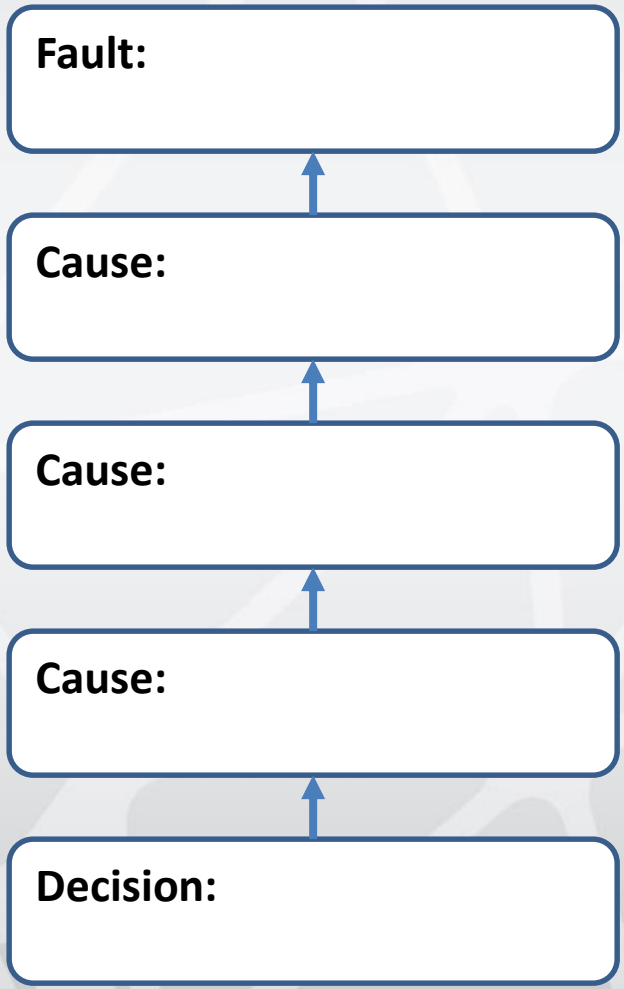
All Faults are Decision Faults – and How to Prevent Them

John Fitch
President/CTO
Decision Driven® Solutions
fitch@decisiondriven.com
www.decisiondriven.com

Let’s start with a thought experiment

Think of a system, product or process that failed you, your family or your team

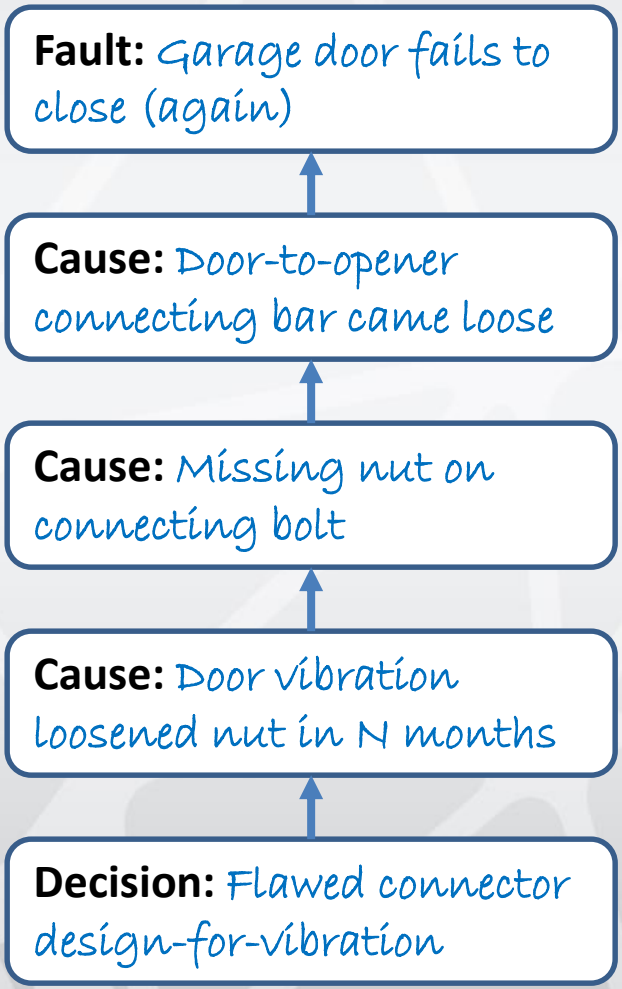
- Name the defect/fault you experienced:
- What caused this defect?
- Continue back the cause-effect chain asking “What caused the cause?”
- How many “hops” does it take to get back to a faulty decision?



Thought experiment - example

Think of a system, product or process that failed you, your family or your team

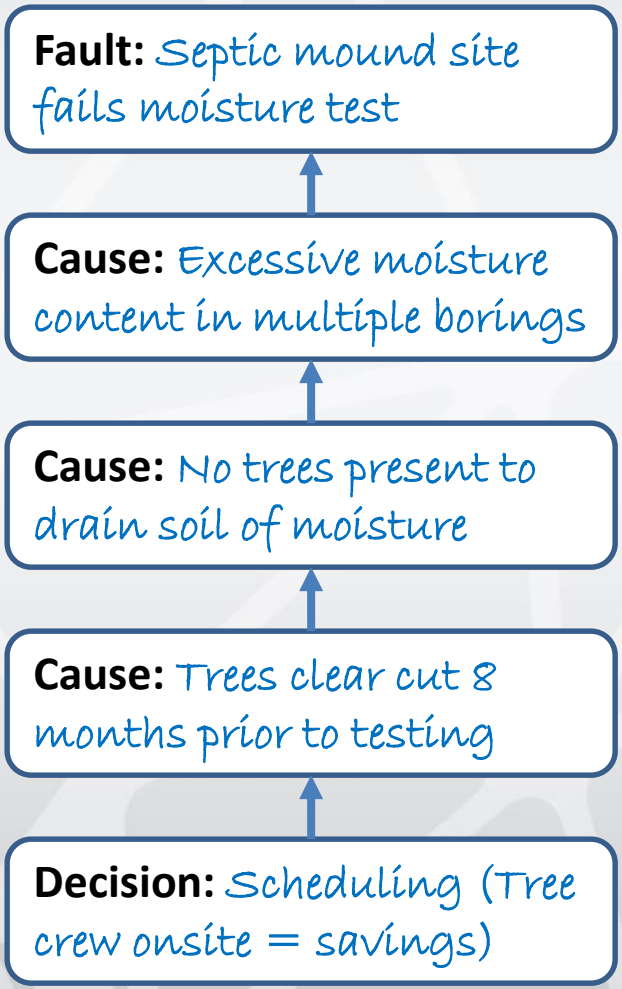
- Name the defect/fault you experienced:
- What caused this defect?
- Continue back the cause-effect chain asking “What caused the cause?”
- How many “hops” does it take to get back to a faulty decision?



