RE Activities, Bespoke RE, Stakeholders

Lecture 2, DAT230, Requirements Engineering
Robert Feldt, 2011-09-07

Recap

- Software Engineering is more than technology
- RE in particular: human-centered => multi-disciplinary
- RE mistakes very costly
- No matter which process: Requirements still key
- Engineers focus on solutions - RE on problem domain
  - Constant “battle” - never enough time/resources
- RE is more than writing requirements
- Req = desired, observable characteristic
- Types: Functional, Quality/NFR, Dev Constraints

Basic concepts and activities

Guide to the

Software Engineering Body of Knowledge

2004 Version

Executive Editors
Alain Abran, École de technologie supérieure
James W. Moore, The MITRE Corp.

Editors
Pierre Bourque, École de technologie supérieure
Robert Dupuis, Université du Québec à Montréal

IEEE COMPUTER SOCIETY

IEEE

BOEING

Canadian Council of Professional Engineers
Conseil canadien des ingénieurs

Construx
delivering software project success

MITRE

NIST

NRC·CNRC

Rational
the software development company

Raytheon

SAP

Guide to the

Software Engineering Body of Knowledge

2004 Version

SWEBOK

Executive Editors
Alain Abran, École de technologie supérieure
James W. Moore, The MITRE Corp.

Editors
Pierre Bourque, École de technologie supérieure
Robert Dupuis, Université du Québec à Montréal

IEEE COMPUTER SOCIETY

IEEE

BOEING

Canadian Council of Professional Engineers
Conseil canadien des ingénieurs

Construx
delivering software project success

MITRE

NIST

NRC·CNRC

Rational
the software development company

Raytheon

SAP

http://swebok.org

Guide to the Software Engineering Body of Knowledge

SWEBOK

2004 Version

Executive Editors
Alain Abran, École de technologie supérieure
James W. Moore, The MITRE Corp.

Editors
Pierre Bourque, École de technologie supérieure
Robert Dupuis, Université du Québec à Montréal

IEEE COMPUTER SOCIETY

IEEE

BOEING

Canadian Council of Professional Engineers
Conseil canadien des ingénieurs

Construx
delivering software project success

MITRE

NIST

NRC·CNRC

Rational
the software development company

Raytheon

SAP

http://swebok.org

Guide to the Software Engineering Body of Knowledge

2004 Version

SWEBOK

Purpose: Consensus definition of what SE is and is not

Executive Editors
Alain Abran, École de technologie supérieure
James W. Moore, The MITRE Corp.

Editors
Pierre Bourque, École de technologie supérieure
Robert Dupuis, Université du Québec à Montréal

IEEE COMPUTER SOCIETY

IEEE

BOEING

Canadian Council of Professional Engineers
Conseil canadien des ingénieurs

Construx
delivering software project success

MITRE

NIST

NRC·CNRC

Rational
the software development company

Raytheon

SAP

http://www.computer.org/portal/web/swebok

eader | GScholar | GNews | Post to delicious | CTH | Vä | BTH | Lexin | Samsok | Time Zones | RE Schema | Västtrafik

IEEE computer society

Search | Advanced

The world's leading membership organization for computing professionals

IEEE

Guide to the Software Engineering Body of Knowledge (SWEBOK)

SWEBOK
IEEE computer society

Get the 2004 SWEBOK Guide

» HTML (free)
» PDF
» Book

SWEBOK News

Editors Picked for SWEBOK Guide Update

2 July 2010 - Volunteer editors have been selected to oversee the updating of specific knowledge areas for the IEEE Computer Society's Guide to the Software Engineering Body of Knowledge.

Click Here

SWEBOK Guide Set for 2010 Refresh

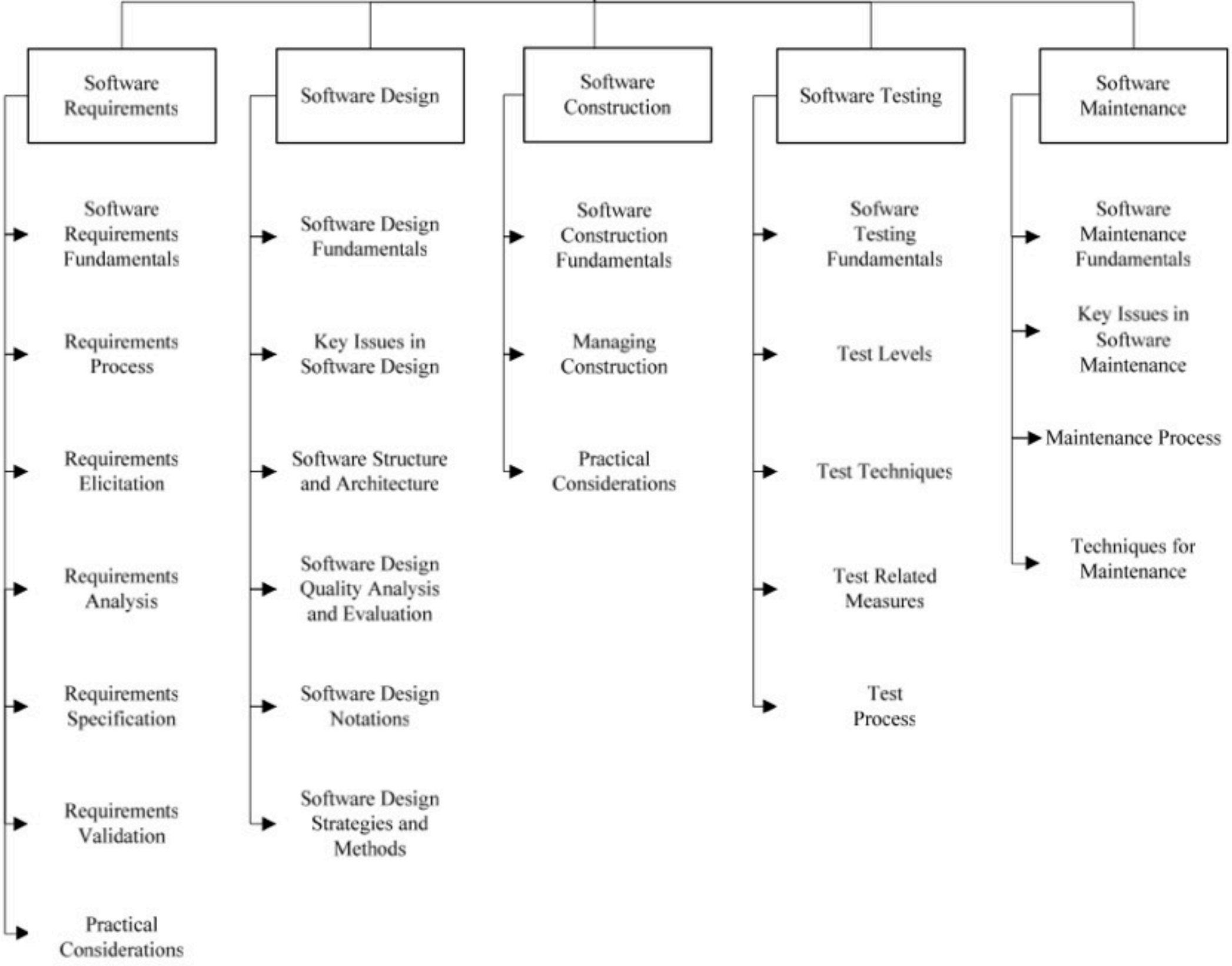
Volunteers are gearing up to refresh the *Guide to the Software Engineering Body of Knowledge*—SWEBOK—intending to add new knowledge areas (KAs) and to revise others.

