Goals

- Get Whole Picture of RE (Requirements Engineering)
  - RE Consists Diverse Disciplines on Business, Systems and Software

- Understand 4 Core Technique of RE
  - Elicitation, Analysis, Specification, and Verification, Validation and Evaluation

- Understand Framework of REBOK
  - REBOK is a Map for Navigating the RE
Scenario

1. RE is the Key to Success
2. REBOK: Why and What?
3. RE Fundamentals and Process
4. RE Core Techniques
5. RE Practice for Success

1. RE is the Key to Success
RE is the Key to Success
Requirements is the Source of the Development

Requirements is the Source of the Development

“Without Right Requirements, Any Development Will Fail”


System/Software Development

Requirements Engineering

Input Requirements Specification

Output (Software) System

Software Development

RE is the Key to Success
Big Challenge

Frequently Quoted: “The Biggest Causes for the Failure of Development is Requirements”
Many Horrible Stories

CHAOS Report by Standish Group (1995) *

The Top 3 Project Challenged Factors (36.9%) Lie in Requirements Process

Lack of User Input = 12.8%
Incomplete Requirements & Specifications = 12.3%
Changing Requirements & Specification = 11.8%

RE is the Key to Success

Key to Success (and Failure if Ignoring)

RE is the Most influential to both Success & Failure

Project Management
Human Resource and Organization
Coordination between Project/Organization
Human Skill and Education
Requirements Engineering
Configuration Management
Outsourcing/Contract Management
Time Management
Quality Control Organization
Review

Source: JISA Survey on Japanese Software Development
JISA: Japan Information Technology Services Association [http://www.jisa.or.jp/e/]

RE is the Key to Success
Effect of RE over Software Development

ROI of RE: NASA Statistics*

- RE Reduces Cost Overrun
  - 2-3% for RE: Overrun=80-200%
  - 8-12% for RE: Overrun=0-50%

Statistics from the Ministry of Economy, Trade, and Industry/JUAS**

- If RE is NOT Seriously Conducted, Development within Schedule is 0%, Schedule Significantly Overrun is 60%

RE is the Key to Success
Challenges in RE Practice

4 Most Influential Factors with Users [64.3%]
- Unable to Control Scope
- Spec Don’t Meet Req.
- Users Don’t Know Req.
- Frequent Req. Change

Why RE is So Difficult

Essential Diversity/Instability of Requirements
- Space: Scope of the Req. is Hard to See and Easy to Creep
- Time: Req. Changes over Time
- People: Source of Req. are People Who May NOT Know (or Be Aware of) their Req.
- Social: Req. is not Just Technical Issues, but Suffers the Political and Social Influence

Relatively Young Discipline in Software Engineering
- Req. Definition is Practiced from the Beginning of Software Development, but RE is Relatively Young and Less Matured [Int'l RE Conference Started in 1993]

RE is Beyond SE
- RE is NOT Only for Software, but also System & Business
RE is the Key to Success
Now, RE is “Maturing”

- Many BIG Books Have Been Published by RE Researchers: 700~800 Pages
- More than 40 Books on RE Published in Japan

REBOK: Why and What?
REBOK: Why and What?

Practitioners Frequently Asked on RE

- What is RE?
- What is Requirements?
- What Techniques are Needed to Learn for Requirement Development?
- What is RE Process and What we should do?
- Which RE Techniques are Good/Poor to a Domain?
- How to Practice RE?
- How to Educate/Train RE Professionals?
- Which Book is Suggested to Read?

Emerging BOKs & Certification Programs in RE

- Emerging BOKs and Syllabi Related to RE
  - **NO Comprehensive BOK Covering Whole RE, and from Basic to Expert**

<table>
<thead>
<tr>
<th>BOK</th>
<th>SWEBOK (Software Engineering BOK)</th>
<th>BABOK (Business Analysis BOK)</th>
<th>Syllabi for CPRE (Certified Professional Req. Engineer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org.</td>
<td>IEEE CS</td>
<td>IIBA (Canada)</td>
<td>IREB (Germany)</td>
</tr>
<tr>
<td>Profession</td>
<td>Software Engineer</td>
<td>BA (Business Analyst)</td>
<td>Requirements Engineer</td>
</tr>
<tr>
<td>Knowledge</td>
<td>Software Engineering (Chap. 2 devoted to RE)</td>
<td>Business Analysis</td>
<td>Basic Knowledge on RE</td>
</tr>
<tr>
<td>Certification</td>
<td>CSDP, CSDA</td>
<td>CBAP</td>
<td>CPRE</td>
</tr>
<tr>
<td>Prerequisite &amp; Level of Certification</td>
<td>Software Development Experience over 4 yr (7,000hr)[2 yr if]</td>
<td>BA Experience over 5yr (7,500 hr)</td>
<td>Foundation Level (Advance, Expert is Under Planning)</td>
</tr>
</tbody>
</table>
REBOK: Why and What? Related BOKs on RE: SWEBOK 2004

SWEBOK (Software Engineering Body of Knowledge)
- RE is a KA of 10 KAs on Software Engineering

