New Directions in Risk: *A Success-Oriented Approach*

SEPG 2009

Audrey Dorofee
Christopher Alberts

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

© 2009 Carnegie Mellon University

Mission Success in Complex Environments (MSCE)

Part of the SEI Acquisition Support Program (ASP), the MSCE Project develops methods, tools, and techniques for

- Advancing the state-of-the-practice for risk management
- Assuring success in complex, uncertain environments

The project builds on more than 16 years of SEI research and development in risk management.

- Continuous Risk Management for software-development projects
- Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE®) for organizational security

© 2009 Carnegie Mellon University

Topic Areas

Current Problem Space

Risk Management: *A Review*

A Different Perspective: *Key Concepts*

The Mission Diagnostic

Beyond the Basic Mission Diagnostic

Summary

© 2009 Carnegie Mellon University

Learning Objectives

Understand the limitations of traditional risk management approaches for today's complex, multi-organizational, system-of-system programs.

Understand how current program conditions can be used to estimate the program's current momentum towards success.

Learn how to use the Mission Diagnostic to evaluate a program's key drivers of success and failure and determine it's current potential for success.

© 2009 Carnegie Mellon University

CURRENT PROBLEM SPACE

© 2009 Carnegie Mellon University

Widespread Use of Risk Management Approaches

Most programs or organizations implement some type of risk management approach.

- Risk management plan
- Processes
- Tools

However, preventable failures continue to occur.

- Uneven and inconsistent application of risk-management practice
- Significant gaps in risk-management practice
- Ineffective integration of risk-management practice

© 2009 Carnegie Mellon University

Increasing Complexity

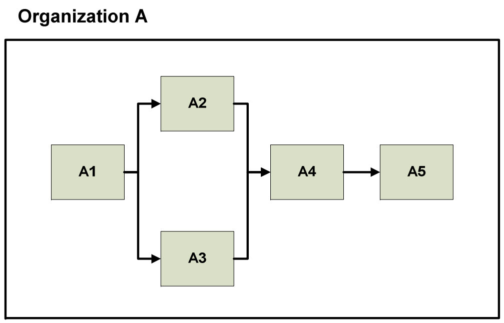
Managers are overseeing increasingly complex projects, programs, and operational processes.

- Multiple models, frameworks, and standards
- Multiple points of management control
- Complex, distributed support technologies
- A variety of management techniques (project, security, financial, technology, etc.)
- Complex tasks
- Multiple detailed status reports

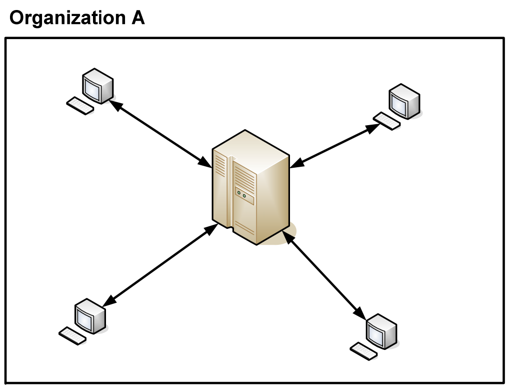
© 2009 Carnegie Mellon University

Single Point of Management Control

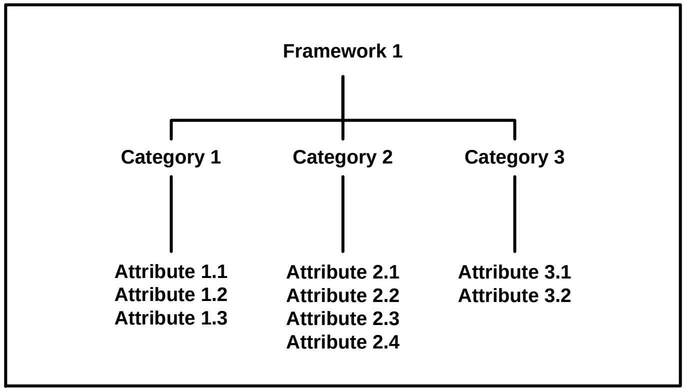
Many management approaches are based on a single point of management control.



Single Program



Single IT System

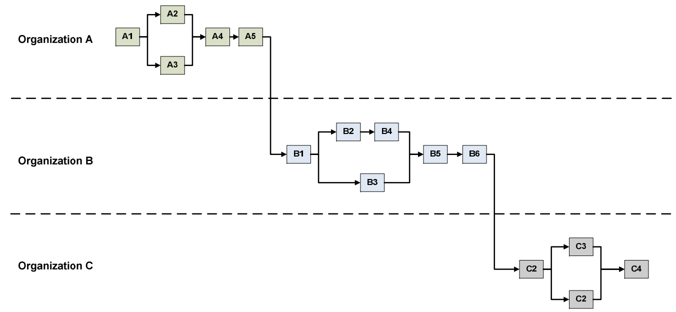


Single Management Framework

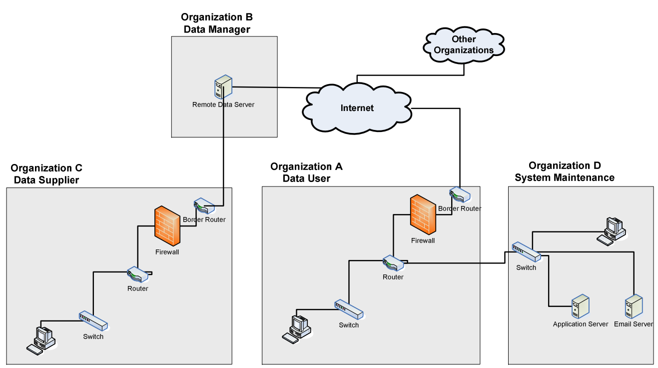
© 2009 Carnegie Mellon University

Distributed Management Environments

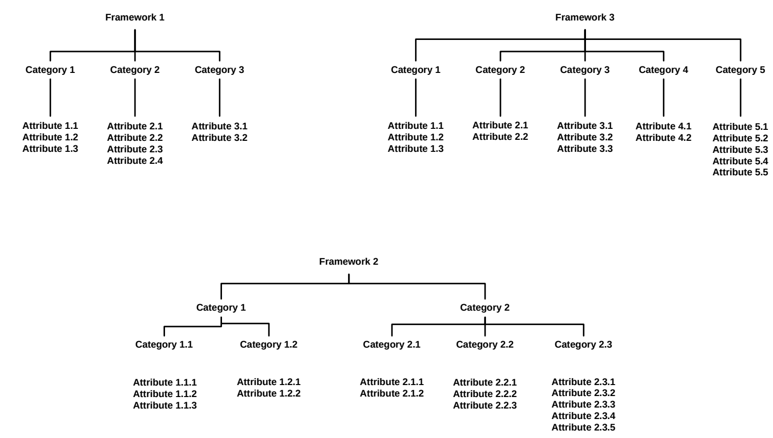
A distributed management environment comprises multiple, points of management control.



Multiple Organizations



Multiple Systems



Multiple Frameworks

© 2009 Carnegie Mellon University

Changing Management Environment

Old Environment	New Environment
• Centralized knowledge, experience, and expertise • Single point of management control (few decision makers) • Command and control	• Distributed knowledge, experience, and expertise • Multiple points of management control (many decision makers) • Communication and coordination

© 2009 Carnegie Mellon University

Management Challenges

Implementing an integrated management approach

- Synthesizing the results of point solutions
- Focusing on operational success across the life cycle and supply chain
- Strategically allocating resources based on greatest need
- Striking the proper balance between risk and opportunity

Coordinating management efforts in distributed environments

Balancing strategic objectives (mission) and tactical objectives (local)

© 2009 Carnegie Mellon University

Issues with Current Risk Management Solutions

Prevalence of point solutions

- Specific point in the life-cycle
- Narrow range of threats

Inability to scale point solutions to distributed management environments

Lack of risk-management solutions specifically designed for distributed management environments (e.g., system-of-system environments)

© 2009 Carnegie Mellon University

Risk Management
A REVIEW

© 2009 Carnegie Mellon University

Exercise One

Refer to Tutorial Workbook, Exercise #1

1. Read the Scenario

2. Consider:
    - What led to the program's failure?
    - Who should have been responsible for resolving these issues and preventing this failure?

© 2009 Carnegie Mellon University

What Is Risk?

Risk is the likelihood of loss.

Risk requires the following conditions:

- A potential loss
- Likelihood
- Choice

© 2009 Carnegie Mellon University

Risk Perspectives

Speculative Perspective

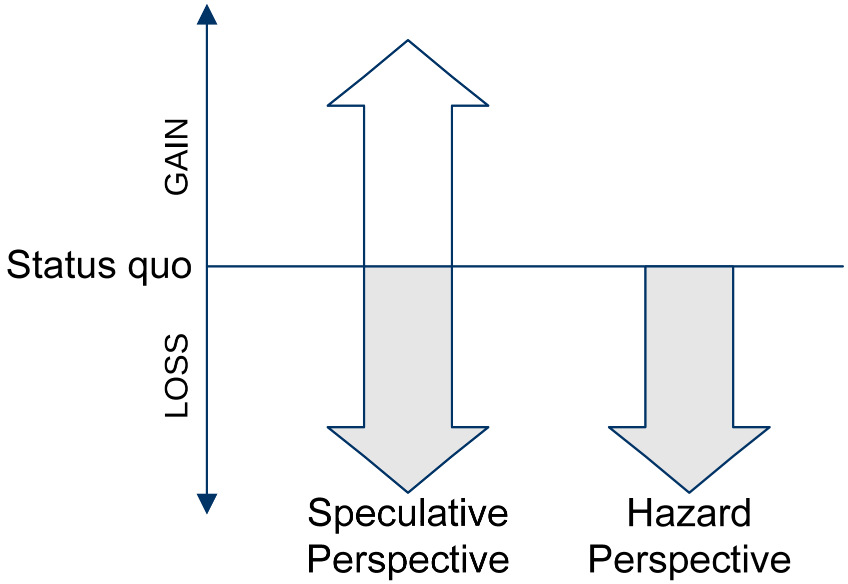
Provides the potential for gain as well as the potential for loss

Brings the potential to improve the current situation relative to the status quo

Hazard Perspective

Provides no opportunity to improve upon the current situation

Brings only the potential for loss



© 2009 Carnegie Mellon University

Risk in Software-Intensive Systems

When developing or operating software-intensive systems, risk is traditionally viewed:

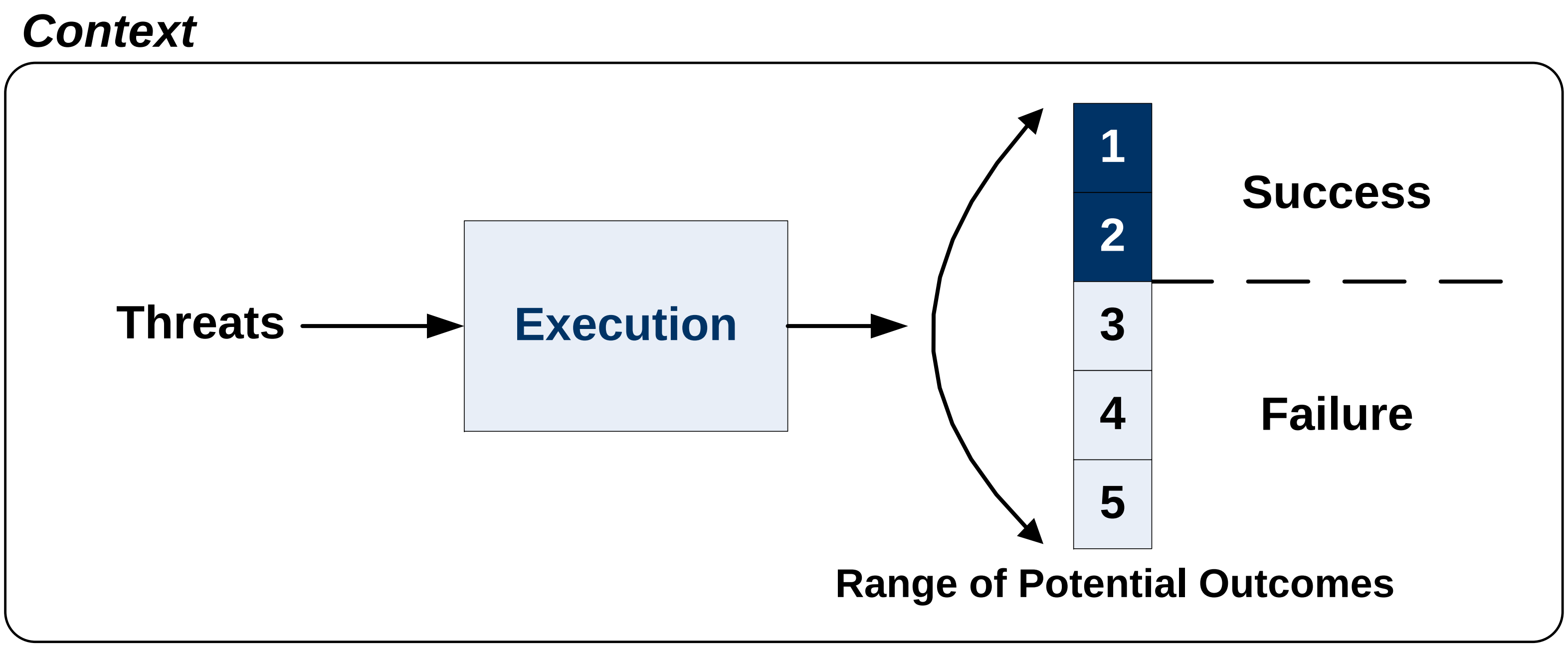
- From a hazard perspective
- As a potential obstacle that can interfere with progress

© 2009 Carnegie Mellon University

The Traditional View of Risk: *Focus on Threats*

A threat is a circumstance with the potential to cause harm or loss.