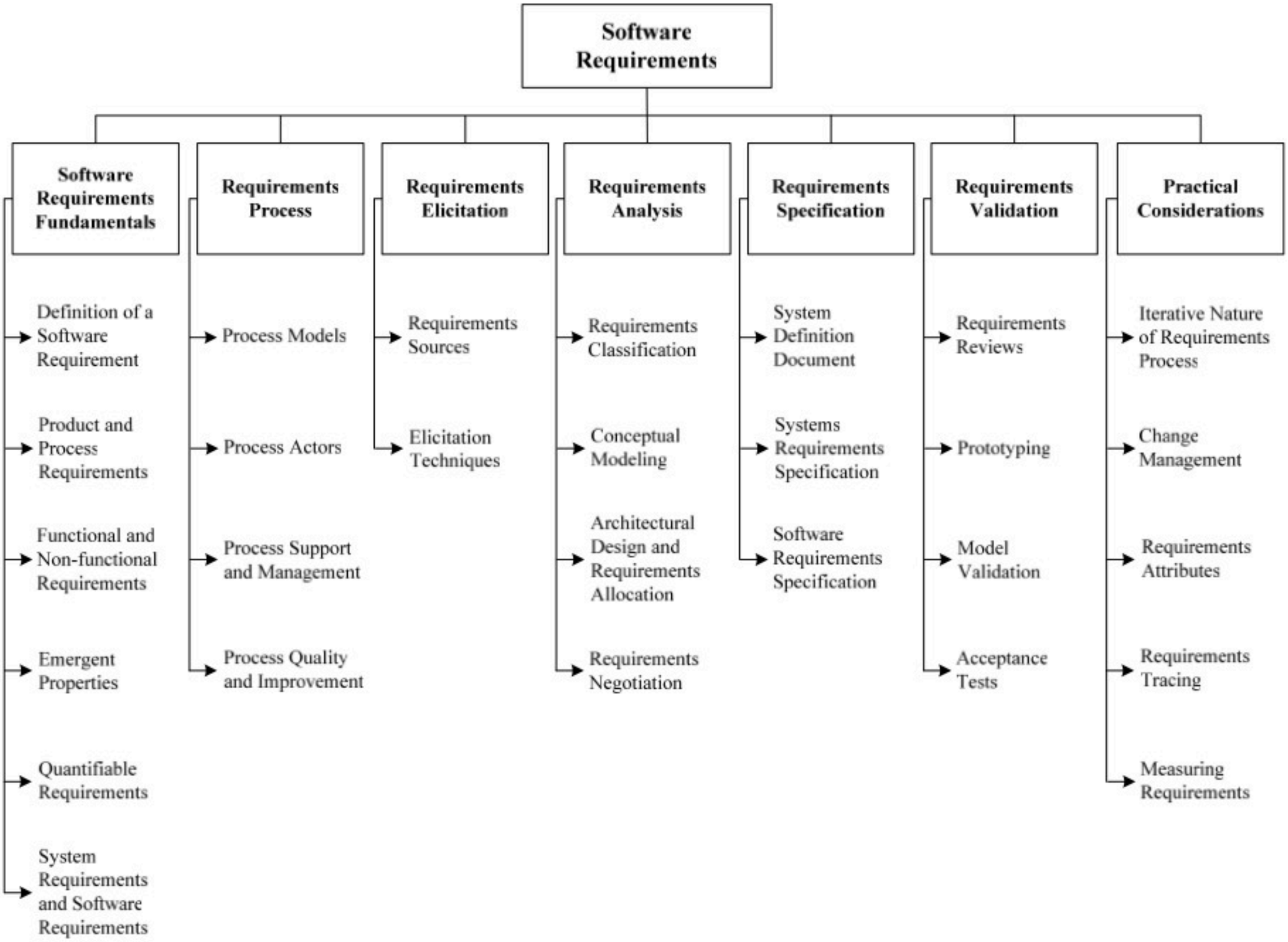
Developed concurrently, the SWEBOK Guide, the Software Engineering 2004 (SE2004) curriculum guide, and the Certified Software Development Professional (CSDP) certification each provided a characterization of the discipline of software engineering. Despite nearly independent development, the three instruments agreed to a remarkable extent. The primary purpose of the current revision of the SWEBOK Guide is to perfect the correspondence between the three items, notably by adding a KA on professional practices—a subject currently covered by the CSDP—and adding "foundation" KAs on related subjects that software engineers learn about during their undergraduate education—subjects currently covered by SE2004.

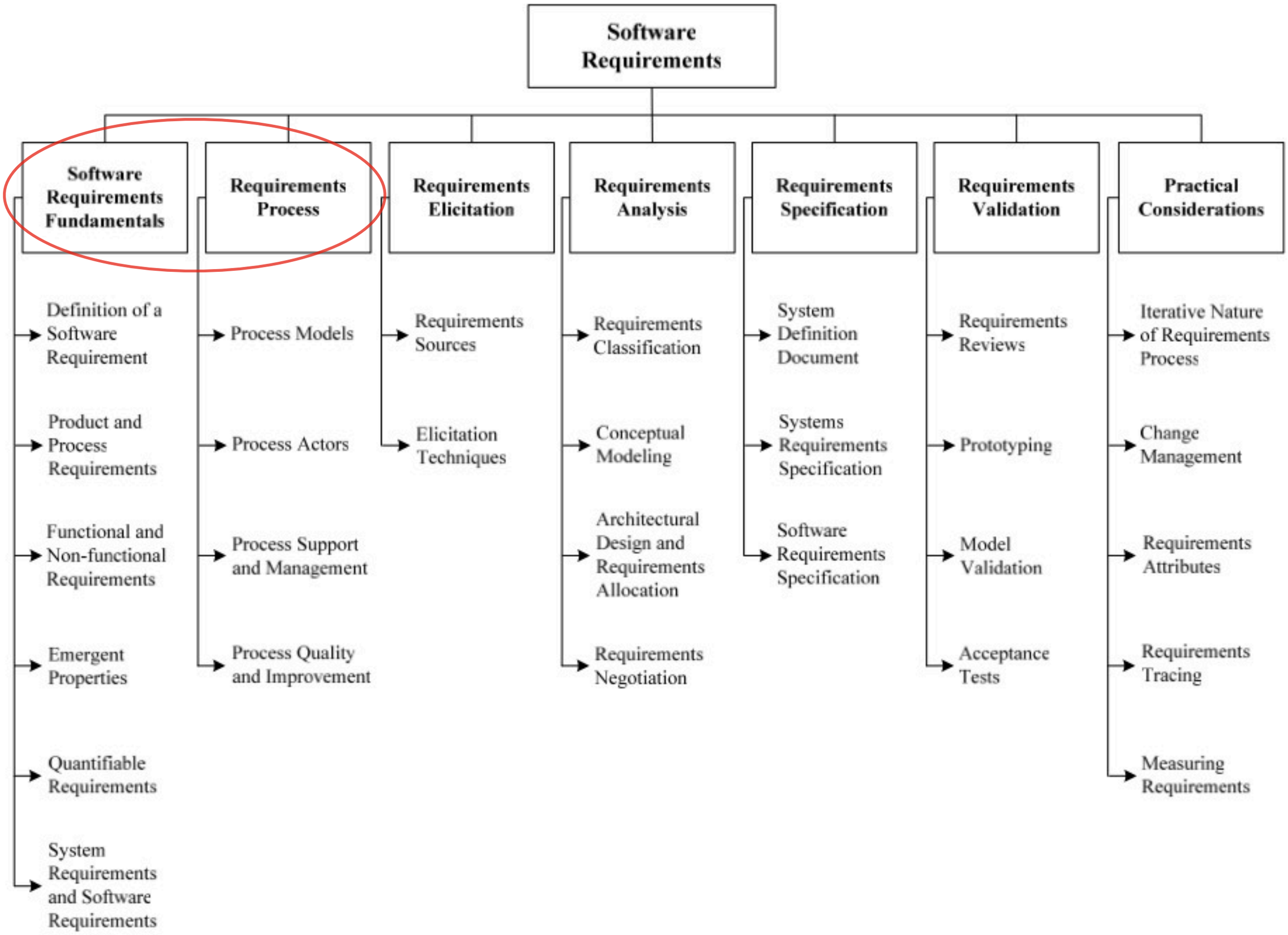
Table 1 The SWEBOK Knowledge Areas (KAs)

Software requirements
Software design
Software construction
Software testing
Software maintenance
Software configuration management
Software engineering management
Software engineering process
Software engineering tools and methods
Software quality

Guide to the Software Engineering Body of Knowledge
2004 Version




Software Requirements
Software Requirements Fundamentals
Definition of a Software Requirement
Product and Process Requirements
Functional and Non-functional Requirements
Emergent Properties
Quantifiable Requirements
System Requirements and Software Requirements
Requirements Process
Process Models
Process Actors
Process Support and Management
Process Quality and Improvement
Requirements Elicitation
Requirements Sources
Elicitation Techniques
Requirements Analysis
Requirements Classification
Conceptual Modeling
Architectural Design and Requirements Allocation
Requirements Negotiation
Requirements Specification
System Definition Document
Systems Requirements Specification
Software Requirements Specification
Requirements Validation
Requirements Reviews
Prototyping
Model Validation
Acceptance Tests
Practical Considerations
Iterative Nature of Requirements Process
Change Management
Requirements Attributes
Requirements Tracing
Measuring Requirements


Software Requirements
Software Requirements Fundamentals
Requirements Process
Requirements Elicitation
Requirements Analysis
Requirements Specification
Requirements Validation
Practical Considerations
Definition of a Software Requirement
Product and Process Requirements
Functional and Non-functional Requirements
Emergent Properties
Quantifiable Requirements
System Requirements and Software Requirements
Process Models
Process Actors
Process Support and Management
Process Quality and Improvement
Requirements Sources
Elicitation Techniques
Requirements Classification
Conceptual Modeling
Architectural Design and Requirements Allocation
Requirements Negotiation
System Definition Document
Systems Requirements Specification
Software Requirements Specification
Requirements Reviews
Prototyping
Model Validation
Acceptance Tests
Iterative Nature of Requirements Process
Change Management
Requirements Attributes
Requirements Tracing
Measuring Requirements

SWEBOK KA1.1.1 Definition

SWEBOK KA1.1.1 Definition

Req = property a SW must exhibit to solve real-world problem

SWEBOK KA1.1.1 Definition

Req = property a SW must exhibit to solve real-world problem

Reqs must be verifiable

SWEBOK KA1.1.1 Definition

Req = property a SW must exhibit to solve real-world problem

Reqs must be verifiable

Reqs often have other attributes like priority rating

SWEBOK KA1.1.1 Definition

Req = property a SW must exhibit to solve real-world problem

Reqs must be verifiable

Reqs often have other attributes like priority rating

Reqs have unique identifier for configuration control and management throughout lifecycle

