Fundamentals of RE
Chapter 3
Requirements Evaluation
Negotiation-based decision making:
as introduced in Chapter 1 ...

- Identification & resolution of inconsistencies
  - conflicting stakeholder viewpoints, non-functional reqs, ...
  - to reach agreement

- Identification, assessment & resolution of system risks
  - critical objectives not met, e.g. safety hazards, security threats, development risks, ...
  - to get new reqs for more robust system-to-be

- Comparison of alternative options, selection of preferred ones
  - different ways of: meeting same objective, assigning responsibilities, resolving conflicts & risks

- Requirements prioritization
  - to resolve conflicts, address cost/schedule constraints, support incremental development
Requirements evaluation: outline

- Inconsistency management
  - Types of inconsistency
  - Handling inconsistencies
  - Managing conflicts: a systematic process
- Risk analysis
  - Types of risk
  - Risk management
  - Risk documentation
  - DDP: quantitative risk management for RE
- Evaluating alternative options for decision making
- Requirements prioritization

Inconsistency management

- Inconsistency = violation of consistency rule among items
- Inconsistencies are highly frequent in RE
  - inter-viewpoints: each stakeholder has its own focus & concerns (e.g. domain experts vs. marketing dept)
  - intra-viewpoint: conflicting quality reqs (e.g. security vs. usability)
- Inconsistencies must be detected and resolved ...
  - not too soon: to allow further elicitation within viewpoint
  - not too late: to allow software development
  
  (anything may be developed from inconsistent specs)
Types of inconsistency in RE

- **Terminology clash**: same concept named differently in different statements
  
  e.g. library management: "borrower" vs. "patron"

- **Designation clash**: same name for different concepts in different statements
  
  e.g. "user" for "library user" vs. "library software user"

- **Structure clash**: same concept structured differently in different statements
  
  e.g. "latest return date" as time point (e.g. Fri 5pm) vs. time interval (e.g. Friday)

---

Types of inconsistency in RE (2)

- **Strong conflict**: statements not satisfiable together
  
  - i.e. logically inconsistent: \( S, \text{not} \ S \)
  
  e.g. "participant constraints may not be disclosed to anyone else" vs. "the meeting initiator should know participant constraints"

- **Weak conflict** (divergence): statements not satisfiable together under some boundary condition
  
  - i.e. strongly conflicting if \( B \) holds: potential conflict
  
  - MUCH more frequent in RE

  e.g. (staff’s viewpoint)
  
  "patrons shall return borrowed copies within \( X \) weeks" vs. (patron’s viewpoint)
  
  "patrons shall keep borrowed copies as long as needed"

  \( B: \) "a patron needing a borrowed copy more than \( X \) weeks"
Handling inconsistencies

- Handling clashes in terminology, designation, structure: through agreed glossary of terms to stick to
  - For some terms, if needed: accepted synonym(s)
  - To be built during elicitation phase

- Weak, strong conflicts: more difficult, deeper causes...
  - Often rooted in underlying personal objectives of stakeholders => to be handled at root level and propagated to requirements level
  - Inherent to some non-functional concerns (performance vs. safety, confidentiality vs. awareness, ...) => exploration of preferred tradeoffs
  - Example: spiral, negotiation-based reconciliation of win conditions [Boehm et al, 1995]

Managing conflicts: a systematic process

- Overlap = reference to common terms or phenomena
  - precondition for conflicting statements
  - e.g. gathering meeting constraints, determining schedules

- Conflict detection ... (see Chapters 16, 18)
  - informally
  - using heuristics on conflicting req categories
    - "Check information req & confidentiality req on related objects"
    - "Check reqs on decreasing & increasing related quantities"
  - using conflict patterns
  - formally (theorem proving techniques)
Detected conflicts should be documented

- For later resolution, for impact analysis
  - statement in multiple conflicts, most conflicting statements, ...
- Using documentation tools, query tools along Conflict links recorded in requirements database
- Or in interaction matrix:

<table>
<thead>
<tr>
<th>Statement</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0</td>
<td>1000</td>
<td>1</td>
<td>1</td>
<td>1002</td>
</tr>
<tr>
<td>S2</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>S3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>S4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>1002</td>
<td>1000</td>
<td>2</td>
<td>2</td>
<td>2006</td>
</tr>
</tbody>
</table>

\[ S_{ij} = \begin{cases} 1 & \text{conflict} \vspace{1em} \\ 0 & \text{no overlap} \vspace{1em} \\ 1000 & \text{no conflict} \end{cases} \]

\#Conflicts(S_i) = \text{remainderOf}(1002 \text{ div } 1000)
\#nonConflictingOverlaps(S_i) = \text{quotientOf}(1002 \text{ div } 1000)

Managing conflicts: a systematic process (2)

- For optimal resolution, better to ...
  - explore multiple candidate resolutions first,
  - compare, select/agree on most preferred next
- To generate candidate resolutions, use ...
  - elicitation techniques
    (interviews, group sessions)
  - resolution tactics

© 2009 John Wiley and Sons
www.wiley europe.com/college/vanlamsweerde

© A. van Lamsweerde
**Conflict resolution tactics**

- **Avoid** boundary condition
  - e.g. “Keep copies of highly needed books unborrowable”
- **Restore** conflicting statements
  - e.g. “Copy returned within X weeks and then borrowed again”
- **Weaken** conflicting statements
  - e.g. “Copy returned within X weeks unless explicit permission”
- **Drop** lower-priority statements
- **Specialize** conflict source or target
  - e.g. “Book loan status known by staff users only’

**Transform conflicting statements or involved objects, or introduce new requirements**

**Managing conflicts: a systematic process (3)**

1. Identify overlapping statements
2. Detect conflicts among them, document these
3. Generate conflict resolutions
4. Evaluate resolutions, select preferred

- **Evaluation criteria for preferred resolution:**
  - contribution to critical non-functional requirements
  - contribution to resolution of other conflicts & risks

- **See ...**
  - Sect. 3.3 in this chapter (“Evaluating alternative options”)  
  - Chapters 16, 18
Requirements evaluation: outline

- Inconsistency management
  - Types of inconsistency
  - Handling inconsistencies
  - Managing conflicts: a systematic process

- Risk analysis
  - Types of risk
  - Risk management
  - Risk documentation
  - DDP: quantitative risk management for RE

- Evaluating alternative options for decision making
- Requirements prioritization

What is a risk?

- Uncertain factor whose occurrence may result in loss of satisfaction of a corresponding objective
  - e.g. a passenger forcing doors opening while train moving
  - a meeting participant not checking email regularly

- A risk has...
  - a likelihood of occurrence,
  - one or more undesirable consequences
  - e.g. passengers falling out of train moving with doors open

- Each risk consequence has ...
  - a likelihood of occurrence if the risk occurs
    (not to be confused with risk likelihood)
  - a severity: degree of loss of satisfaction of objective
Types of RE risk

- **Product-related** risks: negative impact on functional or non-functional objectives of the system
  => failure to deliver services or quality of service
  e.g. security threats, safety hazards
- **Process-related** risks: negative impact on development objectives
  => delayed delivery, cost overruns, ...
  e.g. personnel turnover

RE risk management

- Risk management is iterative
  - countermeasures may introduce new risks
- Poor risk management is a major cause of software failure
  - natural inclination to conceive over-ideal systems
    (nothing can go wrong)
  - unrecognized, underestimated risks => incomplete, inadequate reqs
Risk identification: risk checklists

- Instantiation of risk categories to project specifics
  - associated with corresponding req categories (cf. Chap. 1)
- Product-related risks: req unsatisfaction in functional or quality req categories
  - info inaccuracy, unavailability, unusability, poor response time, poor peak throughput, ...
  - e.g. ? inaccurate estimates of train speed, positions ?
- Process-related risks: top 10 risks [Boehm, 1989]
  - req volatility, personnel shortfalls, dependencies on external sources, unrealistic schedules/budgets, ...
  - poor risk management
  - e.g. ? unexperienced developer team for train system ?