- Conditions with negative consequences
- Events with negative consequences

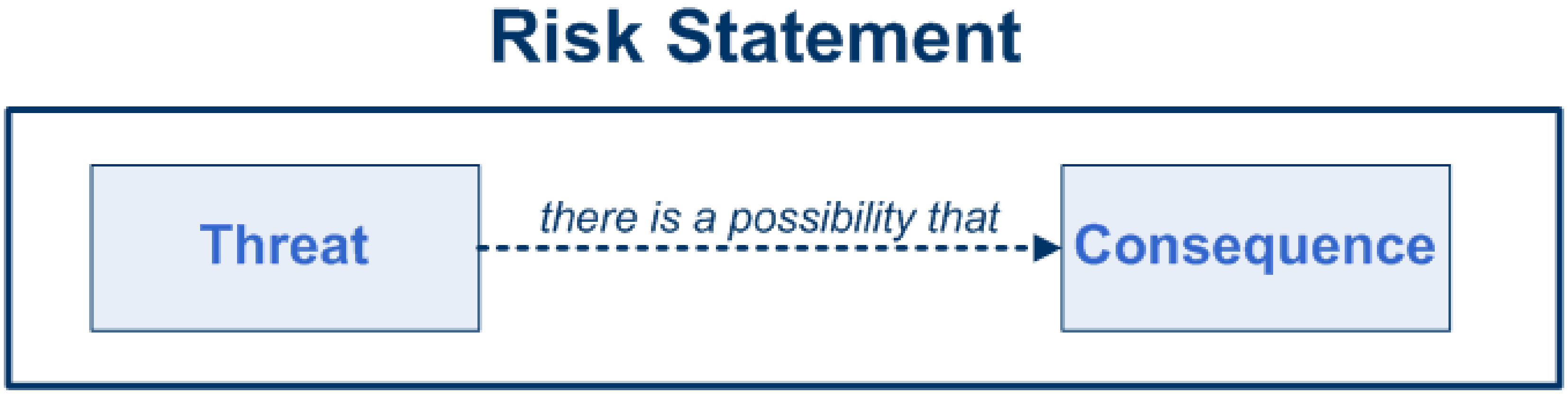


© 2009 Carnegie Mellon University

Risk Statement

Threat
A phrase or sentence that briefly describes the circumstances and situations that are causing concern, doubt, anxiety, or uncertainty

Consequence
A phrase or sentence that describes the negative outcome(s) resulting from the current conditions



© 2009 Carnegie Mellon University

Example: *Risk Statements*

Threat	Consequence
Staffing levels are insufficient	The program could fail to achieve its product, cost, and schedule objectives
If our subcontractor is late in getting its modules completed on time	Then the program's schedule will likely slip

The first risk statement is a condition-consequence statement, which is effective for articulating risks triggered by current conditions.

The second risk statement is an if-then statement, which is effective for articulating risks triggered by the occurrence of potential events.

© 2009 Carnegie Mellon University

Types of Risk

Contracting
- Contracting/ acquisition risk

Development
- Program risk
- Product risk
- Design risk
- Architecture risk
- COTS risk

Transition to Operations
- Integration risk
- Performance risk
- User acceptance risk

Operations and Maintenance
- Enterprise risk
- Business-process risk
- Operational risk
- IT risk
- Information assurance risk
- Security risk

Many different types of risk are managed across the life cycle.

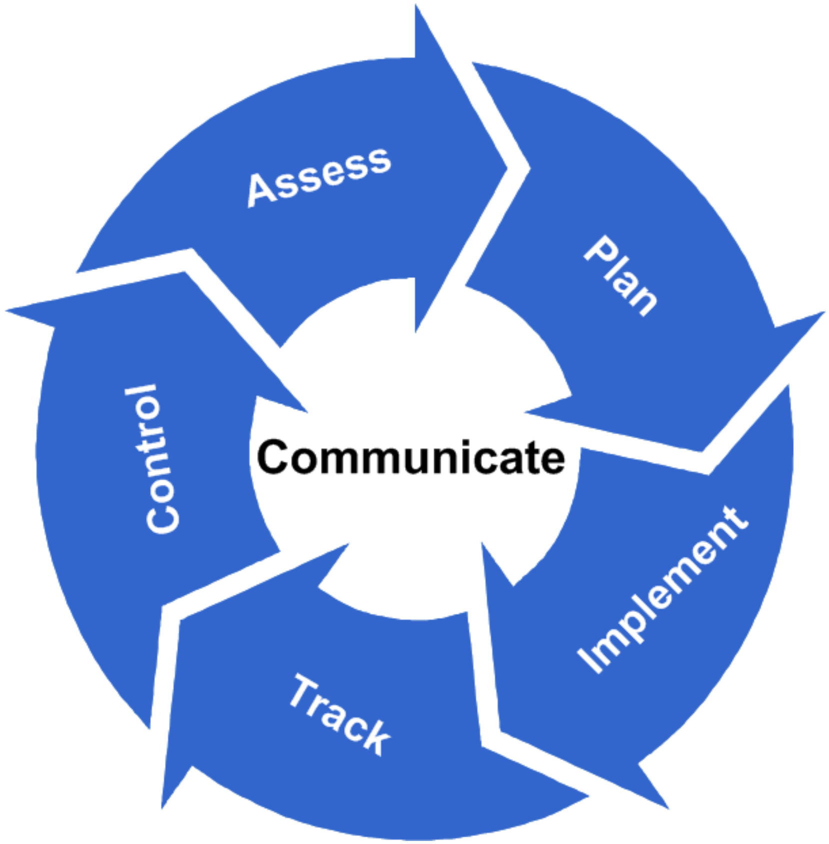
Threat-driven approaches are most commonly used to manage these risks.

© 2009 Carnegie Mellon University

What Is Risk Management?

In a systems context, risk management is traditionally viewed as a proactive, disciplined approach for

- Assessing what can go wrong—risks caused by a range of threats
- Determining which risks are important to address
- Implementing actions to deal with the highest priority risks



© 2009 Carnegie Mellon University

A Different Perspective

KEY CONCEPTS

© 2009 Carnegie Mellon University

Multiple Risk Management Approaches

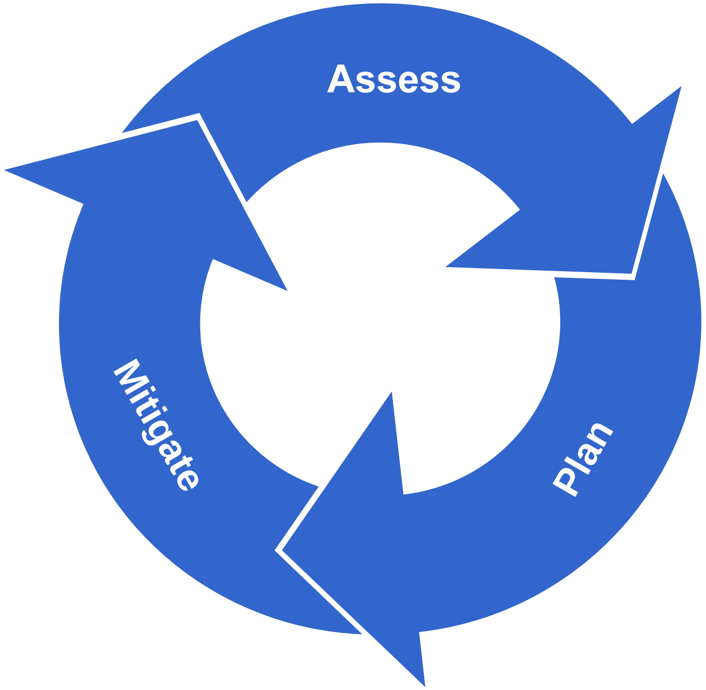
Focus of this tutorial

© 2009 Carnegie Mellon University

Objective-Driven Risk Management - 2

A proactive, disciplined approach for

- Identifying the key objectives that must be achieved
- Establishing what factors, or drivers, can influence the outcome
- Determining which drivers are putting key objectives at risk
- Assessing the probability and severity of each risk
- Developing mitigation approaches for each risk
- Implementing and tracking implementation plans



© 2009 Carnegie Mellon University

Two Types of Risk Management

Threat-Driven Risk Management (Risk to Execution)	Objective-Driven Risk Management (Risk to Objectives)
Key Questions 1. What threats (i.e., potential hazards) can interfere with execution? 2. How likely is each threat? 3. What is the severity of impact for each threat?	***Key Questions*** 1. What key objectives must be achieved? 2. What factors, or drivers, can influence the outcome? 3. Which drivers are putting the key objectives at risk? 4. What is the probability and severity of each risk?
Tactical focus **Bottom-up analysis**	**Systemic focus** **Top-down analysis**

© 2009 Carnegie Mellon University

Managing Risk in Distributed Environments

Objective-driven risk management is a structured approach for assessing and managing in distributed environments.

- Multiple teams, groups, or organizations working toward common objectives (*current focus*)
- Systems of systems and networked systems
- Management of multiple models, standards, & frameworks
- Others

© 2009 Carnegie Mellon University

Mission-Oriented Success Analysis and Improvement Criteria (MOSAIC)

What

A suite of methods that enable objective-driven risk management

Benefits

Enables continuous management of risk to objectives

Applicable across all life-cycle phases

Designed for distributed management environments

Provides a means of analyzing risk in relation to management models, frameworks, and standards

© 2009 Carnegie Mellon University

MOSAIC: *Overarching Goal*

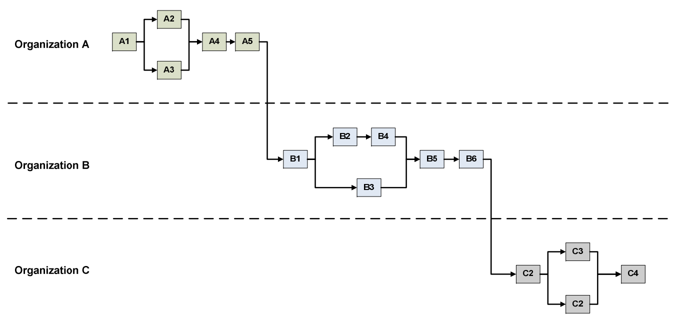
To establish and maintain *confidence* that a software-intensive system or system of systems will achieve its key operational objectives

- Support of operations
- Functionality
- Performance
- Reliability
- Interoperability
- Information assurance
- Usability and maintainability
- Others

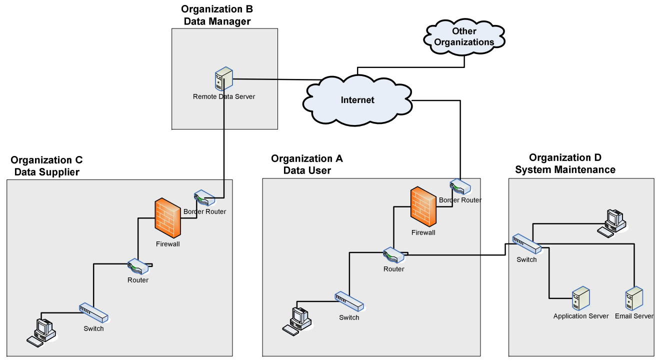
© 2009 Carnegie Mellon University

Risk as an Integrating Theme

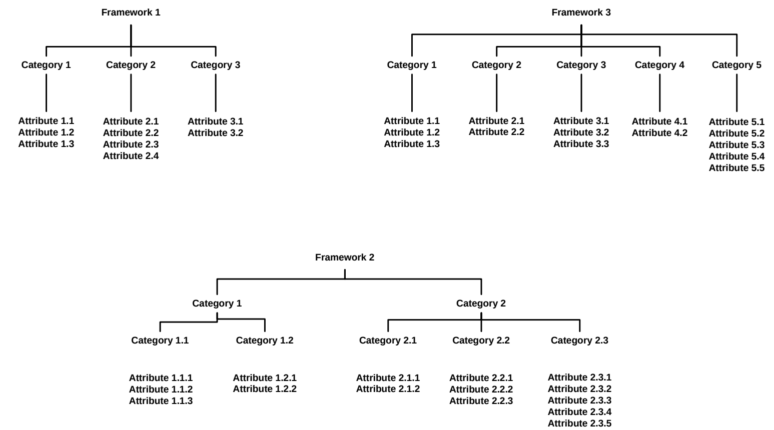
The risk to objectives is used to create a single, integrated view of the current state across multiple, disparate entities.