SWEBOK KA1.1.2 Product & Process Reqs

SWEBOK KA1.1.2 Product & Process Reqs

Product Req = req on software to be developed

SWEBOK KA1.1.2 Product & Process Reqs

Product Req = req on software to be developed

Process Req = development constraint

SWEBOK KA1.1.2 Product & Process Reqs

Product Req = req on software to be developed

Process Req = development constraint

SWEBOK KA1.1.3 FR & NFR

Functional Req describes functions of SW

Non-Functional Reqs constrain the solution (also called Constraints or Quality Reqs)

SWEBOK KA1.1.4 Emergent Properties

SWEBOK KA1.1.4 Emergent Properties

Some reqs represent Emergent Properties

SWEBOK KA1.1.4 Emergent Properties

Some reqs represent Emergent Properties

EPs cannot be satisfied by single component, determined by how all components interoperate

SWEBOK KA1.1.4 Emergent Properties

Some reqs represent Emergent Properties

EPs cannot be satisfied by single component, determined by how all components interoperate

SWEBOK KA1.1.5 Quantifiable

Reqs stated clearly, unambiguously & quantitatively

SWEBOK KA1.1.5 Quantifiable

SWEBOK KA1.1.5 Quantifiable

Reqs stated clearly, unambiguously &
quantitatively

SWEBOK KA1.1.5 Quantifiable

Reqs stated clearly, unambiguously &
quantitatively

Should not rely on subjective judgment

SWEBOK KA1.1.5 Quantifiable

Reqs stated clearly, unambiguously & quantitatively

Should not rely on subjective judgment

"The software shall be reliable"

SWEBOK KA1.1.5 Quantifiable

Reqs stated clearly, unambiguously & quantitatively

Should not rely on subjective judgment

"The software shall be reliable"

"The software should be user-friendly"

SWEBOK KA1.1.5 Quantifiable

Reqs stated clearly, unambiguously & quantitatively

Should not rely on subjective judgment

"The software shall be reliable"

"The software should be user-friendly"

"The call center software must increase the center's throughput by 20%"

SWEBOK KA1.1.5 Quantifiable

Reqs stated clearly, unambiguously & quantitatively

Should not rely on subjective judgment

"The software shall be reliable"

"The probability of a fatal error during one hour of operation should be less than 10^-8"

"The software should be user-friendly"

"The call center software must increase the center's throughput by 20%"

SWEBOK KA1.1.6 System & Software Reqs

SWEBOK KA1.1.6 System & Software Reqs

System = interacting combination of elements to accomplish a given objective

SWEBOK KA1.1.6 System & Software Reqs

System = interacting combination of elements to accomplish a given objective

Elements include hardware, software, firmware, people, information, techniques, facilities, services and other support elements

SWEBOK KA1.1.6 System & Software Reqs

System = interacting combination of elements to accomplish a given objective

Elements include hardware, software, firmware, people, information, techniques, facilities, services and other support elements

System reqs are for the system as a whole

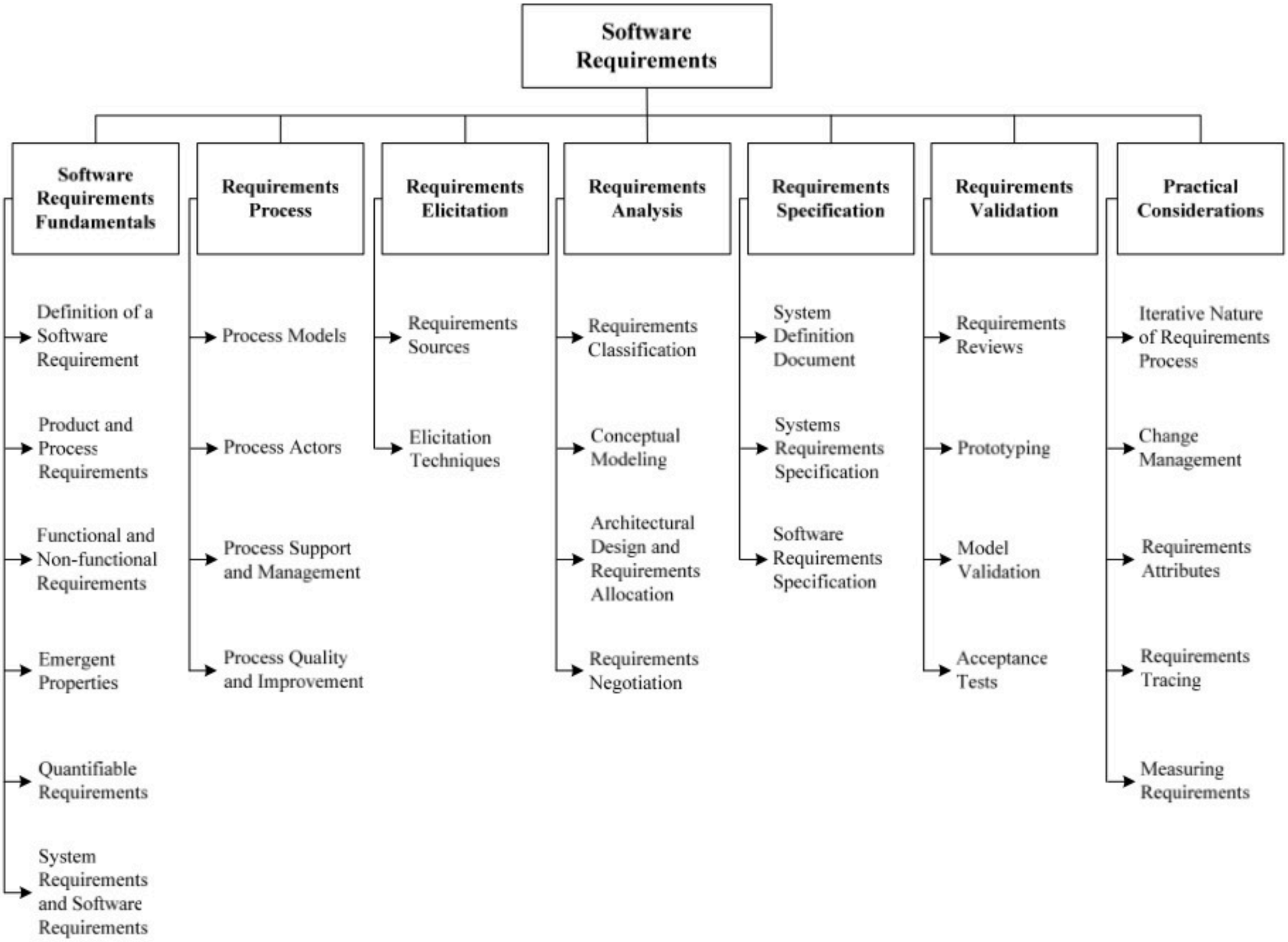
SWEBOK KA1.1.6 System & Software Reqs

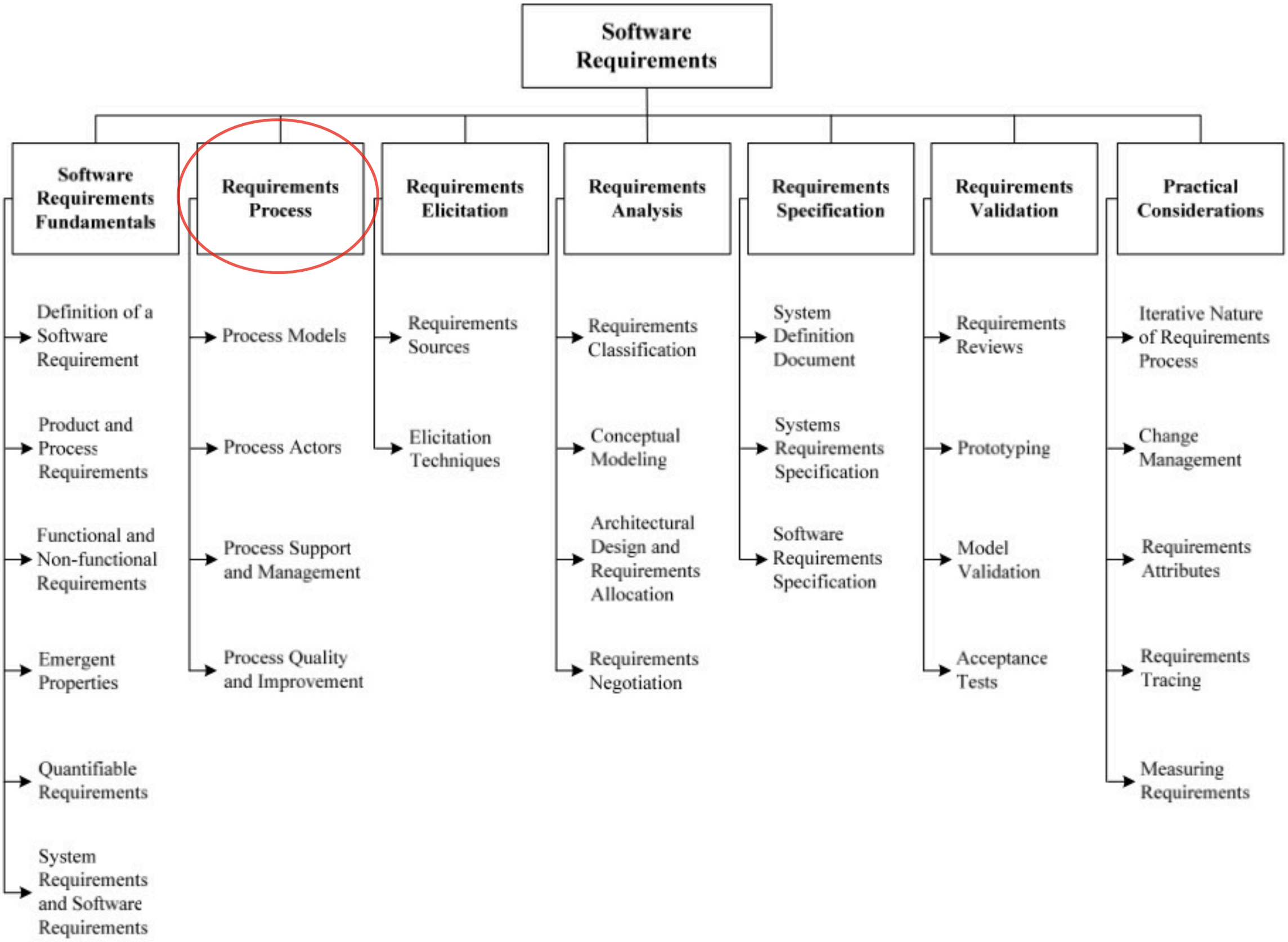
System = interacting combination of elements to accomplish a given objective

