



# Processes for Requirements Engineering

- Importance of Requirements Engineering
- Activities and processes
- Terminology

# The Importance of RE in SWE



- Software is complex
  - It is invisible and abstract
  - It is highly modifiable since no fabrication step is involved
- Our society increasingly relies on all types of software
  - Information Systems
    - Software supports organisational work, e.g., payroll, customer records, accounting, ...
    - Software includes databases, standard applications as well as Internet applications
  - Embedded Systems
    - Software controls complex hardware systems, e.g., aircrafts, cars, industrial plants, cash machines, lifts, ...
    - In a decade, 60% of the value of a car will be in its software

# Early Modelling and Analysis is Important

RE is a technical activity employing computer sciences

- Modelling and analysis techniques
  - Semi-formal techniques are widely used today
    - E.g., Unified Modelling Language (UML) and Object-Oriented Analysis (OOA)
  - Formal techniques not yet widely adopted in practice
    - E.g., Software Cost Reduction (SCR)
- Systems analysis
  - As used in the information systems world
- Systems theory and practice
  - Relevant in the whole-system context

“The cost of good requirements gathering and systems analysis is minor compared to the cost of poor requirements.” [Robertson 1999]



# Early Modelling and Analysis is Not Enough



- There is a need to
  - Communicate requirements to everyone
  - Seek agreement from all stakeholders
  - Understand the context of the system
  - Understand the context of the development process
  - Keep up-to-date as the requirements evolve
- RE involves many non-technical disciplines...

# Non-Technical Disciplines of RE



- **Cognitive psychology** helps in understanding people's difficulties by describing their needs
  - Domain experts often have tacit knowledge that is not amenable to introspection
  - Cognitive psychologists are able to model the users' understanding of user interfaces and/or user behaviour (e.g. HTA, GOMS, ACT-R)
- **Anthropology** provides methods for observing human activities, including techniques for analysing collaborative work and team interaction (e.g. Ethnography, ethnomethodology)
- **Sociology** presents an understanding of the political and cultural changes caused by computerisation
- **Linguistics** analyses can improve understandability and avoid ambiguity