REBOK: Why and What? Related BOKs on RE: BABOK 2.0

6 KA Defined as Processes with Input, Task, Output
- Focus on Expert for Business/Enterprise Analysis
- Little Concerns on Engineering System/Software Req.
- Rather Different Definition of KA and their Relationship
- No Explicit KA on Req. Specification


REBOK: Why and What?
Call for a Guidebook to Learn & Practice RE

JISA RE WG [Chair: Aoyama]: Studying RE in Practical Context
- More than 100 Practitioners Joined from 2006 to 2008
- Challenges in RE Practice
  - Wide Scope of RE Practice, and
  - Diversity of Necessary Knowledge and Skills

FY '06: Organizational Approach to RE Practice: RE Coordinator
FY '07: Collecting Best Practices and Publish them as RE Patterns

FY '08: Initiate REBOK Development Program
July 2011: REBOK Version 1.0 Published

REBOK: Why and What?
Mission of REBOK: Map of Whole RE

Mission: Provide a Map (Whole Picture) of RE
- Based on Knowledge of Global RE Research Community
- Provide a Common Language across Related BOKs and Standards
- Continuous Extension from Business/Product Req. to System Req., and Software Req.

Application Domain of REBOK
- Business (Enterprise) and Product (Embedded)

----------------------|------------|---------------|-----------------'
BABOK & CBAP [IRBA]   | REBOK      | SWEBOK and CSDP, CSDA [IEEE] | CPRE [IREB]
REBOK: Why and What?  
Model of Requirements

3 Layer of Req.: Business/Product, System, Software
- Business: Capability and Related Aspects of a Business
- (Information) System: Including Hardware and Software
- Software: Requirements to be Realized by Software

Smooth Elaboration of Req. from Business to Software

REBOK: Why and What?  
Actor and Scopes to RE

Stakeholder: People Involving to Req.
Actor: Role of People Involving to RE
- RA(Req. Analyst): Practice and Lead RE
- User: Use RE Outcome
- Supporter: Promote RE Practice in the Institute
REBOK: Why and What? Scopes of Req. & Req. Analyst

3 Scopes of Req.

- Business/Product, (Information) System, Software
  - Product: Package, Embedded Product

3 Role Models of Req. Analyst Work to 3 Scopes of Req.

- Business/Product Analyst, System Analyst, Software Analyst
- Requirements Analyst as a Generic Role

Environment: Stakeholder, Market, Society, Regulation, etc.

<table>
<thead>
<tr>
<th>Business/Product</th>
<th>Information System</th>
<th>Software System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>System Requirements</td>
<td>Software</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Engineer</td>
</tr>
</tbody>
</table>

REBOK: Why and What? Profession and Role Involving in RE

Professions and Roles Defined by Major BOK

- RA: Requirements Analyst
- BA: Business Analyst, SE: Systems Engineer

<table>
<thead>
<tr>
<th>BOK (Publisher) [Ref.]</th>
<th>REBOK (JISA) [REBOK 11]</th>
<th>BABOK (IIBA) [IIBA 09]</th>
<th>CPRE Syllabus (IREB) [Pohl 11]</th>
<th>SWBOK (IEEE/ACM) [Abran 04]</th>
<th>SEBoK (INCOSE) [Pyster 11]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic</td>
<td>RA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business</td>
<td>BA</td>
<td>BA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product</td>
<td>Product Analyst</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>Systems Analyst</td>
<td></td>
<td></td>
<td></td>
<td>Systems Analyst/SE</td>
</tr>
<tr>
<td>Software</td>
<td>Software Engineer</td>
<td>Requirements Engineer</td>
<td></td>
<td></td>
<td>Software Engineer</td>
</tr>
</tbody>
</table>
REBOK: Why and What?
Organizing Knowledge on RE

Challenges in Organizing RE Knowledge for REBOK

- Developing Model and Architecture for Organizing Knowledge on RE
- Assuring Consistency of REBOK at Certain Level with SWEBOK and BABOK

3+1 Knowledge Model of REBOK

- KA (Knowledge Area): Basic Unit of Knowledge
- KU (Knowledge Unit): Sub-unit of KA
- Technique: A unit of Knowledge Commonly Used across KAs

<table>
<thead>
<tr>
<th>REBOK</th>
<th>SWEBOK Software Req.</th>
<th>BABOK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Category</td>
<td>RE=one of 10 KAs</td>
<td>—</td>
</tr>
<tr>
<td>KA (Knowledge Area) [8]</td>
<td>KU (Knowledge Unit) [7]</td>
<td>KA (Knowledge Area) [7]</td>
</tr>
<tr>
<td>KU (Knowledge Unit)</td>
<td>Sub-Area</td>
<td>Task [38]</td>
</tr>
<tr>
<td>Technique</td>
<td>Topic, Sub-Topic</td>
<td>Technique [34]</td>
</tr>
</tbody>
</table>

REBOK: Why and What?
REBOK Extensible Knowledge Architecture

Knowledge Category: Separation of Commonality and Variability
- Embracing the Knowledge Diversity of RE

REBOK Core
- Common Knowledge of REBOK

REBOK Extension
- Interface to Specific Technical Knowledge
- Ex.: Interface to Domain Knowledge
- Enterprise Analysis, Product Analysis
REBOK: Why and What?
REBOK Core: Structure of 8 KAs

