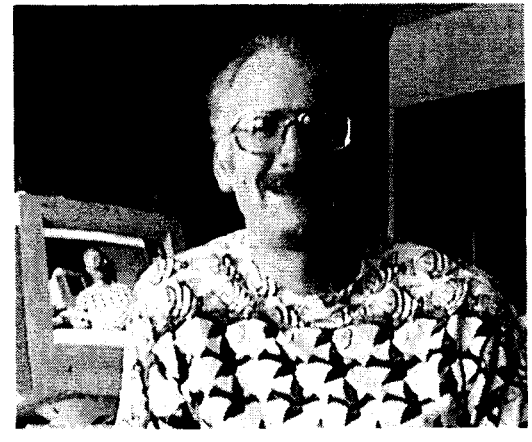
A Risk-centric Decision Process

Martin S. Feather

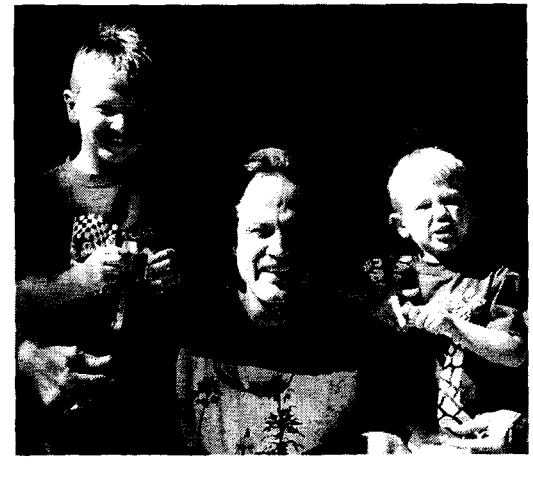
Jet Propulsion Laboratory
California Institute of Technology

Martin.S.Feather@Jpl.Nasa.Gov



This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.
Funded by NASA's Code Q (FDDP program and IV&V ARRT task) and Code R (ECS program).

Reports on use of the DDP tool, the lead for which is JPLer Steve Cornford →



http://ddptool.jpl.nasa.gov

PART I: The Process
PART II: Example of application to technology infusion

NASA's mission challenges

- **Groundbreaking**
  - New mission concepts, new technologies (autonomy, agents, ...), unknown environments

  Past experience provides only a partial guide
- **Multi-disciplinary**
  - Navigation, telecom, fault protection, commanding/sequencing, ...
  - Cross-coupled interactions

  No individual is an expert in all areas

  No individual can juggle all the details at once
- **Resource constrained**
  - Schedule (e.g., celestial mechanics dictate launch date window), budget, testbeds,
  - CPU, RAM, data storage, bandwidth,

  Many risks that, if untamed, lead to cancellation, underachievement, or even loss of mission
- **Need good decisions early**
  - Cost of correcting a bad decision escalates over time

  Early on, lack information (e.g., detailed design) on which to base decisions

What do you want?

"Objectives"
"Requirements"
"Goals"

Mick Jagger
(Rolling Stones):

"You can't always get what you want"

Descoping – strategic abandonment of objectives.

Reprioritize objectives; primary, secondary…

Determine attainment if given additional resources ($, mass, …)

What can get in the way?

"Risks"
"Failure Modes"
"Defects"

Dr. Michael Greenfield
(NASA HQ):

"Risk as a resource"

Trade risk for other resources.

Use risk as an intermediary between other resources.

What can you do about it?

"Mitigations"
"Solution Options"
"Preventions, Analyses, Controls,Tests – PACTs"

Matt Landano
(JPL):

"Do the right thing & do it right"

Can't afford all possible mitigations, so must choose judiciously.

Know the purpose(s) of each mitigation.

Objectives	Risks	Mitigations
~~Return data~~	~~Software bug ridden~~	~~Become CMM level 3~~

Insufficient detail for decision making. Elaborate!

In flight s/w upgrades	Requirements risks	Requirements practices
Code/Data separable	Unstable	Documented
Real-time control loops	Incomplete	Formal CM
Sync to external clock	Unclear	Peer review
Tolerate memory errors	Invalid	Formal inspections
Run time memory =…	Infeasible	Formal reviews
Storage = …	Unprecedented	Criticality analyses
CPU utilization = …	Large size/complex	Verifiability check
…	…	…

Risk x Objective :
How much **of objective will be lost if risk occurs? – "Impact"**

Mitigation x Risk:
How much **will risk be reduced if mitigation applied? – "Effect"**

Elaborate enough to be able to say by *how much*

Day 1 – day of the pessimists!

Objectives – *what you want*

Risks* – *what could occur to detract from attaining objectives*

Impact (Objective x Risk) - *proportion of the Objective lost if Risk occurs*

* ***All*** risks, ***including*** those whose mitigation is planned:

- Makes available for scrutiny explicit assertions of risk reduction
- Allows risk and its mitigation to be involved in trades
- Reveals dependencies on mitigations (what if can't do it on time?)

Day 2 – day of the optimists!

Mitigations – *what could be done to reduce risk*

Effect (Mitigation x Risk) – *proportion by which Mitigation reduces Risk*

Experts' estimates, past experience if available, models & simulations…

Day 3 – day of the realists!

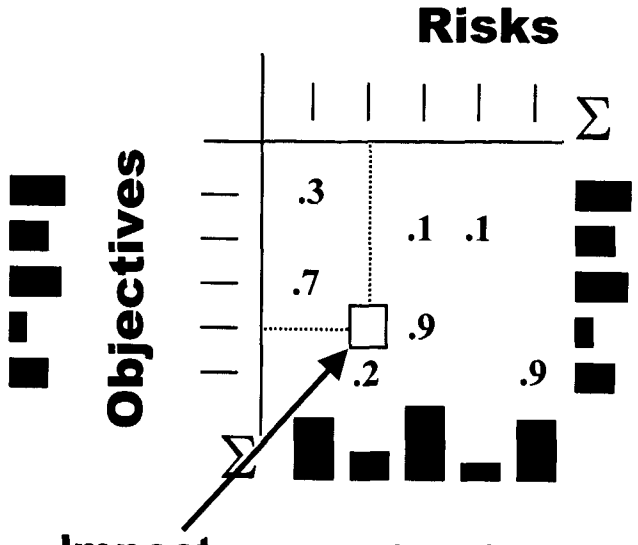
Select – Mitigations to perform
Objectives to discard
Resources to ask for

Decision-making guided by accumulated information

Getting the right people is key!!!

Mission scientists, technologists, relevant disciplines' engineers, assembly/integration, testing, QA, operation, programmatics

Day 1 – day of the pessimists!



Impact – proportion of objective lost if risk occurs

Sum the rows: how much each objective is "at risk".
Sum the columns: how much each Risk causes loss of Objectives.

Objectives – *what you want*
have **weights** (their relative importance)

Risks – *what could occur to detract from attaining objectives*
have **a-priori likelihoods** (how likely they are to happen *if not inhibited by Mitigations*), usually left at the default of 1 (certain!)

Impact (Objective x Risk) - *proportion of the Objective lost if Risk occurs*
Combine *additively*: I1 & I2 = I1 + I2 (therefore objectives can be more than 100% killed!)

Disagreement about an impact number usually (always?) resolved by refinement of Objective and/or Risk

Day 2 – day of the optimists!