# Requirements Need a Process

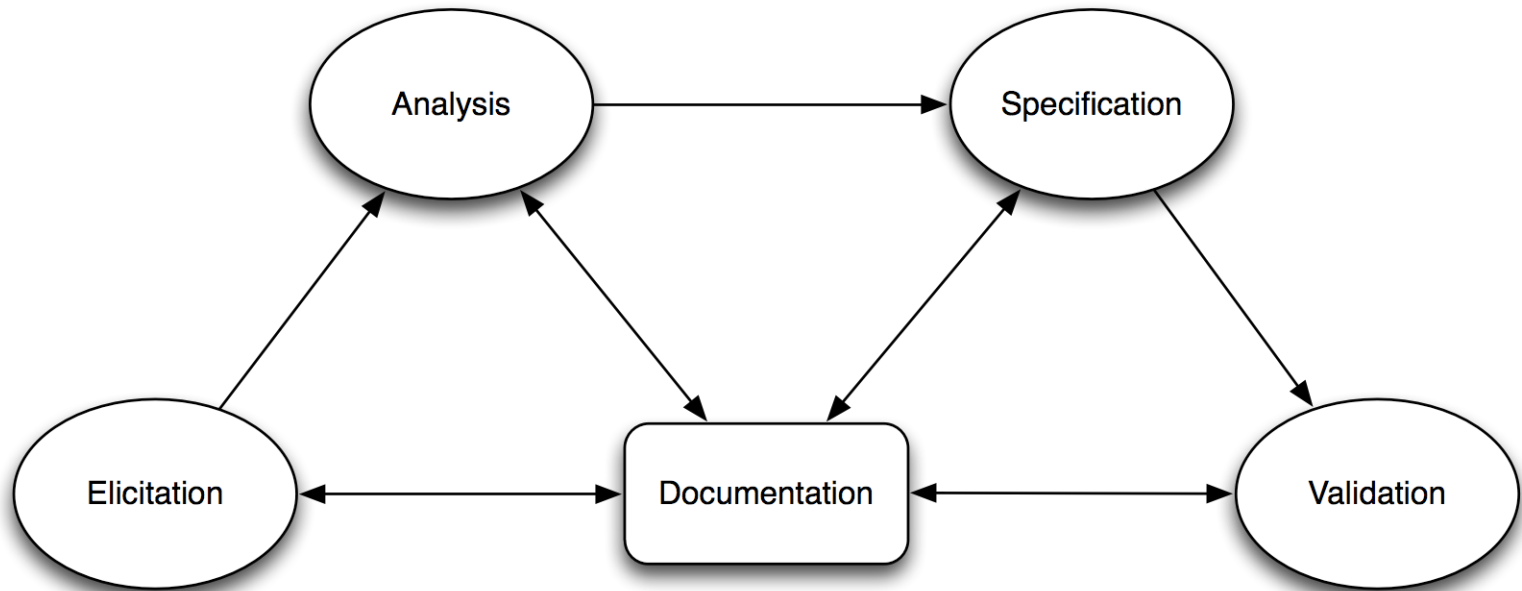


- The development of requirements
  - Involves two or more individuals ...
  - ... co-operating to reach agreement
  - Consumes resources
- A process is needed to do this efficiently
  - Requirements engineering

## The RE process:

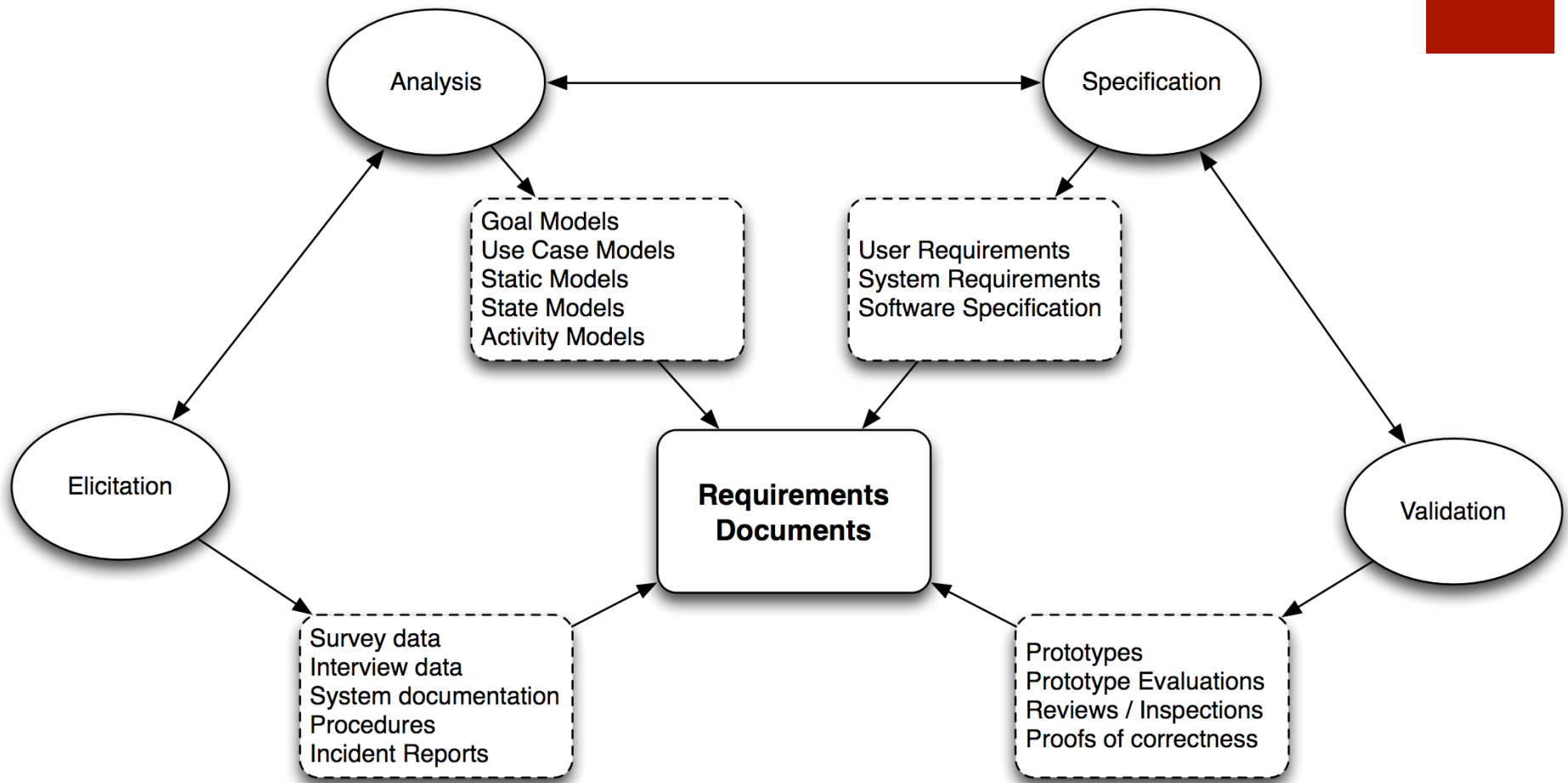
- Who are the participants?
- In which activities do they engage?
- Which process is followed to coordinate the activities?