Risk identification: component inspection

- For product-related risks
- Review each component of the system-to-be: human, device, software component ...
  - can it fail?
  - how?
  - why?
  - what are possible consequences?
  - e.g. on-board train controller, station computer, tracking system, communication infrastructure, ...
- Finer-grained components => more accurate analysis
  - e.g. acceleration controller, doors controller, track sensors, ...
Risk identification: risk trees

- Tree organization for causal linking of failures, causes, consequences
  - similar to fault trees in safety, threat trees in security
- **Failure node** = independent failure event or condition
  - decomposable into finer-grained nodes
- **AND/OR links**: causal links through logical nodes ...
  - **AND-node**: child nodes must all occur for parent node to occur as consequence
  - **OR-node**: only one child node needs to occur

Risk tree: example

- Door opens while train moving
- Train is moving
- Software controller fails
- Door actuator fails
- Speedometer fails
- Passenger forces doors to open
- Wrong requirement
- Wrong assumption
- Wrong specification
- Wrong implementation

© 2009 John Wiley and Sons
www.wileyeurope.com/college/vanlamsweerde

© A. van Lamsweerde
Building risk trees: heuristic identification of failure nodes

- Checklists, component failure
- **Guidewords** = keyword-based patterns of failure
  - NO: “something is missing”
  - MORE: “there are more things than expected”
  - LESS: “there are fewer things than expected”
  - BEFORE: “something occurs earlier than expected”
  - AFTER: “something occurs later than expected”
- But ... problems frequently due to **combinations** of basic failure events/conditions ...

Analyzing failure combinations: cut set of a risk tree

- **Cut set** of risk tree RT: set of minimal **AND-combinations** of RT's leaf nodes sufficient for causing RT's root node
  - **Cut-set tree** of RT: set of its leaf nodes = RT's cut set
- Derivation of cut-set tree CST of RT:
  - CST's top node := RT's top logical node
  - If current CST node is **OR**-node: expand it with RT's corresponding alternative child nodes
    - If current CST node is **AND**-node: expand it in single aggregation of RT's conjoined child nodes
  - Termination when CST's child nodes are all aggregations of leaf nodes from RT
Cut-set tree derivation: example

Cut set = \{(TM, WR), (TM, WA), (TM, WS), (TM, WI), (TM, DAF), (TM, SF), (TM, PFDO)\}

all combinations of bad circumstances for root risk to occur

Risk identification: using elicitation techniques

- **Scenarios** to point out failures from WHAT IF questions
  - interactions not occurring
  - interactions occurring too late
  - unexpected interactions (e.g. under wrong conditions), ...

- **Knowledge reuse**: typical risks from similar systems

- **Group sessions** focussed on identification of project-specific risks
**Risk assessment**

- **Goal**: assess likelihood of risks + severity, likelihood of consequences, to control high-priority risks
- **Qualitative assessment**: use qualitative estimates (levels)
  - for **likelihood**: \{very likely, likely, possible, unlikely, \...\}
  - for **severity**: \{catastrophic, severe, high, moderate, \...\}

  => risk likelihood-consequence table for each risk
  => risk comparison/prioritization on severity levels

---

**Qualitative risk assessment table: example**

Risk: “Doors open while train moving”

<table>
<thead>
<tr>
<th>Consequences</th>
<th>Likely</th>
<th>Possible</th>
<th>Unlikely</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss of life</td>
<td>Catastrophic</td>
<td>Catastrophic</td>
<td>Severe</td>
</tr>
<tr>
<td>Serious injuries</td>
<td>Catastrophic</td>
<td>Severe</td>
<td>High</td>
</tr>
<tr>
<td>Train car damaged</td>
<td>High</td>
<td>Moderate</td>
<td>Low</td>
</tr>
<tr>
<td>#passengers decreased</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Bad airport reputation</td>
<td>Moderate</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

- **Easy to use**
- **Limited conclusions**: coarse-grained, subjective estimates
  likelihood of consequences not considered
Risk assessment (2)

- **Quantitative** assessment: use numerical estimates
  - for likelihoods: \( \{0, 0.1, 0.2, ..., 0.9, 1.0\} \) probability values
    - or \( \{0-0.3, 0.3-0.5, 0.5-0.7, 0.7-1.0\} \) probability intervals
  - for severity: scale from 1 to 10