- Extended from the SWEBOK Software Req.
  - Add KA of Req. Planning and Management
  - Extend KA of Req. V&V and Evaluation

- Clearly Define the Technical and Process Knowledge

Note: Number in the Box Indicates the Chapter of REBOK

REBOK: Why and What?
REBOK Core: Overview of 8 KAs

- Technical Knowledge
  - RE Fundamentals, RE Process, Req. Planning and Management, Practical Consideration

- Process Knowledge
  - Elicitation, Analysis, Specification, V & V & Evaluation

<table>
<thead>
<tr>
<th>KA</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. RE Fundamentals</td>
<td>Definition and essential properties on requirements.</td>
</tr>
<tr>
<td>2. RE Process</td>
<td>Concept and models of requirements engineering process.</td>
</tr>
<tr>
<td>3. Req. Elicitation</td>
<td>Sources and techniques for requirements elicitation</td>
</tr>
<tr>
<td>4. Req. Analysis</td>
<td>Techniques for analyzing requirements elicited</td>
</tr>
<tr>
<td>5. Req. Specification</td>
<td>Specification techniques for requirements analyzed</td>
</tr>
<tr>
<td>6. Req. Verification, Validation &amp; Evaluation</td>
<td>Techniques validating requirements specification</td>
</tr>
<tr>
<td>7. Req. Planning and Management</td>
<td>Properties, metrics and management techniques of requirements</td>
</tr>
<tr>
<td>8. Practical Consideration</td>
<td>Patterns and best practices for practicing requirements engineering</td>
</tr>
</tbody>
</table>
REBOK: Why and What?
Req. Scope and Knowledge Scope

Certain Consistency with:
- Bridging from Business/Product to Solution
- Solution Req. is Decomposed to Systems Req. and Software Req.

<table>
<thead>
<tr>
<th>Scope</th>
<th>REBOK</th>
<th>BABOK</th>
<th>ISO/IEC 12207</th>
<th>ISO/IEC/IEEE 29148</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business/Product</td>
<td>Business Req.</td>
<td>Product Req.</td>
<td>Business Req.</td>
<td>-</td>
</tr>
<tr>
<td>Software</td>
<td>Software Req.</td>
<td></td>
<td>Software Req.</td>
<td>Software Req.</td>
</tr>
<tr>
<td>Operation</td>
<td>Transition Req.</td>
<td>Transition Req.</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Operation Req.</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>


3.
RE Fundamentals and Process
**RE Fundamentals**

**Definition of System and Context**

- **System**: Is Not Isolated, but Depends on Environment
- **Context**: Environmental Factors Influential to a System
  - User, Other Systems, Law
- **System Boundary**: Define the Scope of System
  - Identification of the Boundary of System/Context is the Key to Control of Scope
- **Boundary Area**: Boundary Might be Unclear

![Diagram of System and Context](image)

**RE Fundamentals**

**System, Requirements and RE**

- **Systems-As-Is**: Current System with Problems
- **System-To-Be**: Desired System for Solving Problems
- **Requirements**
  - Capabilities and Related Properties Necessary to Realize System-To-Be from System-As-Is
RE Fundamentals
Definition of Requirement

- Conventional Definitions
  - BABOK: Replaced "User" in IEEE 610.12 with "Stakeholder"
  - REBOK: Follows IEEE 610.12/BABOK

- A requirement is
  - A condition or capability needed by a stakeholder to solve a problem or achieve an objective.
  - A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
  - A documented representation of a condition or capability as in (1) or (2)


RE Fundamentals
Definition of Requirements Engineering (RE)

- Definition of Requirements Engineering (RE)
  - RE is a coordinated set of activities for exploring, evaluating, documenting, consolidating, revising and adapting the objectives, capabilities, qualities, constraints and assumptions that the system-to-be should meet based on problems raised by the system-as-is and opportunities provided by new technologies.

- RE(Int’l Requirements Engineering Conference)
  - Annually Since 1993, Kyoto Japan in 2004

RE Fundamentals
Requirements=FR+NFR

FR (Functional Requirements)
- Functional Effects that the System-To-Be is required to Have on its Environment

NFR (Non-Functional Requirements)
- Constraints on the Way the System-To-Be Should Satisfy its Functional Requirements or on the Way it Should be Developed

NFR Defined Quality, and Helps to Design Architecture
- Multiple Architecture Candidates for a Single Function
- ASR (Architecture Significant Requirements): NFR
  Strongly Influencing on Architecture Design


NFR=Quality Req. + Constraint

Standards on Quality Characteristics

A Taxonomy of NFR

RE Core Techniques

RE Process: Incremental & Iterative Process

Incremental RE Process
- From Business/Product, thru System to Software

Iterative RE Process within a Scope
- Elicitation, Analysis, Specification, V & V & Evaluation

Requirements Source
- Business Strategy, Stakeholder Req., Document, Environment

Iterative Process within a Scope
- Iterate from Elicitation, Analysis, Specification and V&V&E

Elicitation, Analysis, Specification, V&V, and Evaluation
REBOK: Why and What?