Mitigations

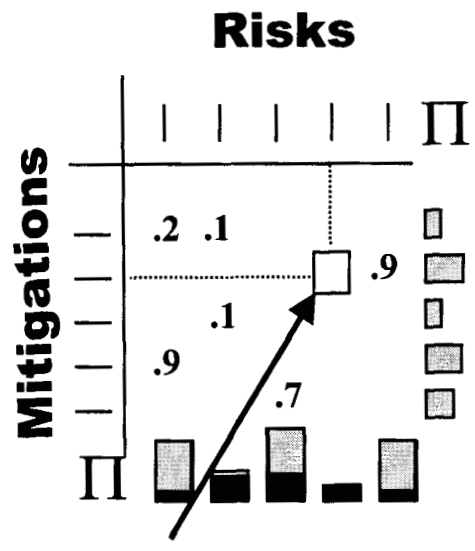
– *what could be done to reduce risk*
have **costs** ($, schedule, high fidelity test beds, memory, CPU, ...)
have **type** (prevention, detection – *reduce risk likelihood*; alleviation – *reduce risk impact*)
have **status** applied / not applied: major purpose is to decide which to apply!

Effect (Mitigation x Risk) – *proportion by which Mitigation reduces Risk*
Combine as serial "filters":
E1 & E2 = (1 – (1-E1)*(1-E2))

e.g., a 0.8 effectiveness Mitigation catches 80% of incoming Risk ,
a 0.3 effectiveness Mitigation catches 30% of incoming Risk ;
100% -> 20% -> 14% so together have 86% effectiveness

(1 – (1 – 0.8)*(1 – 0.3)) = (1 – 0.2*0.7) = (1 – 0.14) = 0.86

Note: a law of diminishing returns as apply additional Mitigations

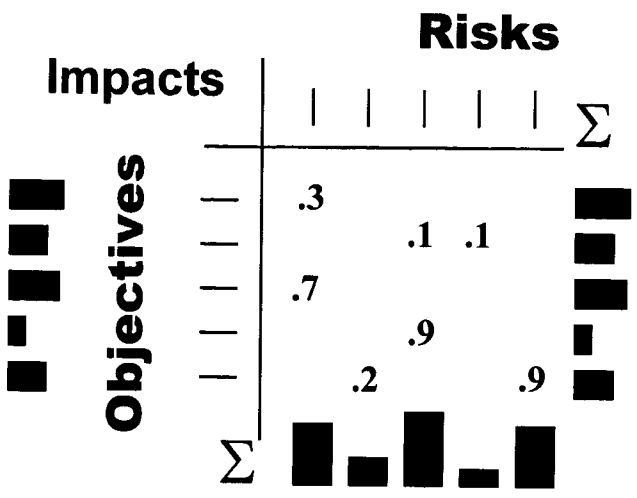


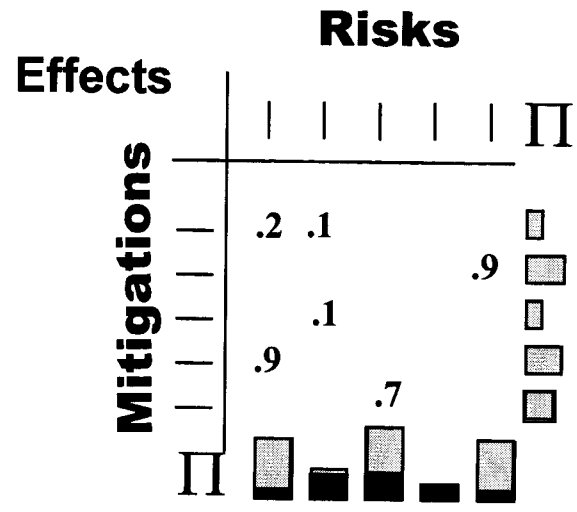
Effect – proportion by which risk reduced if mitigation applied

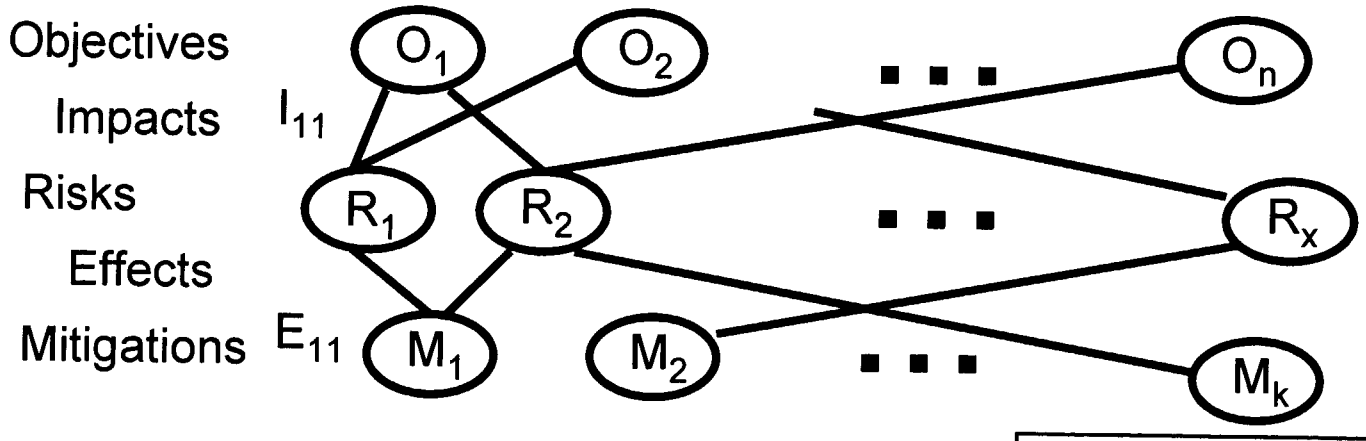
Sum the rows: how much each Mitigation reduces Risks; "solo" or "delta".
Sum the columns: how much each Risk detracts from Objectives (1) when Mitigations off, (2) when Mitigations on.

Note: some mitigations can make risks *worse* (increase likelihood or impact)!

Day 3 – day of the realists!

