Software Risk Estimation and Risk Reduction using the "Defect Detection and Prevention" Risk Approach & Tool(*)

Dr. 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).

* Steve Cornford (JPL) leads the DDP effort

http://ddptool.jpl.nasa.gov
http://eis.jpl.nasa.gov/~mfeather

SMC-IT July 13th 2003 Feather
Roadmap

Part I – Motivation, approach and experience

Part II - Software specific focus

Part III – Live demo!

(separate slides)

Guide to the DDP tool
Technology Infusion Impediments

New mission concepts enabled by new technologies (autonomy, agents, ...), but...

**Infusion of advanced technologies problematic:**

- **Requirements-related:** Miscommunicated, misunderstood or under-defined customer (mission) requirements
- **Readiness-related:** Technology deemed non-flightworthy (unforeseen unresolved engineering issues)
- **Competitiveness-related:** Near-equivalent technologies are or will become available

**Needs:**

- Clearer definition of mission requirements
- Early identification of technology-specific engineering difficulties
- Ability to decide among architectures, technologies
- Projected status of competing technologies

**Challenges:**

- **Groundbreaking nature of new mission concepts and autonomies:**
  Past experience provides only a partial guide
- **Multi-disciplinary nature:**
  No individual is an expert in all areas; No individual can juggle all the details at once
- **Resource constrained**
  Many risks that, if untamed, lead to abandonment or wasted resources
- **Need good decisions early**
  Early on, lack information (e.g., detailed design) on which to base decisions
People + Process + Program

Human Expertise

Information gathering and decision making done by relevant subject area experts

Risk-Based Process

Stakeholders → objectives → risks → mitigations → costing → selection → documentation

DDP

Software Support

Information repository, calculations, visualizations, decision support
Risk-based Roadmapping

What do you want?
“Objectives”
“Requirements”
“Goals”

What can get in the way?
“Risks”
“Failure Modes”
“Defects”

What can you do about it?
“Mitigations”
“Solution Options”
“Preventions, Analyses, Controls, Tests – PACTs”

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, ...)

Dr. Michael Greenfield (NASA HQ):
“Risk as a resource”
Trade risk for other resources.
Use risk as an intermediary between other resources.
Issues outside of technologist expertise.
Issues unique to flight development.

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.

SMC-IT July 13th 2003 Software Risk Estimation... Feather
<table>
<thead>
<tr>
<th>Objectives</th>
<th>Risks</th>
<th>Mitigations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return data</td>
<td>Software bug ridden</td>
<td>Become CMM level 3</td>
</tr>
<tr>
<td>Insufficient detail for decision making. Elaborate!</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- In flight s/w upgrades
- Code/Data separable
- Real-time control loops
- Sync to external clock
- Tolerate memory errors
- Run time memory = ...  
- Storage = ...
- CPU utilization = ...

<table>
<thead>
<tr>
<th>Requirements risks</th>
<th>Requirements practices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstable</td>
<td>Documented</td>
</tr>
<tr>
<td>Incomplete</td>
<td>Formal CM</td>
</tr>
<tr>
<td>Unclear</td>
<td>Peer review</td>
</tr>
<tr>
<td>Invalid</td>
<td>Formal inspections</td>
</tr>
<tr>
<td>Infeasible</td>
<td>Formal reviews</td>
</tr>
<tr>
<td>Unprecedented</td>
<td>Criticality analyses</td>
</tr>
<tr>
<td>Large size/complex</td>
<td>Verifiability check</td>
</tr>
</tbody>
</table>

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

Day 3 – day of the realists!

Select – Mitigations to perform

Objectives to discard

Resources to ask for

Getting the right people is key!!!

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

Decision-making guided by accumulated information

SMC-IT July 13th 2003

Software Risk Estimation...

Feather
Day 1 – day of the pessimists!

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: \( I_1 + I_2 = I_1 + I_2 \)
(therefore objectives can be more than
100% killed!)