REBOK Core: Process Structure

- REBOK Process: Iterative Process in a Scope
- RE = RD(Req. Development) + RM(Req. Management)

ISO/IEC/IEEE 29148 Requirements Engineering

- RE Process Conforming to Related Life Cycle Processes:
  - ISO/IEC 12207(Software Life Cycle Processes)
  - ISO/IEC 15288(System Life Cycle Processes)
- 3-Layer Hierarchical Req. Model [Conforming to REBOK]
  - Business(Stakeholder), System, Software
- Incremental and Iterative RE Process [Conforming to REBOK]
  - Iterative Application of Process: Repeat within a Scope
  - Recursive Application of Process: Incremental across Successive Scopes
4. RE Core Techniques

- **Requirements Elicitation**
- **Requirements Analysis**
- **Requirements Specification**
- **Requirements Verification, Validation and Evaluation**

**RE Core Techniques**

**Req. Elicitation: Shaping Requirements**

- Stakeholder Req. Include Undocumented/Unspoken Req. and Even Unaware/Unrealistic Req.
- Elicitation Needs Structured Activities Backed by Techniques

- **Stakeholders' View (Req. Source)**
- **Strategy, Roadmap**
- **Documents on Current System**
- **Req. Elicitation**
- **Collection of Req. Items**
- **Documented Req. Items**
- **Selected and Structured Req. Items**
- **Req. Analysis**
- **Req. Specification**
- **Req. Verification, Validation, and Evaluation**
- **Well-Formed and Valid Req. Spec.**
- **Scope of Feasible Req.**
- **Vendors' View**
3. Elicitation

- System-As-Is
- Stakeholder

Req. Analyst
SME

RE Core Techniques
Req. Elicitation: Elicitation Process

Legend
Control, Constraint
Input → Activity → Output
Mechanism, Technique, Actor

SME: Subject Matter Expert
i.e. Domain Expert

4. Analysis

Req. Analyzed
Model of System As-Is
Goal Model
Model of System-To-Be
Req. Items
Stakeholder

3.2 Understand System-As-Is
Technique to Understand System-As-Is

3.3 Model System-As-Is
Modeling Techniques

3.4 Identify Problems & Causes
Problem Analysis

3.5 Identify Goals
Goal Analysis

3.6 Analyze Goals
Goal Analysis Technique

3.7 Model System-To-Be
Model of System-To-Be

3.8 Describe Req. Items
Documentation Technique

3.1 Analyze Stakeholder
Stakeholder
SME
Req. Analyst

RE Core Techniques
Req. Elicitation: Elicitation Process (Detail)
RE Core Techniques

Req. Elicitation: 3 Key Techniques

- Elicitation = Entry of RE Process = Key to RE
- Core Techniques in Req. Elicitation:
  - Stakeholder Analysis: Seize Key Person
  - Goal Analysis: Elicit “True” Req. and Agree on It with Users
  - Scenario Analysis: Visualize UX (User Experience) and Elicit Req. from User’s Viewpoint

Stakeholder Analysis
- Identify Stakeholders and their Relationships
- Analyze Influence and Risk of Stakeholders

Stakeholder(s)
- People or Organization Involving in a System
- Req. Source
- The First Step in Req. Elicitation

Ref.: A. Rotem-Gal-Oz, From Stakeholders to Models: It Is All a Matter of Viewpoints, Apr. 2007,
http://msdn2.microsoft.com/en-us/library/bb447667.aspx#04_03_views_topic1
RE Core Techniques

Req. Elicitation: Stakeholder Analysis(2)

Stakeholder Matrix: Influence and Importance
- Visualize the Position of Stakeholders and their Risk

Influence: Power to Decision Making
- Classification Example
  - Primary Customers: Target Stakeholders
  - Secondary Customer: Other Stakeholders
- Stakeholder by Regulation
  - Complier: Enforcing Compliance

Importance: Necessity for the Realization of System
- Mandatory
- Desirable
- Nice to Have

Stakeholder Matrix

Risk

Influence

Importance

Key to Success
“Find Key Person”

RE Core Techniques

Req. Elicitation: Understanding Users(1)

Understanding Users
- By Deep Understanding Users
  - Identification of Target Users for Market-Driven Req.
  - Improve Usability and User Experience
- What is “Deep” Understanding of User
  - Attitude/Behavior of User to a System
  - Usage Context
  - Detail Scenario/User Story

“Understanding User’ and “User Modeling”
- “User Modeling” is Intended to Elaborate Cognitive Model of User in Human-Computer Interaction
RE Core Techniques

Req. Elicitation: Understanding Users (2)

- Method for Understanding User
  - User Modeling
    - Persona
  - Collecting User Information
    - Observation
    - User Profiling
    - Life Log
  - Analysis Methods of User Information
    - Conjoint Analysis (Popular in Marketing)
    - Data Mining
    - Machine Learning
    - Collaborative Filtering
    - Baysian Network

Req. Elicitation: Understanding System-As-Is

- Purpose
  - Understand a System-As-Is Based on the Information from Stakeholders
- Methods to Understand System-As-Is
  - User-Driven (Bottom Up)
    - Scenario, User Story
    - Ethnomethodology/Ethnography
  - Model-Driven (Top Down)
    - Conceptual Modeling [Domain Modeling]
    - Zachman Framework
  - Domain-Specific
    - Ergonomics
RE Core Techniques