Multiple Organizations



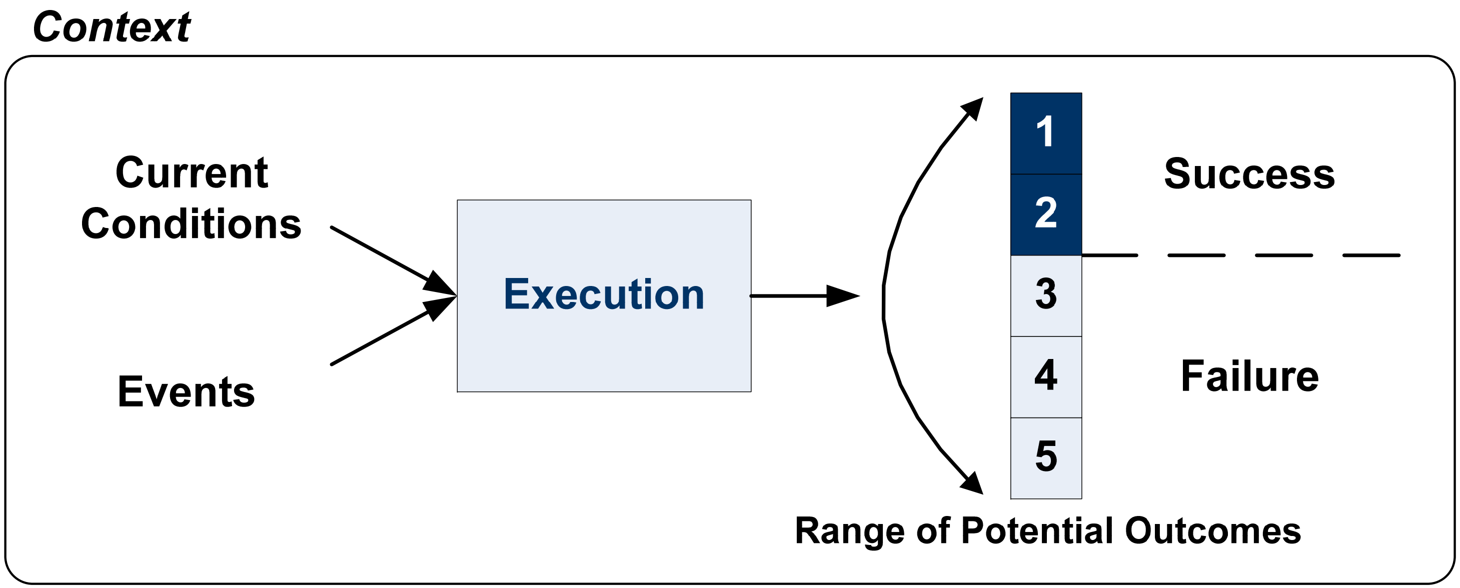
Multiple Systems



Multiple Frameworks

© 2009 Carnegie Mellon University

MOSAIC: *Establishing and Maintaining Confidence*

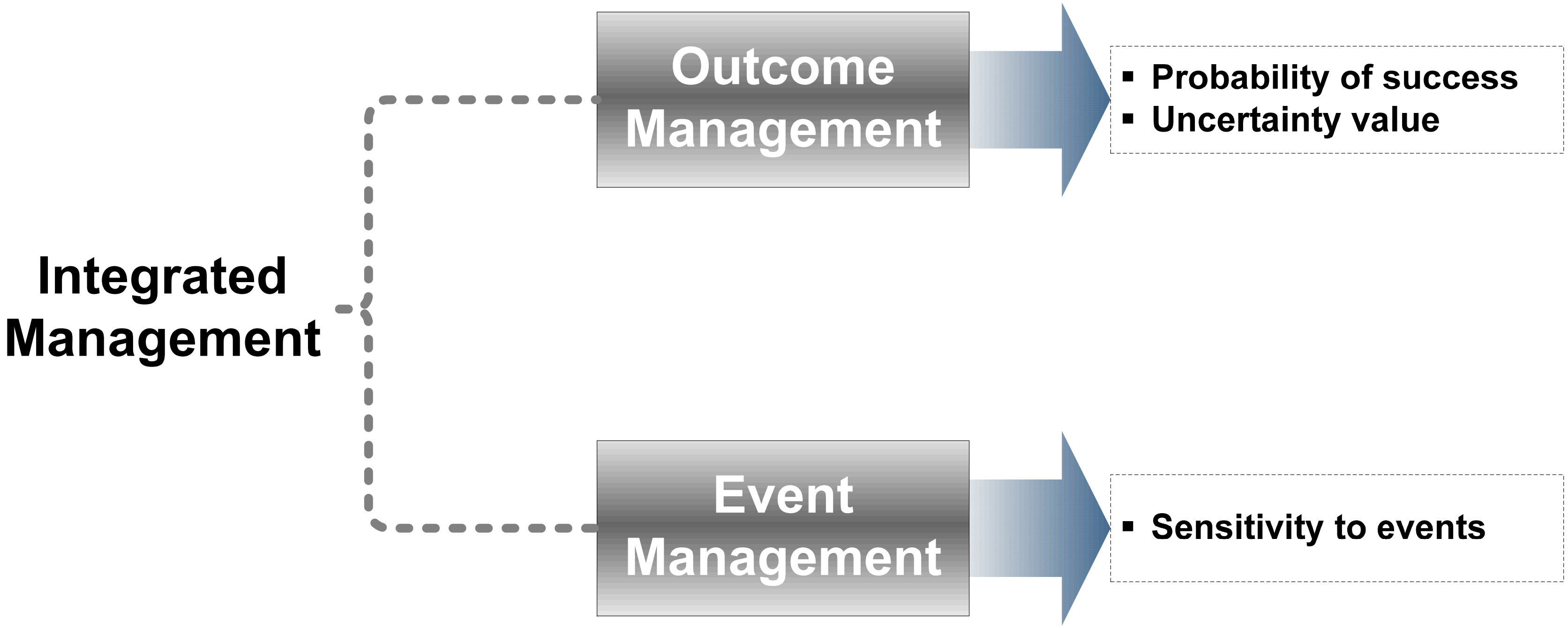


Confidence is established by analyzing the effects of

- Current conditions (with positive and negative consequences)
- Events (with positive and negative consequences)

© 2009 Carnegie Mellon University

MOSAIC: *Managing Outcomes and Events - 1*



Achieving key objectives requires

1. Establishing sufficient momentum toward key objectives (outcome management)
2. Maintaining momentum by managing events (event management)

© 2009 Carnegie Mellon University

MOSAIC: *Managing Outcomes and Events - 2*

The goal of outcome management is to maximize the overall likelihood of

- Achieving key objectives
- Realizing the business/mission opportunity

The goal of event management is to

- Maximize the potential, positive consequences of events (tactical opportunities)
- Minimize the potential, negative consequences of events (tactical risks)

© 2009 Carnegie Mellon University

MOSAIC: *Focus on Key Objectives - 1*

An objective is a desired outcome, or future result.

A ***key objective***

- Is a vital outcome intended to be achieved in the future
- Provides a benchmark against which success will be judged

A set of key objectives define the mission, or *picture of success*, for a project or process.

© 2009 Carnegie Mellon University

MOSAIC: *Focus on Key Objectives - 2*

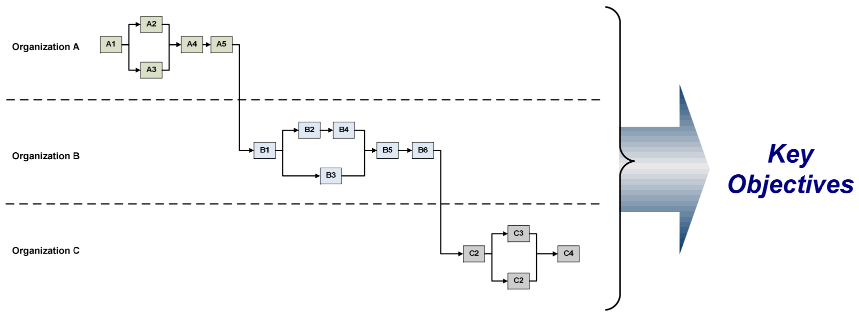
Key objectives typically incorporate multiple perspectives, including

- Program
- Operational and mission
- Business
- Enterprise
- Stakeholders
- Near- and long-term views of success
- Others

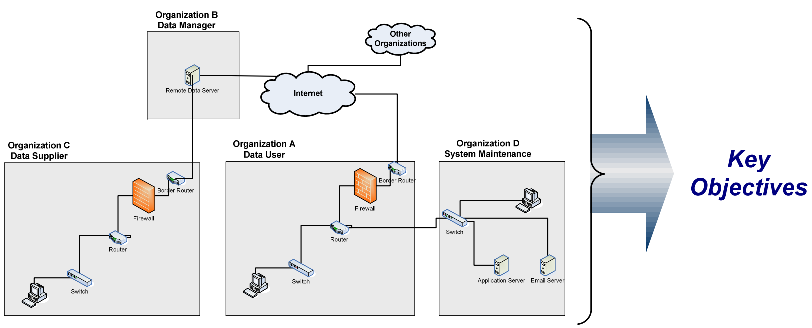
© 2009 Carnegie Mellon University

MOSAIC: *Focus on Key Objectives - 3*

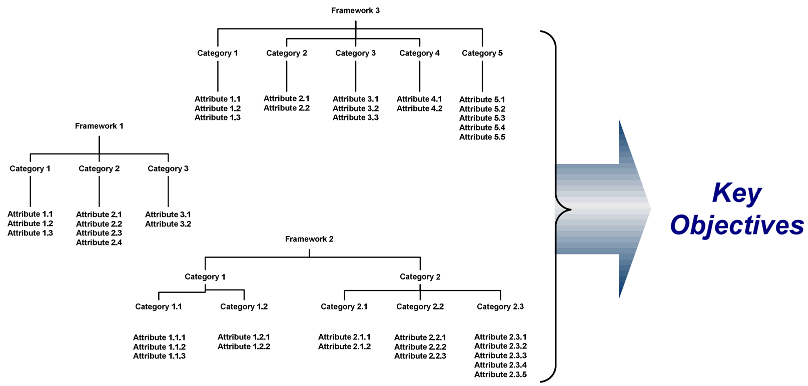
Key objectives are used to set the scope when assessing in a distributed environment.



Multiple Organizations



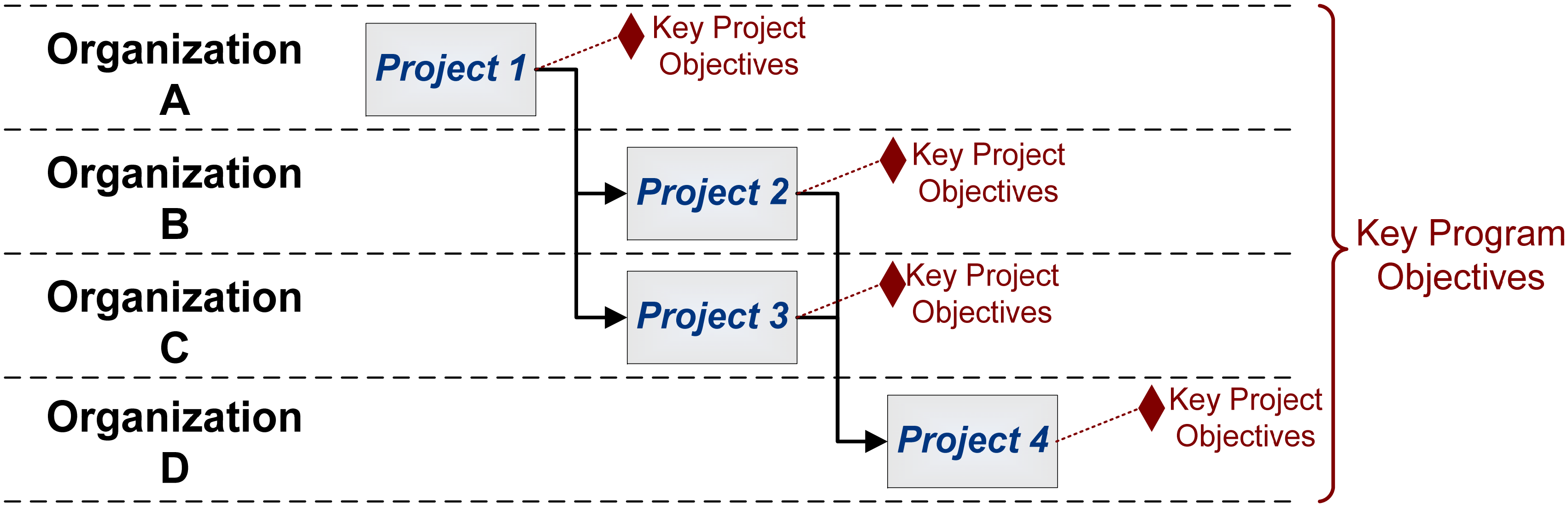
Multiple Systems



Multiple Frameworks

© 2009 Carnegie Mellon University

Network of Objectives



Success of a collaborative venture requires ensuring that all of the key objectives within the network are aligned and on track for success.

MOSAIC is designed to manage risk across a network of objectives.

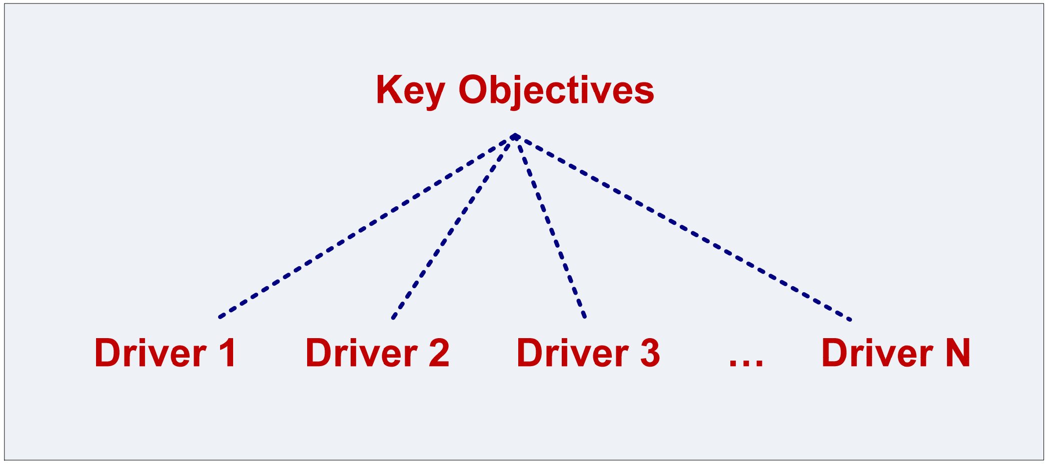
© 2009 Carnegie Mellon University

Drivers of Program Success (and Failure)

A driver is a key factor that steers a program towards success or failure.

Drivers are derived from key objectives

- Drivers are contextual – specific to a program.
- Many drivers are, however, common to most programs.