⇒ **Risk exposure** for risk \( r \) with independent consequences \( c \):
  \[
  \text{Exposure}(r) = \sum_c \text{Likelihood}(c) \times \text{Severity}(c)
  \]

⇒ Risk comparison/prioritization based on exposures
  (with risks weighted by their likelihood)

😄 Finer-grained than qualitative assessment

😖 Sill subjective estimates: not grounded on system phenomena
  ⇒ to be elicited from domain experts
  or data collection from accumulated experiments

Risk control

- **Goal**: Reduce high-exposure risks through countermeasures
  - yields new or adapted requirements
  - should be cost-effective

- Cf. conflict management:
Exploring countermeasures

- Using elicitation techniques
  - interviews, group sessions
- Reusing known countermeasures
  - e.g. generic countermeasures to top 10 risks [Boehm, 1989]
    - simulation \(\preceq\) poor performance
    - prototyping, task analysis \(\preceq\) poor usability
    - use of cost models \(\preceq\) unrealistic budgets/schedules
- Using risk reduction tactics

Risk reduction tactics

- **Reduce risk likelihood**: new reqs to ensure significant decrease
  e.g. “Prompts for driver reaction regularly generated by software”
- **Avoid risk**: new reqs to ensure risk may never occur
  e.g. “Doors may be opened by software-controlled actuators only”
- **Reduce consequence likelihood**: new reqs to ensure significant decrease of consequence likelihood
  e.g. “Alarm generated in case of door opening while train moving”
- **Avoid risk consequence**: new reqs to ensure consequence may never occur
  e.g. “No collision in case of inaccurate speed/position estimates”
- **Mitigate risk consequence**: new reqs to reduce severity of consequence(s)
  e.g. “Waiting passengers informed of train delays”
Selecting preferred countermeasures

- Evaluation criteria for preferred countermeasure:
  - contribution to critical non-functional requirements
  - contribution to resolution of other risks
  - cost-effectiveness

- Cost-effectiveness is measured by risk-reduction leverage:

\[
RRL(r, \text{cm}) = \frac{(\text{Exp}(r) - \text{Exp}(r|\text{cm}))}{\text{Cost}(\text{cm})}
\]

\text{Exp}(r): \text{exposure of risk } r \\
\text{Exp}(r|\text{cm}): \text{new exposure of } r \text{ if countermeasure cm is selected}

\Rightarrow \text{Select countermeasures with highest RRLs}
- refifiable through cumulative countermeasures & RRLs

Risks should be documented

- To record/explain why these countermeasure reqs, to support system evolution

- For each identified risk:
  - conditions/events for occurrence
  - estimated likelihood
  - possible causes & consequences
  - estimated likelihood & severity of each consequence
  - identified countermeasures + risk-reduction leverages
  - selected countermeasures
  \(\approx\) annotated risk tree

- More on risk management & documentation in Chaps. 9, 16, 18
Requirements evaluation: outline

- Inconsistency management
  - Types of inconsistency
  - Handling inconsistencies
  - Managing conflicts: a systematic process
- Risk analysis
  - Types of risk
  - Risk management
  - Risk documentation
  - DDP: quantitative risk management for RE
- Evaluating alternative options for decision making
- Requirements prioritization

DDP: quantitative risk management for RE

- DDP = Defect Detection Prevention
- Technique & tool developed at NASA [Feather, 2003]
- Quantitative support for Identify-Assess-Control cycles
- Three steps:
  - Elaborate risk Impact matrix
  - Elaborate countermeasure Effectiveness matrix
  - Determine optimal balance risk reduction / countermeasure cost
### Step 1: Elaborate the Impact matrix

- **Build a risk-consequence table** with domain experts for...
  - prioritizing risks by critical impact on all objectives
  - highlighting the most risk-driving objectives

- For each objective \( \text{obj} \), risk \( r \):
  \[
  \text{Impact}(r, \text{obj}) = \text{estimated loss of satisfaction of} \ \text{obj} \ \text{by} \ \text{r}
  \]
  \[
  \text{0 (no loss)} \rightarrow 1 \text{ (total loss)}
  \]

- Last line, for each risk \( r \):
  \[
  \text{Criticality}(r) = \text{Likelihood}(r) \times \sum_{\text{obj}} \text{Impact}(r, \text{obj}) \times \text{Weight(\text{obj})}
  \]

