Lecture 5

- ➔ (Week 5 - Feb 5, 2024)

Requirements Classification and FURPS+ Model

- **Description and Priority**
    - Authenticated cafeteria Patrons can seamlessly place meal orders, choosing between delivery to a specified company location or pickup at the cafeteria.
    - Flexibility: Patrons can modify or cancel orders until preparation.
    - Priority: High – indicating critical functionality for user satisfaction.
- **Stimulus/Response Sequences**
- *Stimulus 1:* Patron initiates a meal order.
    - *Response:* System prompts the Patron for meal details, payment, and delivery instructions.
- *Stimulus 2:* Patron requests a change in a meal order.
    - *Response:* If the order status is "Accepted," the system facilitates seamless editing.
- *Stimulus 3:* Patron requests to cancel a meal order.
    - *Response:* If the status is "Accepted," the system cancels the order.
- **Functional Requirements**
    - The system allows logged-in Patrons to effortlessly place meal orders.
    - Patron eligibility for payroll deduction is verified before order placement.

Non-Functional Requirements: FURPS+ Model

- **FURPS+ Model Categories:**
    - **Functionality:** Encompasses features, capabilities, security, and documented user requirements (e.g., system use cases).
    - **Usability:** Specifies requirements for a user-friendly interface, including user-friendliness, accessibility, look and feel, online help, and visual design guidelines.
    - **Reliability:** Covers system performance under routine and nonroutine conditions, with metrics like MTBF, MTBSI, and MTTR.
    - **Performance:** Describes system behavior concerning time and resources, including speed, efficiency, availability, accuracy, response time, recovery time, start-up time, resource usage, and throughput.
    - **Supportability:** Outlines requirements related to monitoring, maintenance, testing, configuration, installation, and upgrades.
- **Plus (+) Category:**
- Additional Constraints:
        - Design requirements.
        - Implementation requirements: Constraints on coding, construction, platforms, coding languages, and standards.
        - Interface requirements: Capability to interact with external systems, specifying protocols, formats, etc.
        - Physical requirements: Constraints on hardware, including size, temperature control, and materials.
    - Legal, compliance, regulatory, and copyright requirements and constraints.

FURPS+ Checklist:

- **Functionality:**
    - Emphasis on well-defined features, capabilities, security, and user-centric requirements.
- **Usability:**
    - Focus on an intuitive user interface, encompassing user-friendliness, accessibility, and visual design guidelines.

- **Reliability:**
    - Clear metrics for system performance under routine and nonroutine conditions.
- **Performance:**
    - Comprehensive coverage of system behavior regarding time and resources.
- **Supportability:**
    - Detailed requirements for monitoring, maintenance, testing, configuration, installation, and upgrades.
- **Plus (+):**
    - Explicit constraints and considerations, ensuring a holistic understanding of the system's operational landscape.

Estimating Cost & Value and Selection

1. Approaches for Estimating Cost & Value

- **Absolute Scale vs. Relative Values:**
- *Absolute Scale (Dollar Values):*
    - Requires extensive domain experience.
- *Relative Values (Less/More; A Little, Somewhat, Very):*
        - Easier to elicit, simplifies prioritization.
    - Prioritization becomes a sorting problem.

2. Complications in Estimation

- **Quantifying Differences:**
    - Challenges in quantifying differences between requirements.
- **Non-Comparable Requirements:**
    - Differences in abstraction levels or core functionality vs. enhancements.
- **Dependency and Consistency:**
    - Requirements may not be independent; stakeholders may lack consistency.

3. Hierarchical Prioritization

- **Grouping into Hierarchy:**
    - Requirements grouped hierarchically (e.g., goal tree, NFR tree).
    - Comparisons made within branches of a single node to minimize costs.

4. Analytic Hierarchy Process (AHP)

- **Matrix Creation (n x n):**
    - Values (1, 3, 5, 7, 9) indicate preference strength.
    - Reciprocal values entered for (y, x).
- **Eigenvalue Estimation:**
    - Averaging over normalized columns.
    - Calculation of column and row sums.
    - Provides estimated percentage of total value for each requirement.

5. AHP Example - Estimating Costs

- **Normalization and Calculation:**
    - Normalizing columns and summing rows.
    - Results in a percentage breakdown of cost allocation for each requirement.

6. ROI Graph and Other Selection Criteria

- **ROI Calculation:**
    - AHP process performed twice for value and cost estimation.
    - ROI ratio calculated: Cost (percent) / Value (percent).
    - Allows for prioritization based on ROI.
- **Other Selection Criteria:**

- **Above Average Cost and Value:**
    - Grouping requirements with both high cost and high value.
- **Relative Loss, Probability, and Risk Exposure:**
    - Introducing additional criteria for grouping and prioritization.