Thought experiment - example

Think of a system, product or process that failed you, your family or your team

- Name the defect/fault you experienced:
- What caused this defect?
- Continue back the cause-effect chain asking “What caused the cause?”
- How many “hops” does it take to get back to a faulty decision?



Decisions create the future!

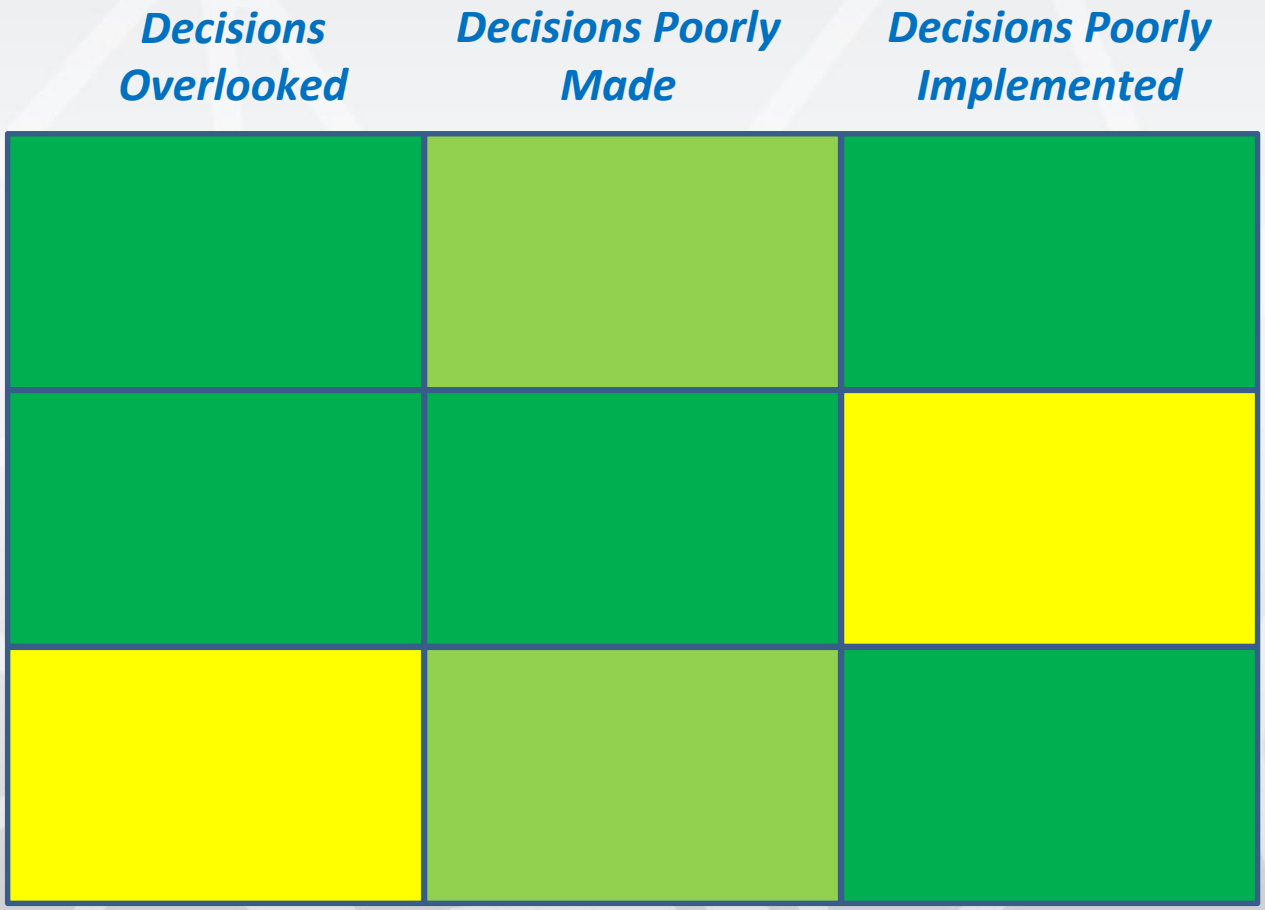
for good or ill

- Systems Engineers (aka human beings) create the future with each decision that we make
- Faulty results can be traced back to 3 types of decision failure modes:
  - A decision overlooked
  - A decision poorly made
  - A decision poorly implemented
- These faults can be prevented (or at least reduced) by:
  - A robust, proactive Decision Management methodology
  - The use of Decision Patterns to frame the problem space
  - The maintenance of Decision-to-Everything traceability

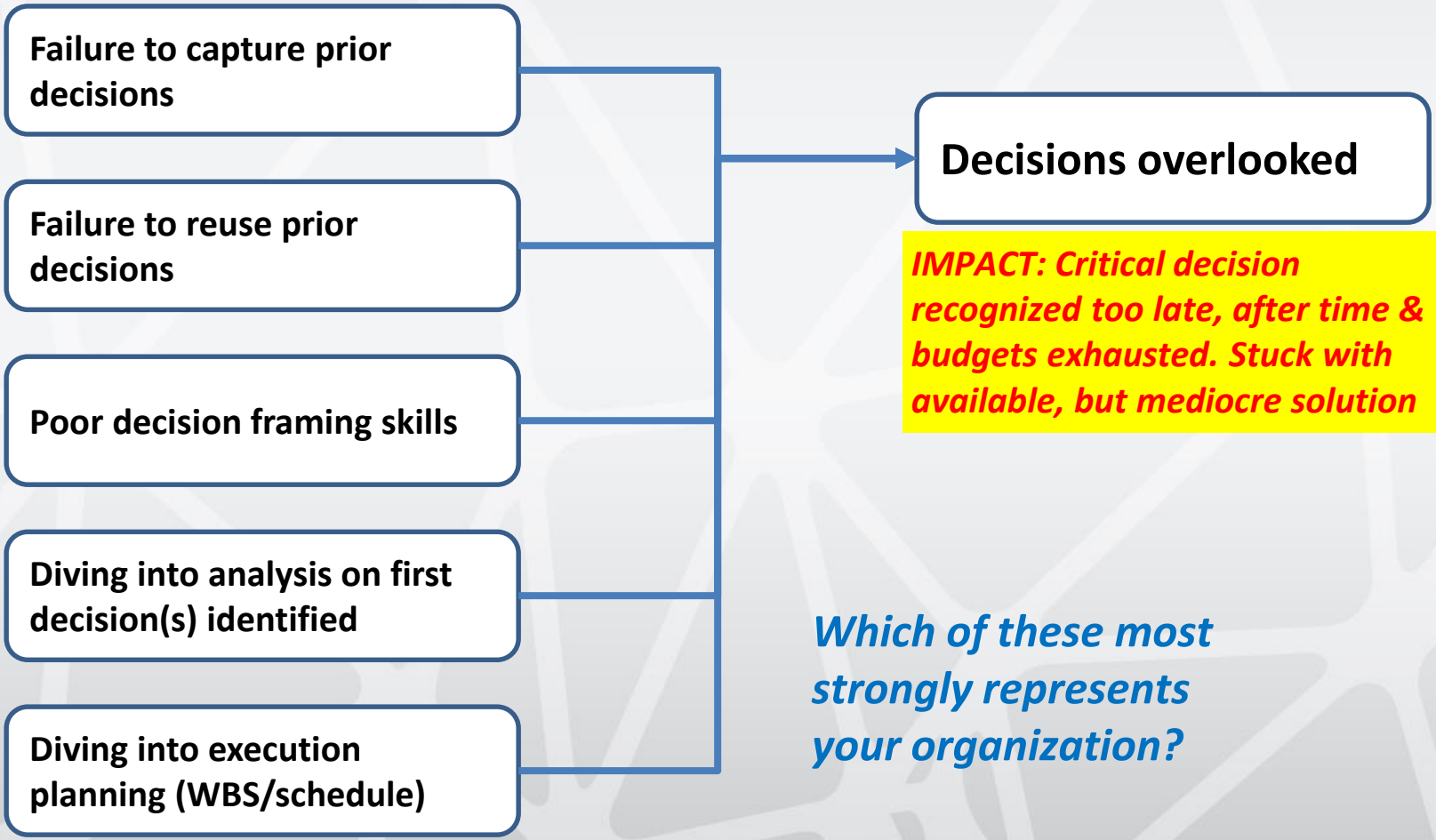
Which of these do you use today?

