APSEC 2011 Tutorial

RE BOK Requirements Engineering Body Of Knowledge

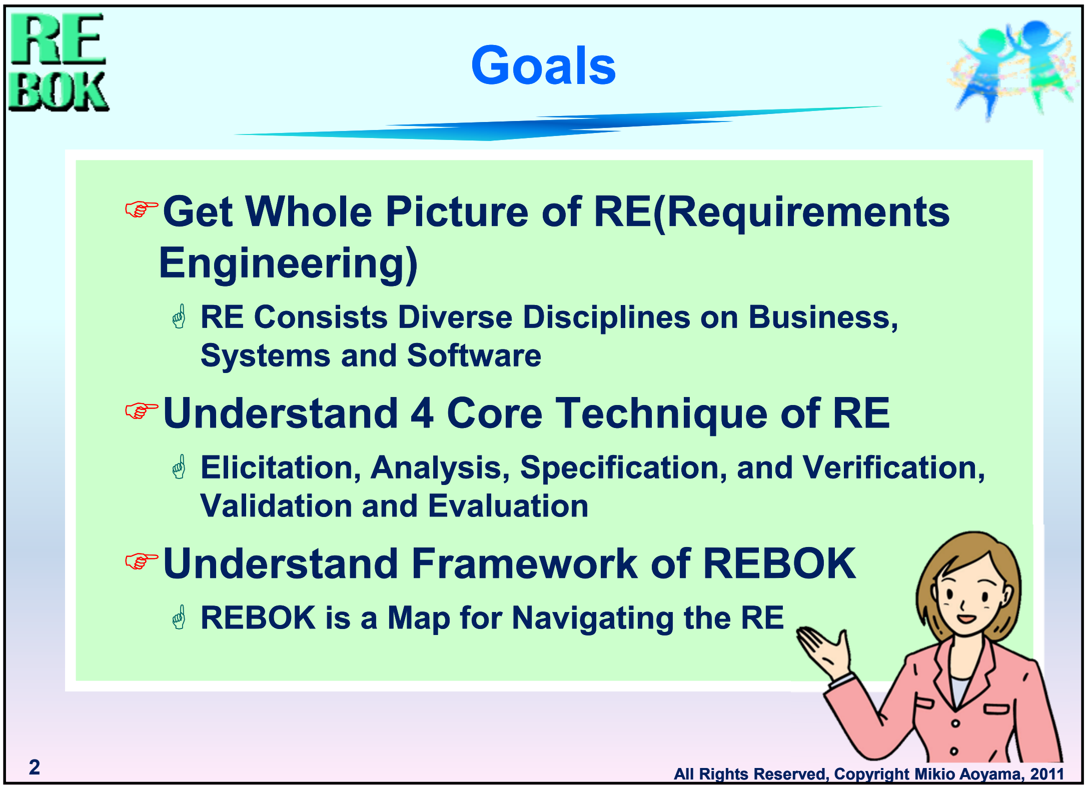
Requirements Engineering Based on REBOK (Requirements Engineering Body Of Knowledge)

Mikio Aoyama
Dep. of Software Engineering, Nanzan University
mikio.aoyama@nifty.com
http://www.nise.org/

We are NISE: Network Information and Software Engineering

December 5, 2011

Ho Chi Minh City, Vietnam

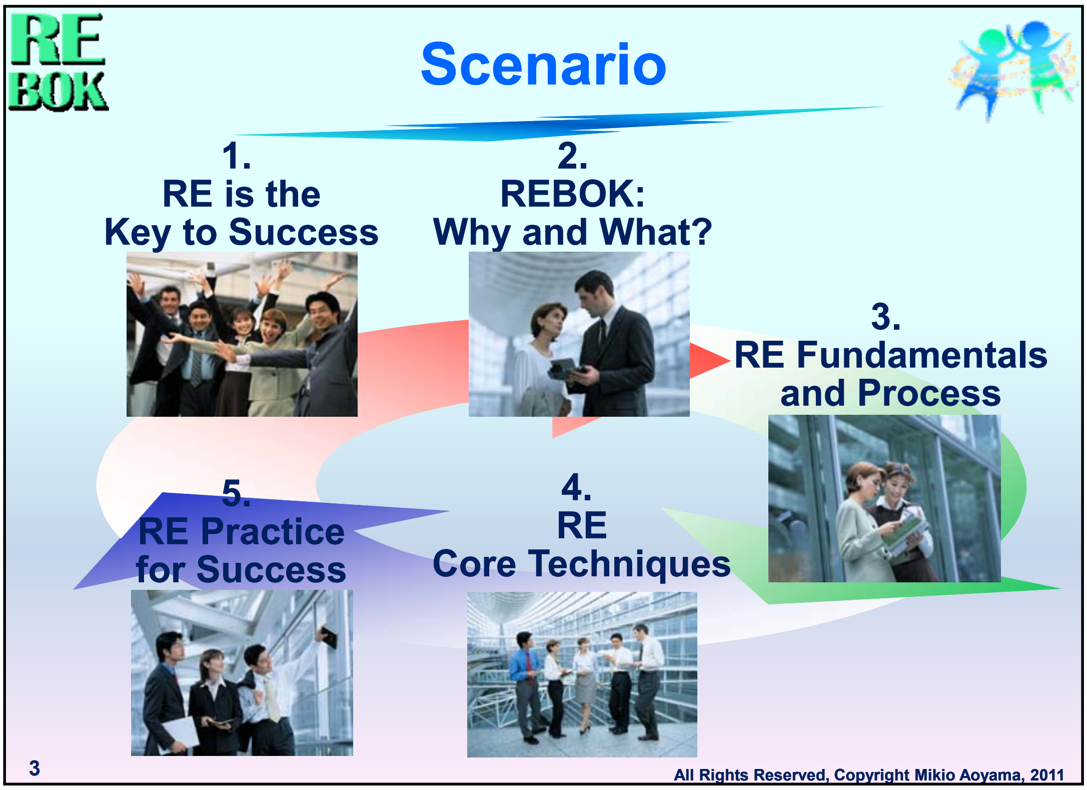


RE BOK

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

2



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

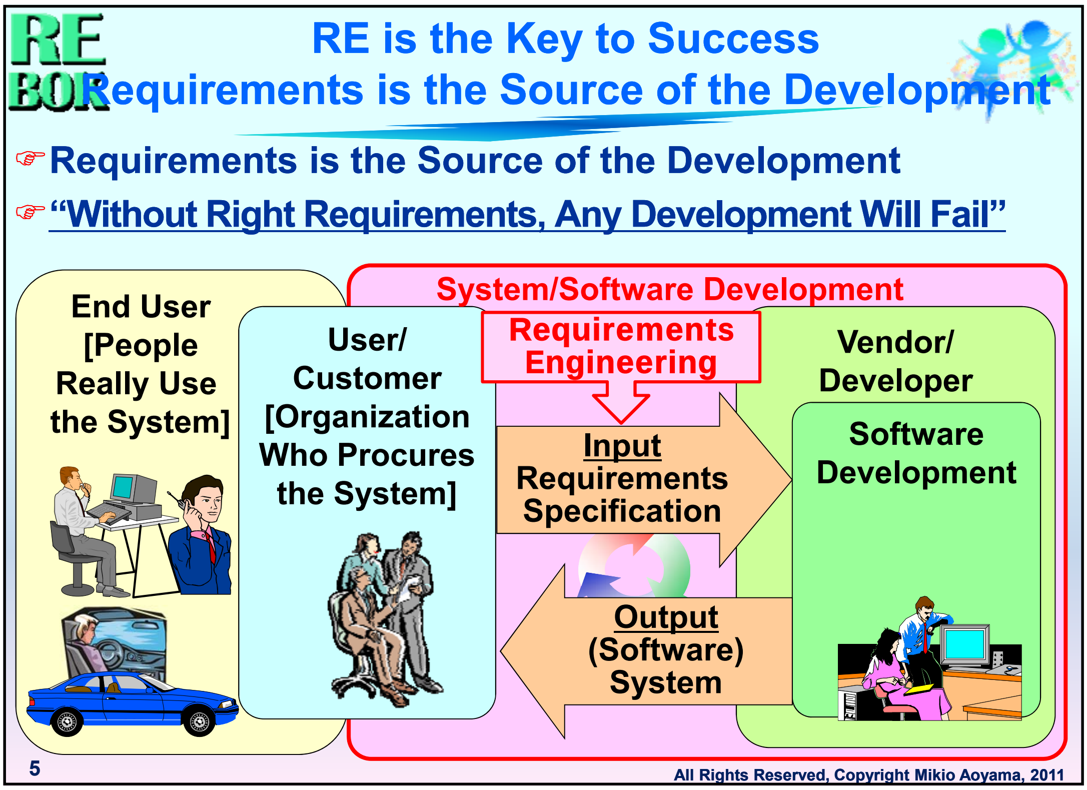
3

All Rights Reserved, Copyright Mikio Aoyama, 2011

1. RE is the Key to Success

4

All Rights Reserved, Copyright Mikio Aoyama, 2011



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

End User [People Really Use the System]

User/ Customer [Organization Who Procures the System]

Requirements Engineering

Input Requirements Specification

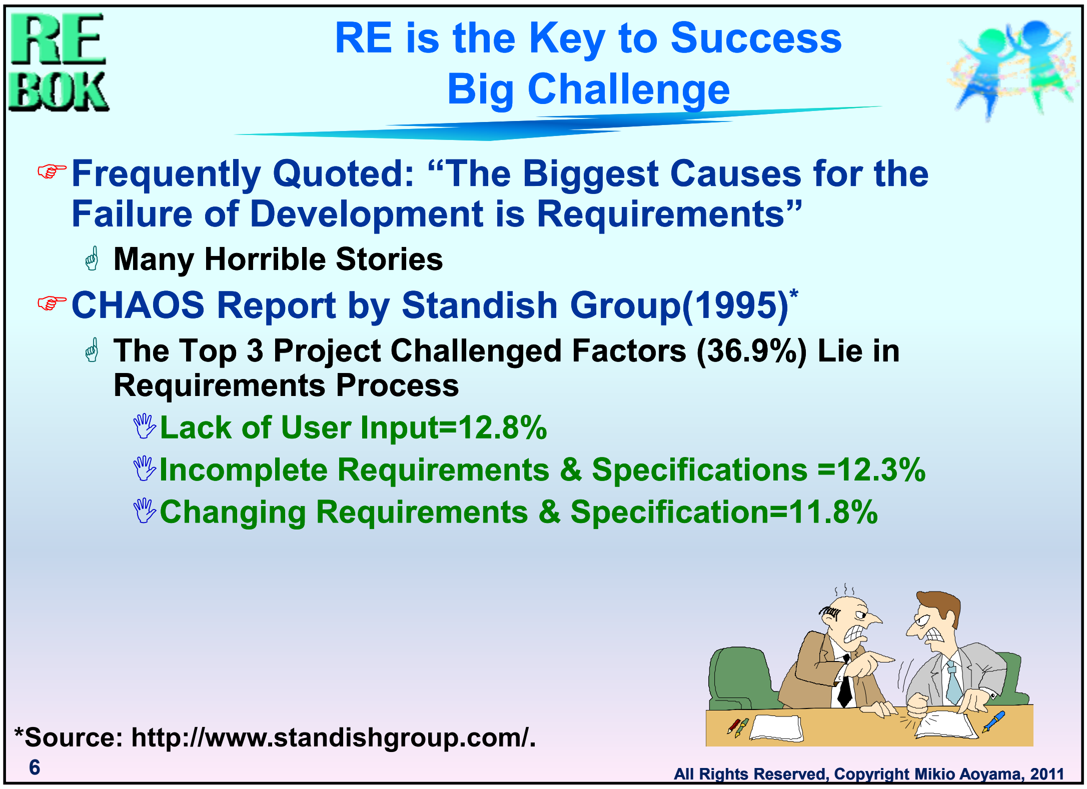
Output (Software) System

Vendor/ Developer

Software Development

5

All Rights Reserved, Copyright Mikio Aoyama, 2011



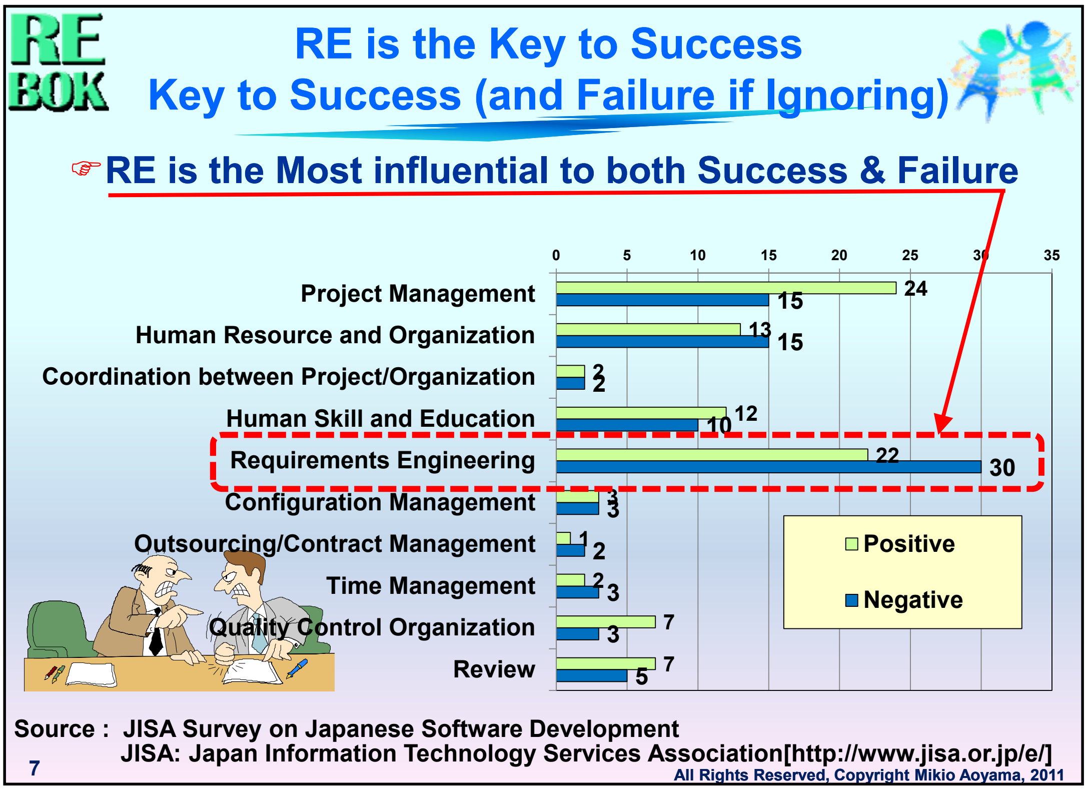
RE is the Key to Success
Big Challenge

- 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%

*Source: http://www.standishgroup.com/.

6

All Rights Reserved, Copyright Mikio Aoyama, 2011



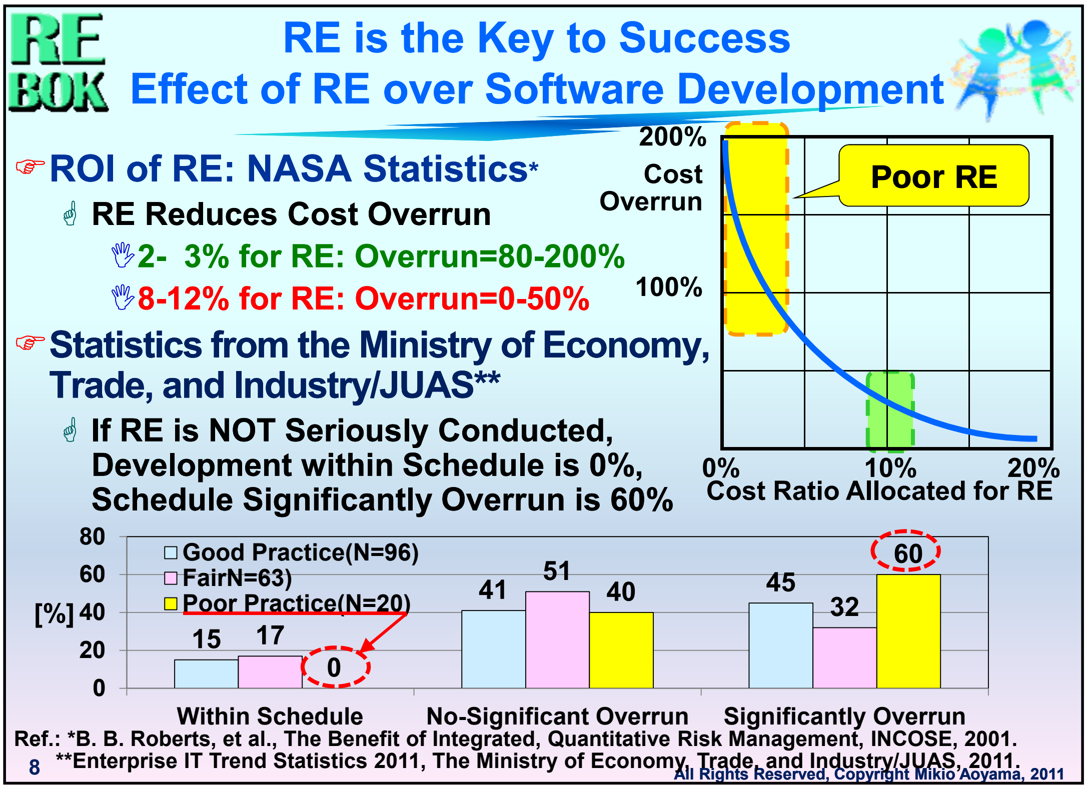
RE is the Key to Success
Key to Success (and Failure if Ignoring)

☞RE is the Most influential to both Success & Failure

	Positive	Negative
Project Management	24	15
Human Resource and Organization	13	15
Coordination between Project/Organization	2	2
Human Skill and Education	12	10
Requirements Engineering	22	30
Configuration Management	3	3
Outsourcing/Contract Management	1	2
Time Management	2	3
Quality Control Organization	7	3
Review	7	5

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

7



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%

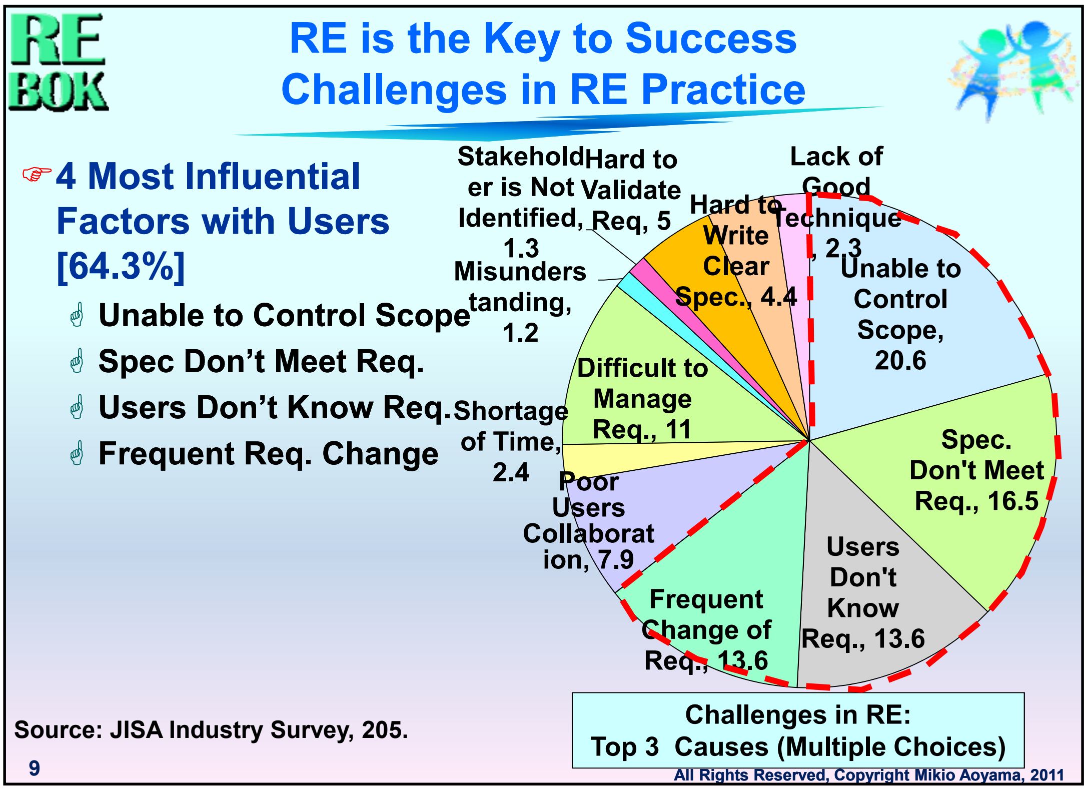
☞Statistics from the Ministry of Economy, Trade, and Industry/JUAS**

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

[%]	Good Practice(N=96)	FairN=63)	Poor Practice(N=20)
Within Schedule	15	17	0
No-Significant Overrun	41	51	40
Significantly Overrun	45	32	60

Ref.: *B. B. Roberts, et al., The Benefit of Integrated, Quantitative Risk Management, INCOSE, 2001.
**Enterprise IT Trend Statistics 2011, The Ministry of Economy, Trade, and Industry/JUAS, 2011.

8



RE is the Key to Success
Challenges in RE Practice

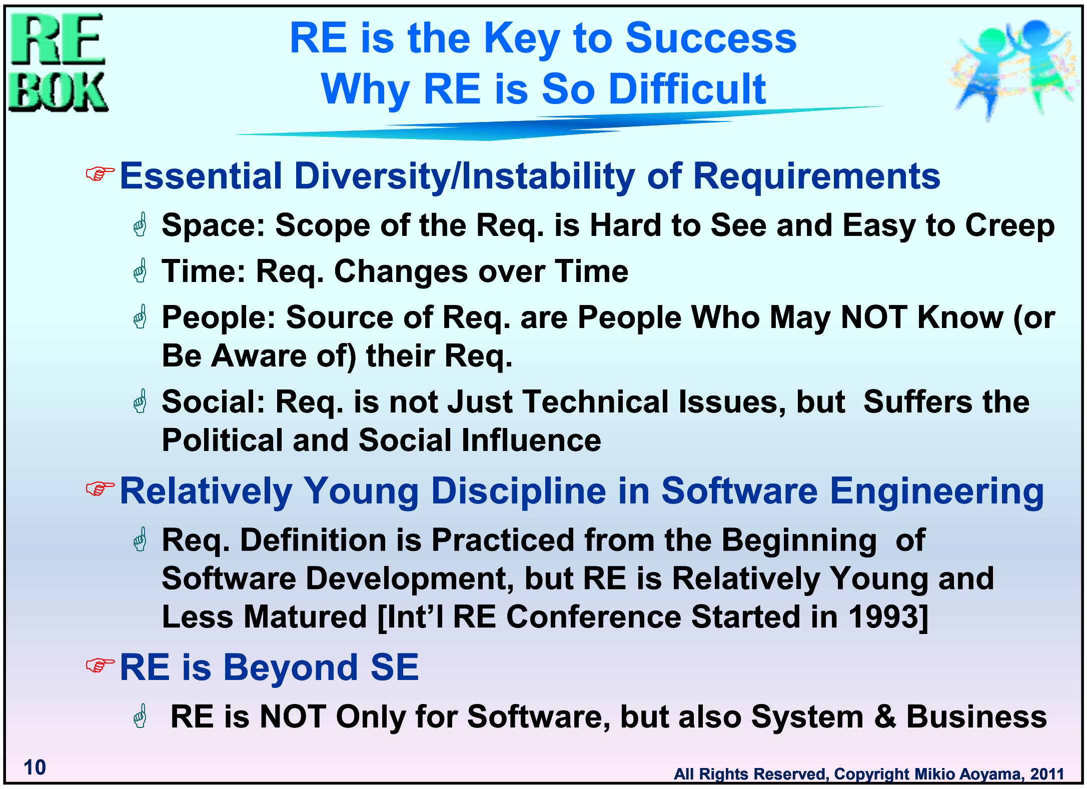
- 4 Most Influential Factors with Users [64.3%]
  - Unable to Control Scope
  - Spec Don't Meet Req.
  - Users Don't Know Req.
  - Frequent Req. Change

Challenges in RE:
Top 3 Causes (Multiple Choices)

Source: JISA Industry Survey, 205.

9

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE is the Key to Success
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

10

All Rights Reserved, Copyright Mikio Aoyama, 2011

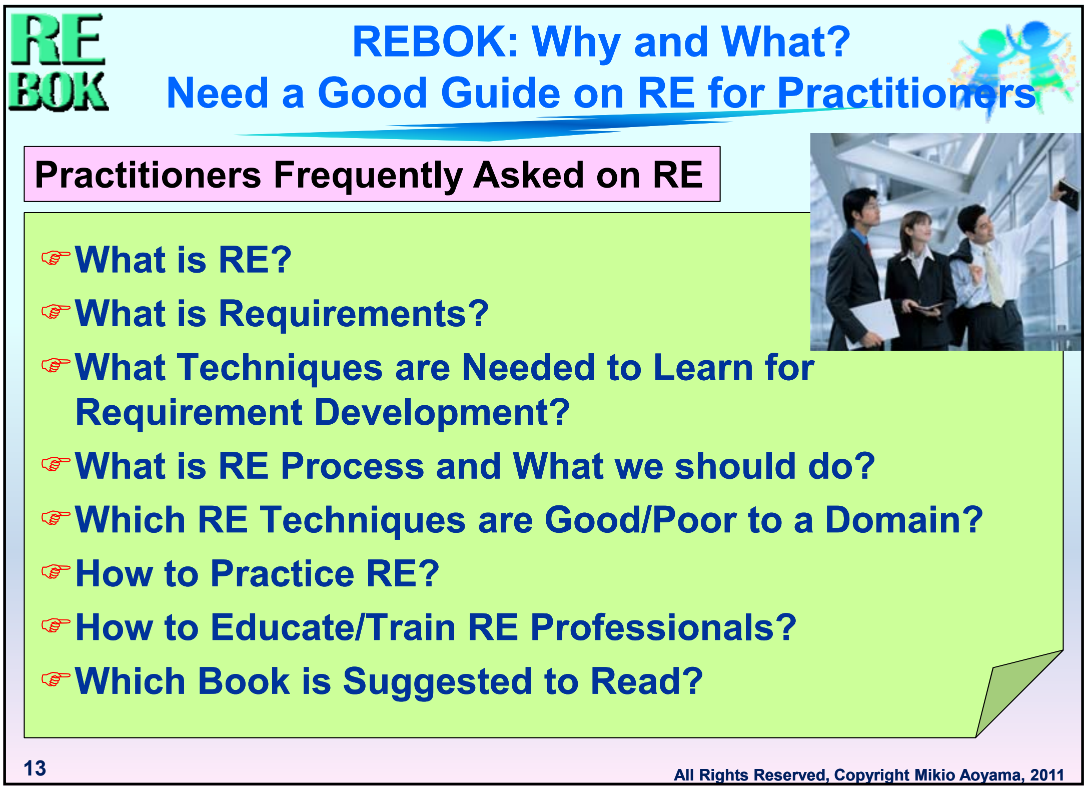
RE is the Key to Success
Now, RE is "Maturing"

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

11

2.
REBOK: Why and What?

12



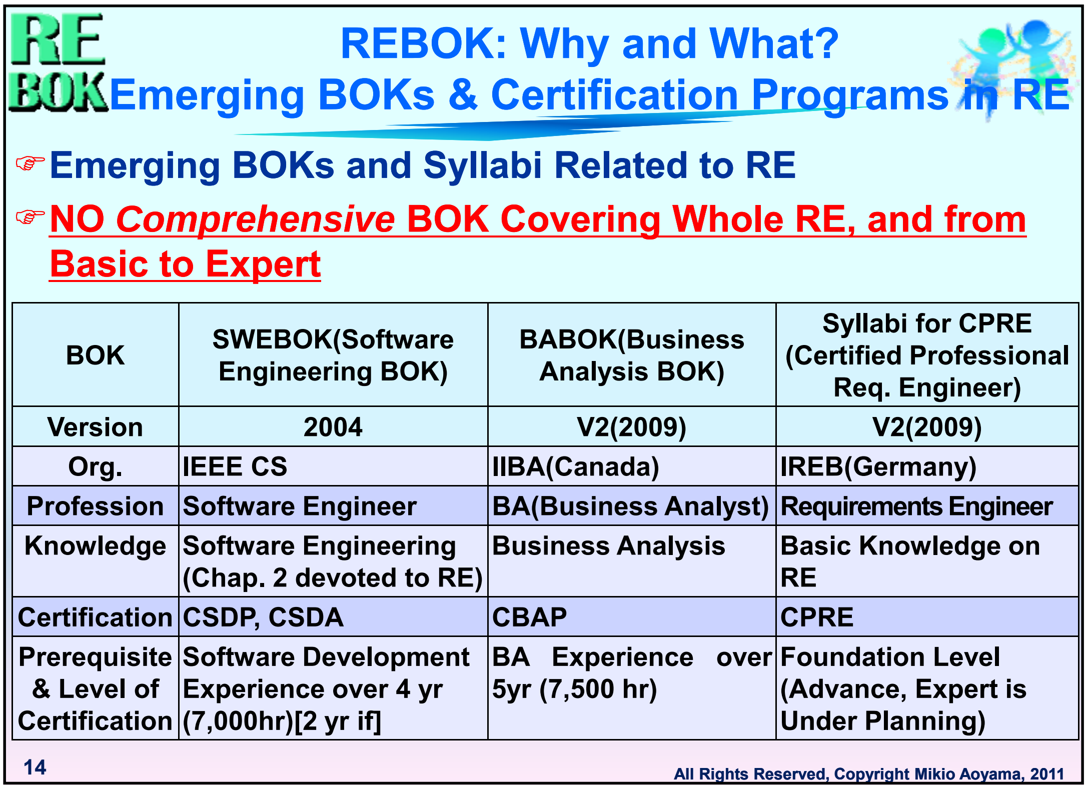
REBOK: Why and What?
Need a Good Guide on RE for Practitioners

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?