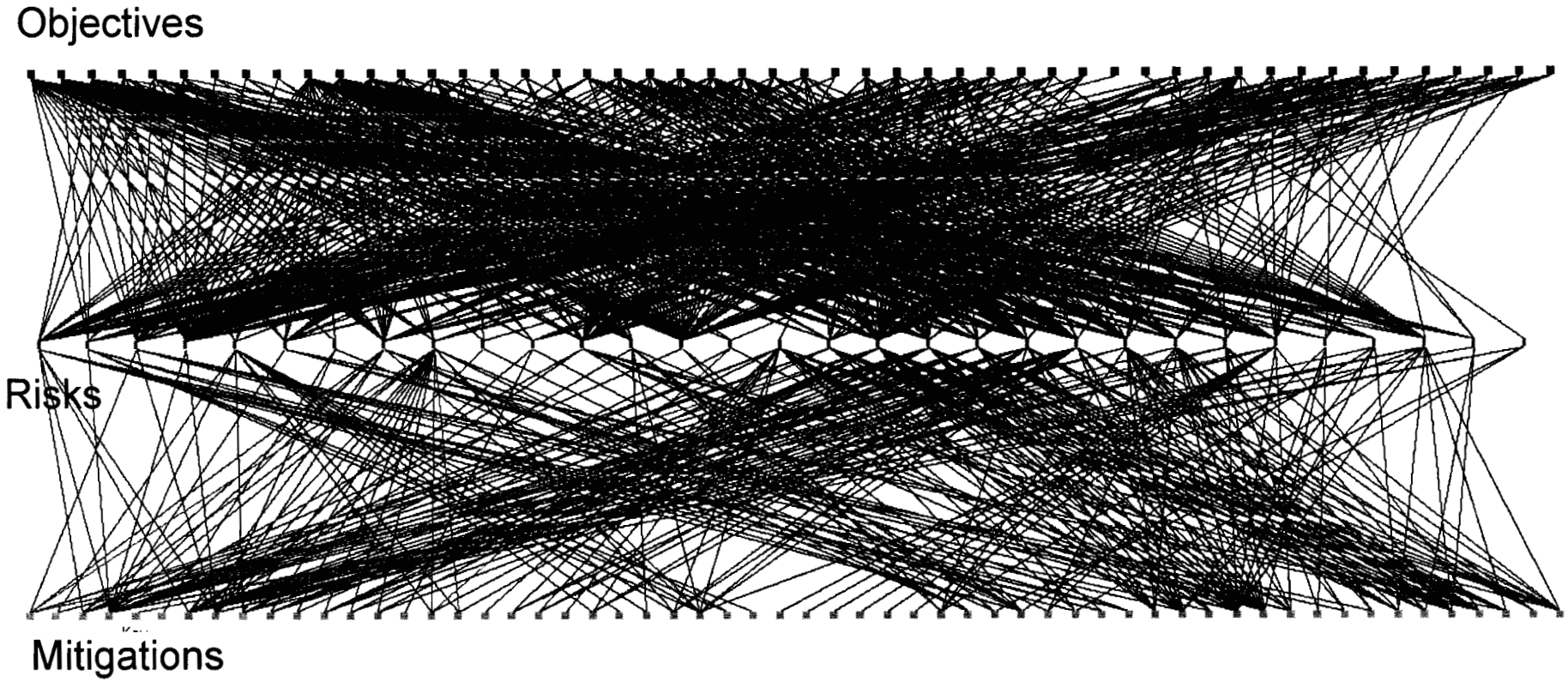






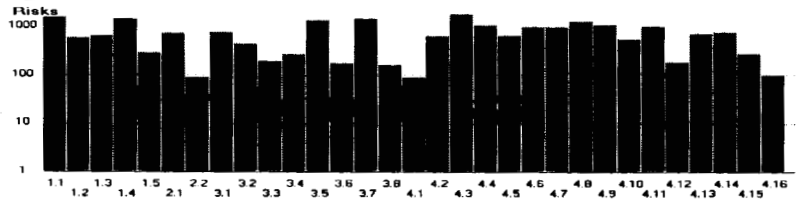
Goal: select mitigations so as to cost-effectively reduce risk

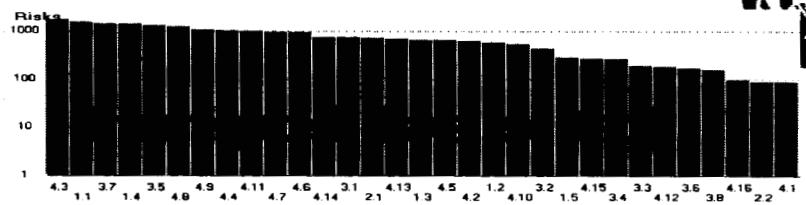
Typical DDP information set: 50 objectives, 31 risks, 58 mitigations

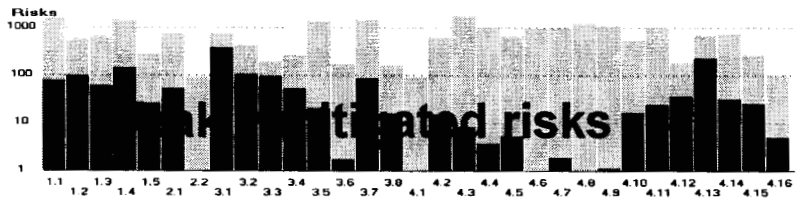


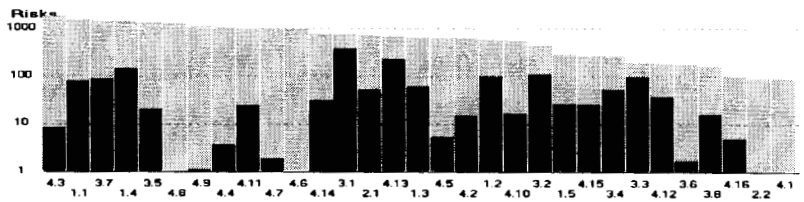
DDP process and custom tool enables models of this scale to be built and used effectively

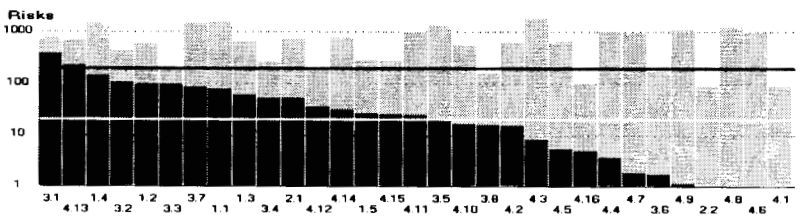
Visualizations of aggregate information



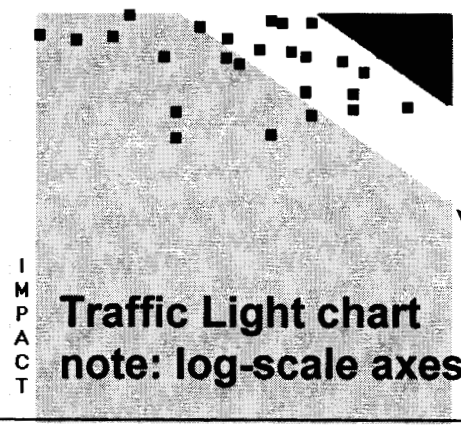








Red = remaining risk
Green = mitigated risk (but at a cost)



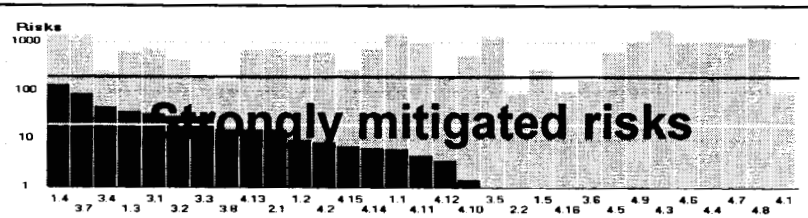
Traffic Light chart
note: log-scale axes

$ Objvs: 2,120 out of 3,610

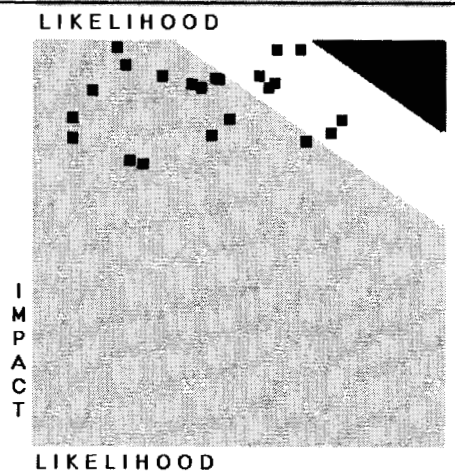
	Cost
ALL	4,740,000
Checked	396,000

$-o-meter

boundary = iso-risk line!