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

SMC-IT July 13th 2003
Software Risk Estimation...
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, alleviation)
  - 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”:
- \( E_1 \& E_2 = (1 - (1 - E1) \times (1 - E2)) \)
  - 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
- \( (1 - (1 - 0.8) \times (1 - 0.3)) = (1 - 0.2 \times 0.7) = (1 - 0.14) = 0.86 \)

Note: a law of diminishing returns as apply additional Mitigations

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)!
Motivation
Steve Cornford’s flow-down image: assurance activities “filter out” risk

overfiltered risk

unfiltered risk

MISSION FAILURE MODES

DESIGN RULES
MATERIALS SELECTION
ROBUST DESIGN

QML VENDORS
PROCESS CONTROLS

INSPECTIONS
VERIFICATIONS

RELIABILITY ANALYSES

ANALYSES

TECHNOLOGY
QUALIFICATION

LIFE TESTING
MISSION SIMULATION

ASSEMBLY TESTING
PERFORMANCE TESTING

SYSTEM TESTING
PERFORMANCE TESTING

MISSION SUCCESS?

singly filtered risk

Software Risk Estimation...

SMC-IT July 13th 2003

Feather
Day 3 – day of the realists!

Impacts | Risks | Effects | Risks
---|---|---|---
| | | |

Decision step: select mitigations so as to cost-effectively reduce risk

Outcome – Final Report:

- Background: Capsule description of technology, relevance to mission, placement viz-à-viz. Competition.

- Risk profile: Detailed list of risks, magnitudes derive from trace to objectives, reductions traced to (time ordered) mitigations.

- Risk mitigations: Detailed list of mitigations, how, when and where they reduce risk, purposes of individual mitigations (track & control), cost estimates.

- Recommendations: Defensible selections of mitigations, costed plans, justifiable expectations.

SMC-IT July 13th 2003

Software Risk Estimation... Feather
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

$-o-meter
boundary = iso-risk line!

Big-picture view and ability to drill down to detailed level

SMC-IT July 13th 2003
Software Feather
Example benefits:

- **Cost & Time Saved** (per study cost: $10K - $30K)
  - At least two instances of savings > $1M
    - E.g., Storage technology study revealed *problematic (at risk) overly-stringent requirement*, whose removal permitted dramatic cost & time savings. Technology near cancellation became proposal-winning concept. Requirements honed to requisite level of mission specificity.

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

- **Thorough and Early Risk Identification and Mitigation**
  - Technology-to-flight entire range of risks identified, and mitigations planned
    - E.g., testing commensurate with anticipated mission radiation dosages; pinpointed use of antiquated design tools as a contributing risk factor; ...

- **Technology Adoption**
  - Achieved sufficient understanding of benefits/risks to make “go” decision
    - E.g., GUI-driven autocoding adapted to run as flight instrument controller: benefits understood, risks unknown; *identified risks* (e.g., unrelocatable code) & *mitigations*. Agency usage, industry business case expansion.
Flight Project risk insights from DDP

#1 risk: Lack of L2 cache proves to be EDL bottleneck

MSL: 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 insights

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

Significant improvement possible; excellent case for more funding!

High Cost, High Benefit

Low Cost, High Benefit

Sub-optimal interior

Each point represents a selection of mitigations, located by its cost (horizontal position) and benefit (vertical position).

Thousands of such points!

Risk basis for calculations: risk detracts from attainment of benefits; mitigation of risk costs resources.
Affordable & high value – “shadow” = loss of benefit if omitted

1.4: ($160,000) Develop documented flight manufacturing and test/qualification practices

3.9: ($75,000) Use automation for bump bonding (requires pattern recognition capability)

Affordable & modest value

1.7: ($60,000) Develop a rework plan

High value but increasingly expensive – use only when can afford to do so

4.5: Conduct qualification/reliability tests
4.5.2: Thermal
4.5.2.1: ($200,000) 3000 temp. cycles of mechanical thermal cycling

1.3: ($500,000) Develop die packaging selection criteria (flip-chip, micro BGA, wire bond)

4.4: ($700,000) Develop/procure/fabricate test equipment/structures to qualify COB manufacturing approaches

Low value & inexpensive – may need to scrutinize more close up

3.3: inspection
3.3.5: ($5,000) Ultrasonic (CSAM)

Very dubious value!

