

Rethinking Risk Management

NDIA Systems Engineering Conference 2009

Audrey Dorofee
Christopher Alberts

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Mission Success in Complex Environments (MSCE) Project

Part of the SEI Acquisition Support Program (ASP), the MSCE Project develops methods, tools, and techniques for assuring success in complex, uncertain environments.

The project builds on more than 17 years of SEI research and development in managing uncertainty.

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

Current work is *Mosaic*, a structured approach for assessing and managing for success in distributed environments.

This tutorial is derived from the Mosaic work.





Topic Areas

Risk Management: *Key Concepts*

A Different Perspective

The Mission Diagnostic

The Risk Diagnostic

Implementation Options

Summary





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 its current potential for success

Understand how to use the Risk Diagnostic to evaluate a program's mission risks

Understand some options for implementing these concepts



RISK MANAGEMENT: *KEY* *CONCEPTS*



What Is Risk?

The likelihood of loss

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

Risk requires the following conditions¹

- A potential loss
- Likelihood
- Choice

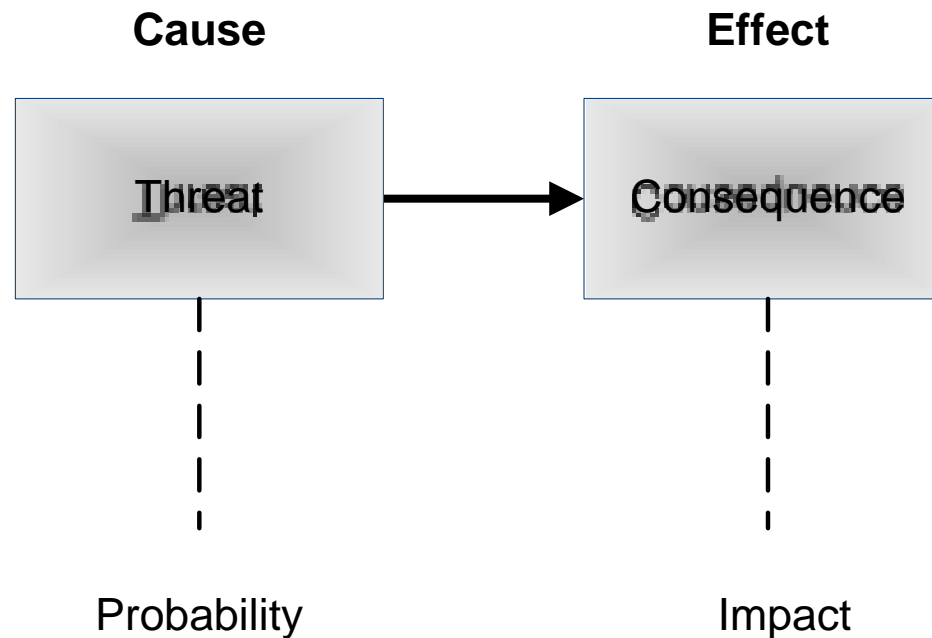
1. Charette, Robert N. *Application Strategies for Risk Analysis*. New York, NY: McGraw-Hill Book Company, 1990



Components of Risk

Risk comprises two core components.

- **Threat** – a circumstance with the potential to produce loss
- **Consequence** – the loss that will occur when a threat is realized



Issue/Problem

A loss or adverse consequence that has occurred or is certain to occur

No uncertainty exists—the loss or adverse consequence has taken place or is certain to take place

An issue or problem can also lead to (or contribute to) other risks by

- Creating a circumstance that produces a new threat
- Making an existing threat more likely to occur
- Aggravating the consequences of existing risks



Opportunity

The likelihood of realizing a gain from an allocation or reallocation of resources

- Defines a set of circumstances that provides the potential for a desired gain
- Requires an investment or action to realize the desired gain (i.e., take advantage of the opportunity)

Pursuit of an opportunity can produce

- New risks or issues
- Change existing risks or issues

Tactical opportunity provides a localized gain (e.g., to program or part of a program)

Business opportunity is a gain for the organization



Types of Risk

Speculative

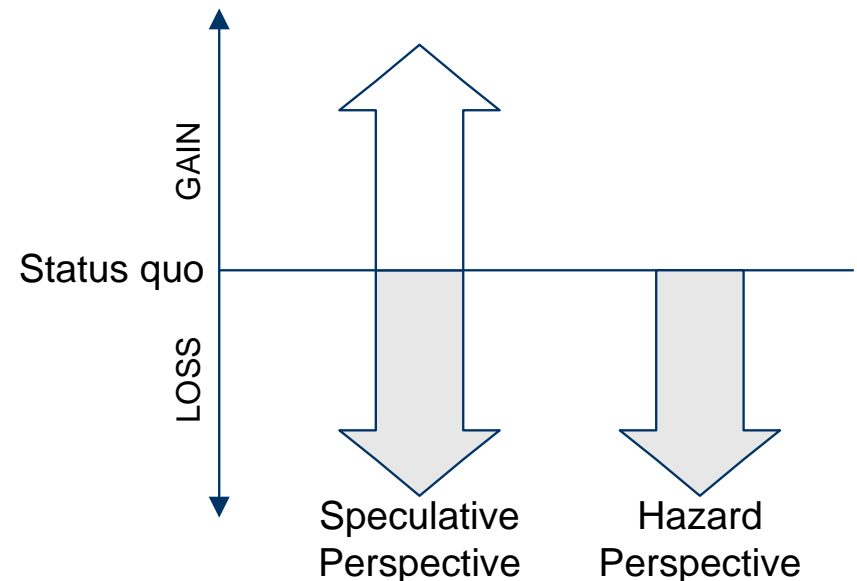
Provides the potential for gain as well as the potential for loss

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

Hazard

Provides no opportunity to improve upon the current situation

Brings only the potential for loss



Widespread Use of Risk Management

Most programs and organizations implement some type of risk management approach when developing and operating software-intensive systems.

- 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
- Increasingly complex management environment
- Confusion among issues, risks, and opportunities

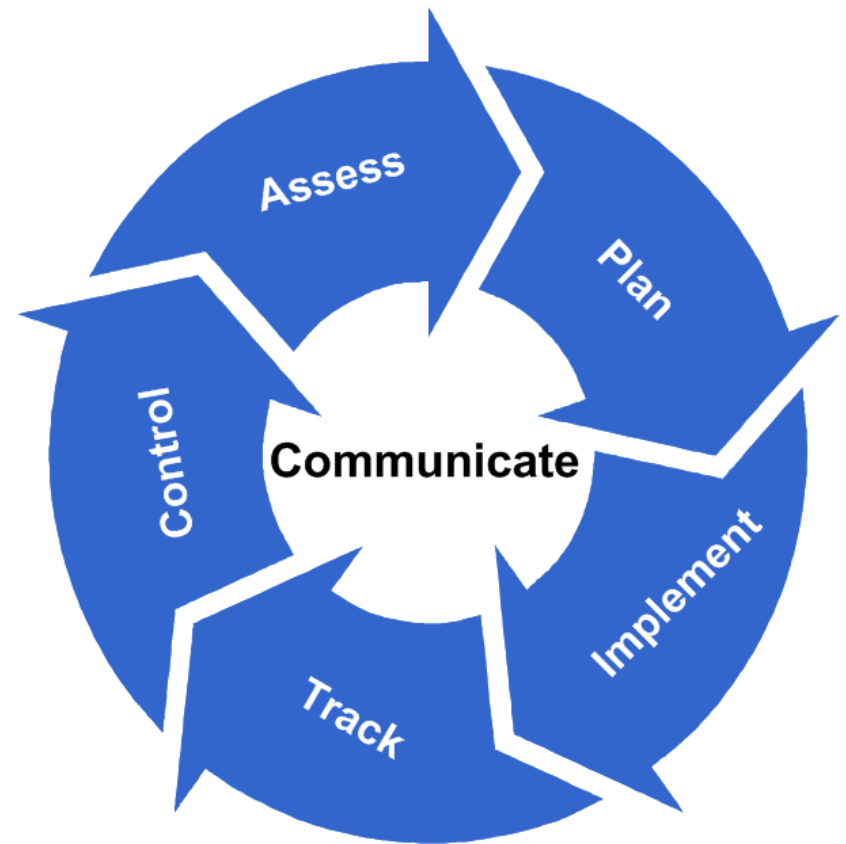


What Is Traditional Risk Management?

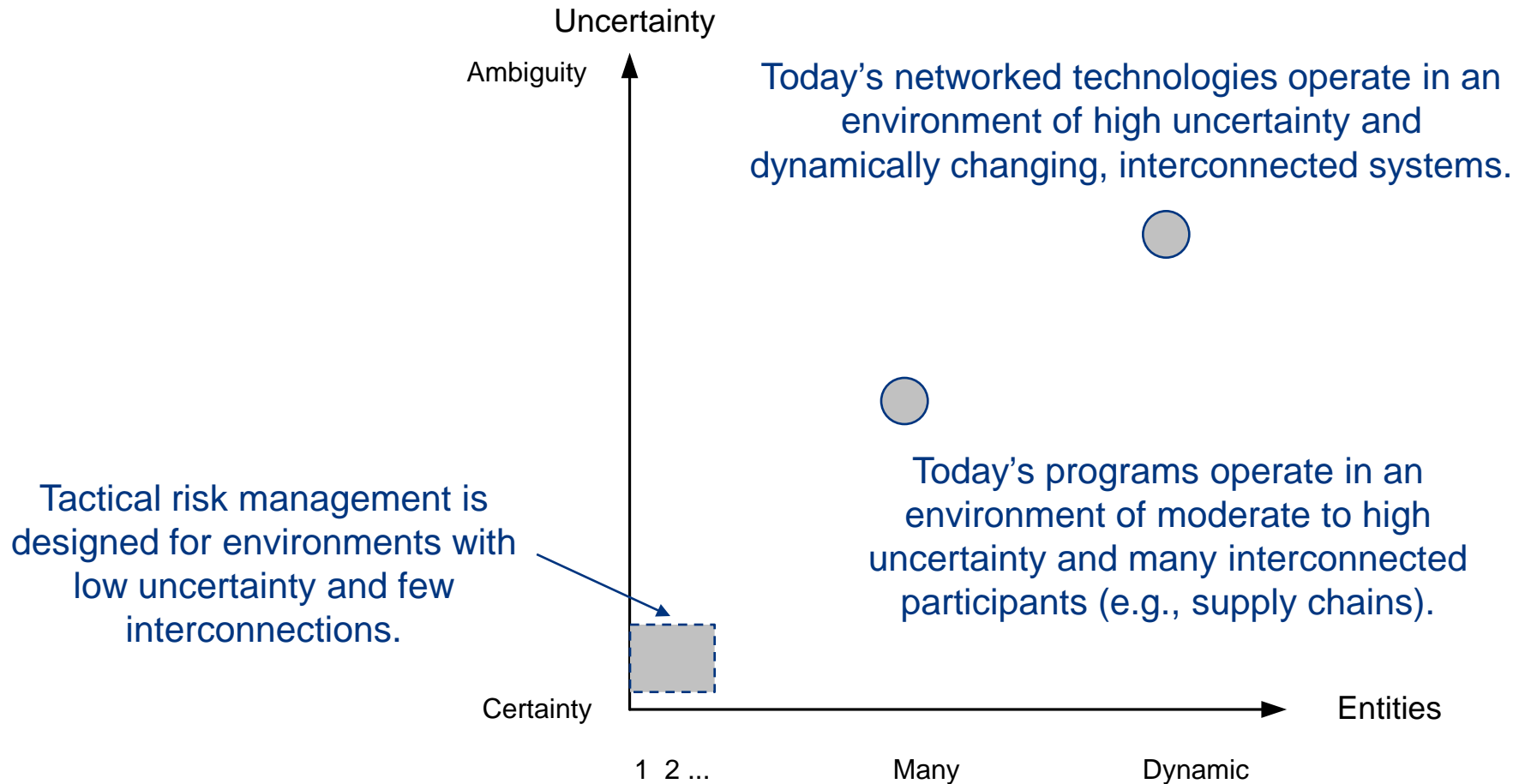
In a systems context, risk management is traditionally viewed as a proactive, disciplined approach for

- 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

Traditional risk management is generally considered to be *tactical* in nature.

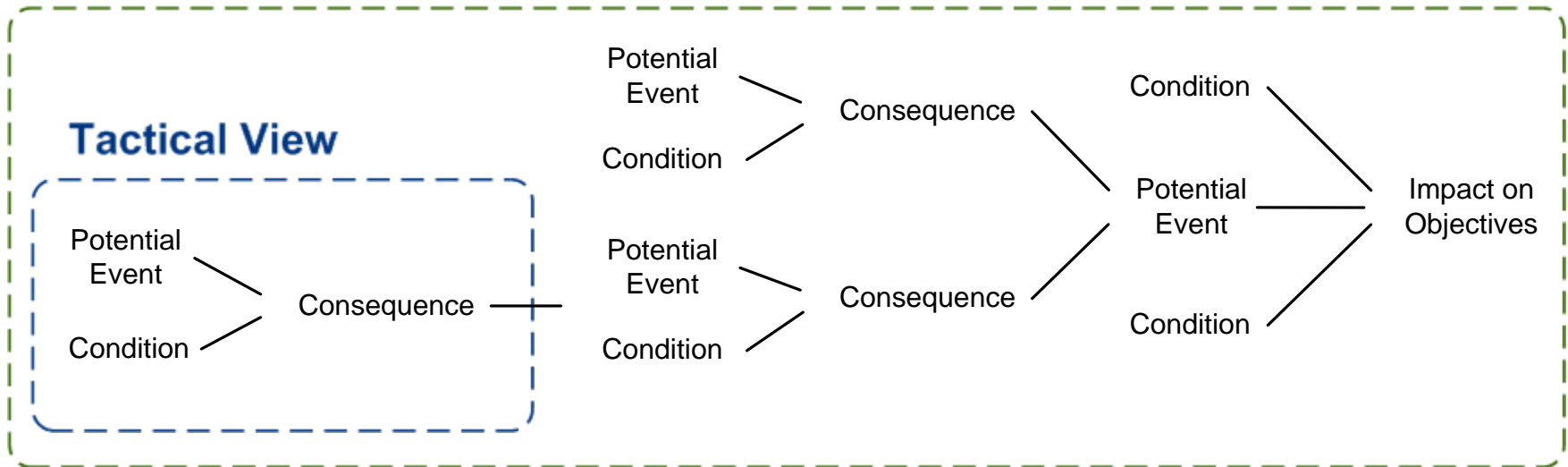


Tactical Risk Management and Complex Environments



Tactical and Systemic Approaches

Systemic View



Tactical Approaches for Analyzing Risk - 1

Have traditionally been used when developing and operating software-intensive systems

View a threat as a potential event that might or might not occur and is focused on the direct consequences of that threat

- Threat directly affects program performance
- The impact on a program's key objectives is an indirect consequence

Employ bottom-up analysis (based on the causes of risk)

Lead to the development of many distinct point solutions, where each is intended to mitigate a specific risk statement



Tactical Approaches for Analyzing Risk - 2

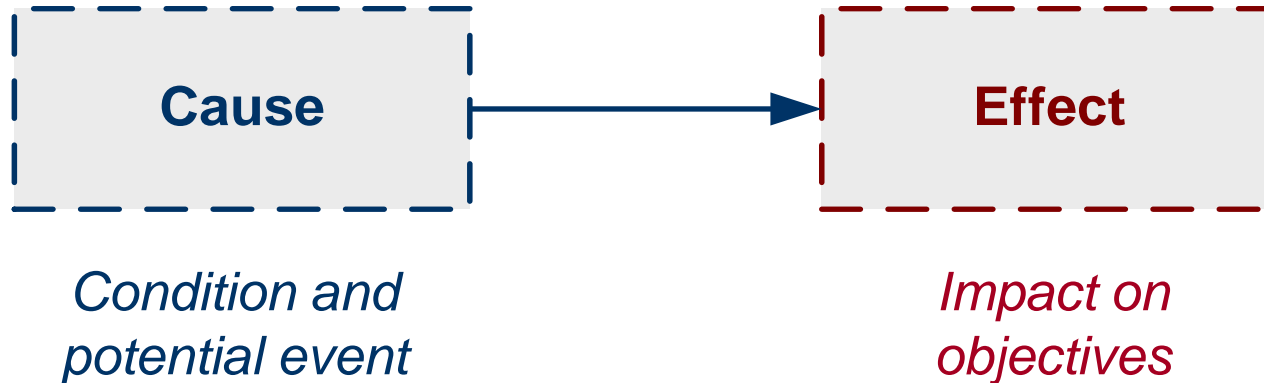
Require a separate statement to be documented for each risk

- Some programs identify hundreds of risk statements.
- Interrelationships and dependencies among conditions and events are not usually established.
- It can be time consuming to aggregate individual risk statements into risk groups.

Implement Pareto analysis to generate a Top N list of risk statements



Tactical Analysis of Risk



Tactical risk analysis views a risk as a simple cause-and-effect pair.

The cause is the combination of a condition and a potential event.

The effect is the impact on objectives.





If-Then Risk Statement

	If	Then
Risk 1	If we miss our next milestone	Then the program will fail to achieve its product, cost, and schedule objectives
Risk 2	If our subcontractor is late in getting their modules completed on time	Then the program's schedule will slip


probability





Condition-Concern Risk Statement

	Condition	Concern
Risk 1	Data indicate that some tasks are behind schedule and staffing levels may be inadequate.	The program could fail to achieve its product, cost, and schedule objectives.
Risk 2	Our subcontractor has not provided much information regarding the status of its tasks.	The program's schedule could slip.


probability

NOTE: Some risk management methods refer to a condition-concern statement as a *condition-consequence* statement





Condition-Event-Consequence Risk Statement

	Condition	Event	Consequence
Risk 1	Data indicate that some tasks are behind schedule and staffing levels may be inadequate.	We could miss our next milestone.	The program will fail to achieve its product, cost, and schedule objectives.
Risk 2	The subcontractor has not provided much information regarding the status of its tasks.	The subcontractor could be late in getting its modules completed on time.	The program's schedule will slip.


probability

NOTE: This is similar to a vulnerability-threat-consequence statement.

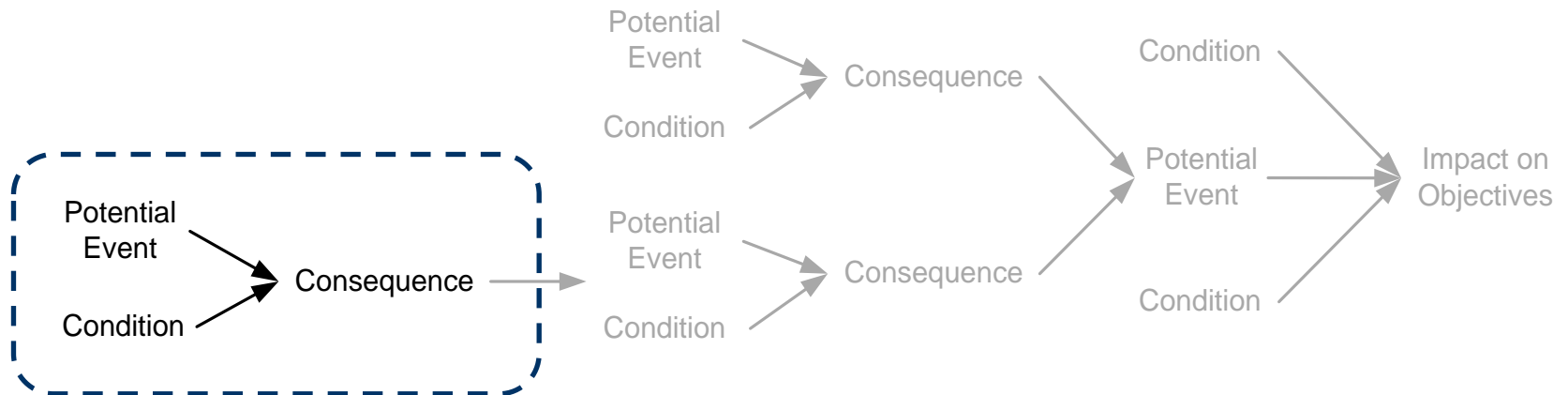


Question: *Risk Statements*

What type of risk statements do you use?