Strongly mitigated risks

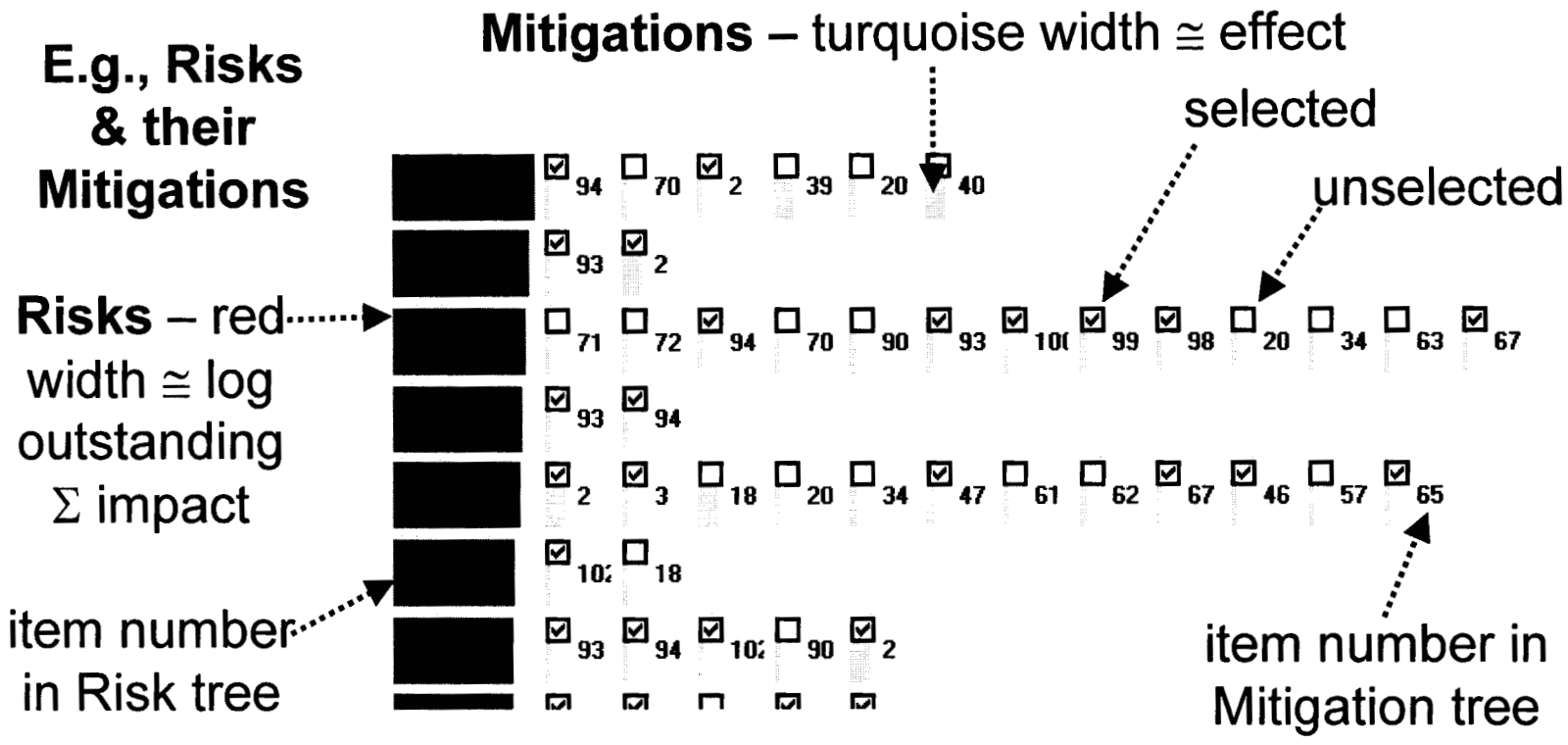


$ Objvs: 3,110 out of 3,610

	Cost
ALL	4,740,000
Checked	996,000

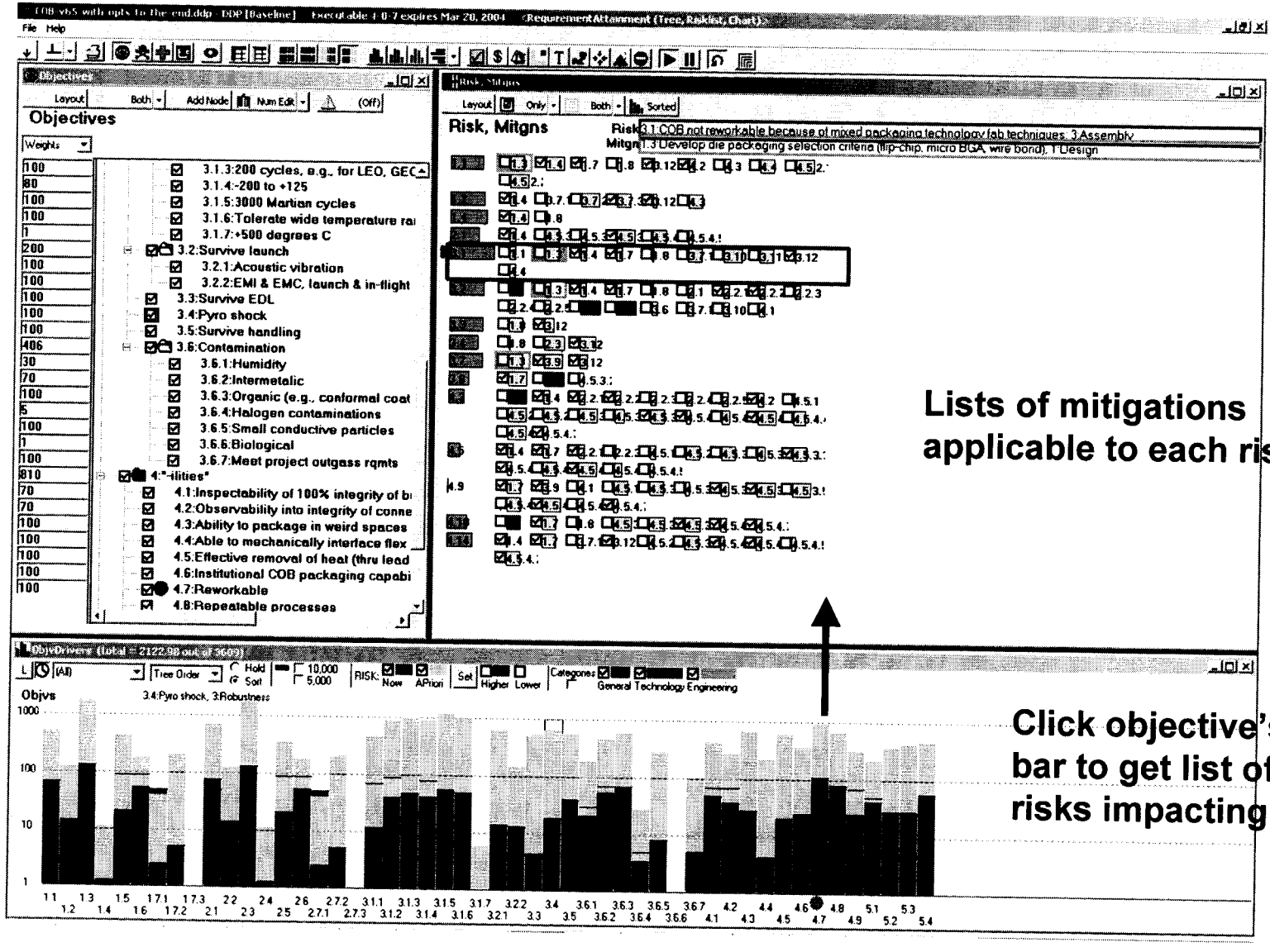
Goal: select mitigations so as to cost-effectively reduce risk

"Stem-and-leaf"(*) visualization of DDP sparse matrices



(*) Tufte attributes these to John W. Tukey, "Some Graphical and Semigraphic Displays" Their usage was introduced into RBP (DDP without numbers) by **Denise Howard** & **Chris Hartsough**, extended further by us in DDP.

