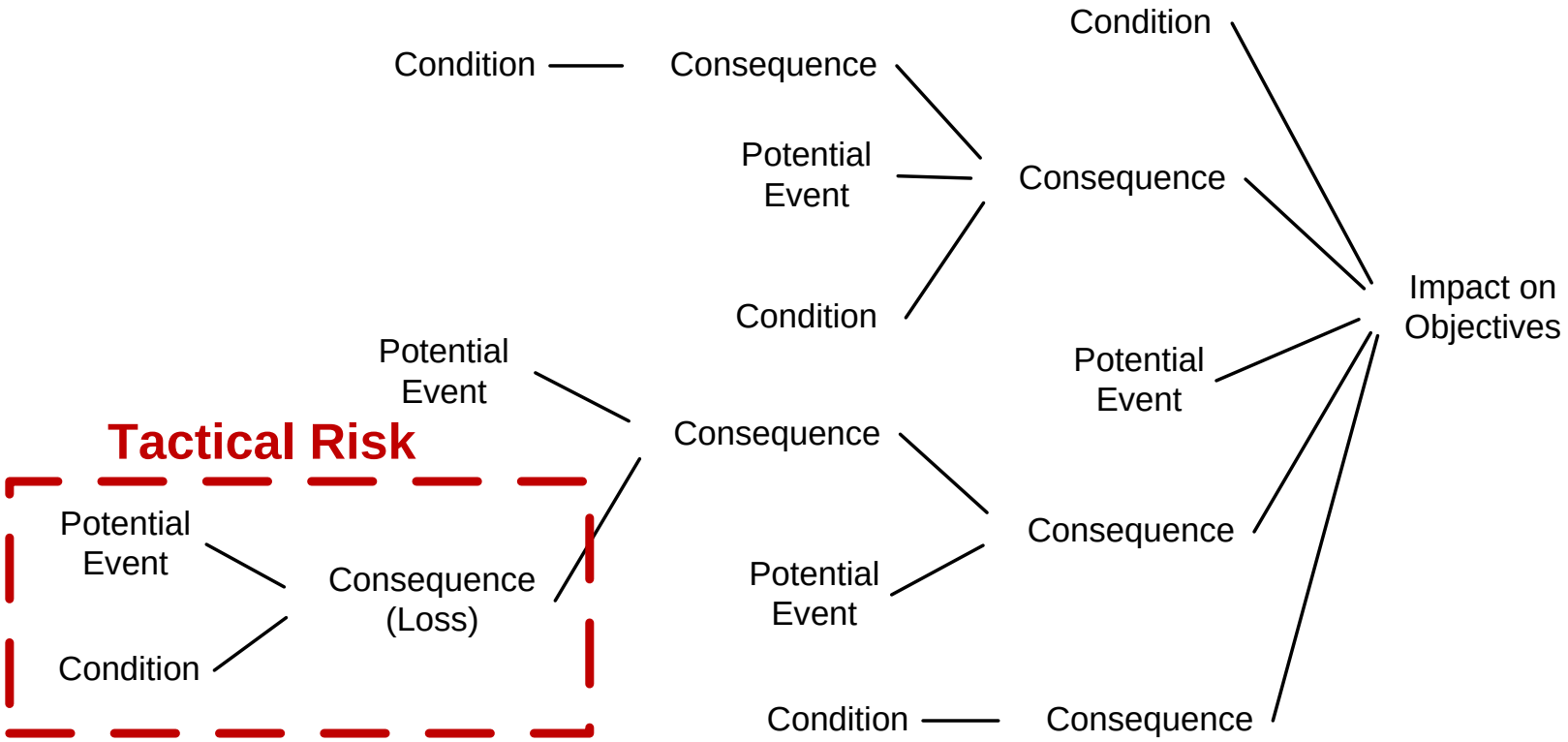
Two Type of Risk Analysis

Two distinct risk analysis approaches can be used when evaluating systems:

1. Tactical risk analysis
2. Mission risk analysis

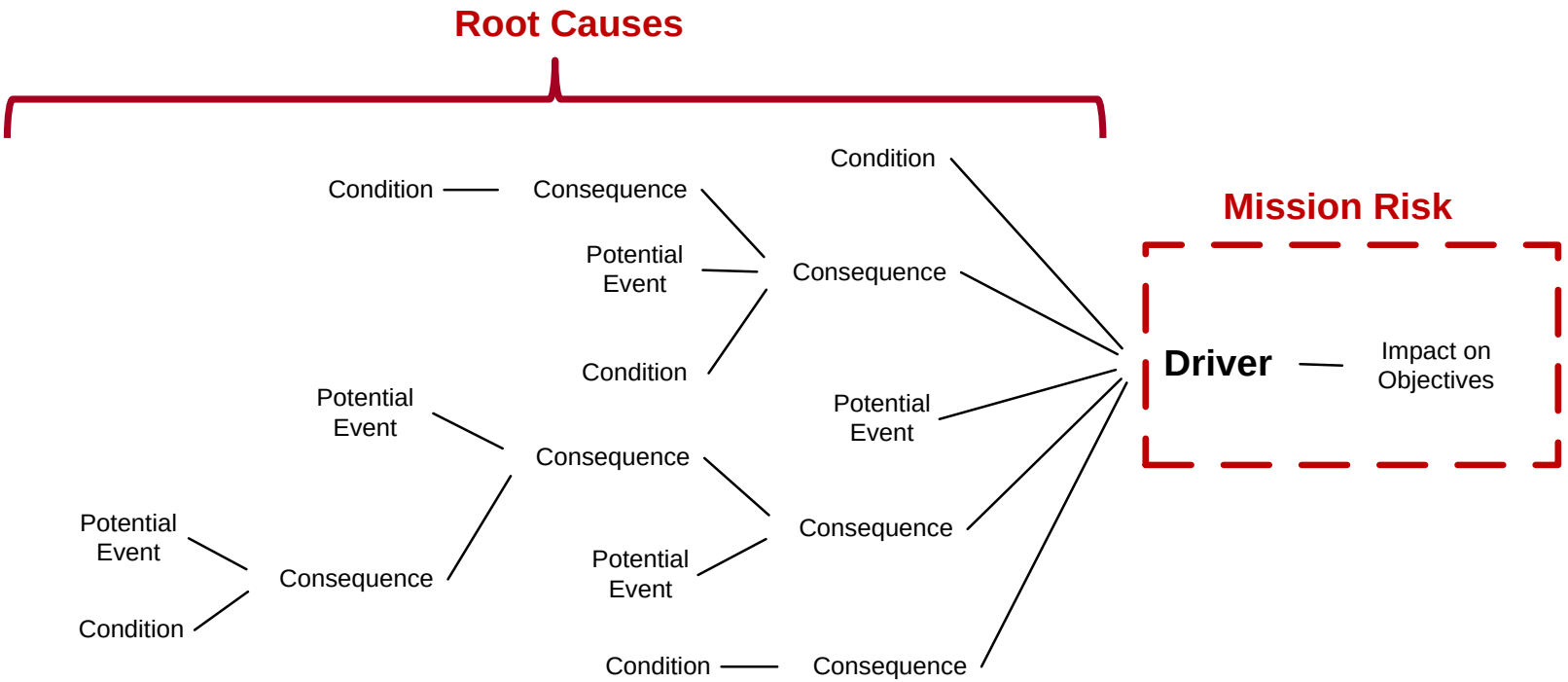
Both types of risk analysis are addressed in this topic.

Elements of Tactical Risk



Tactical risk is the probability that an event will lead to a negative consequence or loss

Elements of Mission Risk



Mission risk is the probability of mission failure (i.e., not achieving key objectives).

Mission risk aggregates the effects of multiple conditions and events on a system's ability to achieve its mission.

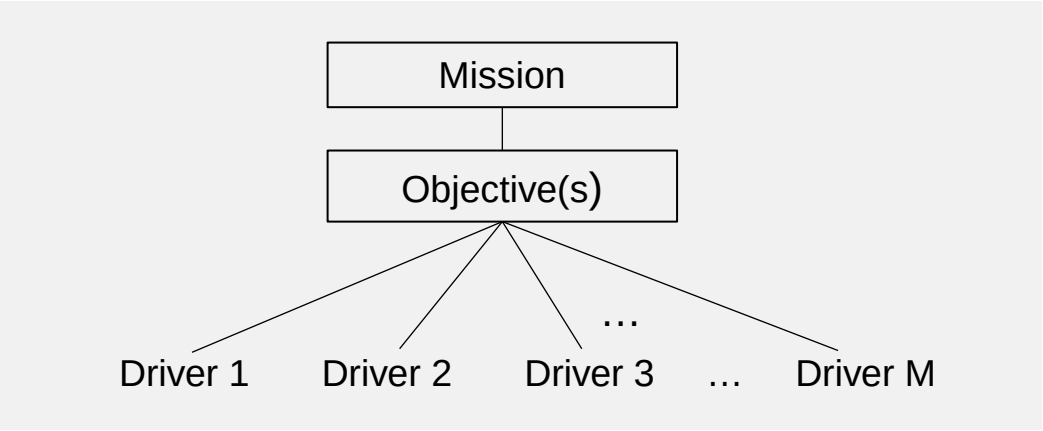
Mission Risk Diagnostic (MRD)

Mission Risk Diagnostic (MRD)

The MRD assesses risk in interactively complex, socio-technical systems, such as

- Projects and programs
- Business processes and mission threads
- IT processes

The goal is to gauge the extent to which a system is in position to achieve its mission and objective(s).



Core MRD Tasks

Identify the mission and objective(s)

Identify drivers

Analyze drivers

Goals of Identifying the Mission and Objective(s)

The overarching goals when identifying the mission and objective(s) are to

- Define the fundamental purpose, or mission, of the system that is being examined
- Establish the specific aspects of the mission that are important to decision makers

Once they have been established, the mission and objective(s) provide the foundation for conducting the assessment.

Mission

The fundamental purpose of the system that is being examined

After the basic target has been established, the next step is to identify which specific aspects of the mission need to be analyzed in detail.

Example

The XYZ Program is providing a new, web-based payroll system for our organization.

Objective

A tangible outcome or result that must be achieved when pursuing a mission

Example

By the end of the development and deployment phase (18 months),

- The web-based payroll system will provide payroll services at all sites across the enterprise
- Development and deployment costs cannot exceed 20 percent of original estimates

SMART Objectives

Objectives identified during the MRD should meet the following criteria:

- ***Specific***—The objective is concrete, detailed, focused, and well defined. It emphasizes action and states a specific outcome to be accomplished.
- ***Measurable***—The objective can be measured, and the measurement source is identified.
- ***Achievable***—The expectation of what will be accomplished is attainable given the time period, resources available, and so on.
- ***Relevant***—The outcome or result embodied in the objective supports the broader mission being pursued.
- ***Time-Bound***—The time frame in which the objective will be achieved is specified.