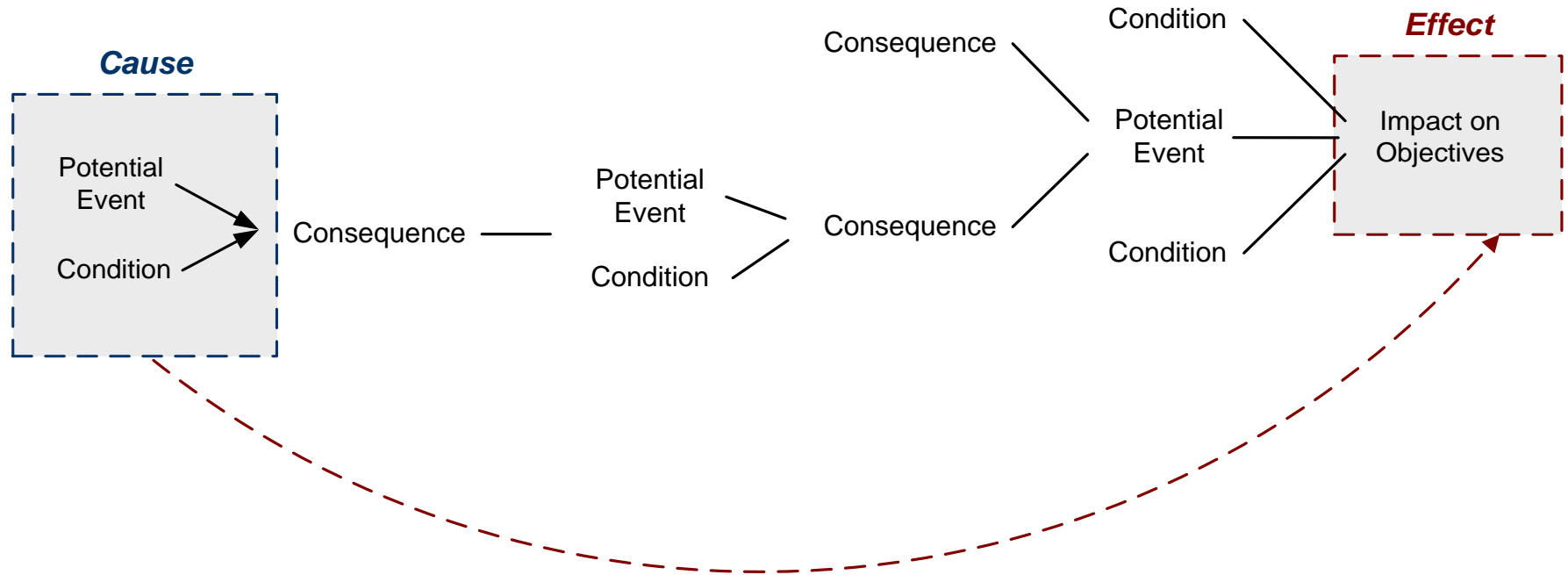
Limitations of Tactical Analysis - 1



The tactical risk can miss the real impact on objectives and lead to localized mitigation efforts.



Limitations of Tactical Analysis - 2



The tactical view assumes a direct connection between a risk's cause and its impact on objectives, which may not be true.

Risk will not be characterized effectively if the connection between a risk's cause and its impact on objectives is **indirect**.



Systemic Approaches for Analyzing Risk

Assume a holistic view of risk to objectives by examining the aggregate effects of multiple conditions and potential events

Employ top-down analysis (based on objectives)

Focus on a small (e.g., 10-20) set of mission* risks (or drivers)

- Enable mapping of multiple root causes to mission risks
- Allow for analysis of interrelationships and dependencies among root causes

Incorporate a system view of risk that is

- Holistic
- Broad-based

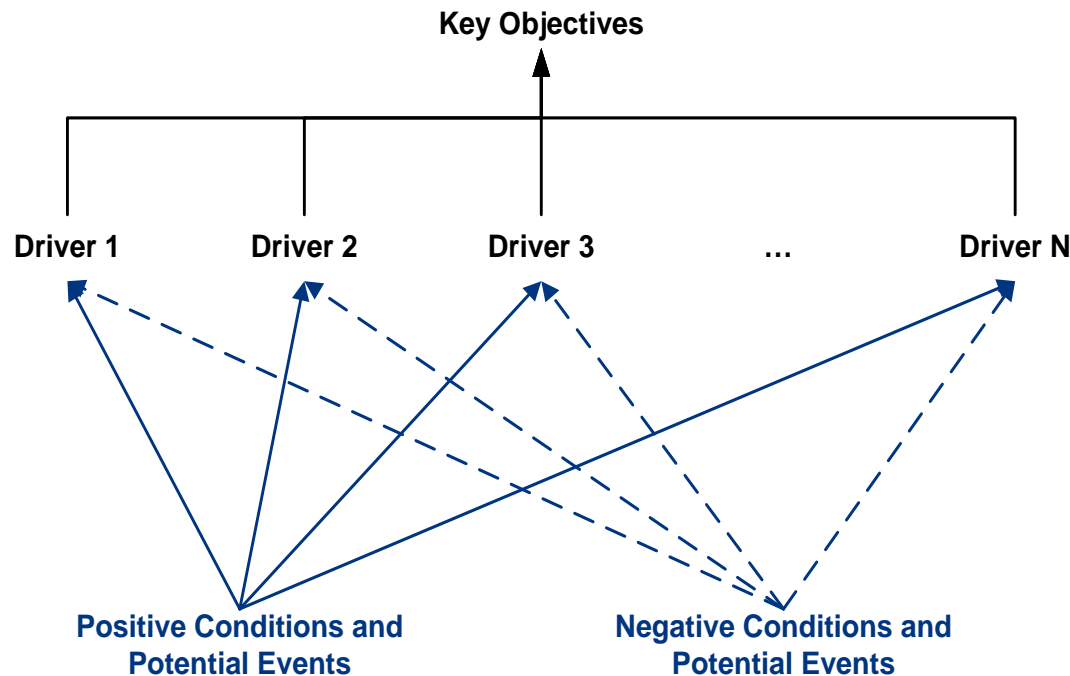
* Systemic and mission risk are used synonymously in this tutorial



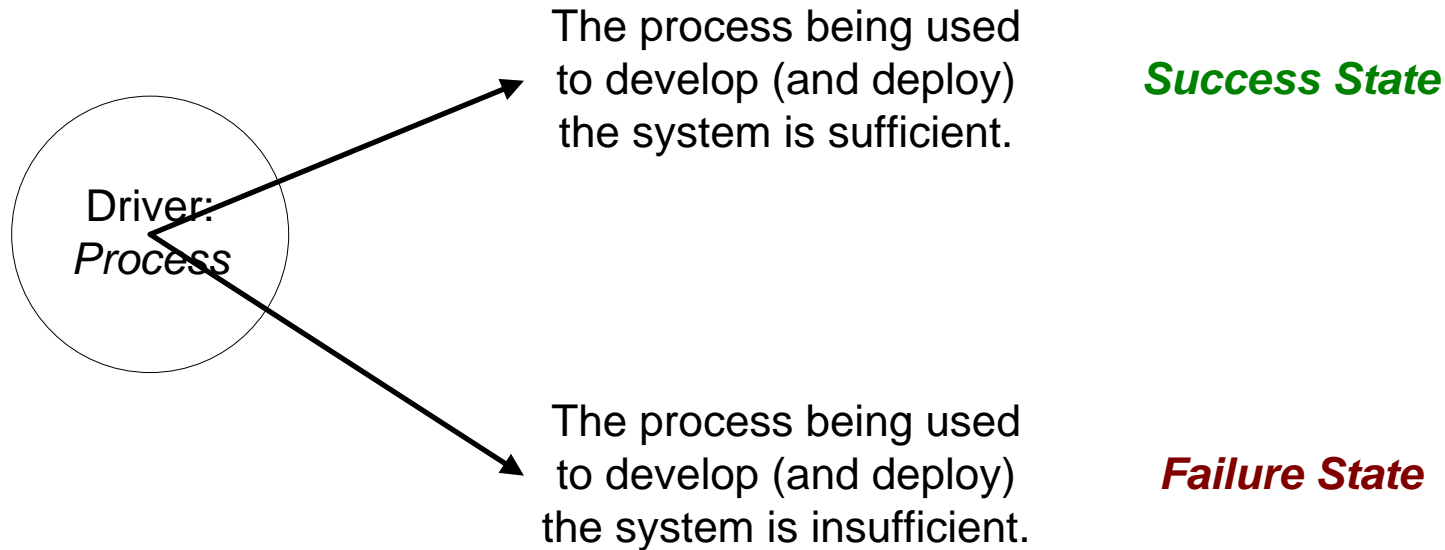
Drivers Aggregate Positive and Negative Aspects

A driver is a factor that has a strong influence on the eventual outcome or result.

Drivers enable a systemic approach to risk management by aggregating the effects of conditions and potential events.



Drivers: *Success and Failure States*

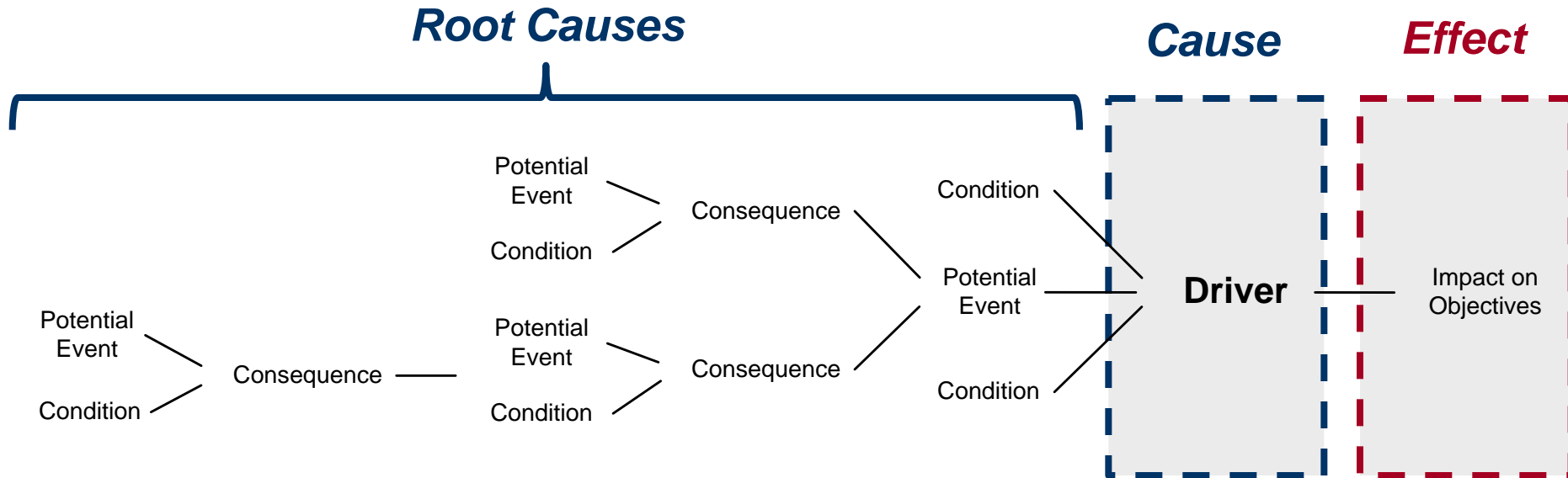


A driver can guide the outcome toward key objectives (success state) or away from them (failure state).

A driver's current state determines whether it is acting as a success or failure driver.



Systemic Analysis of Risk



A driver is a factor that has a strong influence on the eventual outcome or result.

By definition, a driver has a direct connection to the impact on objectives.

Conditions and potential events form the root causes of a systemic risk.





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?



A DIFFERENT PERSPECTIVE



Mosaic

What

A systemic approach for managing risk and uncertainty across the life cycle and supply chain



Why

To provide a risk management approach **that meets the needs of today's complex programs**

Core Technologies

Risk Management Framework

Suite of Assessment Methods

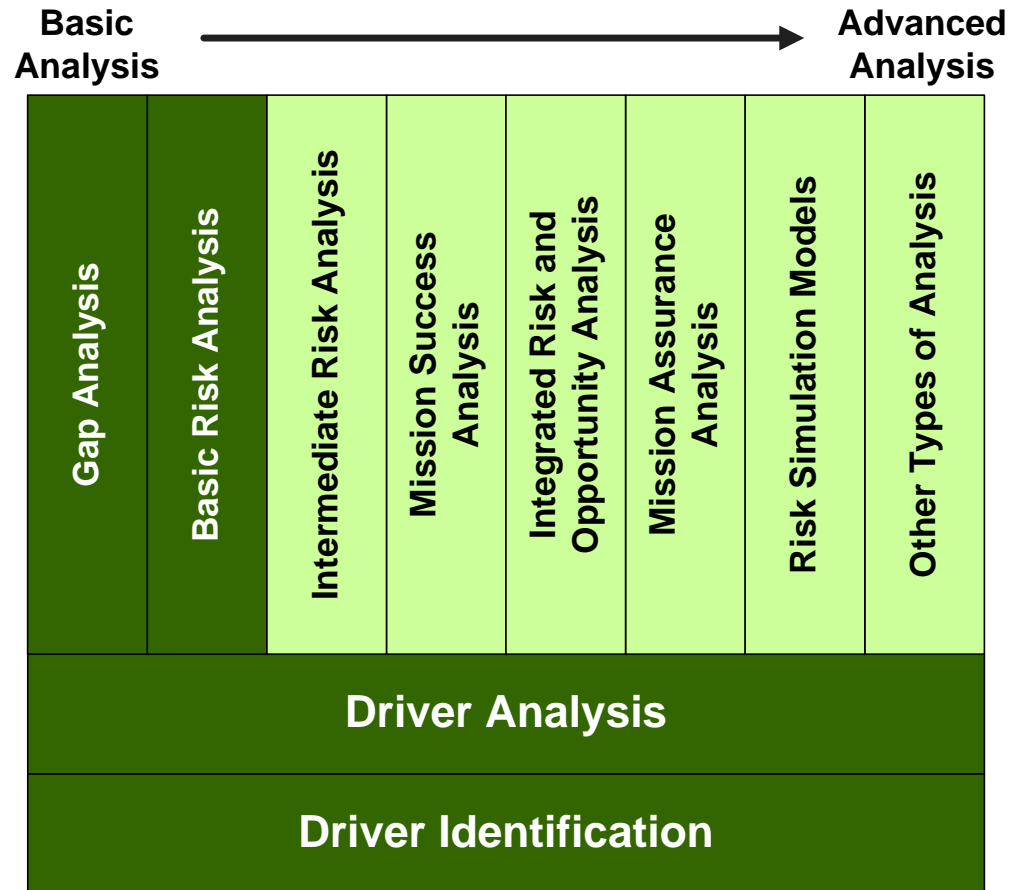


Suite of Assessment Methods on a Common Foundation

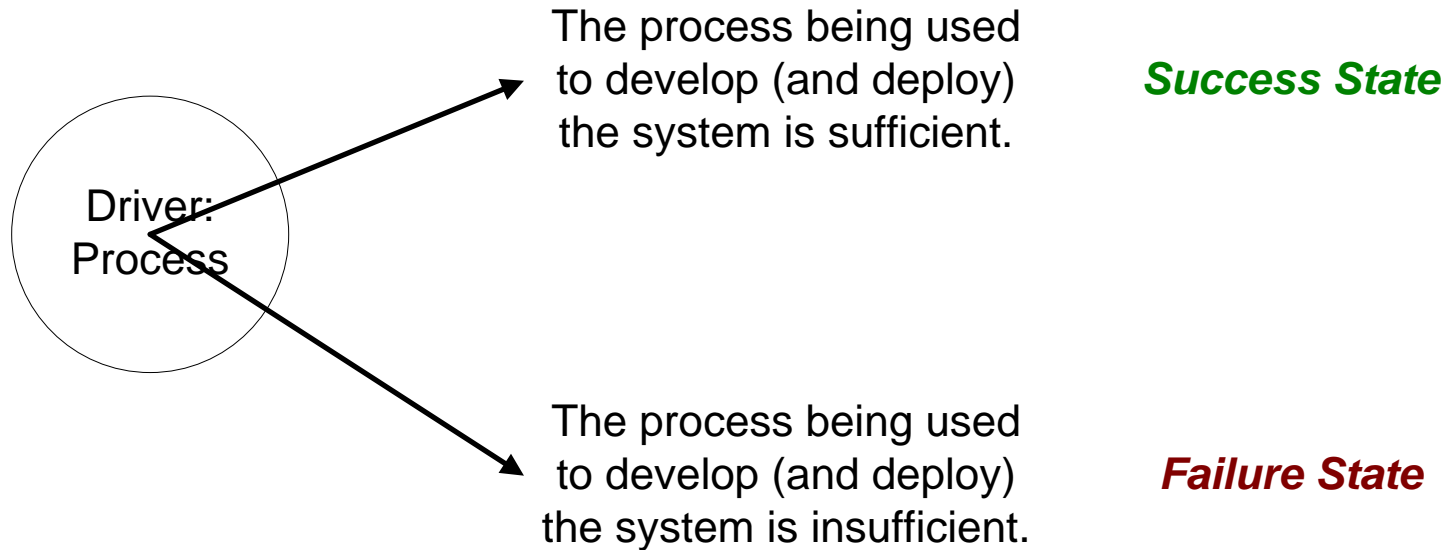
Driver identification and analysis provide a common foundation for multiple back-end analyses