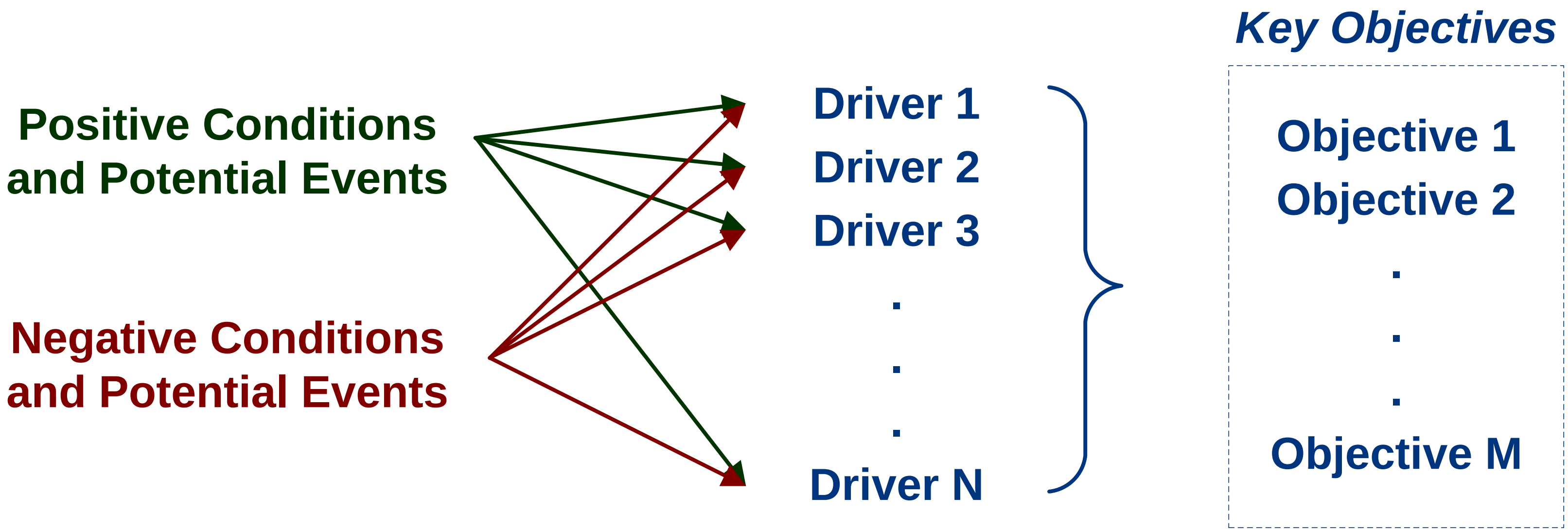


© 2009 Carnegie Mellon University

Evaluating Drivers

Each driver is evaluated to determine how it is currently affecting the outcome, or result.

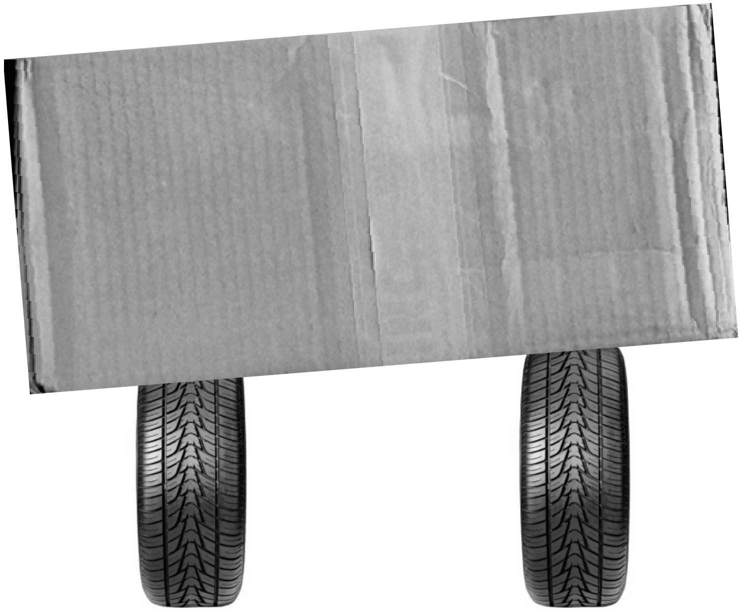
- A ***success driver*** guides the outcome toward the desired state (i.e., key objectives).
- A ***failure driver*** guides the outcome away from the desired state, creating risk to objectives.



© 2009 Carnegie Mellon University

Drivers and Momentum

The current values of the success and failure drivers indicate how much momentum towards success the program has and how resilient it is to risks and unexpected events.



Vs.

© 2009 Carnegie Mellon University

Driver Attributes

Attribute	Description	Example
Name	Short, 1-3 word identifier	Task Execution
Success State	Driver acting as a factor for success (in statement form)	Tasks and activities are performed effectively and efficiently.
Failure State	Driver acting as a factor for failure (in statement form)	Tasks and activities are not performed effectively and efficiently.
Category	Each driver belongs to one of six categories	Execution

© 2009 Carnegie Mellon University

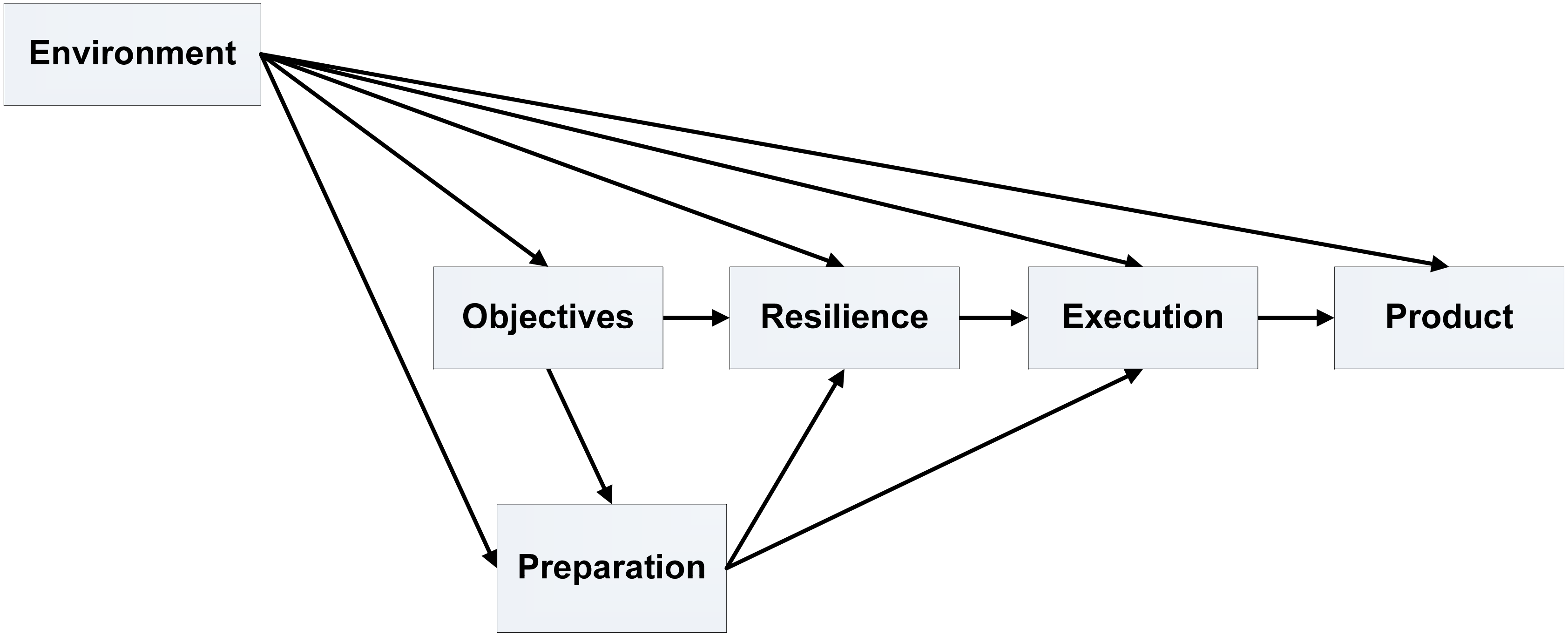
Driver Categories

Six standard categories of drivers are considered when assessing software programs.

- Objectives
- Preparation
- Environment
- Execution
- Resilience
- Product

© 2009 Carnegie Mellon University

Primary Relationships Among the Driver Categories



© 2009 Carnegie Mellon University

Example: *Drivers for Distributed Software Programs*

Driver Framework

Objectives

1. Program Objectives

Preparation

2. Plan
3. Process

Environment

4. Organizational Conditions
5. Compliance

Execution

6. Task Execution
7. Coordination
8. External Interfaces
9. Information Management
10. Technology
11. Facilities and Equipment

Resilience

12. Event Management

Product

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

© 2009 Carnegie Mellon University

Exercise Two

Refer to Tutorial Workbook, Exercise #2 and the Scenario from Exercise #1

Consider the following question:

- Which failure drivers contributed to the problems experienced by the program?

© 2009 Carnegie Mellon University

Changing Risk Management Paradigm

Threat-Driven Risk Management	Objective-Driven Risk Management
• Managing hazards	• Achieving key objectives
• Centralized management	• Distributed/collaborative management
• Point solutions	• Integrated approach
• Tactical focus	• Systemic focus
• Bottom-up analysis	• Top-down analysis

© 2009 Carnegie Mellon University

THE MISSION DIAGNOSTIC

© 2009 Carnegie Mellon University

Mission Diagnostics: *Foundation for MOSAIC*

The Mission Diagnostic approach provides the foundation for all MOSAIC assessments.

- Identify key objectives
- Select and tailor the drivers
- Analyze drivers
- Analyze risk

© 2009 Carnegie Mellon University

Mission Diagnostics

A class of driver-based assessments that incorporate a basic back-end analysis, such as

- Gap analysis (mission gap diagnostic)
- Risk analysis (mission risk diagnostic)
- Success analysis (mission success diagnostic)

The Mission Diagnostic in this tutorial is the mission success diagnostic.

© 2009 Carnegie Mellon University

What Is a Mission?

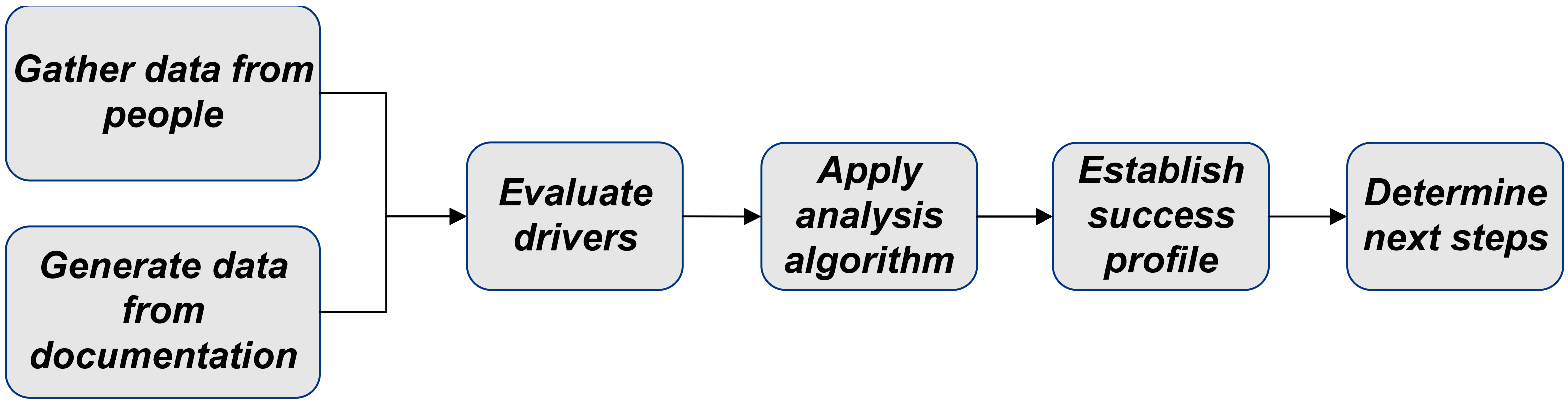
The term mission has multiple meanings, depending on the context in which it is used.

For example, mission is used to describe any of the following:

- Purpose of an organization
- Goals of a specific department or group within a larger organization
- Objectives of each activity in a work process
- Function of each technology (e.g., a software-intensive system) that supports a project or process
- Specific result being pursued when executing a project or process