Req. Elicitation: Zachman Framework

DRaw Whole Picture: Classify with 5W1H in Top Down
*
Combining Stepwise Refinement and Separation of Concerns

<table>
<thead>
<tr>
<th>(What)</th>
<th>(How)</th>
<th>(Where)</th>
<th>(Who)</th>
<th>(When)</th>
<th>(Why)</th>
</tr>
</thead>
<tbody>
<tr>
<td>System/Logical</td>
<td>Data Model</td>
<td>DFD</td>
<td>Distribution Architecture</td>
<td>WBS</td>
<td>Event Diagram</td>
</tr>
<tr>
<td>Details/SubContractor</td>
<td>Data Schema</td>
<td>Program (Function)</td>
<td>Network Architecture</td>
<td>SOW</td>
<td>Event Details</td>
</tr>
</tbody>
</table>


All Rights Reserved, Copyright Mikio Aoyama, 2011

RE Core Techniques

Req. Elicitation: Scenarios and User Stories(1)

Method to Discover Scenarios and User Stories
* Questionnaire/Interview
* Observation
* RE Workshop
* Documents Analysis

Methods of Describing Scenarios and User Stories
* UML
  * Use Case: Illustrating Context and Functions/Services
  * Scenario of Use Case: Interaction between Users and a System
  * Sequence Diagram: Description of Temporal Interaction
  * Rich Picture

All Rights Reserved, Copyright Mikio Aoyama, 2011
RE Core Techniques

Req. Elicitation: Scenarios and User Stories(2)

Security/Safety Req.
- “Harmful Thing Should NOT Happen” is Difficult to Define and Validate
- Change the Req. Model: Define Threat, and Find the Req. to Protect the System from the Threat

Elicitation of Security/Safety Req.
- Identify Threat
- Identify Use Cases as Req. to Mitigate the Threat
- Ensure that the Req. Mitigate the Threat


RE Core Techniques

Req. Elicitation: Scenarios and User Stories(3)

Identify Security Use Case by Misuse Case
- Use Case: Functions Requested
- Misuse Case: Use Case Should NOT Be Happen
  - Driven By Threat: Malicious Actor

Combinatorial Use of Use Case and Misuse Case
- Identify Misuse Case to Threaten the System
- Identify Use Case to Mitigate the Misuse Case

RE Core Techniques
Req. Elicitation: Goal Analysis(1)

Goal[Objective, Purpose, Intention]
- State of System-To-Be
- Goals are Represented as State or Behavior
- Ex: “Stock products that customers want when they want them according to changing needs”*

Why Goals (How Goals Work)
- To Meet Goals is To Solve the Problems
- Goals are “True” Req., FR/NFR are Means to Meet the Goals
- Agree Goals to Stakeholders is the Key to Success

Discovering Goals
- From Problem: Problem of System-As-Is, Claims
- From Opportunity: Strategy, Market, Technology

Key to Success
“Agree Goals with Stakeholders before FR/NFR”

FR/NFR
Real-Time Inventory Mgmt
Forecasting Customer Needs

Goal
“Stock products…”

Meet
Realize
System


RE Core Techniques
Req. Elicitation: Goal Analysis(2)

Soft Goal[Strategic, Abstract]
- Qualitative Characteristics of a System Should Have

Hard Goal[Tactical, Concrete]
- Concrete/Quantitative Characteristics of a System Should Have
- Classification of hard Goals
  - Achieve: Goal Requires to Transit from As-Is to To-Be
  - Retain/Mitigate: To Retain As-Is(Good)/Mitigate to Transit to Bad State

Task: Means to Meet Goals
- Candidates for FR/NFR

Soft Goal

Hard Goal

FR/NFR

Task
RE Core Techniques
Req. Elicitation: Goal Analysis(3)

Goal Graph
- DAG (Directed Acyclic Graph) Structuring Goals and their Relationships

Relationships
- AND-Refinement
  - Goal G is Met if All of Sub-Goals are Met
  - \( G = G_1 \cap G_2 \cap \cdots \cap G_n \)
  - Completeness, Consistency, Minimality
- OR-Refinement
  - Goal G is met if One of Sub-Goals is Met
  - \( G = G_1 \cup G_2 \cup \cdots \cup G_n \)

RE Core Techniques
Req. Elicitation: Goal Analysis(4)

Contribution: Weight Associated to an Arc [KAOS]
- Positive (Contribute): +, ++
- Negative (Conflict): −, − −

Conflict
- Two Goals are Unable to be Met Simultaneously
- Cases
  - Goals from Multiple Sources
  - Goals from Multiple Viewpoints

RE Core Techniques
Req. Elicitation: Goal Analysis

- **GORE (Goal-Oriented RE)**
  - KAOS (Knowledge Acquisition in automated Specification) [A. van Lamsweerde, et al., 1991]
  - Formal Model of Goal in Tree-Structure
  - Model of NFR
  - i*(eye star)/URN (User Requirements Notation) [E. Yu, 2011]
  - Network Relationship Model of Goals and Actors: SD (Strategic Dependency), SR (Strategic Rationale)


---

RE Core Techniques
Req. Elicitation: Goal Analysis

- **Hard Goal: Hierarchical Decomposition with AND/OR**

**Analysis of Hard Goals**

(1) Select a Goal of Terminal Node
  - "By System", "From All", "Automatically"

(2) Infer along with Goal Graph toward the Root
  - To See Whether the Root Goal is Met or Not

[Diagram of scheduling meeting, collecting constraints, generating schedules, etc.]