Typical DDP screenshot



Lists of mitigations applicable to each risk

Click objective's bar to get list of risks impacting it

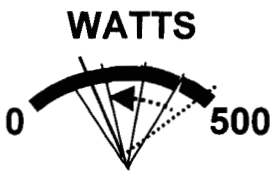
Examples of DDP-assisted improvements

- Cost & Time Saved

$$$

  - At least two instances of savings > $1M (per study cost: $10K - $30K)
    - E.g., Storage technology study revealed problematic overly-stringent requirement, whose removal permitted dramatic cost & time savings

- Designs Improved

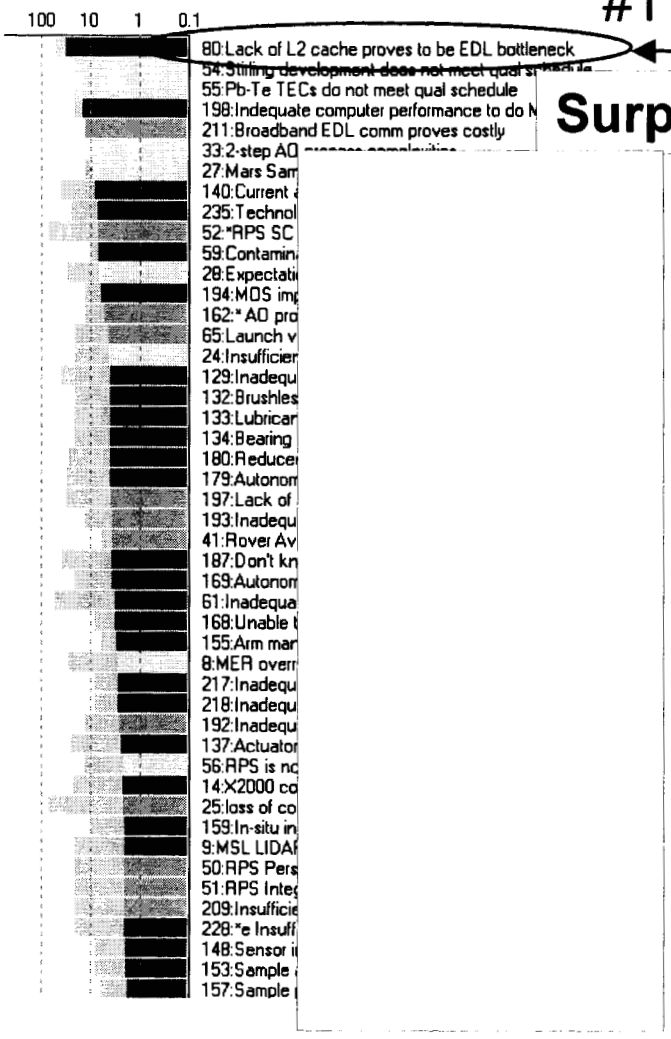


  - Savings of critical resources (power, mass, ...) seen in comparison of designs before & after DDP sessions
    - E.g., Flight experiment redesign: power needs decreased by 68%, mass decreased by 13%, cost decreased by 9%, major category of risk changed from architectural to well-understood design

- Reliability and Safety Increased

  - Non-obvious significant risks identified and mitigated
    - E.g., Lander – Sufficient L2 cache size on computer identified as critical to successful EDL

Flight Project risk insights from DDP



#1 risk:

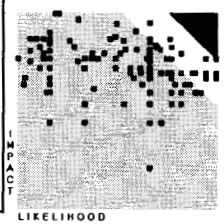
Lack of L2 cache proves to be EDL bottleneck

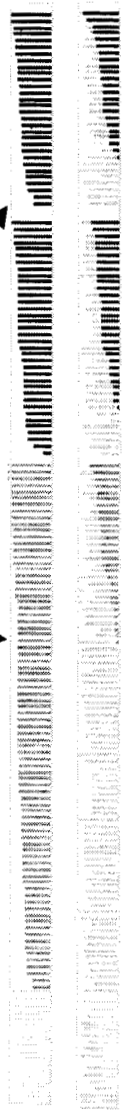
Surprise!

flight project application

- Large number of risks (>150) and mitigations (>300)
- Mix of several *kinds* of risks (Technology, Engineering, Programmatic, …)

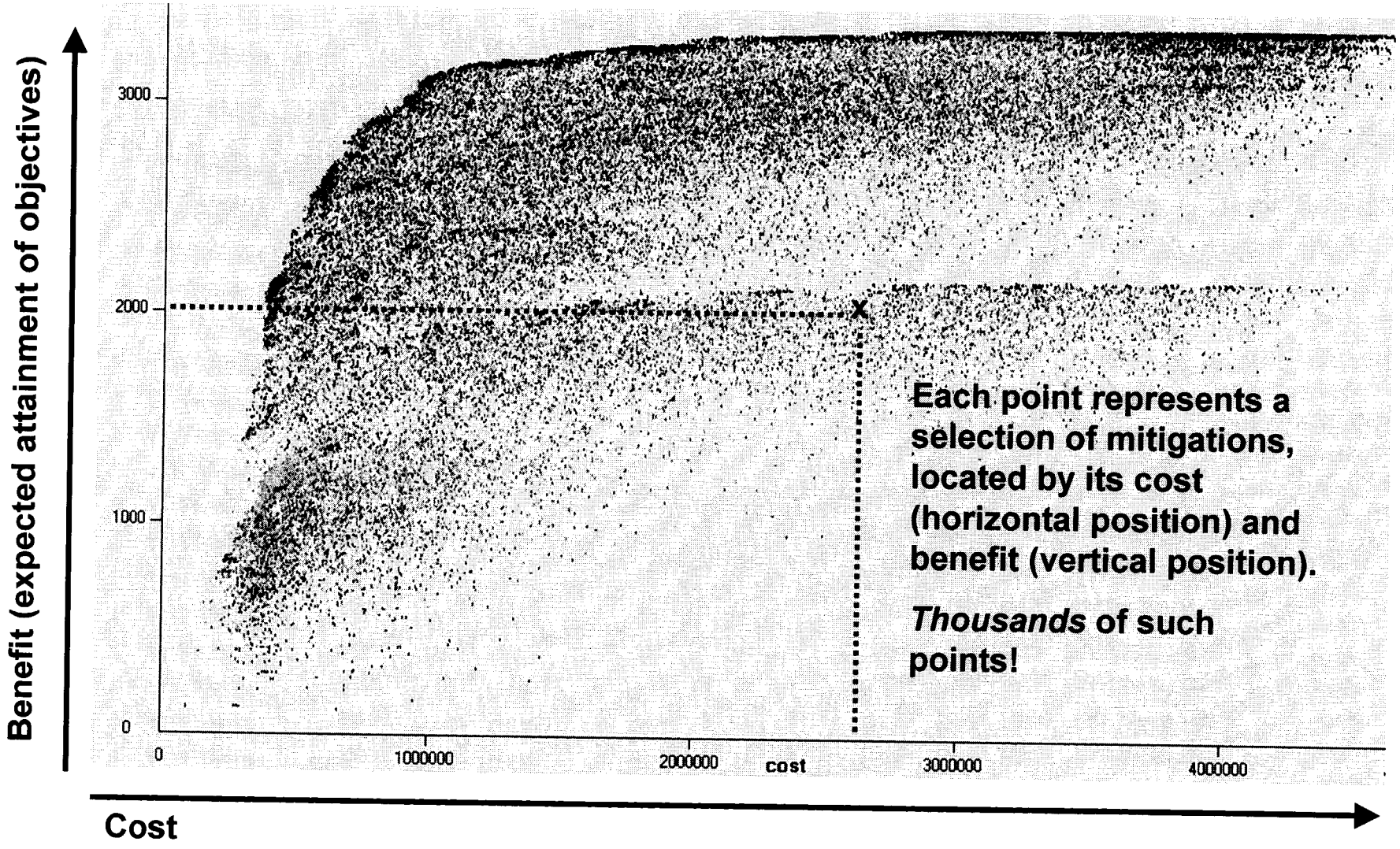
DDP enables both big-picture understanding and detailed scrutiny