This tutorial will focus on the first two types of assessment

- Gap Analysis: Mission Diagnostic
- Basic Risk Analysis: Risk Diagnostic



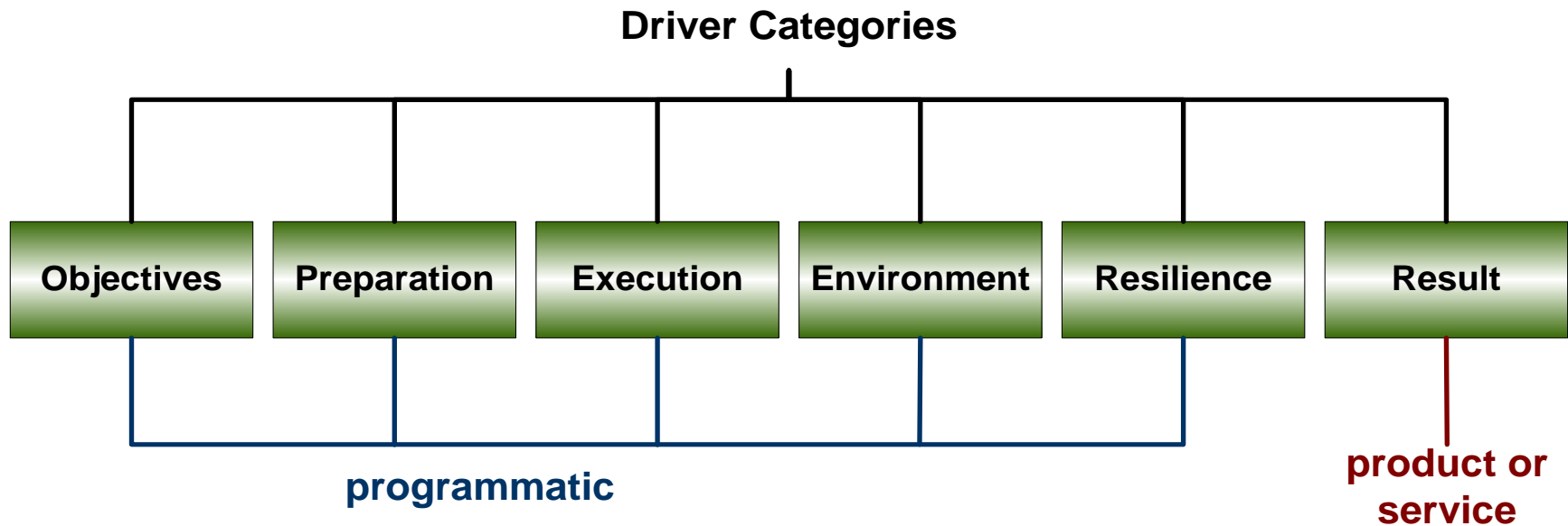
Reminder: *Drivers*



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



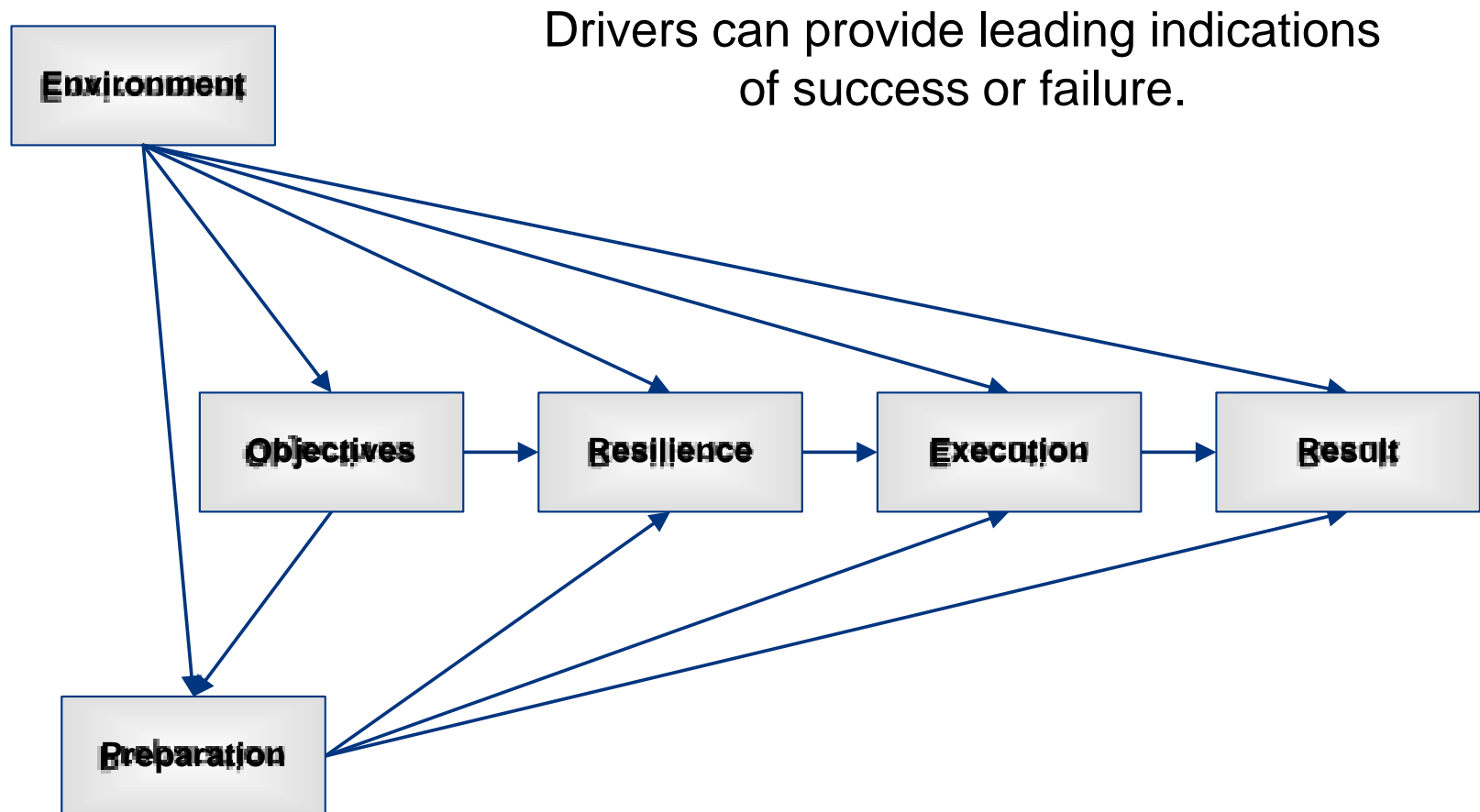
Driver Framework



The driver framework is a common structure for classifying a set of drivers.



Primary Relationships among Driver Categories



Standard Set of Drivers for Software/System Development and Deployment

Objectives

1. Program Objectives

Preparation

2. Plan
3. Process

Execution

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

(Programmatic drivers)

Environment

10. Organizational Conditions
11. Compliance

Resilience

12. Event Management

Result

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



Drivers: *Multiple Format Variations*

Variations for drivers include the following:

- Each driver is embodied in a yes/no question, where each question is phrased from the **success perspective**.
- Each driver is embodied in a yes/no question, where each question is phrased from the **failure perspective**.
- Each driver's **success state** is used as a true/false statement.
- Each driver's **failure state** is used as a true/false statement.

For the Mission Diagnostic, we convert drivers into yes/no questions that are phrased from the success perspective.

For Risk Diagnostic, we use the failure state as a true/false statement and you determine the probability that the failure state exists.



Driver 1: *Program Objectives*

Are program objectives (product, cost, schedule) realistic and achievable?

Considerations:

- Alignment of technical, cost, and schedule objectives
- Inherent technical risk
- Technology maturity
- Resources available



Driver 2: *Plan*

Is the plan for developing and deploying the system sufficient?

Considerations:

- Acquisition or development strategy
- Program plan
- Resources
- Funding
- Schedule
- Roles and responsibilities



Driver 3: *Process*

Is the process being used to develop and deploy the system sufficient?

Considerations:

- Process design
- Measurements and controls
- Process efficiency and effectiveness
- Acquisition and development life cycles
- Training



Driver 4: *Task Execution*

Are tasks and activities performed effectively and efficiently?

Considerations:

- Experience and expertise of management and staff
- Staffing levels
- Experience with the acquisition and development life cycles



Driver 5: *Coordination*

Are activities within each team and across teams coordinated appropriately?

Considerations:

- Communication
- Information sharing
- Dependencies
- Relationships
- Partners and collaborators



Driver 6: *External Interfaces*

Will work products from suppliers, partners, or collaborators meet the program's quality and timeliness requirements?

Considerations:

- Applications
- Software
- Systems or sub-systems
- Hardware



Driver 7: *Information Management*

Is the program's information managed appropriately?

Considerations:

- Usability
- Confidentiality
- Integrity
- Availability



Driver 8: *Technology*

Does the program team have the tools and technologies it needs to develop the system and transition it to operations?

Considerations:

- Software applications
- Infrastructure
- Systems
- Databases



Driver 9: *Facilities and Equipment*

Are facilities and equipment sufficient to support the program?

Considerations:

- Building
- Physical work spaces
- Support equipment
- Supplies
- Other resources



Driver 10: *Organizational Conditions*

Are enterprise, organizational, and political conditions facilitating completion of program activities?

Considerations:

- Stakeholder sponsorship
- Actions of upper management
- Effect of laws, regulations, and policies



Driver 11: *Compliance*

Does the program comply with all relevant policies, laws, and regulations?

Considerations:

- Policies
- Laws
- Regulations
- Standards of care



Driver 12: *Event Management*

Does the program have sufficient capacity and capability to identify and manage potential events and changing circumstances?

Considerations:

- Risk management plan, process, and tools
- Schedule slack
- Funding reserve
- Risk mitigation plans
- Program continuity and contingency plans
- Opportunity management plan, process, and tools



Driver 13: *Requirements*

Are system requirements well understood?

Considerations:

- Customer, user, and stakeholder requirements and needs
- Functional and non-functional requirements
- Operational requirements
- System growth and expansion needs
- Technology maturity



Driver 14: *Design and Architecture*

Are the design and architecture sufficient to meet system requirements and provide the desired operational capability?

Considerations:

- Interfaces
- Dependencies
- Software and system architecture
- Operational requirements
- Technology maturity



Driver 15: *System Capability*

Will the system satisfactorily meet its requirements?

Considerations:

- Functional
- Performance
- Operational
- Reliability
- Security
- Safety
- Usability
- Maintainability
- Technology maturity



Driver 16: *System Integration*

Will the system sufficiently integrate and interoperate with other systems when deployed?

Considerations:

- Interfaces
- Applications
- Tools
- Hardware
- Data
- Technology maturity



Driver 17: *Operational Support*

Will the system effectively support operations?

Considerations:

- Business and operational workflows
- Support of organizational and enterprise missions
- Operational risk mitigation
- Disaster recovery, contingency and business continuity plans
- Technology maturity



Driver 18: *Adoption Barriers*

Have barriers to customer/user adoption of the system been managed appropriately?

Considerations:

- User acceptance
- Stakeholder sponsorship
- Transition to operations
- User support



Driver 19: *Operational Preparedness*

Will people be prepared to operate, use, and maintain the system?

Considerations:

- Policies
- Procedures
- Training



Driver 20: *Certification and Accreditation*

Will the system be appropriately certified and accredited for operational use?

Considerations:

- Compliance with policies, laws, and regulations
- Acceptable mitigation of risk





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?



THE MISSION DIAGNOSTIC





What Is a Mission?

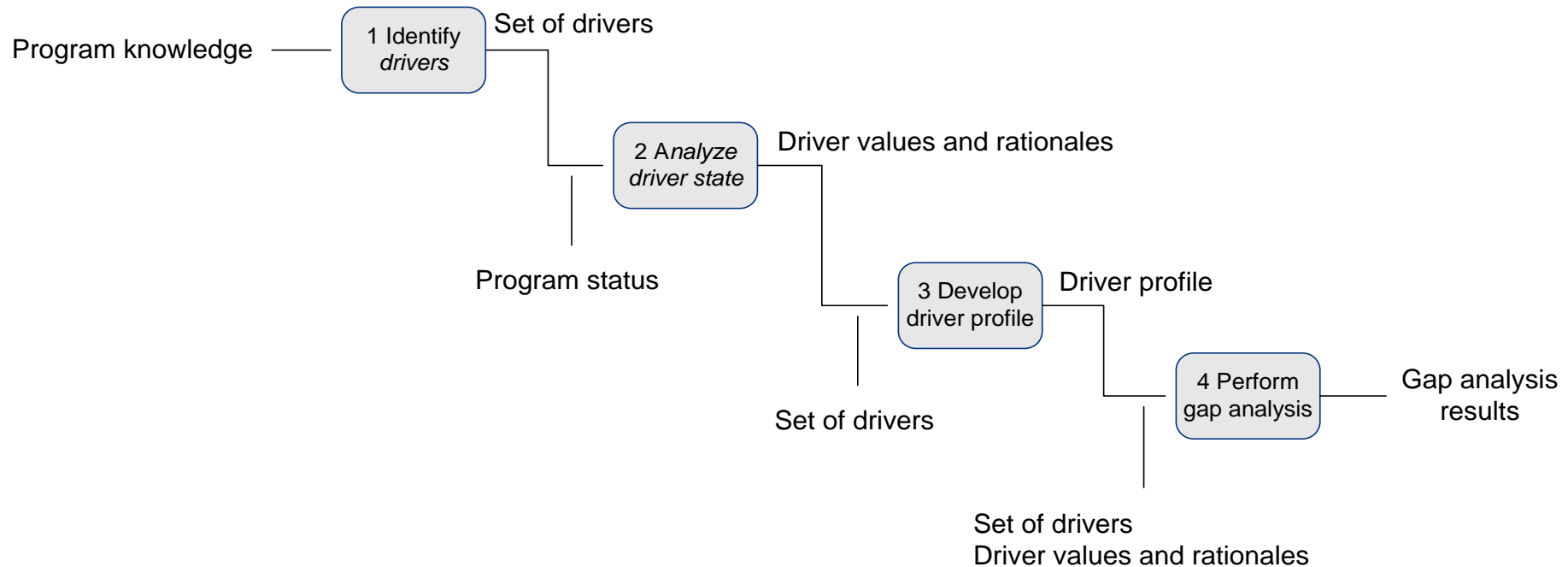
The term ***mission*** has multiple meanings, depending on the context in which it is used.

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



Core Mission Diagnostic Activities





Who Performs the Mission Diagnostic?

External, independent team:

- Outside the organization
- Provide unbiased results
- Will need to gather considerable data

Internal, independent team

- Inside the organization
- Provide unbiased results
- Will likely need to gather less data than an external team

Project team:

- Part of the project
- For routine, frequent applications
- Will need to gather considerably less data, if any



The Mission Diagnostic **WHAT SET OF DRIVERS?**



Identifying Drivers: *Two Basic Steps*

Establish key objectives

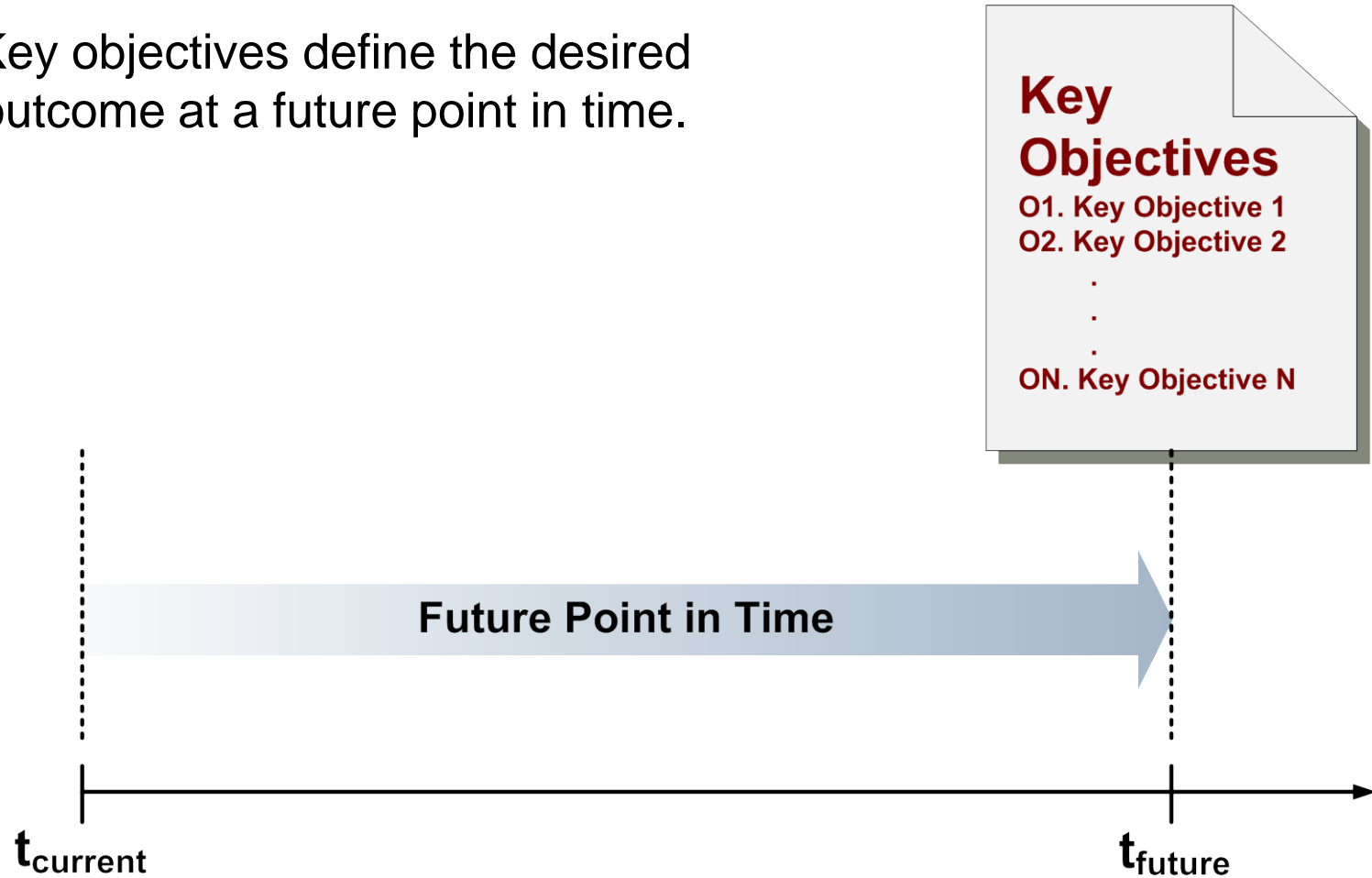
Identify set of drivers to use

- Deriving a set of drivers
- Tailoring a set of drivers



Establishing Key Objectives

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





Types of Key Program Objectives

Product Objectives

- Define the nature of the products produced (or services provided)
- For software-intensive systems, the product (i.e., technical) objectives minimally define

Cost Objectives

- Define the budget allocated to developing a product (or providing a service)

Schedule Objectives

- Define the time period allocate to developing a product (or providing a service)

Other Objectives

- Define additional goals of a program, e.g., business, financial, or compliance



Identifying Drivers: *Deriving Drivers*

To establish a set of drivers for specific objectives, talk to people with ***experience*** and ***expertise*** relevant to those objectives.

Ask the experts the following types of questions:

- What circumstances, conditions, and events will drive your program toward a *successful* outcome?
- What circumstances, conditions, and events will driver your program toward a *failed* outcome?

Organize the information they provided (i.e., circumstances, conditions, and events) into approximately 10-20 groups that share a central idea or theme.

- The driver is the central idea or theme of each group.
- Make sure to include at least one driver for each of the six driver categories



Identifying Drivers: *Tailoring* - 1

Select a predefined set of drivers consistent with the program's key objectives to use as the basis for tailoring.

Meet with management and staff from the program to

- Learn about what the program is trying to accomplish
- Gain an appreciation for its unique context and characteristics



Identifying Drivers: *Tailoring* - 2

Based on the program's key objectives and the data that you have gathered

- Determine which drivers do not apply to the program; eliminate extraneous drivers from the set.
- Establish whether any drivers are missing from the list; add those drivers to the set.
- Decide if multiple drivers from the set should be combined into a single, high-level driver; replace those drivers with a single driver that combines them.
- Decide if any drivers should be decomposed into multiple, more detailed drivers; recompose each of those drivers into multiple drivers.

Adjust the wording of each driver attribute to be consistent with the program's terminology and language.



Questions: *Tailoring Drivers*

For the starter set of drivers, consider the following questions:

- Which drivers would you decompose? Why?
- Which drivers would you consolidate? Why?
- For which drivers would you change the wording? Why?



