ARRT
Advanced Risk Reduction Tool

Presentation to the 1st Annual NASA
Office of Safety and Mission Assurance (OSMA)
Software Assurance Symposium (SAS)

Dr. Martin S. Feather
ARRT Center Initiative Lead*
Jet Propulsion Laboratory
California Institute of Technology

Martin.S.Feather@Jpl.Nasa.Gov
http://eis.jpl.nasa.gov/~mfeather

*Initiative began in 1999 with Dr. John Kelly as Lead
ARRT Heritage & Contributors

ARRT is inspired by, and based on JPLer Steve Cornford’s Defect Detection and Prevention (DDP) and JPLer Tim Larson’s Risk Balancing Profiles (RBP).

contributors (JPL)

John Kelly
Burt Sigal
James Eddingfield
Steve Cornford
Phil Daggett
Julia Dunphy
Roger Klem

contributors

Jim Kiper (U. Miami, Ohio)
William Evanco (Drexel)
Steve Fickas (U. Oregon)
Martha Wetherhold (NASA Glenn)
Richard Hutchinson (Wofford, SC)

primary collaborators

Tim Menzies (U. British Columbia)
Tim Kurtz (NASA Glenn)
Hoh In (Texas A&M)

funding, management & guidance

NASA Code Q, NASA Goddard IV&V Facility
Siamak Yassini, Ken McGill, Marcus Fisher
The Universe of ARRT Customers

optimists
“Hello, I’m from Software Quality Assurance / IV&V and I’m here to help you”

pessimists
“Got Risk?”
“Too much…”
“Too little…”
“Don’t know…”

pragmatists
“Let’s do it!”
“How?”
"Hello, I’m from Software Quality Assurance / IV&V and I’m here to help you"

Many attendees of this symposium are likely to already believe in the net value of assurance activities, but optimism alone is not sufficiently contagious!

What is needed is the means to quantitatively assess the cost/benefit of assurance activities applied to specific projects. This will:

• be more convincing
• determine best use of limited resources
• identify alternatives (e.g., requirements to discard)
The Optimists

Cost/benefit data & reasoning has been applied to:

**Individual activities**, e.g., Regression testing [Graves et al, 1998].

**Pairwise comparisons**, e.g., “Peer reviews are more effective than function testing for faults of omission and incorrect specification” [Basili & Boehm, 2000].

\[
\begin{align*}
\text{Gap!} & \quad \text{ARRT performs quantitative cost/benefit calculation for} \\
& \quad \text{suite of assurance activities applied to a specific project}
\end{align*}
\]

**Lifecycle process improvement**, e.g., Quality, productivity and estimation gains from CMM-like process improvement [McGarry et al, 1998].
ARRT's Quantitative Cost/Benefit Model

Risk mitigations subdivided into

Preventions – prevent problems from appearing in the first place
  e.g., training programmers → fewer coding errors
  cost = performing prevention
  benefit = reduction of risk likelihood

Detections – detect problems so that they can be corrected
  e.g., unit testing → detects internal coding errors
  cost = performing detection +
    performing the repair (cost depends on when!)
  benefit = reduction of risk likelihood

Alleviations – applied to decrease the severity of problems
  e.g., robust coding → tolerant of out-of-bound input values
  cost = performing alleviation
  benefit = reduction of risk severity
Cost/Benefit – Simple Scenario

Requirements phase

Poorly written requirements
Use ARM to do Requirements Analysis ($)
Correct ambiguous requirements ($)

Programming errors
Misinterpret ambiguous requirements

Test phase

System tests, observed by spacecraft engineers($$)
Reimplement misinterpreted requirements ($$$)
Correct programming errors ($$)

Operations phase

RISKS
Mission loss due to misinterpretation of requirements
Mission loss due to programming errors

Low costs to analyze with ARM & correct flaws now

assurance choices

High cost to reimplement requirements this late in development

mistakes happen

NASA OSMA SAS 2001

Advanced Risk Reduction Tool - M.S. Feather
Cost/Benefit – Simple Scenario (cont.)

- Use ARM to do Requirements Inspection ($).
- Correct ambiguous requirements ($).
- System tests, observed by spacecraft engineers ($$$).
- Reimplement misinterpreted requirements (-/$$$).
- Correct programming errors ($$).

\[
\begin{align*}
0 + 0 + 0 + 0 + 0 &= 0 \\
0 + 0 + $$ + $$$ + $$ &= $$$$$$$ \\
$$ + $$ + 0 + 0 + 0 &= $$ \\
$$ + $$ + $$ + 0 + $$ &= $$$$$$ \\
\end{align*}
\]

Lowest risk, but NOT highest cost - savings from correcting problems early.

Risk decreases as the checks are completed.

Risk of mission loss decreases with each step completed.
Return On Investment of Assurance/IV&V

\[ \square \quad \square \quad \checkmark \quad \checkmark \quad S + S + 0 + 0 + 0 = $$ \]

Is it worth paying $$$$ to save this much risk?