Drivers -1

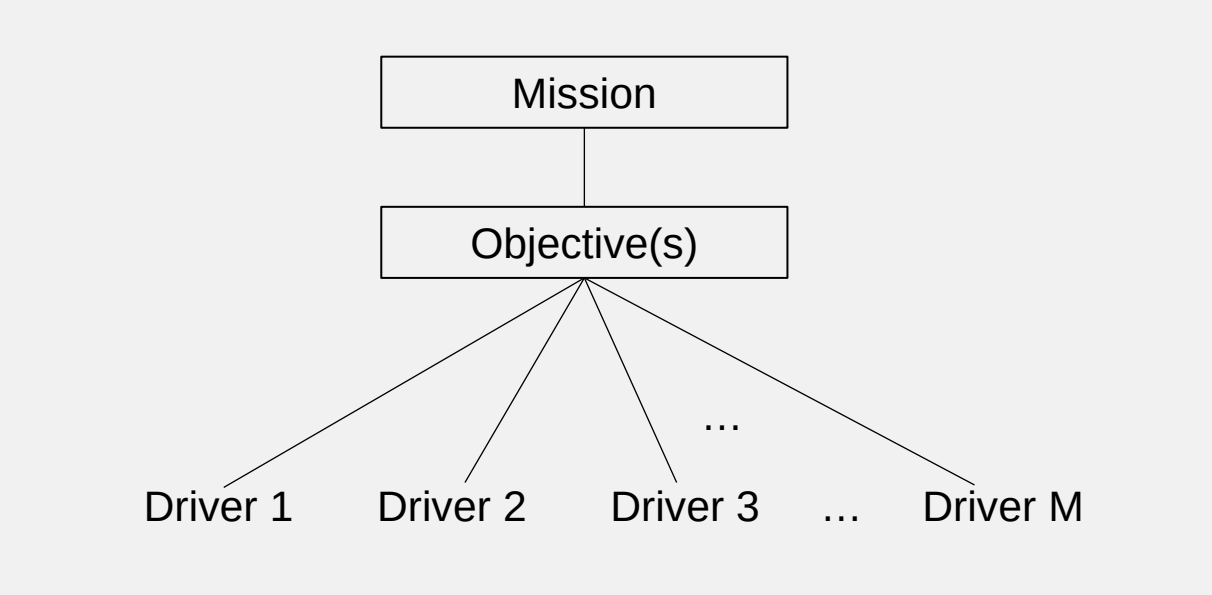
Definition

- A factor that has a strong influence on the eventual outcome or result

Examples

- ***Process***: Is the process being used to develop and deploy the system sufficient?
- ***Task Execution***: Are tasks and activities performed effectively and efficiently?
- ***System Integration***: Will the system sufficiently integrate and interoperate with other systems when deployed?