Standard Set of Drivers for Software/System Development

Objectives

1. Program Objectives

Preparation

2. Plan
3. Process

Execution

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

(Programmatic drivers)

Environment

10. Organizational Conditions
11. Compliance

Resilience

12. Event Management

Result

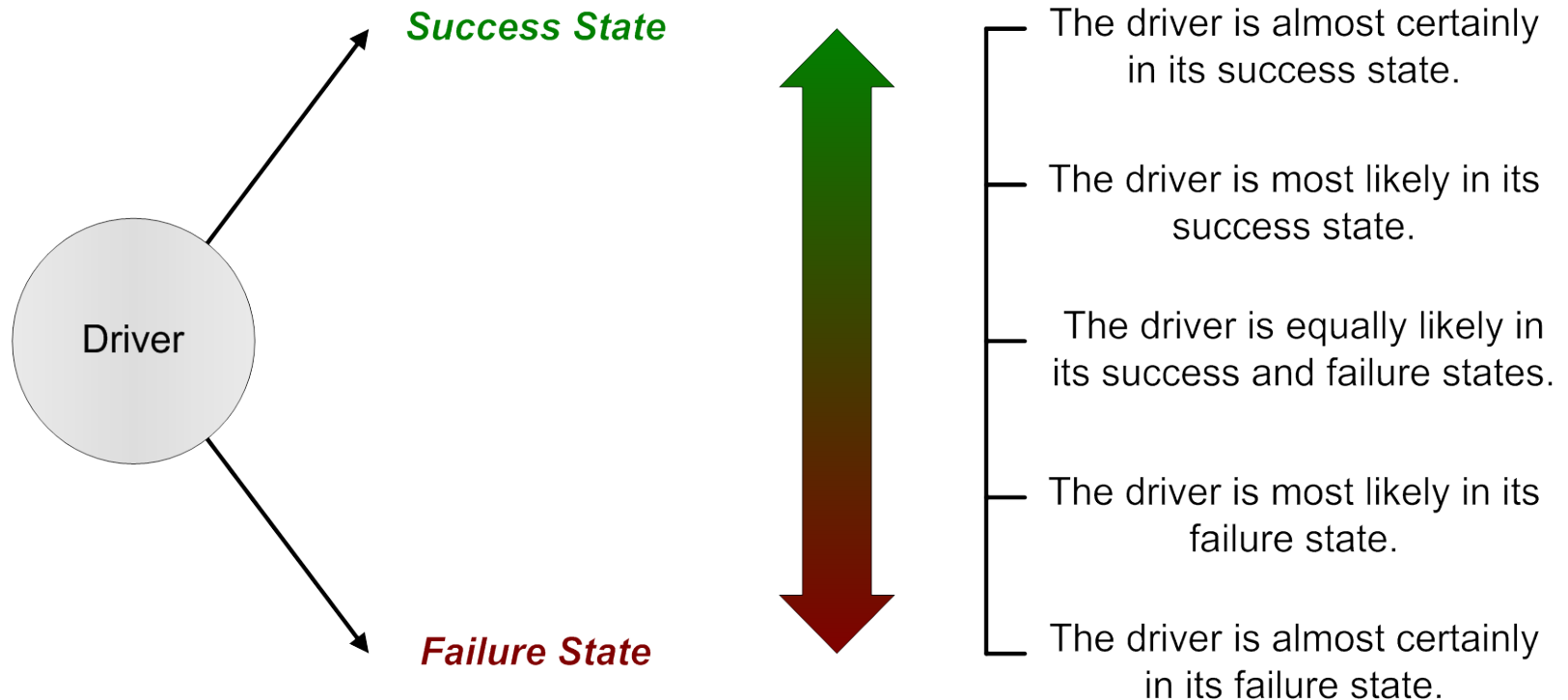
13. Requirements
14. Design and Architecture
15. System Capability
16. System Integration
17. Operational Support
18. Adoption Barriers
19. Operational Preparedness
20. Certification and Accreditation



The Mission Diagnostic **ANALYZING DRIVERS**



Analyzing Driver State



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



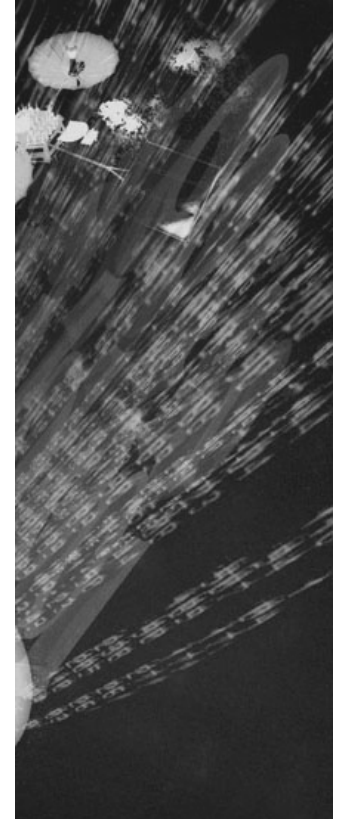
Collect Information

To analyze 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



Data Collection: *Obtaining Status Information*

When analyzing drivers, you need information about the program's current status.

- Positive conditions (i.e., what is working well)
- Potential events that could improve program performance
- Negative conditions (i.e., what is *not* working well)
- Negative events that could degrade program performance

Sometimes separate tasks are required to get sufficient information about a program's current status.

- External team with little knowledge of the program
- Internal, independent team with minimal knowledge of program
- Program team with need to supplement their knowledge



Data Collection: *Techniques for Obtaining Status Information*

Two main techniques are used to obtain status information.

- Gather data from people
- Generate data from documentation

There is usually some connection and iteration between these two activities.

- The organization chart and overall program information is used to explain the nature of the program and identify good candidates for interviews
- Documentation reviews can identify additional groups of people to interview
- Interviews can identify additional documents to collect

Another technique, *observe task execution*, is used in some cases to acquire information about actual performance of specific, key tasks or activities.



Data Collection: *Candidate People*

Status information can be gathered from people who perform program activities, such as

- Managers
- Programmers
- Customers
- Contractors and partner organizations
- Staff responsible for the infrastructure
- Staff responsible for training
- Other relevant groups (e.g., human resources, legal, contracting)



Data Collection: *Techniques for Gathering Data from People*

Technique	Description	Use When
Workshops	Facilitated session with groups of people who work together	Need a less structured format to encourage more free form discussion or to encourage discussion of previously unidentified topics.
Interviews	Facilitated session where participants answer a series of specific questions asked by one or more interviewers	Have structured set of questions and a finite amount of time. Need formal structure to control the process of getting data.
Surveys	Electronic or paper-based surveys are distributed and collected, with or without any follow-on discussion	Need to quickly gather data from large number of people. Surveys are very clear and not subject to misinterpretation.



Data Collection: *Generating Data From Documentation*

A comprehensive review of documents can be used to obtain information about a program's current status to supplement or verify the information gathered from people.

The nature of the documentation reviewed depends upon the

- Specific program being assessed
- Objectives of the program
- Scope of the assessment

Normally, a small team of experienced people reviews documents and records relevant status information.



Example: *Program Documents*

Document Type	Document Type
Plans, such as program plan, deployment plan, integration plan, testing plan, contingency plan <ul style="list-style-type: none">• Tasks• Budget• Schedule• Roles and responsibilities	Requirements specifications for <ul style="list-style-type: none">• Software and system• Interfaces to other applications, infrastructure, databases• Supporting infrastructure and technologies
Design and architecture documentation	User guides
Training materials for users, operators, maintainers, installers, etc.	Procedures for installation, maintenance, use, etc.



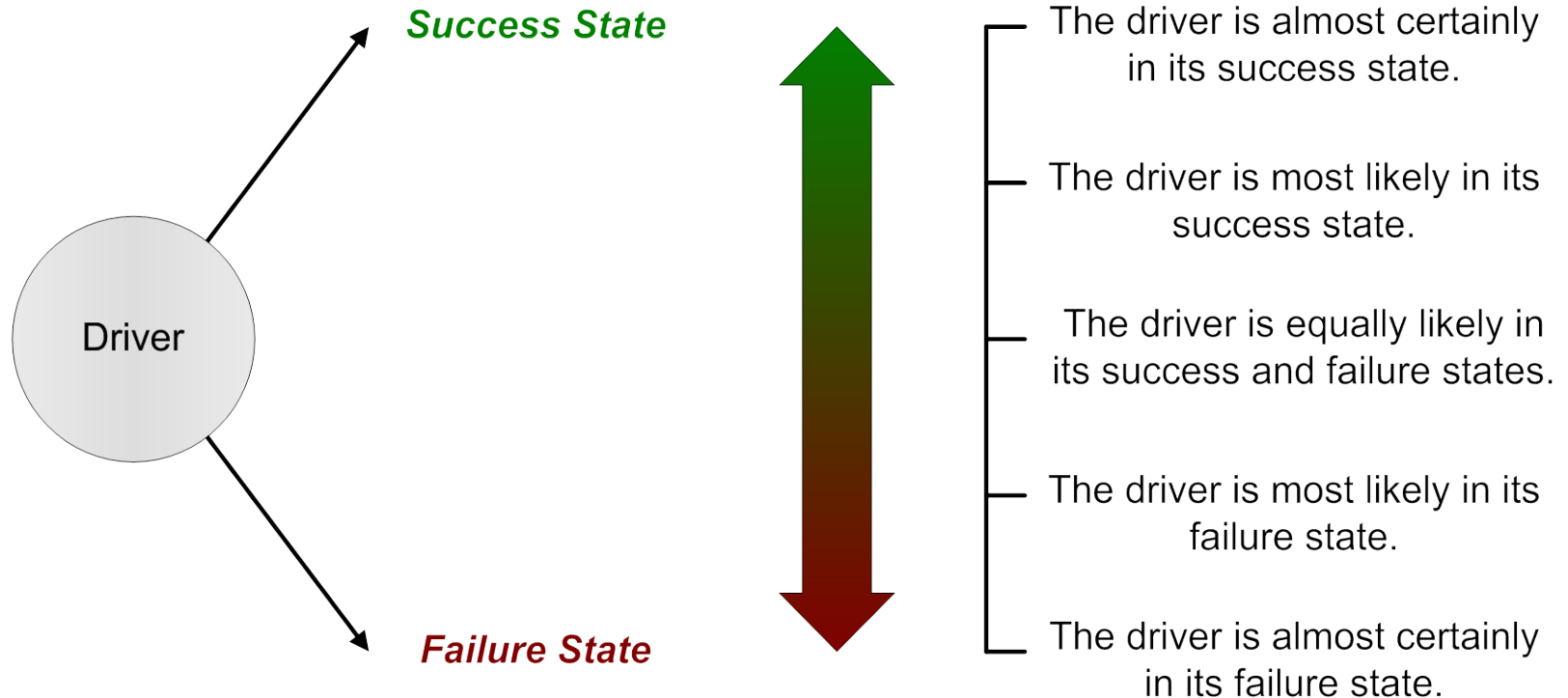
Data Collection: *Techniques for Generating Data from Documentation*

Technique	Description	Strategies
Document Identification	Gather written information, such as policies, procedures, reports, and work products	Ask for all documentation Ask for a focused list of documents
Document Analysis	Analyze the gathered information to transform raw, unfiltered information into data that are usable during the assessment	Have a set of questions or focal points to guide analysis Use expertise and experience to find relevant data*

* In practice, both of these techniques are generally used together.



Use Data to Analyze Driver State



Example: *Driver Question*

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?	<input type="checkbox"/> Yes
	<input type="checkbox"/> Likely Yes
	<input type="checkbox"/> Equally Likely
	<input type="checkbox"/> Likely No
	<input type="checkbox"/> No
	<input type="checkbox"/> Don't Know
	<input type="checkbox"/> Not Evaluated

Consider:

Process design; measurements and controls; process efficiency and effectiveness; acquisition and development life cycles; training

These driver questions are phrased from the success perspective.
Probability is incorporated into the range of answers for each driver.



Example: *Driver Value Criteria*

Each driver is evaluated against predefined criteria.

Response	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)
Don’t know	More information is needed to answer the question.
Not Evaluated	The driver question is not relevant at this point in time. It was not evaluated.



Example: *Evaluating Drivers*

Directions: Select the appropriate response to the driver question.

Driver Question	Response
<p>3. Is the process being used to develop and deploy the system sufficient?</p> <p><i>Consider:</i></p> <p>Process design; measurements and controls; process efficiency and effectiveness; acquisition and development life cycles; training</p>	<p><input type="checkbox"/> Yes</p> <p><input type="checkbox"/> Likely Yes</p> <p><input type="checkbox"/> Equally Likely</p> <p><input checked="" type="checkbox"/> Likely No</p> <p><input type="checkbox"/> No</p> <p><input type="checkbox"/> Don't Know</p> <p><input type="checkbox"/> Not Evaluated</p>

Each driver is evaluated using information about the program's current status.



Documenting Rationale for Driver State

You must document the reasons underlying the analysis of each driver.

- Conditions that support an answer of **yes**
- Conditions that support an answer of **no**
- Potential events that support an answer of **yes**
- Potential events that support an answer of **no**
- Gaps in information that is available for driver analysis
- Any assumptions that have been made



Example: *Rationale for Driver Value - 1*

Driver Question	Driver Value
3. Is the process being used to develop and deploy the system sufficient?	Likely no

Rationale

Previous programs have a 90% history of delivering on-time. (+)

The process for integration testing is likely inadequate. Historically, integration testing has used “verbal” agreements between a few managers who already know each other. With this system, there are managers and team leads who have never worked together and there are other barriers in place that make “verbal” agreements tenuous. (-)





Example: *Rationale for Driver Value - 2*

Rationale (cont.)

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. (-)

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



Option: Driver Weights

Beyond the basic driver analysis, you can also consider how important the driver is to meeting program objectives

- *Critical*
- *High*
- *Medium*
- *Low*
- *Minimal*

Drivers are considered to be essential to the success of the mission, therefore the starting point is that all drivers are weighted as *Critical*

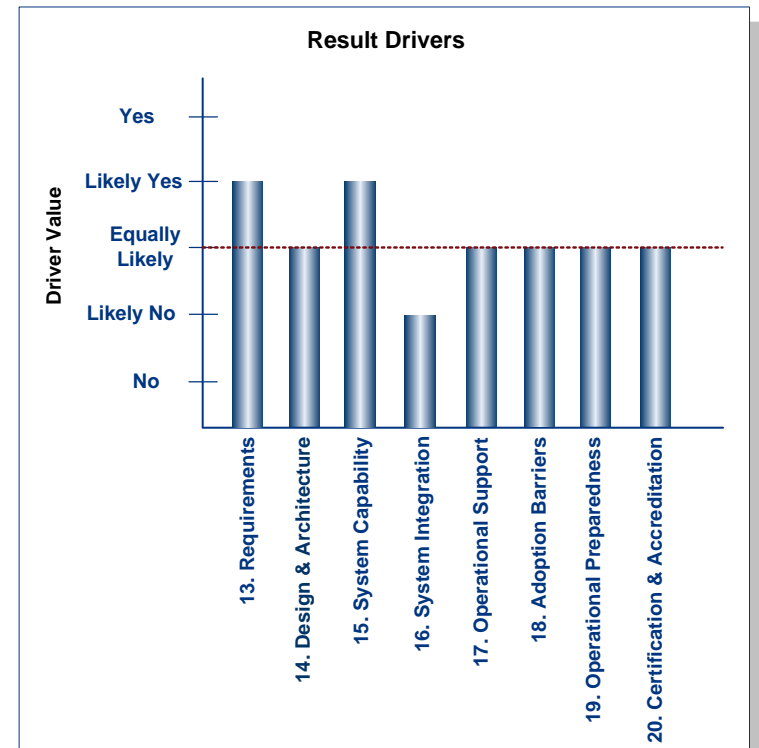
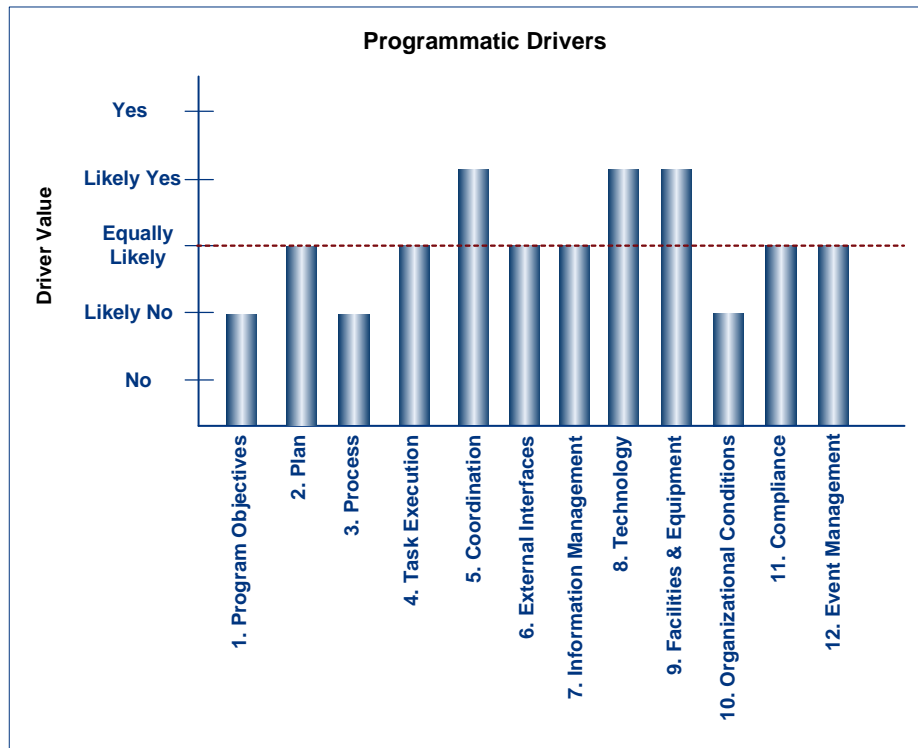
Drivers may increase or decrease their importance or weight depending on where you are in the program life cycle



The Mission Diagnostic **DRIVER PROFILE**



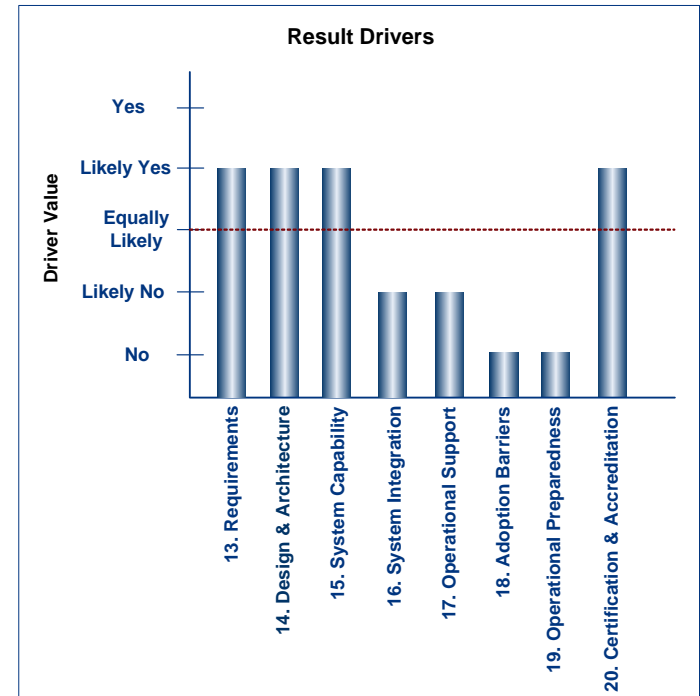
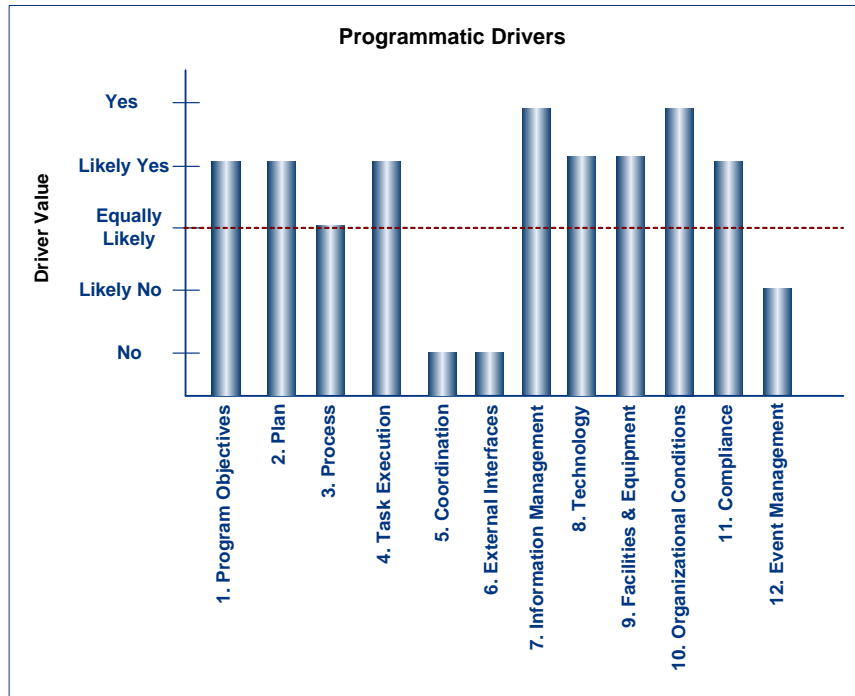
Example: *Driver Profile - 1*



While this simple profile at first glance appears to show roughly equivalent momentum towards success and failure, notice which drivers are failure drivers.