Decision Faults vs Preventive Measures

Level of prevention



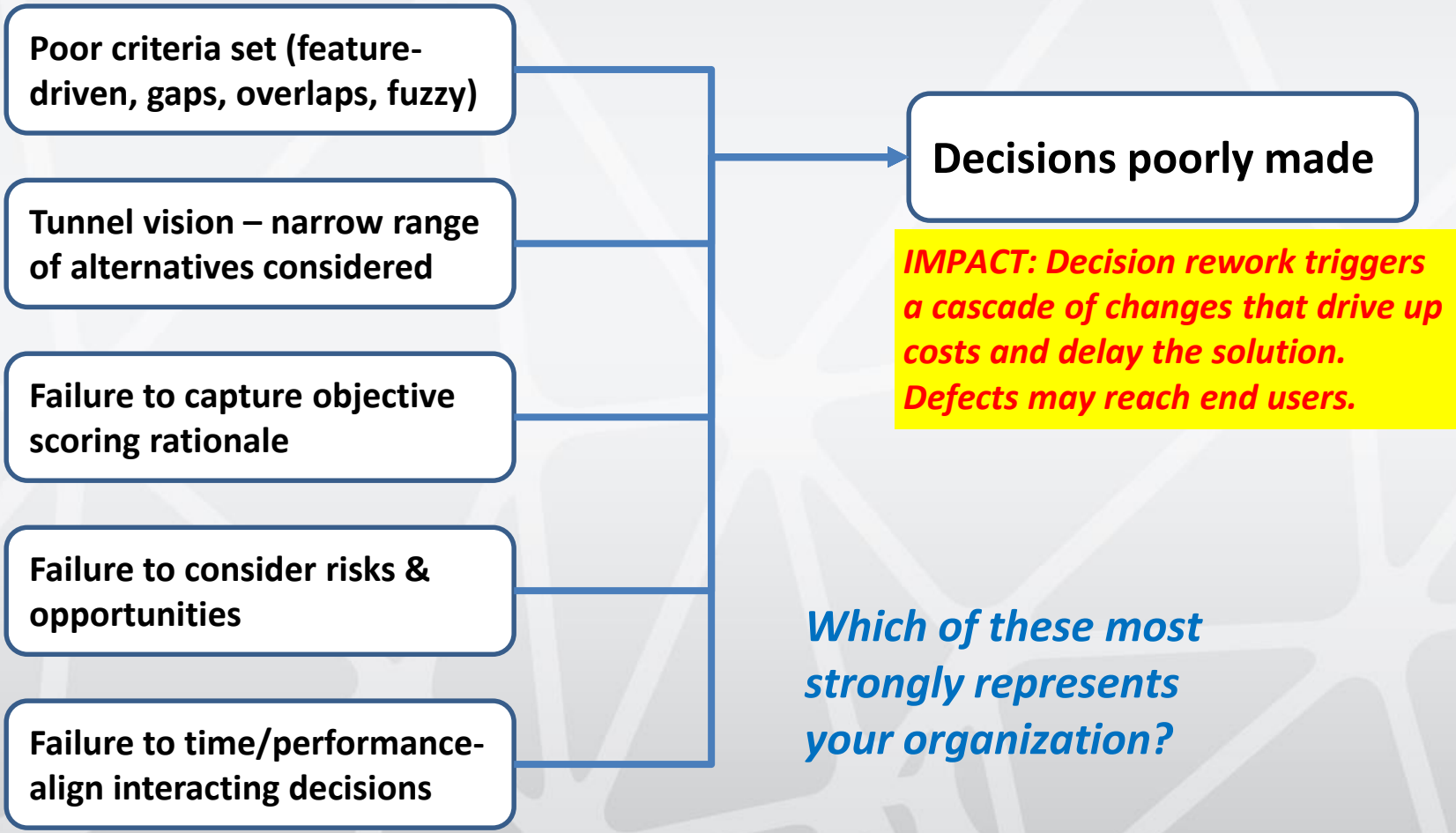
Causes: Decisions Overlooked



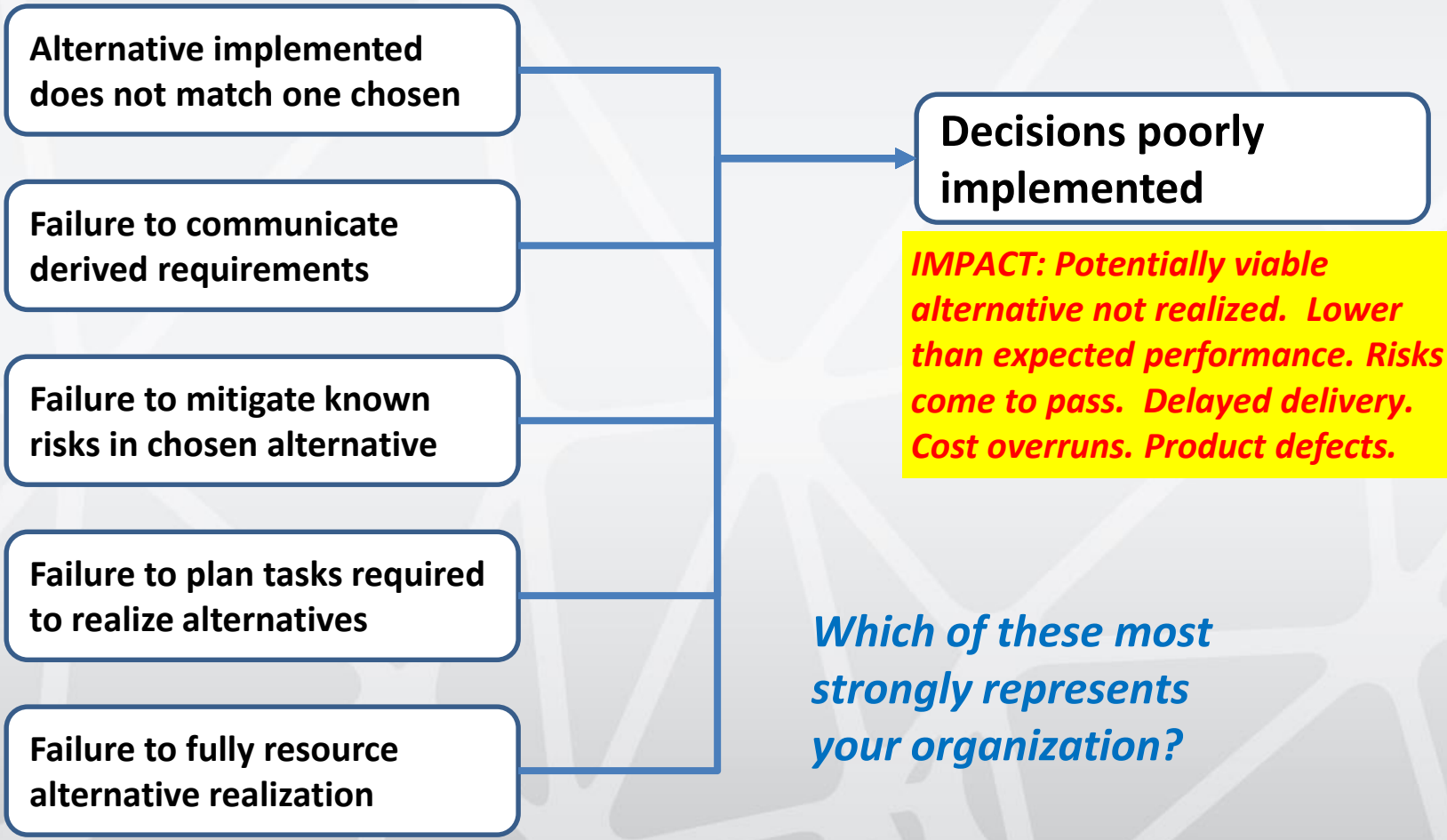
IMPACT: Critical decision recognized too late, after time & budgets exhausted. Stuck with available, but mediocre solution

Which of these most strongly represents your organization?