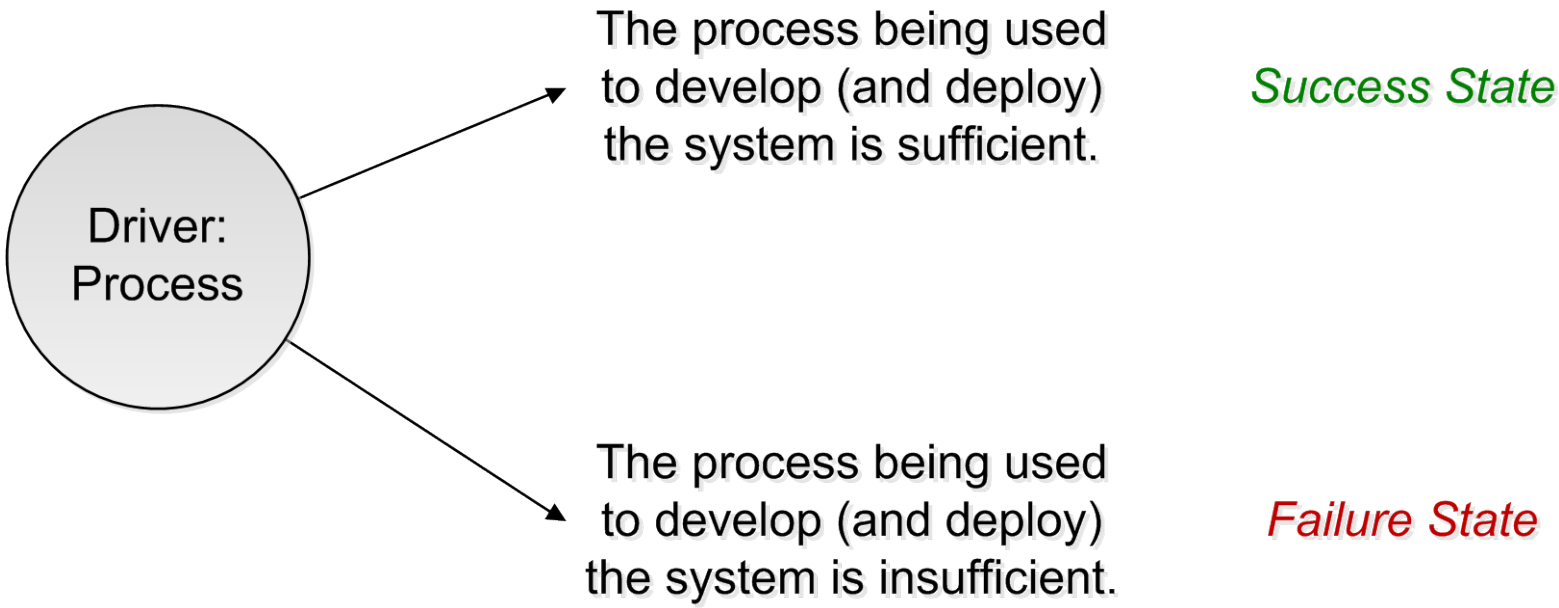
Drivers -2



By definition, a driver has a direct connection to the mission and objectives.

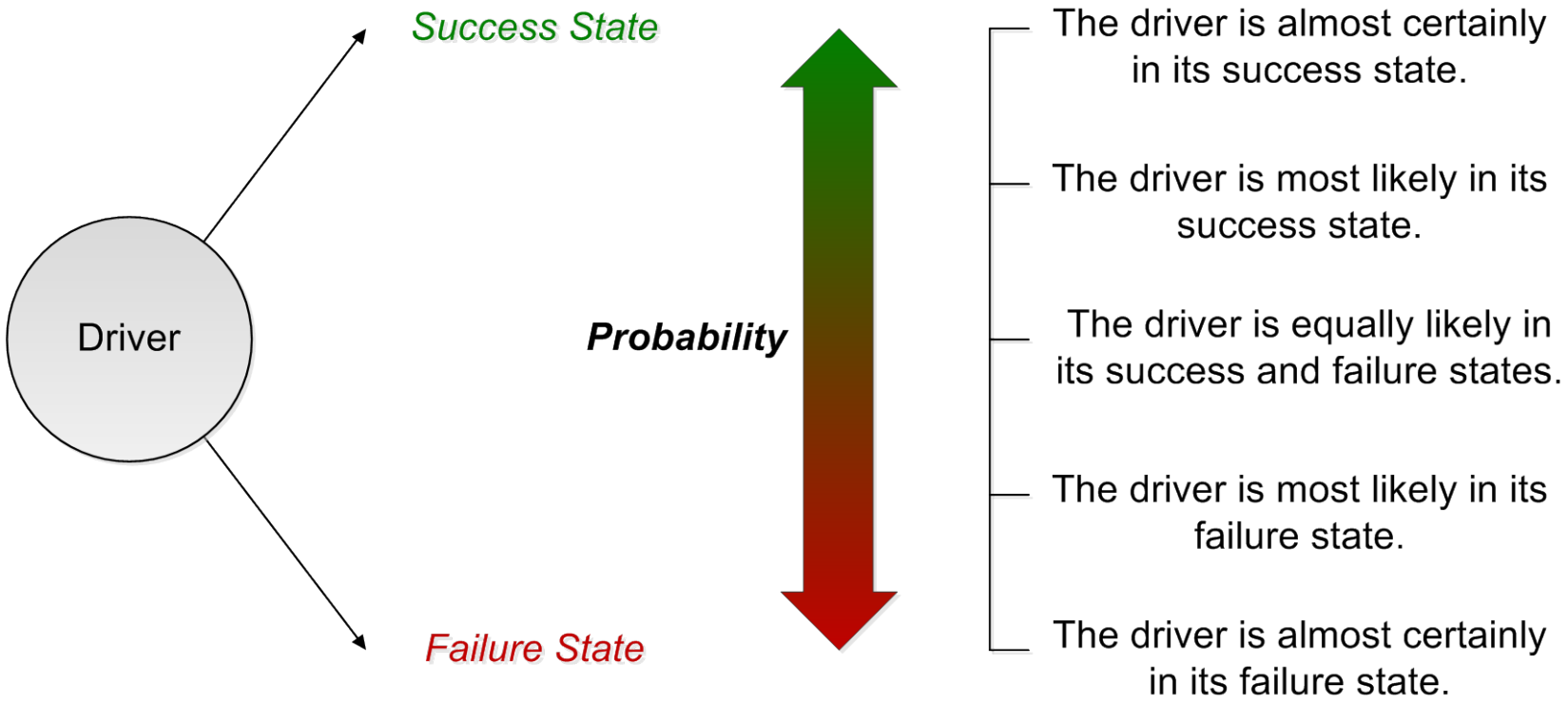
A small set of drivers (typlically10-25) can provide insight into a mission's potential for success.

Drivers: *Success and Failure States -1*



A driver can guide the outcome toward key objectives (success state) or away from them (failure state).

Drivers: *Success and Failure States -2*



The objective when analyzing a driver's state is to determine how each driver is currently acting.

Evaluating Drivers

Directions: Select the appropriate response to the driver question.

Driver Question	Response
3. Is the process being used to develop and deploy the system sufficient? *Consider:* Process design; measurements and controls; process efficiency and effectiveness; acquisition and development life cycles; training	❑ Yes ❑ Likely Yes ❑ Equally Likely ☒ Likely No ❑ No ❑ Not Applicable

Evaluating Drivers: *Items to Consider*

The following items should be examined for each driver that is analyzed:

- Positive conditions that support a response of *yes* (strengths)
- Negative conditions that support a response of *no* (issues/problems)
- Potential events with positive consequences that support a response of *yes* (tactical opportunities)
- Potential events with negative consequences that support a response of *no* (tactical risks)
- Unknown factors that contribute to uncertainty regarding the response (uncertainties)
- Assumptions that might bias the response (assumptions)

Rationale and Supporting Evidence

The rationale and supporting evidence for each response to a driver question is recorded.