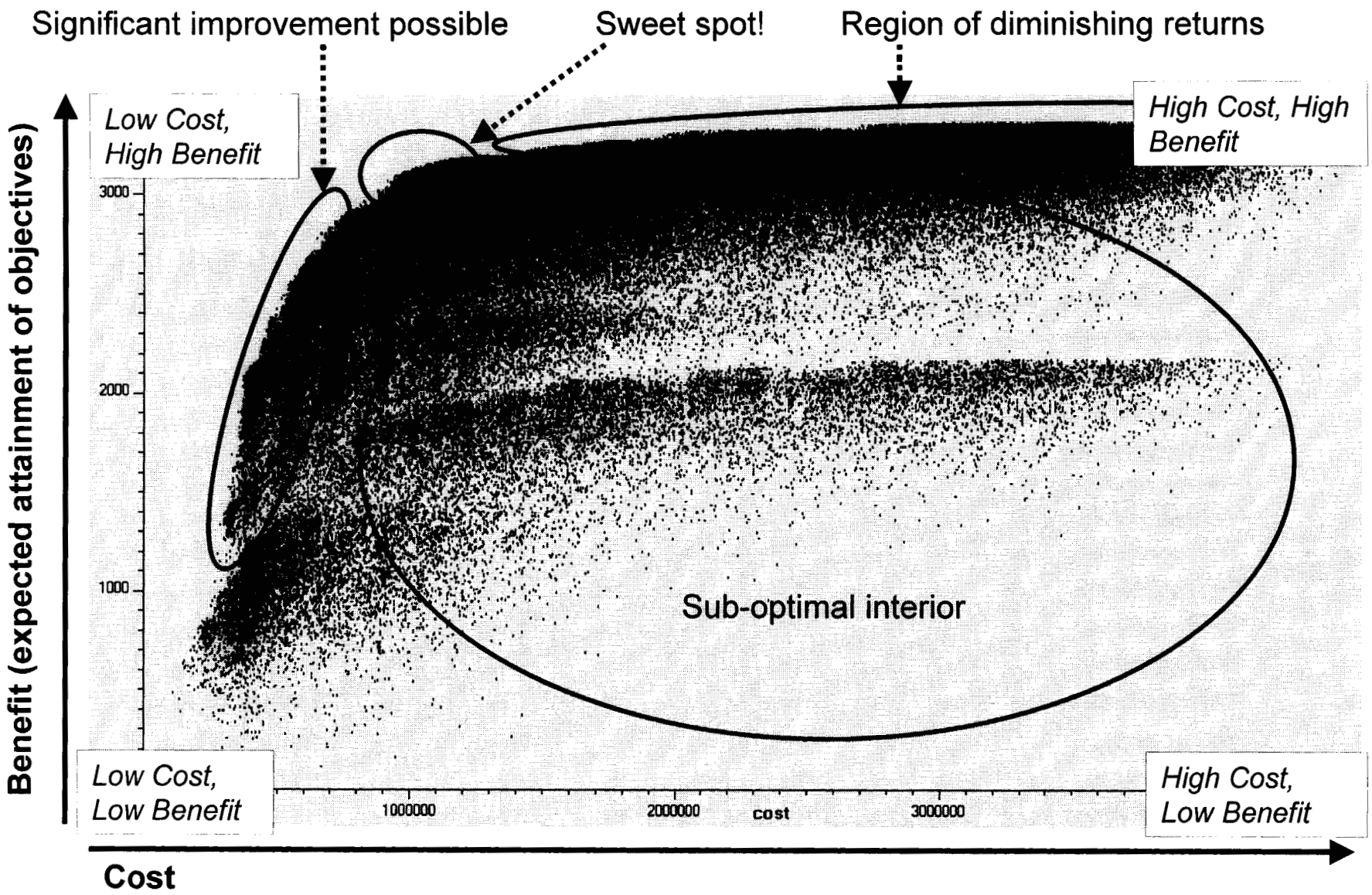


Cost-Benefit trade space

58 mitigations = 2^{58} (approx 10^{18}) ways of selecting.
Simulated Annealing used to search for near-optimal selections.



Cost-Benefit trade space insights



DDP Timeline

Code Q "Failure Detection and Prevention Program" (1998-?)

PEMs: Tim Larson, Kelly Moran
PI: Steve Cornford

Code Q / IV&V "Advanced Risk Reduction Tool" (1999-2002)

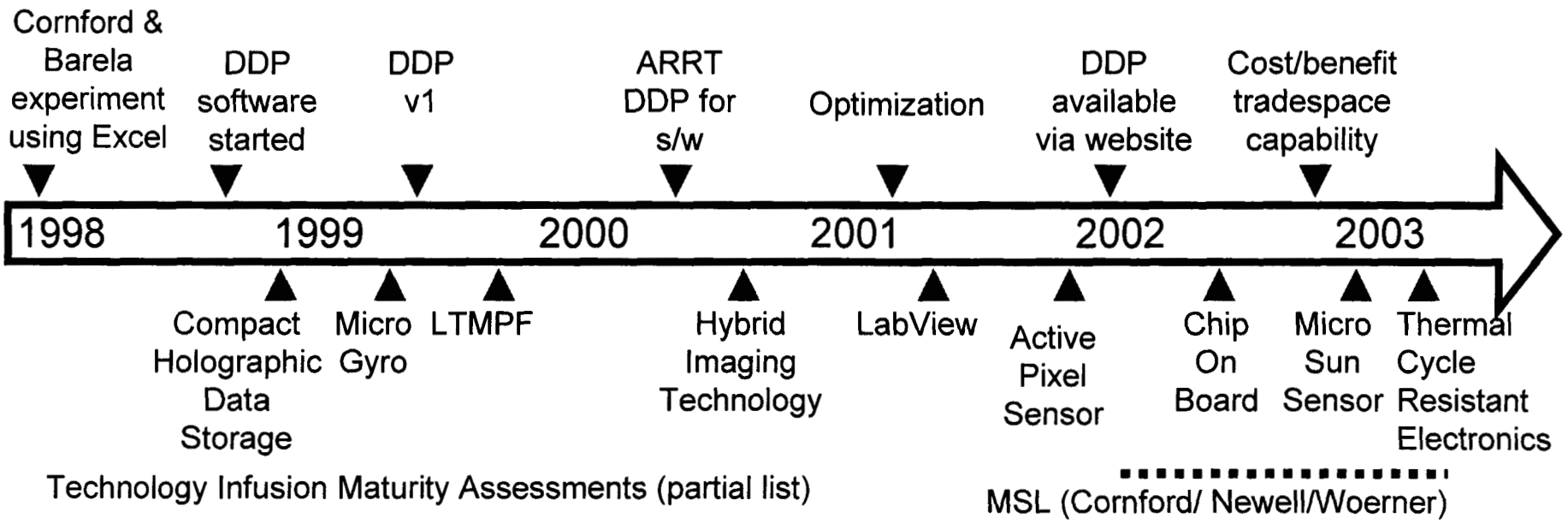
PEMs: John Kelly, Burton Sigal, Allen Nikora
PIs: John Kelly, Martin Feather