# An RE Model



Adapted from Sommerville 2005 &  
Van Vliet, 2008

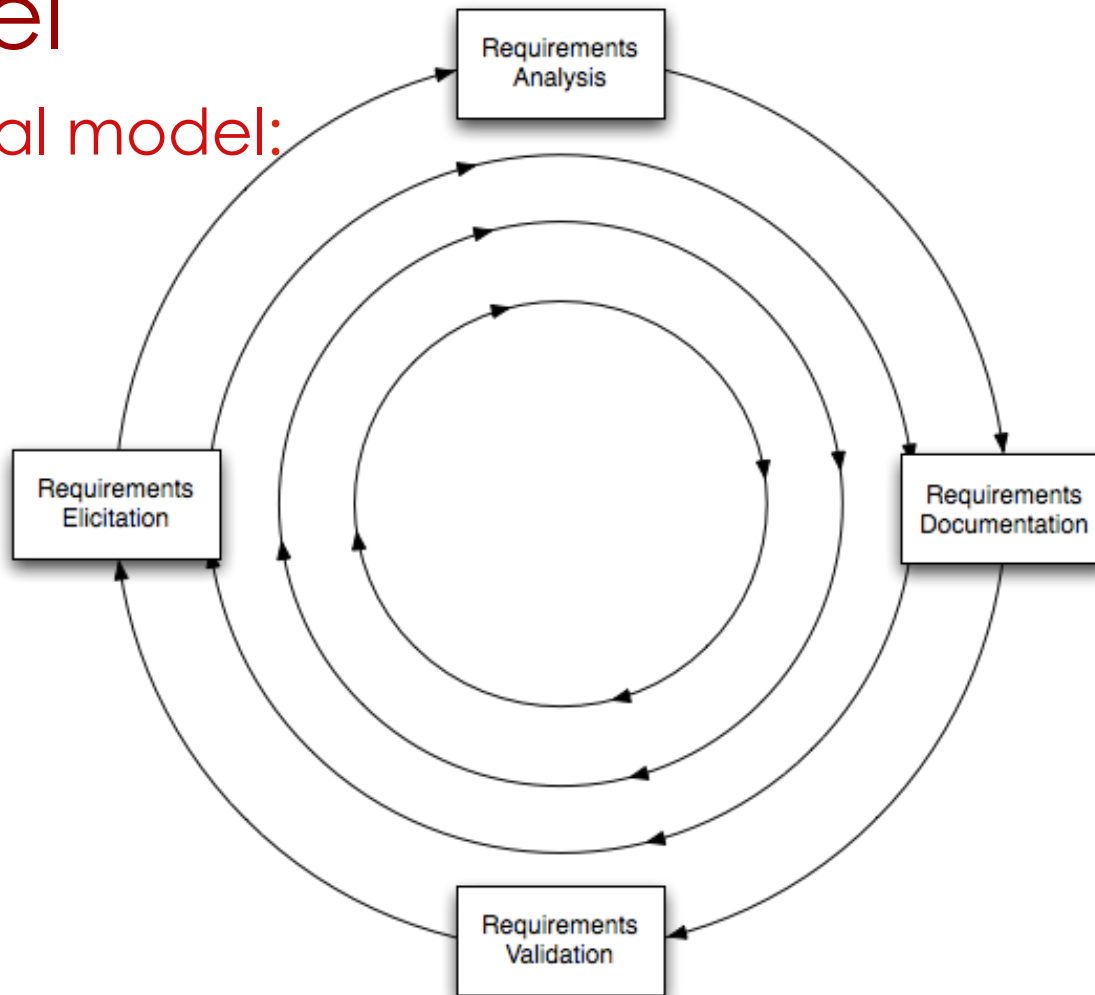
# A more detailed version ...