Evidence can include: interview data, documentation, reports, observations, demonstrations, and measurement data

Example Rationale

+ Previous programs have a 90% history of delivering on-time.
- The process for integration testing is not documented.
- There are a lot of brand new programmers (45%).
- This program required a significant change in our standard processes. There was no new training created for the new processes.
- QA did not have a chance to review the new and revised processes before they were put into practice.

Drivers for Software/System Development

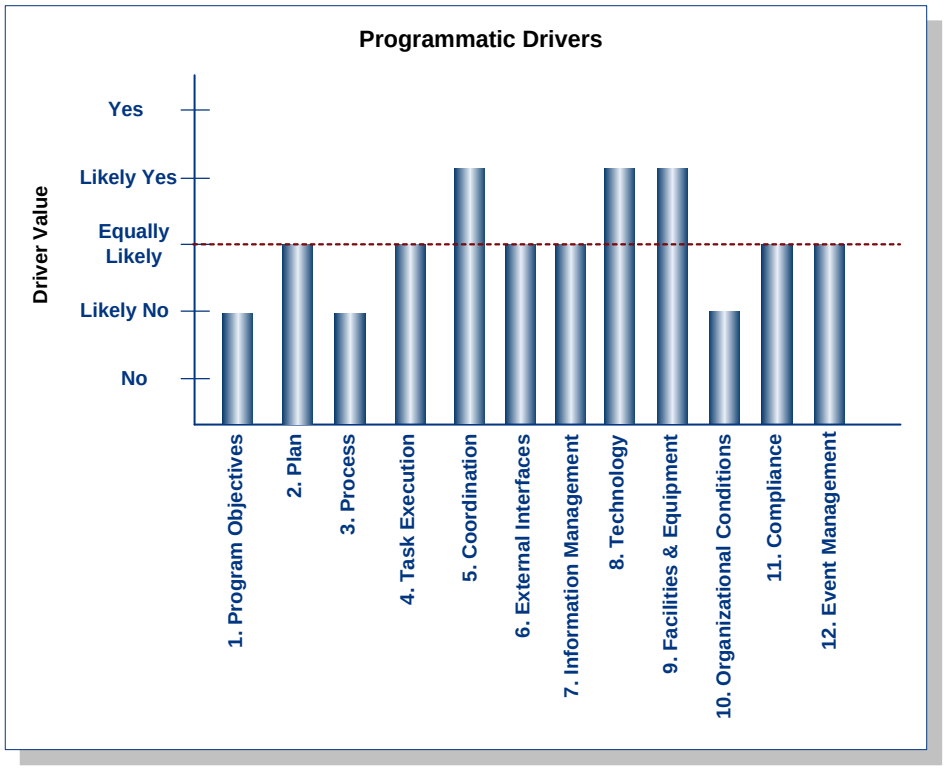
Programmatic Drivers

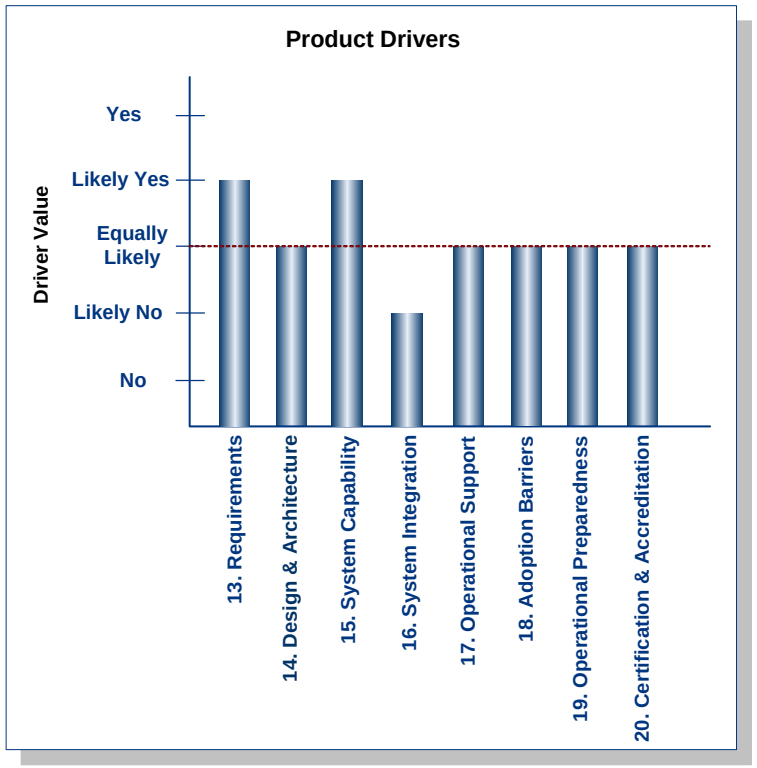
1. Program Objectives
2. Plan
3. Process
4. Task Execution
5. Coordination
6. External Interfaces
7. Information Management
8. Technology
9. Facilities and Equipment
10. Organizational Conditions
11. Compliance
12. Event Management