Elements include hardware, software, firmware, people, information, techniques, facilities, services and other support elements

System reqs are for the system as a whole

A system with software components has software requirements


Software Requirements
Software Requirements Fundamentals
Definition of a Software Requirement
Product and Process Requirements
Functional and Non-functional Requirements
Emergent Properties
Quantifiable Requirements
System Requirements and Software Requirements
Requirements Process
Process Models
Process Actors
Process Support and Management
Process Quality and Improvement
Requirements Elicitation
Requirements Sources
Elicitation Techniques
Requirements Analysis
Requirements Classification
Conceptual Modeling
Architectural Design and Requirements Allocation
Requirements Negotiation
Requirements Specification
System Definition Document
Systems Requirements Specification
Software Requirements Specification
Requirements Validation
Requirements Reviews
Prototyping
Model Validation
Acceptance Tests
Practical Considerations
Iterative Nature of Requirements Process
Change Management
Requirements Attributes
Requirements Tracing
Measuring Requirements


Software Requirements
Software Requirements Fundamentals
Requirements Process
Requirements Elicitation
Requirements Analysis
Requirements Specification
Requirements Validation
Practical Considerations
Definition of a Software Requirement
Product and Process Requirements
Functional and Non-functional Requirements
Emergent Properties
Quantifiable Requirements
System Requirements and Software Requirements
Process Models
Process Actors
Process Support and Management
Process Quality and Improvement
Requirements Sources
Elicitation Techniques
Requirements Classification
Conceptual Modeling
Architectural Design and Requirements Allocation
Requirements Negotiation
System Definition Document
Systems Requirements Specification
Software Requirements Specification
Requirements Reviews
Prototyping
Model Validation
Acceptance Tests
Iterative Nature of Requirements Process
Change Management
Requirements Attributes
Requirements Tracing
Measuring Requirements

SWEBOK KA1.2.1 Process Models

SWEBOK KA1.2.1 Process Models

Req Process is NOT discrete front-end activity

SWEBOK KA1.2.1 Process Models

Req Process is NOT discrete front-end activity

Req Process configuration manages all reqs

SWEBOK KA1.2.1 Process Models

Req Process is NOT discrete front-end activity

Req Process configuration manages all reqs

Req Process needs adaptation to organization and project context

SWEBOK KA1.2.1 Process Models

Req Process is NOT discrete front-end activity

Req Process configuration manages all reqs

Req Process needs adaptation to organization and project context

Req Process includes input activities like marketing and feasability studies

SWEBOK KA1.2.2 Process Actors

SWEBOK KA1.2.2 Process Actors

Req specialist must mediate between domain of stakeholder and that of SE

SWEBOK KA1.2.2 Process Actors

Req specialist must mediate between domain of stakeholder and that of SE

User = operates the software

SWEBOK KA1.2.2 Process Actors

Req specialist must mediate between domain of stakeholder and that of SE

User = operates the software

Customer = commisioned software or is target market

SWEBOK KA1.2.2 Process Actors

Req specialist must mediate between domain of stakeholder and that of SE

User = operates the software

Customer = commisioned software or is target market

Market analysts = establish market or are proxy customers

SWEBOK KA1.2.2 Process Actors

Req specialist must mediate between domain of stakeholder and that of SE

User = operates the software

Customer = commisioned software or is target market

Market analysts = establish market or are proxy customers

Regulators = establish regulations sw must comply with

SWEBOK KA1.2.2 Process Actors

Req specialist must mediate between domain of stakeholder and that of SE

User = operates the software

Customer = commisioned software or is target market

Market analysts = establish market or are proxy customers

Regulators = establish regulations sw must comply with

SW Engs job to negotiate trade-offs; not all stakeholders can be perfectly satisfied