RE Core Techniques
Req. Elicitation: Goal Analysis(7)

Analysis of Soft Goals
- Evaluate the Contribution of Sub-Goals by Weight of the Arcs
- Maximize Positive Contribution While Minimizing Negative ones

Quality of Schedule
- Minimal Effort
- Matching Effort
- Minimal Conflicts
- Degree of Participation

Collection effort
- Scheduling Meeting
- Collect Timetables
- Collect Constraints
- Collect Other Constraints

Matching Effort
- By Person
- From All
- Automatically

Minimal Efforts
- By Sharing
- By System
- By Person
- From Initiator Only

Degree of Participation
- By E-mail
- By All Means
- Interactively

Soft Goal
- Maximize Positive Contribution
- Minimize Negative ones

Analysis of Soft Goals
- Evaluate the Contribution of Sub-Goals by Weight of the Arcs
- Maximize Positive Contribution While Minimizing Negative ones

Real-Time Order Mgmt
Order on Demand
Demand Forecasting
Real-Time Inventory Mgmt at Store
Real-Time Inventory Mgmt
Collect Info. on Customer Behavior
Send Arrival Notice
Customer Profile & Purchase Info
Input Customer Profile
Input Purchase Info via POS

Legend
- Soft Goal
- Hard Goal
- Task


All Rights Reserved, Copyright Mikio Aoyama, 2011
4. RE Core Techniques

- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Verification, Validation and Evaluation

4.1 Classification: Classify Req. Items Based on Properties
4.2 Structuring: Find the Relationship among Req. Items
4.3 Allocation: Allocate Req. to Architecture and Analyze Feasibility of Req.
4.4 Prioritization: Set Priority to the Req.
4.5 Negotiation: Agree on Req. to be Realized with Stakeholders

RE Core Techniques
Req. Analysis: Analysis Process

4.1 Classification: Classify Req. Items Based on Properties
4.2 Structuring: Find the Relationship among Req. Items
4.3 Allocation: Allocate Req. to Architecture and Analyze Feasibility of Req.
4.4 Prioritization: Set Priority to the Req.
4.5 Negotiation: Agree on Req. to be Realized with Stakeholders
**RE Core Techniques**

**Req. Analysis: Structuring Requirements**

**Structuring Requirements**
- Find the Relationships among Req. Items
- Represents the Whole Structure of Req. with Diagram/Table

**Techniques for Structuring Req. w.r.t. 5W1H**
- **Why**: Goal, Rule, Policy
- **Who**: Organization Model, User Model, Role(Responsibility)
  - Ex.: RACI Matrix by Classifying Req. with Responsibility, Accountable, Consulted, and Informed
- **What**: Structure Model, Function Model, Dictionary(Ontology)
- **When**: Event Model, State Transition Model
- **Where**: Deployment Model
- **How**: Flow Model, Scenario

**View Point and View**
- **Viewpoint**: A Concern to a System
- **View**: Sight from a Viewpoint, Information Filtered through a Specific Concern

**Modeling by Multiple Viewpoints**
- Separating Complex Information on a System into a Manageable Set of Information through Multiple Viewpoints
- Understanding Correct
- Ex: 3D Physical Object
  - Represented with 2D Drawings from 3 Different Viewpoints
RE Core Techniques

Req. Analysis: Structuring Requirements

- Selecting Viewpoints
  - Viewpoint = A Set of Concerns to a System
  - Modeling a System with a Set of (Small Number of) Viewpoints

- Concern
  - A Set of Properties of a System Meaningful to Stakeholder

- Separation of Concerns
  - Concerns can be Diverse, and Need to be Separated into a Set of Appropriate View(point)s
  - MDSoC (Multi-Dimensional Separation of Concerns)

Stakeholder Concerns View(point)s

- Safety
- Cost
- Usability


Analyzing a System with 3 Views by Structuring Different Properties

- Functional: Capability of a System
- Structure [Static]: Components and their Relationships
- Behavior [Dynamic]: Temporal Properties

Functional View (Data Flow Diagram)

Customer → Card Entry

1.0 Authentication

5.0 Error Handling

Structural View (Class Diagram)

Terminal System

Teller → ATM

Behavioral View (State-Transition Diagram)

Card Entry → Authentication

Ref.: All Rights Reserved, Copyright Mikio Aoyama, 2011
RE Core Techniques
Req. Analysis: Structuring Requirements

- **Functional View**
  - Function = Process
  - Model: IPO (Input-Process-Output)
  - Analysis: Data Flow Analysis

- **Structural View**
  - Entity = Abstract Data Type
  - Model: ER (Entity-Relationship)/Class
  - Analysis: Conceptual Data Modeling

- **Behavioral View**
  - State and Transition
  - Model: STD (State Transition Diagram)/State Chart
  - Analysis: State Analysis