13

All Rights Reserved, Copyright Mikio Aoyama, 2011



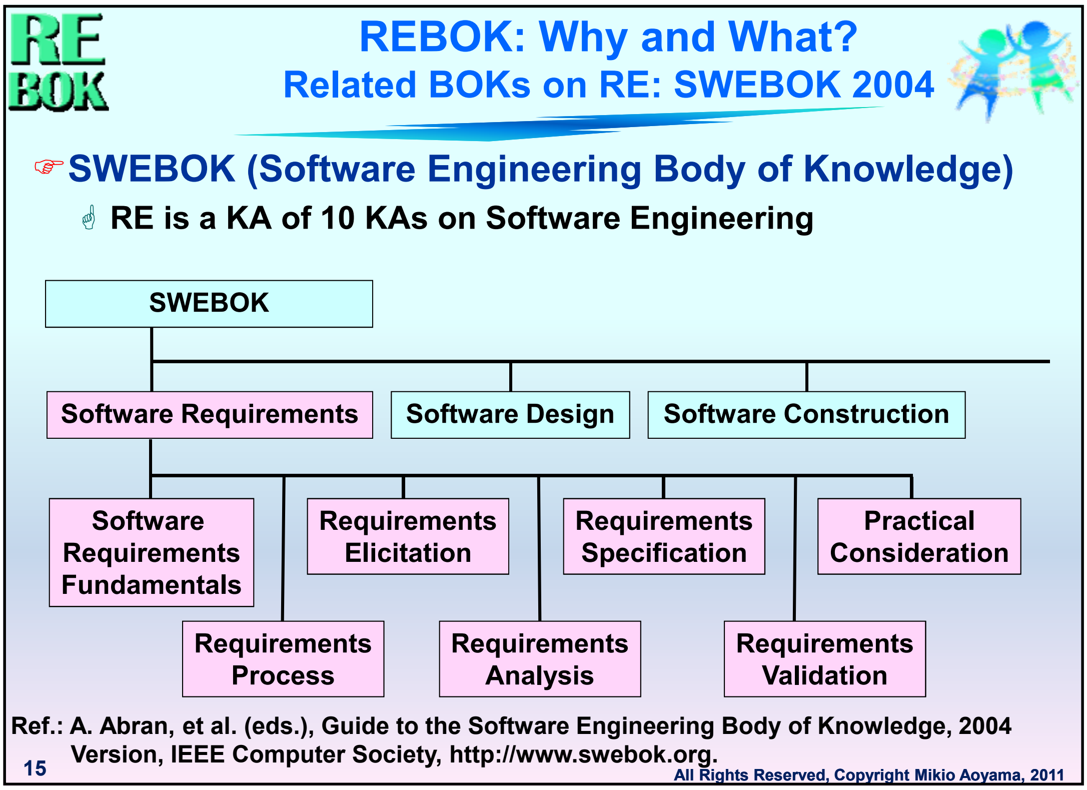
REBOK: Why and What?
Emerging BOKs & Certification Programs in RE

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

BOK	SWEBOK(Software Engineering BOK)	BABOK(Business Analysis BOK)	Syllabi for CPRE (Certified Professional Req. Engineer)
Version	2004	V2(2009)	V2(2009)
Org.	IEEE CS	IIBA(Canada)	IREB(Germany)
Profession	Software Engineer	BA(Business Analyst)	Requirements Engineer
Knowledge	Software Engineering (Chap. 2 devoted to RE)	Business Analysis	Basic Knowledge on RE
Certification	CSDP, CSDA	CBAP	CPRE
Prerequisite & Level of Certification	Software Development Experience over 4 yr (7,000hr)[2 yr if]	BA Experience over 5yr (7,500 hr)	Foundation Level (Advance, Expert is Under Planning)

14

All Rights Reserved, Copyright Mikio Aoyama, 2011



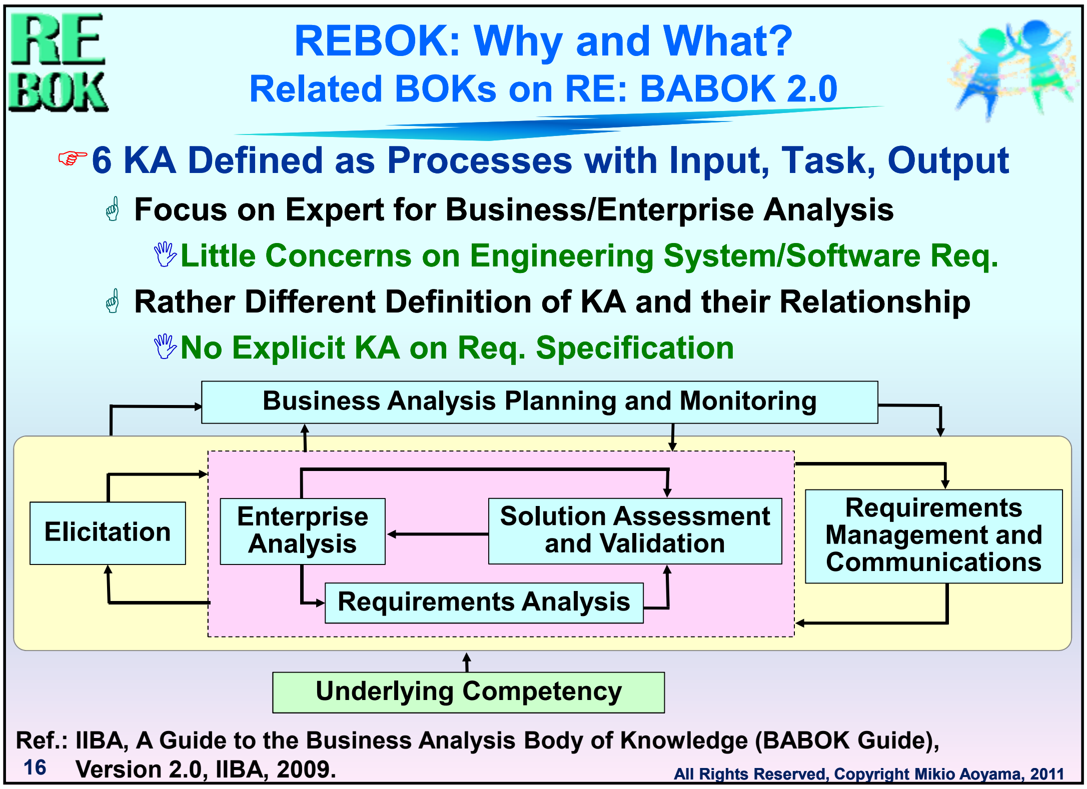
REBOK: Why and What?
Related BOKs on RE: SWEBOK 2004

☞ SWEBOK (Software Engineering Body of Knowledge)

- RE is a KA of 10 KAs on Software Engineering

Ref.: A. Abran, et al. (eds.), Guide to the Software Engineering Body of Knowledge, 2004 Version, IEEE Computer Society, http://www.swebok.org.

15



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

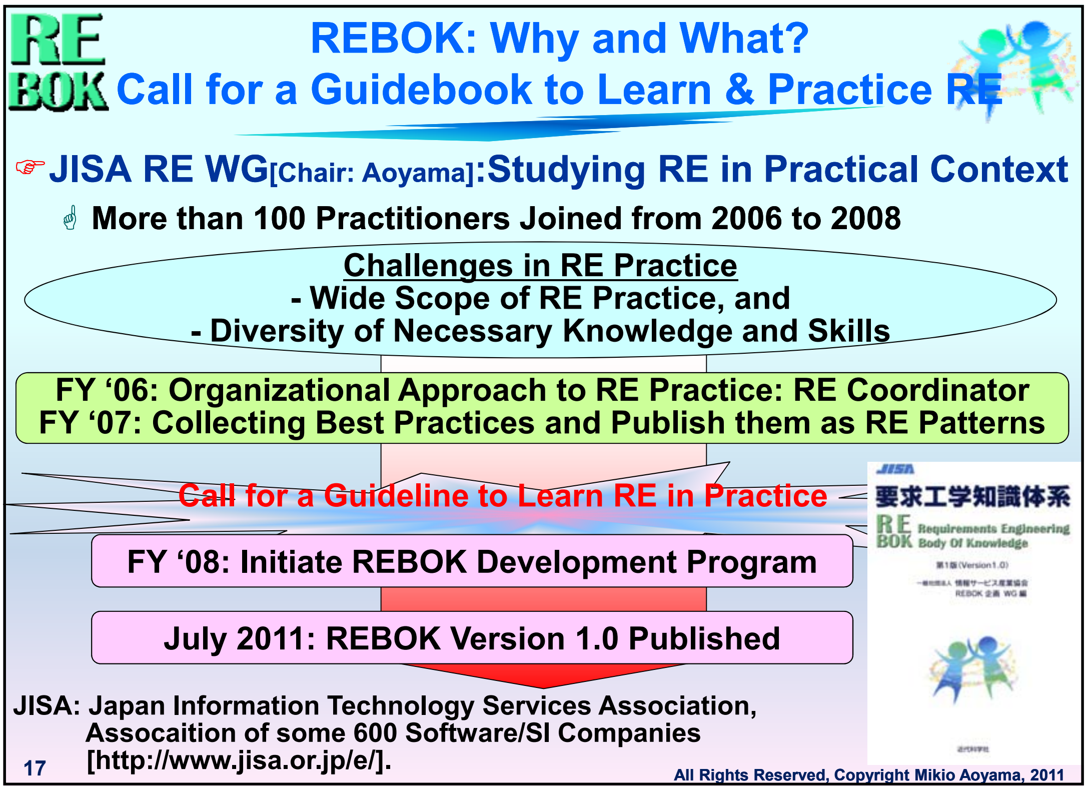
☞ 6 KA Defined as Processes with Input, Task, Output

- 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

Business Analysis Planning and Monitoring
Elicitation
Enterprise Analysis
Solution Assessment and Validation
Requirements Analysis
Requirements Management and Communications
Underlying Competency

Ref.: IIBA, A Guide to the Business Analysis Body of Knowledge (BABOK Guide), Version 2.0, IIBA, 2009.

16



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

Call for a Guideline to Learn RE in Practice

FY '08: Initiate REBOK Development Program

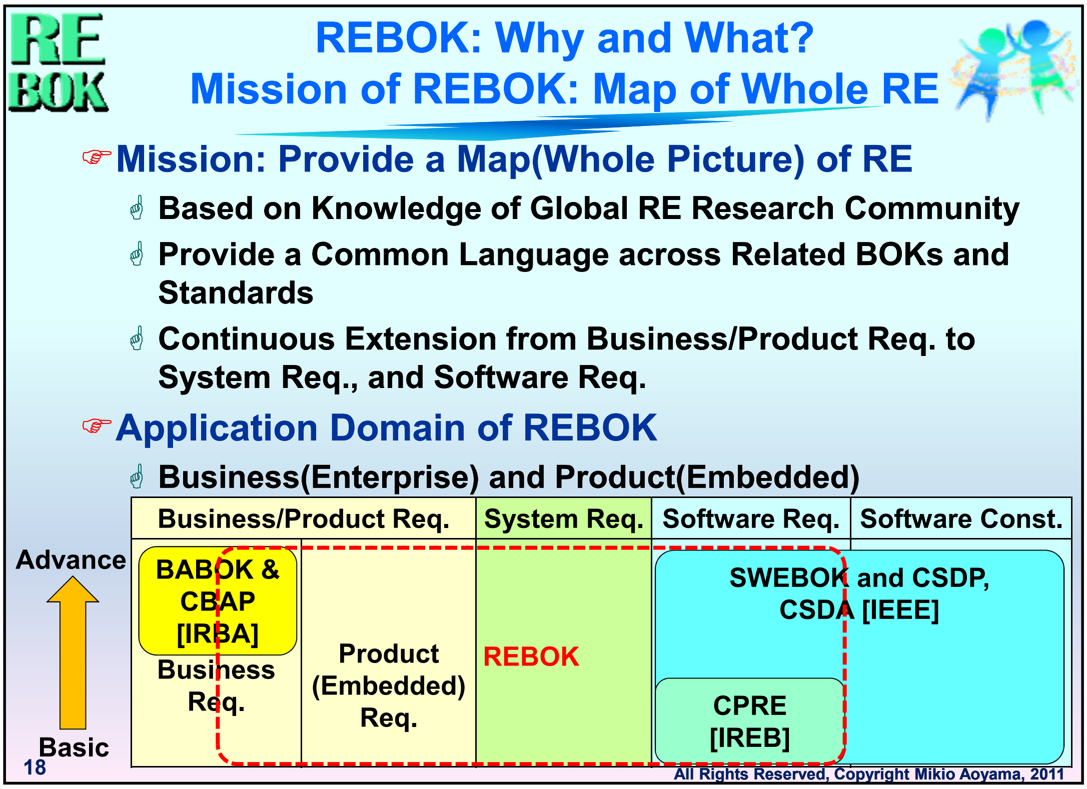
July 2011: REBOK Version 1.0 Published

JISA: Japan Information Technology Services Association,
Assocaition of some 600 Software/SI Companies
[http://www.jisa.or.jp/e/].

JISA
要求工学知識体系
RE BOK Requirements Engineering Body Of Knowledge
第1版(Version1.0)
一般社団法人 情報サービス産業協会 REBOK企画 WG 編
近代科学社

17

All Rights Reserved, Copyright Mikio Aoyama, 2011



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.

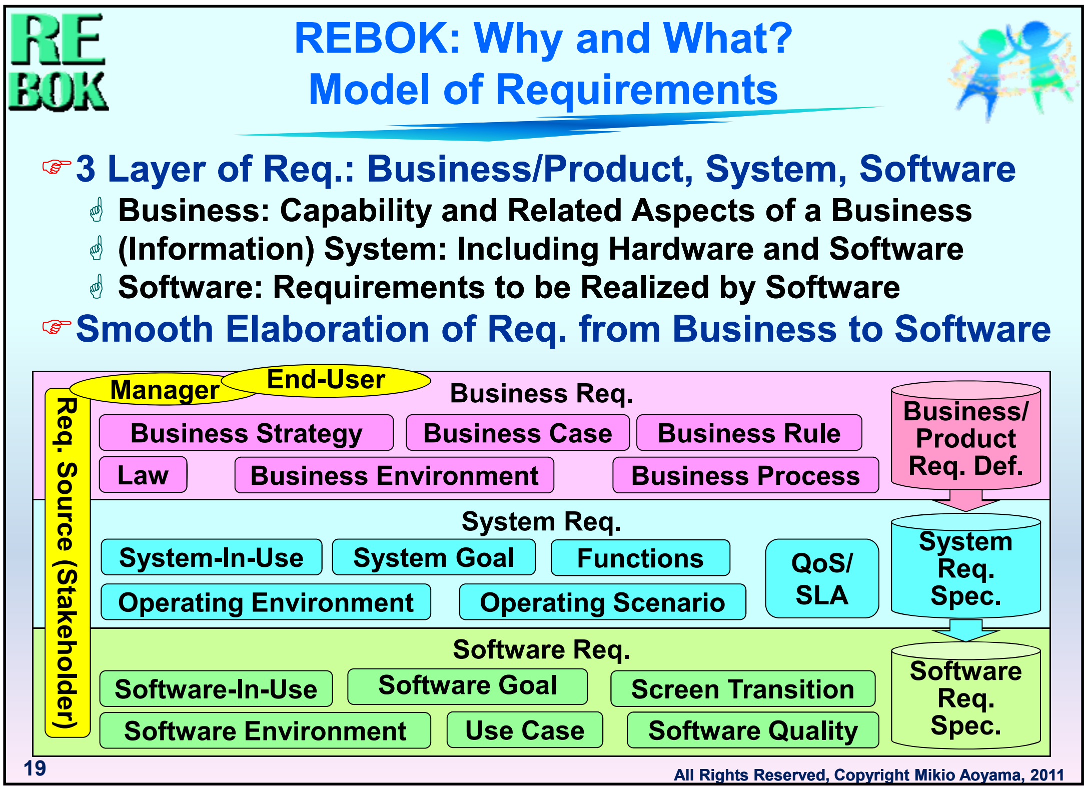
☞ Application Domain of REBOK

- Business(Enterprise) and Product(Embedded)

Business/Product Req.		System Req.	Software Req.	Software Const.
BABOK & CBAP [IRBA] Business Req.	Product (Embedded) Req.	REBOK	SWEBOK and CSDP, CSDA [IEEE] CPRE [IREB]	

18

All Rights Reserved, Copyright Mikio Aoyama, 2011



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

Req. Source (Stakeholder)

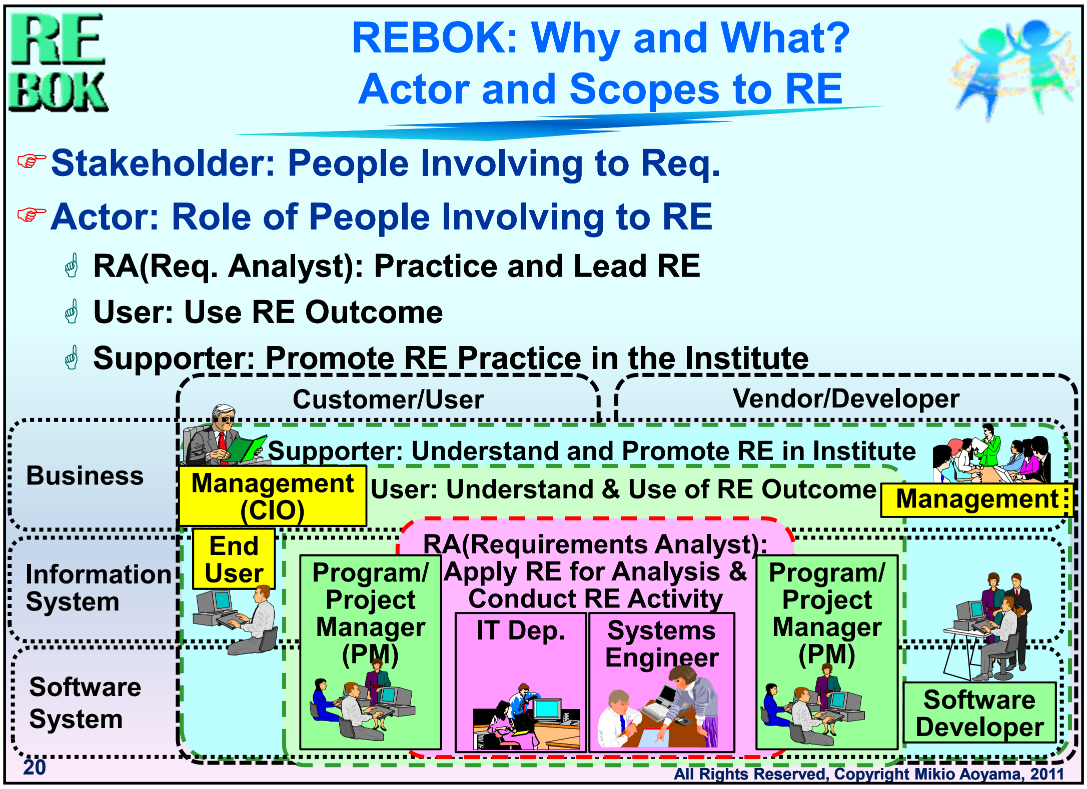
Manager
End-User

Business Req.
Business Strategy
Business Case
Business Rule
Law
Business Environment
Business Process
Business/ Product Req. Def.

System Req.
System-In-Use
System Goal
Functions
Operating Environment
Operating Scenario
QoS/ SLA
System Req. Spec.

Software Req.
Software-In-Use
Software Goal
Screen Transition
Software Environment
Use Case
Software Quality
Software Req. Spec.

19

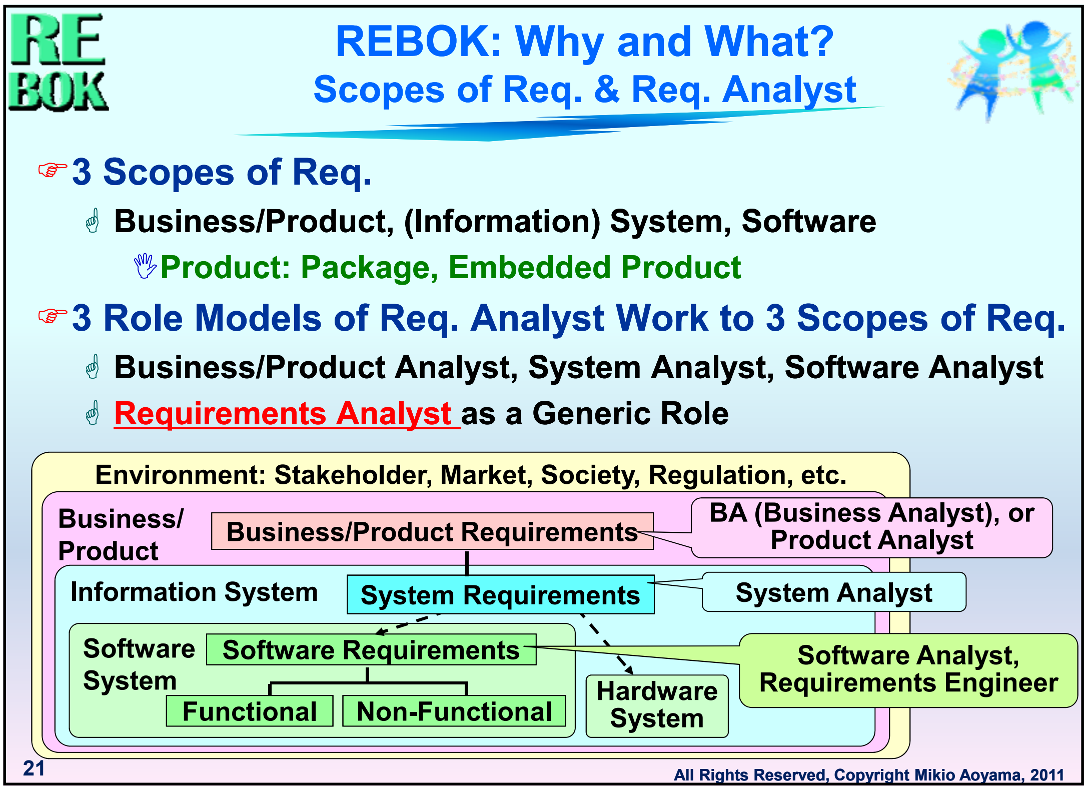


REBOK: Why and What? Actor and Scopes to RE

- 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

Customer/User
Vendor/Developer
Business
Information System
Software System
Supporter: Understand and Promote RE in Institute
Management (CIO)
User: Understand & Use of RE Outcome
Management
End User
RA(Requirements Analyst): Apply RE for Analysis & Conduct RE Activity
Program/ Project Manager (PM)
IT Dep.
Systems Engineer
Program/ Project Manager (PM)
Software Developer

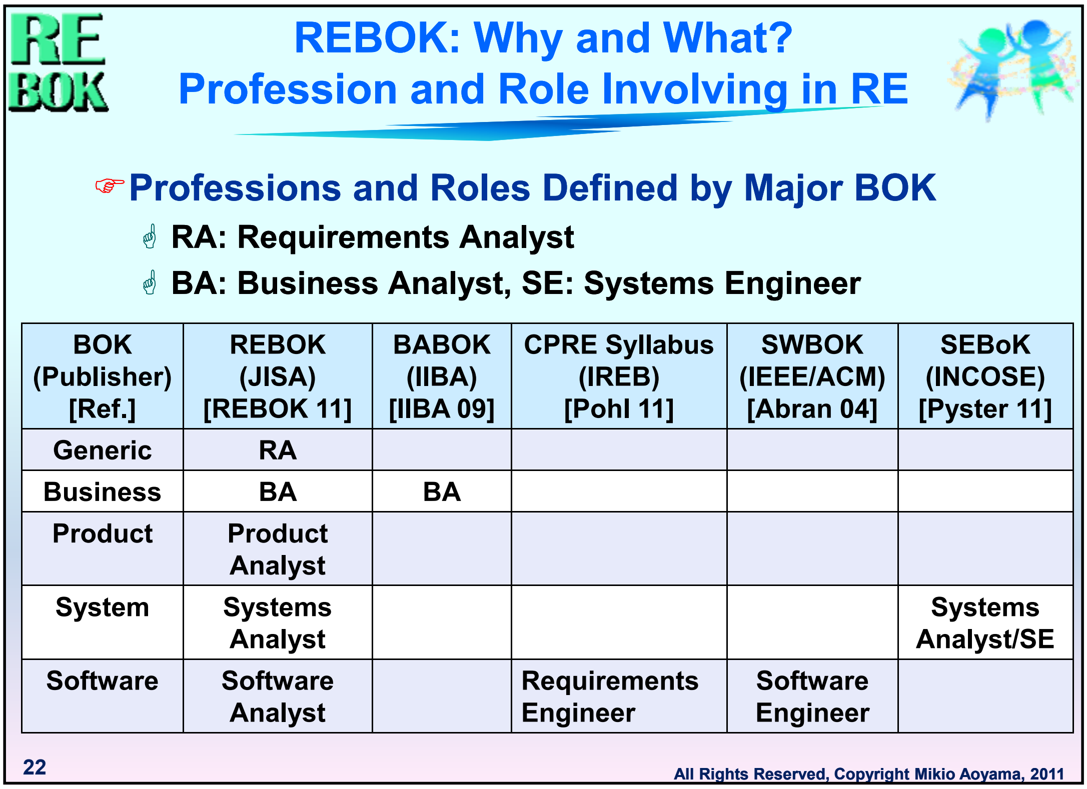
20



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

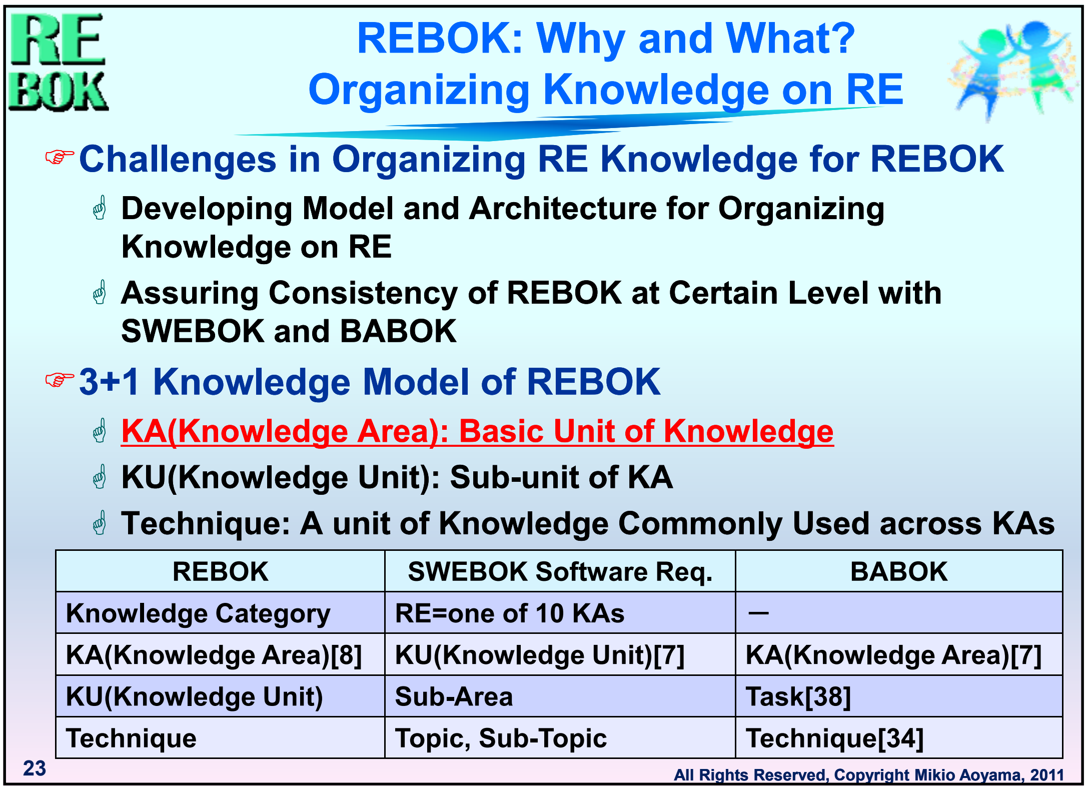
Environment: Stakeholder, Market, Society, Regulation, etc.
Business/ Product
Business/Product Requirements
BA (Business Analyst), or Product Analyst
Information System
System Requirements
System Analyst
Software System
Software Requirements
Functional
Non-Functional
Hardware System
Software Analyst, Requirements Engineer



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

BOK (Publisher) [Ref.]	REBOK (JISA) [REBOK 11]	BABOK (IIBA) [IIBA 09]	CPRE Syllabus (IREB) [Pohl 11]	SWBOK (IEEE/ACM) [Abran 04]	SEBoK (INCOSE) [Pyster 11]
Generic	RA				
Business	BA	BA			
Product	Product Analyst				
System	Systems Analyst				Systems Analyst/SE
Software	Software Analyst		Requirements Engineer	Software Engineer	

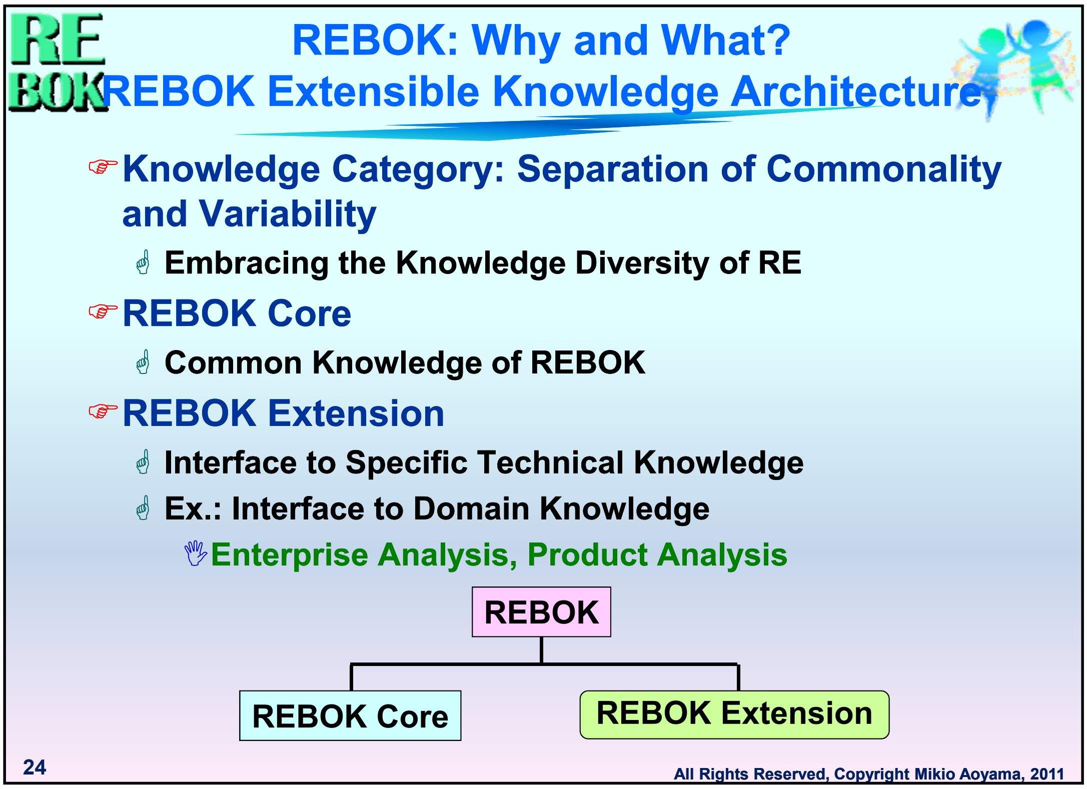


REBOK: Why and What?
Organizing Knowledge on RE

- 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

REBOK	SWEBOK Software Req.	BABOK
Knowledge Category	RE=one of 10 KAs	—
KA(Knowledge Area)[8]	KU(Knowledge Unit)[7]	KA(Knowledge Area)[7]
KU(Knowledge Unit)	Sub-Area	Task[38]
Technique	Topic, Sub-Topic	Technique[34]

23

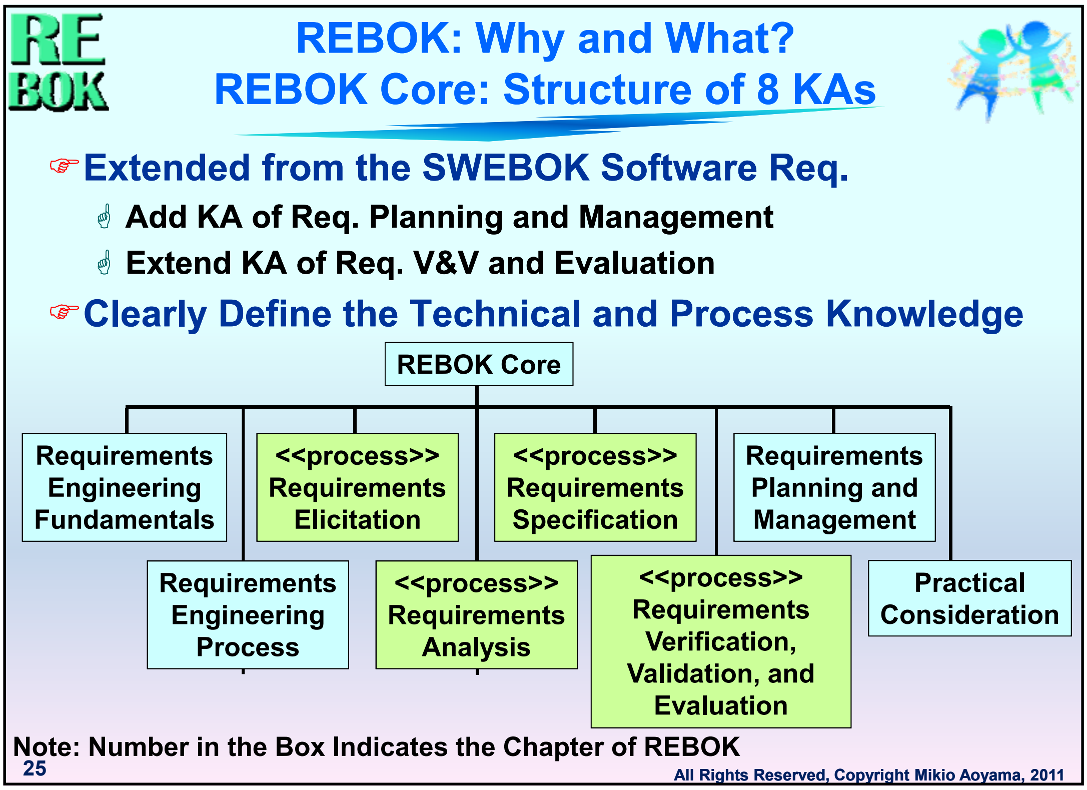


REBOK: Why and What?
REBOK Extensible Knowledge Architecture

- 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
REBOK Core
REBOK Extension