© 2009 Carnegie Mellon University

Mission Diagnostic Activities



© 2009 Carnegie Mellon University

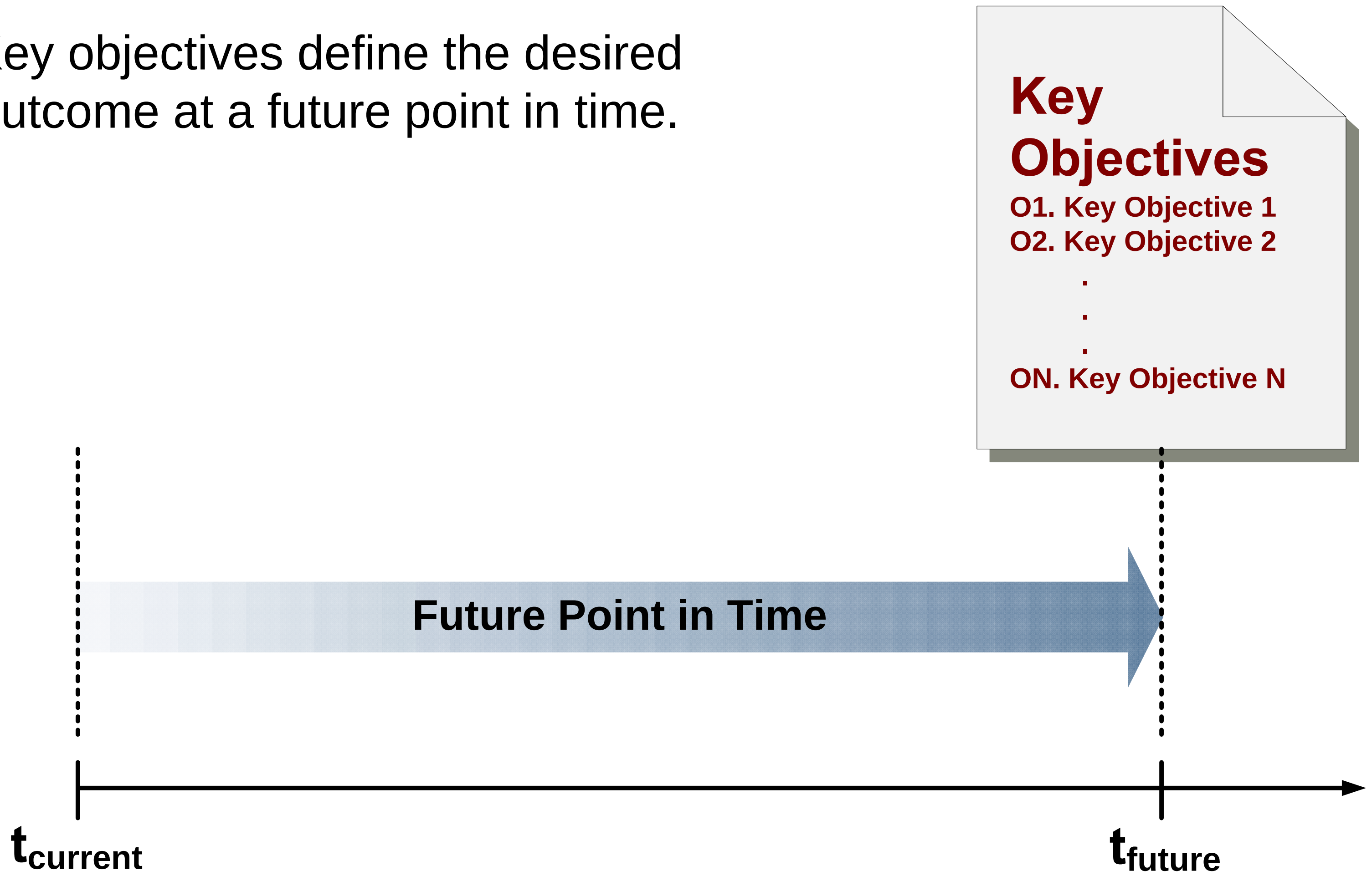
The Mission Diagnostic

ESTABLISHING KEY OBJECTIVES

© 2009 Carnegie Mellon University

Establishing Key Objectives

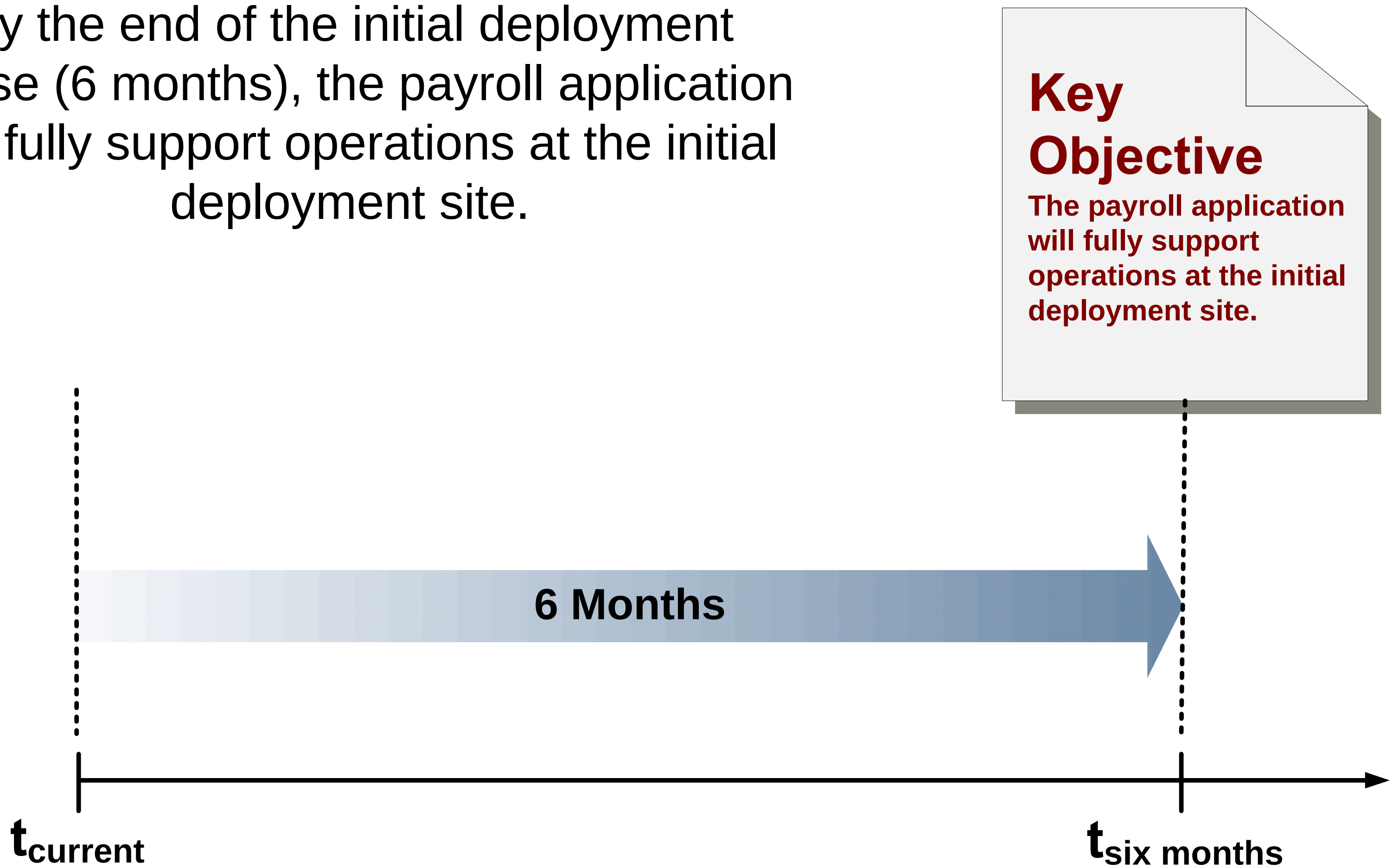
Key objectives define the desired outcome at a future point in time.



© 2009 Carnegie Mellon University

Example: *Key Objective*

By the end of the initial deployment phase (6 months), the payroll application will fully support operations at the initial deployment site.



© 2009 Carnegie Mellon University

The Mission Diagnostic

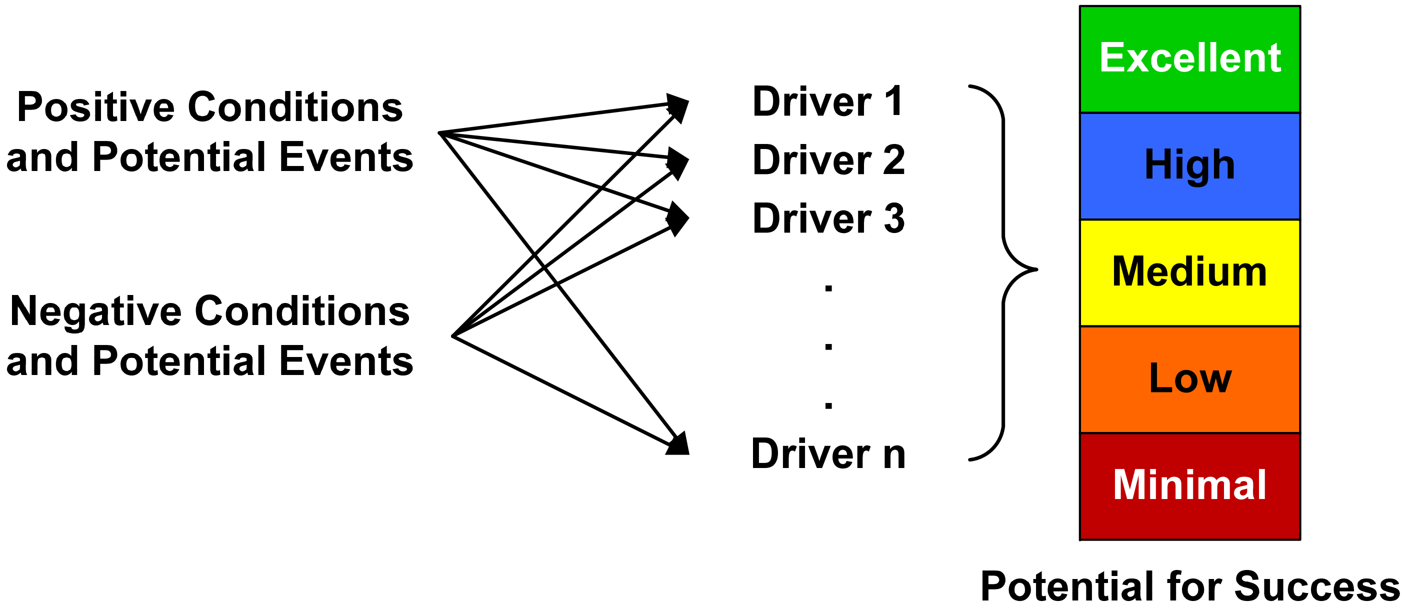
TAILOR DRIVERS

© 2009 Carnegie Mellon University

Reminder: *What Are Drivers?*

A driver is a situation or circumstance that has a strong influence on the outcome or result.

- A ***success driver*** guides the outcome toward key objectives.
- A ***failure driver*** guides the outcome away from key objectives.



© 2009 Carnegie Mellon University

Example: *Tailoring Drivers - 1*

1. Program Objectives
2. Plan
3. Process
4. Organizational Conditions
5. Compliance
6. Task Execution
7. Coordination
8. External Interfaces
9. Information Management
10. Technology
11. Facilities and Equipment
12. Event Management
13. Requirements
14. Design and Architecture
15. System Capability
16. System Integration
17. Operational Support
18. Adoption Barriers
19. Operational Preparedness
20. Certification and Accreditation

© 2009 Carnegie Mellon University

Example: *Tailoring Drivers - 2*

Adjust the drivers based on your current context and objectives, for example

- Where you are in the life cycle
- Type of project, such as new development vs. maintenance
- Terminology

Expand drivers

- Plan can be expanded to Plan, Budget, and Schedule

Collapse drivers

- Operational Support, Adoption Barriers, and Operational Preparedness could be collapsed into Operations

Add drivers as needed for completely new aspects or new types of objectives

© 2009 Carnegie Mellon University

The Mission Diagnostic

EVALUATE DRIVERS

© 2009 Carnegie Mellon University

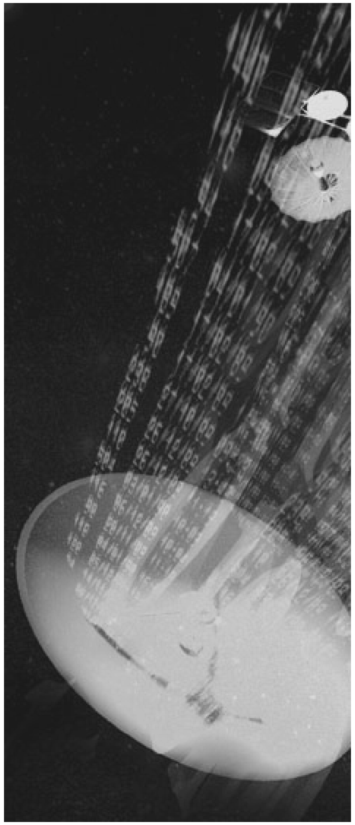
Collect Information

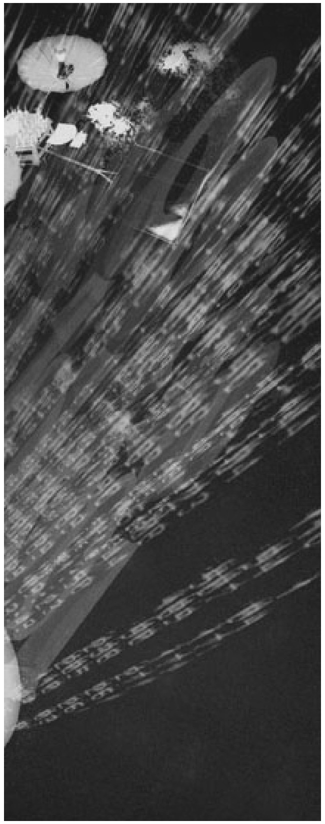
To evaluate a driver, you need information from

- Program personnel, all levels and groups
- Program documentation
- Other sources

Gather information from

- Interviews
- Documentation reviews
- Group meetings to reach consensus on drivers





© 2009 Carnegie Mellon University

Example: *Driver Question*

Question	Answer				
	No	Likely no	Equally likely	Likely yes	Yes
1. Is the process being used to develop (and deploy) the system sufficient?	❑	❑	❑	❑	❑

Driver questions are phrased from the success perspective.

Probability is incorporated into the range of answers for each driver.

© 2009 Carnegie Mellon University

Example: *Driver Value Criteria*

Each driver is evaluated against predefined criteria.

You must also record the rationale for your evaluation of each driver.

Measure	Description
Yes	The answer is almost certainly "yes." Almost no uncertainty exists. There is little or no probability that the answer could be "no." (~ > 95% probability of yes)
Likely yes	The answer is most likely "yes." There is some chance that the answer could be "no." (~ 75% probability of yes)
Equally Likely	The answer is just as likely to be "yes" or "no." (~ 50% probability of yes)
Likely no	The answer is most likely "no." There is some chance that the answer could be "yes." (~ 25% probability of yes)
No	The answer is almost certainly "no." Almost no uncertainty exists. There is little or no probability that the answer could be "yes." (~ < 5% probability of yes)

© 2009 Carnegie Mellon University

Example: *Evaluating Drivers*

Question		Answer				
		No	Likely no	Equally likely	Likely yes	Yes
1.	Is the process being used to develop (and deploy) the system sufficient?	❑	X	❑	❑	❑

Each driver is evaluated based on the data that have been collected.

The rationale for selecting an answer is recorded.

© 2009 Carnegie Mellon University

Example: *Rationale for Driver Value - 1*

Driver	Value
1. Is the process being used to develop (and deploy) the system sufficient?	Equally Likely Yes or No

Rationale

+ Previous programs have a 90% history of delivering on-time with required functionality.

+ The lead engineers are skilled at adapting to new processes.

- This program required a significant change in our standard processes. There was no new training created for the new processes.

© 2009 Carnegie Mellon University

Example: *Rationale for Driver Value - 2*

Rationale (cont.)

- QA did not have a chance to review the new and revised processes before they were put into practice.

- The person who developed the new processes quit last week.

- There are a lot of brand new programmers (45%).