7. Result Plotting:

- **ROI Graph:**
    - Utilizes the ROI ratio to categorize requirements into low, medium, and high priority.
- **Other Selection Criteria Graphs:**
    - Illustrates alternative groupings based on different criteria.

8. Considerations:

- **Comprehensive Decision-Making:**
    - Choosing the appropriate approach based on project context.
    - Ensuring a holistic understanding of cost, value, and risk factors.

9. Continuous Refinement:

- **Iterative Process:**
    - Regularly revisit and refine estimations based on evolving project dynamics.
    - Engage stakeholders for consistent evaluation and adjustments.

Requirement and Stakeholder Conflicts

1. Nature of Conflicts:

- **Functional vs. Non-Functional Requirements:**
    - Inherent conflict between different types of requirements.
- **Stakeholder Defensiveness:**
    - Stakeholders may strongly advocate for their requirements, leading to conflicts.
- **Negotiation and Compromise:**
    - Regular negotiations are essential to reach compromises.
    - Documenting conflicts and decisions is crucial for archival purposes.

2. Examples of Conflicting Requirements:

- **Definition:**
    - Conflict arises when implementing one requirement impedes the fulfillment of another.
- **Example:**
    - Conflict between making a product "available to all" and ensuring it is "fully secure."
    - E.g., Human Resources requesting employee age vs. Data Privacy team opposing its capture.

3. Severity of Conflicts:

- **Definition:**
    - Assessing the magnitude and impact of conflicts.
- **Considerations:**
    - Severity depends on the nature of conflicting requirements and their implications for the project.

4. Causes of Conflict:

- **Deutsch's Factors (1973):**
    - Control over resources, preferences, values, beliefs, and nature of relationships.
- **Robbins' Factors (1989):**
    - Communicational, structural, and personal factors.

5. Interesting Results from Conflict Studies:

- **Deviant Behavior and Conflict:**
    - Normal in small group decision-making.
- **Communication Impact:**
    - More aggression with restricted communication; conflict intensifies.

- **Team Heterogeneity:**
    - More conflict, but also more diverse perspectives.
- **Homogeneous Groups:**
    - Prone to high-risk decisions (groupthink).

6. Conflict Resolution Approaches:

- **Negotiation:**
    - Collaborative exploration seeking a mutually satisfactory settlement.
    - Integrative behavior or constructive negotiation.
- **Third-Party Resolution:**
    - Participants appeal to an external source, e.g., rule-book, authority figure, or coin toss.
    - Judicial, Extra-judicial, and Arbitrary types.
- **Bidding and Bargaining:**
    - Bidding involves stating desired terms.
    - Bargaining seeks a satisfactory integration of bids.

7. Key Takeaways:

- **Normalcy of Conflict:**
    - Conflicts are natural in decision-making processes.
- **Communication's Role:**
    - Restricted communication intensifies conflict; contact hypothesis.
- **Team Dynamics:**
    - Heterogeneous teams experience more conflict; homogeneous groups prone to groupthink.
- **Personality Factors:**
    - Overshadowed by situational and perceptual elements.

8. Practical Implications:

- **Documentation:**
    - Record all conflicts and decisions for transparency and archival purposes.
- **Iterative Conflict Resolution:**
    - Conflict resolution is an ongoing, iterative process.
- **Stakeholder Engagement:**
    - Active involvement and communication to manage conflicting requirements effectively.

Goal Models in Requirements Engineering

1. When and Why to Use Goal Models:

- **Timing:** Early requirements elicitation phase.
- **Purpose:**
    - Focus on identifying problems and exploring system solutions.
- **Sequence:** Conducted before UML modeling.
- **Rationale:**
    - Provides a clear rationale for requirements.
    - Identifies stable information to guide requirement elaboration.

2. Identifying Stakeholders' Goals:

- **Approach:**
    - Focus on the "why" of a system's requirement.
    - Express the 'why' as a set of stakeholder goals.
    - Use goal refinement to derive specific requirements.
- **Goal Analysis:**
    - Document, organize, and classify goals.
    - Goal evolution involves refining, elaborating, and operationalizing goals.

        - Goal hierarchies display refinements and alternatives.

3. Softgoals:

- **Definition:**
    - Some goals can never be fully satisfied, treated as softgoals.
    - Examples include "system be easy to use" or "access be secure."
    - Also known as 'non-functional requirements' or 'quality requirements.'
- **Approach:**
    - Look for factors contributing to satisficing the softgoals.

4. Goal Refinement:

- **Definition:**
    - Expressing how a more abstract goal can be achieved through lower-level goals.
    - Includes AND and OR refinements.