24



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

REBOK Core

Requirements Engineering Fundamentals

<<process>> Requirements Elicitation

<<process>> Requirements Specification

Requirements Planning and Management

Requirements Engineering Process

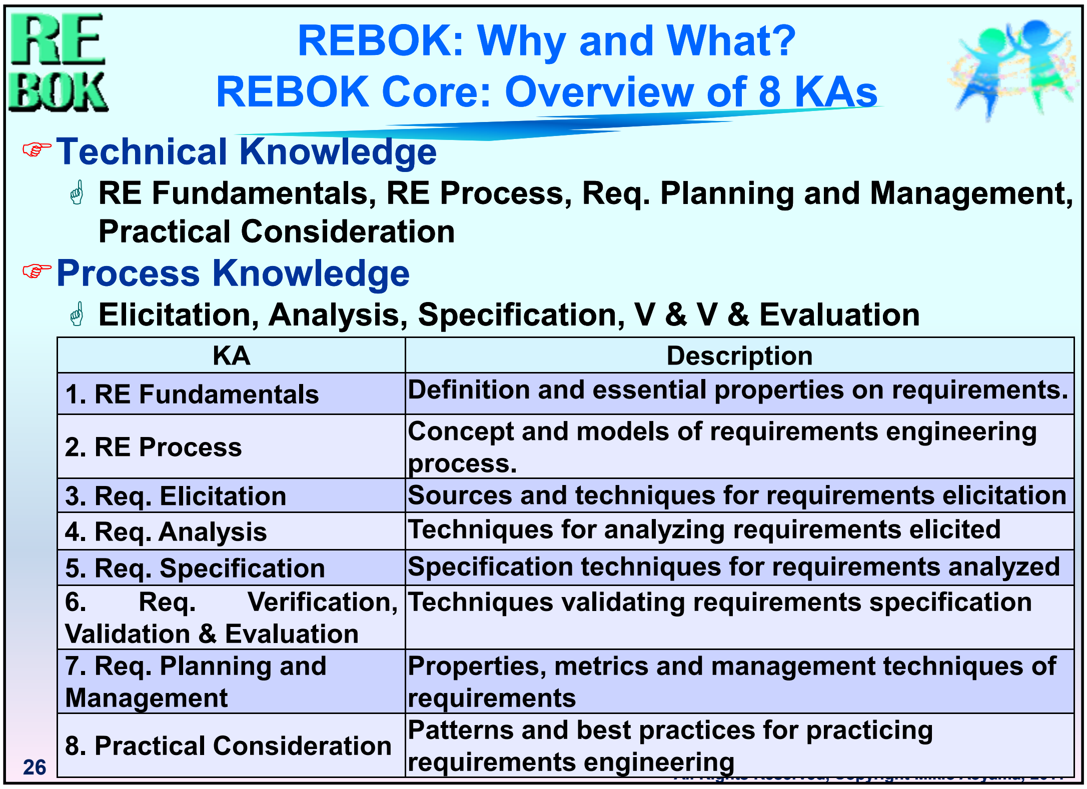
<<process>> Requirements Analysis

<<process>> Requirements Verification, Validation, and Evaluation

Practical Consideration

Note: Number in the Box Indicates the Chapter of REBOK

25

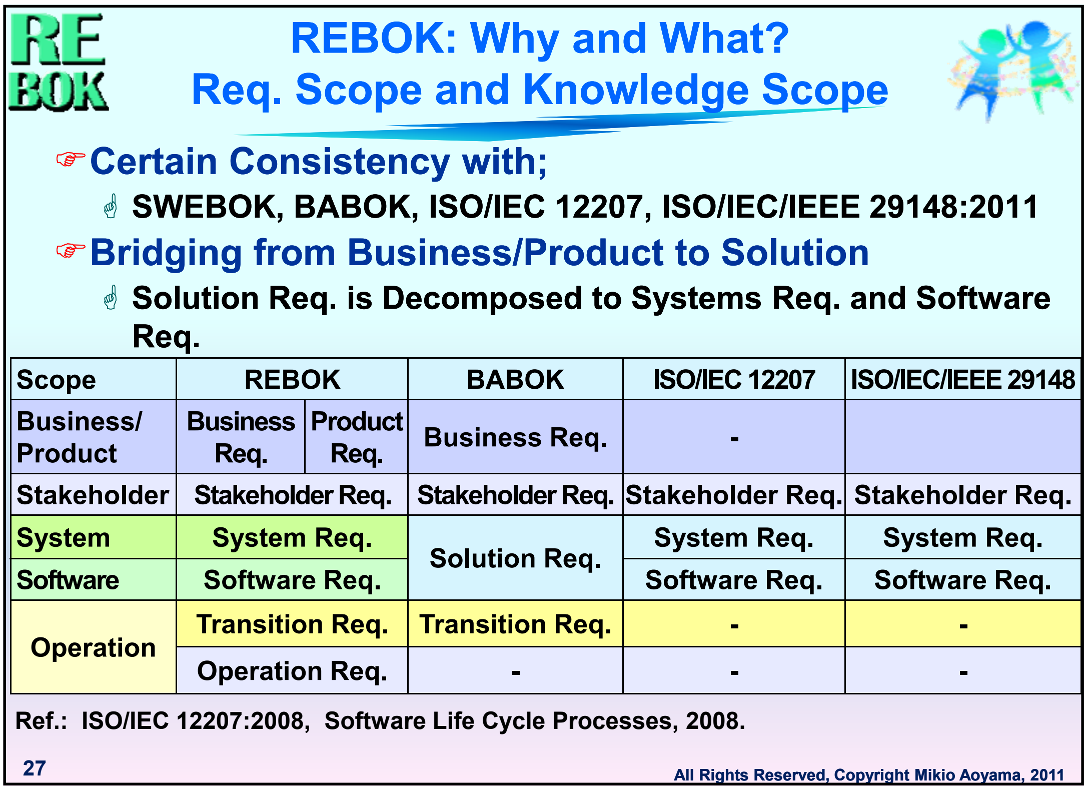


REBOK: Why and What?
REBOK Core: Overview of 8 KAs

- ☞Technical Knowledge
  - RE Fundamentals, RE Process, Req. Planning and Management, Practical Consideration
- ☞Process Knowledge
  - Elicitation, Analysis, Specification, V & V & Evaluation

KA	Description
1. RE Fundamentals	Definition and essential properties on requirements.
2. RE Process	Concept and models of requirements engineering process.
3. Req. Elicitation	Sources and techniques for requirements elicitation
4. Req. Analysis	Techniques for analyzing requirements elicited
5. Req. Specification	Specification techniques for requirements analyzed
6. Req. Verification, Validation & Evaluation	Techniques validating requirements specification
7. Req. Planning and Management	Properties, metrics and management techniques of requirements
8. Practical Consideration	Patterns and best practices for practicing requirements engineering

26



REBOK: Why and What?
Req. Scope and Knowledge Scope

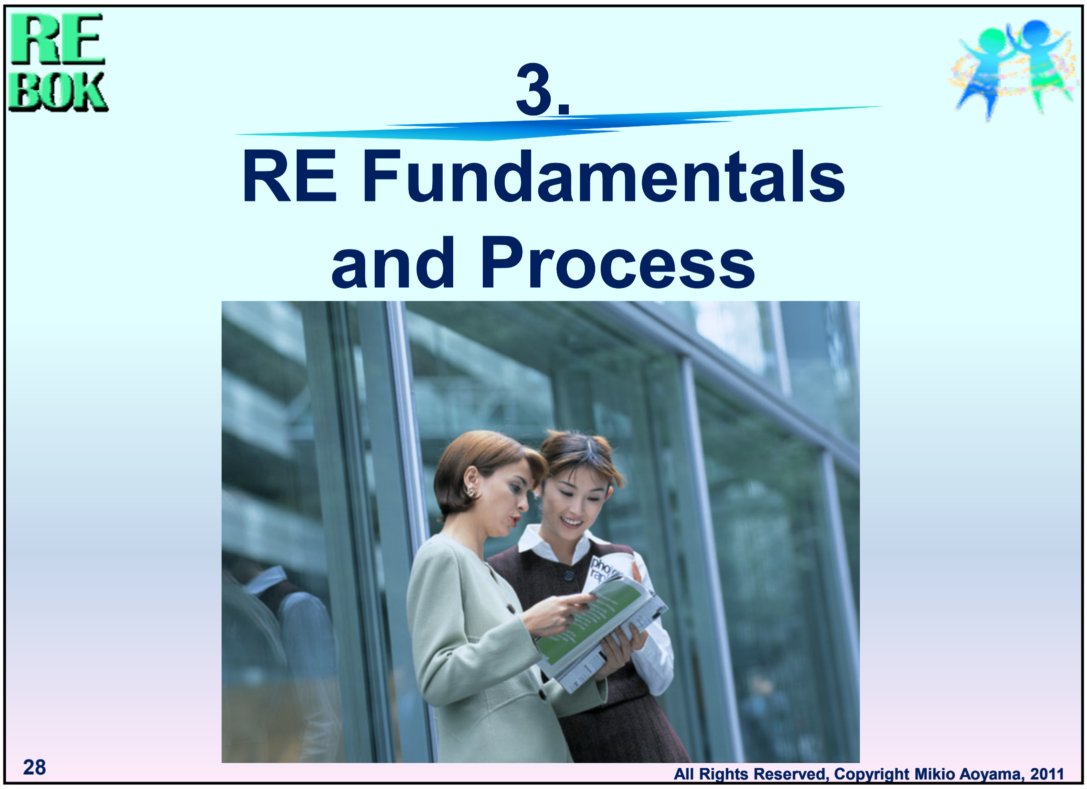
- Certain Consistency with;
  - SWEBOK, BABOK, ISO/IEC 12207, ISO/IEC/IEEE 29148:2011
- Bridging from Business/Product to Solution
  - Solution Req. is Decomposed to Systems Req. and Software Req.

Scope	REBOK		BABOK	ISO/IEC 12207	ISO/IEC/IEEE 29148
Business/ Product	Business Req.	Product Req.	Business Req.	-	
Stakeholder	Stakeholder Req.		Stakeholder Req.	Stakeholder Req.	Stakeholder Req.
System	System Req.		Solution Req.	System Req.	System Req.
Software	Software Req.			Software Req.	Software Req.
Operation	Transition Req.		Transition Req.	-	-
	Operation Req.		-	-	-

Ref.: ISO/IEC 12207:2008, Software Life Cycle Processes, 2008.

27

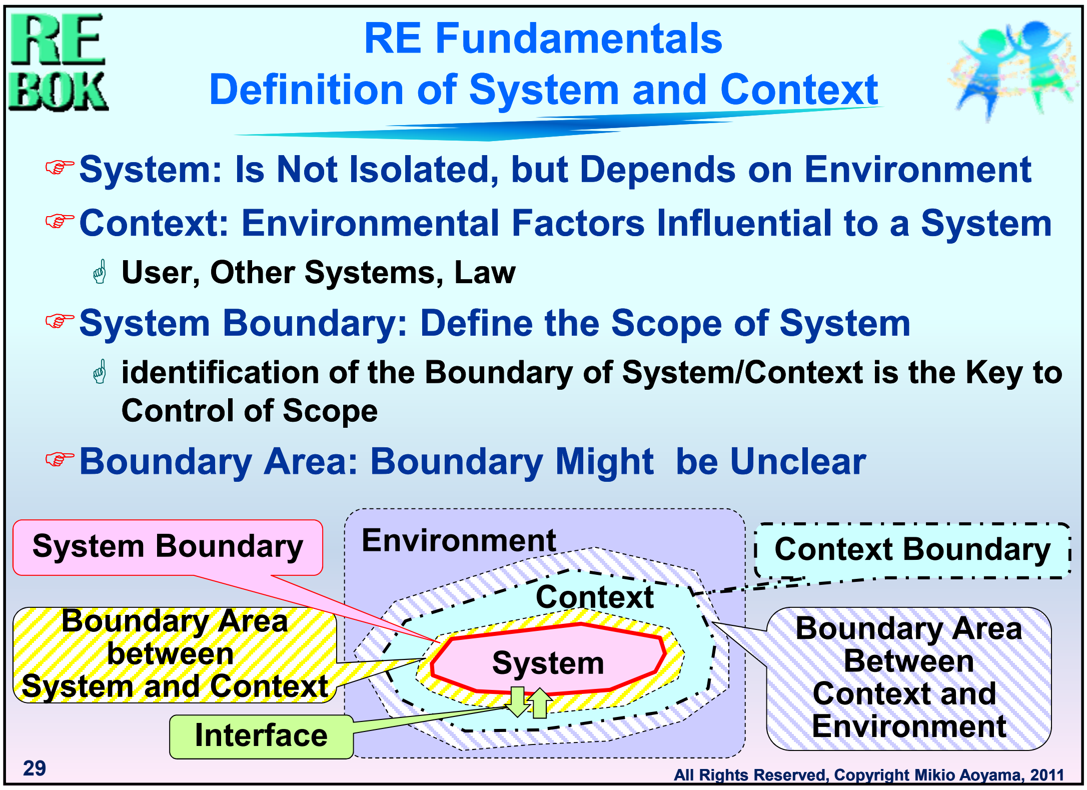
All Rights Reserved, Copyright Mikio Aoyama, 2011



3.
RE Fundamentals and Process

28

All Rights Reserved, Copyright Mikio Aoyama, 2011



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

System Boundary

Environment

Context Boundary

Context

Boundary Area between System and Context

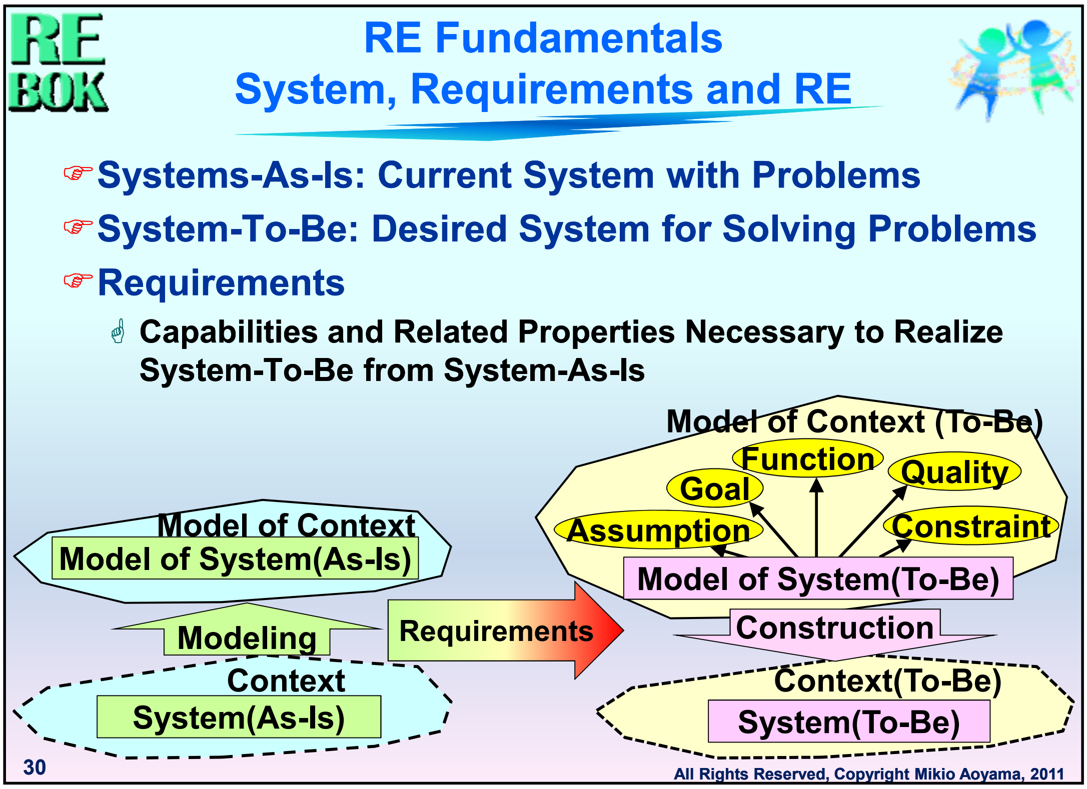
System

Boundary Area Between Context and Environment

Interface

29

All Rights Reserved, Copyright Mikio Aoyama, 2011



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

Model of Context (To-Be)

Function

Goal

Quality

Assumption

Constraint

Model of Context

Model of System(As-Is)

Model of System(To-Be)

Modeling

Requirements

Construction

Context

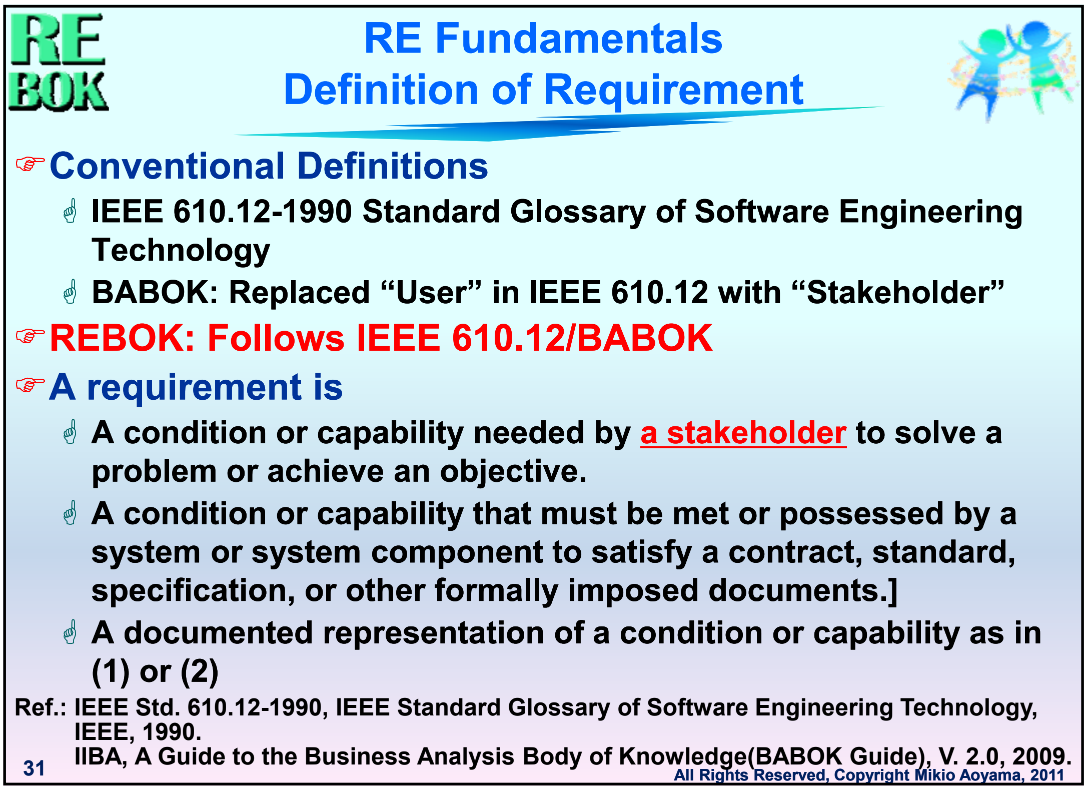
System(As-Is)

Context(To-Be)

System(To-Be)

30

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE Fundamentals
Definition of Requirement

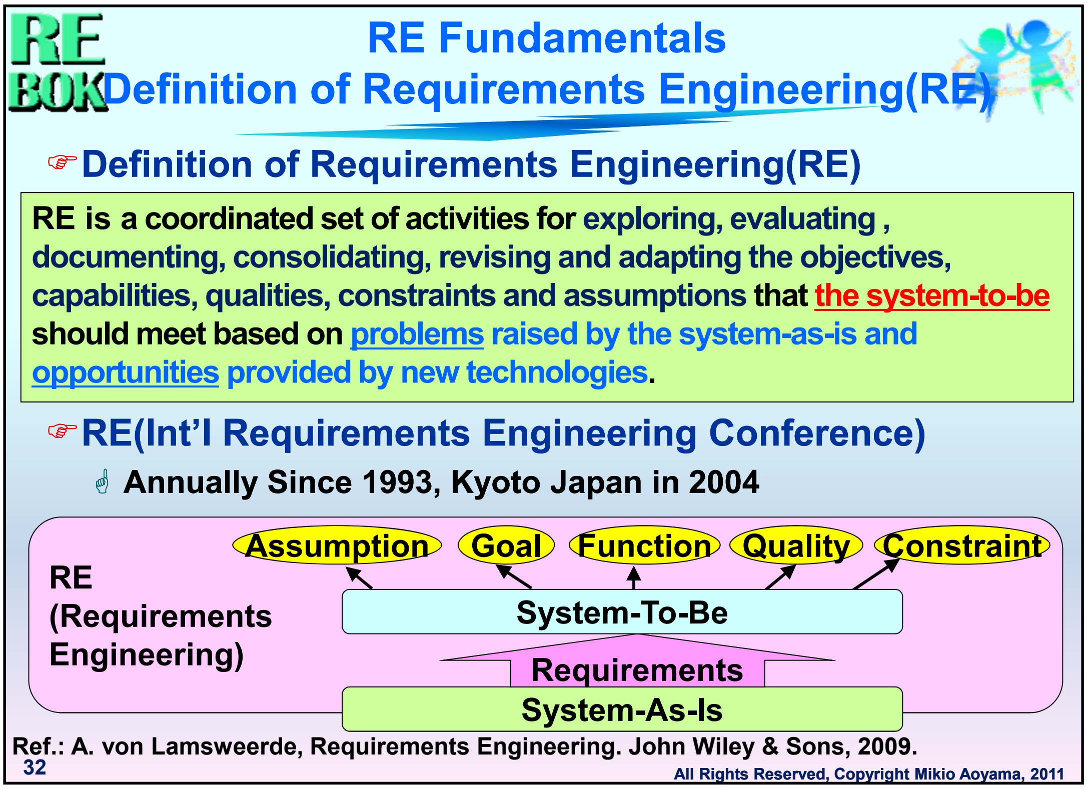
☞ Conventional Definitions
- IEEE 610.12-1990 Standard Glossary of Software Engineering Technology
- BABOK: Replaced “User” in IEEE 610.12 with “Stakeholder”

☞ 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)

Ref.: IEEE Std. 610.12-1990, IEEE Standard Glossary of Software Engineering Technology, IEEE, 1990.
IIBA, A Guide to the Business Analysis Body of Knowledge(BABOK Guide), V. 2.0, 2009.

31



RE Fundamentals
Definition of Requirements Engineering(RE)

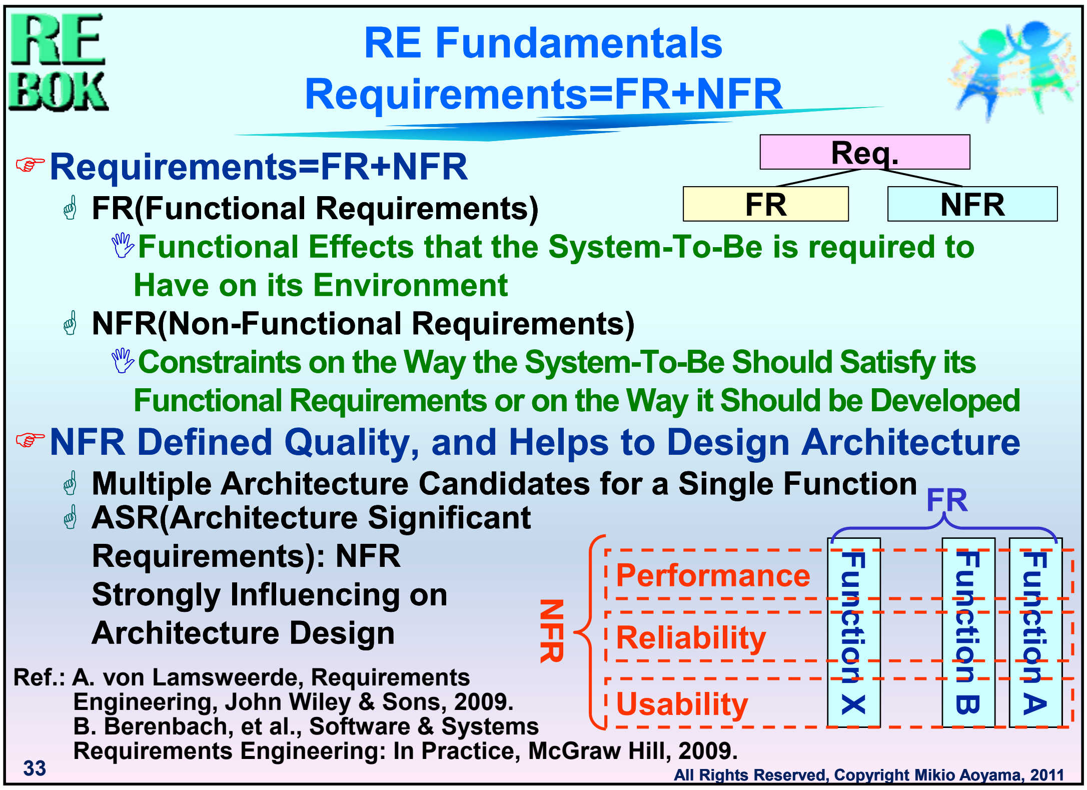
☞ Definition of Requirements Engineering(RE)

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

Ref.: A. von Lamsweerde, Requirements Engineering. John Wiley & Sons, 2009.

32



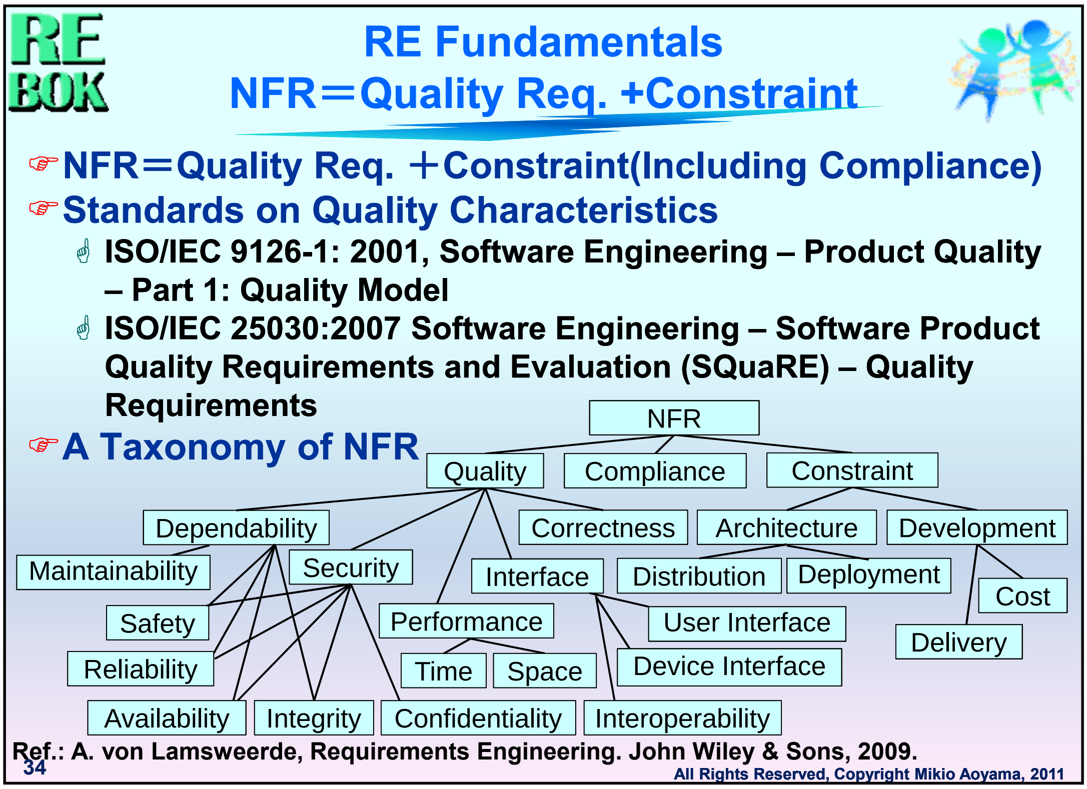
RE Fundamentals
Requirements=FR+NFR

- 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

Ref.: A. von Lamsweerde, Requirements Engineering, John Wiley & Sons, 2009.
B. Berenbach, et al., Software & Systems Requirements Engineering: In Practice, McGraw Hill, 2009.

33

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE Fundamentals
NFR＝Quality Req. +Constraint

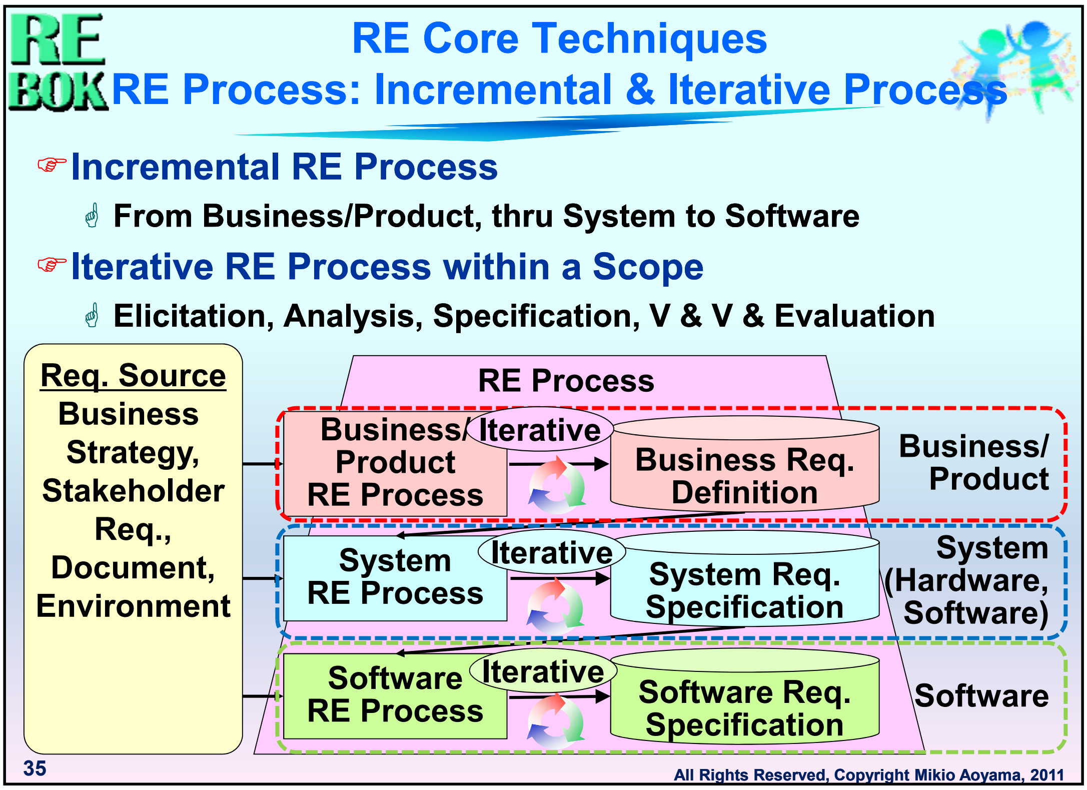
- NFR＝Quality Req. ＋Constraint(Including Compliance)
- Standards on Quality Characteristics
  - ISO/IEC 9126-1: 2001, Software Engineering – Product Quality – Part 1: Quality Model
  - ISO/IEC 25030:2007 Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Quality Requirements
- A Taxonomy of NFR

RE BOK
NFR
Quality
Compliance
Constraint
Dependability
Correctness
Architecture
Development
Maintainability
Security
Interface
Distribution
Deployment
Cost
Safety
Performance
User Interface
Delivery
Reliability
Time
Space
Device Interface
Availability
Integrity
Confidentiality
Interoperability

Ref.: A. von Lamsweerde, Requirements Engineering. John Wiley & Sons, 2009.