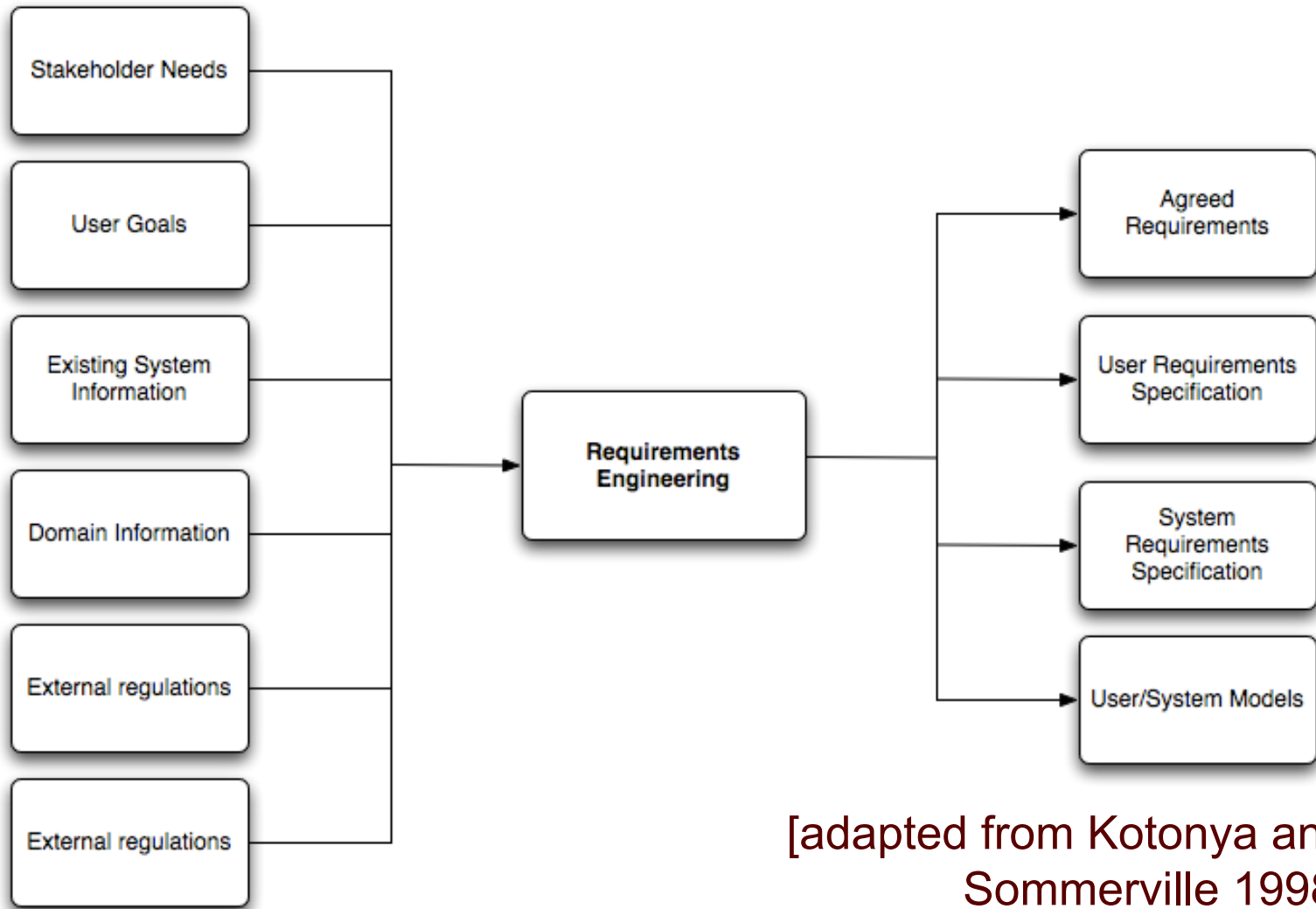
# Another view of the Process Model

The spiral model:



...recognizes the traditional view of the RE process