5. Goal Analysis:

- **Goal Elaboration:**
    - "Why" questions explore higher goals (context).
    - "How" questions explore lower goals (operations).
    - "How else" questions explore alternatives.
- **Relationships:**
    - Goals can help, hurt, make, break, or have precedence over others.
- **Obstacle Analysis:**
    - Explores potential obstructions to goals and their consequences.

6. Softgoals as Selection Criteria:

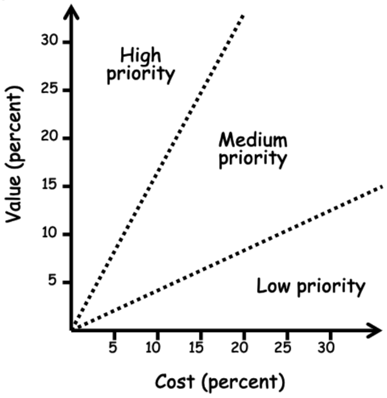
- **Approach:**
    - Evaluate hard goal contributions to soft goals.
    - Softgoals guide the selection of alternatives based on their contributions.

7. Summary:

- **Requirements Classifications:**
    - Categorization of requirements based on functionality, usability, reliability, performance, and supportability.
- **Requirements Prioritization:**
    - Process of organizing and ranking requirements based on their importance.
- **Conflicts:**
    - Identification and resolution of conflicts arising from differing requirements.
- **Requirements Negotiation:**
    - Collaborative exploration to find mutually satisfactory settlements.
- **Stakeholder Goals and Goal Analysis:**
    - Identifying the "why" of system requirements and analyzing goals hierarchically.

8. Next Week:

- **Goal Modeling:**
    - Further exploration and practical application of goal modeling in the context of requirements engineering.



Different categories of Goals (Requirements)

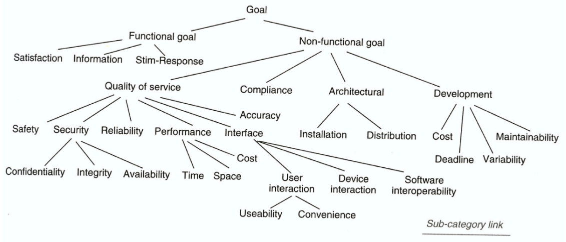
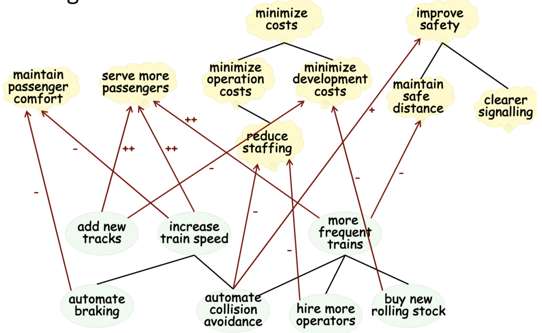


Figure 7.5 Goal categories

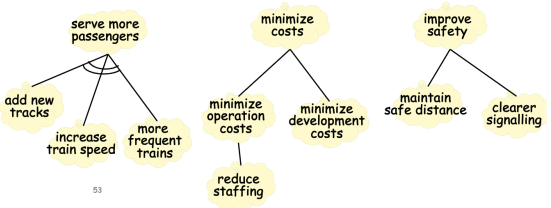
This is the same as the classification of requirements into functional and non-functional (with all its sub-categories)

Softgoals as selection criteria



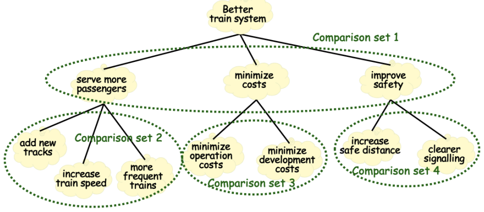
Softgoals

- Some goals can never be fully satisfied
  - Treat these as softgoals
    - E.g. "system be easy to use"; "access be secure"
    - Also known as 'non-functional requirements'; 'quality requirements'
  - Will look for things that contribute to satisficing the softgoals
  - E.g. for a train system:

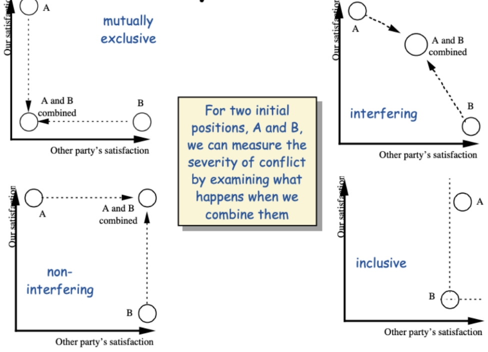


Hierarchical Prioritization

- Group Requirements into a hierarchy
  - E.g. A goal tree
  - E.g. A NFR tree
- Only make comparisons between branches of a single node:



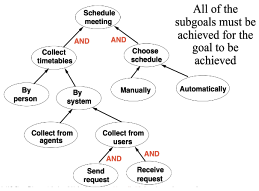
Severity of Conflict



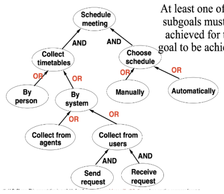
For two initial positions, A and B, we can measure the severity of conflict by examining what happens when we combine them

Goal refinement

- Goal refinement: expressing how a more abstract goal can be established by a set of more low-level goals
  - AND refinement
  - OR refinement



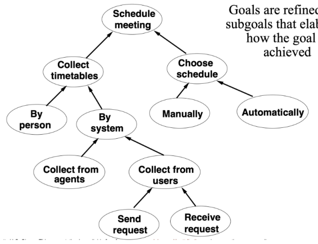
All of the subgoals must be achieved for the goal to be achieved



At least one of the subgoals must be achieved for the goal to be achieved

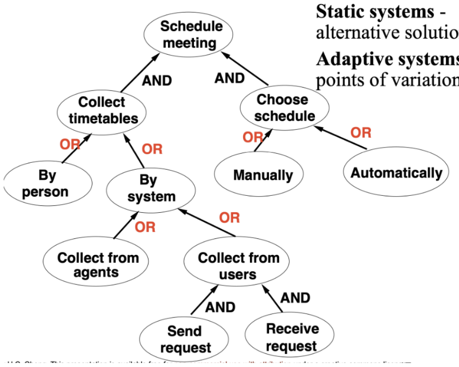
Running Example

- Meeting Scheduler
  - Assists the initiator in scheduling a meeting
  - Meeting should be convenient for participants
    - Participants should be available
  - Modeled using the i* goal notation



Goals are refined into subgoals that elaborate how the goal is achieved

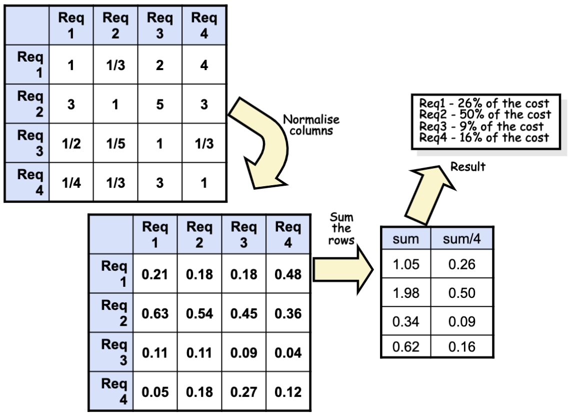
Interpretations of OR Refinements



Static systems - alternative solutions

Adaptive systems – points of variation

AHP example - estimating costs



	Req 1	Req 2	Req 3	Req 4
Req 1	1	1/3	2	4
Req 2	3	1	5	3
Req 3	1/2	1/5	1	1/3
Req 4	1/4	1/3	3	1

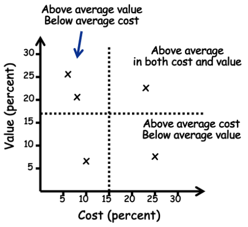
	Req 1	Req 2	Req 3	Req 4
Req 1	0.21	0.18	0.18	0.48
Req 2	0.63	0.54	0.45	0.36
Req 3	0.11	0.11	0.09	0.04
Req 4	0.05	0.18	0.27	0.12

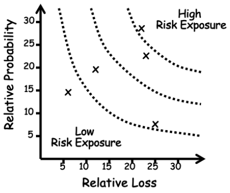
sum	sum/4
1.05	0.26
1.98	0.50
0.34	0.09
0.62	0.16

Req1 - 26% of the cost
Req2 - 50% of the cost
Req3 - 9% of the cost
Req4 - 16% of the cost

Other selection criteria

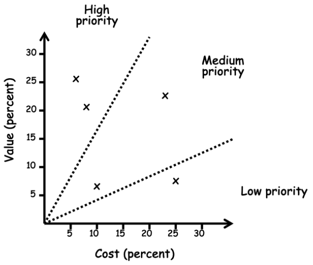
- ROI ratio is not the only way to group requirements





Plot ROI graph

- Do AHP process twice:
  - Once to estimate relative value
  - Once to estimate relative cost
- Use results to calculate ROI ratio:


High
priority
Medium
priority
Low priority
Value (percent)
Cost (percent)