---

RE Core Techniques
Req. Analysis: Requirements Allocation

- **Twin Peaks Model**
  - Concurrent Activity of Req. Analysis and Architecture Analysis
  - Req. and Architecture are Mutually Dependent
  - Hierarchy of 3 Scopes and Corresponding Architectural Issues

RE Core Techniques

Req. Analysis: Requirements Prioritization

- Qualitative/Quantitative Value Analysis
  - Prioritization Matrix
  - Four-Quadrant
  - Multi-Objective optimization
  - Voting
  - MoSCoW

- Prioritization Scale
  - Importance and Urgency

<table>
<thead>
<tr>
<th>Req.</th>
<th>Relative Benefit</th>
<th>Relative Penalty</th>
<th>Total Value</th>
<th>Relative Cost</th>
<th>Relative Risk</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>A001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An Example of Prioritization Matrix


4.

RE Core Techniques

- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Verification, Validation and Evaluation
5 RE Core Techniques

Req. Specification: Specifications from Multiple Views

Representation with Multiple Views/Viewpoints

Structural View
- Conceptual Data Modeling
  - E-R (Entity-Relationship) Diagram
  - Class Diagram

Functional View
- SA/SD
  - DFD (Data Flow Diagram)
- Use Case Diagram
- Activity Diagram

UML
- Sequence Diagram
- Communication Diagram
- State Chart

Behavioral View
- Real-Time
  - SA/SD
  - STD (State Transition Diagram)


RE Core Techniques

Req. Specification: Specification Process

Specification: Write Req. with a Set of Specified Forms
- Text: Specific Style of Natural Language
- Model: Diagram and Table, Prototype

Standard Specification Templates
RE Core Techniques

Req. Specification: Business Req. Definition

**Business Requirements Definition**

- Documenting Business-To-Be and Associated Information of Capability, Rules, etc.
- IEEE Std. 1362-1998

**Product Requirements**

- Definition of Products Including Embedded Systems and Package
- No Common Standard Yet be Defined, Domain Specific Documentation Styles


---

RE Core Techniques


**System Requirements Specification (SyRS)**

- Documenting System-To-Be
- Template for SyRS: IEEE Std.1233-1998
- May Need to Taylor

RE Core Techniques

Software Requirements Specification (SRS)

- Documenting Software-To-Be
- Template for SRS: IEEE Std. 830-1998
- Suggested to Some Improvement Due to Change of Related Std.
- Attributes Can Use the Classification of ISO/IEC 25030:2007

Structure
1. Introduction
2. Overall Description
   2.1 Product Perspective
   2.2 Product Functions
   2.3 User Characteristics
   2.4 Constraints
   2.5 Assumptions and Dependencies
3. Specific Requirements
   3.1 External Interface Requirements
   3.2 Functional Requirements
   3.3 Performance Requirements
   3.4 Design Constraints
   3.5 Software System Attributes
      a) Reliability, b) Availability, c) Security,
      d) Maintainability, e) Portability
   3.6 Other Requirements


4.

RE Core Techniques

- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Verification, Validation and Evaluation
RE Core Techniques

Req. V & V & Evaluation: Definition

Concept for V&V in RE. Has Been Evolved
- Was NOT Consistent With V&V in SE,
  - SWEBOK: Only “Validation” is defined, But Suggests Some Verification Activities within
- It Became Consistent with V&V in SE

Definition of ISO/IEC/IEEE 29148 Requirements Engineering
- Verification: confirmation by examination that requirements is well formulated
- Validation: confirmation by examination that the requirements define the right system as intended by the stakeholders

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification</td>
<td>Yes</td>
<td>Yes</td>
<td>Partly</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Evaluation</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>


RE Core Techniques

Req. V & V & Evaluation: V&V&E Process

Verification
- Confirm that Req. Spec. is Well-Formed w.r.t. a Set of Specification Properties to be Met

Validation
- Confirm that Req. Spec. Fulfill Stakeholder Req.

Evaluation [Need Further Study]
- Evaluate the Value and Risk of Req. Spec. w.r.t. a Set of Criteria
**RE Core Techniques**

**Req. V & V & Evaluation: Verification**

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unambiguousness</td>
<td>Every Requirement Stated Has Only One Interpretation</td>
</tr>
<tr>
<td>Completeness</td>
<td>No Missing Definition/Meaning of Requirement Stated</td>
</tr>
<tr>
<td>Consistency</td>
<td>No Subset of Individual Req. Described in it Conflict</td>
</tr>
<tr>
<td>Verifiable</td>
<td>Every Requirement Stated is Verifiable, i.e. Realistically Checkable</td>
</tr>
<tr>
<td></td>
<td>Counter Example: “Like Current System”, “Work Well”</td>
</tr>
<tr>
<td>Modifiable</td>
<td>The Structure and Style are Such That Any Changes to the</td>
</tr>
<tr>
<td></td>
<td>Requirements Can be Made Easily</td>
</tr>
<tr>
<td>Traceable</td>
<td>The Origin of Each of its Requirements is Clear and the</td>
</tr>
<tr>
<td></td>
<td>Specification Facilitates the Referencing of each Requirement</td>
</tr>
<tr>
<td></td>
<td>in Future Development or Enhancement Documentation</td>
</tr>
</tbody>
</table>