# Inputs and Outputs of the RE Process



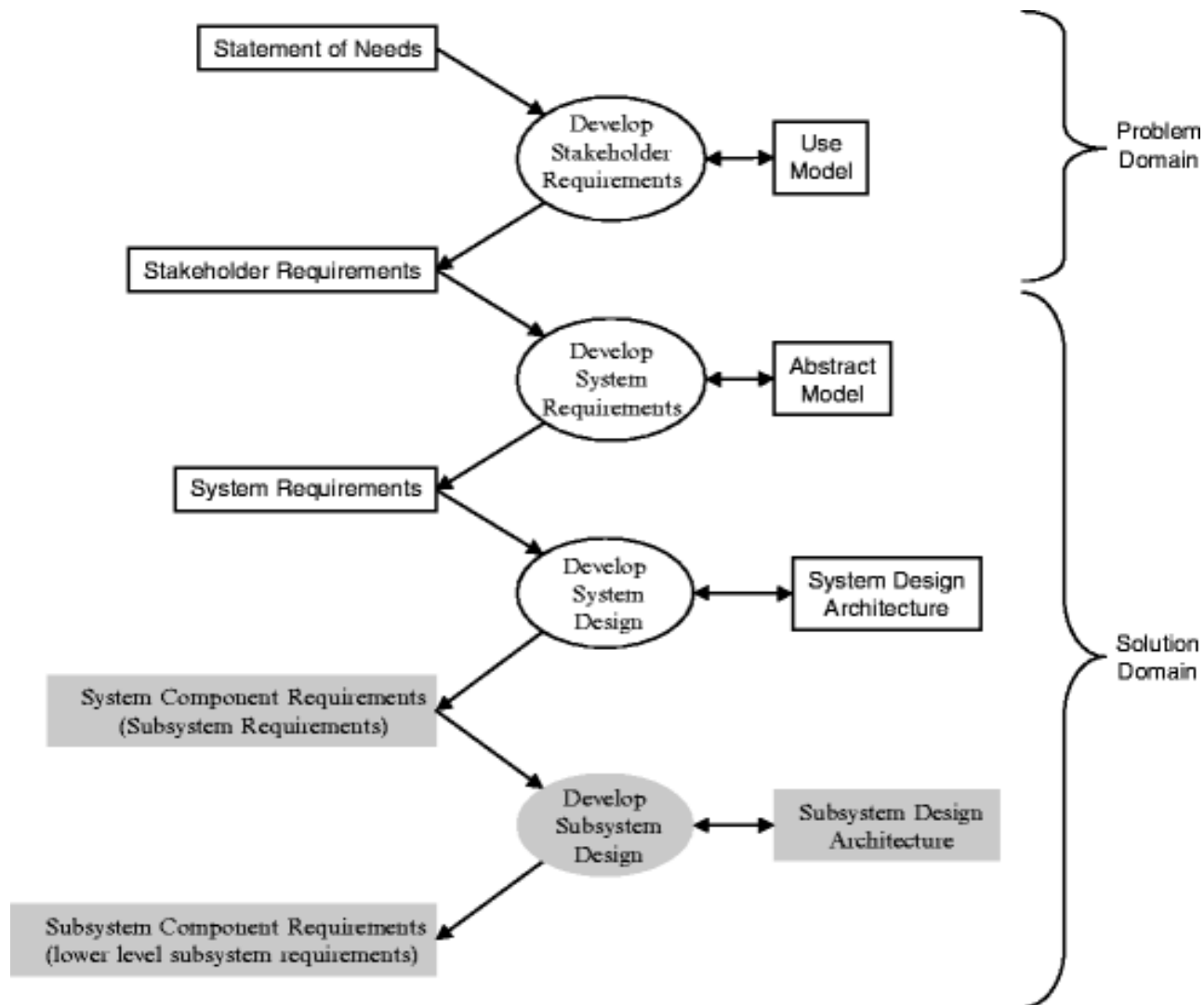
[adapted from Kotonya and Sommerville 1998]