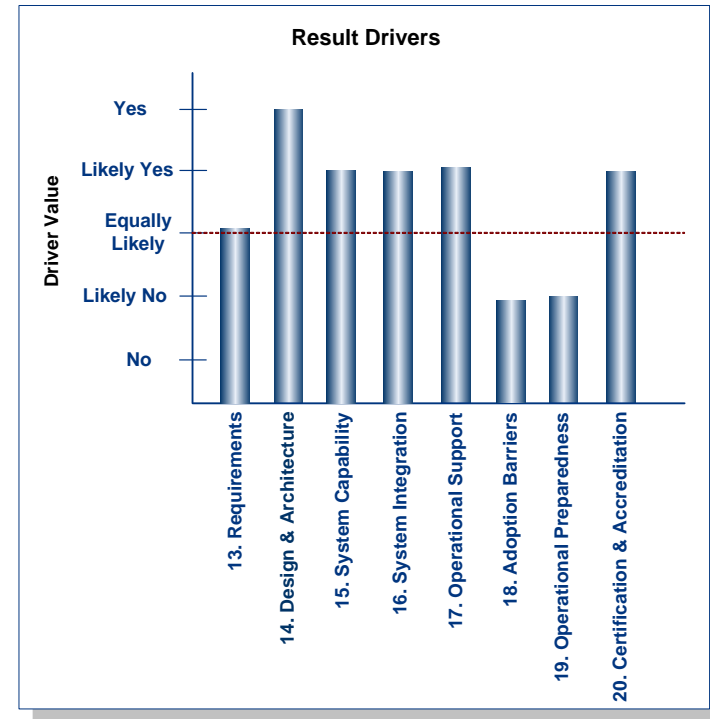
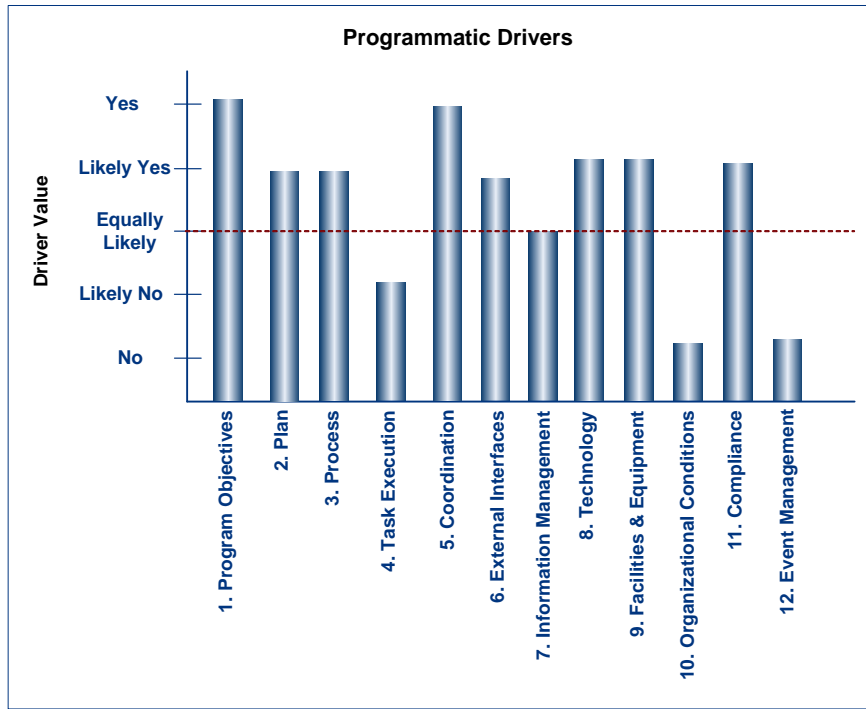
Example: *Driver Profile - 2*



Here, it looks like the program has severe issues in coordinating and working with all external groups. This is leading to issues with integration and preparing the end users and operators to accept and use the system.



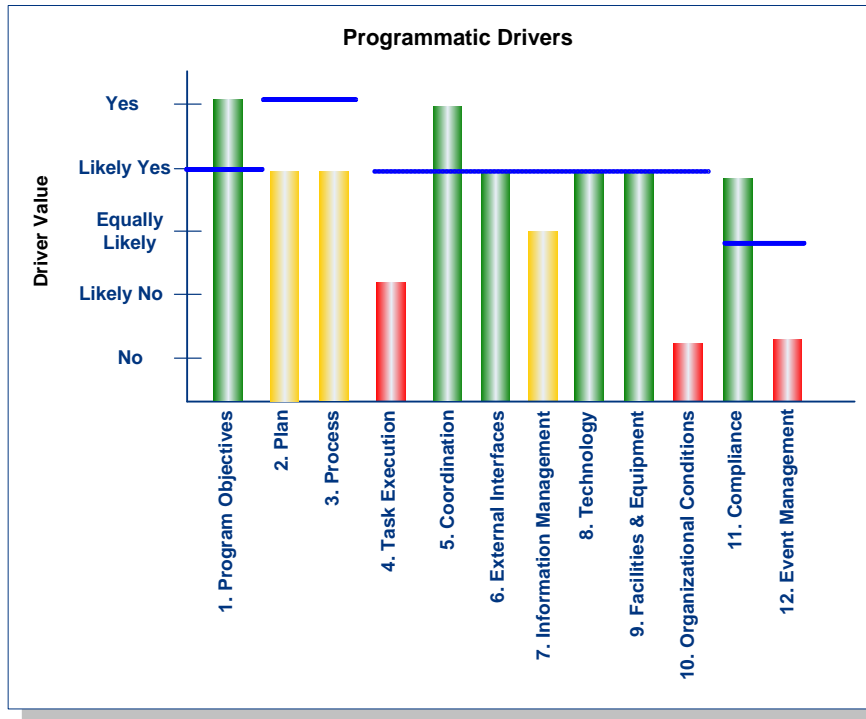
Example: *Driver Profile - 3*



Most of this program is going well. Trouble spots are with excessive sponsor interference in 10, which has caused a lot of rework, affecting task execution and decreased the reserves (12). This particular sponsor is the user, thus the low confidence in being able to prepare users and system maintainers.



Example: *Driver Profile - 4*



This profile also shows the relationship of current value to desired value, depicted by the blue lines. Management now needs to consider whether or not their expectations also need to be adjusted as improvements are planned.



Potential for Success

The potential for success is the likelihood that key objectives will be achieved

An additional analysis of the drivers is used to establish the current potential for success

- Simple aggregation of driver values
- Weighted aggregation of driver values (using driver weights)
- Mean or median driver value
- Rule-based algorithms

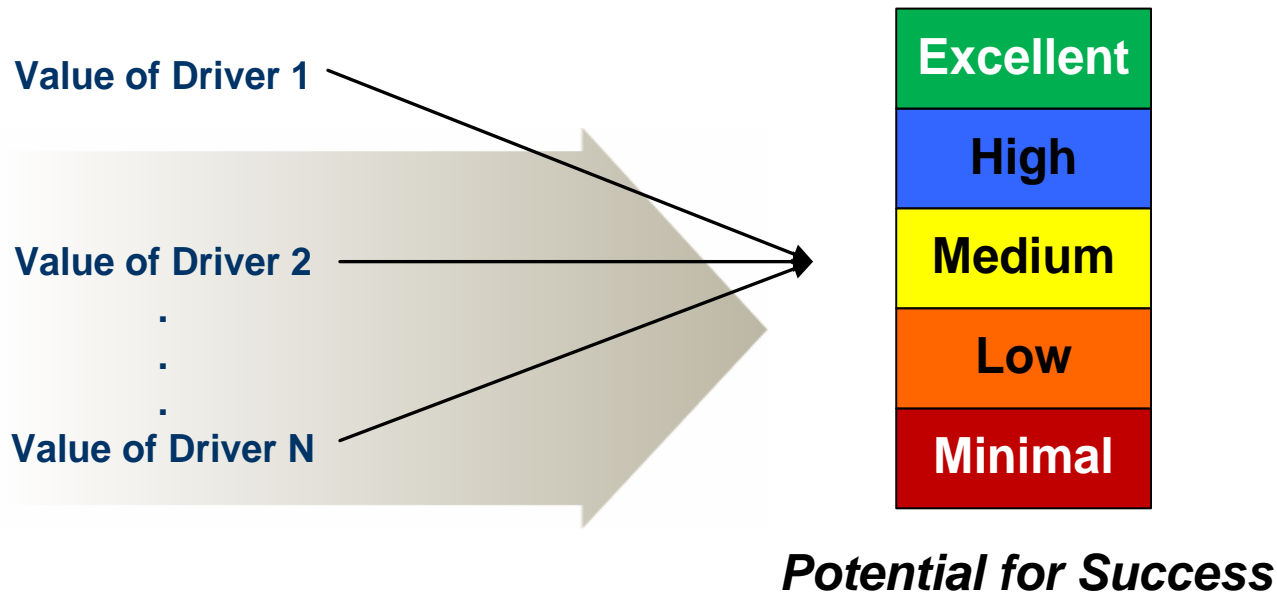


Example: *Success Criteria*

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)



Example: *Potential for Success*



An analysis of drivers is used to determine the *current potential for success* for meeting key objectives



Example: *Potential for Success*

Objectives: 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 these objectives is **Low**

Rationale

- Several drivers with *Critical* weight had values of *Likely No*
- 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.
-



Mission Diagnostic: *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



The Mission Diagnostic **YOUR PROGRAM**



Exercise Three: *Evaluate Your Program*

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:

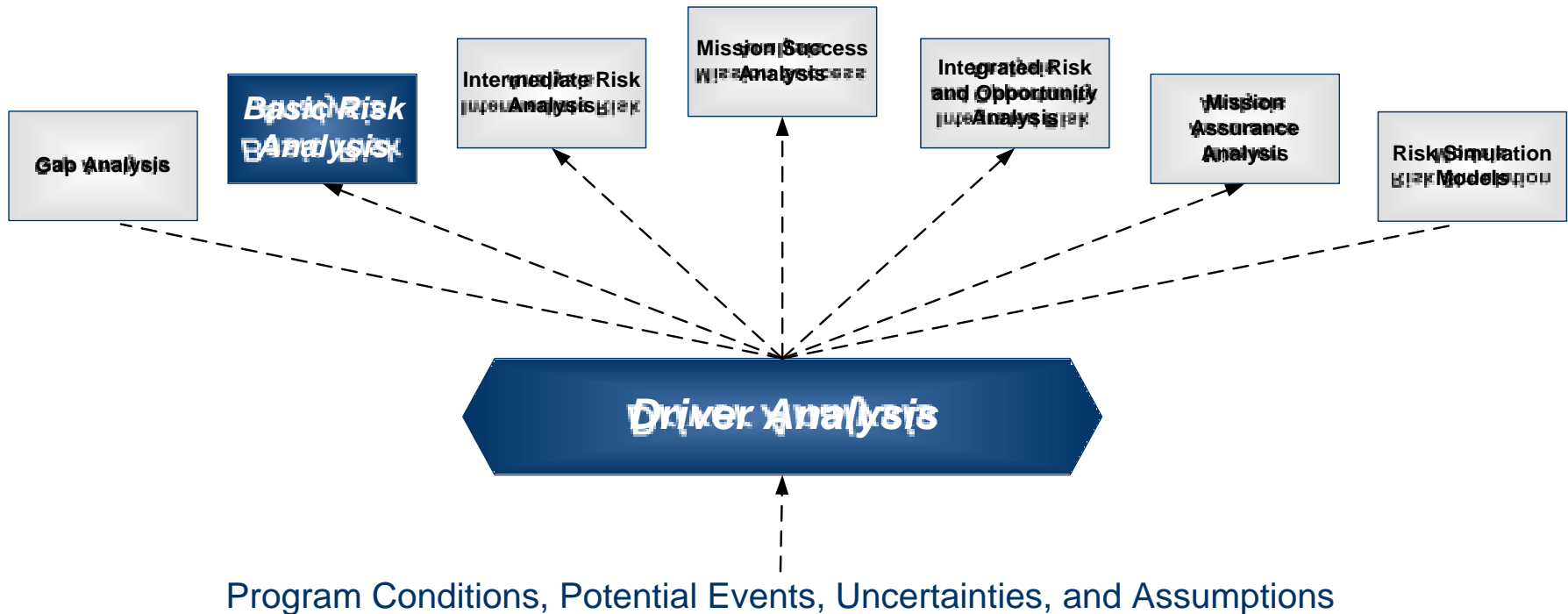
- *Are there some drivers for which you need more information?*
- *Where would you get that information?*



THE RISK DIAGNOSTIC



Risk Diagnostic

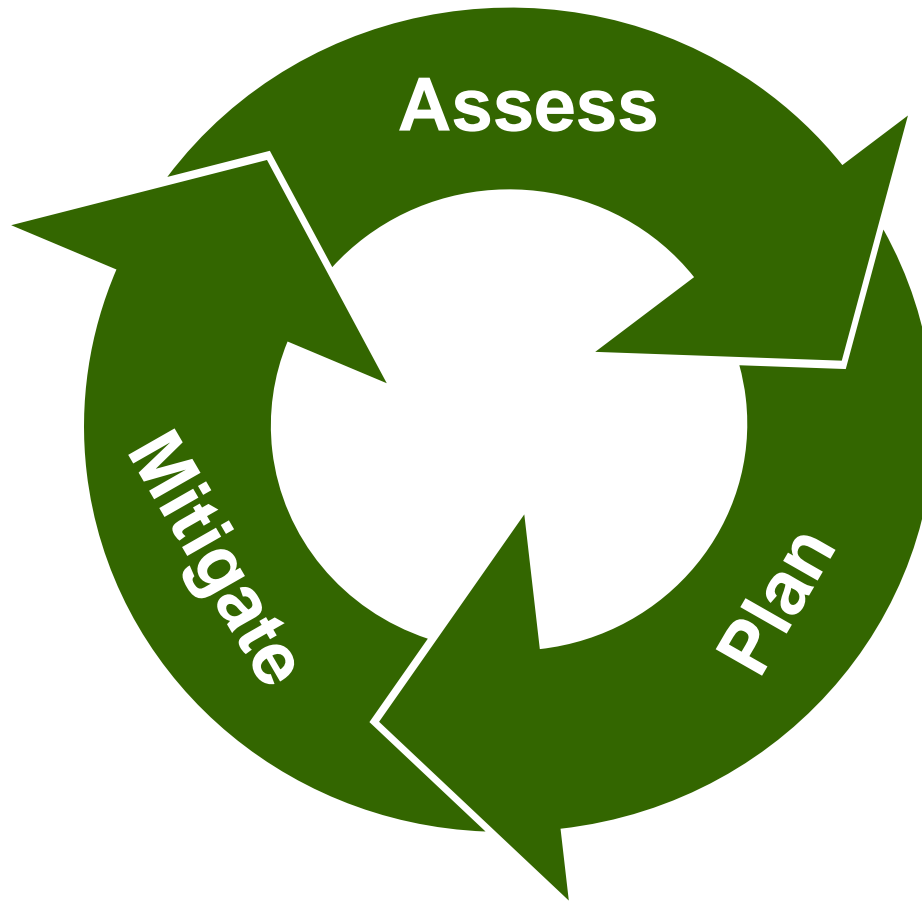


Risk Diagnostic incorporates a basic back-end risk analysis.

Risk Diagnostic is the focus of the this section.



Core Risk Management Activities



What is Mission Risk?

A systemic (i.e., aggregate) risk that affects a program's ability to achieve its key objectives

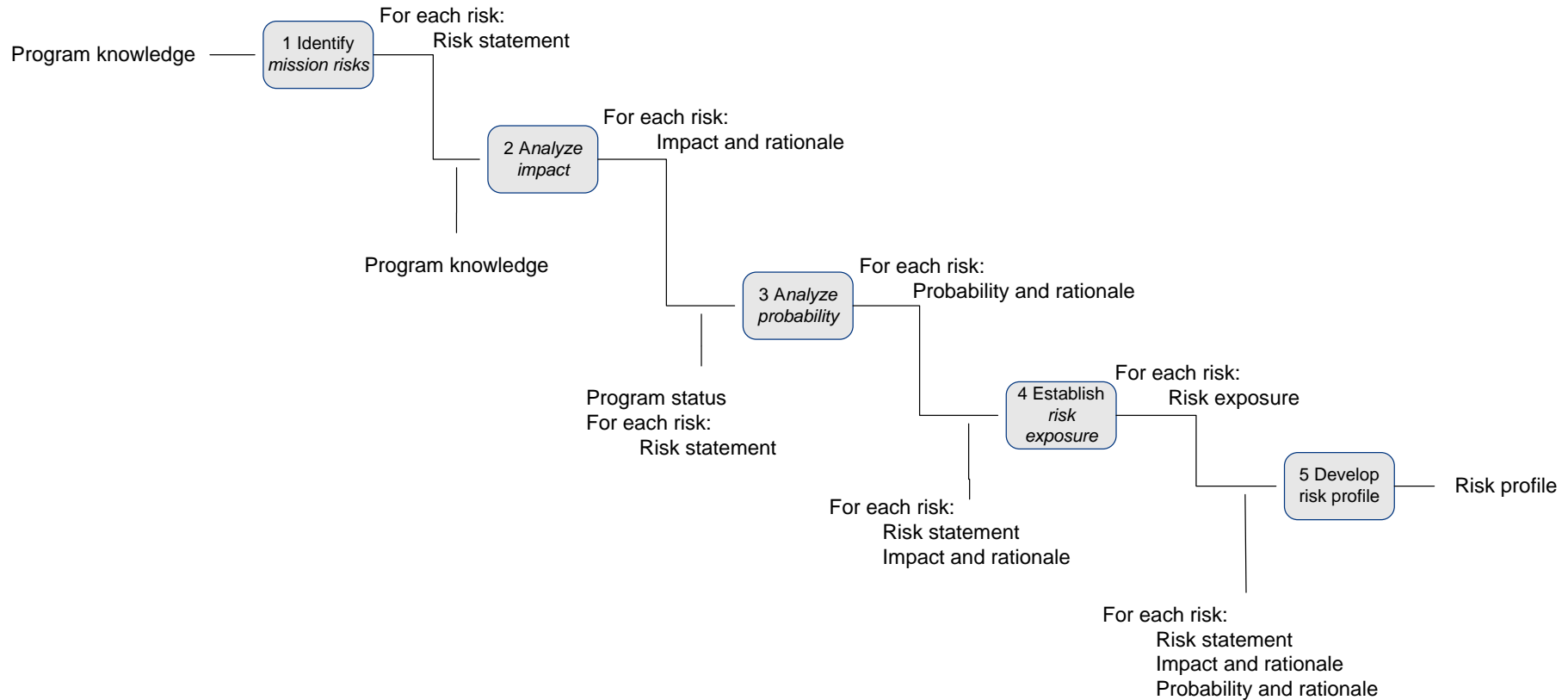
A measure of potential loss in relation to key objectives

- Probability that a driver is in its failure state
- Impact on objectives if a driver is in its failure state

Each driver produces a mission risk.



Risk Diagnostic Method: *Dataflow Diagram*



From Drivers to Mission Risks

The purpose of a risk statement is to provide a unique, succinct, and meaningful descriptor of a risk using a standard format to facilitate communication.