Product Drivers

13. Requirements
14. Architecture and Design
15. System Capability
16. System Integration
17. Operational Support
18. Adoption Barriers
19. Operational Preparedness
20. Certification and Accreditation

Driver Profile

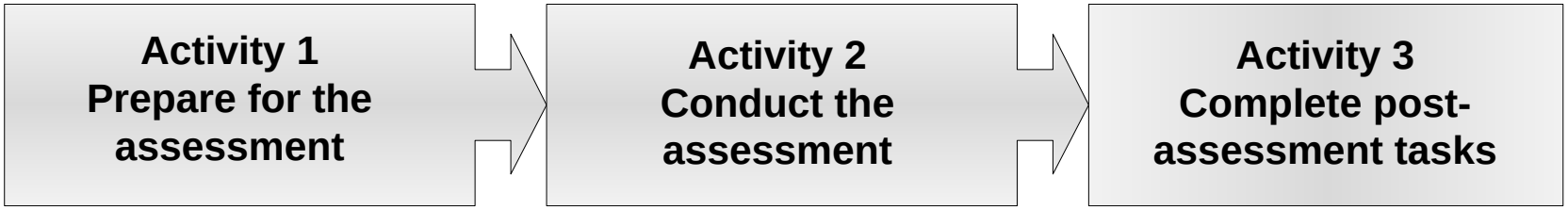




The driver profile provides an indication of systemic risk to the mission (i.e., mission risk).

It can be used as a dashboard for program decision makers.

MRD Method: *Activities and Tasks*



Tasks

- 1.1 Form the assessment team
- 1.2 Develop stakeholder sponsorship
- 1.3 Set the scope of the assessment
- 1.4 Develop the assessment plan
- 1.5 Coordinate logistics
- 1.6 Tailor method and tools

Tasks

- 2.1 Identify mission and objective(s)
- 2.2 Identify drivers
- 2.3 Analyze drivers
- 2.4 Determine next steps

Tasks

- 3.1 Communicate results
- 3.2 Conduct assessment postmortem
- 3.3 Improve assessment process

Standard Driver Sets

Drivers for Software/System Development -1

Mission

The [program/project] is developing and deploying the [software-reliant system].

Objective

By the end of the development and deployment phase (N months),

- The system will provide agreed-upon services to users
- Development and deployment costs cannot exceed X percent of original estimates

Drivers for Software/System Development -2

Programmatic Drivers

1. Program Objectives
2. Plan
3. Process
4. Task Execution
5. Coordination
6. External Interfaces
7. Information Management
8. Technology
9. Facilities and Equipment
10. Organizational Conditions
11. Compliance
12. Event Management

Product Drivers

13. Requirements
14. Architecture and Design
15. System Capability
16. System Integration
17. Operational Support
18. Adoption Barriers
19. Operational Preparedness
20. Certification and Accreditation

See handout.

Drivers for Secure Software/System Development -2

Mission

The [program/project] is developing and deploying the [software-reliant system].

Objective

When the system is deployed, security risks to the deployed system will be within an acceptable tolerance.

Drivers for Secure Software/System Development -2

Programmatic Drivers

1. Program Security Objectives
2. Security Plan
3. Contracts
4. Security Process
5. Security Task Execution
6. Security Coordination
7. External Interfaces
8. Organizational and External Conditions
9. Event Management