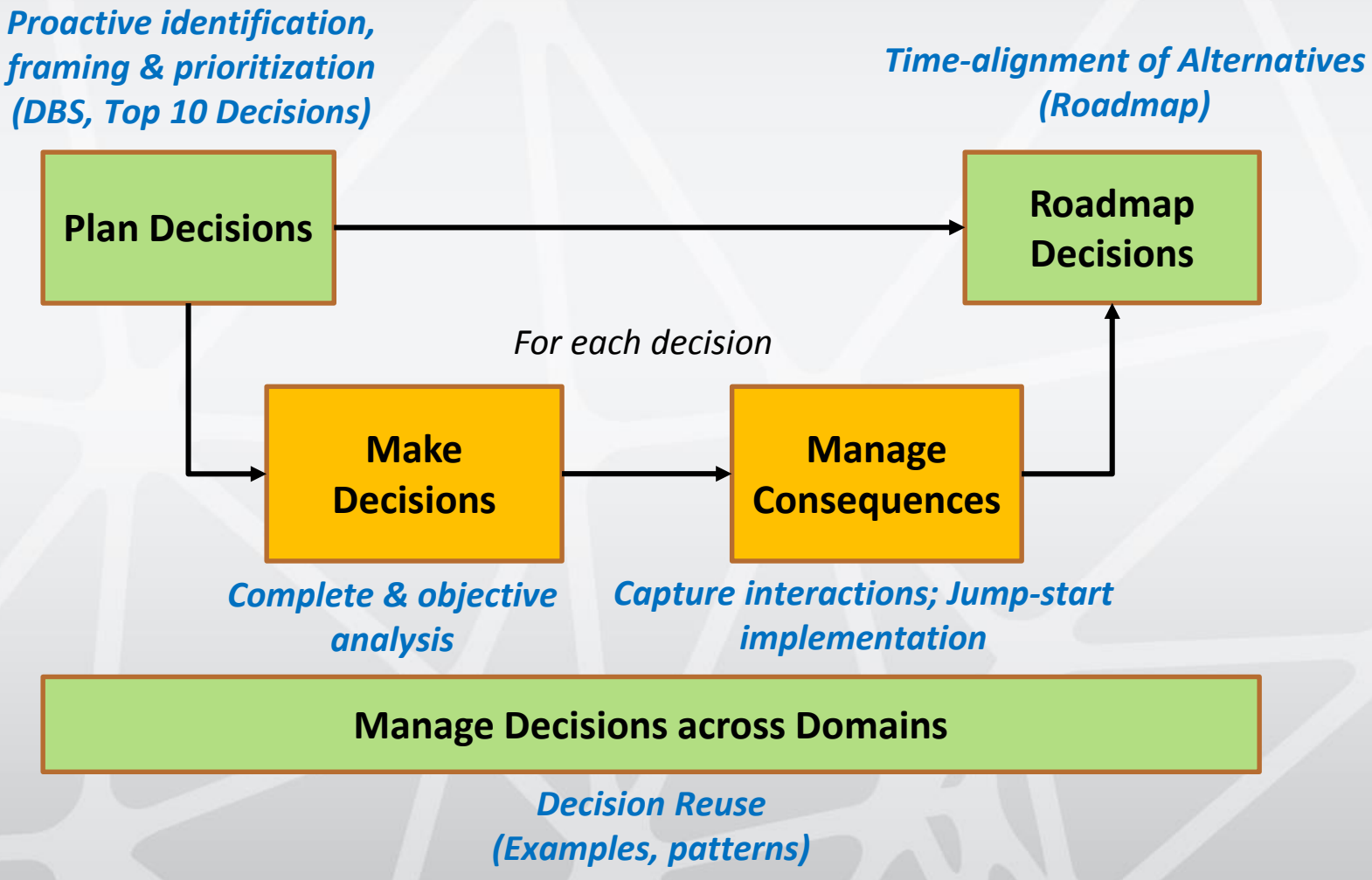
Causes: Decisions Poorly Made



Causes: Decisions Poorly Implemented



Decision Driven® Methods Engine



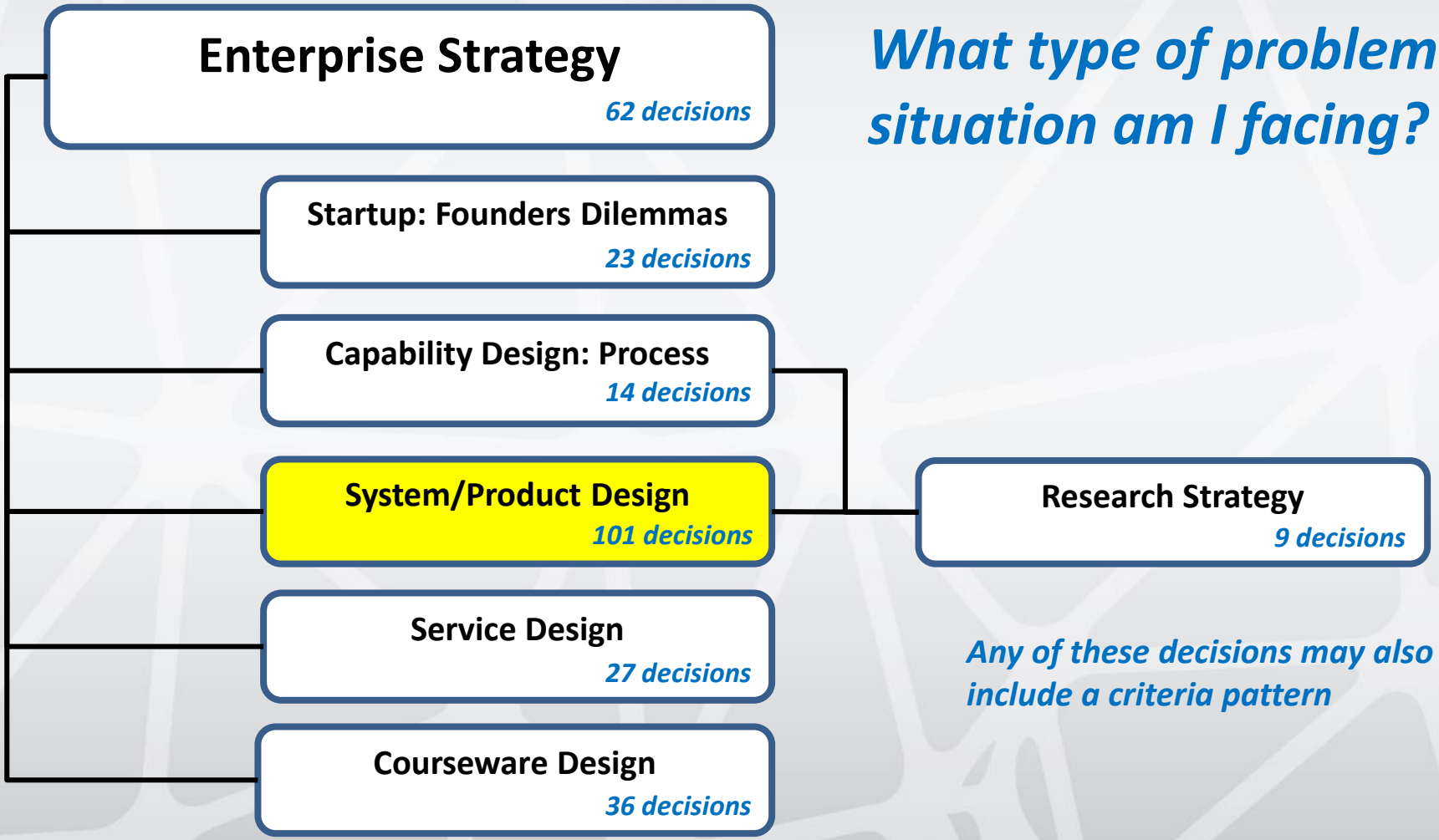
There is a pattern of decisions behind every System Design/Architecture

Exposing this decision pattern gives you a unique set of controls:

- A Decision Breakdown Structure to tame complexity
- An Innovation Framework to visualize new possibilities
- An Evaluation Framework to guide and capture your analysis
- A Collaboration Framework to engage stakeholders
- A Roadmap to fast-forward and align your plans proactively

Never make a decision from scratch!

Decision Patterns

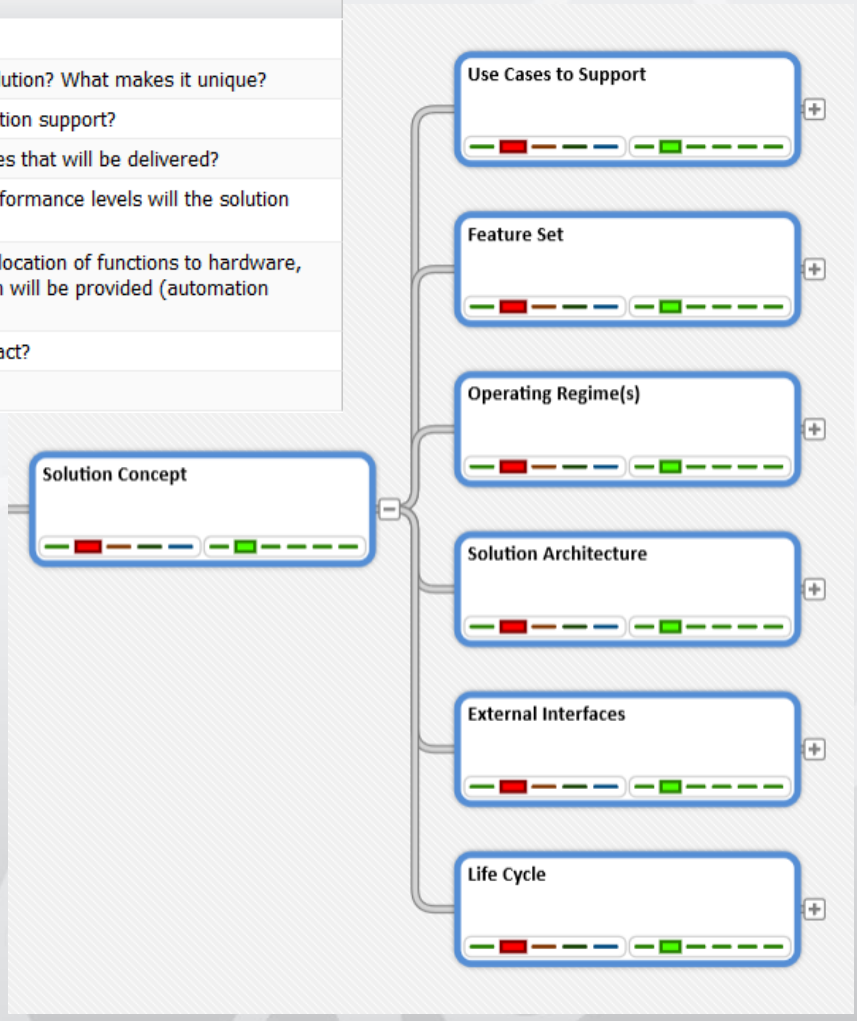


What type of problem situation am I facing?

Any of these decisions may also include a criteria pattern

System/Product Design Decision Pattern: Top Level

Object Number	Decision Title	Decision Description
	Pattern-Product Design	Subsystem Decision Pattern
1	Solution Concept	What is the top-level concept for this system or solution? What makes it unique?
1.1	Use Cases to Support	What use cases (scenarios, missions) will this solution support?
1.2	Feature Set	What are the primary features or groups of features that will be delivered?
1.3	Operating Regime(s)	In what range of conditions, environments and performance levels will the solution operate?
1.4	Solution Architecture	What is the solution's top-level architecture; the allocation of functions to hardware, software or user actions? What level of automation will be provided (automation boundary)?
1.5	External Interfaces	With which external systems will the solution interact?
1.6	Life Cycle	What is the solution's life cycle model?