DDP

Code R "Engineering of Complex Systems" (2002-?)

Level 2 Manager: Stephen Prusha

PIs: Ken Hicks (Risk-Based Design),
Steve Cornford (Risk Workstation),
Martin Feather (S/W Risk Characterization & Mitigation)



Reflections: Probabilistic Risk *Reduction*

PRA

Probabilistic Risk **Assessment** computes risk from knowledge of:

- Individual components' reliabilities (e.g., MTBF)
- System architecture (e.g., Fault Tree)

Benefits

- Calculate system risk / reliability
  when system too expensive/complex/long lived/critical to directly measure
- Gain insight into system vulnerabilities
  (e.g., cut-sets indicate key contributors to failure)

PRR

Probabilistic Risk **Reduction** computes risk from knowledge of:

- Individual risk mitigation activities (e.g., inspection, unit testing)
- Potential risks - both product risks and process risks (e.g., late/over-cost)
- Quantitative assessments of mitigations' effectiveness (at reducing risk) and risks' impacts (on system objectives)

Benefits

- Calculate system risk / reliability
  when development process key system assessment (e.g., software)
- Select mitigations to most cost-effectively reduce risk
- Identify problematic objectives (those with expensive-to-reduce risks)
- Gain insight into risks (reduction of, remaining) & mitigations (purpose)

PART I: CONCLUDING SUMMARY

- Information: make use of information available early in lifecycle
  - Combine knowledge from experts and past experience
  - Accommodate both evidence and estimates
- Process: gather the right information the right way
  - Objectives, including their relative importance
  - Risks, and by how much they impact objectives and requirements
  - Mitigations, and by how much their use would reduce risk
- Tool support: effectively handle voluminous amounts of information
  - Capture experts' knowledge on-the-fly during intensive sessions
  - Present information through cogent visualizations
  - Derive additional knowledge via calculation and search

For more details, please see:
[Feather & Cornford, 2003] M.S. Feather & S.L. Cornford "Quantitative risk-based requirements reasoning", to appear in *Requirements Engineering* (Springer), published online 25 February 2003, DOI 10.1007/s00766-002-0160-y.

http://ddptool.jpl.nasa.gov

Steven L. Cornford@Jpl.Nasa.Gov Martin.S.Feather@Jpl.Nasa.Gov

PART II:
Example of application to technology infusion

Next four slides show application of this risk-based reasoning to the "Technology Infusion" problem.
For illustration, model checking technology is considered.

Using the DDP risk-centric decision process, aspects application of model checking (for what, and by whom) are represented as DDP "Objectives", impediments to these are represented as DDP "Risks", and solutions to those impediments are represented as DDP "Mitigations".
Quantitative (albeit coarse-grained) estimates of *by how much* each impediment inhibits model checking, and *by how much* each solution addresses each impediment allows for scrutiny of the net effects of decisions.

JPL Technology infusion as a "risk" centric decision problem

"Objectives"

1:artifacts
- 1.1:rqmts
  - 1.1.1:consistency
  - 1.1.2:completeness
  - 1.1.3:validation
  - 1.1.4:test case generation
- 1.2:design
  - 1.2.1:requirements verification
  - 1.2.2:bug finding
- 1.3:code
  - 1.3.1:requirements verification
  - 1.3.2:unit testing
  - 1.3.3:integration testing
  - 1.3.4:structural, defect detection
  - 1.3.5:functional errors, bug finding
  - 1.3.6:timing errors
- 1.4:models
  - 1.4.1:sanity checking
  - 1.4.2:validation

2:who uses the tool
- 2.1:developers
- 2.2:test engineers
- 2.3:QA
- 2.4:IV&V
- 2.5:model checking gurus

"Risks"

impediments to use of model checking

1:Technical issues
- 1.1:state space explosion
- 1.2:slow turnaround time
- 1.3:notation that mc can't handle
  - 1.3.1:design notation incompatible with model checking
  - 1.3.2:property notation incompatible with model checking
- 1.4:challenging generation of environment models
- 1.5:unknown what applications domains are suitable
- 1.6:unknown how much work it takes
- 1.7:complexity of deciphering error traces

2:Social Issues
- 2.1:resistance to learning new languages and tools
- 2.2:need to have specification expertise
  - 2.2.1:modeling expertise (how to build the model)
  - 2.2.2:LTL etc expertise (how to specify the properties)
- 2.3:well documented requirements are lacking
- 2.4:beneficiaries not the ones who do it
- 2.5:large effort of applying model checking
- 2.6:knowledge of the application domain is required

Note the wide range of issues taken into consideration; our thesis is that technology infusion can and does fail for a variety of reasons.

Choose one from each group. Group 1 is what model checking is to be used for (e.g., 1.1.2, completeness checking of requirements), Group 2 captures the general level of expertise of the users of model checking

JPL Technology infusion as a "risk" centric decision problem (continued)

"Risks" (repeated from preceding slide)

impediments to use of model checking

1:Technical issues
- 1.1:state space explosion
- 1.2:slow turnaround time
- 1.3:notation that mc can't handle
  - 1.3.1:design notation incompatible with model checking
  - 1.3.2:property notation incompatible with model checking
- 1.4:challenging generation of environment models
- 1.5:unknown what applications domains are suitable
- 1.6:unknown how much work it takes
- 1.7:complexity of deciphering error traces

2:Social Issues
- 2.1:resistance to learning new languages and tools
- 2.2:need to have specification expertise
  - 2.2.1:modeling expertise (how to build the model)
  - 2.2.2:LTL etc expertise (how to specify the properties)
- 2.3:well documented requirements are lacking
- 2.4:beneficiaries not the ones who do it
- 2.5:large effort of applying model checking
- 2.6:knowledge of the application domain is required

"Mitigations"

help address impediments to use of model checking

1:tools for abstraction
2:tools for translation into mc Ls
3:hiring PhDs
4:training application engineers
5:increase computing resources
- 5.1:chips get faster
- 5.2:more memory
- 5.3:parallel h/w

6:short training course for LTL el al
7:emphasize the unique role that m/c can play
8:cost of failure a driver
9:specification patterns for properties
10:model checking provided as a "service"
11:Develop cost models
12:Case studies
13:Baselining & benchmarking
14:(Funded) partnerships with projects
15:Search heuristics
16:custom model checkers for programming Ls
17:compositional m/c
18:marketing
19:design for verification
20:tools for visualizing results
21:include mc into existing toolset
22:pick customers