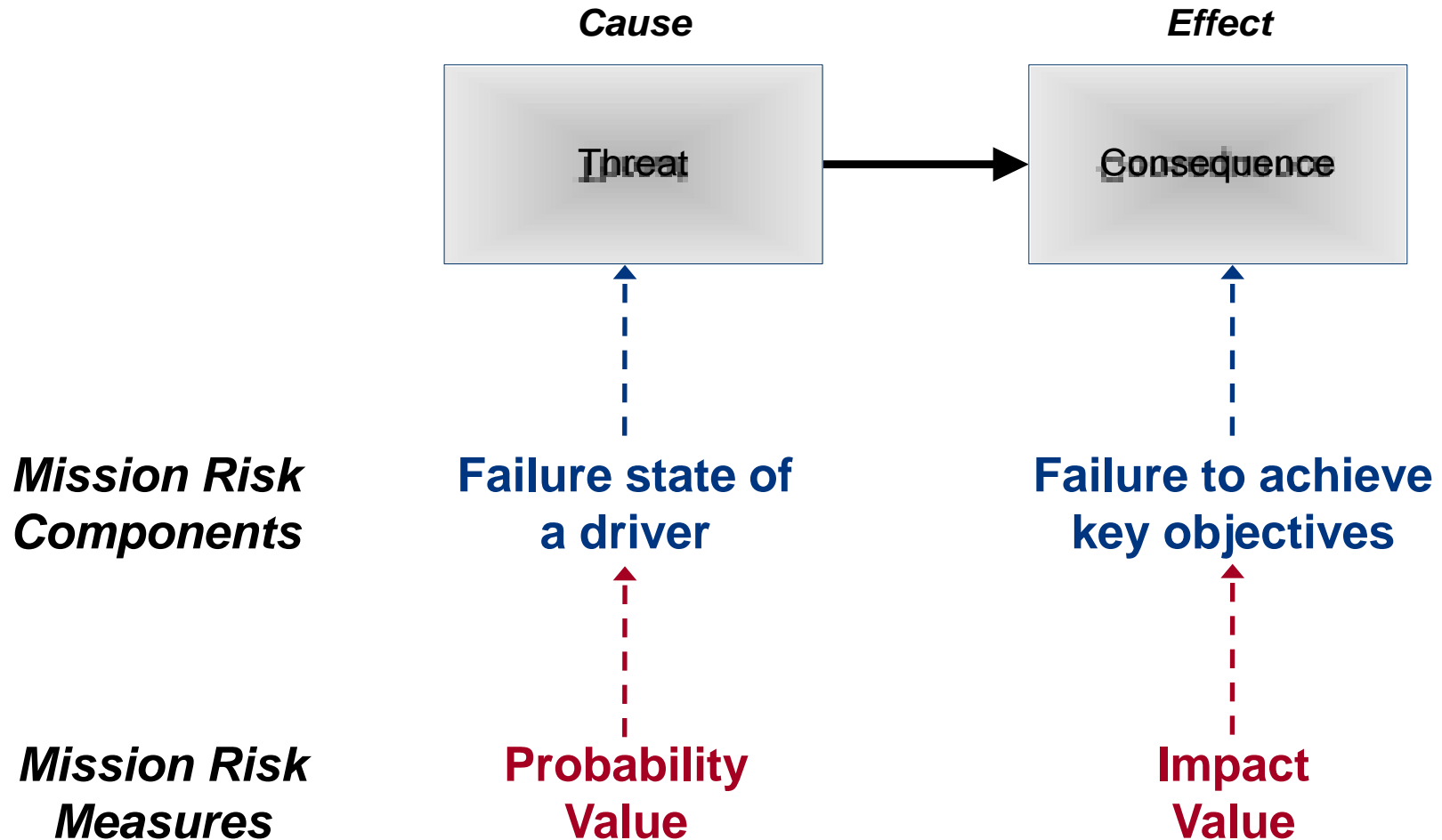
Mosaic uses a driver's failure state as the risk statement for a mission risk.

The consequences for a mission risk are always ***failure to meet key objectives***.

Driver	Risk Statement
Process	The process being used to develop and deploy the system is insufficient.



Components of Mission Risk



Risk Analysis: *Mission Risk*

Mission Risk	Probability	Impact	Risk Exposure
3. The process being used to develop and deploy the system is insufficient.	High	Severe	High

Determined using results of driver analysis

Determined using standard risk analysis methods



Example: *Probability From Driver Response*

Driver Question	Response	Probability
<p>3. Is the process being used to develop and deploy the system sufficient?</p> <p><i>Consider:</i></p> <p>Process design; measurements and controls; process efficiency and effectiveness; acquisition and development life cycles; training</p>	<p><input type="checkbox"/> Yes</p> <p><input type="checkbox"/> Likely Yes</p> <p><input type="checkbox"/> Equally Likely</p> <p><input checked="" type="checkbox"/> Likely No</p> <p><input type="checkbox"/> No</p> <p><input type="checkbox"/> Don't Know</p> <p><input type="checkbox"/> Not Evaluated</p>	<p><input type="checkbox"/> Minimal</p> <p><input type="checkbox"/> Low</p> <p><input type="checkbox"/> Medium</p> <p><input checked="" type="checkbox"/> High</p> <p><input type="checkbox"/> Maximum</p> <p><input type="checkbox"/> Unknown</p> <p><input type="checkbox"/> Not Evaluated</p>



Example: *Impact from Driver Weight*

Driver	Weight	Impact
3. The process being used to develop and deploy the system	<div><input checked="" type="checkbox"/> Critical</div> <div><input type="checkbox"/> High</div> <div><input type="checkbox"/> Medium</div> <div><input type="checkbox"/> Low</div> <div><input type="checkbox"/> Minimal</div> <div><input type="checkbox"/> Don't Know</div>	<div><input checked="" type="checkbox"/> Severe</div> <div><input type="checkbox"/> High</div> <div><input type="checkbox"/> Medium</div> <div><input type="checkbox"/> Low</div> <div><input type="checkbox"/> Minimal</div> <div><input type="checkbox"/> Unknown</div>



Drivers and Mission Risk Statements - 1

Driver	Mission Risk Statement
1. Program Objectives	Program objectives (product, cost, schedule) are unrealistic or unachievable.
2. Plan	The plan for developing and deploying the system is insufficient.
3. Process	The process being used to develop and deploy the system is insufficient.
4. Task Execution	Tasks and activities are performed ineffectively and inefficiently.
5. Coordination	Activities within each team and across teams are not coordinated appropriately.
6. External Interfaces	Work products from suppliers, partners, or collaborators will not meet the program's quality and timeliness requirements.
7. Information Management	The program's information is not managed appropriately.



Drivers and Mission Risks - 2

Driver	Mission Risk Statement
8. Technology	The program team does not have the tools and technologies it needs to develop the system and transition it to operations.
9. Facilities and Equipment	Facilities and equipment are insufficient to support the program.
10. Organizational Conditions	Enterprise, organizational, and political conditions are hindering completion of program activities.
11. Compliance	The program does not comply with all relevant policies, laws, and regulations.
12. Event Management	The program has insufficient capacity and capability to identify and manage potential events and changing circumstances.
13. Requirements	System requirements are not well understood.
14. Design and Architecture	The design and architecture are insufficient to meet system requirements and provide the desired operational capability.



Drivers and Mission Risks - 3

Driver	Mission Risk Statement
15. System Capability	The system will not satisfactorily meet its requirements.
16. System Integration	The system will not sufficiently integrate and interoperate with other systems when deployed.
17. Operational Support	The system will not effectively support operations.
18. Adoption Barriers	Barriers to customer/user adoption of the system have not been managed appropriately.
19. Operational Preparedness	People will not be prepared to operate, use, and maintain the system.
20. Certification and Accreditation	The system will not be appropriately certified and accredited for operational use.



Example: *Probability Criteria for Mission Risks*

Probability	Description
Minimal	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)
Low	The answer is most likely “yes.” There is some chance that the answer could be “no.” (~ 75% probability of yes)
Medium	The answer is just as likely to be “yes” or “no.” (~ 50% probability of yes)
High	The answer is most likely “no.” There is some chance that the answer could be “yes.” (~ 25% probability of yes)
Maximum	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)
Unknown	More information is needed to answer the question.
Not Evaluated	The driver question is not relevant at this point in time. It was not evaluated.



Example: *Impact Criteria for Mission Risks*

Impact	Description
Severe	The driver is vital to the program. It has an extremely strong influence on program success or failure.
High	The driver is very important to the program, but not vital. It has a strong influence on program success or failure.
Medium	The driver is moderately important to the program. It has some influence on program success or failure.
Low	The driver is somewhat important to the program. It has a weak influence on program success or failure.
Minimal	The driver is not important to the program. It has negligible influence on program success or failure.
Unknown	More information is needed to answer the question.





Mission Risk Exposure Criteria

		Impact				
		Severe (5)	High (4)	Medium (3)	Low (2)	Minimal (1)
Probability	Maximum (5)	Severe (5)	High (4)	Medium (3)	Low (2)	Minimal (1)
	High (4)	High (4)	Medium (3)	Low (2)	Minimal (1)	Minimal (1)
	Medium (3)	Medium (3)	Low (2)	Minimal (1)	Minimal (1)	Minimal (1)
	Low (2)	Low (2)	Minimal (1)	Minimal (1)	Minimal (1)	Minimal (1)
	Minimal (1)	Minimal (1)	Minimal (1)	Minimal (1)	Minimal (1)	Minimal (1)





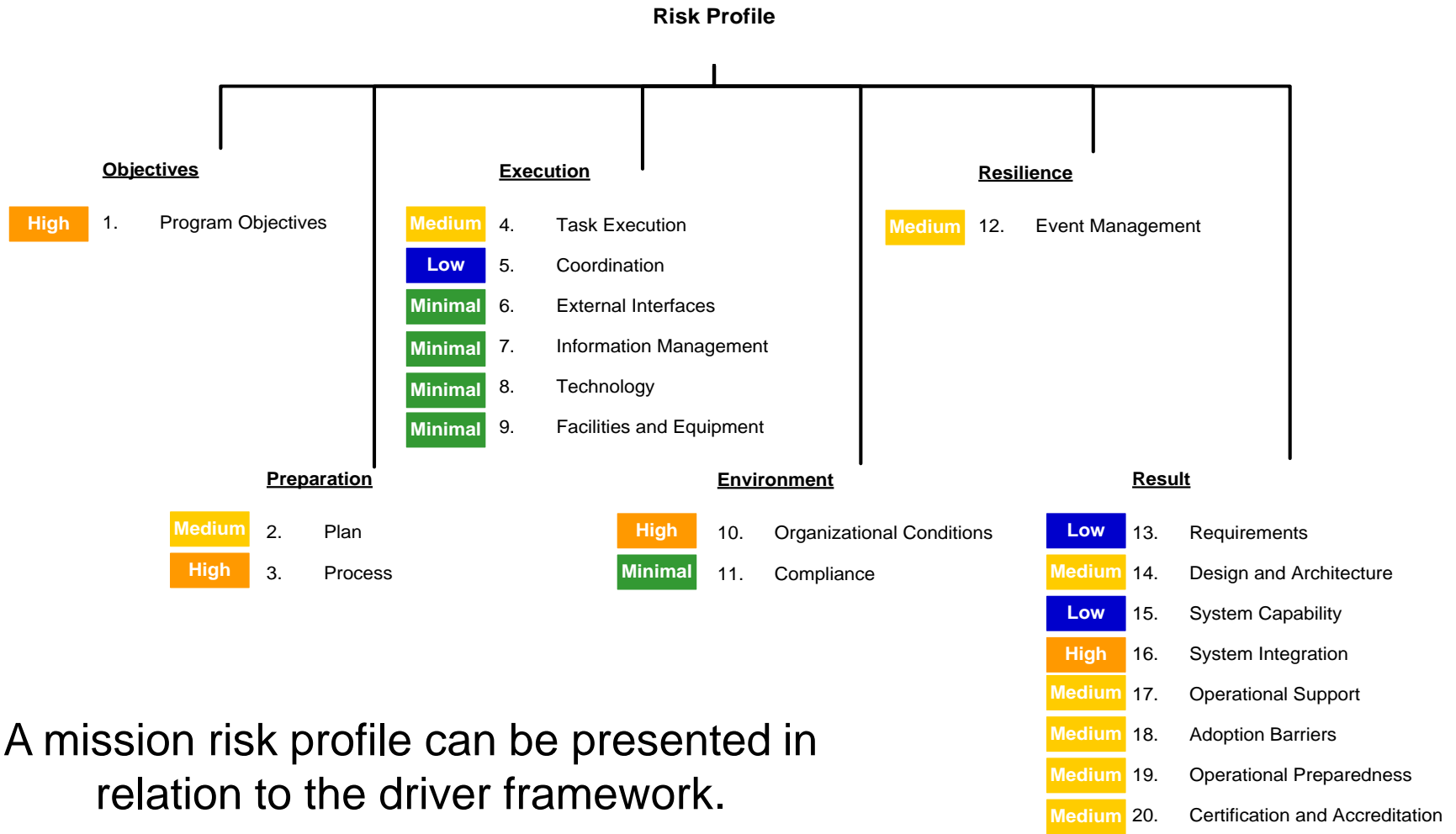
Example: *Mission Risk Profile – List View*

ID	Mission Risk Statement	Prob.	Impact	Risk Exp.
3	The process being used to develop and deploy the system is insufficient.	High (4)	Severe (5)	High (4)
11	The program does not comply with all relevant policies, laws, and regulations.	Medium (3)	Low (2)	Minimal (1)

A risk profile can be presented as a list or spreadsheet.

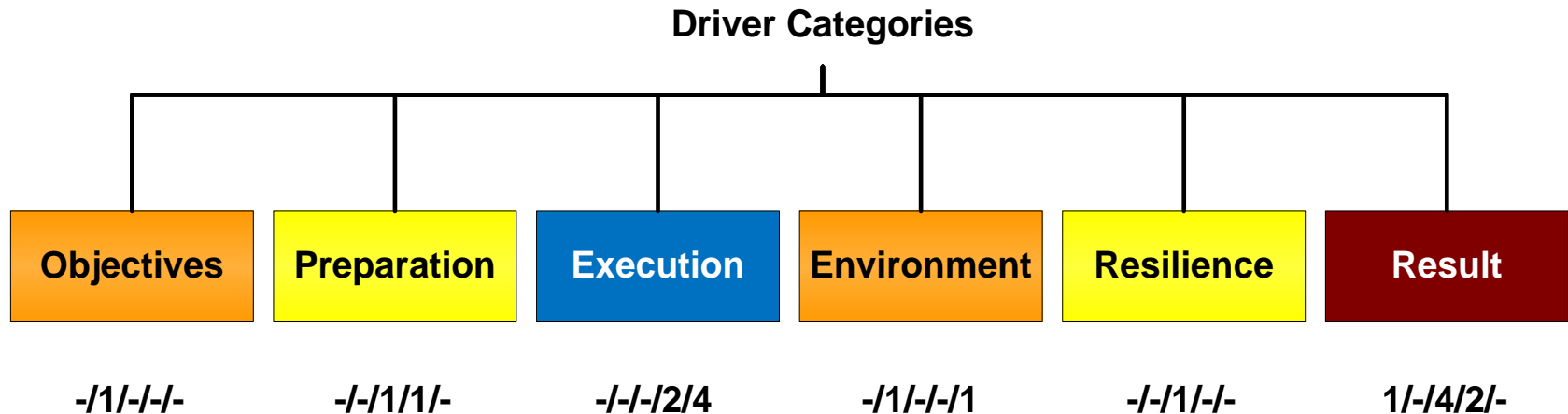


Example: *Mission Risk Profile – Driver Framework View*



A mission risk profile can be presented in relation to the driver framework.

Example: *Mission Risk Profile – Category View*



Each driver category reflects the highest risk exposure for the mission risks associated with that category

Beneath each category is the total number of mission risks at each risk exposure level (severe/high/medium/low/minimal)



Mitigation Approaches for Mission Risks

A mitigation approach defines the strategy for addressing a risk.

Mitigation approaches for mission risks include

- **Control** – Actions are implemented in attempt to reduce or contain a risk.
- **Watch** – Reassess a risk's probability on a more frequent basis than is provided by scheduled, periodic risk assessments
- **Defer** – No mitigation actions will be taken at the present time. The risk will be reassessed during the next scheduled risk assessment.

Mitigation approaches should be shared with all relevant stakeholders as appropriate.

Controlling and watching mission risks require development of mitigation plans



Strategies for Controlling Mission Risks

Maintain strengths

Take action to reinforce positive conditions that are guiding drivers toward their success states.

Resolve weaknesses/issues

Take action to correct weaknesses or issues that are guiding drivers toward their failure states.

Manage tactical opportunities

Take action to leverage tactical opportunities that could guide drivers toward their success states.

Manage tactical risks

Take action to mitigate tactical risks that could guide drivers toward their failure states.



Tracking Decisions

As a mitigation plan is implemented, decisions will be required about what action(s) to take.

Tracking decisions include

- Continue implementing the mitigation plan as intended
- Modify the mitigation plan
- Implement the contingency plan (if one exists)
- Modify the mitigation approach and take any appropriate actions, for example change the mitigation approach from
 - *Watch* to *Control* if risk exposure exceeds a predefined threshold
 - *Control* to *Watch* or *Defer* if strategies for controlling a mission risk have been achieved

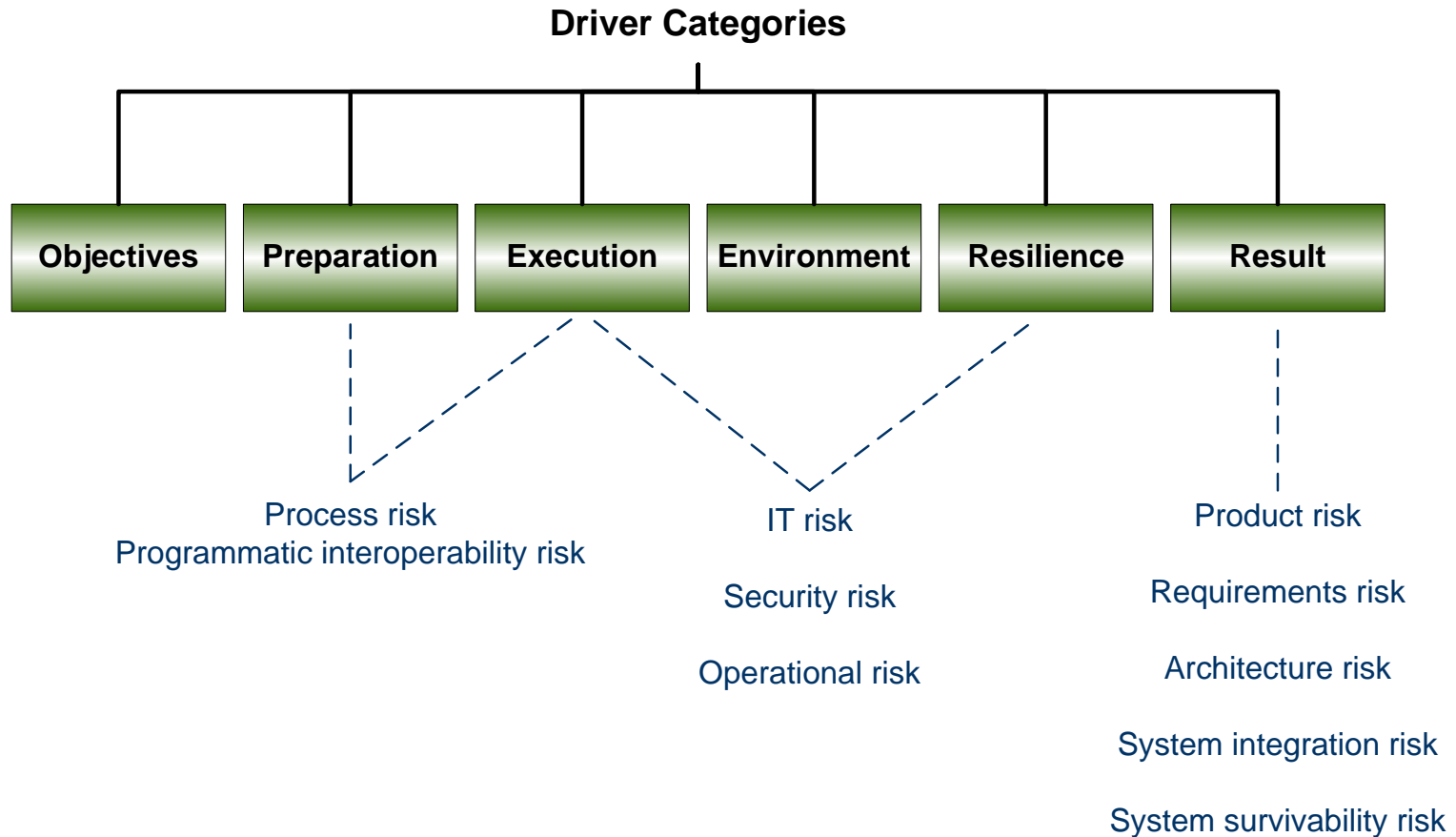
Tracking decisions should be shared with all relevant stakeholders as appropriate.