Each decision is a fundamental question/issue that demands an answer/solution

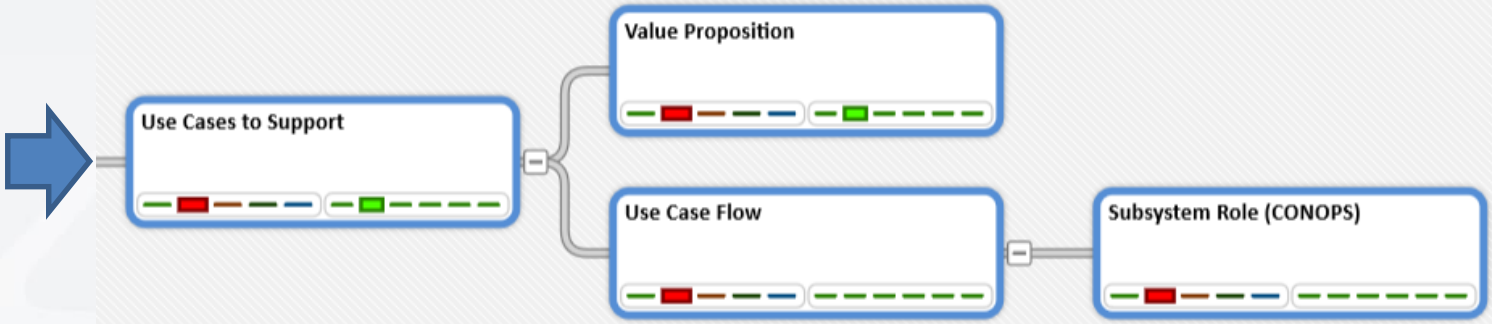
Avoid missing a key decision
Aid to effective decision framing

Use Cases

Add From Pattern

	Name
	Compliance
	Number of users
	Urgency
	Differentiation
	Unmet needs
	Long term needs
	Time to market
	Low cost
	Fit our strategy

Criteria patterns for each decision define success in the eyes of the stakeholders

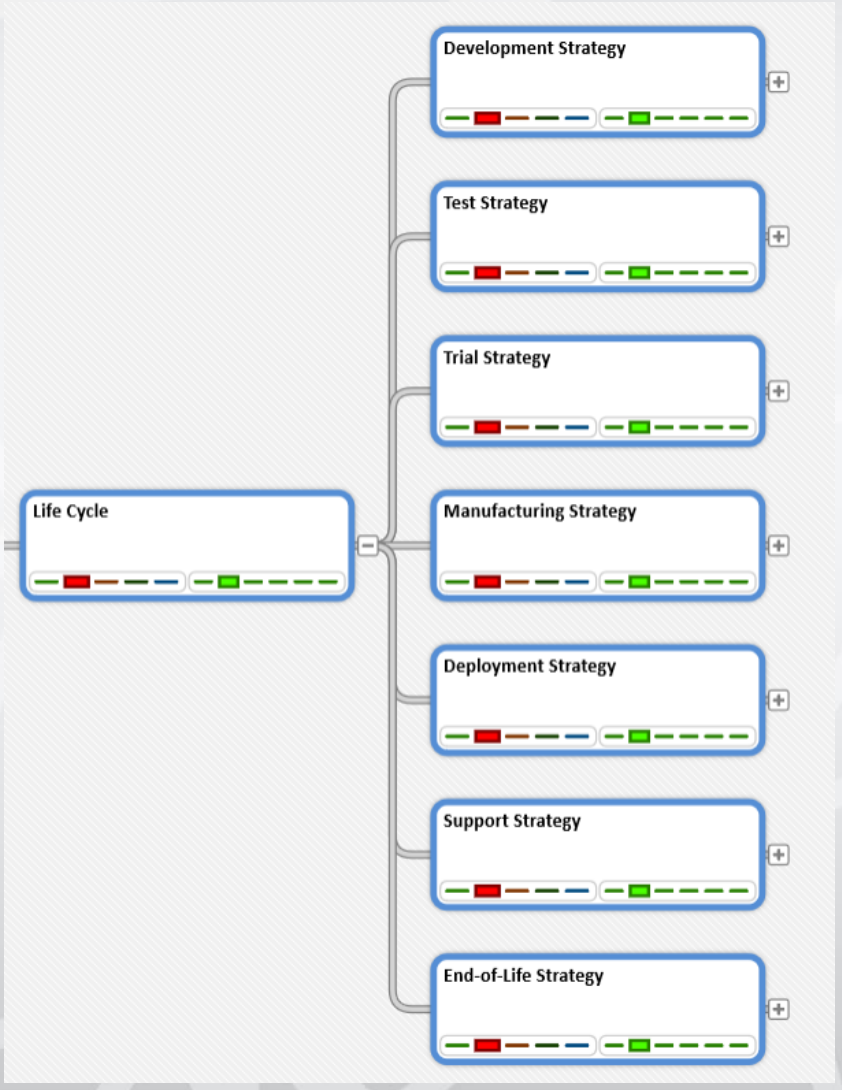


Object Numb	Decision Title	Decision Description
	Pattern-Product Design	Subsystem Decision Pattern
1	Solution Concept	What is the top-level concept for this system or solution? What makes it unique?
1.1	Use Cases to Support	What use cases (scenarios, missions) will this solution support?
1.1.1	Value Proposition	How will the solution deliver value to the end users and customers of this use case?
1.1.2	Use Case Flow	How will this use case be performed? What flow of activities and events will occur?
1.1.2.1	Subsystem Role (CONOPS)	What role with the solution play in this use case? What capabilities and value will it deliver?