34

All Rights Reserved, Copyright Mikio Aoyama, 2011



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

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

RE Process

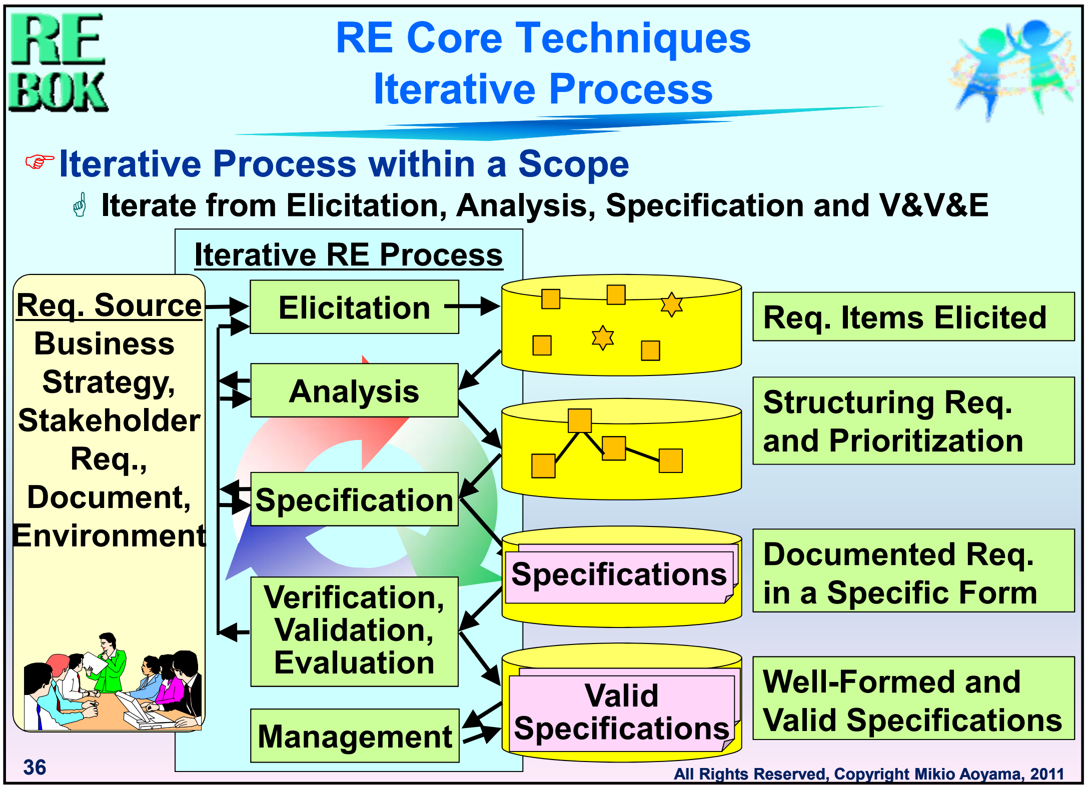
Business/Product RE Process
Iterative
Business Req. Definition
Business/Product

System RE Process
Iterative
System Req. Specification
System (Hardware, Software)

Software RE Process
Iterative
Software Req. Specification
Software

35

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE Core Techniques
Iterative Process

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

Iterative RE Process

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

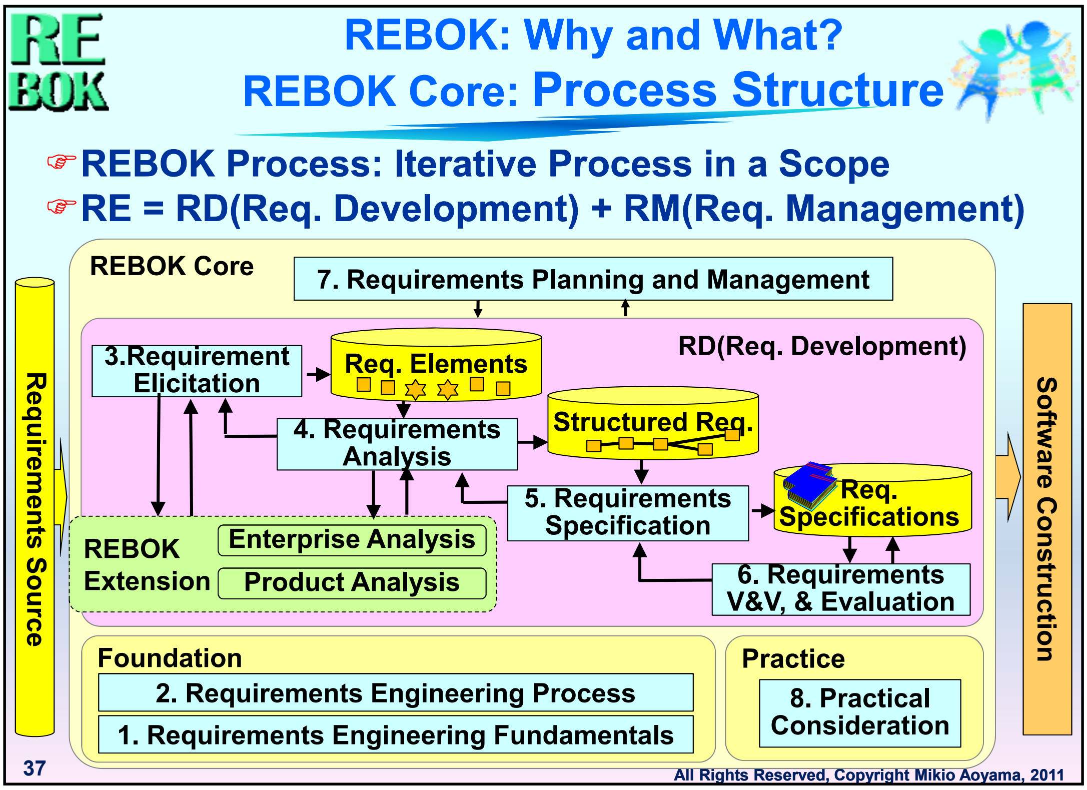
Elicitation
Analysis
Specification
Verification, Validation, Evaluation
Management

Specifications
Valid Specifications

Req. Items Elicited
Structuring Req. and Prioritization
Documented Req. in a Specific Form
Well-Formed and Valid Specifications

36

All Rights Reserved, Copyright Mikio Aoyama, 2011



REBOK: Why and What?
REBOK Core: Process Structure

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

Requirements Source

REBOK Core

7. Requirements Planning and Management

RD(Req. Development)

3.Requirement Elicitation

Req. Elements

4. Requirements Analysis

Structured Req.

5. Requirements Specification

Req. Specifications

6. Requirements V&V, & Evaluation

REBOK Extension

Enterprise Analysis

Product Analysis

Foundation

2. Requirements Engineering Process

1. Requirements Engineering Fundamentals

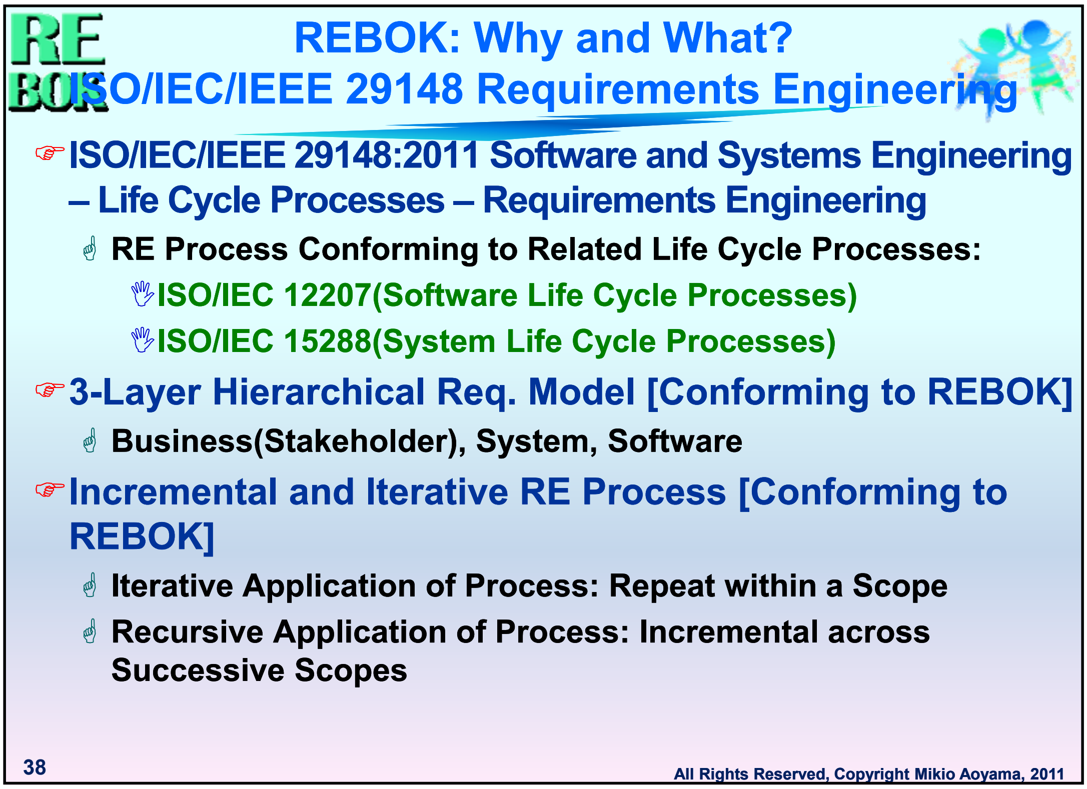
Practice

8. Practical Consideration

Software Construction

37

All Rights Reserved, Copyright Mikio Aoyama, 2011

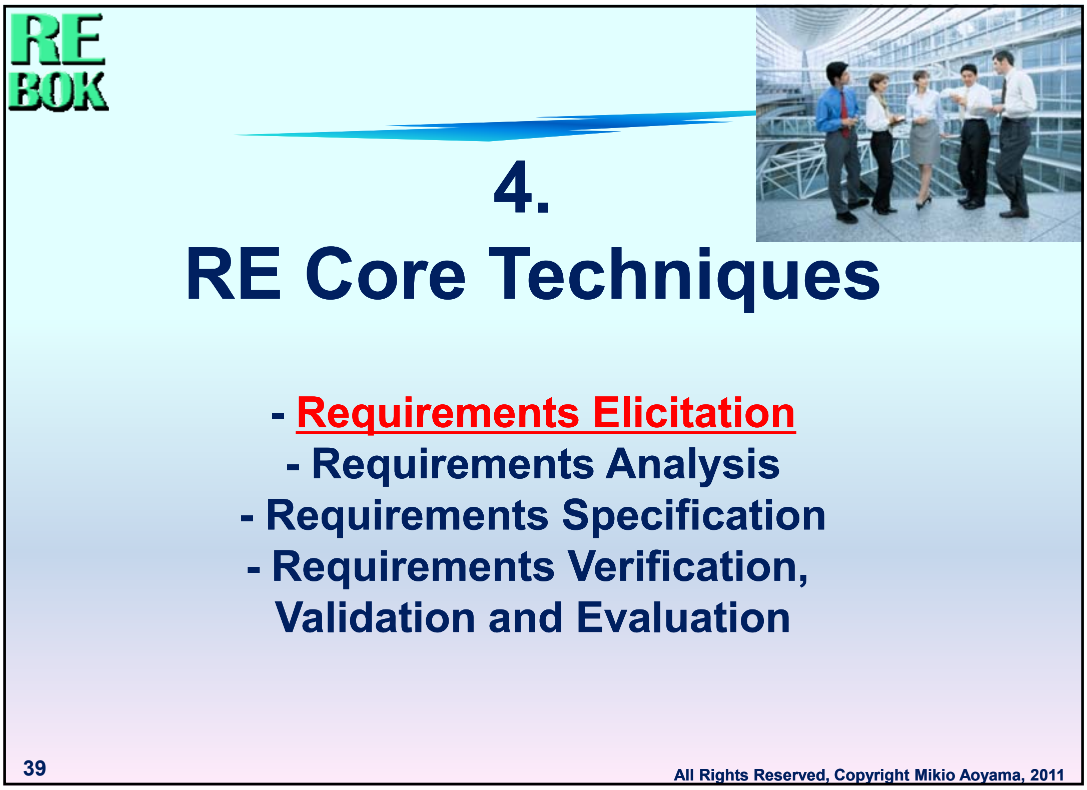


REBOK: Why and What?
ISO/IEC/IEEE 29148 Requirements Engineering

- ISO/IEC/IEEE 29148:2011 Software and Systems Engineering – Life Cycle Processes – 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

38

All Rights Reserved, Copyright Mikio Aoyama, 2011



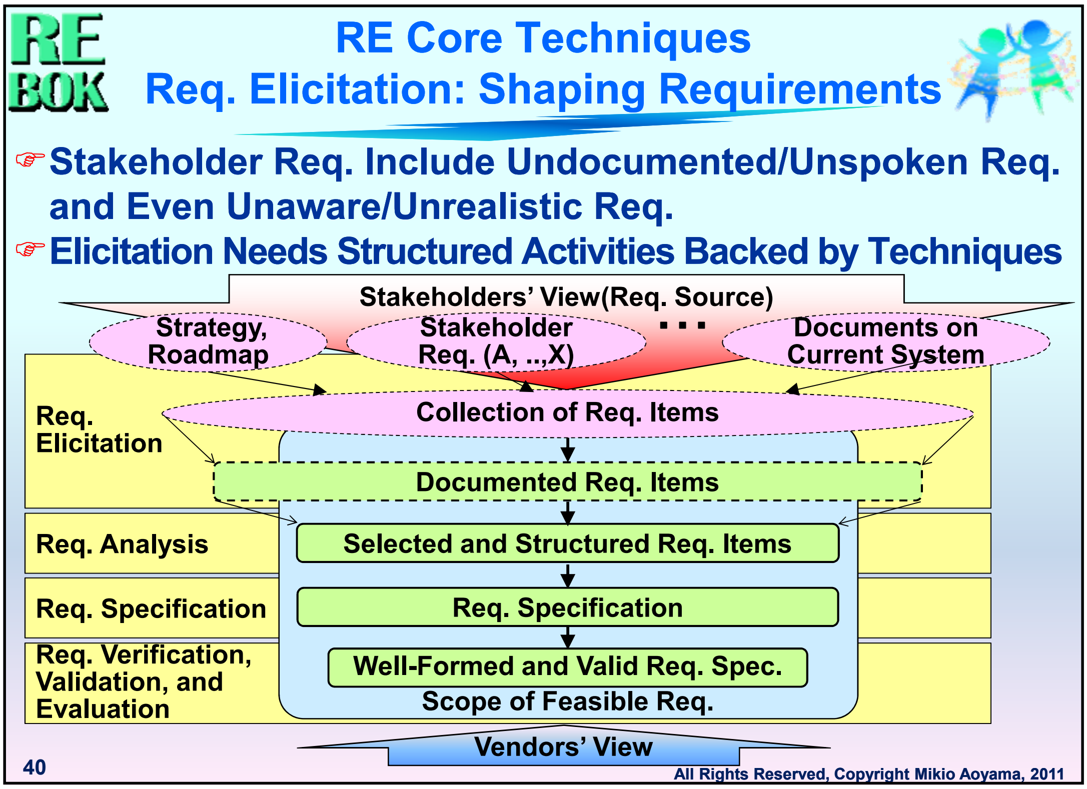
RE BOK

4. RE Core Techniques

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

39

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE BOK

RE Core Techniques
Req. Elicitation: Shaping Requirements

☞ Stakeholder Req. Include Undocumented/Unspoken Req. and Even Unaware/Unrealistic Req.

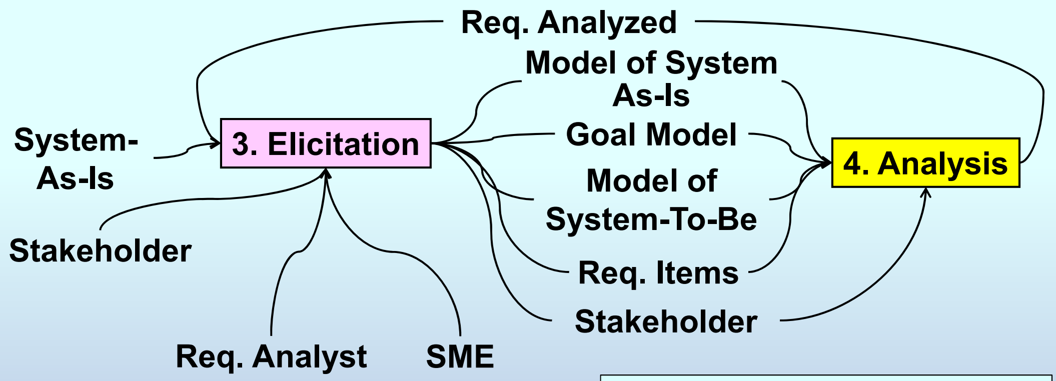
☞ Elicitation Needs Structured Activities Backed by Techniques

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

40

All Rights Reserved, Copyright Mikio Aoyama, 2011

RE Core Techniques
Req. Elicitation: Elicitation Process



Req. Analyzed

System-As-Is

3. Elicitation

Stakeholder

Req. Analyst

SME

Model of System As-Is

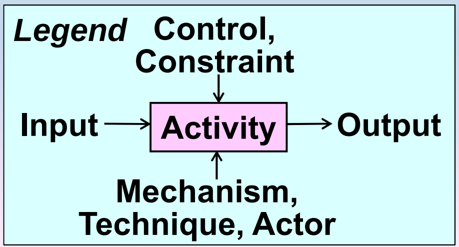
Goal Model

Model of System-To-Be

Req. Items

Stakeholder

4. Analysis



Legend

Control, Constraint

Input

Activity

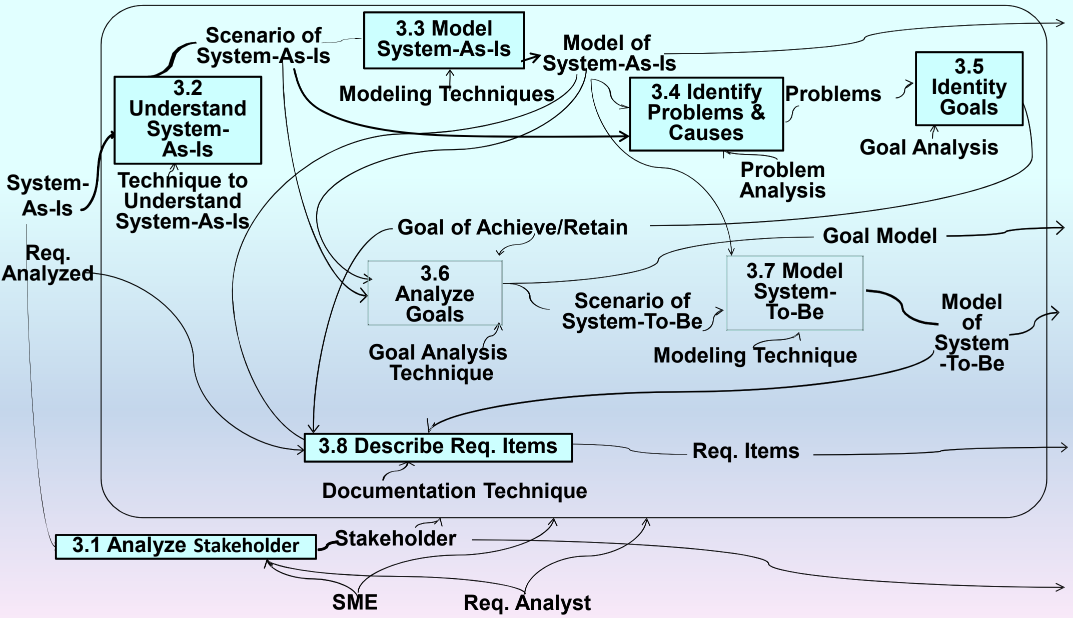
Output

Mechanism, Technique, Actor

SME: Subject Matter Expert
i.e. Domain Expert

41

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



Scenario of System-As-Is

3.3 Model System-As-Is

Model of System-As-Is

Modeling Techniques

3.2 Understand System-As-Is

3.4 Identify Problems & Causes

Problems

3.5 Identity Goals

Goal Analysis

Problem Analysis

System-As-Is

Technique to Understand System-As-Is

Req. Analyzed

Goal of Achieve/Retain

Goal Model

3.6 Analyze Goals

Scenario of System-To-Be

3.7 Model System-To-Be

Model of System -To-Be

Goal Analysis Technique

Modeling Technique

3.8 Describe Req. Items

Req. Items

Documentation Technique

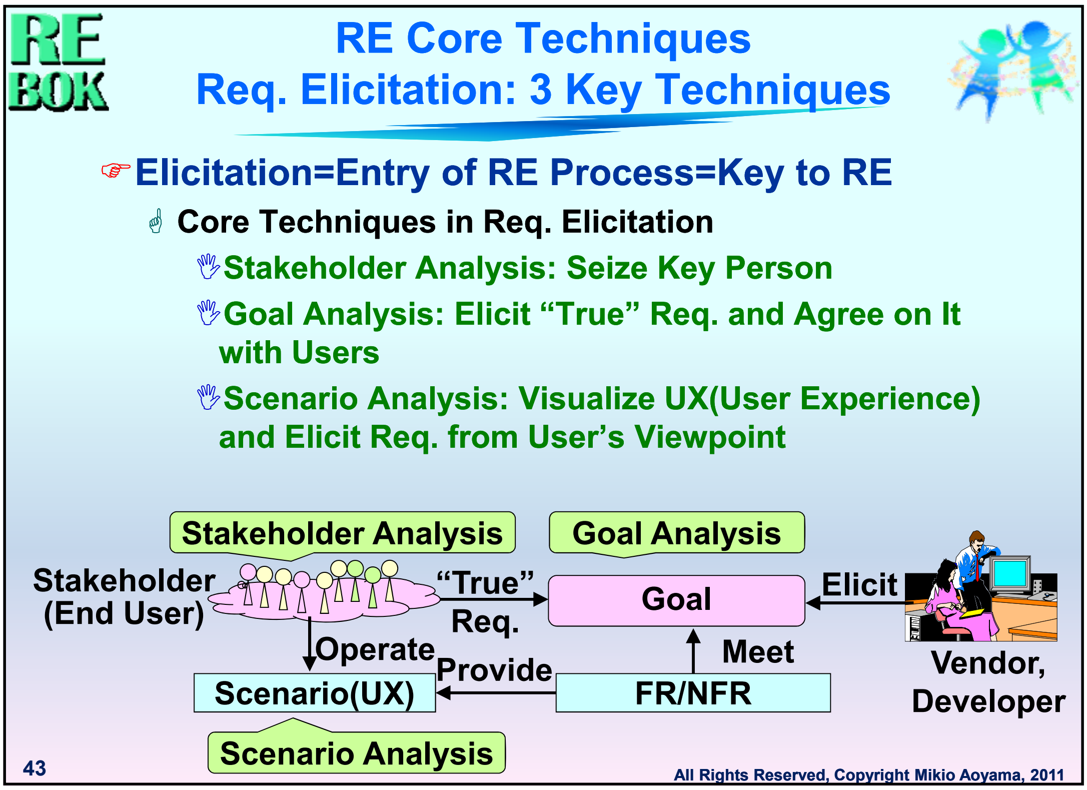
3.1 Analyze Stakeholder

Stakeholder

SME

Req. Analyst

42

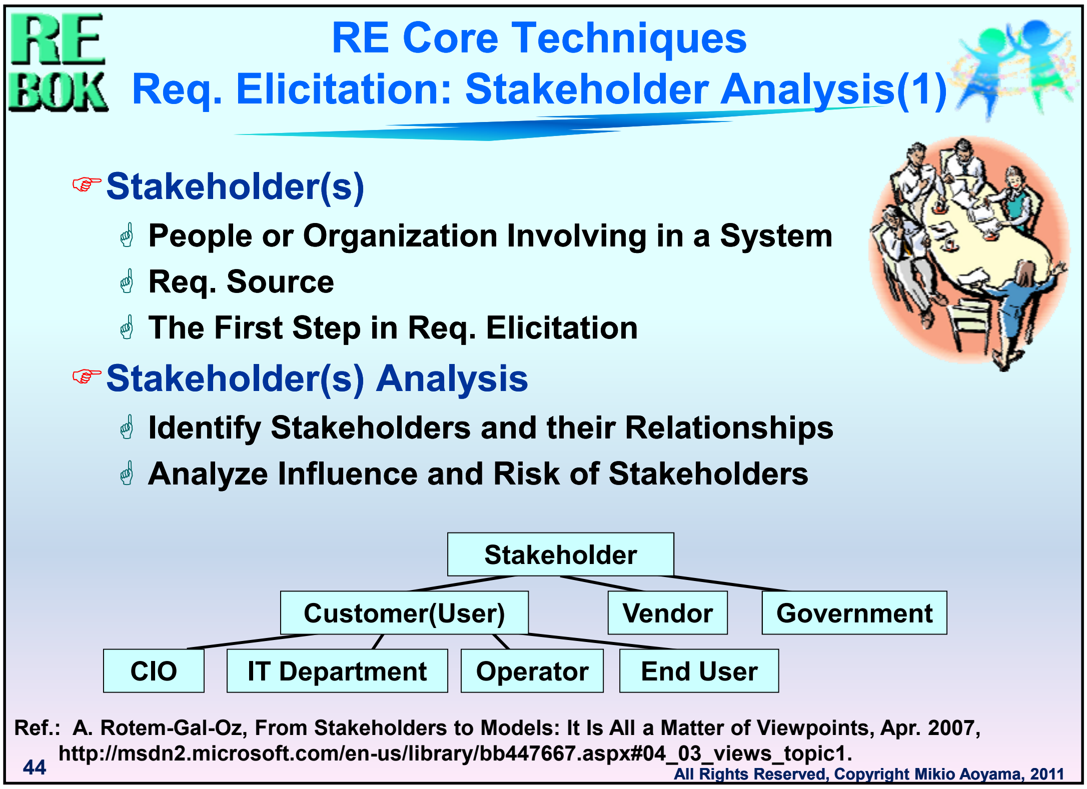


RE Core Techniques
Req. Elicitation: 3 Key Techniques

- 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
Goal Analysis
Stakeholder (End User)
“True” Req.
Goal
Elicit
Operate
Meet
Provide
Scenario(UX)
FR/NFR
Vendor, Developer
Scenario Analysis

43



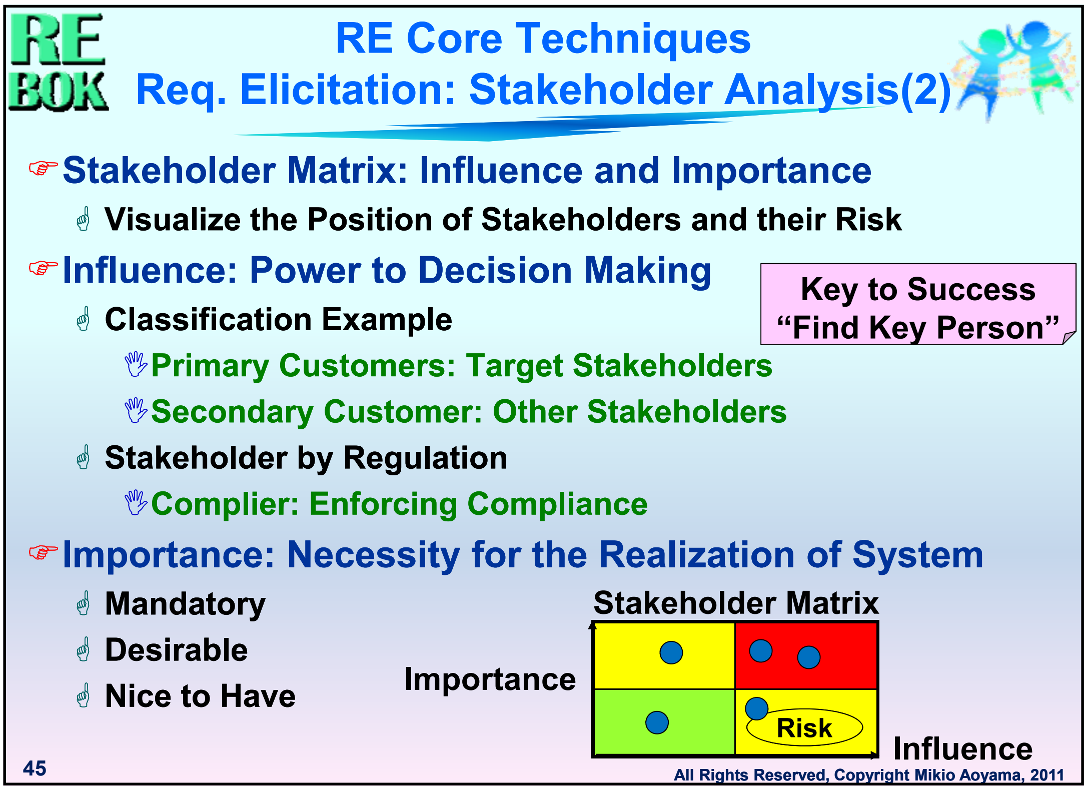
RE Core Techniques
Req. Elicitation: Stakeholder Analysis(1)

- Stakeholder(s)
  - People or Organization Involving in a System
  - Req. Source
  - The First Step in Req. Elicitation
- Stakeholder(s) Analysis
  - Identify Stakeholders and their Relationships
  - Analyze Influence and Risk of Stakeholders

Stakeholder
Customer(User)
Vendor
Government
CIO
IT Department
Operator
End User

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.