3.3: inspection
3.3.4: ($5,000) Visual inspection

1.1: ($20,000) Use embedded passives

SMC-IT July 13th 2003

Software Risk Estimation... Feather
Matching Needs To Research(!)

Needs = Objectives
Areas = Risks
Research = Mitigations


The 198 leaf nodes, placed side-by-side, of the “software” portion of the ACM taxonomy of computer science

9 practitioners

Uh-oh, unmet needs!

E.g., overlap!
Researcher addressing needs shared by several practitioners

19 researchers

SMC-IT July 13th 2003
Software Risk Estimation...
Feather
DDP Timeline

Code Q “Failure Detection and Prevention Program”
PEMs: Tim Larson, Kelly Moran
PI: Steve Cornford
1998-?
Code Q / IV&V “Advanced Risk Reduction Tool”
PEMs: John Kelly, Burton Sigal, Allen Nikora
PIs: John Kelly, Martin Feather
2002-?

Code R “Engineering of Complex Systems”
Level 2 Manager: Stephen Prusha
Pls: Ken Hicks (Risk-Based Design),
Steve Cornford (Risk Workstation),
Martin Feather (S/W Risk Characterization & Mitigation)

Cornford & Barela Blackwood
conceive of DDP

DDP software started
DDP v1.0
ARRT DDP for s/w
Optimization
DDP available via website
Cost/benefit tradespace capability

Technology Infusion Maturity Assessments (partial list)
Hybrid Imaging Technology
Compact Holographic Data Storage
Micro Gyro
LTMPF LabView
Active Pixel Sensor
Chip On Board
Micro Sun Sensor
MSL – technology recipient

PARCS
Thermal Cycle Resistant Electronics
Software Risk Estimation...

SMC-IT July 13th 2003
Reflections: Major Factors

- Information: make most 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
- Utilizable Product:
  - Specific collection of tasks, purposeful, costed, scheduled
  - A traceable rationale, quantitative scoring, reviewable
  - Risk-based understanding, thorough, calibrated
Part II - Software specific focus

Up to this point, approach general enough to apply to software, hardware, research portfolios, ...

Focus shifts now to software, in particular planning its assurance / V&V
Software Engineering Community
Starting Points

Risks: Software Risk Taxonomy (SEI)

Mitigations: several datasets:
1. CMM Key Practices (Infrastructure and Activities)
2. Software Quality Assurance activities from Ask Pete (NASA Glenn tool)
3. JPL QA & V&V activities

Effects: cross-linkings of the above
1. Expert’s best estimates of which help
2. Experts’ 1000+ best estimates of how much (quantified effectiveness) they help

Note: Objectives are PROJECT SPECIFIC

Seeking experience-based data
DDP Trees

1: Product Engineering
2: Requirements Risks
10: Design Risks
11: Functionality: Potential problems in meeting function
12: Difficulty: Difficult design to achieve
13: Interfaces: ill-defined or uncontrolled internal interface
14: Performance: Stringent response time or throughput
15: Testability: Product difficult to test
16: Hardware Constraints: Tight constraints because of
17: Non-Developmental Software: Problems with software
18: Code and Unit Test Risks
19: Feasibility: Implementation of design difficult
20: Unit Test: Level and time for unit test inadequate

Taxonomies are good for reminders, navigation & abstraction (DDP computes aggregate values)

DDP software provides trees, matrices, stem-and-leaf views, and bar charts to enter, organize & view information