- Last column, for each objective \( \text{obj} \):
  \[
  \text{Loss(\text{obj})} = \text{Weight(\text{obj})} \times \sum_r \text{Impact}(r, \text{obj}) \times \text{Likelihood(r)}
  \]

---

### Impact matrix: example for library system

<table>
<thead>
<tr>
<th>Objectives</th>
<th>Late returns (likelihood: 0.7)</th>
<th>Stolen copies (likelihood: 0.3)</th>
<th>Lost copies (likelihood: 0.1)</th>
<th>Long loan by staff (likelihood: 0.5)</th>
<th>Loss obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular availability of book copies (weight: 0.4)</td>
<td>0.30</td>
<td>0.60</td>
<td>0.60</td>
<td>0.20</td>
<td>0.22</td>
</tr>
<tr>
<td>Comprehensive library coverage (weight: 0.3)</td>
<td>0.00</td>
<td>0.20</td>
<td>0.20</td>
<td>0.00</td>
<td>0.02</td>
</tr>
<tr>
<td>Staff load reduced (weight: 0.1)</td>
<td>0.30</td>
<td>0.50</td>
<td>0.40</td>
<td>0.10</td>
<td>0.04</td>
</tr>
<tr>
<td>Operational costs decreased (weight: 0.2)</td>
<td>0.10</td>
<td>0.30</td>
<td>0.30</td>
<td>0.10</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**Risk criticality**

- **0.12**
- **0.12**
- **0.04**
- **0.06**
Step 2: Elaborate the Effectiveness matrix

- **Build a risk-countermeasure table** with domain experts for...
  - estimating risk reduction by alternative countermeasures
  - highlighting most globally effective countermeasures

- For each countermeasure $cm$, weighted risk $r$:
  \[
  \text{Reduction}(cm, r) = \text{estimated reduction of } r \text{ if } cm \text{ applied}
  \]
  \[
  0 \text{ (no reduction)} \rightarrow 1 \text{ (risk elimination)}
  \]

- Last line, for each risk $r$:
  \[
  \text{combinedReduction}(r) = 1 - \prod_{cm} (1 - \text{Reduction}(cm, r))
  \]

- Last column, for each countermeasure $cm$:
  \[
  \text{overallEffect}(cm) = \sum_{r} (\text{Reduction}(cm, r) \times \text{Criticality}(r))
  \]

**Effectiveness matrix:** example for library system

<table>
<thead>
<tr>
<th>Countermeasures</th>
<th>Weighted risks</th>
<th>Overall effect of countermeasure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Late returns</td>
<td>Stolen copies</td>
</tr>
<tr>
<td></td>
<td>(likelihood: 0.7)</td>
<td>(likelihood: 0.3)</td>
</tr>
<tr>
<td>Email reminder sent</td>
<td>0.70</td>
<td>0</td>
</tr>
<tr>
<td>Fine subtracted from registration deposit</td>
<td>0.80</td>
<td>0</td>
</tr>
<tr>
<td>Borrower unregistration + insertion on black list</td>
<td>0.80</td>
<td>0.20</td>
</tr>
<tr>
<td>Anti-theft device</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Combined risk reduction</td>
<td>0.99</td>
<td>1</td>
</tr>
</tbody>
</table>
Step 3: Determine optimal balance
risk reduction vs. countermeasure cost

- Cost of each countermeasure \( cm \) to be estimated with domain experts
- DDP can then visualize ...  
  - risk balance charts: residual impact of each risk on all objectives if \( cm \) is selected
  - optimal combinations of countermeasures for risk balance under cost constraints
    - simulated annealing search for near-optimal solutions
    - optimality criterion can be set by user
      e.g. “maximize satisfaction of objectives under this cost threshold”
      “minimize cost above this satisfaction threshold”

Requirements evaluation: outline

- Inconsistency management
  - Types of inconsistency
  - Handling inconsistencies
  - Managing conflicts: a systematic process
- Risk analysis
  - Types of risk
  - Risk management
  - Risk documentation
  - DDP: quantitative risk management for RE
- Evaluating alternative options for decision making
- Requirements prioritization
Evaluating alternative options for decision making