Risk = loss of Requirements

If you are bold enough to value Requirements in the same unit of currency as costs of Mitigations, then you can calculate Return On Investment (ROI)

Valuation of Requirements can be difficult, e.g.,
- What is the value of discovering water on Mars?
- What is the value of returning a Mars sample to Earth?
- What is the value of an astronaut's life?
ARRT's Quantitative Cost/Benefit Model

Cost/benefit computations in ARRT

- Automatic
- Handle *suite* of assurance activities
- Permit data to be changed if we know better than standard estimates
- Distinguish development phases (requirements, design, ...)
- Distinguish preventions, detections and alleviations
- Combine with underlying risk computation model (see next section)
The Pessimists
GOT RISK?

TOO MUCH – use ARRT to plan how to reduce risk in a cost-effective manner.

TOO LITTLE – use ARRT to plan how to accept more risk in exchange for reduced cost and schedule, more functionality, etc.

JUST RIGHT – use ARRT to maintain a desired risk profile through the lifetime of the project.

DON’T KNOW – use ARRT to assess risk status.

"Risk as a Resource" - Dr. Michael Greenfield
[Greenfield, 1998]
ARRT's treatment of Risk--DDP & RBP concepts, specifically populated with software data

ARRT is inspired by, and based on JPLer Steve Cornford's Defect Detection and Prevention (DDP) and JPLer Tim Larson's Risk Balancing Profiles (RBP).

In particular, ARRT inherits DDP's Risk Model.

DDP is a process [Cornford et al, 2001] supported by a custom tool [Feather et al, 2000a] for quantitative risk management.

RBP is a qualitative risk management tool populated with risk and risk mitigation data.

DDP & RBP merged [Feather et al, 2000b] into DDP

ARRT uses this merged combination of DDP & RBP
ARRT inherits DDP's Risk Model

DDP utilizes three trees of key concepts:

Requirements (what you want)
Failure Modes / Risk Elements (what can get in the way of requirements)
PACTs (what can mitigate risk)

and two matrices that connect those concepts:

Impacts (how much Requirement loss is caused by a FM)
Effectivenesses (how much a PACT mitigates a FM)
Information is derived from user-provided data via built-in computations, e.g.,

- FM's cumulative impact = FM.Likelihood * (Σ (R ∈ Requirements) R.Weight * Impact(R, FM))

Information presented via cogent visualizations

- Bar charts
- Risk Region chart
- Stem-and-leaf plots
- Detailed view of properties of individual element
ARRT/DDP Trees

Taxonomies of Software Requirements / Risks / Risk Mitigations

Contracted

Expanded

Selected

Deselected

Number: Title

Autonumbering: linear 1,2,... or tree 1, 1.1, 1.2, 1.2.1, ...

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
**ARRT/DDP Matrices**

**Effects (Mitigation x Risk):**

Proportion of Risk reduced by Mitigation

**Impacts (Requirement x Risk):**

Proportion of Requirement loss if Risk occurs

<table>
<thead>
<tr>
<th>PACTs</th>
<th>FM</th>
<th>FoMR</th>
<th>Stability</th>
<th>Clarity</th>
<th>Validity</th>
<th>Feasib</th>
<th>Pre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>2.95</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
</tr>
<tr>
<td>Identify</td>
<td>2.3</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td>Mainte</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td>Software</td>
<td>2.65</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Implement</td>
<td>1.85</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.3</td>
</tr>
<tr>
<td>Manag</td>
<td>0.15</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Docum</td>
<td>1.65</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Prep</td>
<td>2.8</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Numbers supplied by experts and/or based on accumulated metrics
ARRT/DDP Visualizations - Bar Charts

Risks bar chart

- Unsorted – order matches leaf elements in Risk tree
- Green: of this Risk’s total Impact on Requirements, that *saved* by Mitigations
- Red: of this Risk’s total Impact on Requirements, that *remaining* despite Mitigations

Sorted – in decreasing order of remaining risk

Requirements bar chart – how much each is impacted

Mitigations bar chart – how much impact each is saving
User defines risk levels demarcating red/yellow/green/(tiny) risk regions.

Log/Log scale: diagonal boundaries = risk contour lines.

Conventional measure of risk as impact (severity) x likelihood.
**ARRT/DDP Visualizations - stem-and-leaf(*) charts**

Compact visualization of DDP’s 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 by D. Howard, extended further by us in DDP.
The Pragmatists

"Has it been used?"

"Where does the data come from?"

"How does it combine with software estimation & planning?"

"What about...?"
Focused study data: Software Assessment Exercise

Steve Cornford, JPL + others

- Focus: code generation by [product name deliberately hidden]
  - Flight code of modest experiment
  - Flight code for future missions
- 15+ experts in 4 x 4-hour sessions, Sept 2000
  - [product] experts
  - Mission experts
  - Software experts (SQA, coders, …)
- Large information set
  - 47 Requirements (unprioritized)
  - 76 Risks (near-term mission-specific & futuristic)
  - 303 Mitigations (pre-populated with large set)
  - 107 Impacts
  - 223 Effects
Portions of the Requirements tree and bar chart

- 2.2.3: Comm IF's
  - 2.2.3.1: IEEE 1394
  - 2.2.3.2: RS-422 (etc)
  - 2.2.3.3: MIL STD 1553/1773