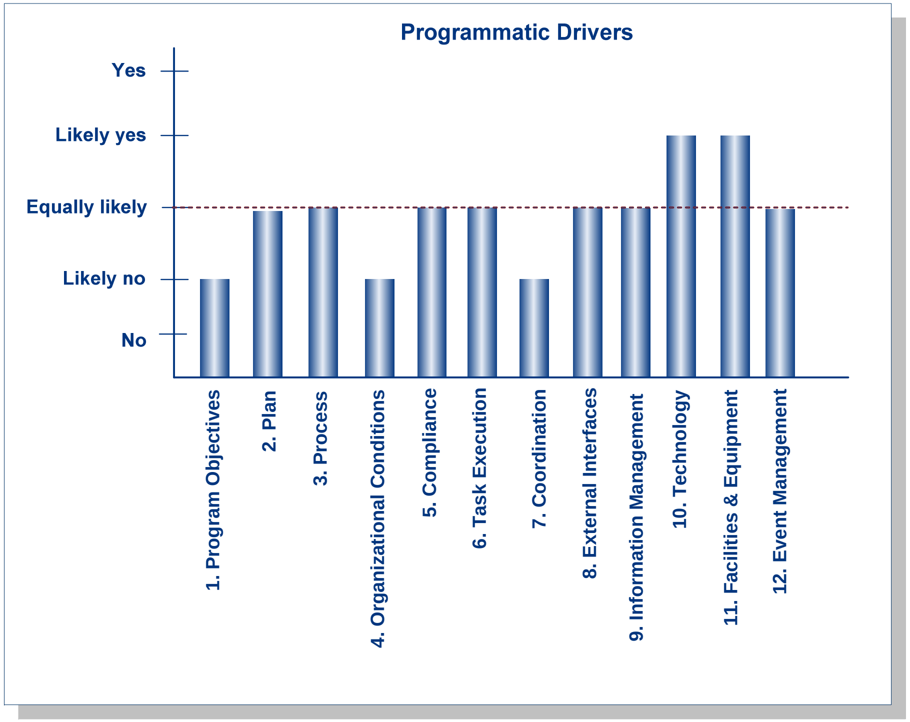
© 2009 Carnegie Mellon University

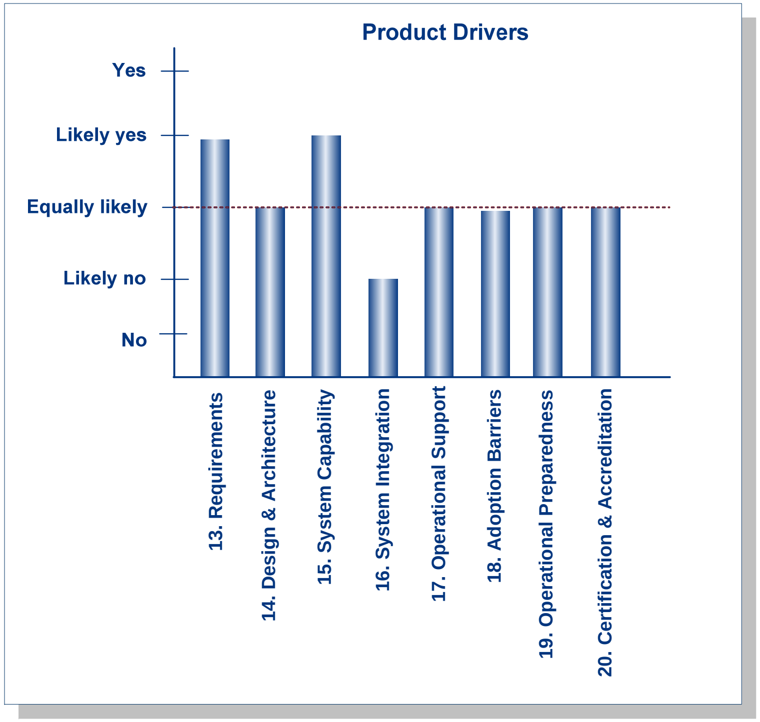
The Mission Diagnostic

RISK PROFILE OR POTENTIAL FOR SUCCESS

© 2009 Carnegie Mellon University

Example: *Driver Profile - 1*





A simple analysis provides insight into the potential for success.

© 2009 Carnegie Mellon University

Mission Diagnostic Results: *Driver Profile - 2*

Driver Framework

Objectives

1. Program Objectives

Environment

4. Organizational Conditions
5. Compliance

Resilience

12. Event Management

Preparation

2. Plan
3. Process

Execution

6. Task Execution
7. Coordination
8. External Interfaces
9. Information Management
10. Technology
11. Facilities and Equipment

Product

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

A driver profile can also present risks in relation to the driver framework.

© 2009 Carnegie Mellon University

Mission Diagnostic Results: *Traditional "Risk List"*

Risk Statement	Risk Probability	Risk Severity	Risk Exposure
1. Project objectives are unrealistic or unachievable.	High	Severe	High
14. The project does not have sufficient capacity and capability to identify or manage unpredictable events and changing circumstances.	Medium	Severe	Medium
7. The project does not comply with all relevant policies, laws, and regulations.	Medium	Low	Minimum

A risk profile can be a list of current risks (impact to key objectives is implied) derived from drivers.

© 2009 Carnegie Mellon University

Potential for Success

The potential for success is the likelihood that a key objective will be achieved (also called the probability of success).

The current potential for success is an indicator of mission risk and opportunity.

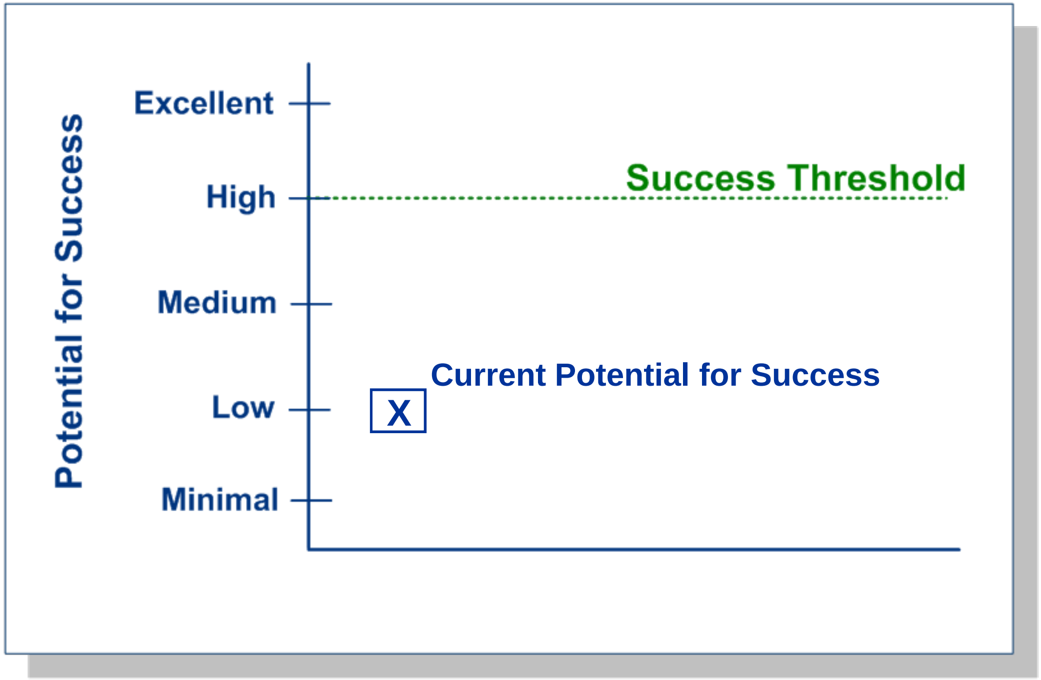
- A high potential for success is an indicator of mission opportunity.
- A low potential for success is an indicator of mission risk.

An analysis of present conditions (as represented by drivers of success) is used to establish the current potential for success.

- Simple aggregation of driver values
- Weighted aggregation of driver values
- Mean or median driver value
- Rule-based algorithms

© 2009 Carnegie Mellon University

Example: *Basic Success Profile*



A basic success profile depicts the current potential for success in relation to the success threshold.

The success threshold defines the desired, or target, potential for success

© 2009 Carnegie Mellon University

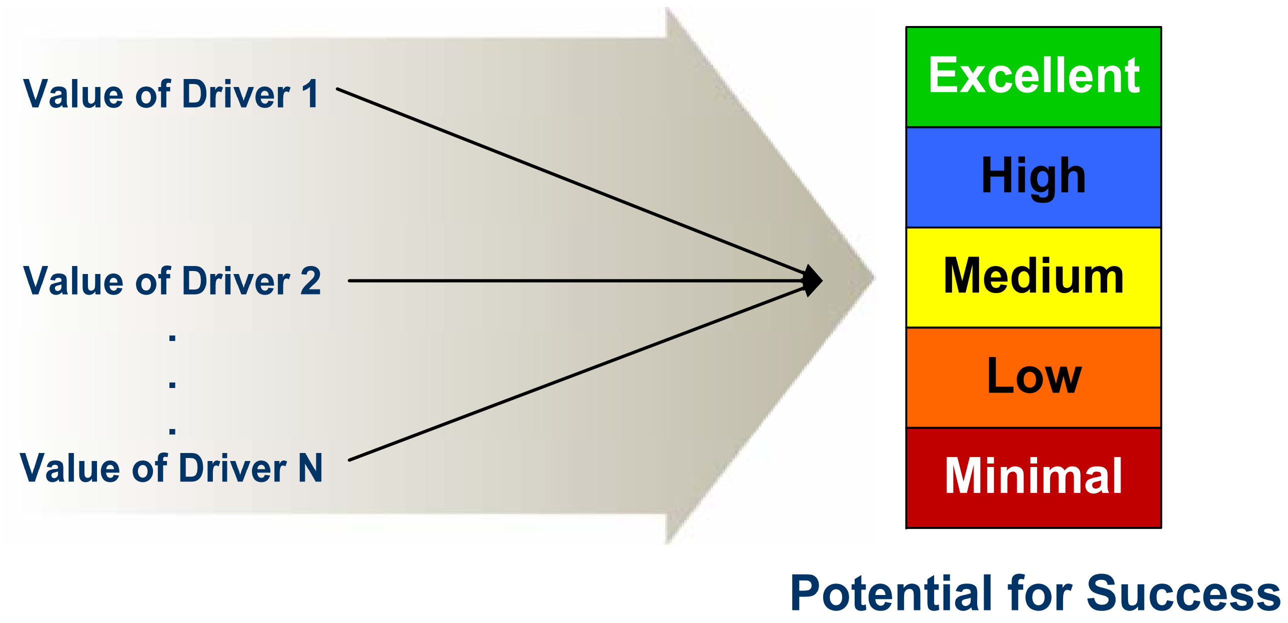
Example: *Success Criteria*

Each key objective is analyzed in relation to a set of success criteria to determine its current potential for success.

Measure	Description
Excellent	Current conditions are extremely favorable for a successful outcome. (~ > 95% chance of success)
High	Current conditions are favorable for a successful outcome. (~ 75% chance of success)
Medium	Current conditions are mixed, making success and failure equally likely. (~ 50% chance of success)
Low	Current conditions are not favorable for a successful outcome. (~ 25% chance of success)
Minimal	Current conditions are extremely unfavorable for a successful outcome. (~ < 5% chance of success)

© 2009 Carnegie Mellon University

Example: *Objective-based Potential for Success*



An analysis of drivers is used to determine the *current potential for success* for each objective.

© 2009 Carnegie Mellon University

Example: *Success Potential of a Key Objective*

By the end of the initial deployment phase (6 months), the payroll application will fully support operations at the initial deployment site.

Current Potential for Success

Current likelihood of achieving this scenario is ***Low***

Rationale

- System functionality was cut to meet the deployment schedule at the initial deployment site.
- The contractor developing the payroll application has not been meeting its milestones.
- The integration task is more complicated than usual.
  - The integration schedule is shorter than usual.
  - The infrastructure is dynamic and evolving.
  - No one is managing the common enterprise infrastructure.
- Changes in senior management could affect the ability to resolve infrastructure issues.

© 2009 Carnegie Mellon University

The Mission Diagnostic

NEXT STEPS

© 2009 Carnegie Mellon University

Next Steps

Determine what areas need

- Further investigation
- Improvement

If further investigation is needed

- Gather additional information to clarify uncertainties
- Continue decomposing drivers to get at deeper issues
- Chose alternate methods to analyze the situation

If improvement is needed

- Determine causes of weaknesses
- Develop and implement improvement plans
- Re-evaluate

© 2009 Carnegie Mellon University

The Mission Diagnostic

YOUR PROGRAM

© 2009 Carnegie Mellon University

Exercise Three

Refer to Tutorial Workbook, Exercise #3

1. Select a program, project, or process with which you are knowledgeable.
2. Evaluate it using the set of drivers provided in the Workbook.
3. Sketch your risk profile.

Consider:

- *Are there some drivers for which you need more information?*
- *Where would you get that information?*

© 2009 Carnegie Mellon University

BEYOND THE BASIC MISSION DIAGNOSTIC

© 2009 Carnegie Mellon University

Expanding the Mission Diagnostic

There are many ways to expand the Mission Diagnostic.

Two will be briefly introduced:

- Dealing with uncertainty
- Handling events

© 2009 Carnegie Mellon University

Uncertainty about Current Conditions

Uncertainty is defined as having doubt or being unsure of something.

As you analyze information, one or more of the following will likely be true:

- Certain information is not available or is unknown.
- You do not trust certain information based on its source.
- Some information is based on people's assumptions or opinions, which might prove to be incorrect.

Some uncertainty will be associated with the potential for success based on having doubts about or being unsure of current conditions (i.e., driver values).

© 2009 Carnegie Mellon University

Changes in Uncertainty Over Time

Many uncertainties related to current conditions can be resolved.