SWEBOK KA1.2.3 Process Support & Mngmnt

SWEBOK KA1.2.3 Process Support & Mngmnt

Link to other SE Management KA

SWEBOK KA1.2.3 Process Support & Mngmnt

Link to other SE Management KA

SWEBOK KA1.2.4 Process Q & Improvement

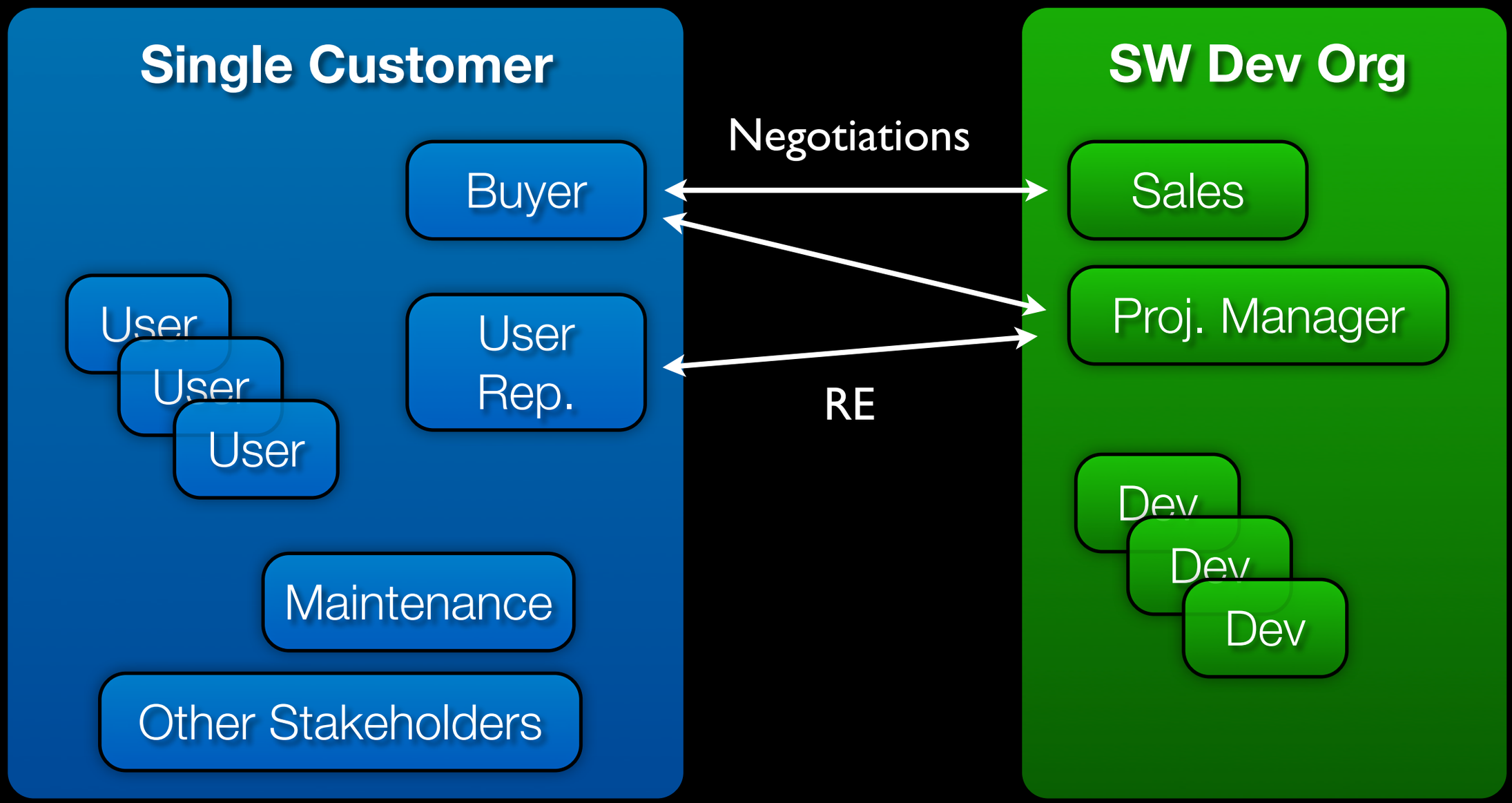
SWEBOK KA1.2.3 Process Support & Mngmnt

Link to other SE Management KA

SWEBOK KA1.2.4 Process Q & Improvement

Link to SE Quality KA & SE Process KA

Bespoke Software Development



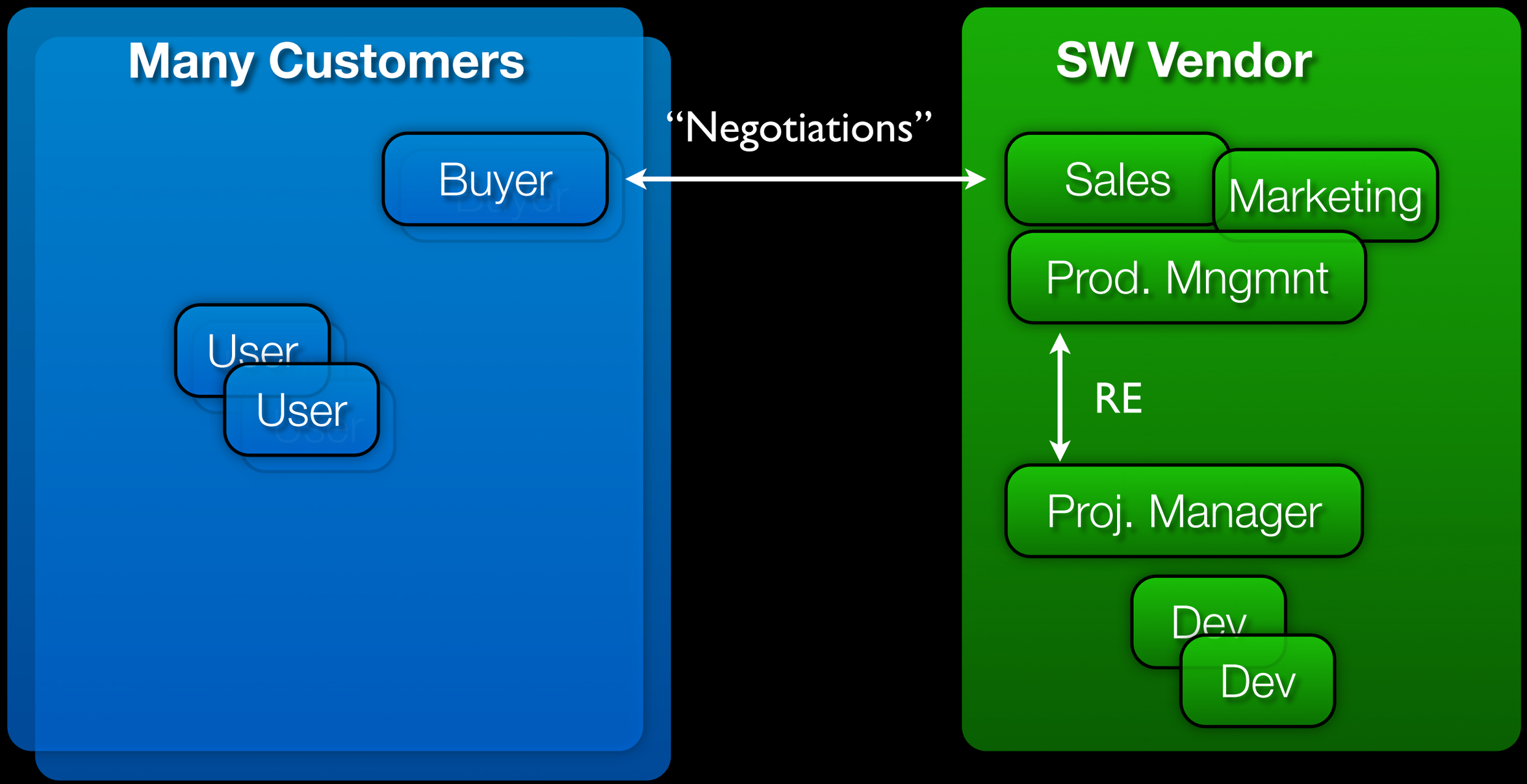
Bespoke Software Development

- Also known as: Custom/Traditional Software Development
- RE primarily startup activity
  - Pre-study/Feasibility study, Contract
  - SW Req Specification (SRS)
  - Changes require negotiations
- Project focus (RE, Analysis, Design, Impl, V&V, Release)
- Domain knowledge from customers/users
- Success = contractual fulfillment & customer satisfaction

Bespoke Software Development - RE steps

1. Customer states need in general terms in Request for Proposals (RFP)
2. Dev company creates proposal = approach, prelim requirements, schedule, budget
3. Customer selects best proposal
4. Dev company prepares SRS & presents
5. Changes => prioritization & negotiations
6. Budget/Schedule problems => prioritization & negotiations

Market-Driven Software Development



Market-Driven Software Development

- Many potential customers (companies and/or end users)
- No “negotiation”, rather elicitation, evaluation, prediction, innovation
- Domain expertise primarily internally
- Success = Sales volume, ROI, Market share, growth

MD Software Development - RE steps

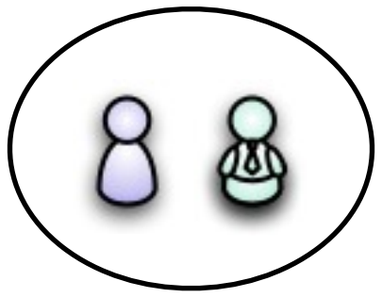
1. Decide what business you are in
2. Select a target market
3. Market research to determine size, competitors, customers, pains/needs, market message
4. Draft high-level features in Market Req Doc (MRD) = desired price, intro date, prioritization
5. Test MRD on potential customers
6. Detailed SRS written
7. Change => internal triage/re-prioritize
8. Budget/Schedule problems => internal triage/re-prioritize

Stakeholder Identification

[Sharp1999]

Stakeholder Identification

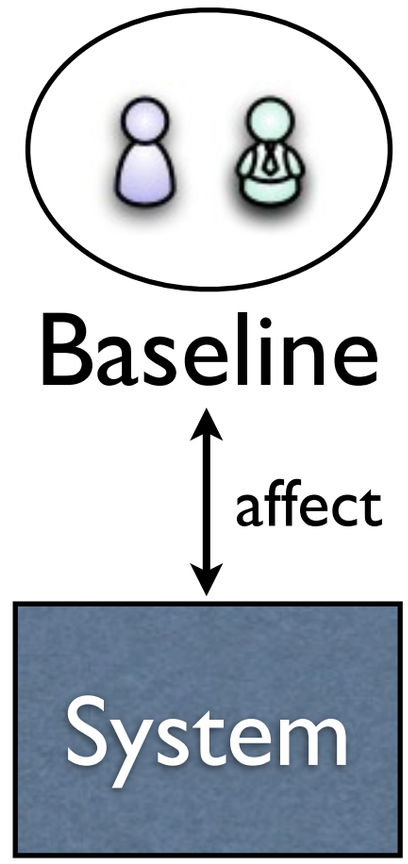
[Sharp1999]



Baseline

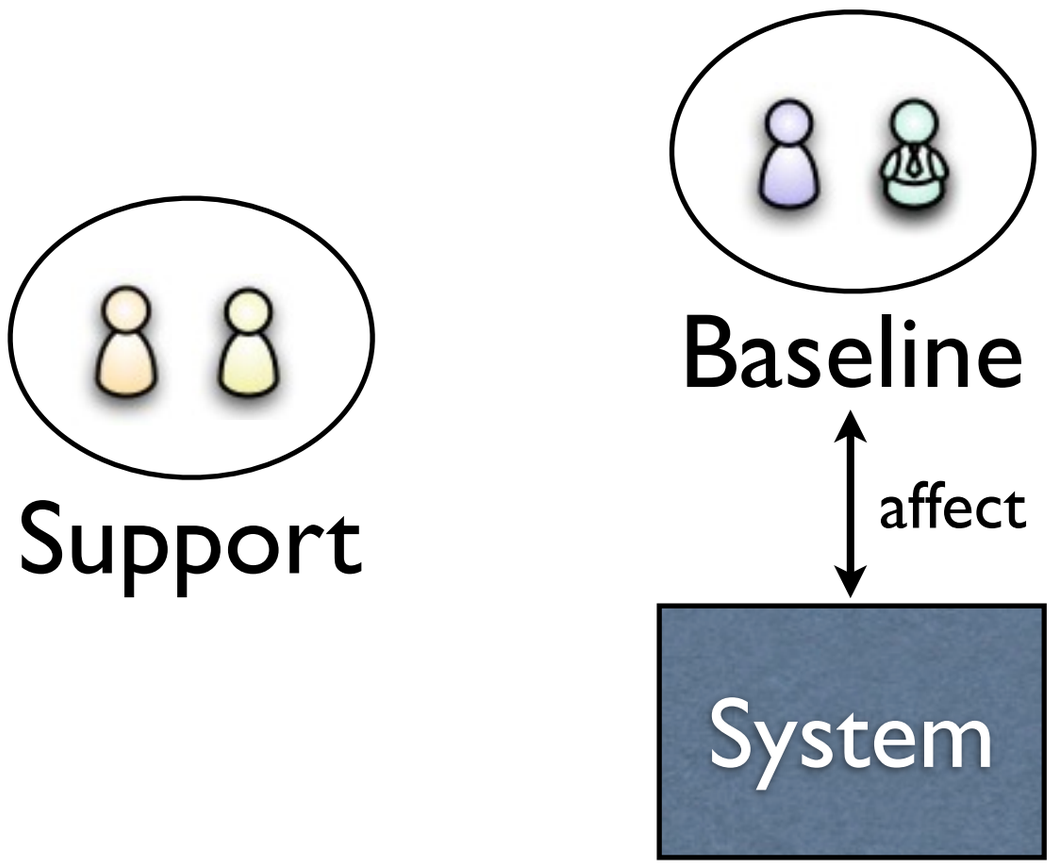
Stakeholder Identification

[Sharp1999]