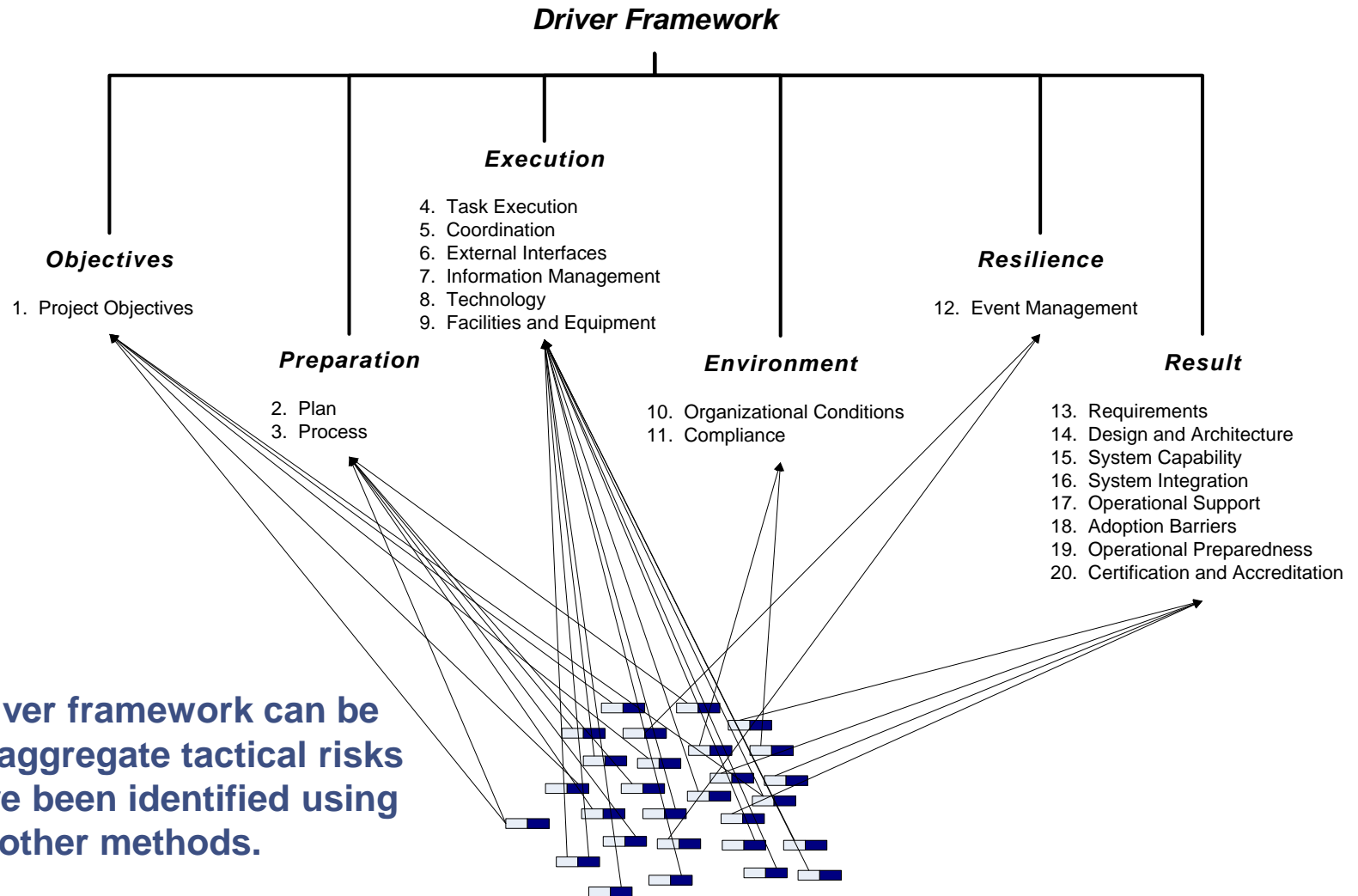
IMPLEMENTATION OPTIONS



Use Drivers to Integrate Multiple Types of Risk



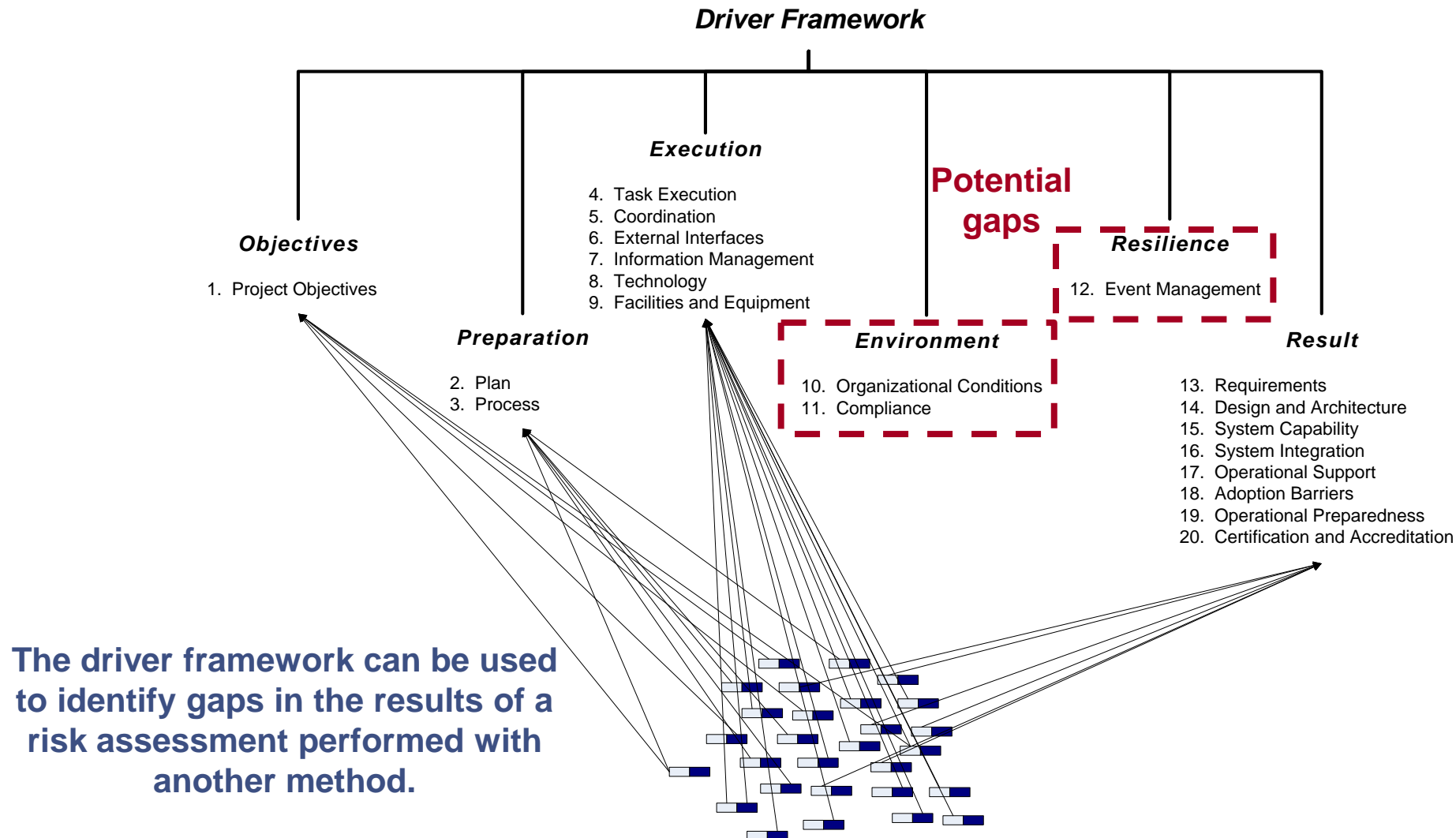
Using the Driver Framework to Aggregate Tactical Risks



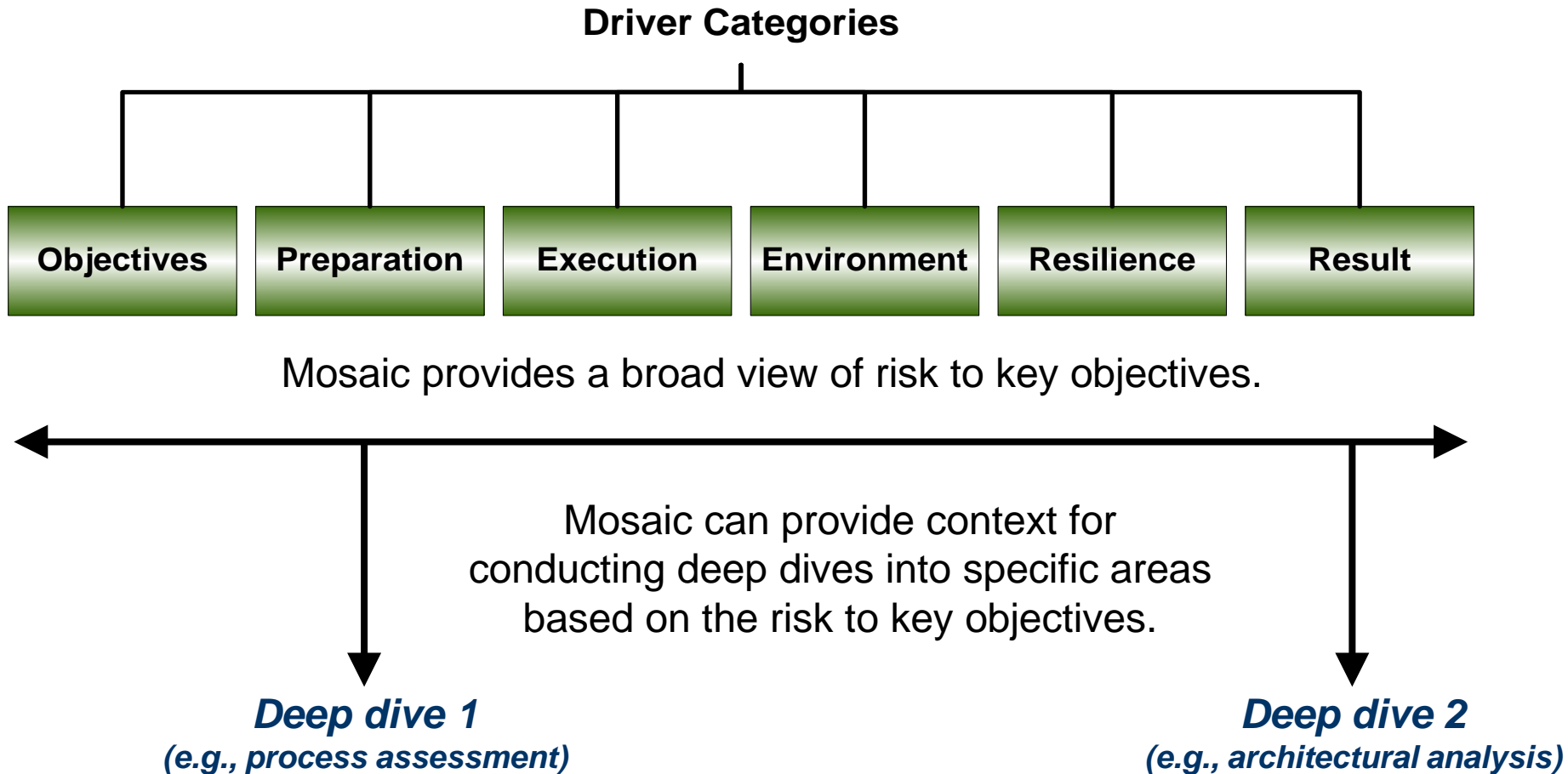
The driver framework can be used to aggregate tactical risks that have been identified using other methods.



Using the Driver Framework to Identify Gaps in Tactical Risk Assessments



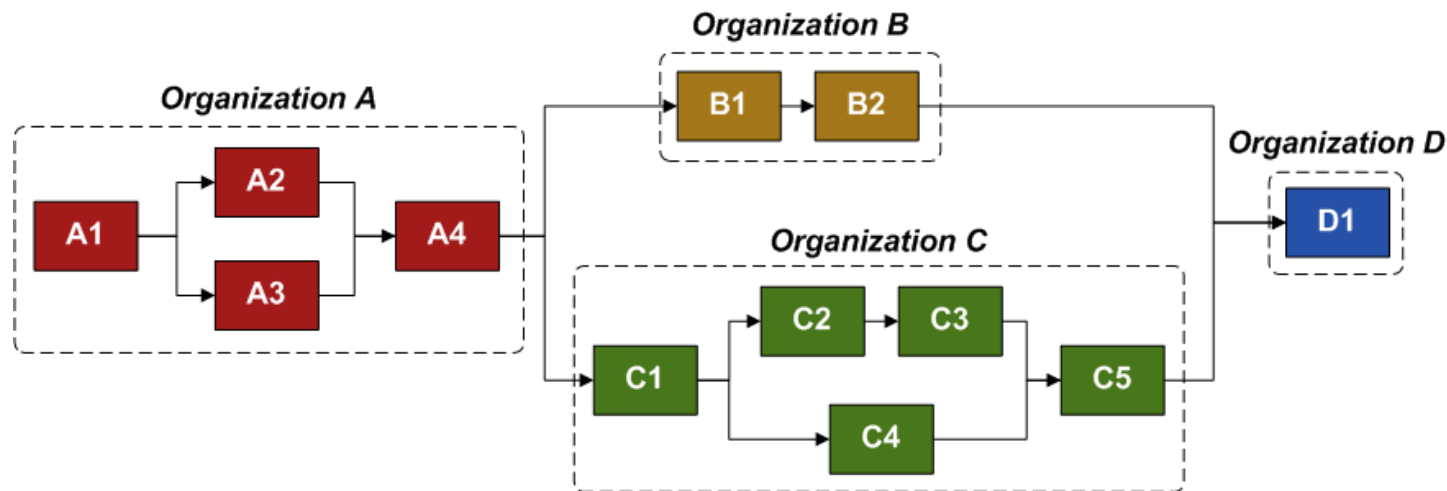
Using Risk Diagnostic as a Broad-Based Assessment



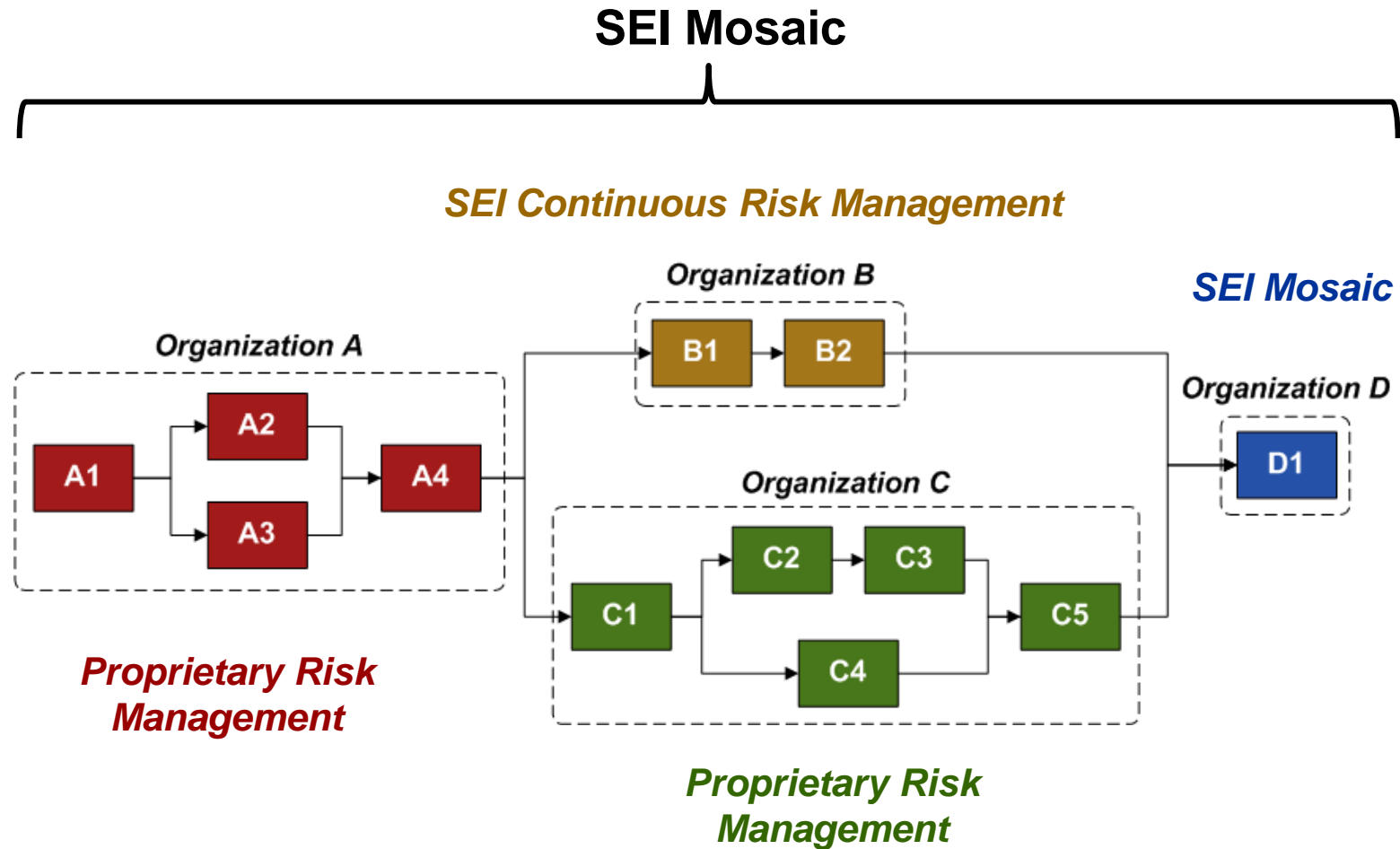
Mosaic in Multi-Enterprise Environments - 1

Distributed programs that cross multiple organizational boundaries require a systemic viewpoint when managing risk.