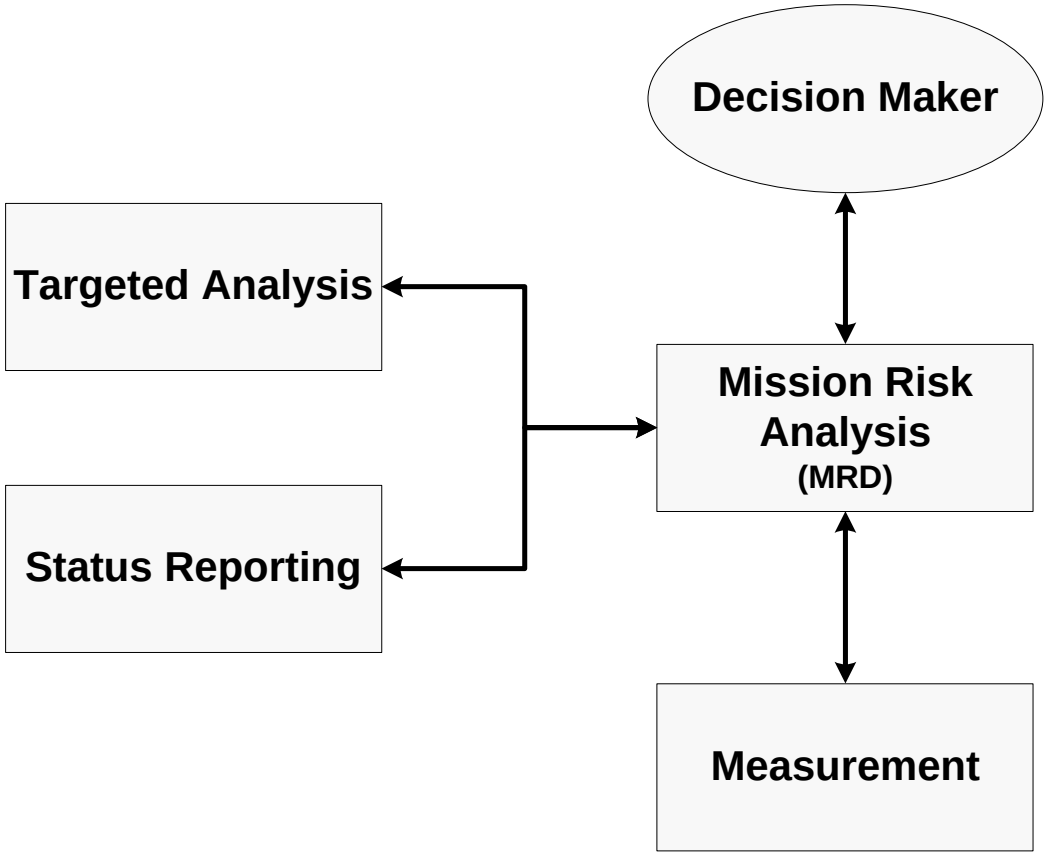
Product Drivers

10. Security Requirements
11. Security Architecture and Design
12. Code Security
13. Integrated System Security
14. Adoption Barriers
15. Operational Security Compliance
16. Operational Security Preparedness
17. Product Security Risk Management

See handout.

Risk-Based Measurement and Analysis

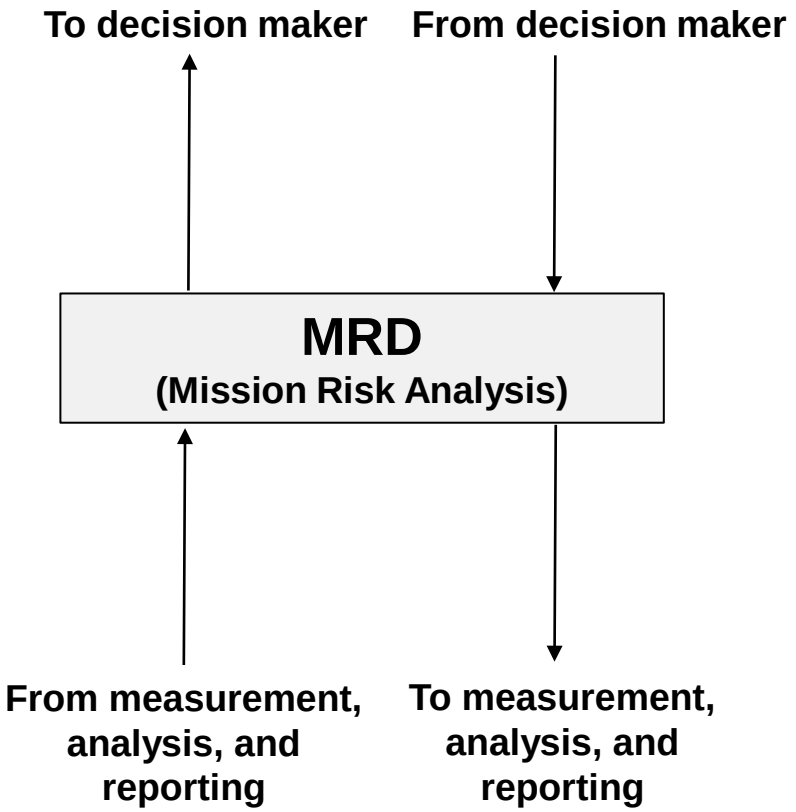
Integrated Measurement and Analysis Framework (IMAF)



The IMAF employs mission risk analysis to provide decision makers with a consolidated view of the performance of interactively complex software-reliant systems.