44

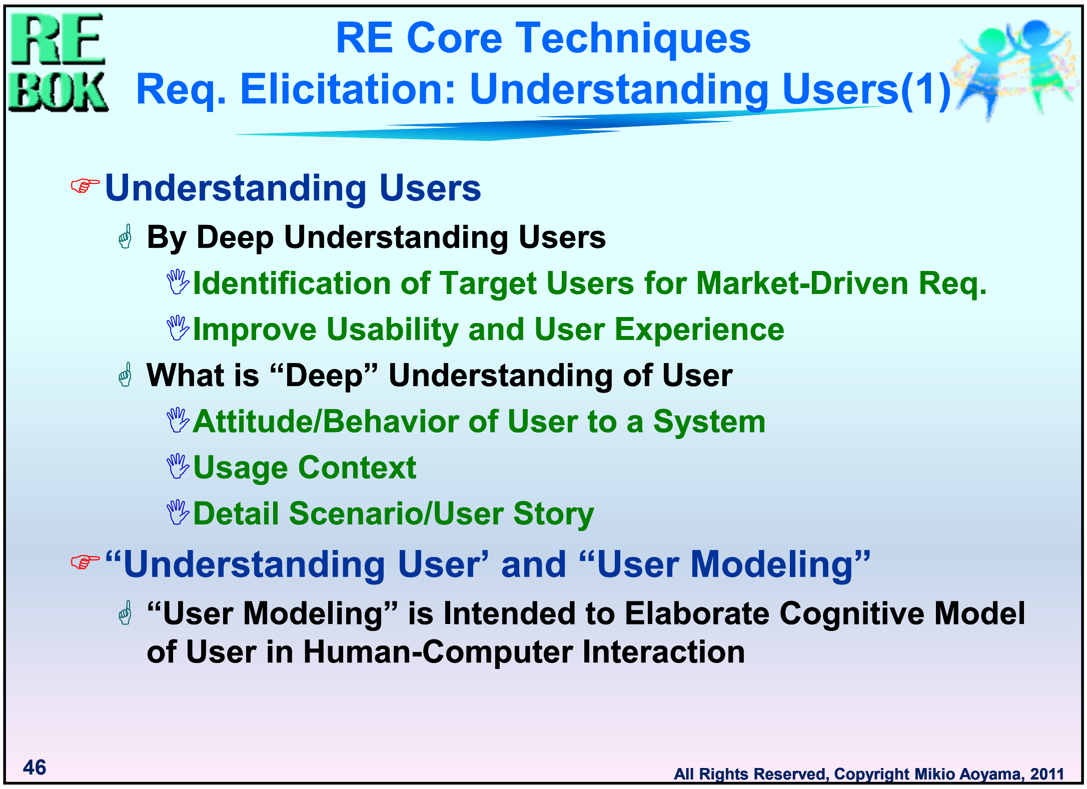


RE Core Techniques
Req. Elicitation: Stakeholder Analysis(2)

- 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

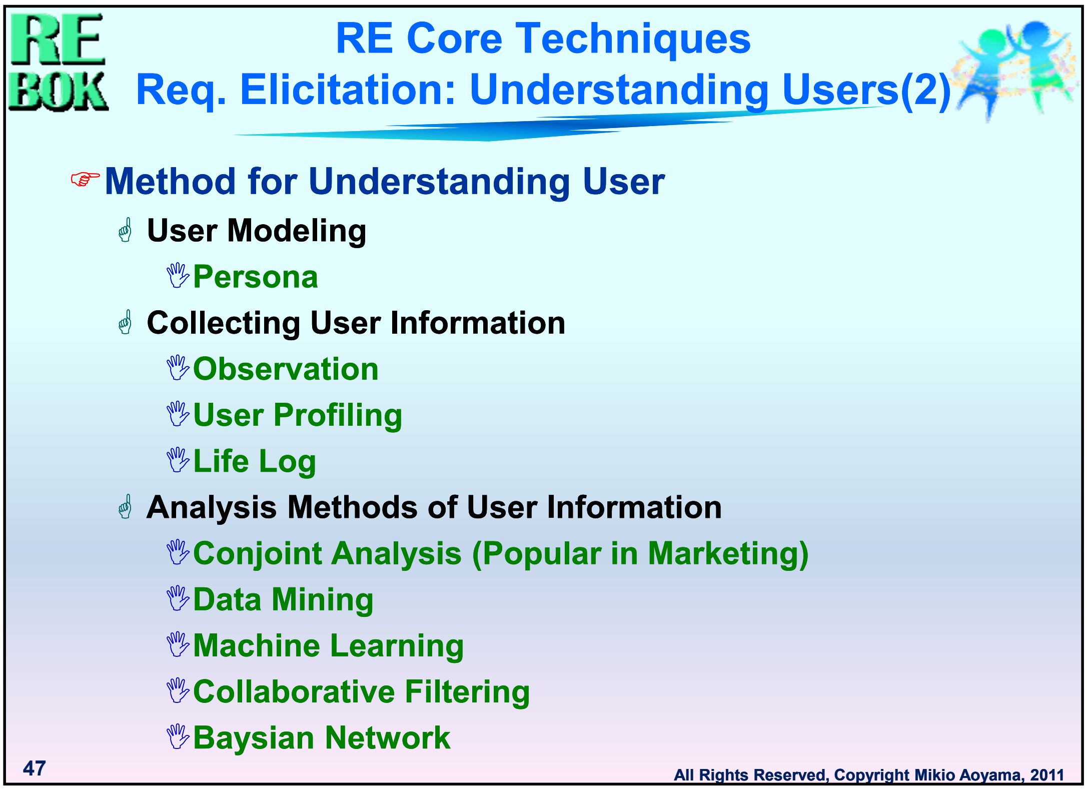
Key to Success
“Find Key Person”

Stakeholder Matrix



RE Core Techniques
Req. Elicitation: Understanding Users(1)

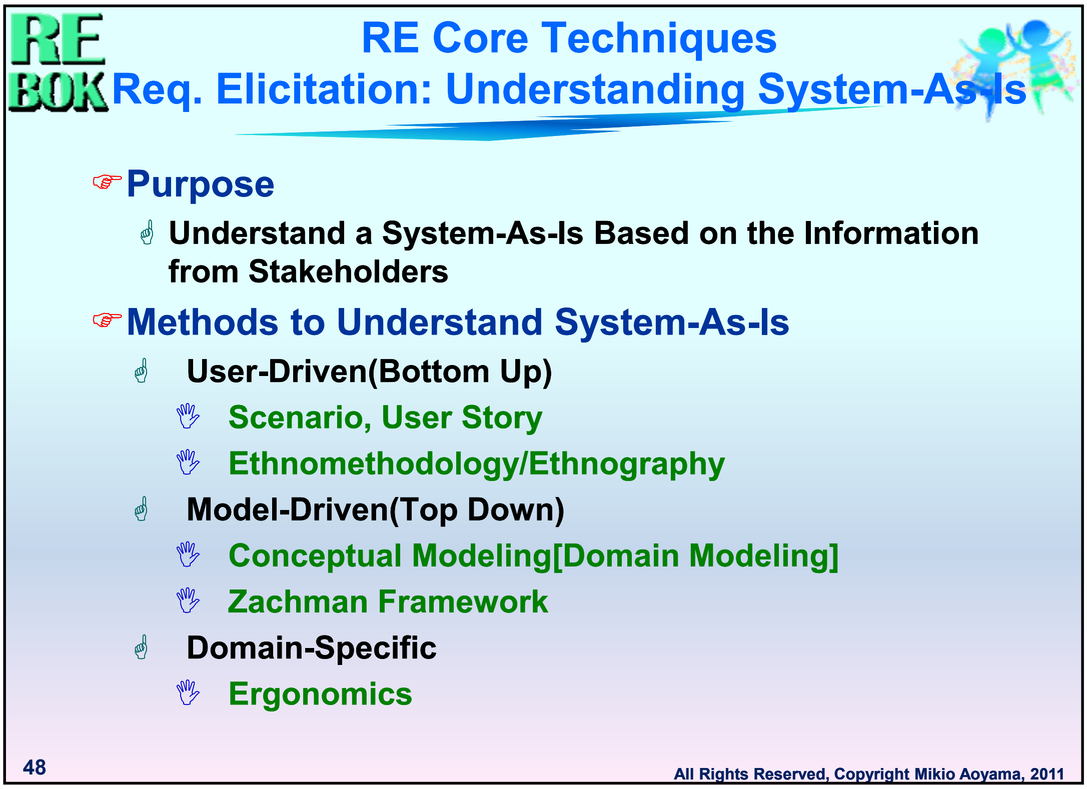
- Understanding Users
  - By Deep Understanding Users
    - Identification of Target Users for Market-Driven Req.
    - Improve Usability and User Experience
  - What is “Deep” Understanding of User
    - Attitude/Behavior of User to a System
    - Usage Context
    - Detail Scenario/User Story
- “Understanding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

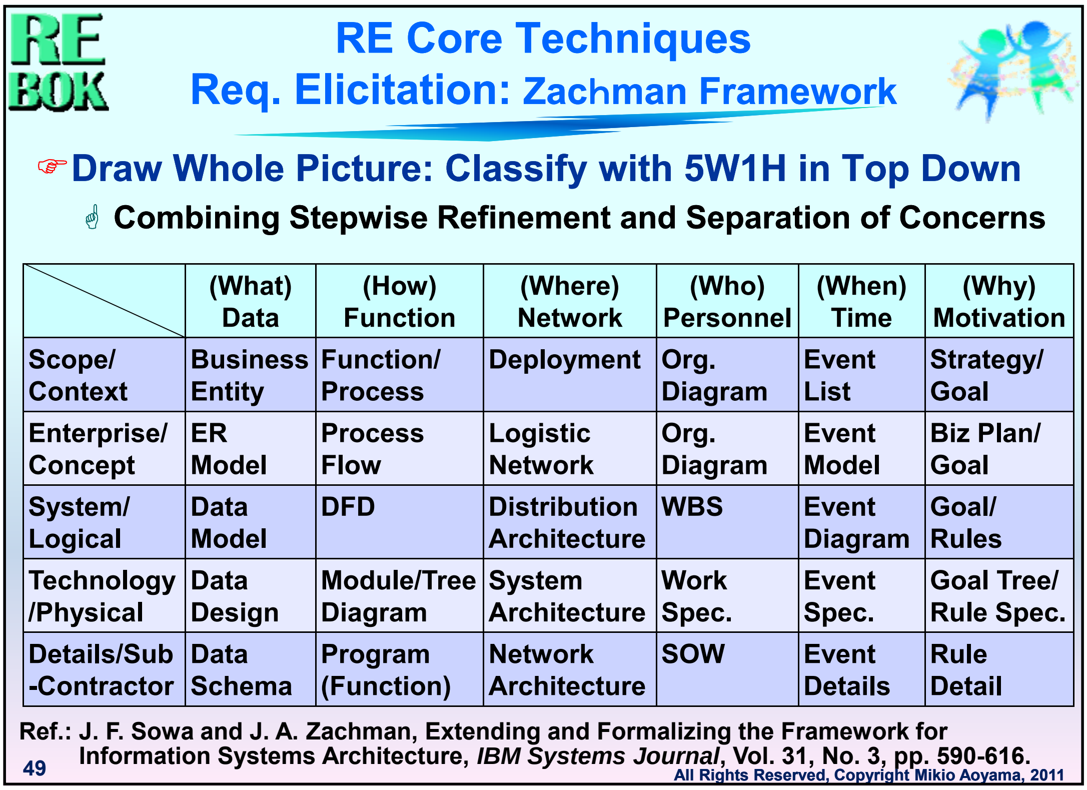
47



RE Core Techniques
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

48



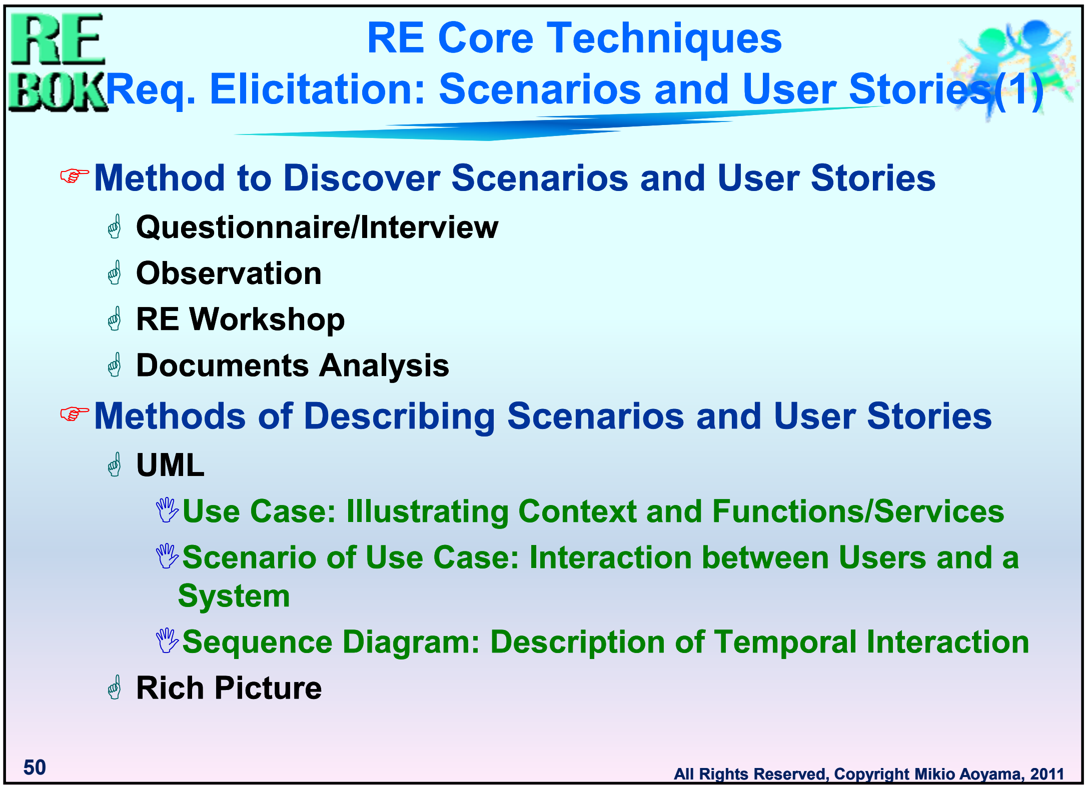
RE Core Techniques
Req. Elicitation: Zachman Framework

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

	(What) Data	(How) Function	(Where) Network	(Who) Personnel	(When) Time	(Why) Motivation
Scope/ Context	Business Entity	Function/ Process	Deployment	Org. Diagram	Event List	Strategy/ Goal
Enterprise/ Concept	ER Model	Process Flow	Logistic Network	Org. Diagram	Event Model	Biz Plan/ Goal
System/ Logical	Data Model	DFD	Distribution Architecture	WBS	Event Diagram	Goal/ Rules
Technology /Physical	Data Design	Module/Tree Diagram	System Architecture	Work Spec.	Event Spec.	Goal Tree/ Rule Spec.
Details/Sub -Contractor	Data Schema	Program (Function)	Network Architecture	SOW	Event Details	Rule Detail

Ref.: J. F. Sowa and J. A. Zachman, Extending and Formalizing the Framework for Information Systems Architecture, *IBM Systems Journal*, Vol. 31, No. 3, pp. 590-616.

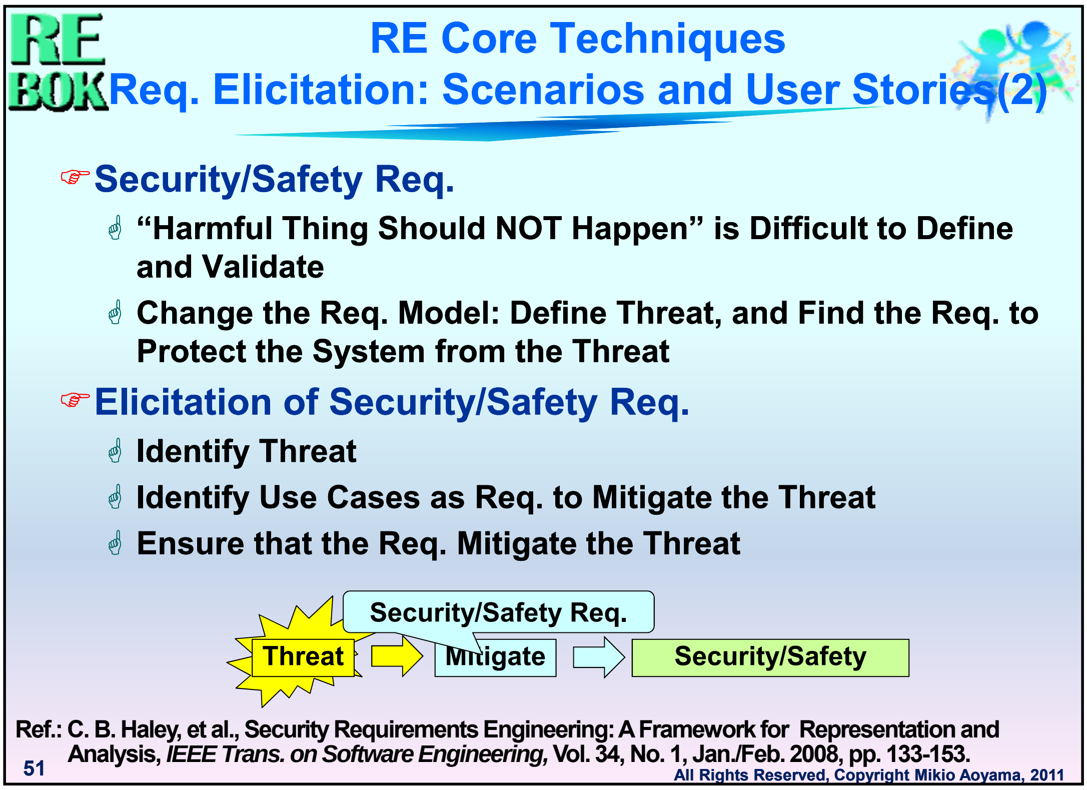
49



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

50

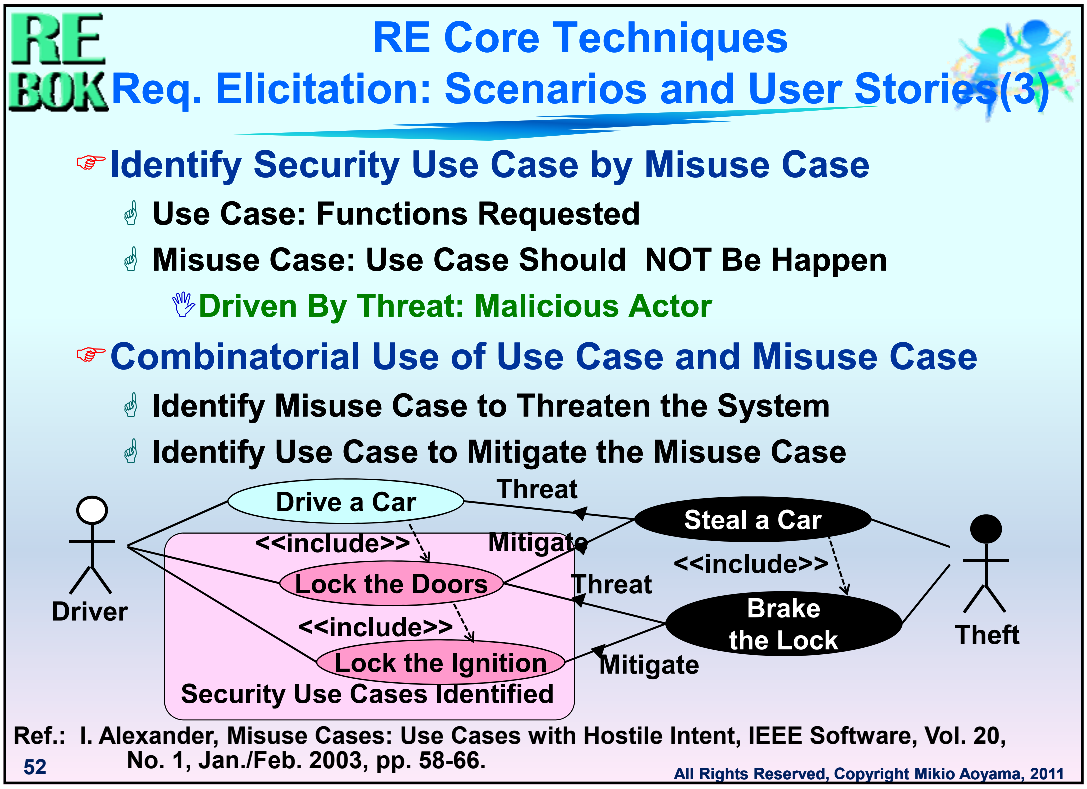


RE Core Techniques
Req. Elicitation: Scenarios and User Stories(2)

- 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

Ref.: C. B. Haley, et al., Security Requirements Engineering: A Framework for Representation and Analysis, *IEEE Trans. on Software Engineering,* Vol. 34, No. 1, Jan./Feb. 2008, pp. 133-153.

51



RE BOK

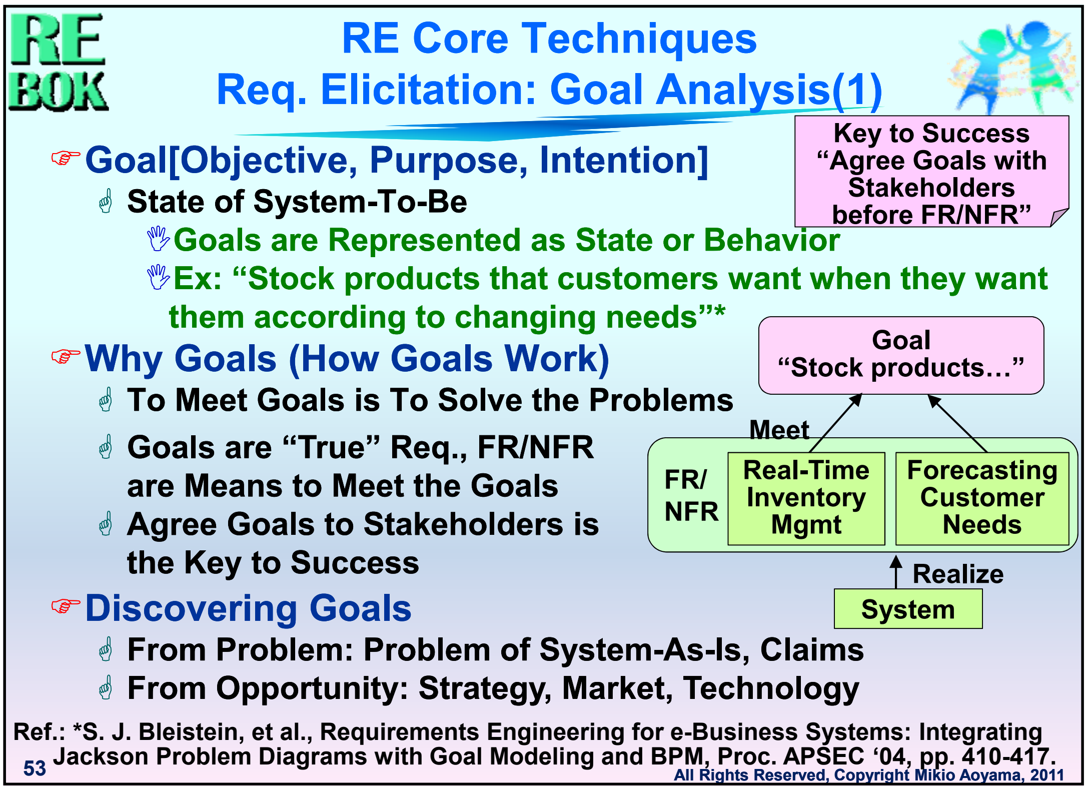
RE Core Techniques
Req. Elicitation: Scenarios and User Stories(3)

- 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

Drive a Car
Threat
Steal a Car
<<include>>
Mitigate
<<include>>
Lock the Doors
Threat
Driver
Brake the Lock
Theft
<<include>>
Lock the Ignition
Mitigate
Security Use Cases Identified

Ref.: I. Alexander, Misuse Cases: Use Cases with Hostile Intent, IEEE Software, Vol. 20, No. 1, Jan./Feb. 2003, pp. 58-66.

52



RE BOK

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

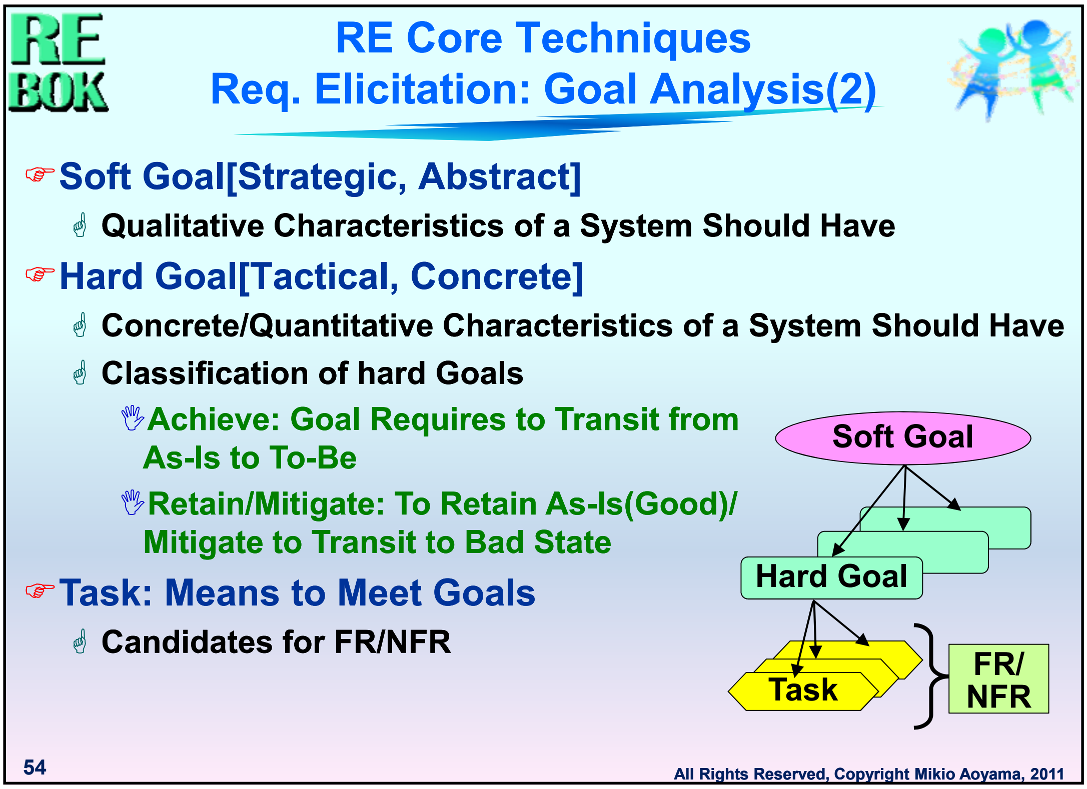
Key to Success “Agree Goals with Stakeholders before FR/NFR”

- 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

Ref.: *S. J. Bleistein, et al., Requirements Engineering for e-Business Systems: Integrating Jackson Problem Diagrams with Goal Modeling and BPM, Proc. APSEC ‘04, pp. 410-417.

53

All Rights Reserved, Copyright Mikio Aoyama, 2011



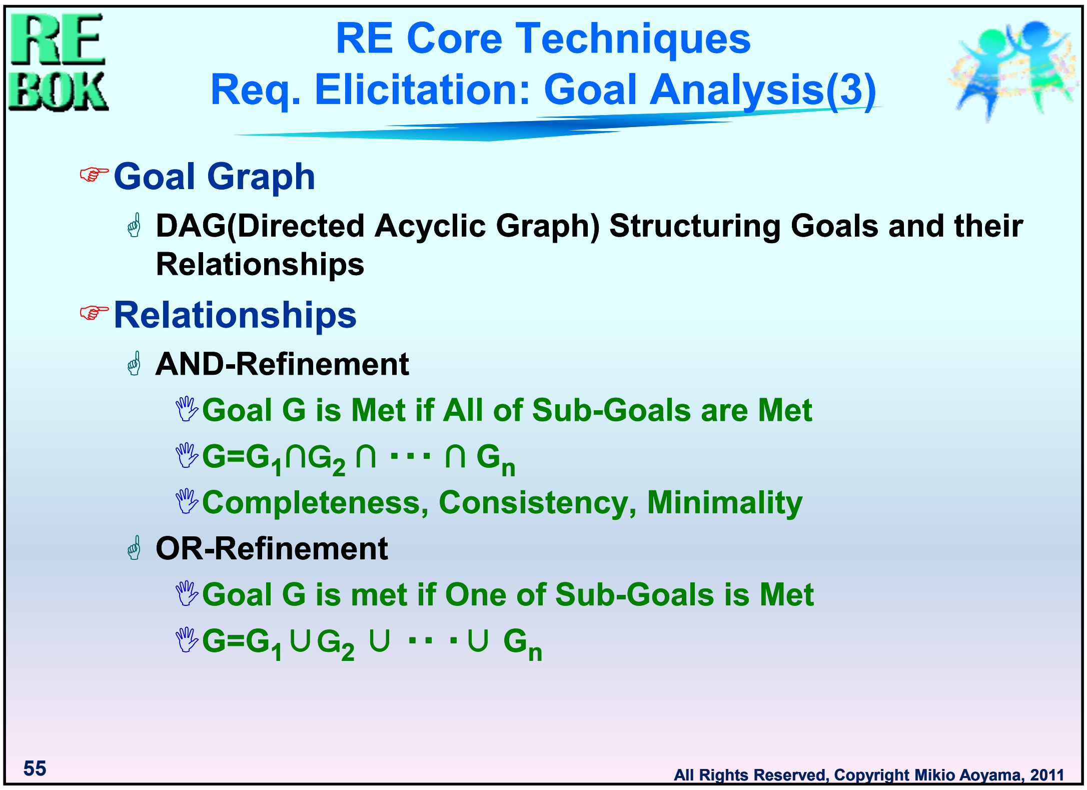
RE BOK

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

54

All Rights Reserved, Copyright Mikio Aoyama, 2011

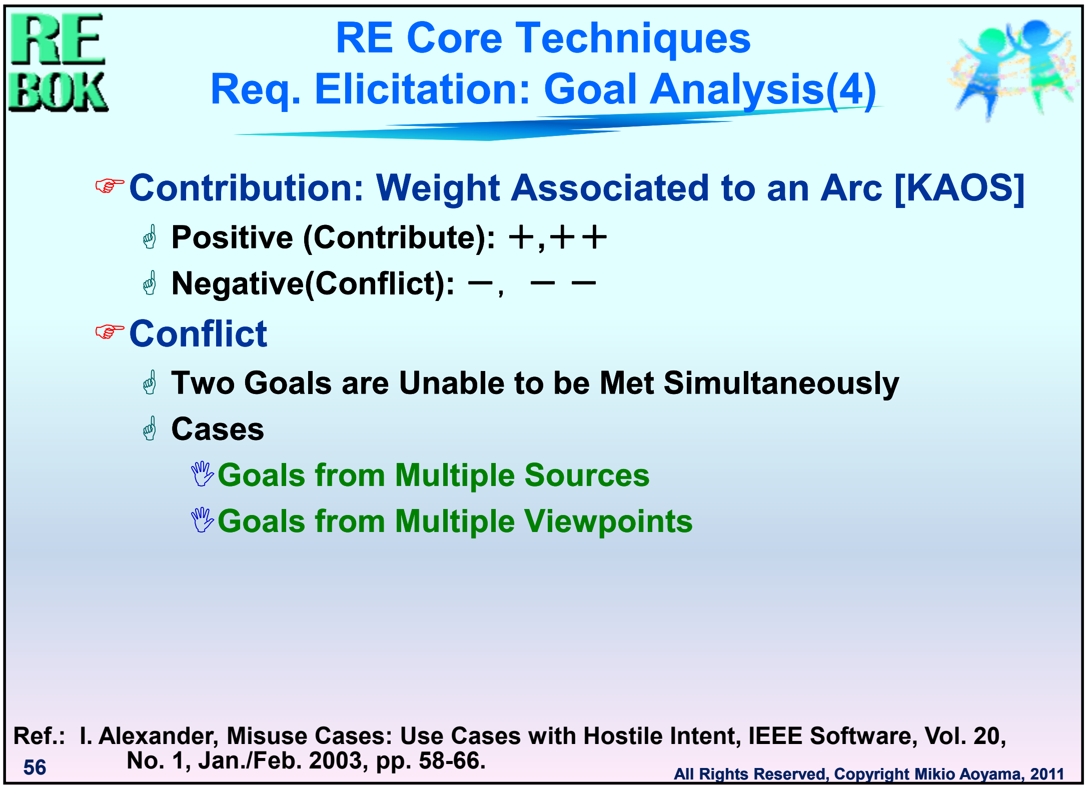


RE Core Techniques
Req. Elicitation: Goal Analysis(3)

- 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
    - $G=G_1 \cup G_2 \cup \cdots \cup G_n$

55

All Rights Reserved, Copyright Mikio Aoyama, 2011



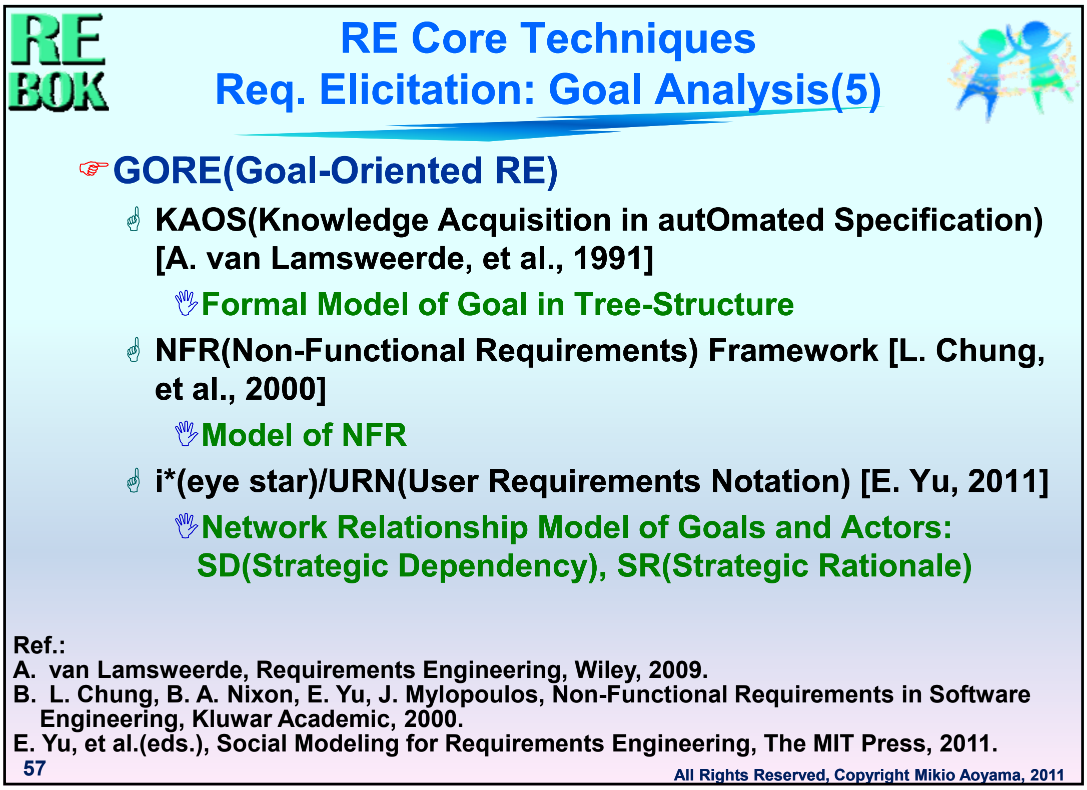
RE Core Techniques
Req. Elicitation: Goal Analysis(4)

- 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

Ref.: I. Alexander, Misuse Cases: Use Cases with Hostile Intent, IEEE Software, Vol. 20, No. 1, Jan./Feb. 2003, pp. 58-66.