SMC-IT July 13th 2003
Software Risk Estimation... Feather
Software Risk Mitigations (partial list)

1: Concept Phase Activities
   2: Concept of operations and operational scenarios
   3: Attend concept reviews.
   4: Identification of risks and mitigations
   5: Review program/project plans
   6: Perform Preliminary Hazard Analysis (PHA)
   7: Determine software safety level of effort/scoped.
   8: Develop list of safety efforts/activities.
   9: Start Software Hazard Tracking Log

10: Requirement Phase Activities
   11: Provide Generic Safety Requirements.
   12: Assist in the generation of overall requirements.
   13: Perform Software Safety Requirements Analysis.
   14: Perform Requirements Criticality Analysis.
   15: Verify inclusion of any safety critical end of life-cycle requirements.
   16: Produce CIL.
   17: Participate in requirement working group or JAD meetings
   18: Attend requirement reviews.
   19: Review and analyze requirements for industry acceptable and required charac
   20: Track and resolve any software safety requirements related issues.
   21: Generation and/or review requirement traceability/verification matrices
   22: Review software development, CM, and test plans
   23: Requirements Flow-down Analysis.
   24: Review and provide guidance on program/project metrics.
   25: Verify software safety requirements.
   26: Determine impact of changes and defects on software safety.
   27: Perform Phase 1 Safety Review.

20: Design Phase Activities
   29: Assess safety related issues during design meetings.
   30: Design reviews and track any issues related to safety.
   31: Analyze software using tools
Quantitatively Relating Software Risks to Mitigations (fragment of 64 x 107 matrix)

<table>
<thead>
<tr>
<th>Mitgn</th>
<th>Risk</th>
<th>Col = Completeness: Incomplete requirements</th>
<th>Row = Conduct Formal Reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Stab Unstab</td>
<td>Clarity Unclear</td>
</tr>
<tr>
<td>Mitgn</td>
<td>Risk</td>
<td>Stab Unstab</td>
<td>Clarity Unclear</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27 26 27 28 19 25 43</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>Implement</td>
<td>11 7 0.9 0.3 0.9 0.9 0.3 0.3 0.1</td>
<td></td>
</tr>
<tr>
<td>Implement</td>
<td>Management</td>
<td>12 9 0.3 0.9 0.9 0.1 0.3 0.3 0.1</td>
<td></td>
</tr>
<tr>
<td>Documenter</td>
<td>Peer review</td>
<td>9 3 0.9 0.9 0.9 0.9 0.9 0.9 0.1</td>
<td></td>
</tr>
<tr>
<td>Conduct</td>
<td>Software</td>
<td>9 9 0.9 0.9 0.9 0.9 0.9 0.9 0.1</td>
<td></td>
</tr>
<tr>
<td>Requirement</td>
<td>Peer review</td>
<td>10 25 0.9 0.9 0.9 0.9 0.9 0.9 0.1</td>
<td></td>
</tr>
<tr>
<td>Implement</td>
<td>Conduct</td>
<td>8 14 0.9 0.9 0.9 0.9 0.9 0.9 0.1</td>
<td></td>
</tr>
<tr>
<td>Conduct</td>
<td>Document</td>
<td>6 8 0.9 0.9 0.9 0.9 0.9 0.9 0.1</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Coarse distinctions (0, 0.1, 0.3, 0.9) do suffice for insightful decision making

n.b. Blank = 0, no benefit
“Stem-and-leaf”(*) visualization of DDP sparse matrices:

E.g., Risks & their Mitigations

Risks – red width ≡ log outstanding Σ impact

Mitigations – turquoise width ≡ effect

(*) 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.
### Quantitatively Relating Software Risks to Mitigations

(fragment of 64 x 107 matrix)

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
Typical DDP screen

Lists of mitigations applicable to each risk

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

SMC-IT July 13th 2003
Software Risk Estimation...
Feather
# 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)
- 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)

SFTA / FMECA's appropriate for software/system boundary (e.g., effect of erroneous inputs to / outputs from software)

## 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)
- Calculate system risk / reliability when development process key to 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)

Appropriate for software itself? Encompass both software structure and unreliability of development
Message to Remember –
Risk-informed Decision Making!