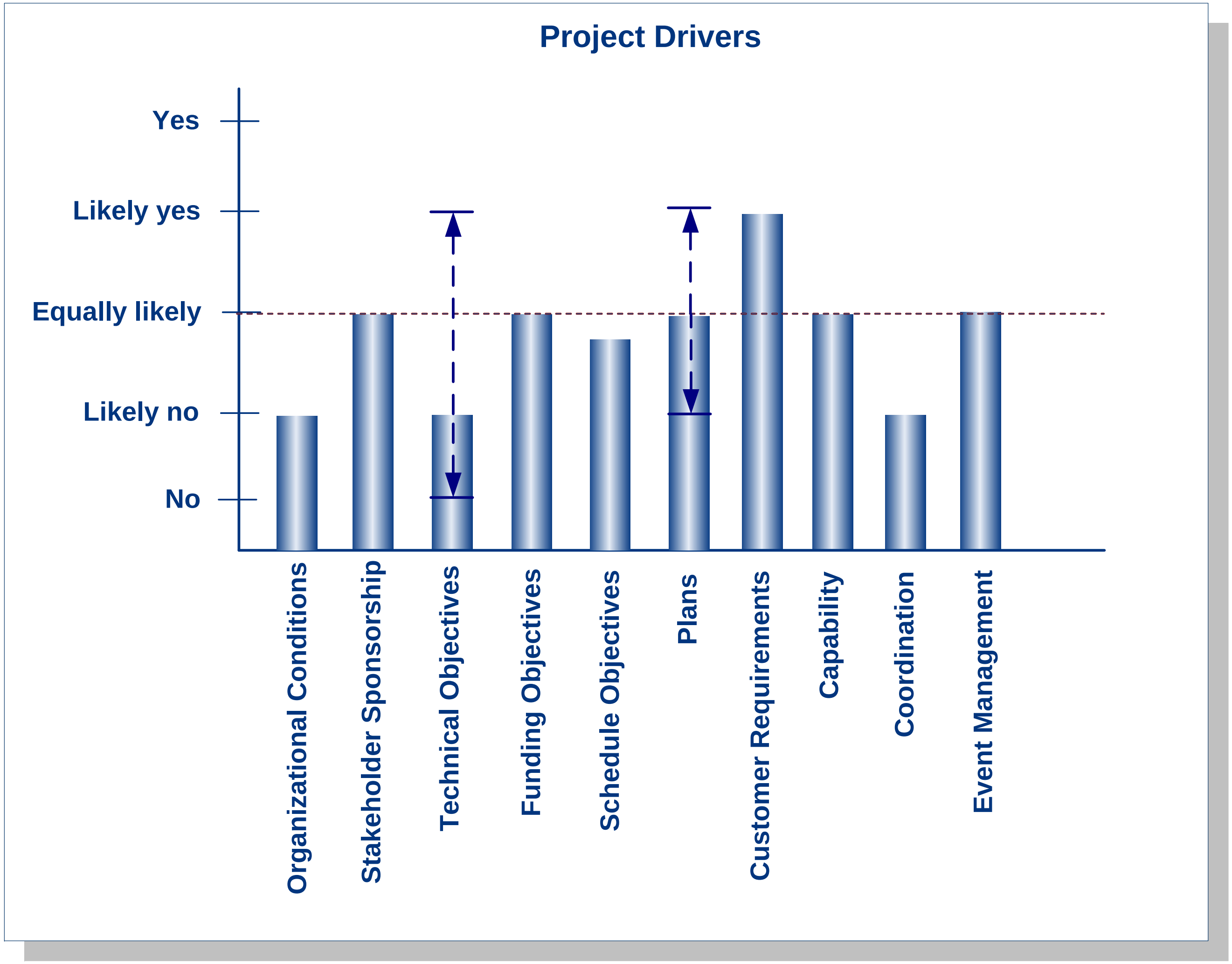
- More information becomes available or known.
- Information from untrusted sources can be verified.
- Assumptions can be tested and proved to be correct or incorrect.

You will almost always have some degree of uncertainty related to current conditions.

- You will not be able to resolve all uncertainties.
- Information will always be imperfect.
- Changing conditions will produce new uncertainties.

© 2009 Carnegie Mellon University

Example: *Uncertainty in Drivers*



© 2009 Carnegie Mellon University

Events

An event is an unpredictable occurrence that changes the current state (i.e., status quo).

Events can have a positive or negative effect on the outcome.

- A decrease in funding would likely produce a negative consequence that might adversely affect a project's outcome.
- An increase in funding would likely produce a positive consequence that might put a project in better position for success.

A sensitivity analysis examines an event's likely effect on the potential for success

- Increase in the potential for success resulting from the occurrence of an event (i.e., tactical opportunity)
- Decrease in the potential for success resulting from the occurrence of an event (i.e., tactical risk)

© 2009 Carnegie Mellon University

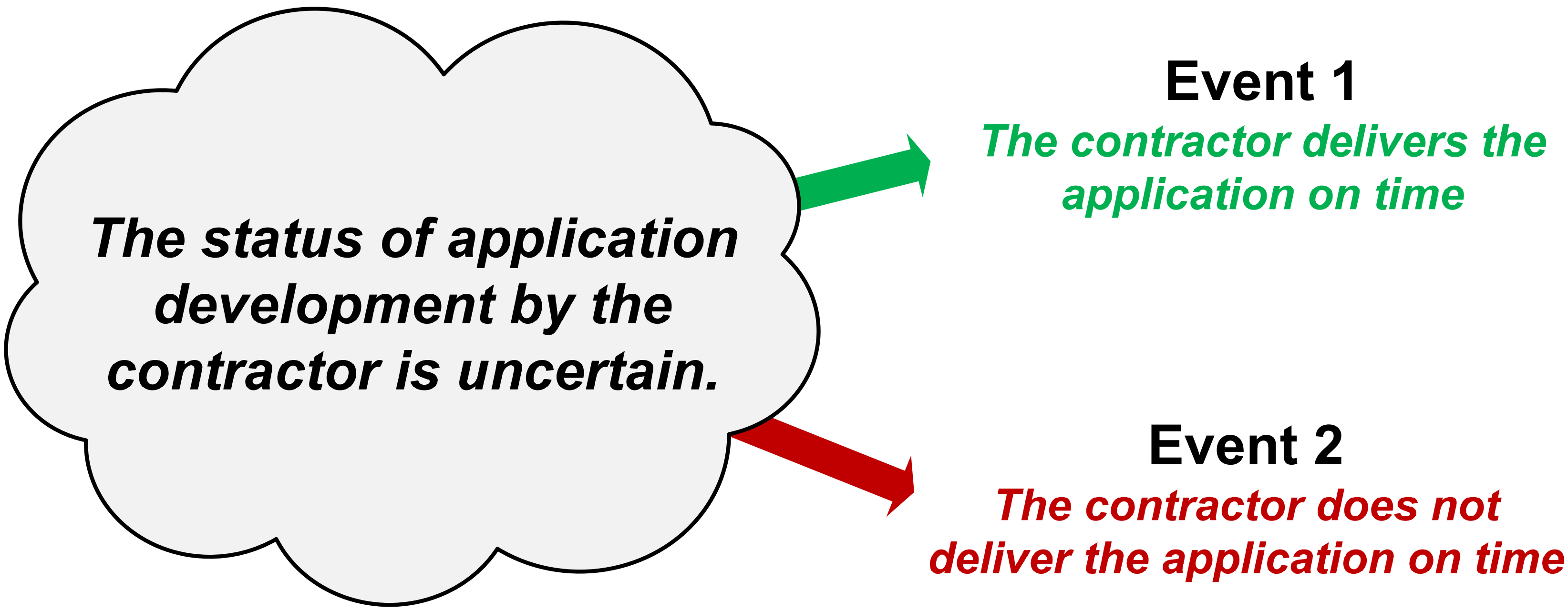
Event Identification

Relevant events are identified based on

- *Uncertainties*
  Resolution of uncertainties can change current conditions and affect the potential for success.
- *Vulnerabilities*
  Inherent weaknesses can expose a program to the effects of unpredictable occurrences.

© 2009 Carnegie Mellon University

Example: *Events Resulting from an Uncertainty*



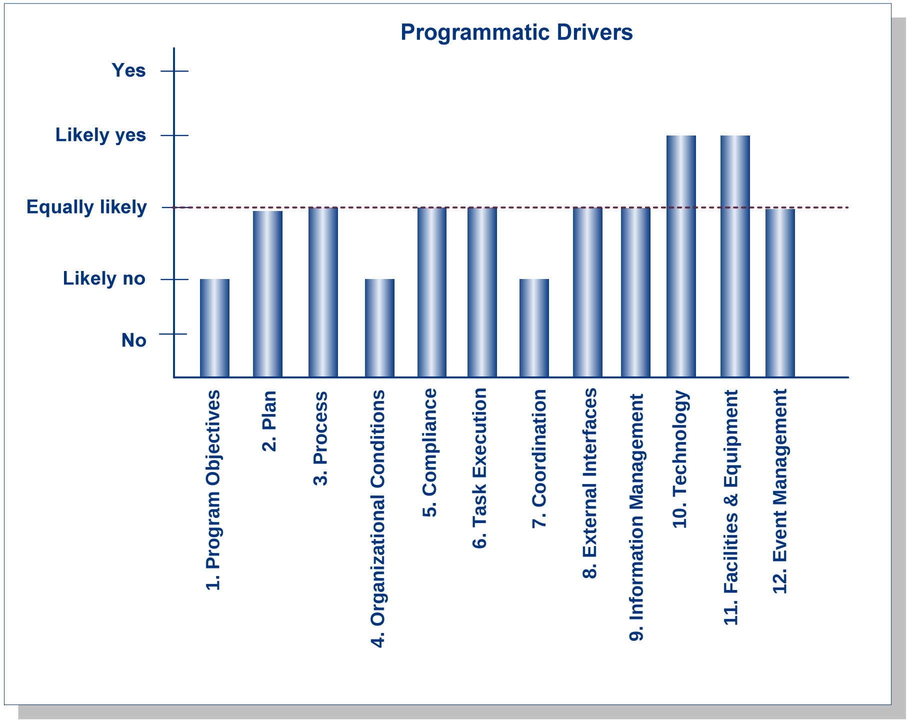
© 2009 Carnegie Mellon University

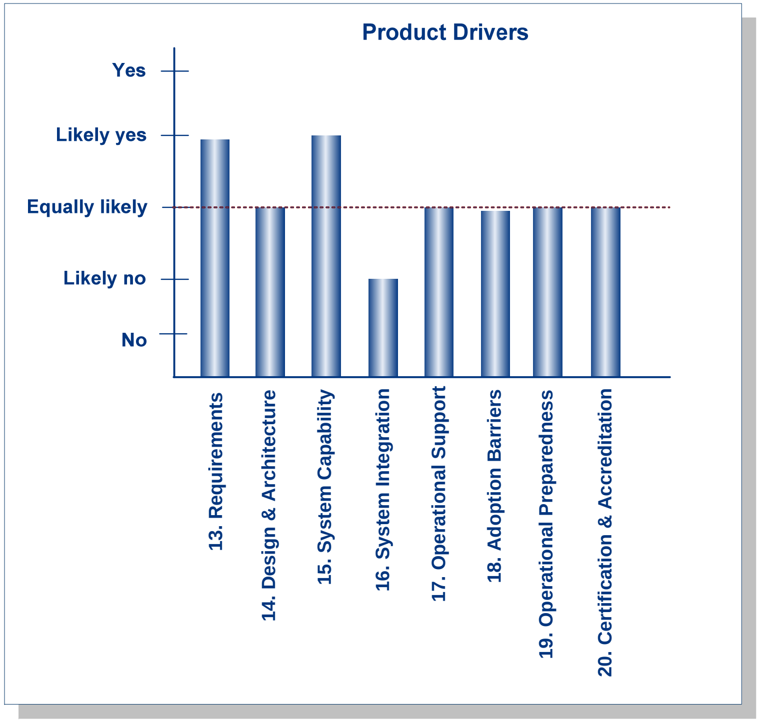
Example: *Events Resulting from an Uncertainty*

Event	Consequence
E1. If the contractor delivers the application on time	Then the project's potential for success will remain ***low*** (i.e., no change)
E2. If the contractor does not deliver the application on time	Then the project's potential for success will fall to ***minimal***

© 2009 Carnegie Mellon University

Example: *If the contractor delivers the application on time (Event 1)*



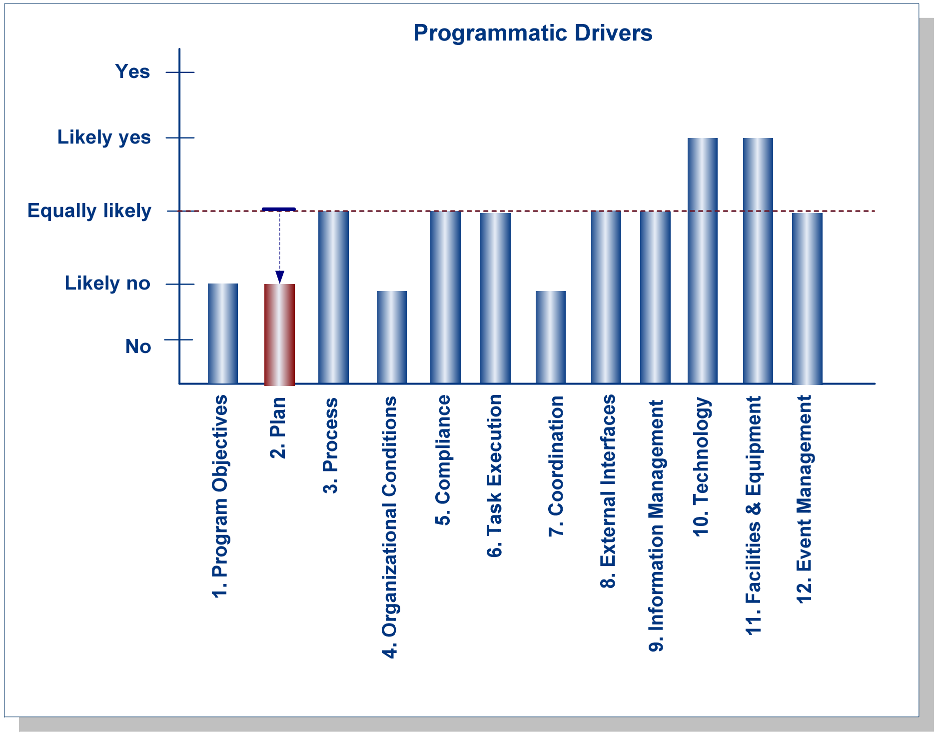


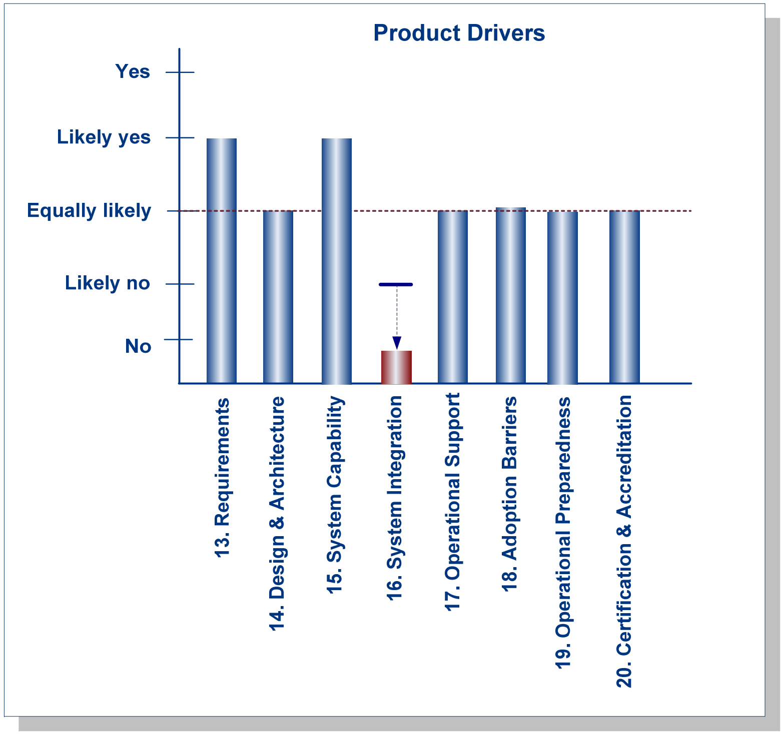
If the contractor delivers the application on time, the values of all drivers will remain at their current levels.