56

All Rights Reserved, Copyright Mikio Aoyama, 2011

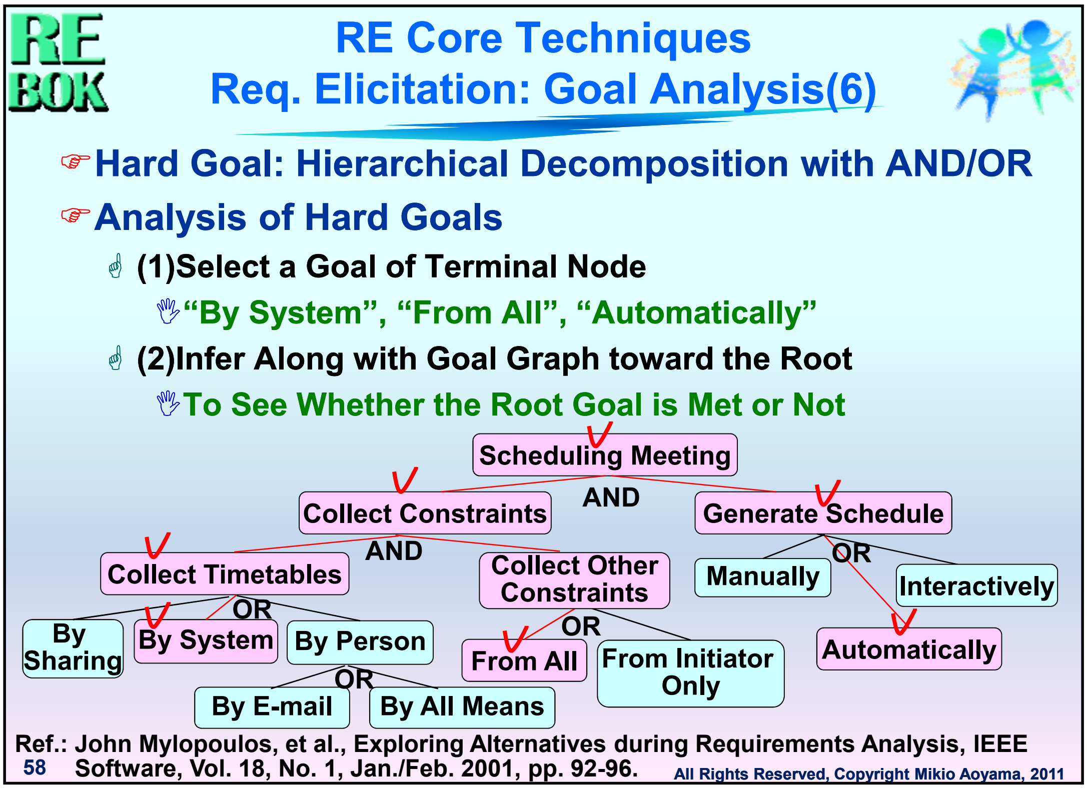


RE Core Techniques
Req. Elicitation: Goal Analysis(5)

- GORE(Goal-Oriented RE)
  - KAOS(Knowledge Acquisition in autOmated Specification) [A. van Lamsweerde, et al., 1991]
    - Formal Model of Goal in Tree-Structure
  - NFR(Non-Functional Requirements) Framework [L. Chung, et al., 2000]
    - 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)

Ref.:
A. van Lamsweerde, Requirements Engineering, Wiley, 2009.
B. L. Chung, B. A. Nixon, E. Yu, J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwar Academic, 2000.
E. Yu, et al.(eds.), Social Modeling for Requirements Engineering, The MIT Press, 2011.

57 All Rights Reserved, Copyright Mikio Aoyama, 2011

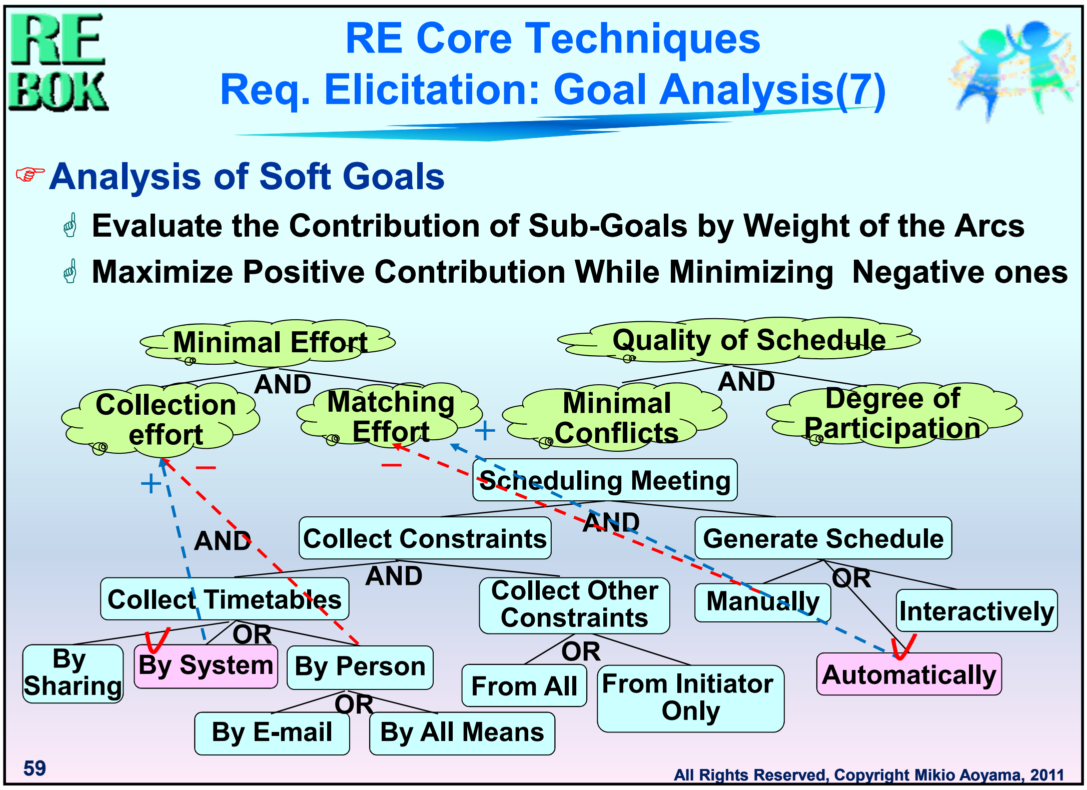


RE Core Techniques
Req. Elicitation: Goal Analysis(6)

- 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

Ref.: John Mylopoulos, et al., Exploring Alternatives during Requirements Analysis, IEEE Software, Vol. 18, No. 1, Jan./Feb. 2001, pp. 92-96.

58 All Rights Reserved, Copyright Mikio Aoyama, 2011

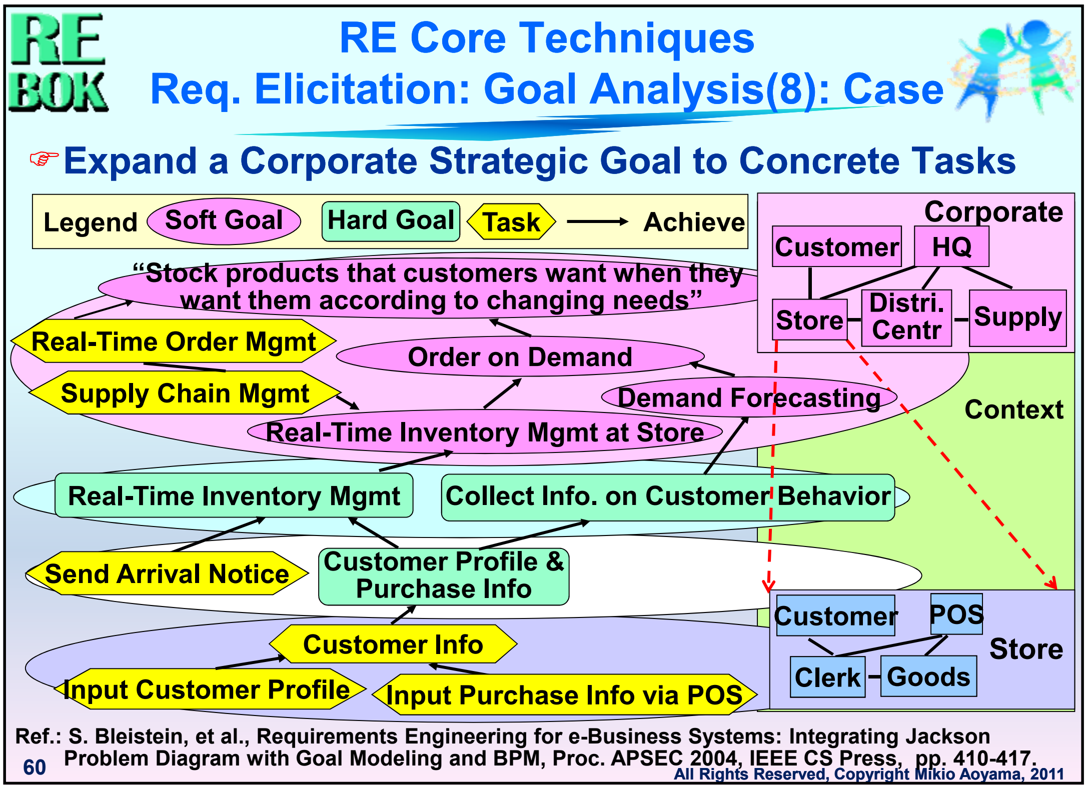


RE Core Techniques
Req. Elicitation: Goal Analysis(7)

Analysis of Soft Goals

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

59

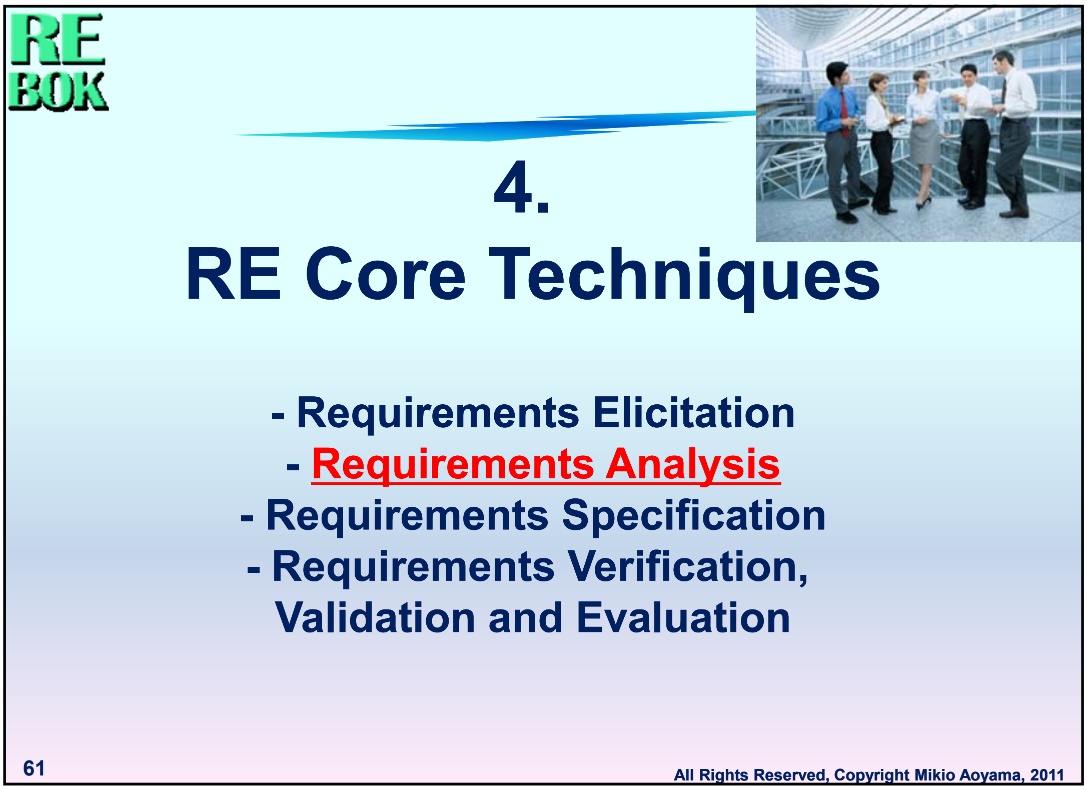


RE Core Techniques
Req. Elicitation: Goal Analysis(8): Case

Expand a Corporate Strategic Goal to Concrete Tasks

Ref.: S. Bleistein, et al., Requirements Engineering for e-Business Systems: Integrating Jackson Problem Diagram with Goal Modeling and BPM, Proc. APSEC 2004, IEEE CS Press, pp. 410-417.

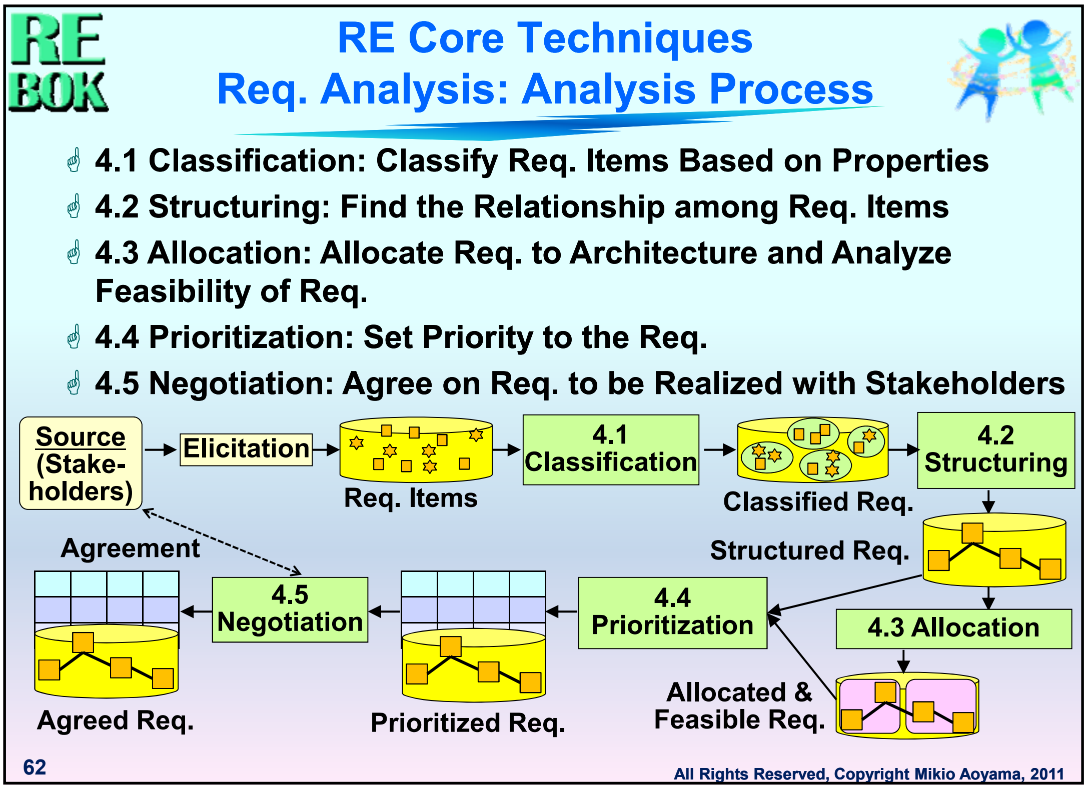
60



4.
RE Core Techniques

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

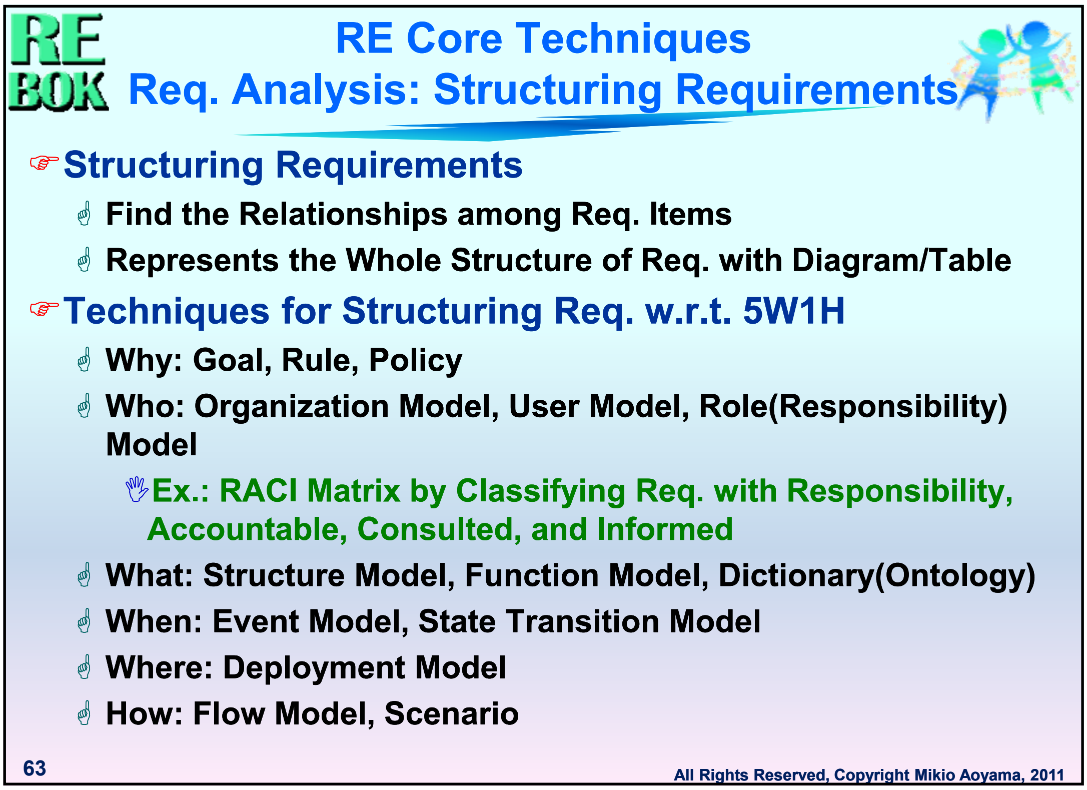
61



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

62

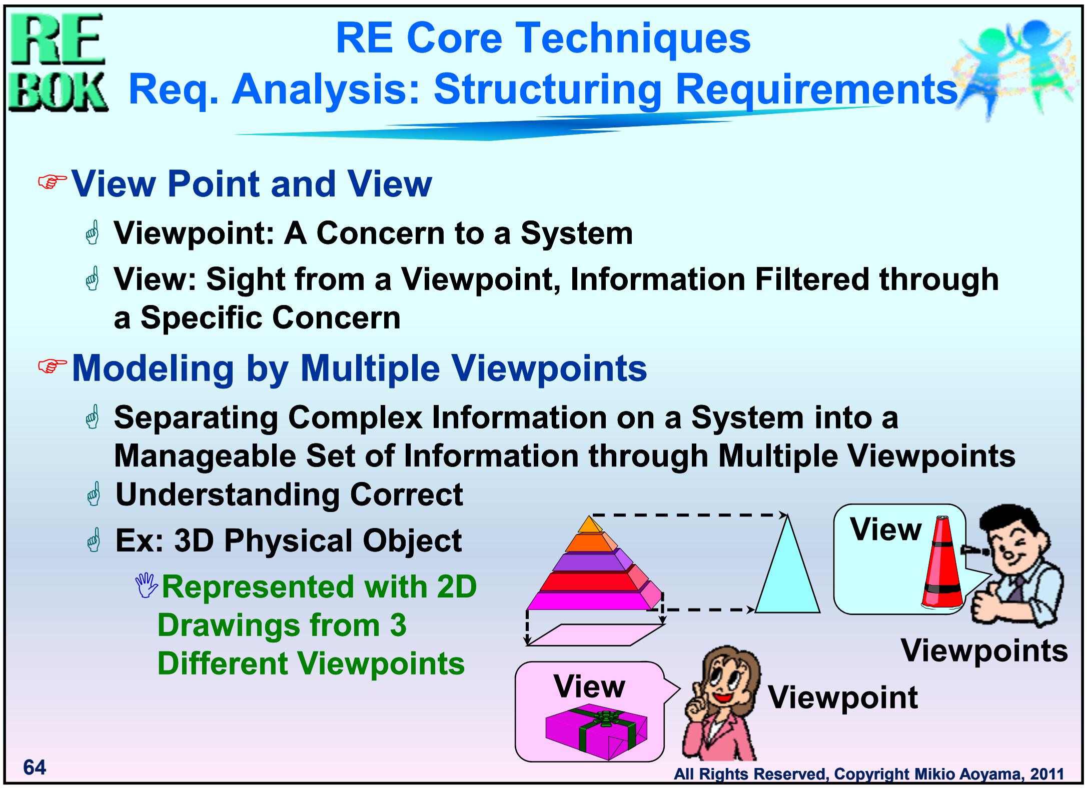


RE Core Techniques
Req. Analysis: Structuring Requirements

- 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) Model
    - 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

63

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE Core Techniques
Req. Analysis: Structuring Requirements

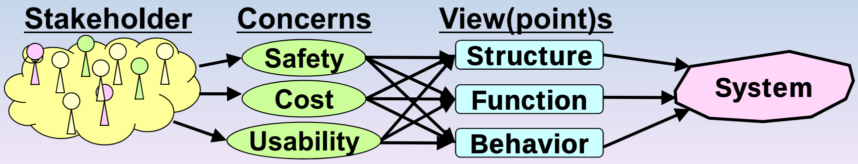
- View Point and View
  - Viewpoint: A Concern to a System
  - View: Sight from a Viewpoint, Information Filtered through a Specific Concern
- Modeling by Multiple Viewpoints
  - Separating Complex Information on a System into a Manageable Set of Information through Multiple Viewpoints
  - Understanding Correct
  - Ex: 3D Physical Object
    - Represented with 2D Drawings from 3 Different Viewpoints

64

All Rights Reserved, Copyright Mikio Aoyama, 2011

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)



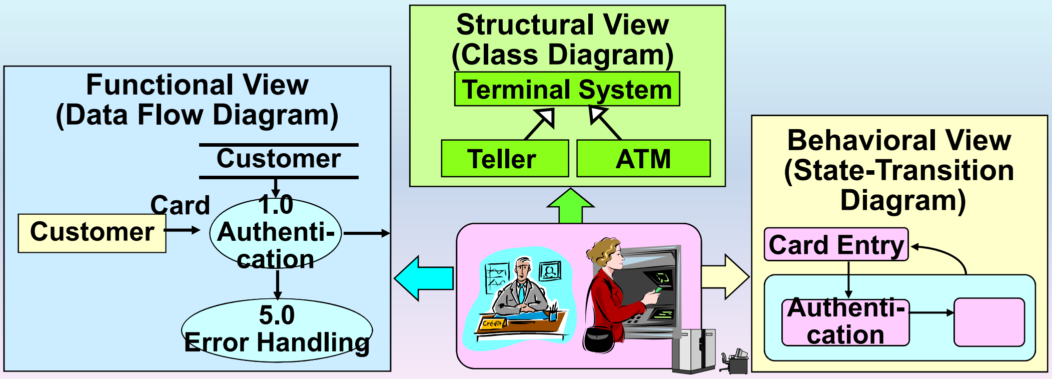
Ref.: P. Lago, et al., Software Architecture: Framing Stakeholders' Concerns, IEEE Software, Vol. 27, No. 6, Nov./Dec., 2010, pp. 20-24.

65

All Rights Reserved, Copyright Mikio Aoyama, 2011

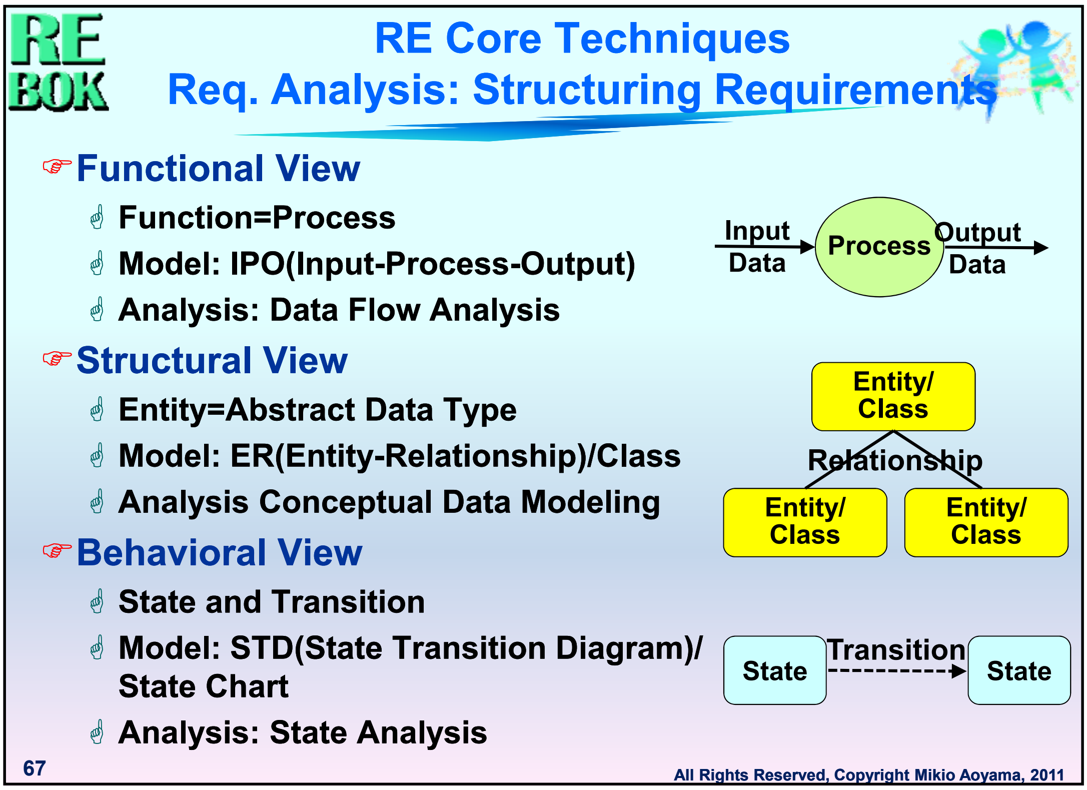
RE Core Techniques
Req. Analysis: Structuring Requirements

- 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



66

All Rights Reserved, Copyright Mikio Aoyama, 2011

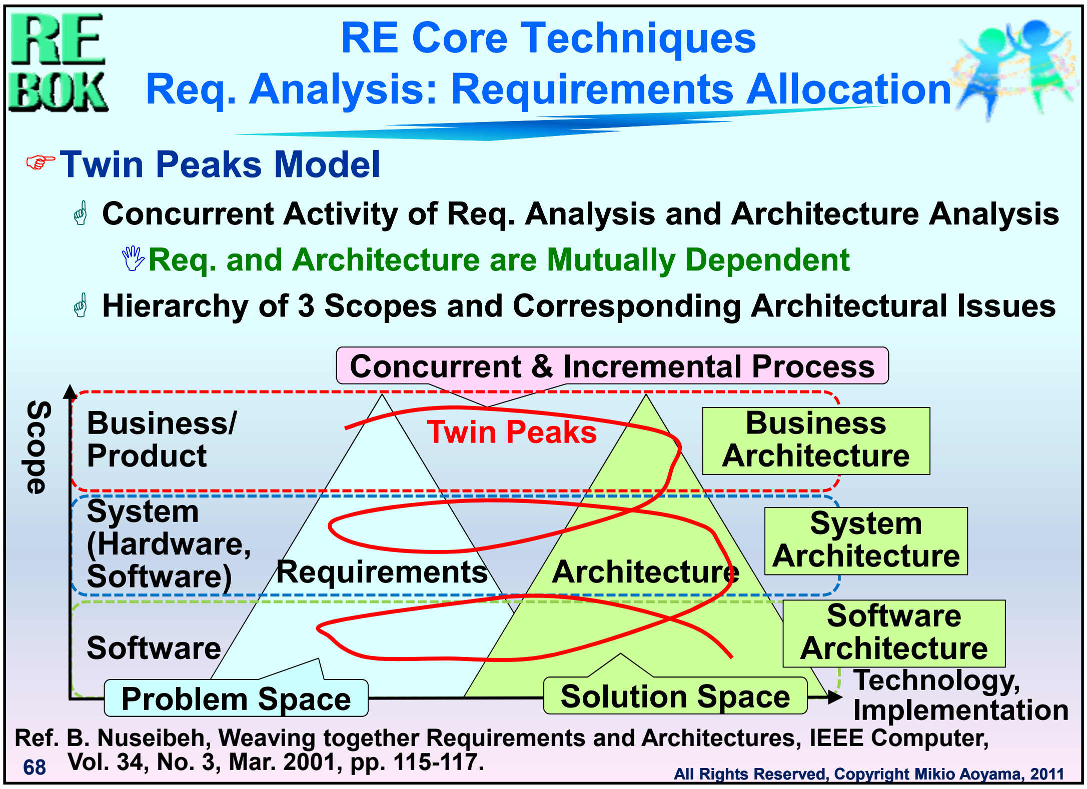


REBOK

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

67



REBOK

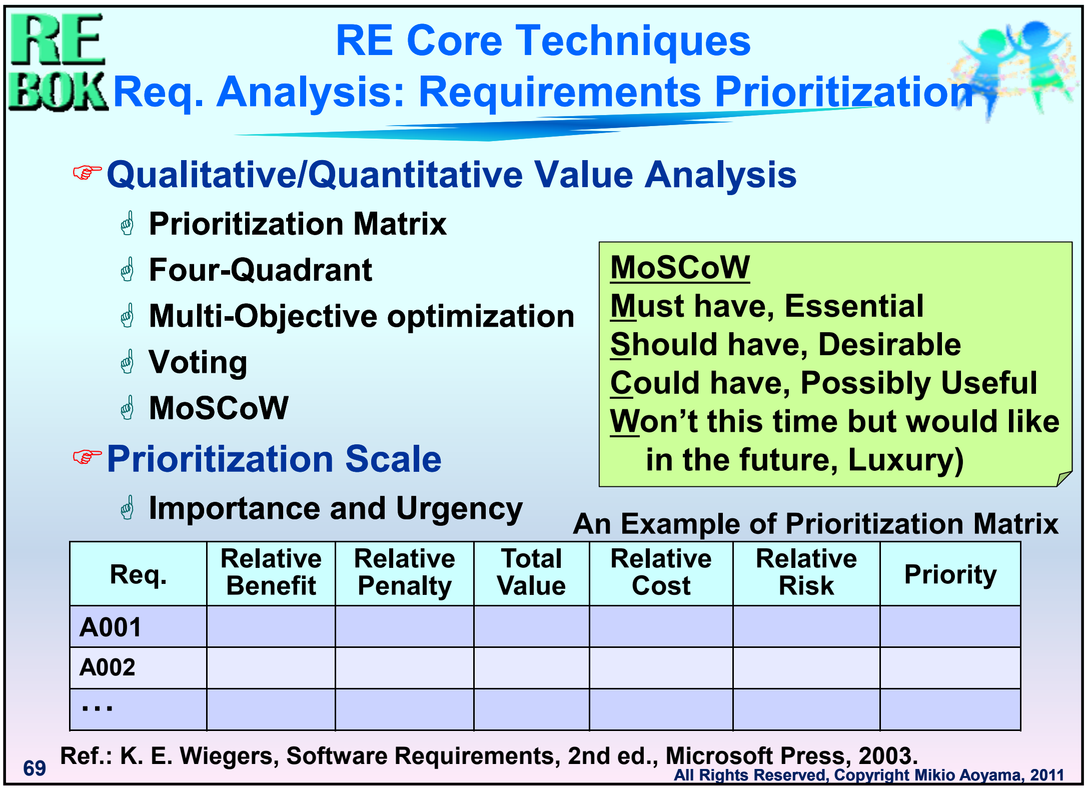
RE Core Techniques
Req. Analysis: Requirements Allocation

- 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

Concurrent & Incremental Process
Scope
Business/ Product
Twin Peaks
Business Architecture
System (Hardware, Software)
Requirements
Architecture
System Architecture
Software
Software Architecture
Problem Space
Solution Space
Technology, Implementation

Ref. B. Nuseibeh, Weaving together Requirements and Architectures, IEEE Computer, Vol. 34, No. 3, Mar. 2001, pp. 115-117.