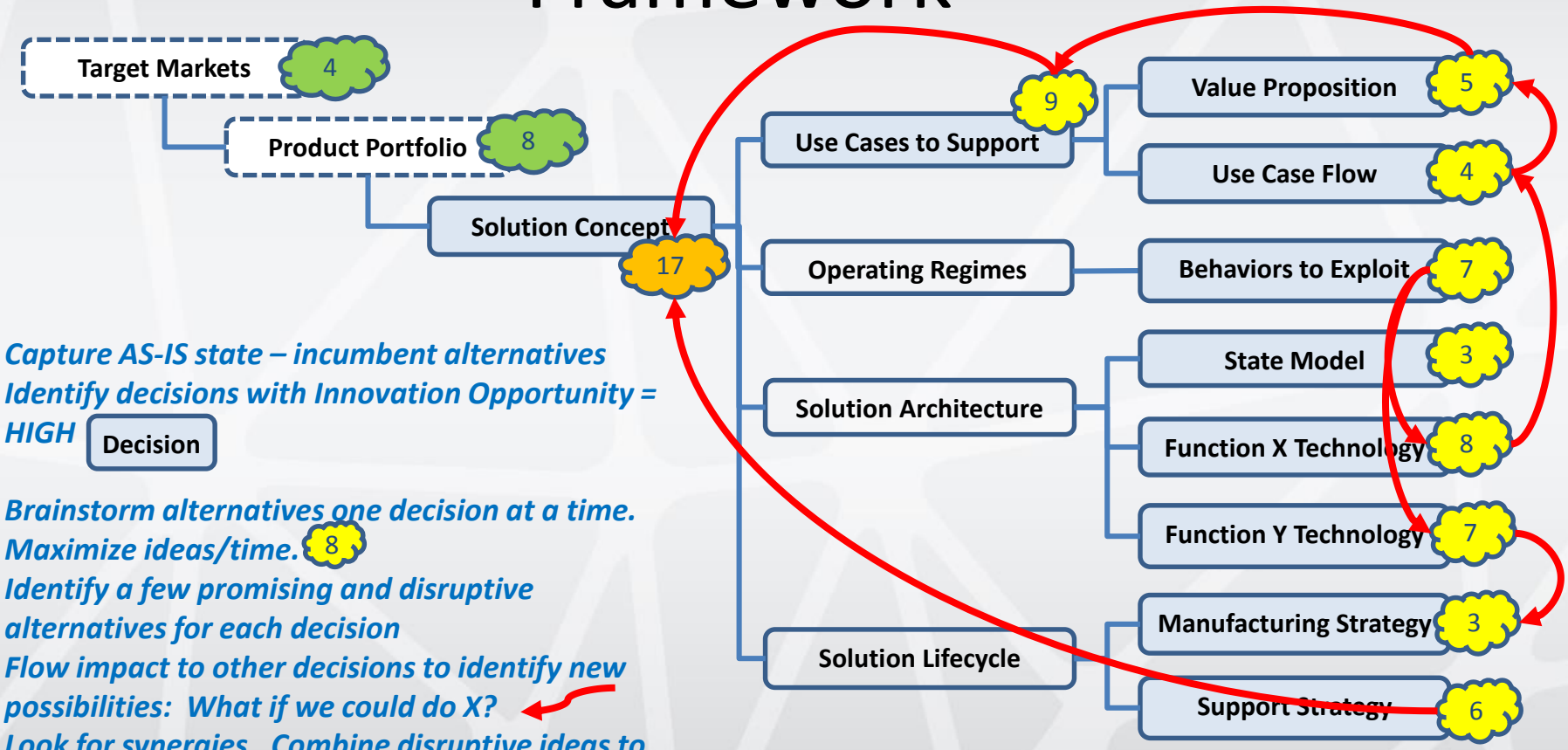
Solution Architecture

System Lifecycle

Object Number	Decision Title	Decision Description
	Pattern-Product Design	Subsystem Decision Pattern
1	Solution Concept	What is the top-level concept for this system or solution? What makes it unique?
1.6	Life Cycle	What is the solution's life cycle model?
1.6.1	Development Strategy	What strategy will we use to develop this solution?
1.6.2	Test Strategy	What strategy will we use to test this solution?
1.6.3	Trial Strategy	What strategy will we use to trial this solution before it is released to market? What set of trials will we conduct? How will these trials interact?
1.6.4	Manufacturing Strategy	What strategy will we use to manufacture this solution?
1.6.5	Deployment Strategy	What strategy will we use to deploy this solution?
1.6.6	Support Strategy	What strategy will we use to support this solution when it is operational?
1.6.7	End-of-Life Strategy	What strategy will we use to replace, disassemble, recycle or dispose of this solution at the end of its life?



Decision Patterns as an Innovation Framework

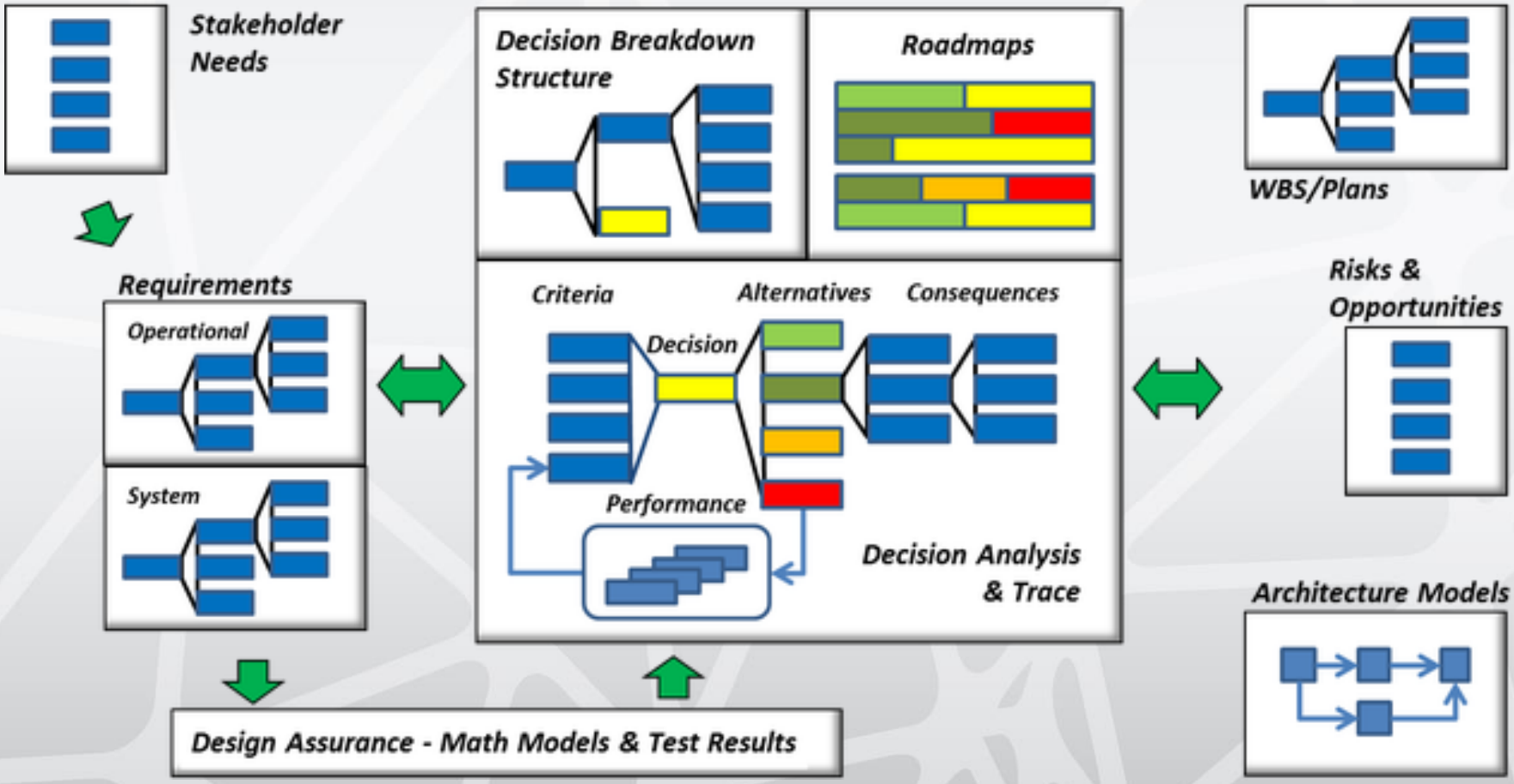


1. *Capture AS-IS state – incumbent alternatives*
2. *Identify decisions with Innovation Opportunity = HIGH* Decision
3. *Brainstorm alternatives one decision at a time. Maximize ideas/time.* 8
4. *Identify a few promising and disruptive alternatives for each decision*
5. *Flow impact to other decisions to identify new possibilities: What if we could do X?*
6. *Look for synergies. Combine disruptive ideas to create new scenarios (futures)* 12
7. *Flow most promising ideas up the DBS structure* Decision 3