# What is the problem with these models?

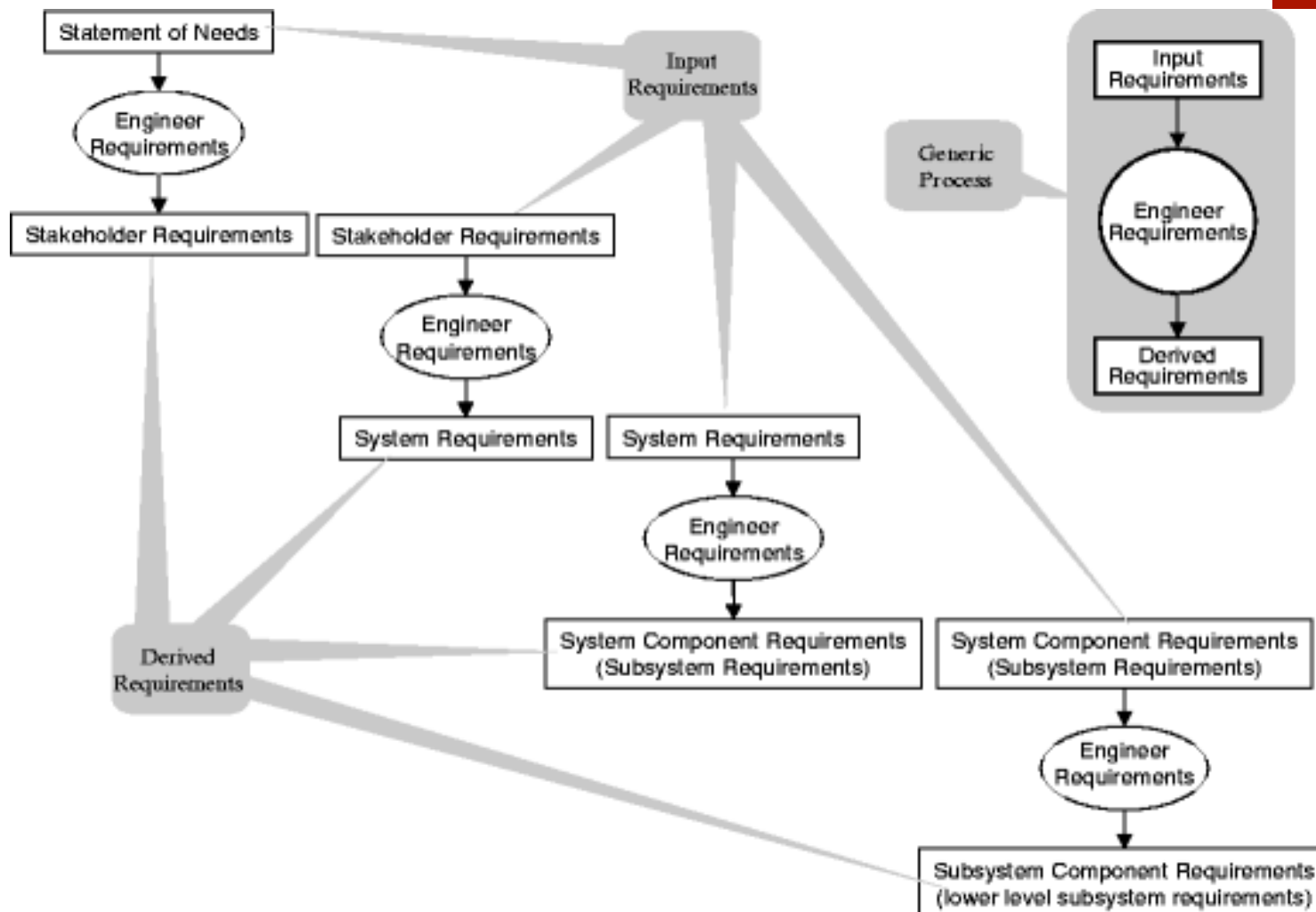


- They do not a bad job of telling you what the outputs are – but they don't really explain the process very clearly
- They all tell you how requirements engineering happens from a very high level
- But what does it actually mean to do requirements engineering?