The potential for success will remain ***Low***.

© 2009 Carnegie Mellon University

Example: *If the contractor does not deliver the application on time (Event 2)*



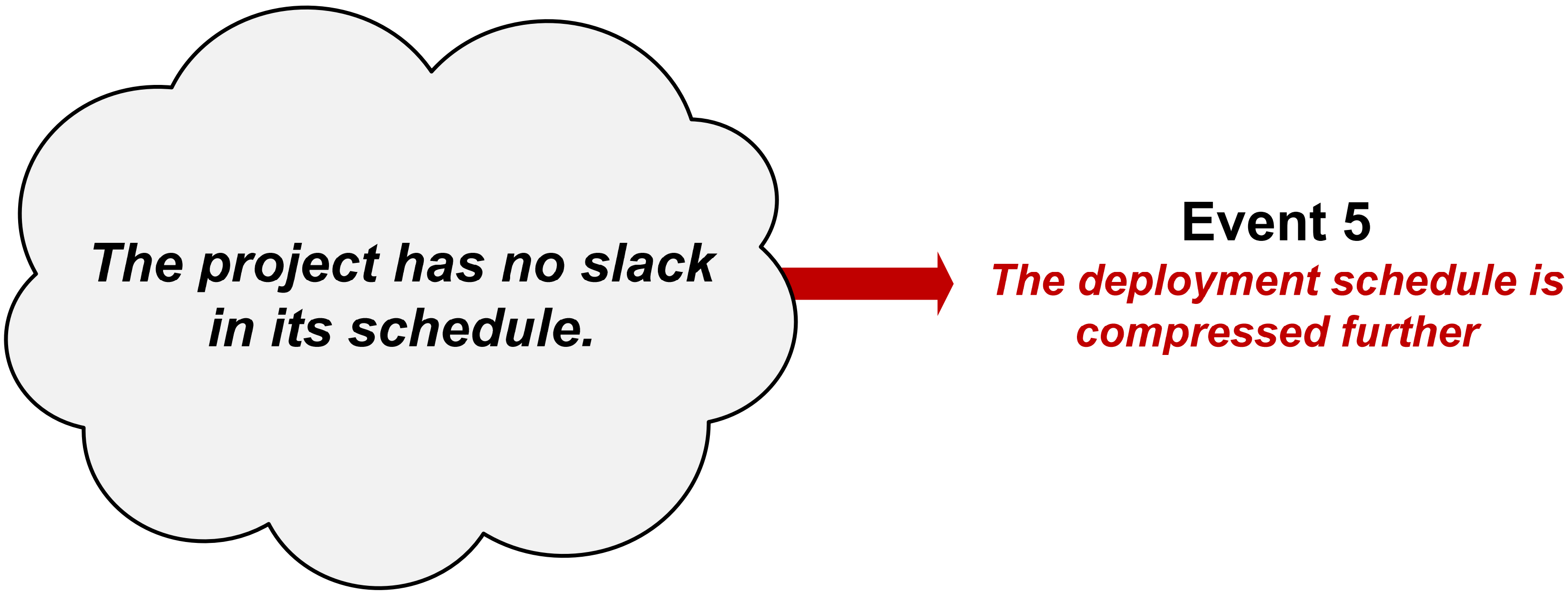


If the contractor does not deliver the application on time, the values of the drivers for plans and system integration will decrease.

The potential for success will drop to ***Minimal***.

© 2009 Carnegie Mellon University

Example: *Event Resulting from a Vulnerability*



© 2009 Carnegie Mellon University

Example: *Event Resulting from a Vulnerability*

Event	Consequence
E5. If the deployment schedule is compressed further	Then the project's potential for success will fall to ***minimal***

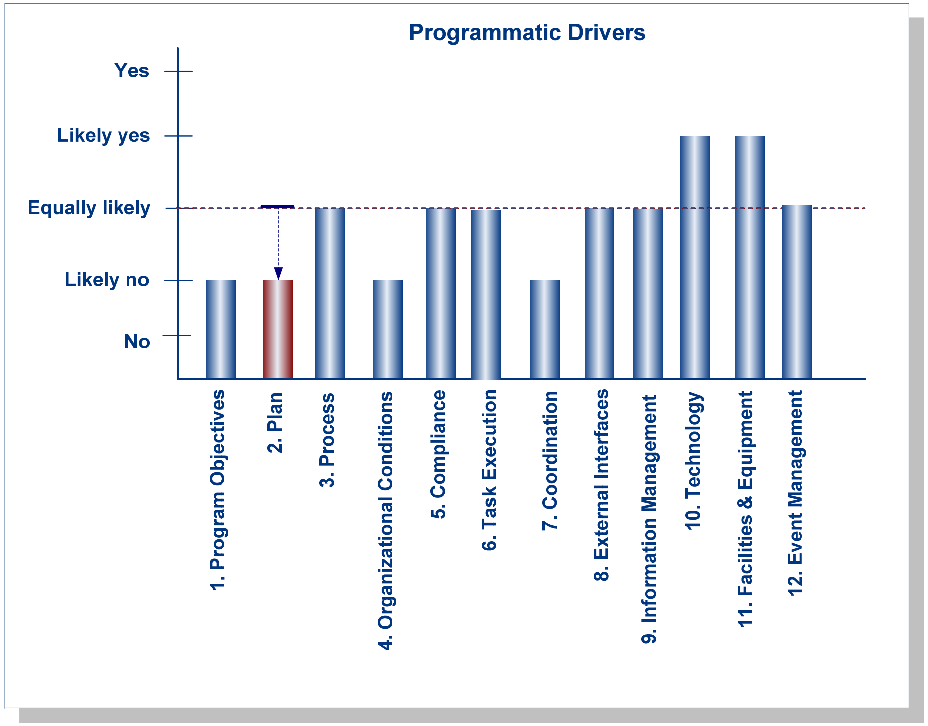
Context

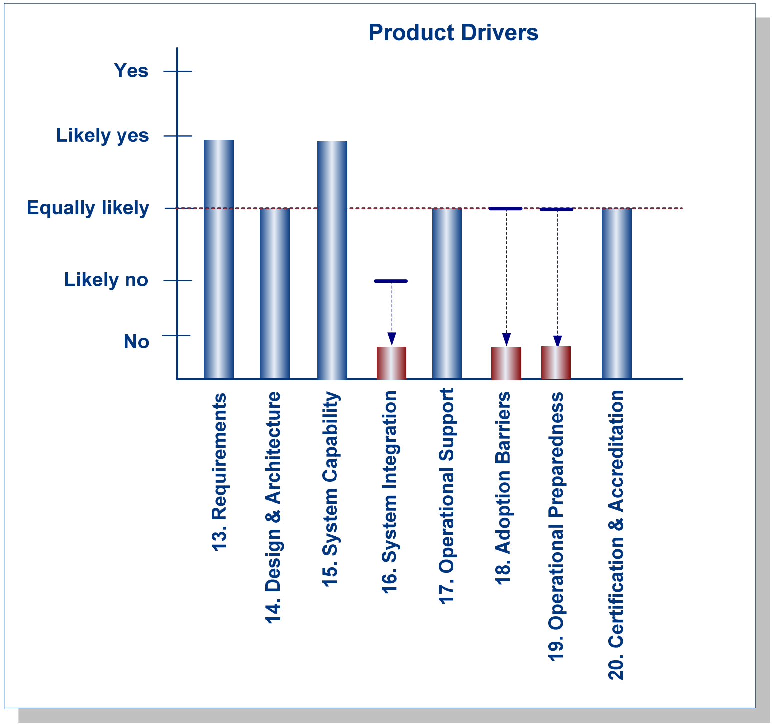
Senior management has had a history of overriding the decisions of the project's management team. Promises made by the CIO had already shortened the development schedule.

The integration issues will be difficult to resolve. These issues make the project especially vulnerable to further schedule changes.

© 2009 Carnegie Mellon University

Example: *If the deployment schedule is compressed (Event 5)*



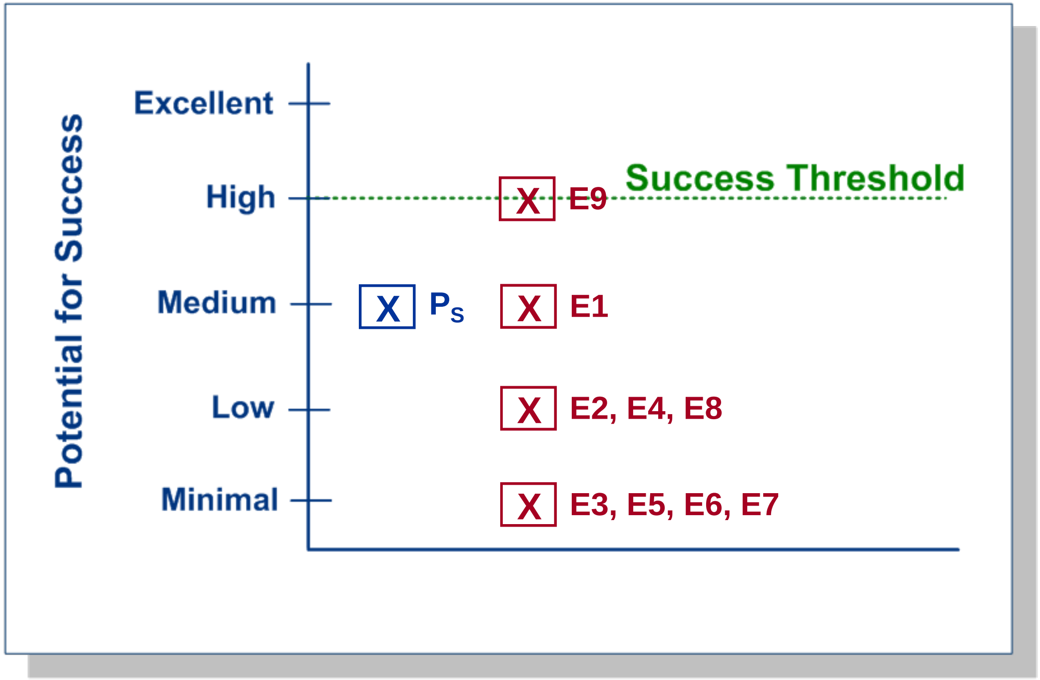


If the deployment schedule is compressed any further, the drivers for plans, system integration, adoption barriers, and operational preparedness will be affected.

The potential for success will drop to ***Minimal***.

© 2009 Carnegie Mellon University

Example: *Success Profile with Event Sensitivity*



The consequences of events can be added to the basic success profile.

This success profile includes one event that increases the potential for success – a tactical opportunity.

© 2009 Carnegie Mellon University

SUMMARY

© 2009 Carnegie Mellon University

Summary of Key Points - 1

The paradigm for managing software programs is changing.

- Increased complexity
- Distributed knowledge, experience, and expertise
- Multiple points of management control
- Focus on communication and coordination

Objective-driven risk management

- Is a structured approach for assessing and managing in distributed environments.
  – Systemic focus
  – Top-down analysis
- Uses the risk to objectives to create a single, integrated view of the current state across multiple, disparate entities

© 2009 Carnegie Mellon University

Summary of Key Points - 2

MOSAIC

- Is a suite of methods that enable objective-driven risk management
- Comprises a range of assessments
  - Basic
  - Intermediate
  - Advanced

The Mission Diagnostic

- Provides a time-efficient means of assessing risks to program objectives
- Focuses on a set of key drivers

© 2009 Carnegie Mellon University

Example: *Drivers for Distributed Software Programs*

Driver Framework

Objectives

1. Program Objectives

Preparation

2. Plan
3. Process

Environment

4. Organizational Conditions
5. Compliance

Execution

6. Task Execution
7. Coordination
8. External Interfaces
9. Information Management
10. Technology
11. Facilities and Equipment

Resilience

12. Event Management

Product

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

© 2009 Carnegie Mellon University

For Additional Information

Christopher Alberts
Email: cja@sei.cmu.edu
Phone: 412-268-3045
Fax: 412-268-5758

Audrey Dorofee
Email: ajd@sei.cmu.edu
Phone: 412-268-6396
Fax: 412-268-5758

WWW	http://www.sei.cmu.edu/msce/index.html
U.S. mail	Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213-3890

For updated slides
http://www.sei.cmu.edu/msce/presentations.html

© 2009 Carnegie Mellon University

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this presentation in its entirety with no modifications is granted.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be directed to the permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

© 2009 Carnegie Mellon University