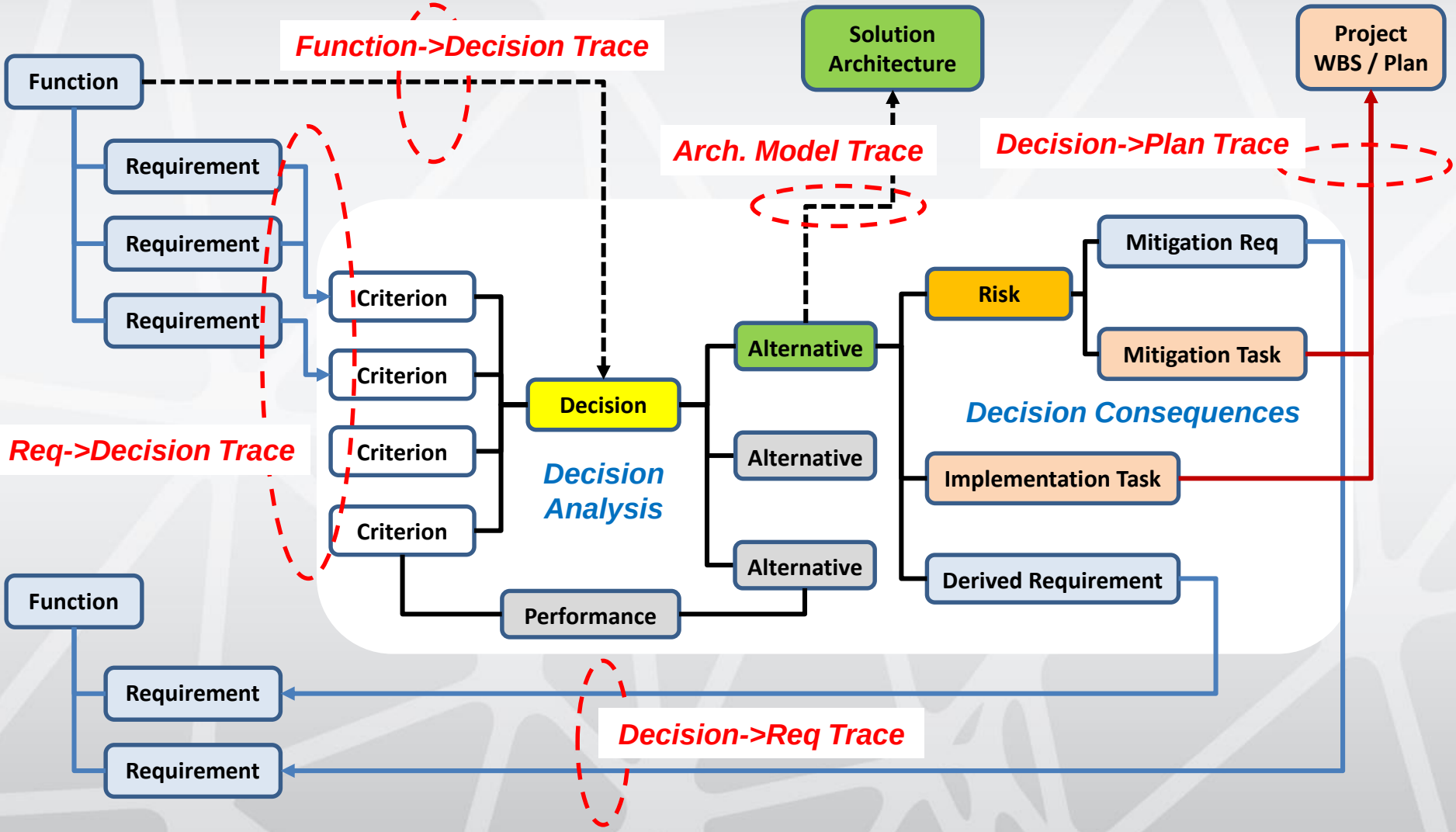
Avoid tunnel vision

Decision-to-Everything Traceability

Decisions are the glue that provide context for other Systems Engineering data



High value decision traceability threads



High payoff shortcuts

Start small – build momentum

- Use a **Top 10 Decisions List** to focus your team on the "vital few" decisions that will drive your project's success
  - Include a **Use Cases to Support** decision to bound the project scope
  - Map the top 3 system functions to a **Technology for Function X** decision
  - Include an **Interface Concept** decision for the primary external interface
- Build an **initial Decision Pattern** from the Top 10 decisions in 2 projects
- Limit scoring effort/data gathering by playing **Knockout**
  - Kill off alternatives that don't meet Threshold values on your top criteria
- For lower priority decisions, **skip the scoring** process. Record alternatives + a brief selection rationale paragraph for the winner. Focus thinking on the **Consequences** of the preferred alternative:
  - The most demanding **derived requirement** that might affect other decisions
  - The most critical **risk** and how to mitigate it **now**!
  - The most complex, costly or time-critical **implementation task**

Additional References

Papers

- Fitch, J.A. 2009. "Exploiting Decision-to-Requirements Traceability, briefing to NDIA CMMI Conference, Denver, CO (US), 9 November, 2009.
- Mendonza, P. and Fitch, J.A. 2012. “Decision Management (DM) as the engine for scalable, cross-domain Systems Engineering”. Paper presented at 22nd Annual International Symposium of the International Council on Systems Engineering, Rome, Italy, 9-12 July.

Blog posts

- https://decisiondriven.wordpress.com/2008/10/10/all-faults-are-decision-faults/
- https://decisiondriven.wordpress.com/2015/08/22/stimulate-grand-challenge-innovation-by-clear-problem-framing/
- https://decisiondriven.wordpress.com/2008/08/11/decision-patterns-unleash-creativity/
- https://decisiondriven.wordpress.com/2008/06/25/multi-decision-innovation-framework/
- https://decisiondriven.wordpress.com/2008/10/17/failure-to-manage-decision-consequences/
- https://decisiondriven.wordpress.com/2008/10/04/all-requirements-are-derived-requirements/
- https://decisiondriven.wordpress.com/2008/09/08/top-ten-decisions-list/