Mitigations (rows) x Risks (columns)

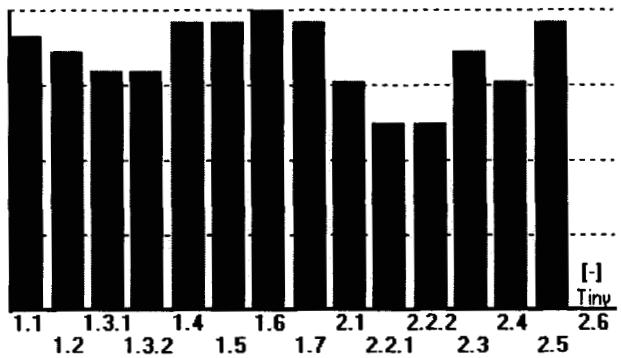
Numeric entries in cells indicates proportion by which mitigation (if applied) would reduce risk; blank = 0

		Risks	[-]Technical issues								[-]Social Issues						
		Risks	state space explosion	slow turnaroun time	[redacted]		challengir generatior of environme models	unknown what applicatio domains are	unknown how much work it takes	complexity of decipherir error	resitance to learning new languages	[-]need to have specification expertise		well document requireme are	beneficiar not the ones who	large effort of applying model checking	knowledg of the applicatio domain is required
		Risks			design notation incompatil with model	[redacted]						modeling expertise (how to build the model)	LTL etc expertise (how to specify the properties				
Mitgns	Mitgns	counts	12	14	7	7	6	12	10	13	10	9	8	7	7	10	11
tools for abstraction		8	0.7	0.7			0.7	0.1		-0.3		0.3				0.3	0.1
tools for translation into mc Ls		10		0.9	0.95			0.1	0.5	0.7	0.99	0.99			0.5	0.9	0.7
hiring PhDs		14	0.8	0.1	0.9	0.9	0.8	0.6	0.3	0.99	0.9	0.8	0.99	0.1		0.3	0.1
training application engineers		15	0.3	0.7	0.9	0.99	0.9	0.9	0.3	0.8	0.1	0.99	0.99	0.5	0.99	0.3	0.99
[-]increase computing resources	chips get	3	0.1	0.7						0.1							
	more	2	0.6	0.3													
	parallel	2	0.7	0.7													
short training course for LTL el al		5				0.99							0.99	0.2	0.2		0.3
emphasize the unique role that m/c can play		2						0.7			0.3						
cost of failure a driver		2						0.7			0.3						
specification patterns for properties		6		0.1		0.7					0.3		0.7	0.3	0.2		
model checking provided as a "service"		14	0.8	0.7	0.9	0.9	0.8	0.6	0.9	0.99	0.9	0.8	0.99	0.5		0.3	0.5
Develop cost models		2						0.3	0.9								
Case studies		7						0.7	0.9	0.3		0.3	0.3			0.2	0.1
Baselining & benchmarking		7						0.7	0.9	0.3		0.3	0.3			0.2	0.1
(Funded) partnerships with projects		15	0.8	0.7	0.9	0.9	0.8	0.6	0.9	0.99	0.9	0.9	0.99	0.7	0.6	0.4	0.9
Search heuristics		3	0.4	0.7						0.8							
custom model checkers for programming Ls		12	0.1	0.7	0.99		0.2	0.2	0.5	0.7	0.99	0.99			0.6	0.9	0.7
compositional m/c		6	0.5	0.1					0.1	0.4						0.2	0.1
marketing		3									0.2			0.2	0.7		
design for verification		4	0.7	0.8	0.99	0.9											
tools for visualizing results		1								0.9							
include mc into existing toolset		0															
pick customers		0															

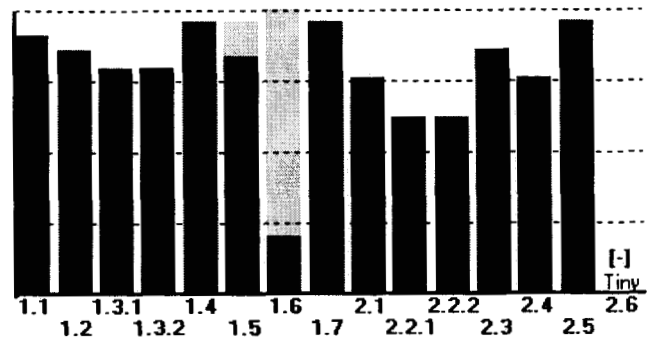
Example: model-checking experts report that teaching people how to write properties using LTL (Linear Temporal Logic) can be *very* effective.

Selection of mitigation measures leads to decreased risk

Initial magnitudes calculated from how much each risk impacts the selected objectives

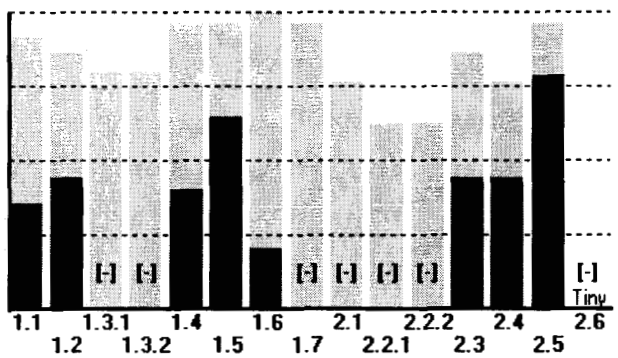


Unmitigated Risk magnitudes

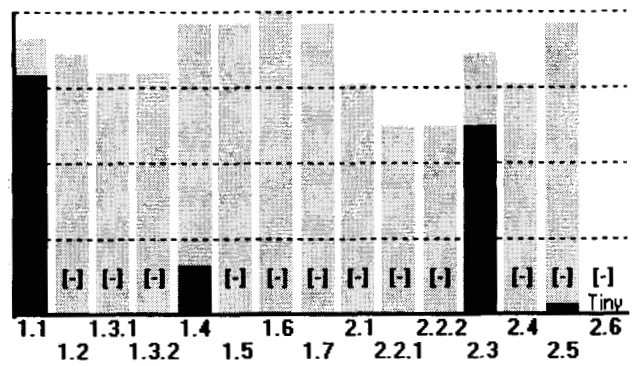


Risk magnitudes, when a lesser Mitigation *(11:Develop cost models)* **adopted**

Green indicates risk that has been reduced because of the selected mitigation(s); different selections achieve different risk reductions



Risk magnitudes, when a key Mitigation *(14:(Funded) partnerships with projects)* **adopted**



Risk magnitudes, when three Mitigations adopted: *"11:Develop cost models" "4:training application engineers"* **and** *"2:tools for translation into mc Ls"*

Judicious selection of mitigations is the goal – reduce risk, but in a cost effective manner (e.g., maybe you can't afford a funded project partnership…)

End of PART II
(Example of application to technology infusion)

For more details, please see:

"Infusing and Selecting V&V Activities" Martin S. Feather
V&V State of the Art: Proceedings of Foundations '02, a Workshop on Model and Simulation Verification and Validation for the 21st Century, Dale K. Pace (ed), held October 22-24, 2002, Johns Hopkins University Applied Physics Laboratory, Laurel, MD.119-129. Published by The Society for Modeling and Simulation International, www.scs.org, ISBN 1-56555-256-3 2002.