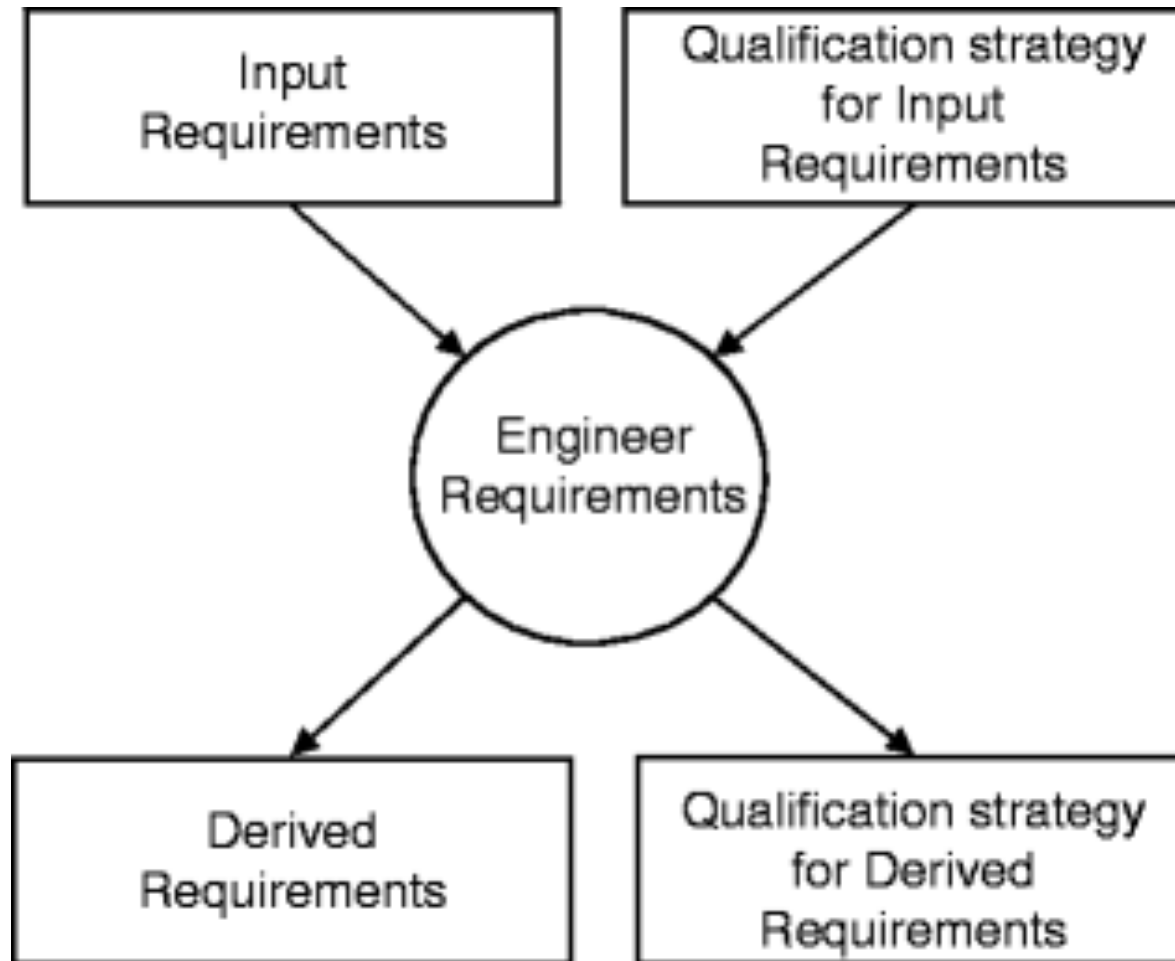
# Two different domains