- 2.3: Data Handling
  - 2.3.1: Accommodate SW upgrades in flig
  - 2.3.2: Telemetry
    - 2.3.2.1: Uplink and downlink accour
      - 2.3.2.1.1: Uplink
      - 2.3.2.1.2: Downlink
  - 2.3.3: Storage

- 2.4: Fault Protection
  - 2.4.1: System level fault protection

- 2.4.2: Code and Data separable
- 2.4.3: Work around memory errors
- 2.4.4: Autonomous failure recovery
  - 2.4.4.1: Due to external failures
  - 2.4.4.2: Due to internal failures

- 3: Programmatic Requirements
  - 3.1: Budget = TBD$
  - 3.2: Deliver in 2003
  - 3.3: Use TBD developmental tools
  - 3.4: Use TBD lifecycle tools
  - 3.5: Use TBD test tools
  - 3.6: Utilize legacy code
  - 3.7: Link w/ other languages

Generic: Synchronize to external clock

<table>
<thead>
<tr>
<th></th>
<th>1.2</th>
<th>2.3.1</th>
<th>2.4.4.2</th>
<th>2.4.4.1</th>
<th>3.7</th>
<th>2.2.2</th>
<th>2.1.1</th>
<th>3.3</th>
<th>3.4</th>
<th>3.5</th>
<th>1.3</th>
<th>2.1.2</th>
<th>1.9</th>
<th>2.1.3</th>
<th>1.6.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Software Engineering Community Data

- **Risks:** Software Risk Taxonomy (SEI)
- **Mitigations:** two datasets:
  1. JPL’s Risk Balance Profile of SQA actions
  2. Assurance activities from Ask Pete (NASA Glenn tool)
- **Effects:** cross-linkings of the above (Jim Kiper)
  1. Expert’s best estimates of yes/no (Prof. J. Kiper)
  2. Experts’ 1000+ best estimates of quantified effectiveness (Prof. J. Kiper & J. Eddingfield)

Note: **Requirements** are PROJECT SPECIFIC
Software Estimation & Planning data: ARRT – Ask Pete collaboration

Ask Pete runs to gather project characteristics, make first cut at suggested selection of risk mitigations.

Mitigation selection passed to ARRT

ARRT runs to allow user to assess risk, provide costs, customize to project (add/remove risks, refine effect values, etc.), tune selection accordingly.

Revised mitigation selection returned to Ask Pete

Ask Pete runs to generate final reports

Tim Kurtz, Tim.Kurtz@grc.nasa.gov
SAIC/NASA Glenn Research Center
http://tkurtz.grc.nasa.gov/pete
Principal Investigator ⇒ Martha Wetherholt

see companion presentation in this symposium
ARRT - Tim Menzies collaboration

Prof. Tim Menzies, U. British Columbia

- Optimization - automated search for (near) optimal mitigations suites
  - Least risk for given cost
  - Least cost for given risk
- Sensitivity analysis
  - On which data values do the results hinge?
    - Scrutinize these values further
    - Identify points of leverage (e.g., problematic requirements; make-or-break decisions)
- Retain human involvement
- Extend reasoning to more complex data
  - Interactions: mitigations that induce risk (e.g., code changes to correct one bug may introduce other bugs)
  - Ranges / distributions of values (e.g., [0.1 – 0.3])

see companion presentation in this symposium
ARRT – Hoh In et al collaboration: IEESIM

Other Tools (e.g., VCR) → ASK PETE

Repository of project data
- Insert & classify,
- Search,
- Retrieve,
- Delete

Accessibility via the web

http://www.cs.tamu.edu/faculty/hohin/

IEESIM

Integrated views (data schema) from local tool views
Exchangeable format based on XML
Extendable interfaces for additional tools
Shared Information Mediator

Prof. Hoh In, Texas A&M University
Hoh In et al – Visualized Conflict Resolution (VCR)

ARRT data passed to VCR. Purposes:

- **Sophisticated Visualization**
  - Intuitive graphical presentations of consensus, conflict trends.
  - Scalable and multi-dimension visualization.

- **Powerful Analysis Support**
  - Identify non-trivial interrelationships (Clustering).
  - Discover stakeholder decision rationales (Profiles).
  - Benefit-cost tradeoff analysis

XML adopted as standard medium of data exchange

**Status:** examples of both kinds of data transferred & visualized

see Friday's demo at this symposium

Hoh's visualization work motivated inclusion of the green/yellow/red Risk chart capability into ARRT - slide 18
Hoh In et al – Visualized Conflict Resolution (VCR)

- Shows the degree of consensus in form of ellipse
- Shows issues, criteria of evaluation
- Shows individual stakeholder perceptions/votes, group perceptions
- Shows clusters spanning all criteria of an issue
- Shows clusters per criterion, mean, max, min values
Concluding Remarks

even this talk maps to ARRT/DDP's concepts!

Requirements:
what ARRT will help you achieve

Risks:
what ARRT will help you avoid

Mitigations:
what it takes to apply ARRT

optimists

pessimists

pragmatists

see Friday's demo at this symposium
References