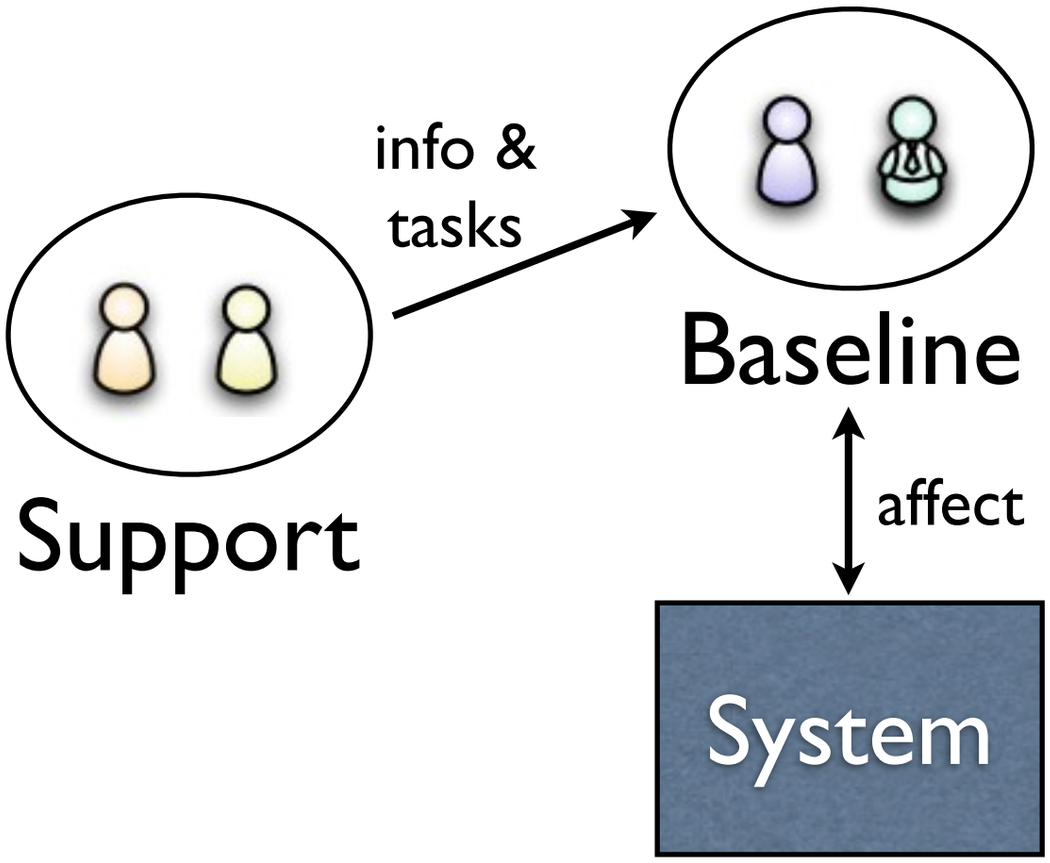
Stakeholder Identification

[Sharp1999]



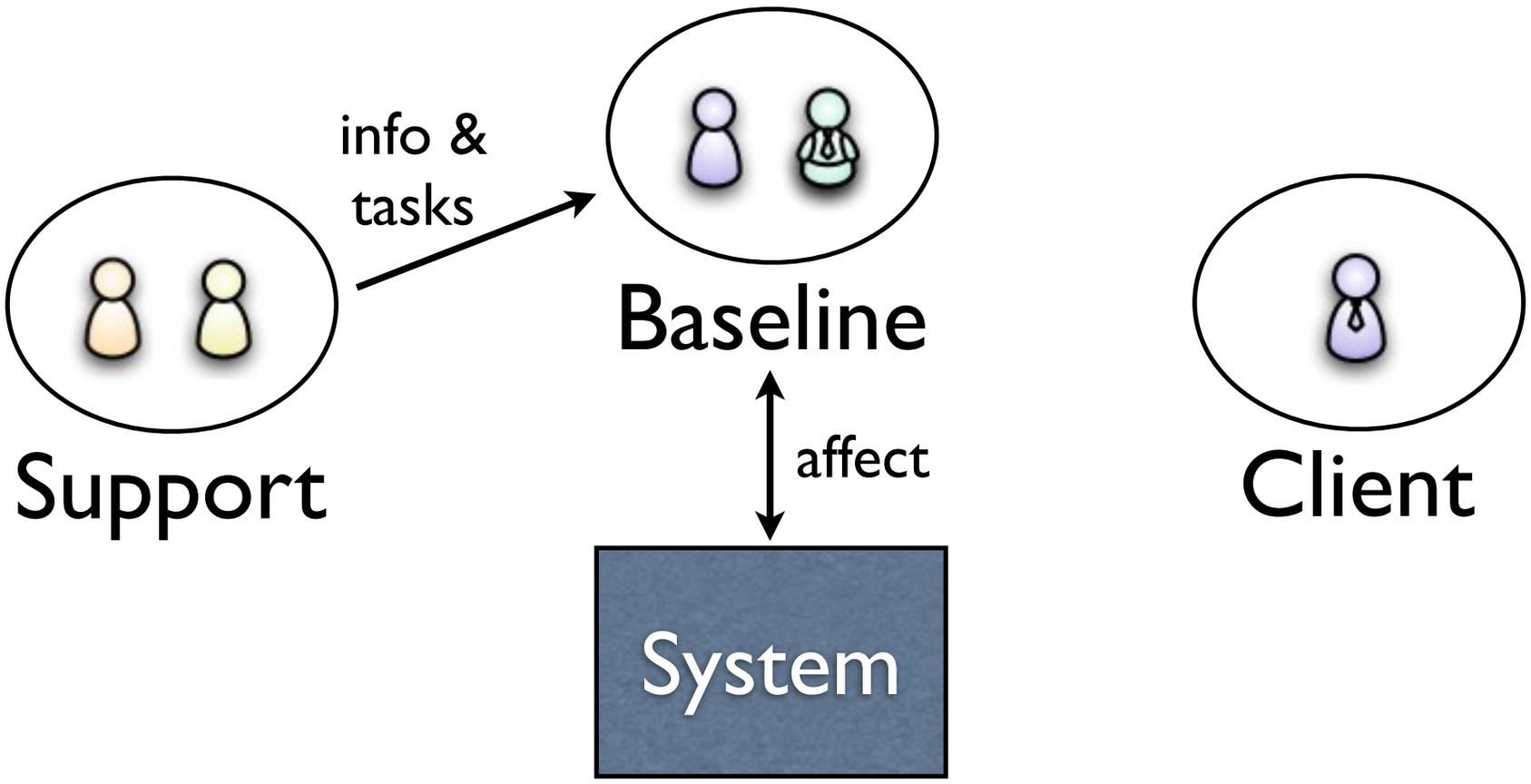
Stakeholder Identification

[Sharp1999]