What do you want? What can get in the way? What can you do about it?

- Information: make most 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
- Utilizable Product:
  - Specific collection of tasks, purposeful, costed, scheduled
  - A traceable rationale, quantitative scoring, reviewable
  - Risk-based understanding, thorough, calibrated
Backup slides
DDP Sensitivity Analysis

1) Menzies’ technique showed optimal solution robust

2) Vary effect values one by one, recompute requirements attainment, tabulate results:

<table>
<thead>
<tr>
<th>Change</th>
<th>% Change</th>
<th>PACT</th>
<th>Failure Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.02</td>
<td>0.76</td>
<td>Select/make laser</td>
<td>Insufficient power</td>
</tr>
<tr>
<td>-0.834</td>
<td>0.314</td>
<td>CCD Qualification</td>
<td>CCD degredation</td>
</tr>
<tr>
<td>-0.6</td>
<td>0.226</td>
<td>System Study</td>
<td>Other technologies are better</td>
</tr>
<tr>
<td>-0.329</td>
<td>0.124</td>
<td>Hermetic packaging</td>
<td>non-Hermetic</td>
</tr>
<tr>
<td>-0.246</td>
<td>0.0926</td>
<td>Fibre qualification</td>
<td>Fibre degredation</td>
</tr>
</tbody>
</table>

3) Use results for relative decision making, not as absolute measures of reliability. Having identified areas of critical concern, apply other techniques (e.g., probabilistic risk assessment).
DDP Risk Model – Refinements (1)
Mitigation Subtypes

Preventions: decrease likelihood of risk arising (e.g., training; coding conventions)

Alleviations: decrease impact (severity) of risk if it occurs (e.g., defensive programming)

Detections: identify risks prior to use (e.g., testing, analysis), leading to repair of risks so detected; net result: decreased likelihood of risk remaining.

Repair cost = risk’s unit repair cost x decrease in risk’s likelihood

DDP - Defect Detection and Prevention
DDP Risk Model – Refinements (2)

Time phases

**Mitigations** grouped into user-defined **phases**

E.g., requirements, designing, coding, unit testing, integration testing ...

or

“DBAT” (hardware) Design, Build, Assemble, Test

or

Pre-phase A, Phase A, Phase B, ... (JPL s/c development terms)

**Uses:**

- Incorporate budget constraints on per-phase expenditure
- Capture cost-of-repair escalation in later phases (1:10:100...)
- Insight into risk profile over time
Risk Reduction Profile Over Time

- Plan A, slipped
- Plan B, slipped

Risk at launch: high vs. low

For terrestrial software, think “release date”!

Launch date

Development time

Smc-IT July 13th 2003
Software Risk Estimation... Feather
DDP Risk Model – Refinements (3)
Induced/Aggravated Risks

Mitigation that make risks worse
- hardware shake test breaks something else
- software bugfix introduces new bugs
- turning on/off array bound checking changes timing
- adoption of design element introduces design-specific risks

Mitigation induces risk by extent \( e \) \((-1 \leq e < 0)\)
Risk likelihood before = \( L \)
Risk likelihood after = \((1 - (1 - L)(1 + e))\)
  E.g., \( e = -0.9 \), Likelihood after = \((1 - (1 - L) \times 0.1)) = 1 - 0.1 + 0.1L = 0.9 + 0.1L\)

Mitigation aggravates risk by extent \( e \) \((e < -1)\)
Risk impact before = \( I \)
Risk impact after = \( I \times (-e)\)
  E.g., \( e = -3 \), Risk impact after = \( I \times 3\)