68



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

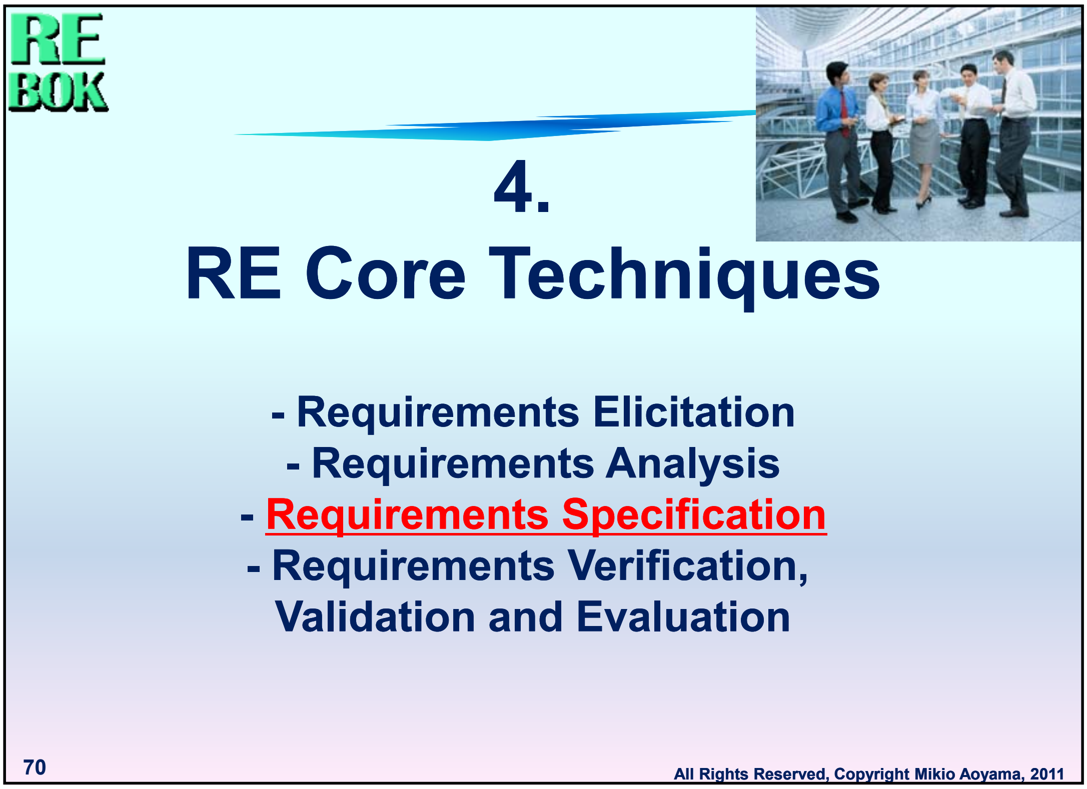
MoSCoW
Must have, Essential
Should have, Desirable
Could have, Possibly Useful
Won't this time but would like in the future, Luxury)

An Example of Prioritization Matrix

Req.	Relative Benefit	Relative Penalty	Total Value	Relative Cost	Relative Risk	Priority
A001						
A002						
...						

Ref.: K. E. Wiegers, Software Requirements, 2nd ed., Microsoft Press, 2003.

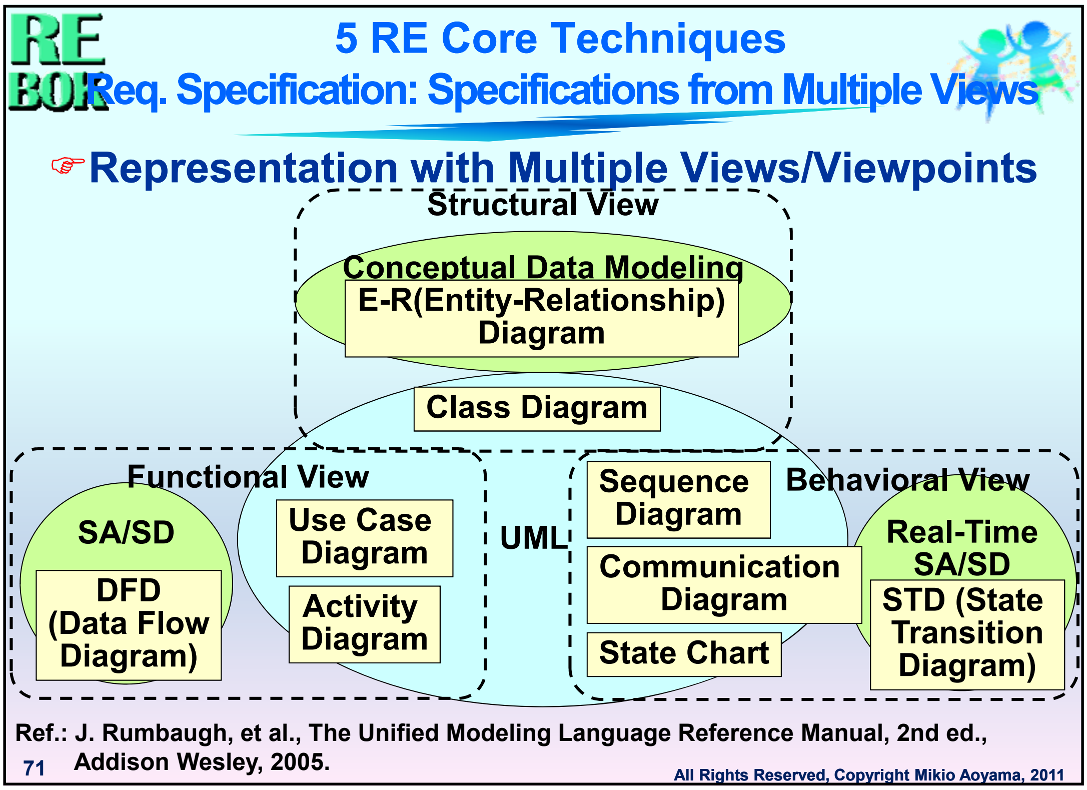
69



4.
RE Core Techniques

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

70



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

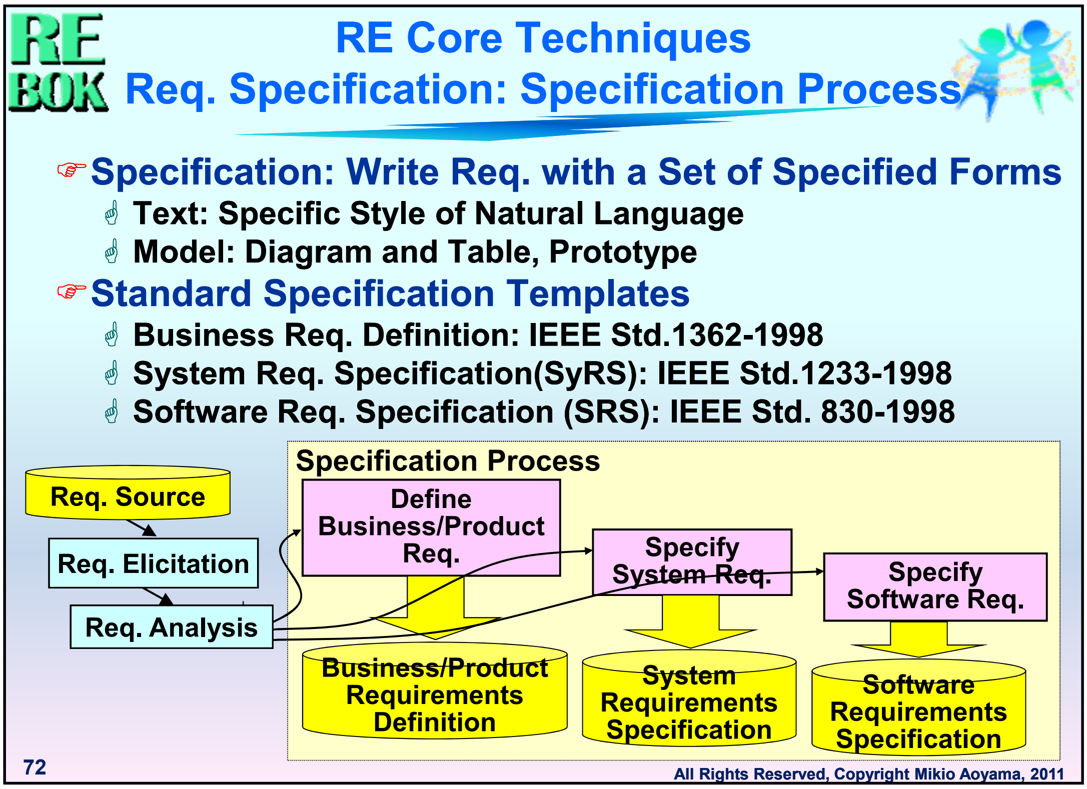
Behavioral View

Real-Time SA/SD

STD (State Transition Diagram)

Ref.: J. Rumbaugh, et al., The Unified Modeling Language Reference Manual, 2nd ed., Addison Wesley, 2005.

71



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

- Business Req. Definition: IEEE Std.1362-1998
- System Req. Specification(SyRS): IEEE Std.1233-1998
- Software Req. Specification (SRS): IEEE Std. 830-1998

Req. Source

Req. Elicitation

Req. Analysis

Specification Process

Define Business/Product Req.

Specify System Req.

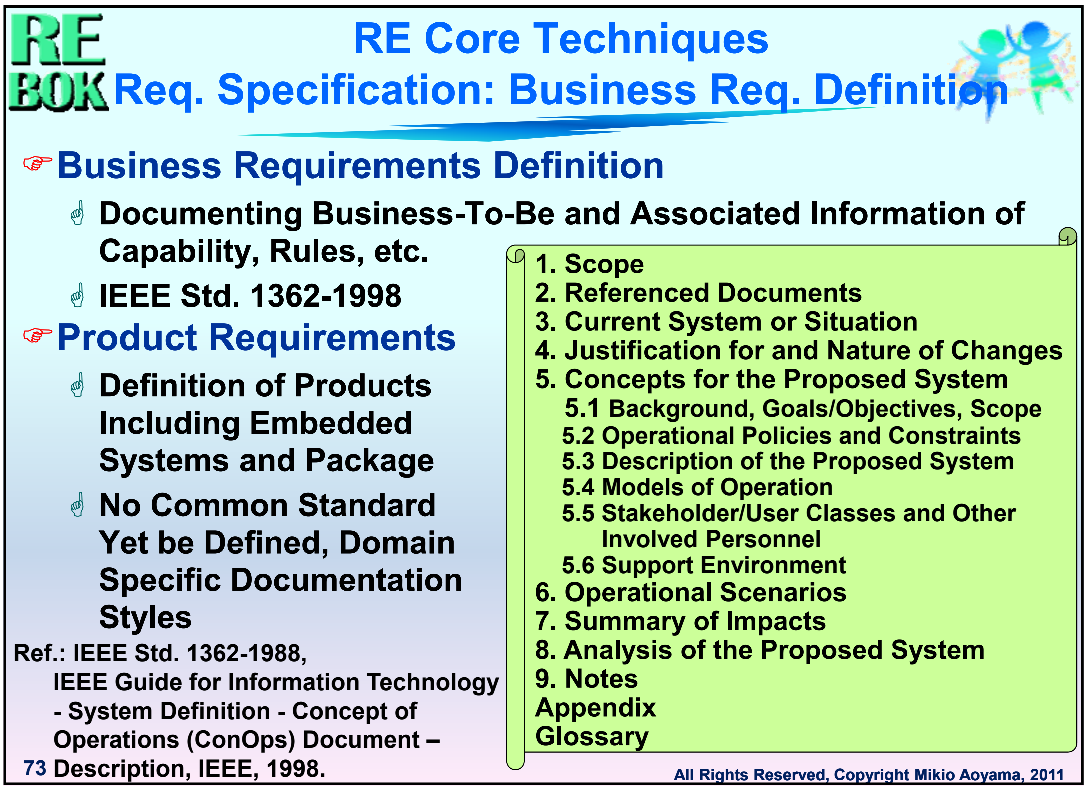
Specify Software Req.

Business/Product Requirements Definition

System Requirements Specification

Software Requirements Specification

72



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

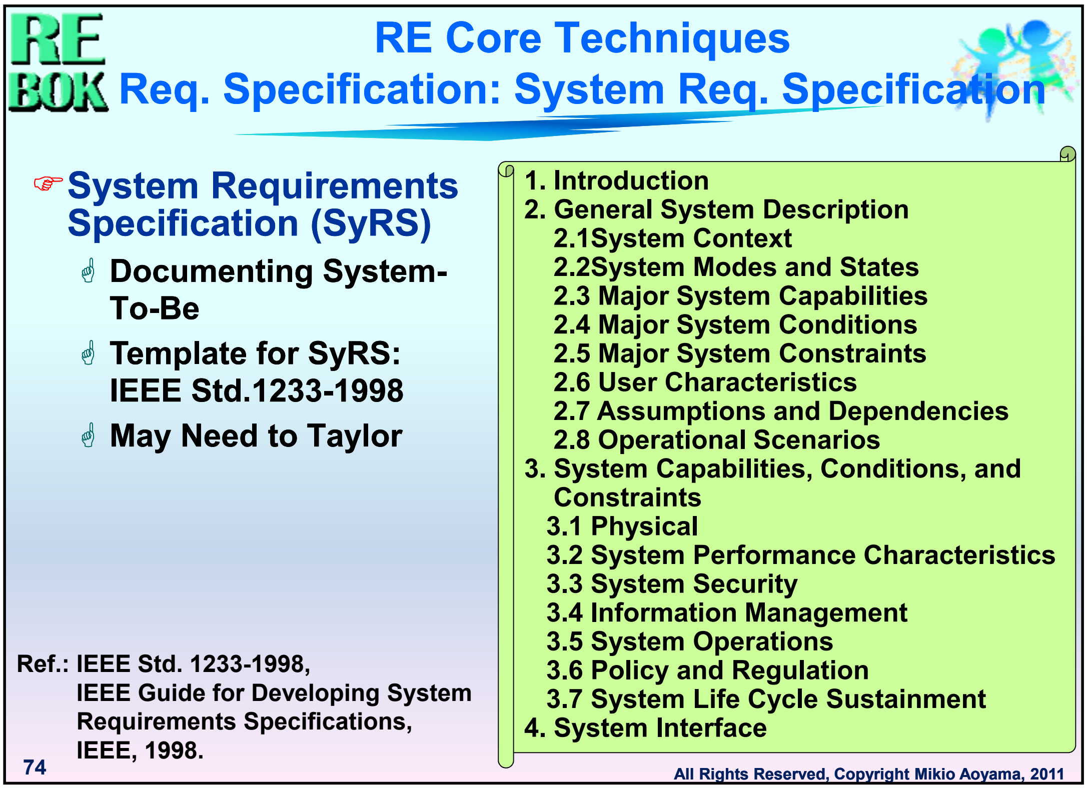
1. Scope
2. Referenced Documents
3. Current System or Situation
4. Justification for and Nature of Changes
5. Concepts for the Proposed System
   5.1 Background, Goals/Objectives, Scope
   5.2 Operational Policies and Constraints
   5.3 Description of the Proposed System
   5.4 Models of Operation
   5.5 Stakeholder/User Classes and Other Involved Personnel
   5.6 Support Environment
6. Operational Scenarios
7. Summary of Impacts
8. Analysis of the Proposed System
9. Notes

Appendix
Glossary

Ref.: IEEE Std. 1362-1988, IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document – Description, IEEE, 1998.

73

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE Core Techniques
Req. Specification: System Req. Specification

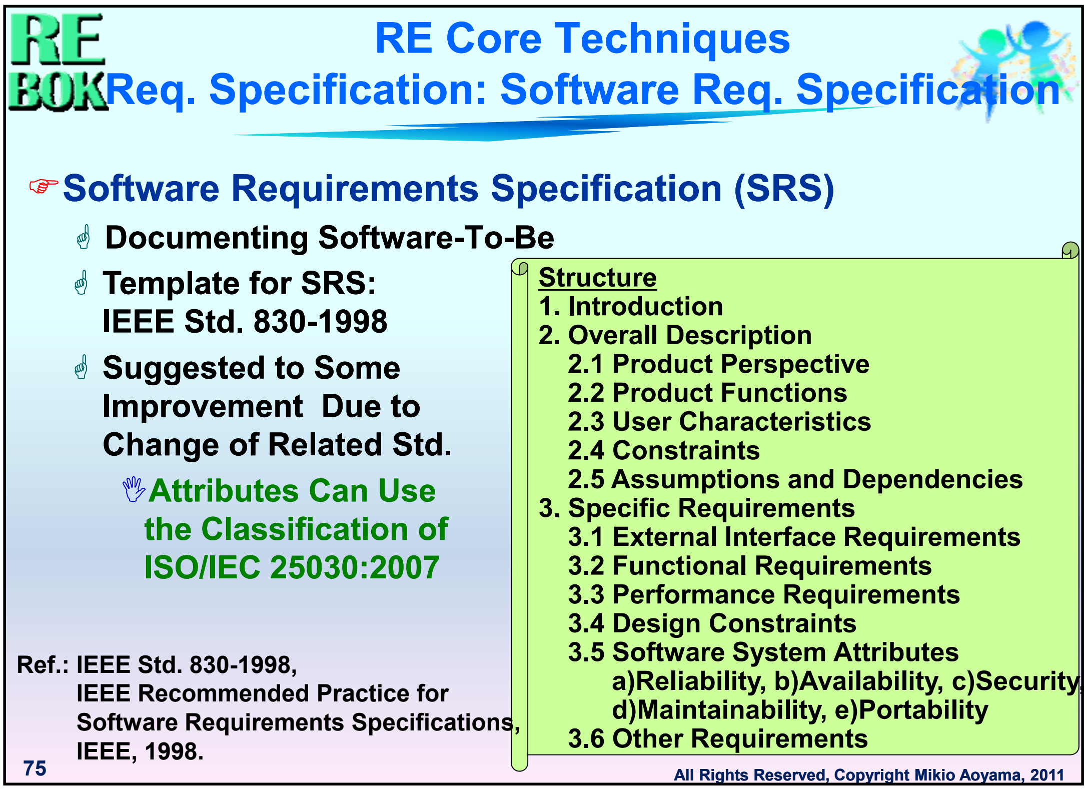
- **System Requirements Specification (SyRS)**
  - Documenting System-To-Be
  - Template for SyRS: IEEE Std.1233-1998
  - May Need to Taylor

1. Introduction
2. General System Description
   2.1System Context
   2.2System Modes and States
   2.3 Major System Capabilities
   2.4 Major System Conditions
   2.5 Major System Constraints
   2.6 User Characteristics
   2.7 Assumptions and Dependencies
   2.8 Operational Scenarios
3. System Capabilities, Conditions, and Constraints
   3.1 Physical
   3.2 System Performance Characteristics
   3.3 System Security
   3.4 Information Management
   3.5 System Operations
   3.6 Policy and Regulation
   3.7 System Life Cycle Sustainment
4. System Interface

Ref.: IEEE Std. 1233-1998, IEEE Guide for Developing System Requirements Specifications, IEEE, 1998.

74

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE BOK

RE Core Techniques
Req. Specification: Software Req. Specification

☞Software Requirements Specification (SRS)

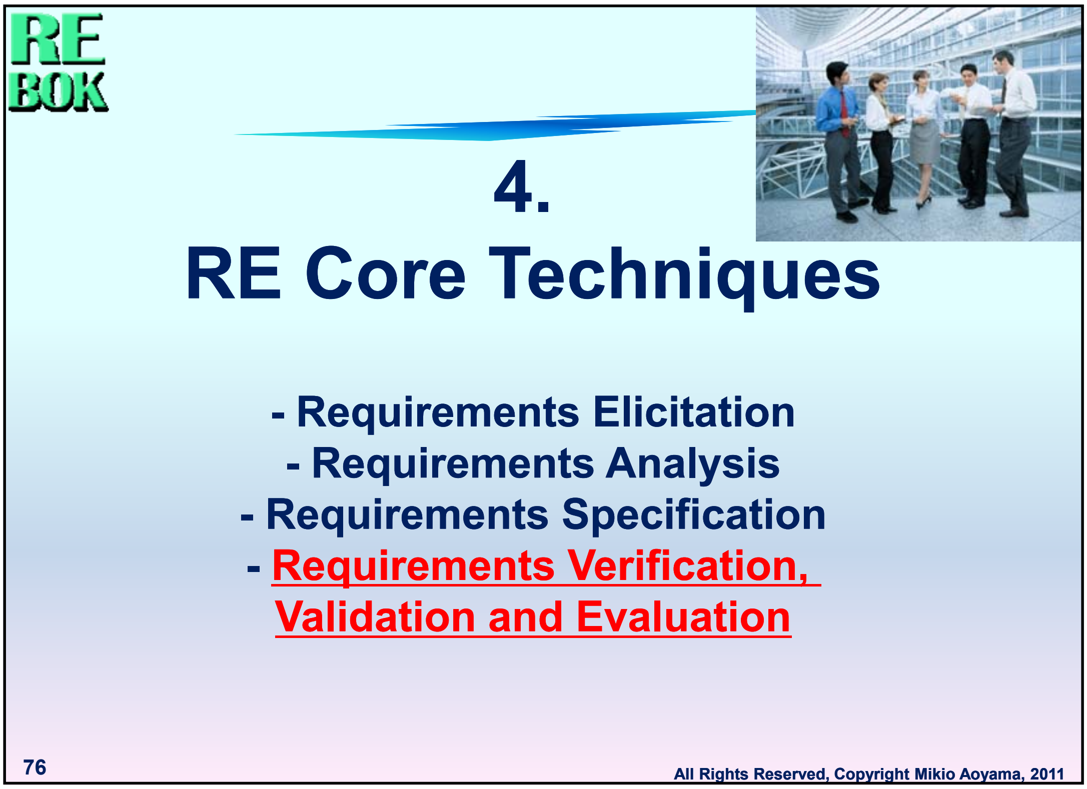
- Documenting Software-To-Be
- Template for SRS: IEEE Std. 830-1998
- Suggested to Some Improvement Due to Change of Related Std.
  - Attributes Can Use the Classification of ISO/IEC 25030:2007

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

Ref.: IEEE Std. 830-1998,
IEEE Recommended Practice for
Software Requirements Specifications,
IEEE, 1998.

75

All Rights Reserved, Copyright Mikio Aoyama, 2011



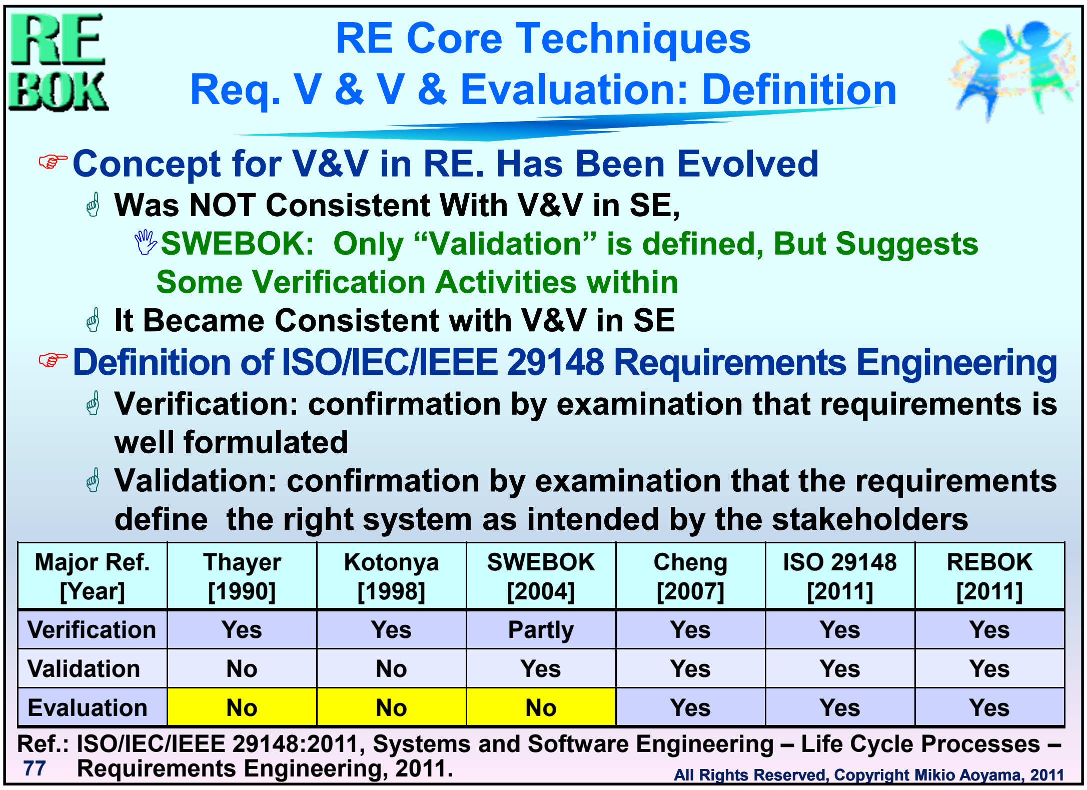
RE BOK

4.
RE Core Techniques

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

76

All Rights Reserved, Copyright Mikio Aoyama, 2011



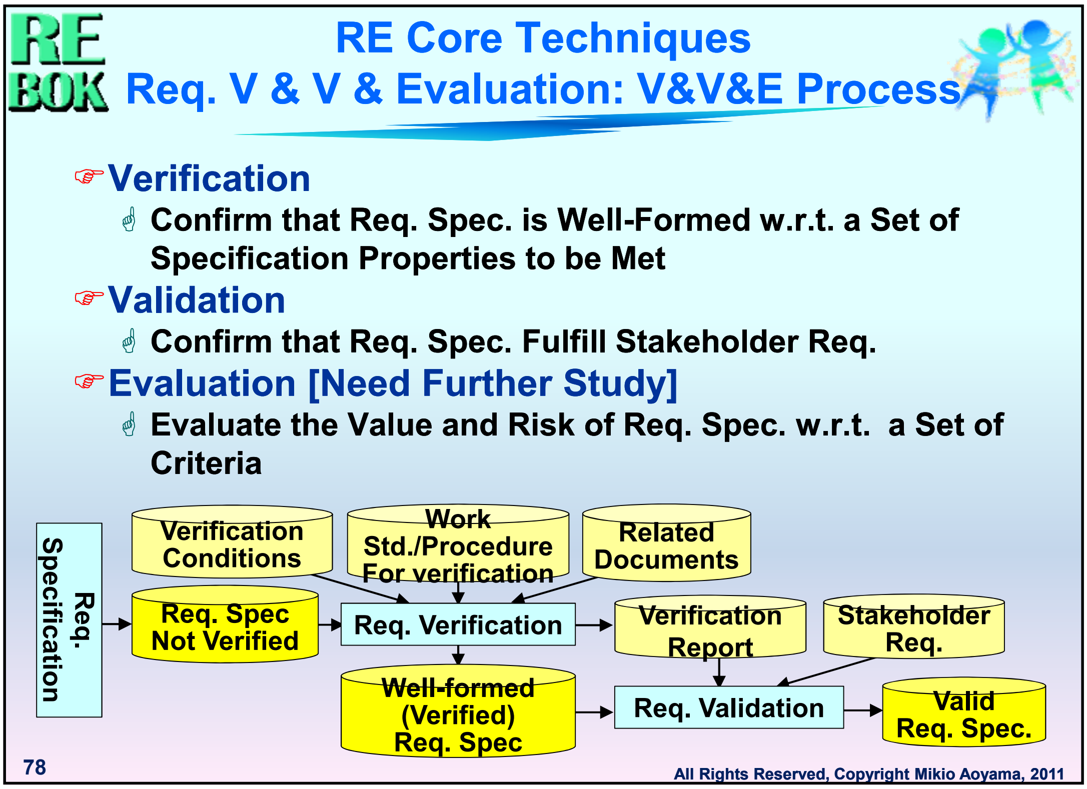
RE Core Techniques
Req. V & V & Evaluation: Definition

- 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

Major Ref. [Year]	Thayer [1990]	Kotonya [1998]	SWEBOK [2004]	Cheng [2007]	ISO 29148 [2011]	REBOK [2011]
Verification	Yes	Yes	Partly	Yes	Yes	Yes
Validation	No	No	Yes	Yes	Yes	Yes
Evaluation	No	No	No	Yes	Yes	Yes

Ref.: ISO/IEC/IEEE 29148:2011, Systems and Software Engineering – Life Cycle Processes – Requirements Engineering, 2011.