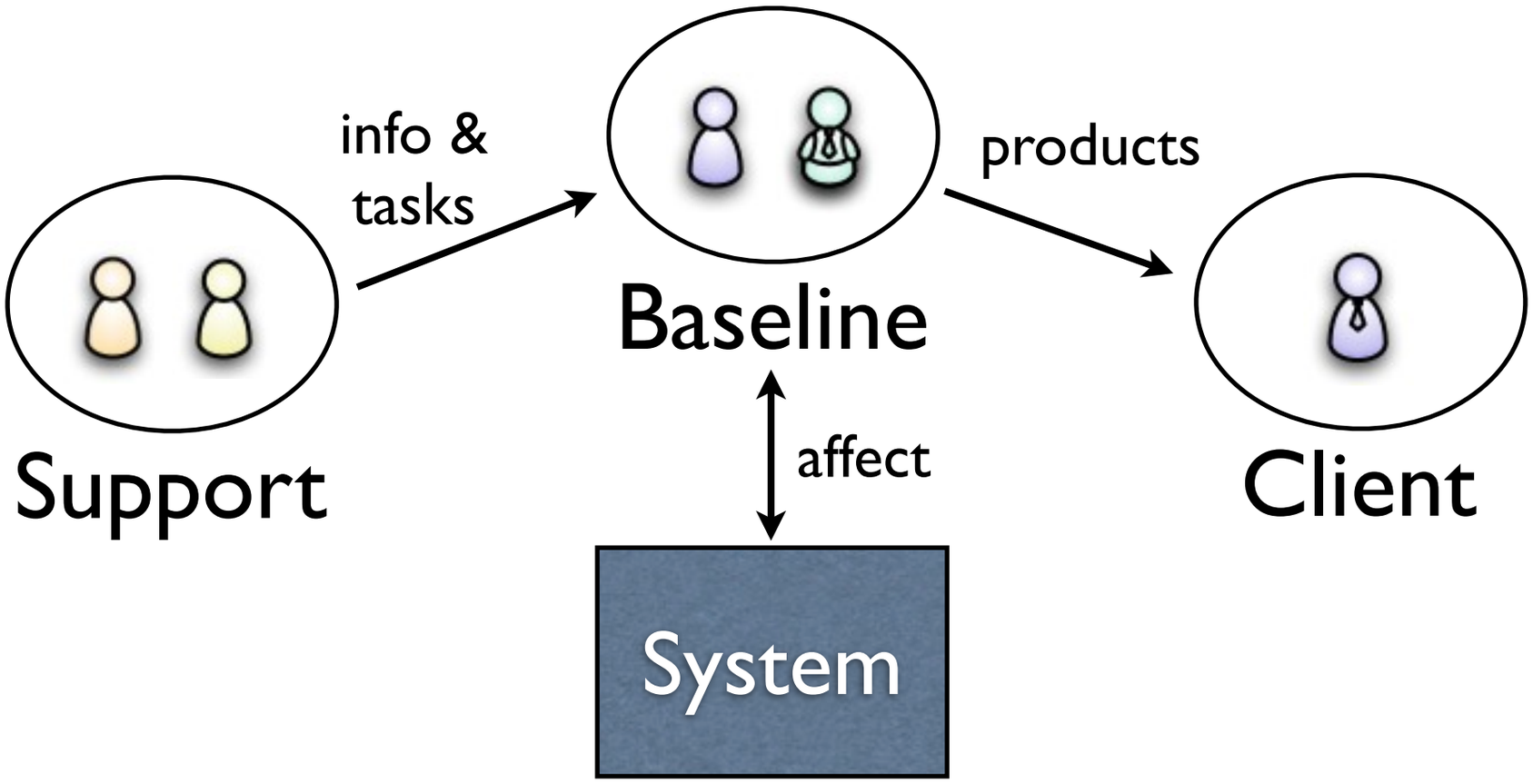
Stakeholder Identification

[Sharp1999]



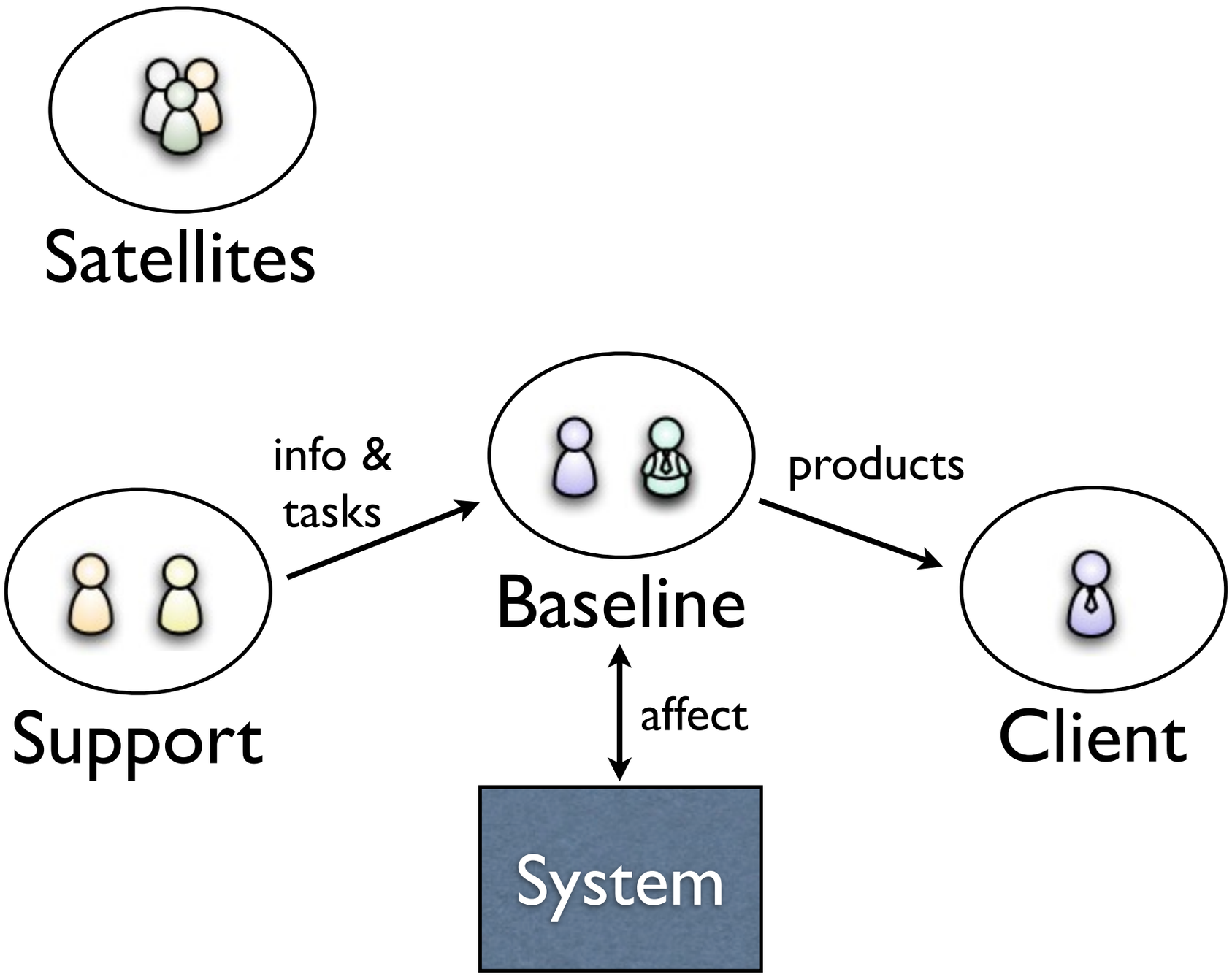
Stakeholder Identification

[Sharp1999]



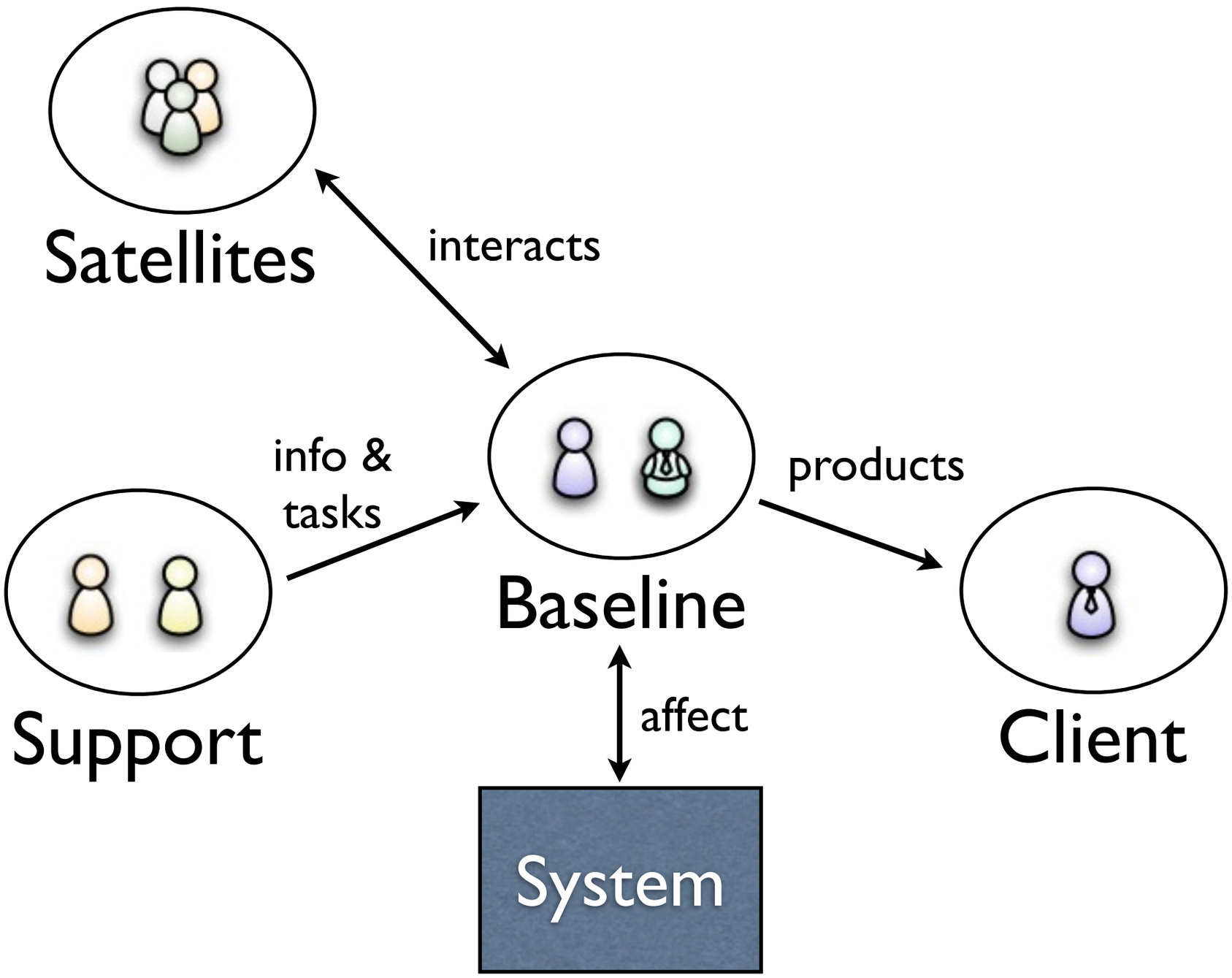
Stakeholder Identification

[Sharp1999]