**RE Core Techniques**

**Req. V & V & Evaluation: Validation & Evaluation**

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctness (External Consistency)</td>
<td>Review against Superior Req. Spec. and Stakeholder</td>
</tr>
<tr>
<td>Feasibility</td>
<td>Req. is Technically Achievable and Fits within Constraints</td>
</tr>
<tr>
<td>Degree of Importance (Priority)</td>
<td>Degree of Stakeholders’ Desire, or</td>
</tr>
<tr>
<td></td>
<td>Degree of Essentialness to the System</td>
</tr>
<tr>
<td>Degree of Stability</td>
<td>Number of the Expected Changes to Any Requirement</td>
</tr>
</tbody>
</table>

*Not Stated in IEEE Std. 830, ISO/IEC 29148*
Requirements Review

- Review is Most Widely Used for V&V&E of Req. Spec.
- Need to Appropriate Selection of Members, and Collaboration with Stakeholders

Review Technique


Major Activities in the Review

- Clarification of Req.
- Check against Properties for V&V&E
- Resolve Any Conflict, and Any Infeasible Req.
- Agree with the Participants on the Req. Spec.
- Report the Agreement, Issues and Further Actions if Needed

Prototyping

- Useful for Validating Dynamic and/or People-Involved Req. Spec.

Classification of Prototyping

- Horizontal[Static, Mockup] Prototyping: Demonstrate Req. Spec. with Paper or Presentation Tools
- Vertical[Dynamic] Prototyping: Demonstrate the Computing Behavior by Developing Software of Limited Capabilities

<table>
<thead>
<tr>
<th>Type of Prototype</th>
<th>Evolutionary Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td>Implementation</td>
</tr>
<tr>
<td>Horizontal</td>
<td>Paper</td>
</tr>
<tr>
<td></td>
<td>Software</td>
</tr>
<tr>
<td>Vertical</td>
<td>Paper</td>
</tr>
<tr>
<td></td>
<td>Software</td>
</tr>
</tbody>
</table>
6. RE Practice for Success

RE Practice for Success

REBOK Can Help You

Apply RE Techniques to Improve RE Practice
- REBOK Can Help You

Analyze RE-As-Is (Current Practice of RE)
- Evaluate Current RE Practice of RE Process, Techniques and Products

Engineering the RE-To-Be (Intended Practice of RE)
- Reengineering RE-As-IS and Identifies Requirements to Move to RE-To-Be
- Education and Training of RA (Req. Analyst)

Knowledge

Reference

RE-As-Is (Current Practice)

RE-To-Be (Intended Practice)

RA-To-Be

RA-As-Is

Education & Training

Req. for Reengineering
RE Practice for Success
Practical Considerations (Tips)

*Best Practices for RE Practice*

(1) A Good Practice Guide*
- by RE-GPG (Requirements Engineering for Critical Systems)
- 3 Level of Guides for basic, Intermediate, and Advanced

(2) RE Best Practice Patterns
- JISA RE WG (Chair: Mikio Aoyama)
- 34 Best Practice Patterns Elaborated from Practice in Japan


REBOK for Success
REBOK Roadmap

*Development of REBOK*
- English Version of REBOK V 1.0
- Development of Education and Training Curriculum

*Application of REBOK*
- Case Study on RE Practice with respect to REBOK
- Feedback to REEBOK from Practice

*Collaboration with Global RE Community*

We Welcome Your Feedback!
Remarks

- **Message to Practitioners**
  - Improve the Entry Process of Development with RE
  - Make RE Work in the Context of Development

- **Message to Educators**
  - Teach RE with REBOK

- **Message to Researchers**
  - Jump into the Big Sea of Research on RE

Twin Angels is a Symbol of REBOK

---

Chapters of REBOK

- 0. Introduction: Overview of REBOK
- 1. Requirements Engineering Fundamentals
- 2. Requirements Engineering Process
- 3. Requirements Elicitation
- 4. Requirements Analysis
- 5. Requirements Specification
- 6. Requirements Verification, Validation and Evaluation
- 7. Requirements Planning and Management
- 8. Practical Consideration
- Glossary, Related Standards and References
References for Further Understanding


Thank You for Your Attentions!

Team REBOK Development
M. Aoyama (Chair: Nanzan University),
S. Saito (NTT Data), T. Nakatani (University of Tsukuba),
M. Suzuki (ASTER), T. Obeyama (Nomura Research Institute),
K. Fujita (Hitachi Solutions), I. Morita (Fujitsu),
Hiroaki Nakazaki (Fujitsu FIP), Y. Saito (RICHO IT Solutions),
Ritsuo Suzuki (Secretary: JISA)

Team REBOK Application
M. Aoyama (Chair: Nanzan University),
M. Sato (ITOCHU-Techno Solutions), T Oketani, A. Kobayashi (Intec),
T. Koike (NEC Nexsolutions), M. Inoki (Toshiba Solutions),
T. Hirayama, A. Tamaki (JUAS), T. Hiraishi (Panasonic),
Y. Ohshita (Hitachi Solutions), Y. Sanagi (Mitsubishi Research Institute DCS)
K. Arimoto (RICHO IT Solutions), Ritsuo Suzuki (Secretary: JISA)