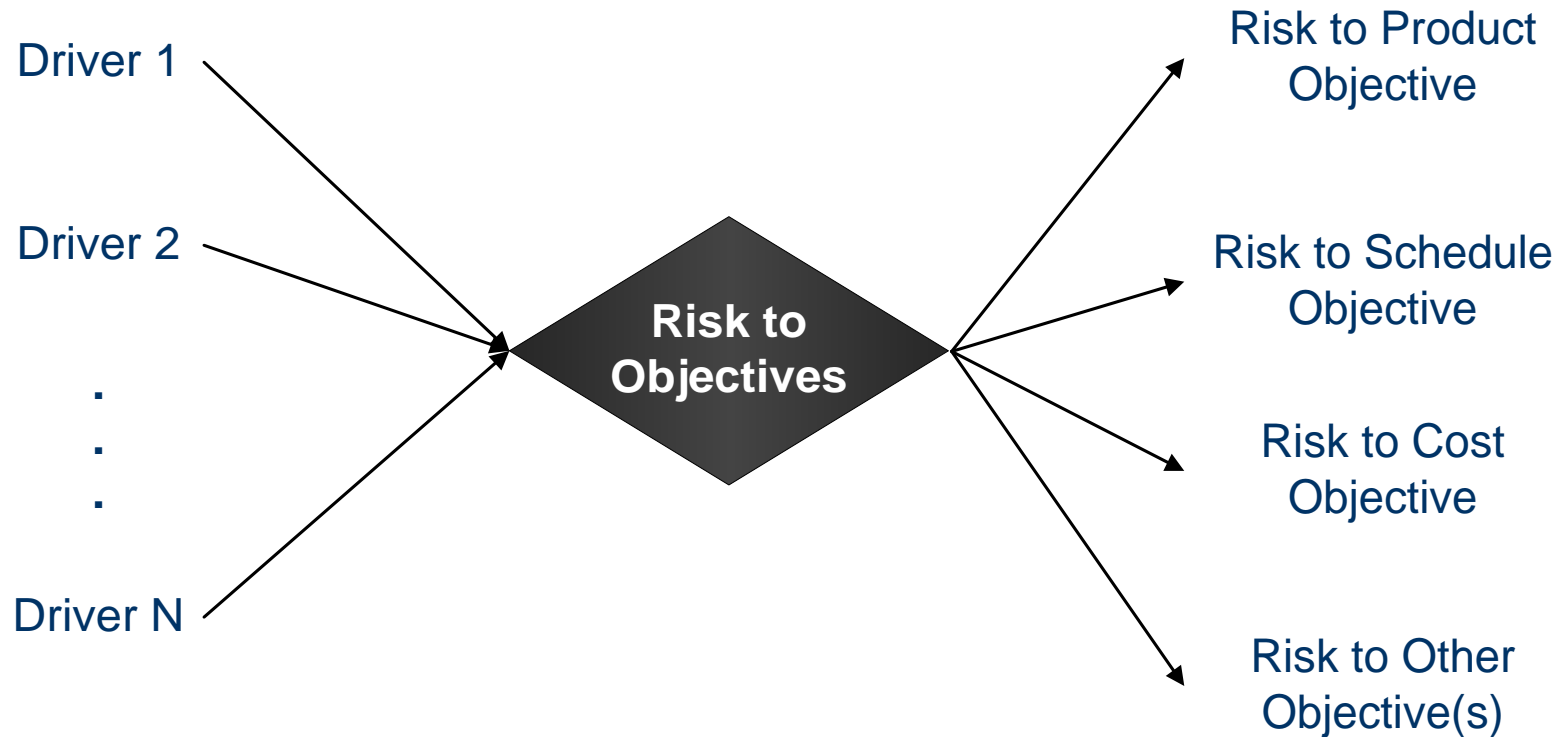
- Acquire and maintain a broad view of the impact to program objectives
- Avoid local optimization of risk that aggravates mission risk
- Keep volume of risk data to a manageable level



Mosaic in Multi-Enterprise Environments - 2



Analyze Risk to Specific Objectives

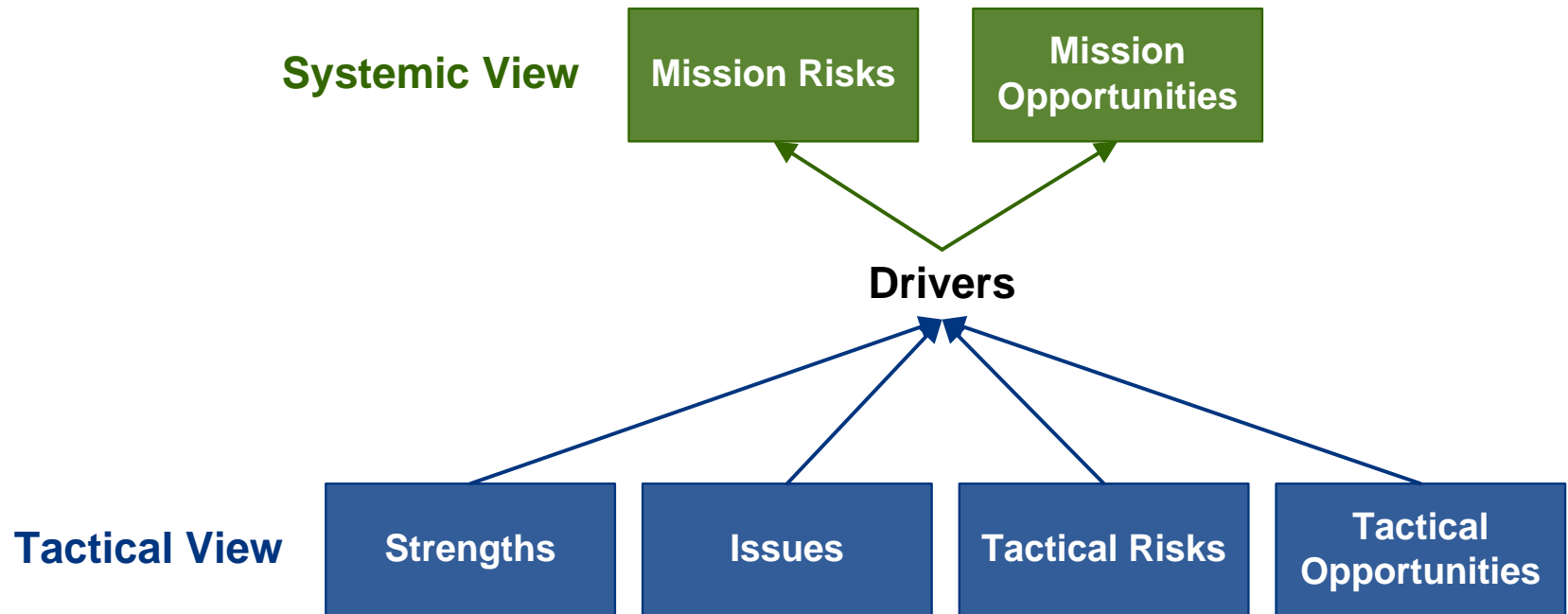


As presented in this course, basic risk analysis determines risk to objectives.

You can also establish the risk to each individual objective based on the objective's criticality to the mission.



Integrated Risk and Opportunity Analysis

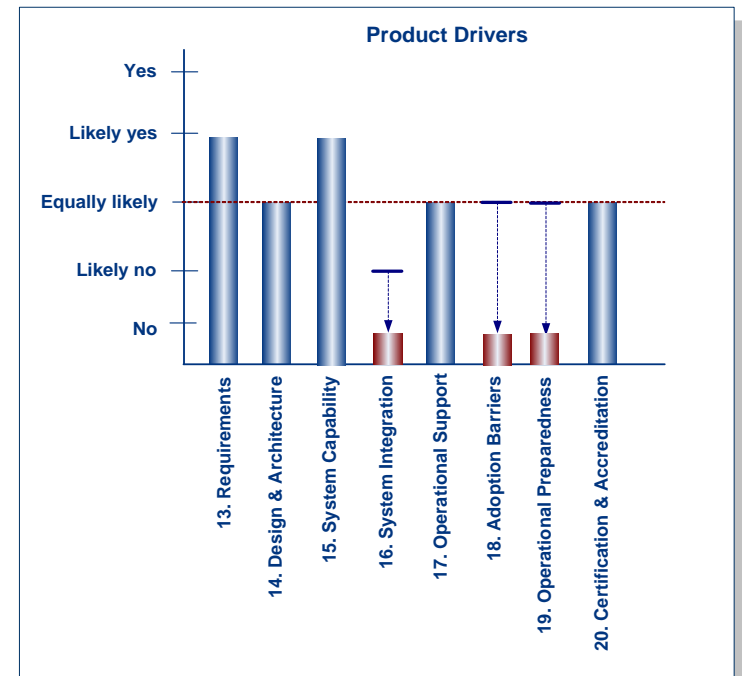
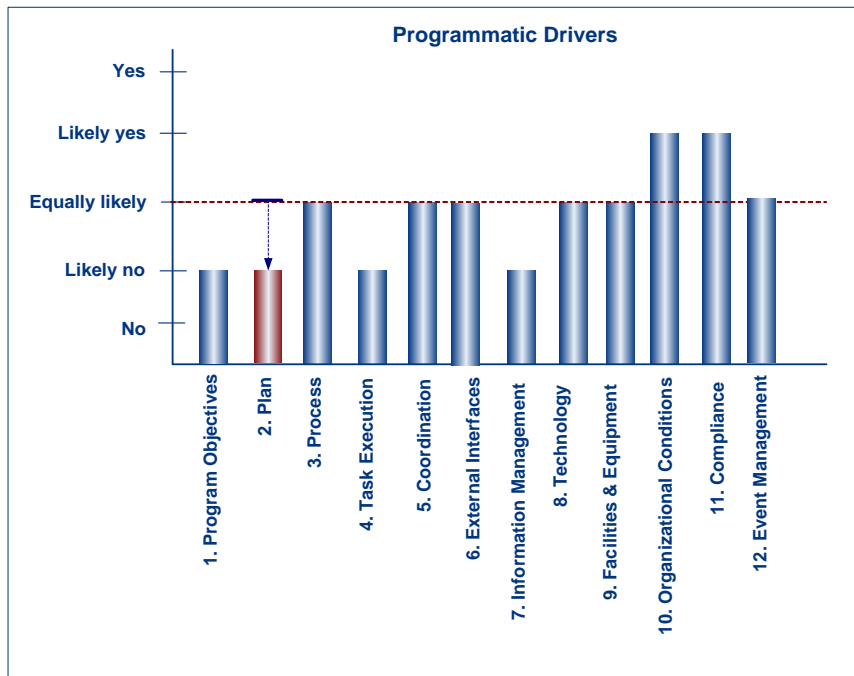


A mission risk is a circumstance that has the potential to cause loss from the business or mission perspective

A mission opportunity is a circumstance that has the potential to provide a gain from the business or mission perspective



Integrated Risk and Opportunity: *Impact of Taking an Opportunity on Mission Drivers*

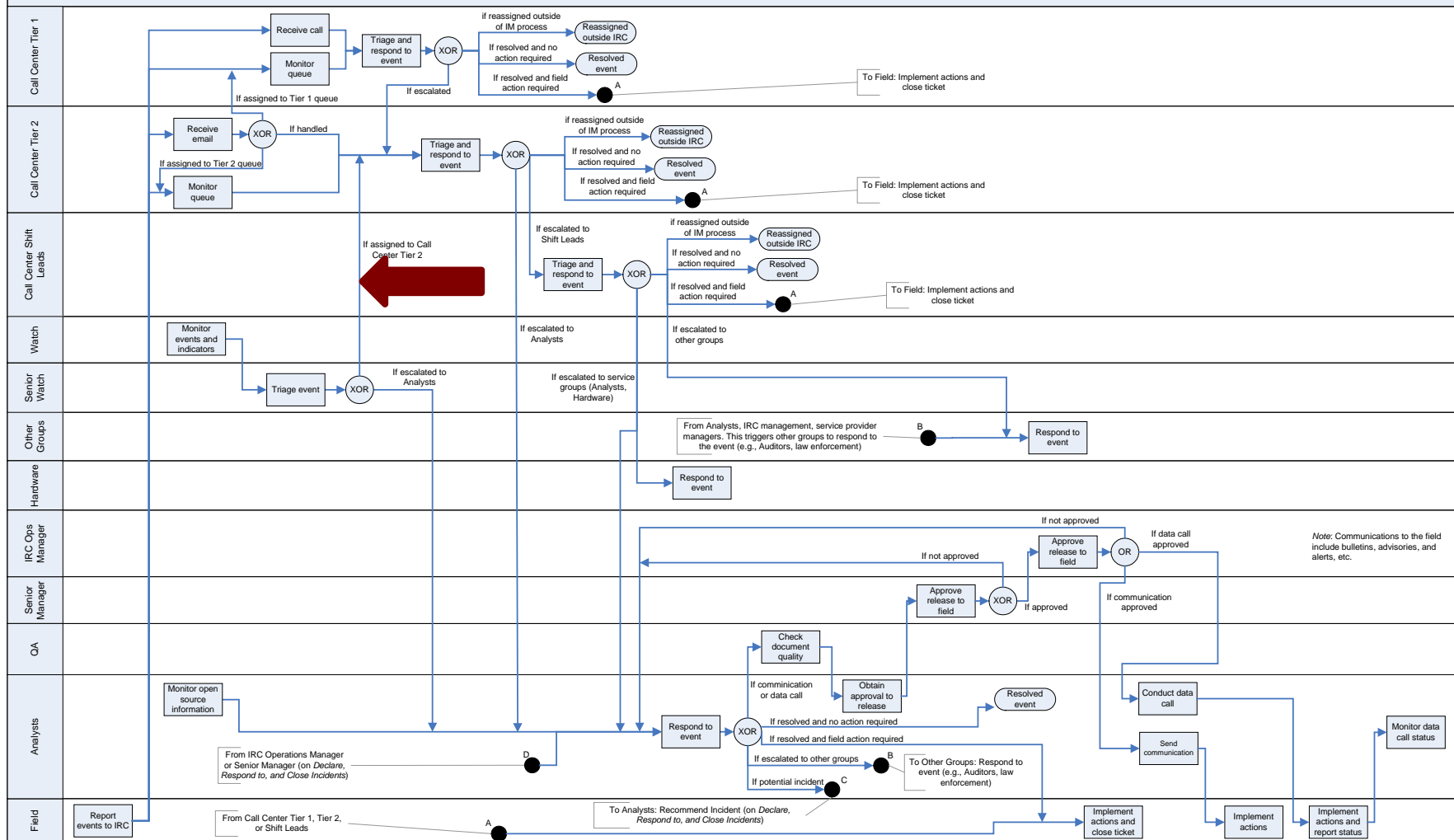


Here, taking an opportunity to use unexpected funds to improve the test facility has unexpectedly negative consequences on planning, system integration, adoption barriers, and operational preparedness.



Analyze Drivers At Interfaces

Detect, Triage, and Respond to Events



SUMMARY



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

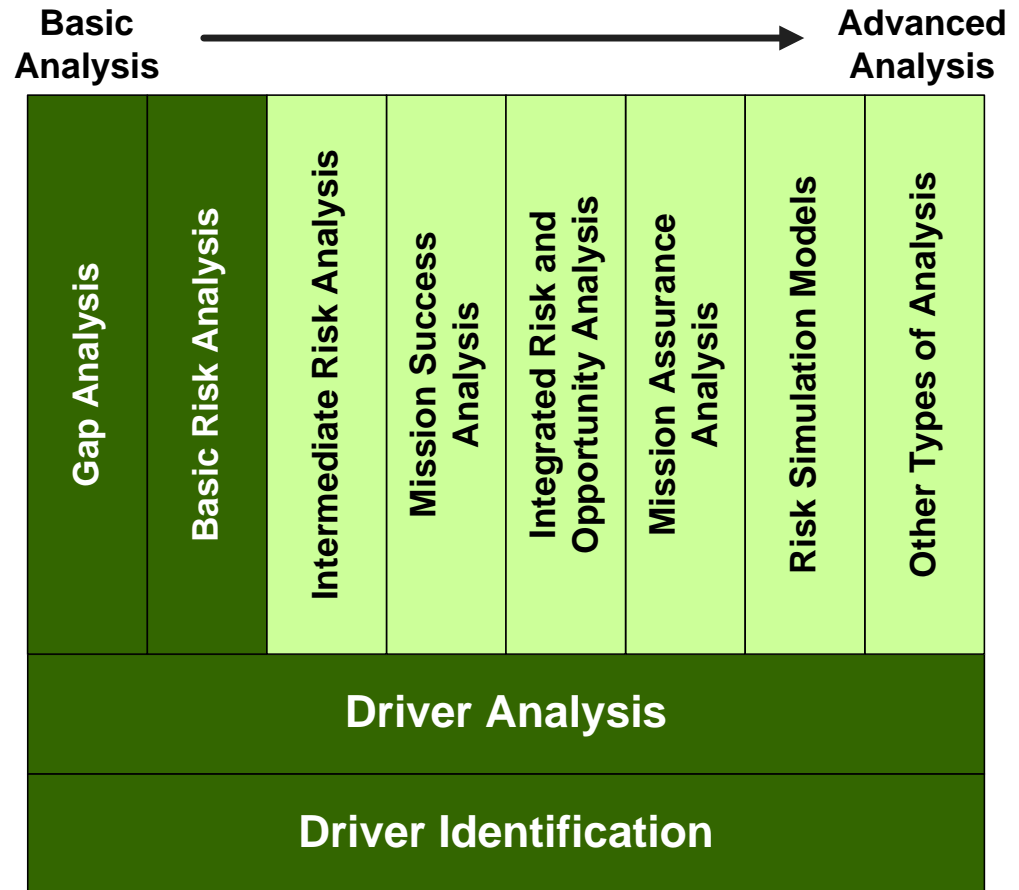
Mosaic

- 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



Summary of Key Points - 2

Driver identification and analysis provide a common foundation for multiple back-end analyses



Summary of Key Points - 3

The Mission Diagnostic

- Provides a time-efficient means of assessing a program's success/failure drivers

The Risk Diagnostic

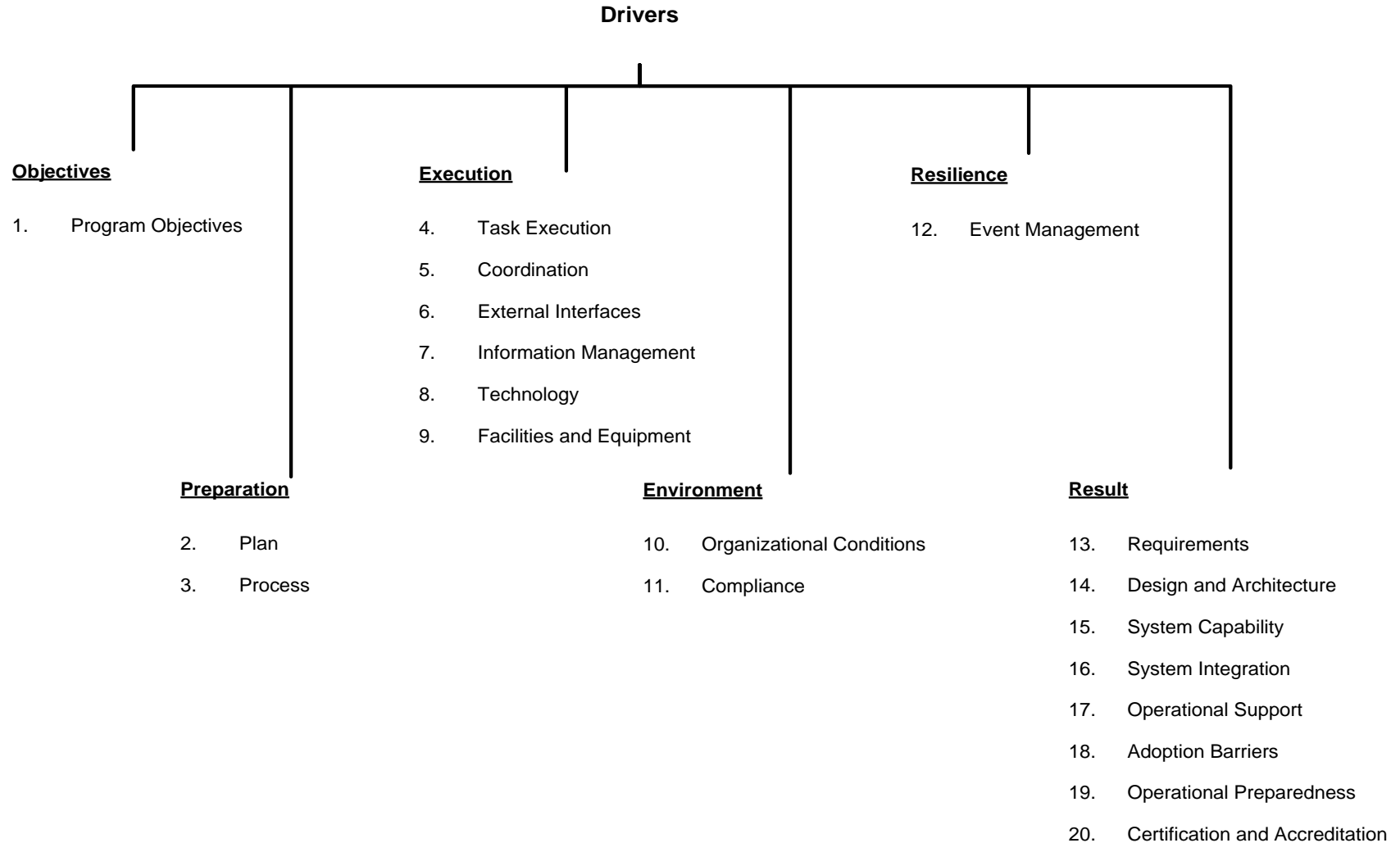
- Provides a time-efficient means of assessing mission risks to program objectives
- Based on a set of key drivers

Drivers can be the foundation for a variety of deeper analyses.

Drivers can be used to integrate tactical information from a variety of sources.



Drivers for Software/System Development and Deployment



Additional Materials Available

1. Streamlined Mission Diagnostic Method
 - Set of worksheets in the form of a short workbook
2. Streamlined Risk Diagnostic Method
 - Set of worksheets in the form of a short workbook



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/risk/>

U.S. mail Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890



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.





Software Engineering Institute

Carnegie Mellon



Software Engineering Institute

Carnegie Mellon

Rethinking Risk Management
Audrey Dorofee & Chris Alberts
© 2009 Carnegie Mellon University