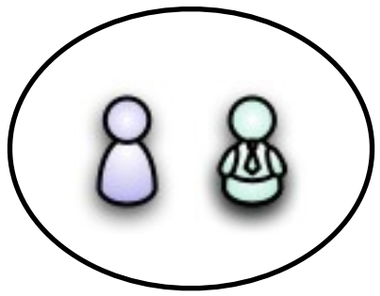
Stakeholder Identification

[Sharp1999]



Stakeholder Identification

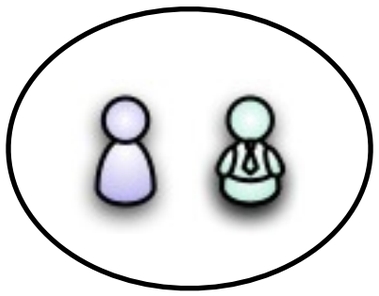
[Sharp1999]



Baseline

Stakeholder Identification

[Sharp1999]



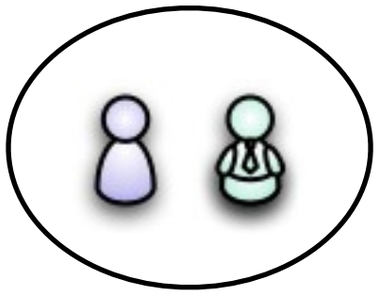
Baseline

Stakeholder Identification

[Sharp1999]

Users - operate the SW

Developers - develop the SW



Baseline

Legislators - constrains the SW

Decision-makers - takes decisions

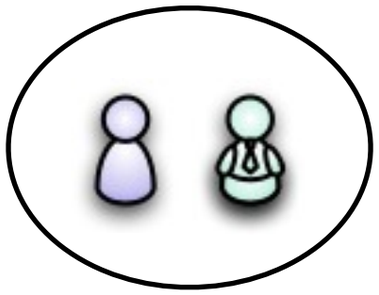
Stakeholder Identification

[Sharp1999]

Users - operate the SW

Frequent users, occasional users,
future & past users, users of products from sw

Developers - develop the SW



Baseline

Legislators - constrains the SW

Decision-makers - takes decisions

Stakeholder Identification

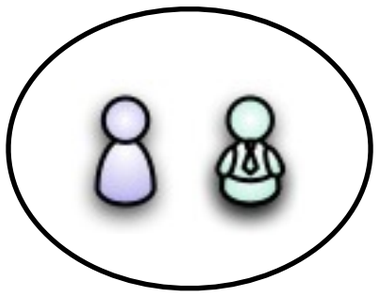
[Sharp1999]

Users - operate the SW

Frequent users, occasional users,
future & past users, users of products from sw

Developers - develop the SW

Developers, Analysts, Designers, QA,
Maintainers, Trainers, Project managers



Baseline

Legislators - constrains the SW

Decision-makers - takes decisions

Stakeholder Identification

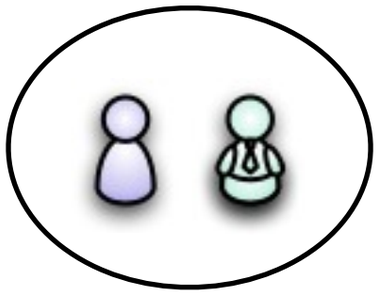
[Sharp1999]

Users - operate the SW

Frequent users, occasional users,
future & past users, users of products from sw

Developers - develop the SW

Developers, Analysts, Designers, QA,
Maintainers, Trainers, Project managers



Baseline

Legislators - constrains the SW

Government, Community, Trade unions,
Legal representatives, Standard bodies (ISO, IEEE),
Auditors (TUV)

Decision-makers - takes decisions

Stakeholder Identification

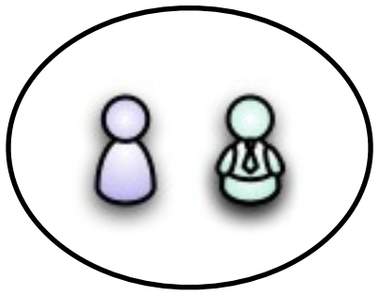
[Sharp1999]

Users - operate the SW

Frequent users, occasional users,
future & past users, users of products from sw

Developers - develop the SW

Developers, Analysts, Designers, QA,
Maintainers, Trainers, Project managers



Baseline

Legislators - constrains the SW

Government, Community, Trade unions,
Legal representatives, Standard bodies (ISO, IEEE),
Auditors (TUV)

Decision-makers - takes decisions

Dev & user managers,
Financial managers/controllers

Stakeholder Identification

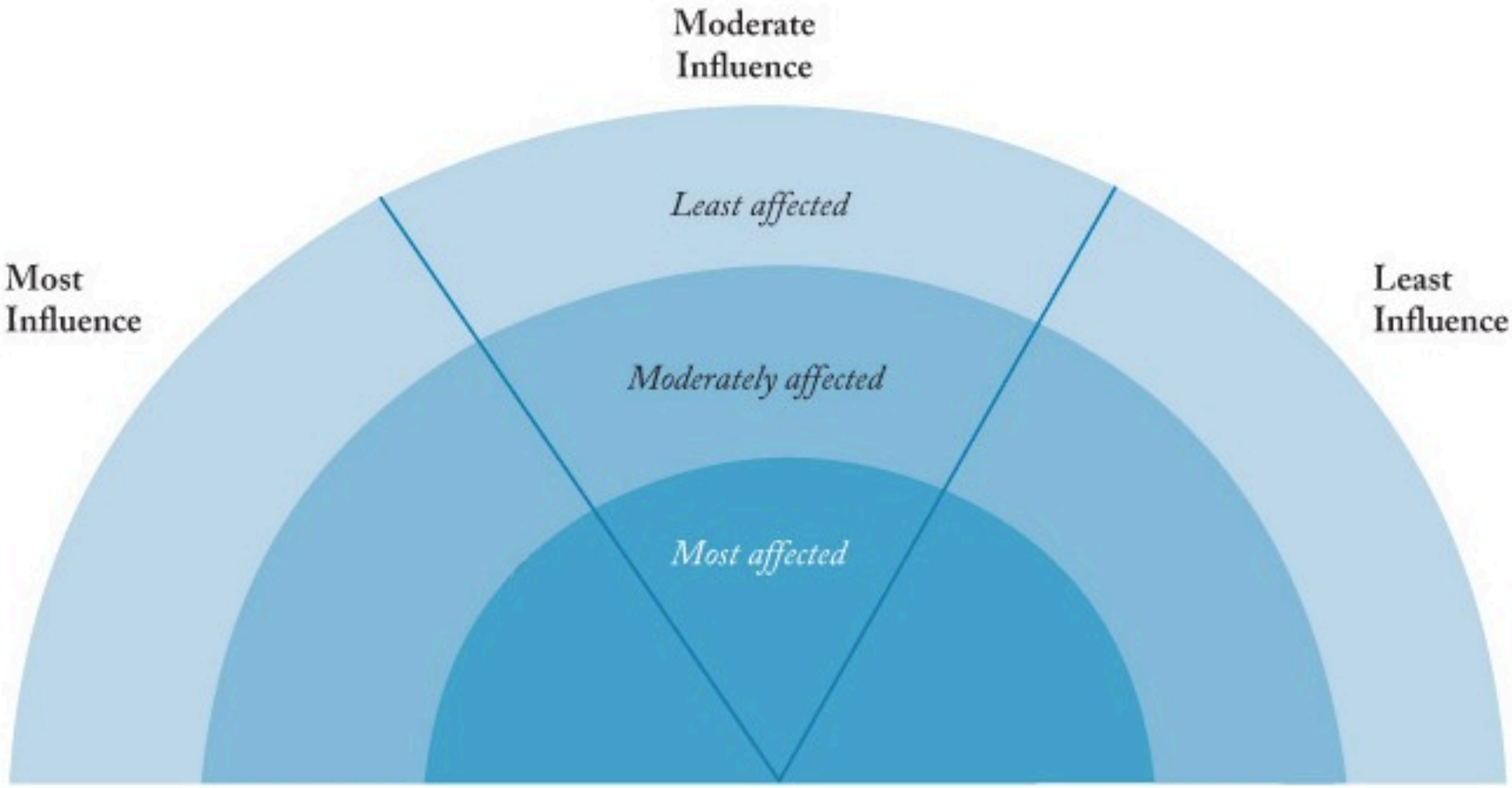
[Sharp1999]

1. Identify all relevant groups of baseline stakeholders
2. Identify all relevant roles within each baseline group
3. For each baseline role:
    1. Who supplies information to this role? Who performs supporting tasks? => Support stakeholders
    2. Who processes or inspects products from this role? => Client
    3. Who interacts with this role in other ways? => Satellite
4. Repeat 3 above for newly found stakeholders
5. Consider relations between identified stakeholders: "in charge of", "supports", "is crucial to", "provides info for", ...

Stakeholder Analysis

- Who are the stakeholders?
- Do we have access to them?
- What are their expectations and interests?
- What are their influence and role in project?

Stakeholder Analysis



Rainbow diagram

Stakeholder Analysis

- Expectations and interests
  - Personal: Work or Family focus, Job satisfaction, Org satisfaction, Improving knowledge, Sufficient appreciation, Workload/Responsibility
  - Social: Peer recognition, Cover incompetence, Sponsorships, Undermining, On the move, Power hierarchies
  - Material: Money, Tools, Office, Travels