- The RE process raises multiple alternative options of different types
  - alternative ways of satisfying a system objective
  - alternative assignments of responsibilities among system components
  - alternative resolutions of a conflict
  - alternative countermeasures to reduce a risk
- Preferred alternatives must be negotiated, selected ...
  - agree on evaluation criteria (e.g. contribution to NFRs)
  - compare options according to criteria
  - select best option

Qualitative reasoning for evaluating options

- Goal: determine qualitative contribution of each option to important non-functional requirements (NFRs):
  - very positively (++), positively (+), negatively (-), very negatively (--) 
- Example: meeting scheduling

<table>
<thead>
<tr>
<th>Options</th>
<th>Non-functional requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fast response</td>
</tr>
<tr>
<td>Get constraints by email</td>
<td>-</td>
</tr>
<tr>
<td>Get constraints from e-agenda</td>
<td>+ +</td>
</tr>
</tbody>
</table>

- Qualitative labels “+”, “-” on higher-level NFRs are obtained by bottom-up propagation from lower-level reqs in goal-subgoal refinement/conflict graph ([Chung et al 2000], see chap. 16)
- Given “+”, “-” contributions of each option to lowest-level reqs, option with best contribution to critical high-level NFRs is taken
Quantitative reasoning for evaluating options

- **Build a weighted matrix** for...
  - estimating score of each option on each evaluation criterion (weighted by relative importance)
  - selecting option with highest overall score on all criteria
- For each option $opt$, criterion $crit$:
  \[
  \text{Score}(opt, crit) = \text{estimated score percentage of } opt \text{ on } crit
  \]
  $0 \rightarrow 1$, $Y/100$ means "$crit$ satisfied in $Y\%$ of cases"
- Last line, for each option $opt$:
  \[
  \text{totalScore}(opt) = \sum_{crit} \text{Score}(opt, crit) \times \text{Weight}(crit)
  \]

<table>
<thead>
<tr>
<th>Evaluation criteria (NFRs)</th>
<th>Significance weighting</th>
<th>Option scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Get constraints by email</td>
</tr>
<tr>
<td>Fast response</td>
<td>0.30</td>
<td>0.50</td>
</tr>
<tr>
<td>Reliable response</td>
<td>0.60</td>
<td>0.90</td>
</tr>
<tr>
<td>Minimal inconvenience</td>
<td>0.10</td>
<td>0.50</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>1.00</strong></td>
<td><strong>0.74</strong></td>
</tr>
</tbody>
</table>

Requirements evaluation: outline

- **Inconsistency management**
  - Types of inconsistency
  - Handling inconsistencies
  - Managing conflicts: a systematic process
- **Risk analysis**
  - Types of risk
  - Risk management
  - Risk documentation
  - DDP: quantitative risk management for RE
- **Evaluating alternative options for decision making**
- **Requirements prioritization**
Requirements prioritization

- Elicited & evaluated reqs must be assigned priorities ...
  - conflict resolution
  - resource limitations (budget, personnel, schedules)
  - incremental development
  - replanning due to unexpected problems

- Some principles for effective req prioritization ...
  1. by ordered levels of equal priority, in small number
  2. qualitative & relative levels (“higher than”, …)
  3. comparable reqs: same granularity, same abstraction level
  4. reqs not mutually dependent (one can be kept, another dropped)
  5. agreed by key players

- Too early ranking at elicitation time might be subjective => risk of inadequate, inconsistent results

Value-cost prioritization

- Systematic technique, meets principles (1) - (3)

- Three steps:
  1. Estimate relative contribution of each req to project’s value
  2. Estimate relative contribution of each req to project’s cost
  3. Plot contributions on value-cost diagram: shows what req fits what priority level according to value-cost tradeoff
Estimating relative contributions
of requirements to project value & cost

- AHP technique from Decision Theory
  ("Analytic Hierarchy Process", [Saati, 1980])
- Determines in what proportion each req $R_i$, ..., $R_N$ contributes
to criterion $Crit$
- Applied twice: $Crit = \text{value}$, $Crit = \text{cost}$
- Two steps:
  1. **Build comparison matrix:**
     estimates how $R_i$'s contribution to $Crit$ compares to $R_j$'s
  2. Determine how $Crit$ distributes among all $R_i$