Using the IMAF

2. The MRD identifies mission risks and uncertainties

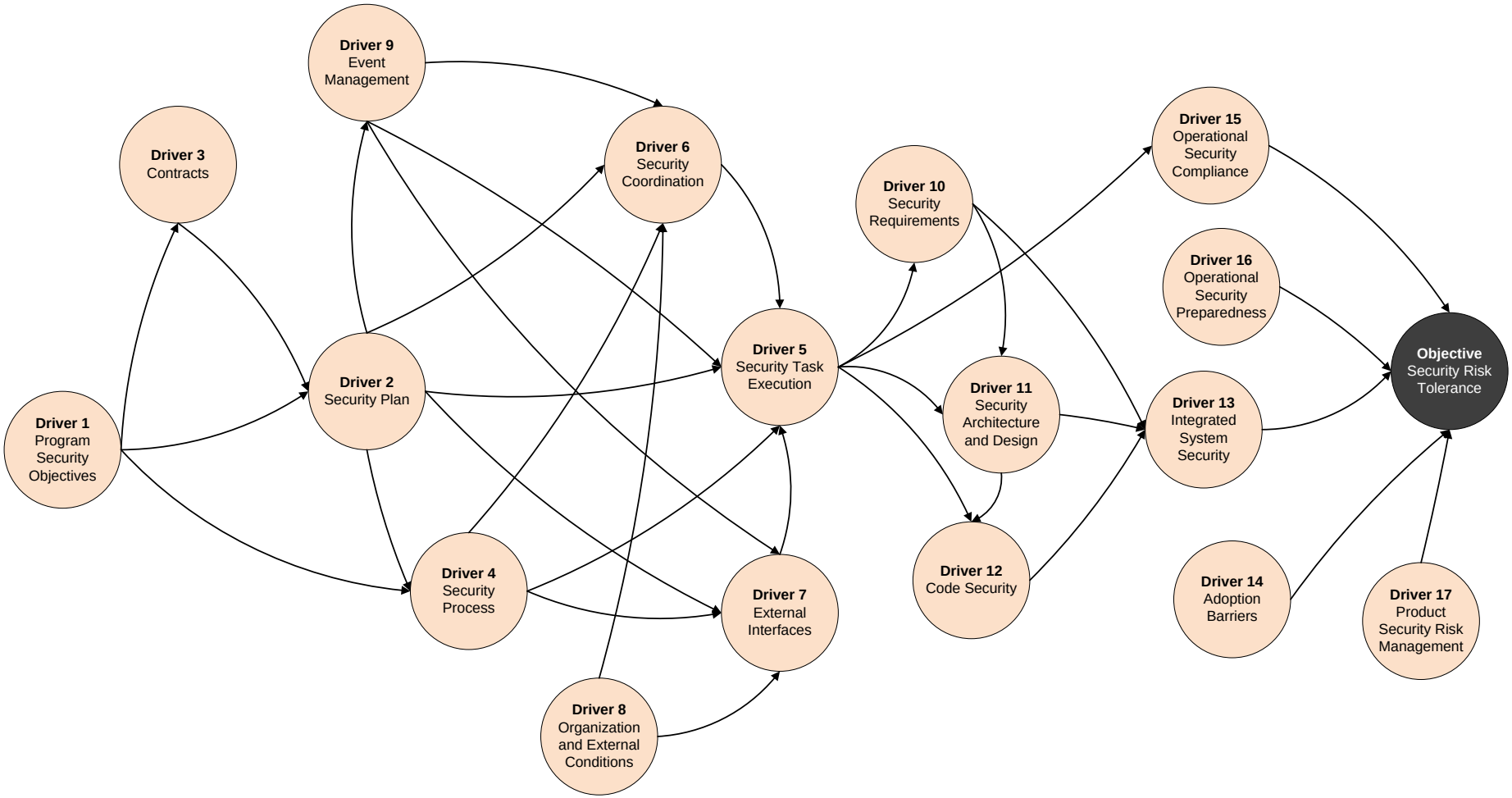


3. Decision maker revises information needs

1. Information is collected (ongoing activity)

4. Information needs are revised

Research Topic: *Quantitative Driver Modeling*



Summary

Key Points -1

The basic goal of risk analysis is to provide decision makers

- With the information they need
- When they need it
- In the right form

If decisions are not influenced by risk analysis activities, then risk analysis provides no added value.

Applying mission risk analysis (e.g., by using the MRD) enables decision makers to confidently assess the behavior of interactively complex systems.

Key Points -2

The IMAF can be used to direct measurement activities based on the degree of risk and uncertainty affecting a system.

The reduction in uncertainty resulting from new data will

- Provide decision makers with more clarity regarding system performance
- Enable better decision making based on more objective data

Case Study -1

Form teams of 4-5 people.

Each team should have 1 or more students working on a software development project that can be used as a software security case study.

The team members should have reasonably compatible schedules in order to accomplish the team work.

Case Study -2

1. Document the mission and objective(s) for the software-development project that you are assessing. See additional guidance in the *Mission Risk Diagnostic (MRD) Workbook*. (15%)
2. Answer all driver questions. Document your answer to each driver question as well as the rationale for your response. See additional guidance in the *Mission Risk Diagnostic (MRD) Workbook*. (50%)
3. Document the top 3 next-step recommendations for the project based on your responses to the driver questions. (25%)
4. Describe what insights you gained (if any) by applying the method. (10%)

Publications and Resources -1

Cyber Security Engineering (CSE) Team Web Page
http://www.cert.org/sse/

Alberts, Christopher & Dorofee, Audrey. *Mission Risk Diagnostic (MRD) Method Description* (CMU/SEI-2012-TN-005). Software Engineering Institute, Carnegie Mellon University, 2012.
http://www.sei.cmu.edu/reports/12tn005.pdf

Alberts, Christopher; Allen, Julia; & Stoddard, Robert. *Risk-Based Measurement and Analysis: Application to Software Security* (CMU/SEI-2012-TN-004), Software Engineering Institute, Carnegie Mellon University, 2012.
http://www.sei.cmu.edu/reports/12tn004.pdf

Publications and Resources -2

Alberts, Christopher & Dorofee, Audrey. *A Framework for Categorizing Key Drivers of Risk* (CMU/SEI-2009-TR-007). Software Engineering Institute, Carnegie Mellon University, 2009.
http://www.sei.cmu.edu/library/abstracts/reports/09tr007.cfm

SEI Mission Success in Complex Environments (CSE) Special Project
http://www.sei.cmu.edu/risk/

Alberts, Christopher J.; Dorofee, Audrey J.; Creel, Rita; Ellison, Robert J.; Woody, Carol. "A Systemic Approach for Assessing Software Supply-Chain Risk." *Proceedings of the 44th Hawaii International Conference on System Sciences*. 2011.
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05718996\

For Additional Information

Christopher Alberts

Principal Engineer

CERT® Program, Software Engineering Institute

Email	cja@sei.cmu.edu
Phone	412-268-3045
Fax	412-268-5758
WWW	http://www.cert.org/sse/
U.S. mail	Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213-3890

Software Engineering Institute | Carnegie Mellon