77



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

Req. Specification

Verification Conditions

Work Std./Procedure For verification

Related Documents

Req. Spec Not Verified

Req. Verification

Verification Report

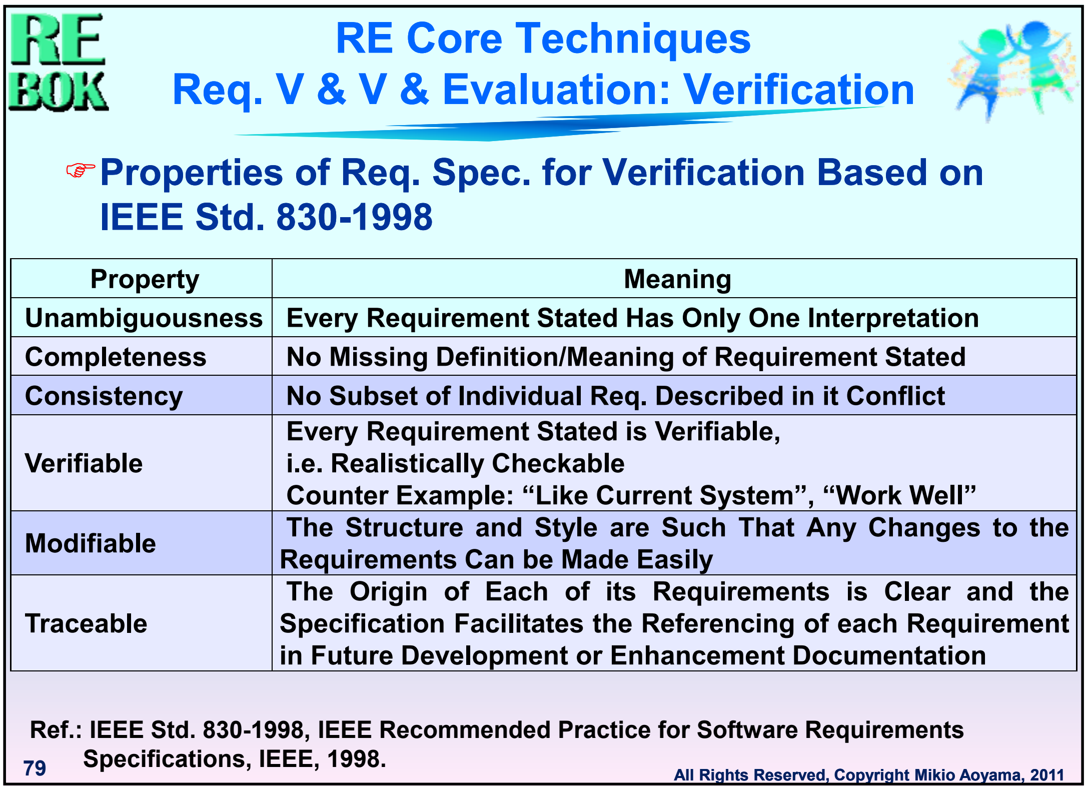
Stakeholder Req.

Well-formed (Verified) Req. Spec

Req. Validation

Valid Req. Spec.

78



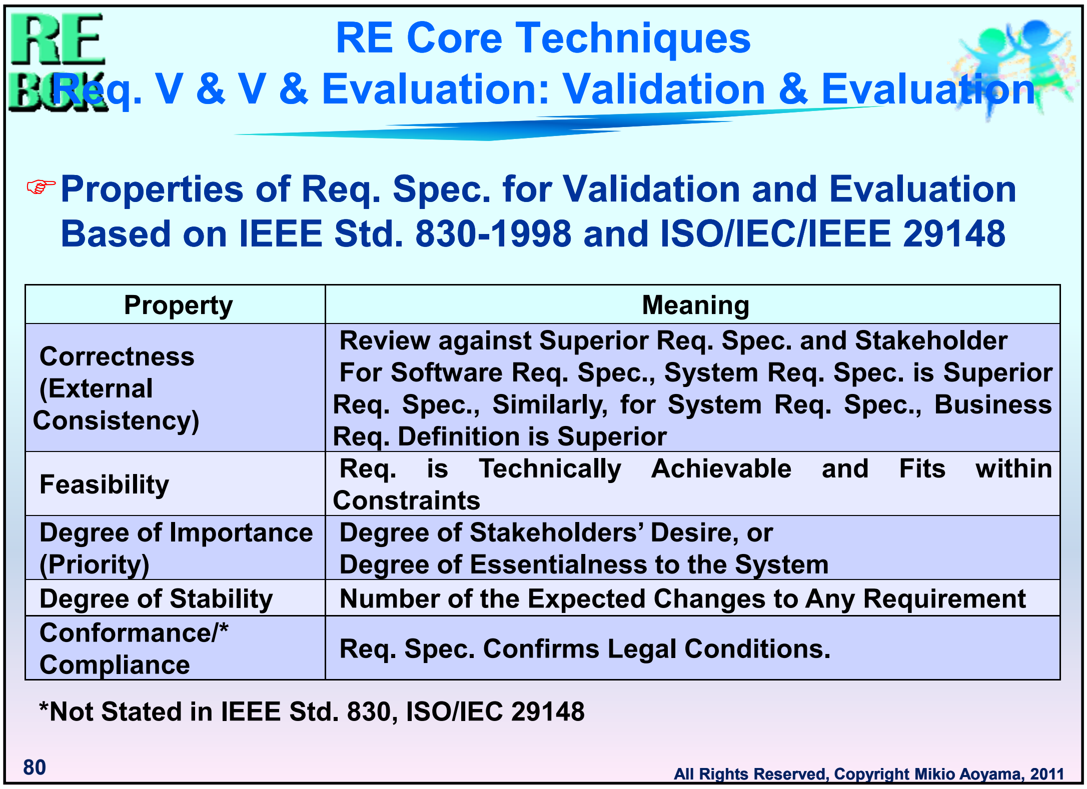
RE Core Techniques
Req. V & V & Evaluation: Verification

☞Properties of Req. Spec. for Verification Based on IEEE Std. 830-1998

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

Ref.: IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE, 1998.

79



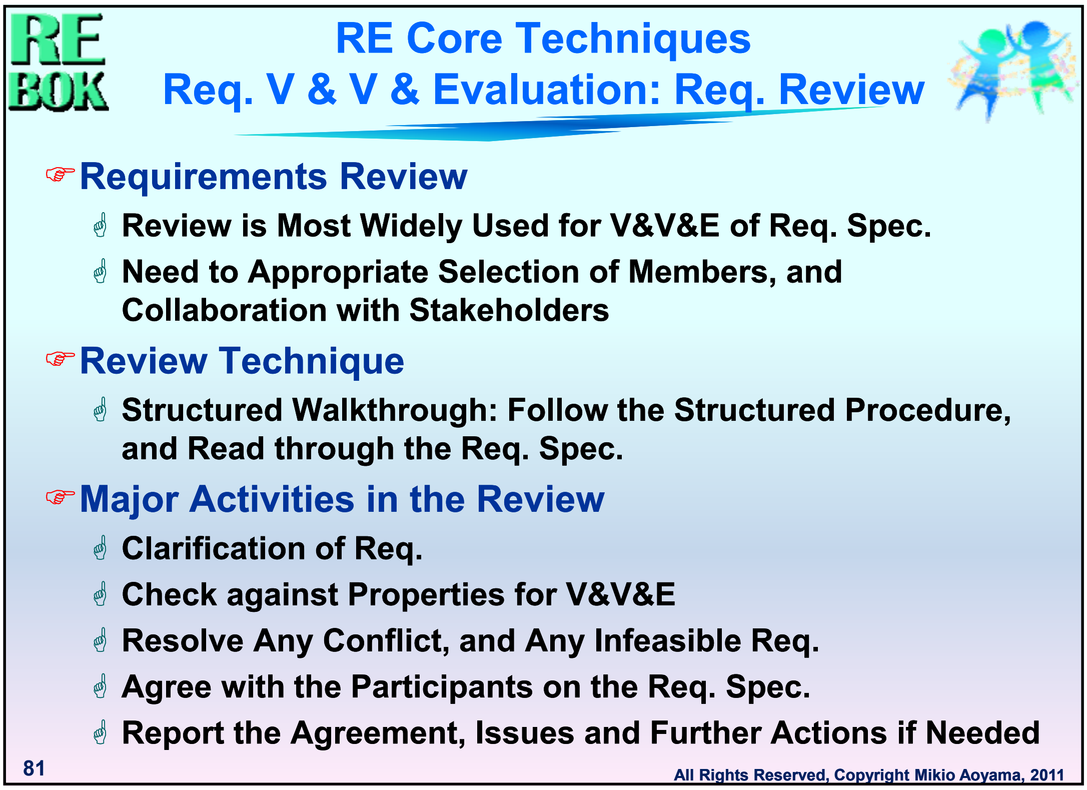
RE Core Techniques
Req. V & V & Evaluation: Validation & Evaluation

☞Properties of Req. Spec. for Validation and Evaluation Based on IEEE Std. 830-1998 and ISO/IEC/IEEE 29148

Property	Meaning
Correctness (External Consistency)	Review against Superior Req. Spec. and Stakeholder For Software Req. Spec., System Req. Spec. is Superior Req. Spec., Similarly, for System Req. Spec., Business Req. Definition is Superior
Feasibility	Req. is Technically Achievable and Fits within Constraints
Degree of Importance (Priority)	Degree of Stakeholders’ Desire, or Degree of Essentialness to the System
Degree of Stability	Number of the Expected Changes to Any Requirement
Conformance/* Compliance	Req. Spec. Confirms Legal Conditions.

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

80



RE Core Techniques
Req. V & V & Evaluation: Req. Review

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

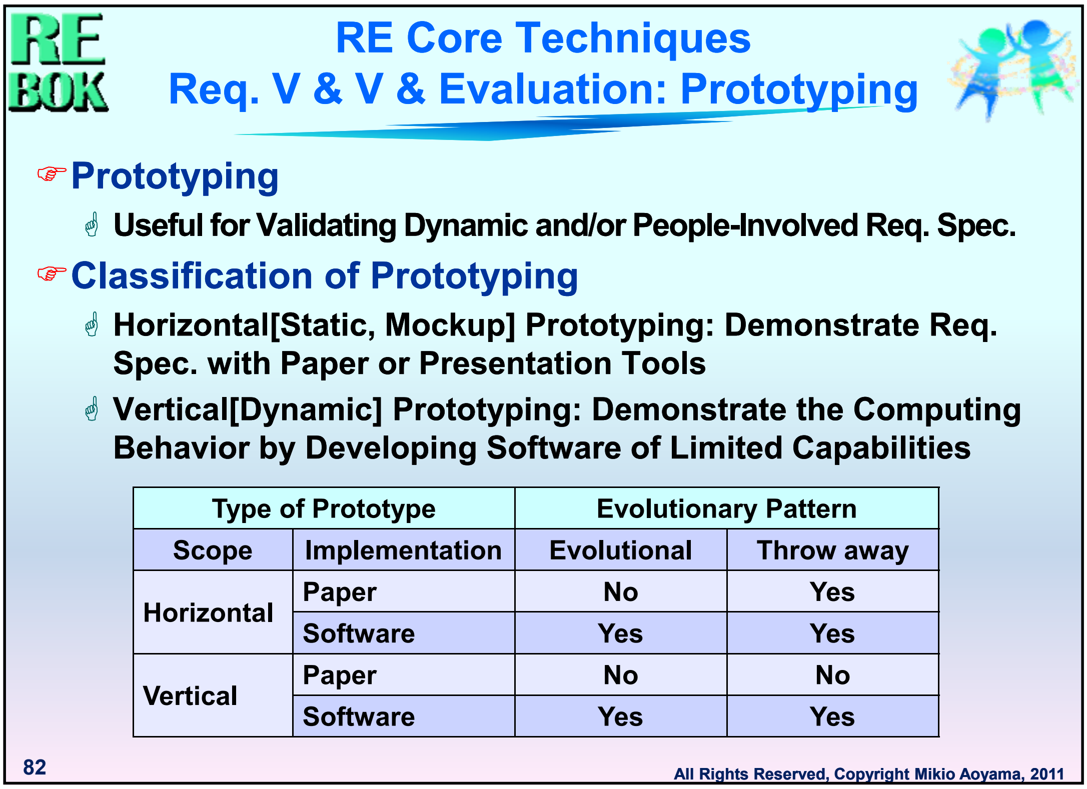
- Structured Walkthrough: Follow the Structured Procedure, and Read through the Req. Spec.

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

81

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE Core Techniques
Req. V & V & Evaluation: Prototyping

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

Type of Prototype		Evolutionary Pattern	
Scope	Implementation	Evolutional	Throw away
Horizontal	Paper	No	Yes
	Software	Yes	Yes
Vertical	Paper	No	No
	Software	Yes	Yes

82

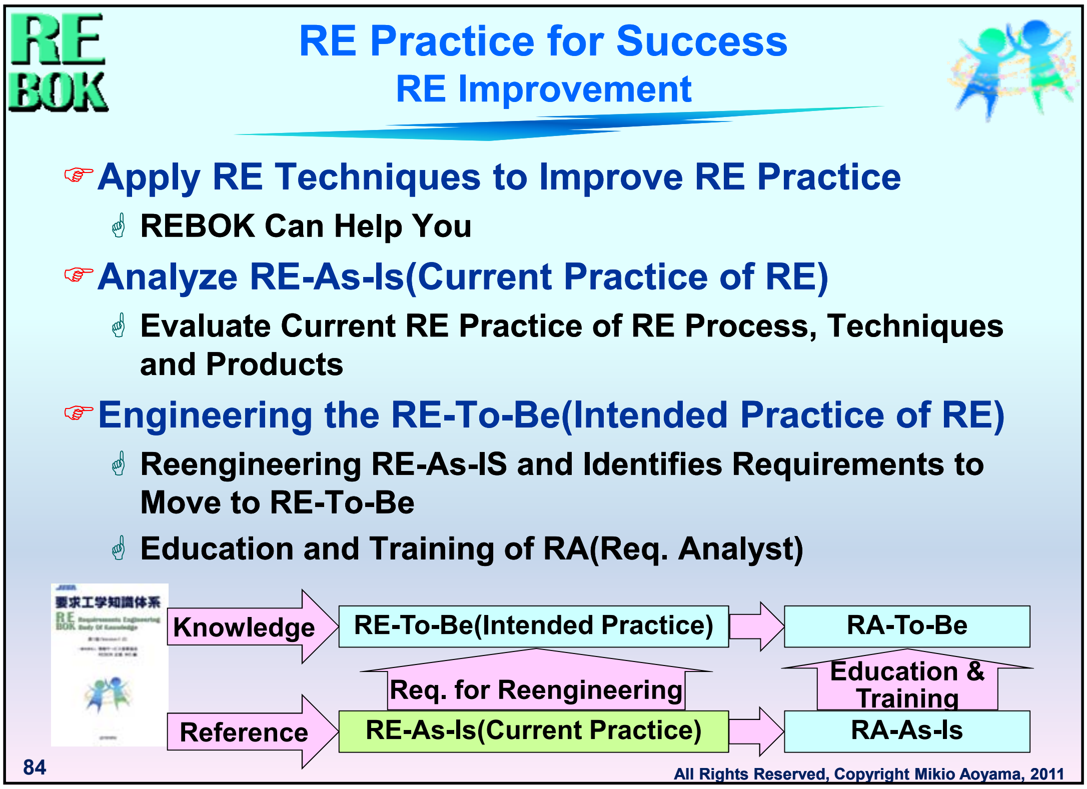
All Rights Reserved, Copyright Mikio Aoyama, 2011

6.

RE Practice for Success

83

All Rights Reserved, Copyright Mikio Aoyama, 2011



RE Practice for Success
RE Improvement

- 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 → RE-To-Be(Intended Practice) → RA-To-Be

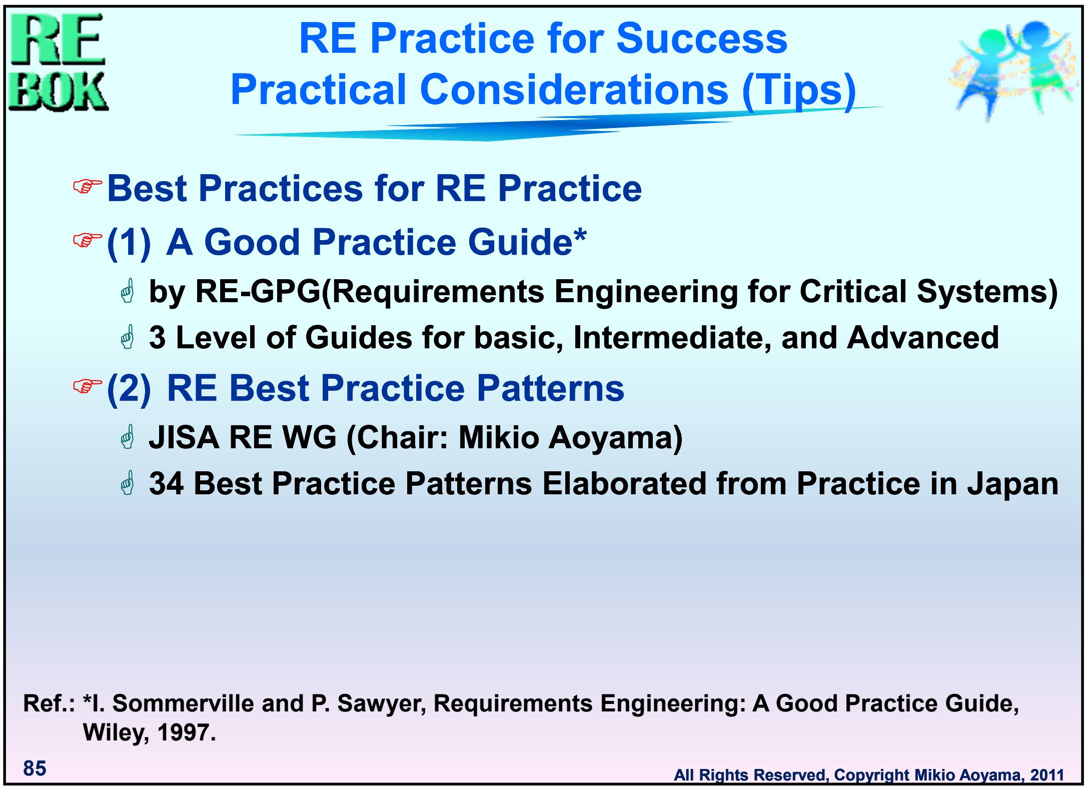
Req. for Reengineering

Education & Training

Reference → RE-As-Is(Current Practice) → RA-As-Is

84

All Rights Reserved, Copyright Mikio Aoyama, 2011

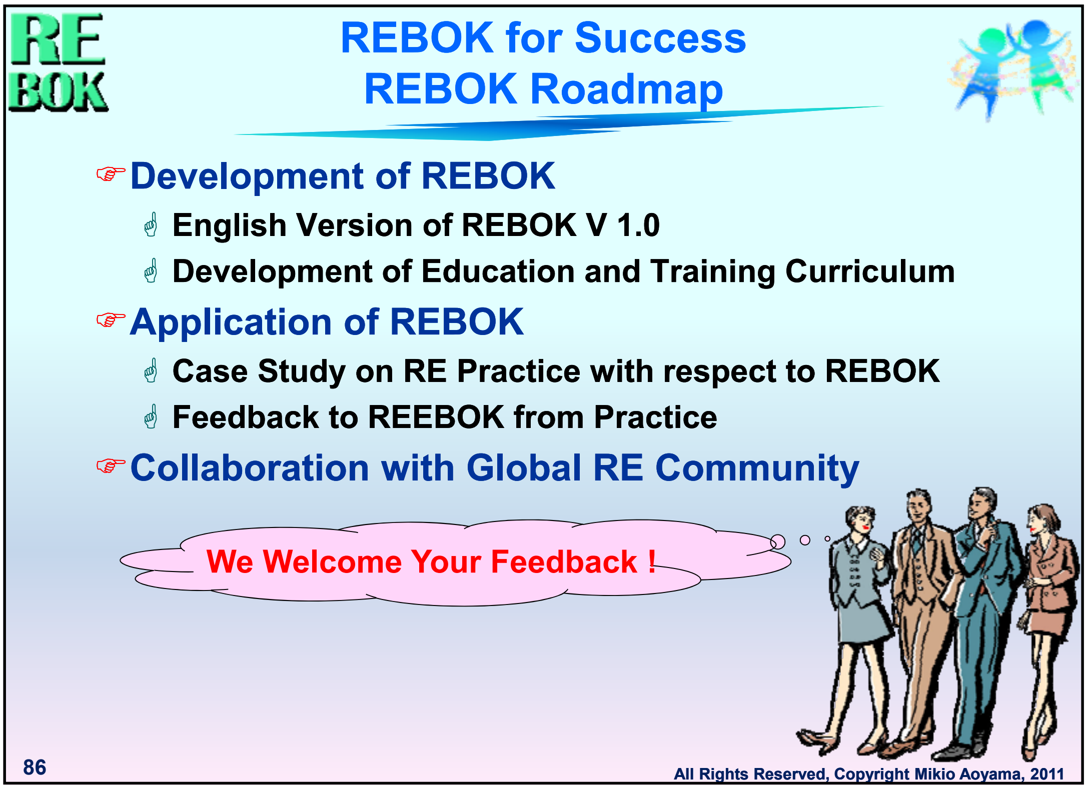


RE Practice for Success
Practical Considerations (Tips)

- ☞ 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

Ref.: *I. Sommerville and P. Sawyer, Requirements Engineering: A Good Practice Guide, Wiley, 1997.

85

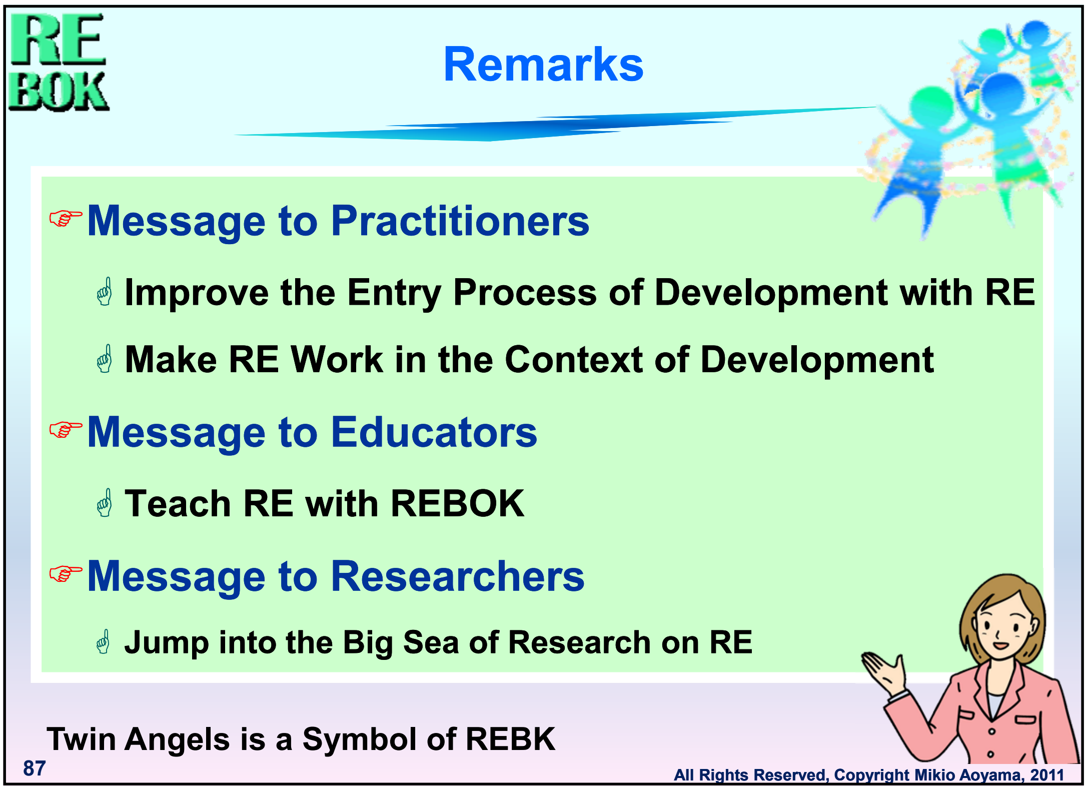


REBOK for Success
REBOK Roadmap

- ☞ 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 !

86



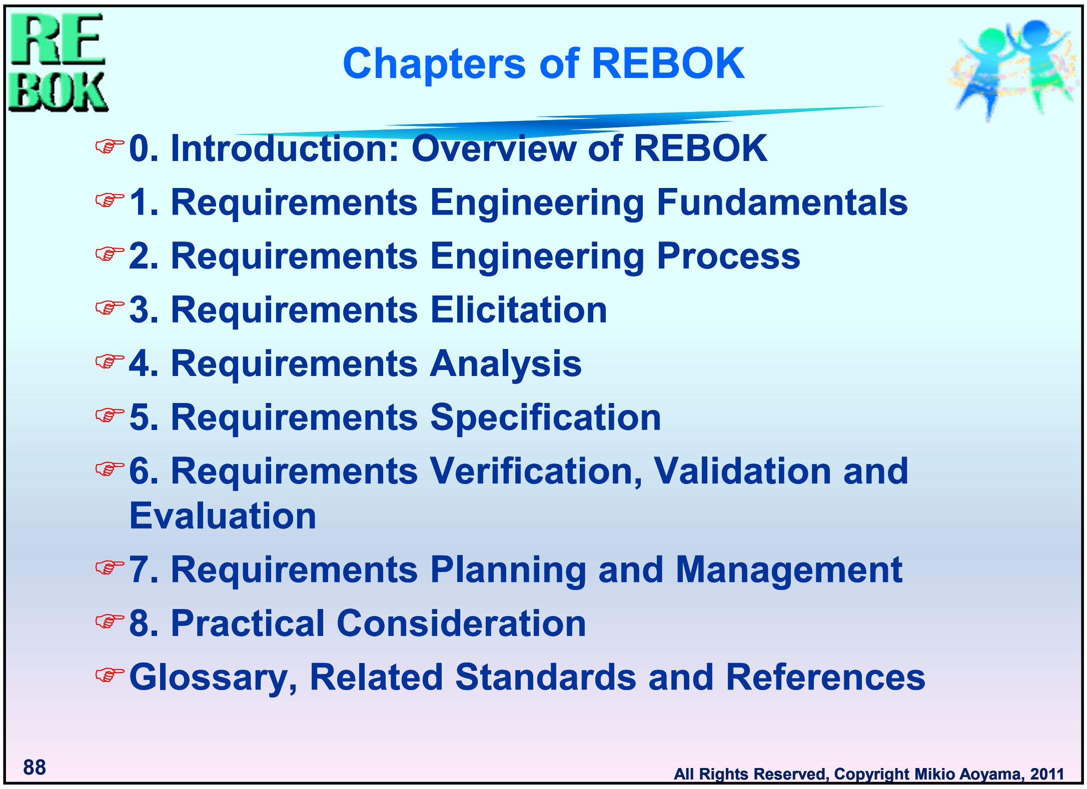
Remarks

- 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 REBK

87

All Rights Reserved, Copyright Mikio Aoyama, 2011

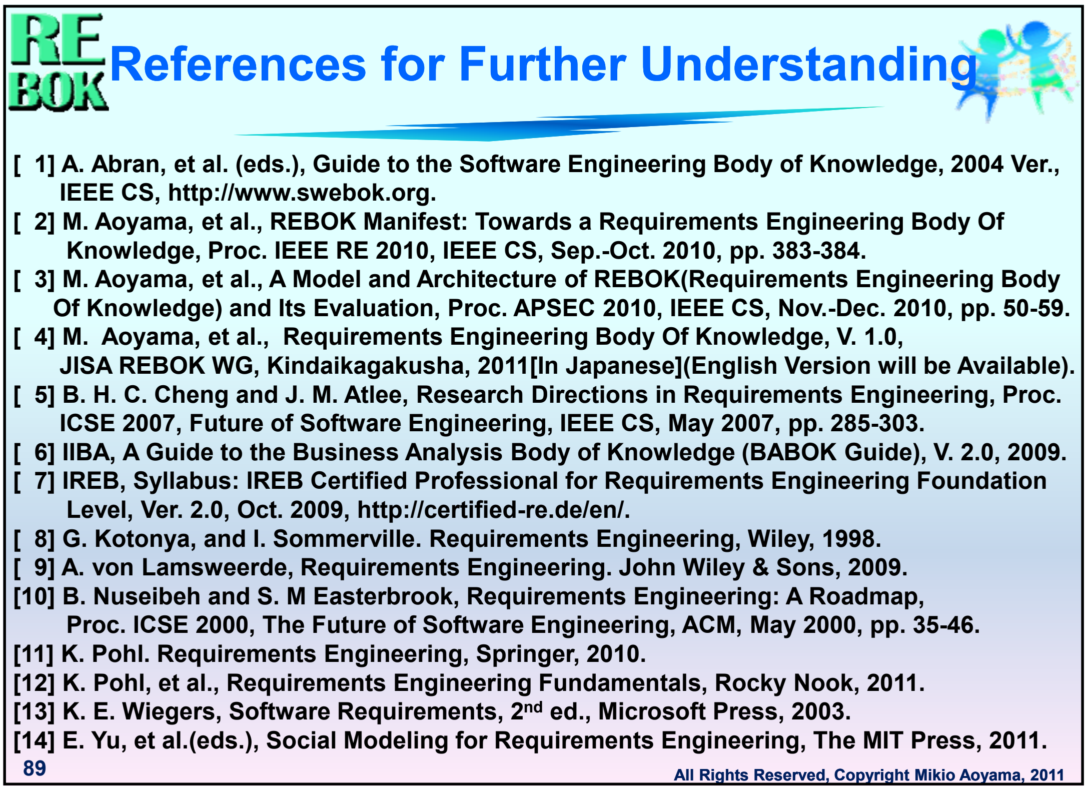


Chapters of REBOK

- 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

88

All Rights Reserved, Copyright Mikio Aoyama, 2011



References for Further Understanding

[1] A. Abran, et al. (eds.), Guide to the Software Engineering Body of Knowledge, 2004 Ver., IEEE CS, http://www.swebok.org.
[2] M. Aoyama, et al., REBOK Manifest: Towards a Requirements Engineering Body Of Knowledge, Proc. IEEE RE 2010, IEEE CS, Sep.-Oct. 2010, pp. 383-384.
[3] M. Aoyama, et al., A Model and Architecture of REBOK(Requirements Engineering Body Of Knowledge) and Its Evaluation, Proc. APSEC 2010, IEEE CS, Nov.-Dec. 2010, pp. 50-59.
[4] M. Aoyama, et al., Requirements Engineering Body Of Knowledge, V. 1.0, JISA REBOK WG, Kindaikagakusha, 2011[In Japanese](English Version will be Available).
[5] B. H. C. Cheng and J. M. Atlee, Research Directions in Requirements Engineering, Proc. ICSE 2007, Future of Software Engineering, IEEE CS, May 2007, pp. 285-303.
[6] IIBA, A Guide to the Business Analysis Body of Knowledge (BABOK Guide), V. 2.0, 2009.
[7] IREB, Syllabus: IREB Certified Professional for Requirements Engineering Foundation Level, Ver. 2.0, Oct. 2009, http://certified-re.de/en/.
[8] G. Kotonya, and I. Sommerville. Requirements Engineering, Wiley, 1998.
[9] A. von Lamsweerde, Requirements Engineering. John Wiley & Sons, 2009.
[10] B. Nuseibeh and S. M Easterbrook, Requirements Engineering: A Roadmap, Proc. ICSE 2000, The Future of Software Engineering, ACM, May 2000, pp. 35-46.
[11] K. Pohl. Requirements Engineering, Springer, 2010.
[12] K. Pohl, et al., Requirements Engineering Fundamentals, Rocky Nook, 2011.
[13] K. E. Wiegers, Software Requirements, 2nd ed., Microsoft Press, 2003.
[14] E. Yu, et al.(eds.), Social Modeling for Requirements Engineering, The MIT Press, 2011.

89

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)

90