AHP, Step 1: Compare requirements pairwise

- Scale for comparing $R_i$'s contribution to $Crit$ to $R_j$'s:
  1: contributes equally
  3: contributes slightly more
  5: contributes strongly more
  7: contributes very strongly more
  9: contributes extremely more

- In comparison matrix, $R_{ji} = 1 / R_{ij}$ (1 ≤ i, j ≤ N)

<table>
<thead>
<tr>
<th>Crit: value</th>
<th>Produce optimal date</th>
<th>Handle preferred locations</th>
<th>Parameterize conflict resolution strategy</th>
<th>Multi-lingual communication</th>
<th>Meeting assistant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produce optimal date</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Handle preferred locations</td>
<td>1/3</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Parameterize conflict resolution strategy</td>
<td>1/5</td>
<td>1/3</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Multi-lingual communication</td>
<td>1/9</td>
<td>1/7</td>
<td>1/5</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>Meeting assistant</td>
<td>1/7</td>
<td>1/7</td>
<td>1/3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
AHP, Step 2: Evaluate how the criterion distributes among all requirements

- Criterion distribution = eigenvalues of comparison matrix

2.a Normalize columns: \[ R'_{ij} := \frac{R_{ij}}{\sum_i R_{ij}} \]
2.b Average across lines: \[ \text{Contrib}(R_i, \text{Crit}) = \frac{\sum_j R'_{ij}}{N} \]

<table>
<thead>
<tr>
<th>Produce optimal date</th>
<th>Handle preferred locations</th>
<th>Param. conflict resolution strategy</th>
<th>Multi-lingual communication</th>
<th>Meeting assistant</th>
<th>Relative value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.56</td>
<td>0.65</td>
<td>0.52</td>
<td>0.36</td>
<td>0.38</td>
<td>0.49</td>
</tr>
<tr>
<td>Handle preferred locations</td>
<td>0.19</td>
<td>0.22</td>
<td>0.31</td>
<td>0.28</td>
<td>0.28</td>
</tr>
<tr>
<td>Parameterize conflict resolution strategy</td>
<td>0.11</td>
<td>0.07</td>
<td>0.10</td>
<td>0.20</td>
<td>0.16</td>
</tr>
<tr>
<td>Multi-lingual communication</td>
<td>0.06</td>
<td>0.03</td>
<td>0.02</td>
<td>0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>Meeting assistant</td>
<td>0.08</td>
<td>0.03</td>
<td>0.03</td>
<td>0.12</td>
<td>0.05</td>
</tr>
</tbody>
</table>

- AHP has rules for ensuring consistent estimates & ratios

Plotting contributions on value-cost diagram

- Replay Steps 1 & 2 of AHP with Crit = \text{cost}
- Visualize value/cost contributions on diagram partitioned in selected priority levels
Requirements evaluation: summary

- Inconsistencies are frequent during req acquisition
  - For clashes in terminology, designation, structure: a glossary of terms is best
  - For weak, strong conflicts: variety of techniques & heuristics to support cycles "identify overlaps, detect conflicts, generate resolutions, select preferred"

- Product-/process-related risks must be carefully analyzed
  - Loss of satisfaction of system/development objectives
  - Variety of techniques for risk identification, incl. risk trees & their cut set
  - Likelihood of risks & consequences + severity need be assessed, qualitatively or quantitatively, with domain experts
  - Heuristics for exploring countermeasures, selecting cost-effective ones
  - DDP: an integrated quantitative approach for RE risk management

Requirements evaluation: summary (2)

- Alternative options need be evaluated for selecting preferred, agreed ones
  - Different types, incl. resolutions of conflicts & risks
  - Qualitative or quantitative reasoning for this (weighted matrices)

- Requirements must be prioritized
  - Due to resource limitations, incremental development
  - Constraints for effective prioritization
  - AHP-based value-cost prioritization: a systematic technique

*Model-driven evaluation provides structure & comparability for what needs to be evaluated* (see Part 2 of the book)