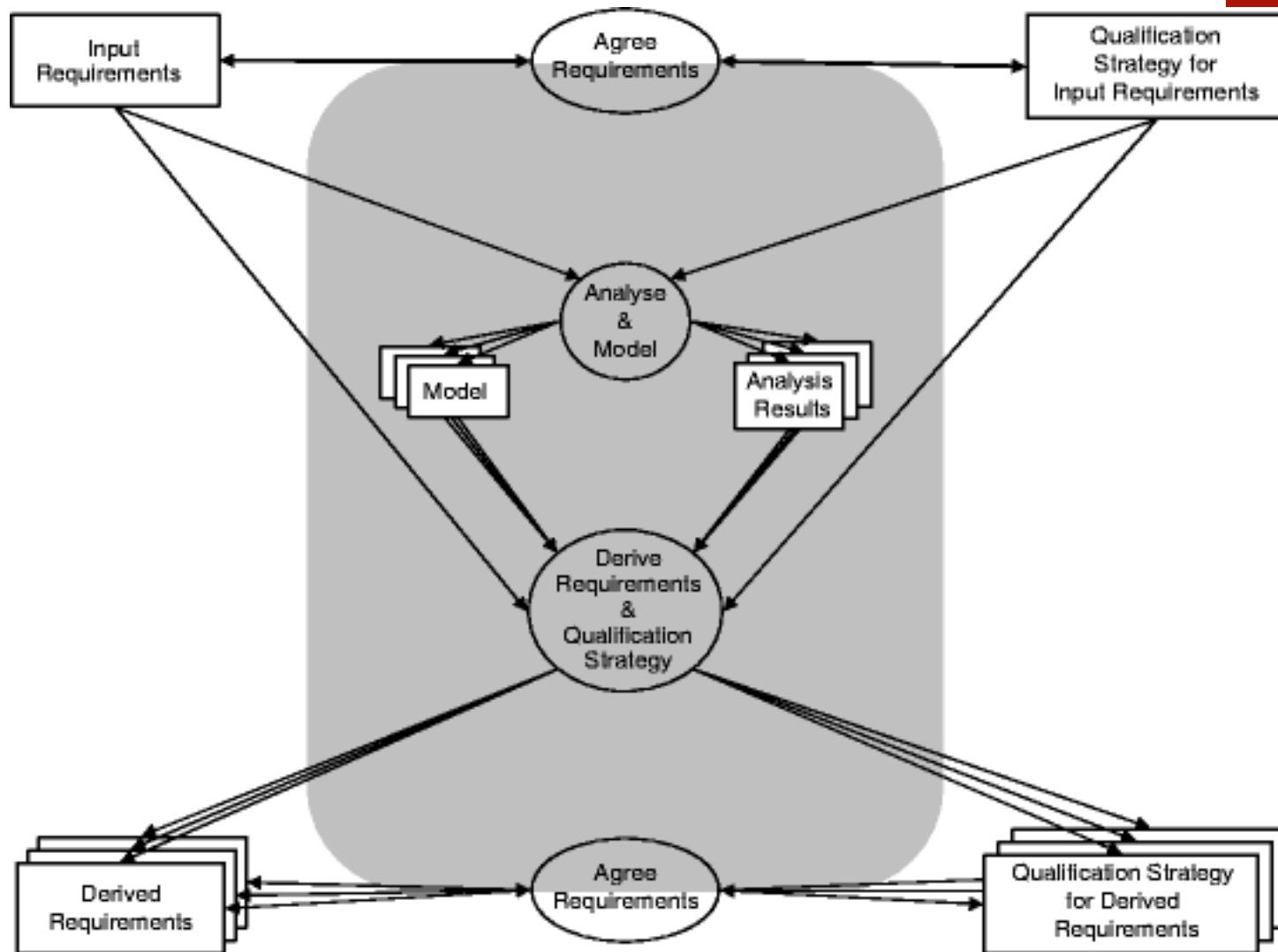
# Input – then derive requirements



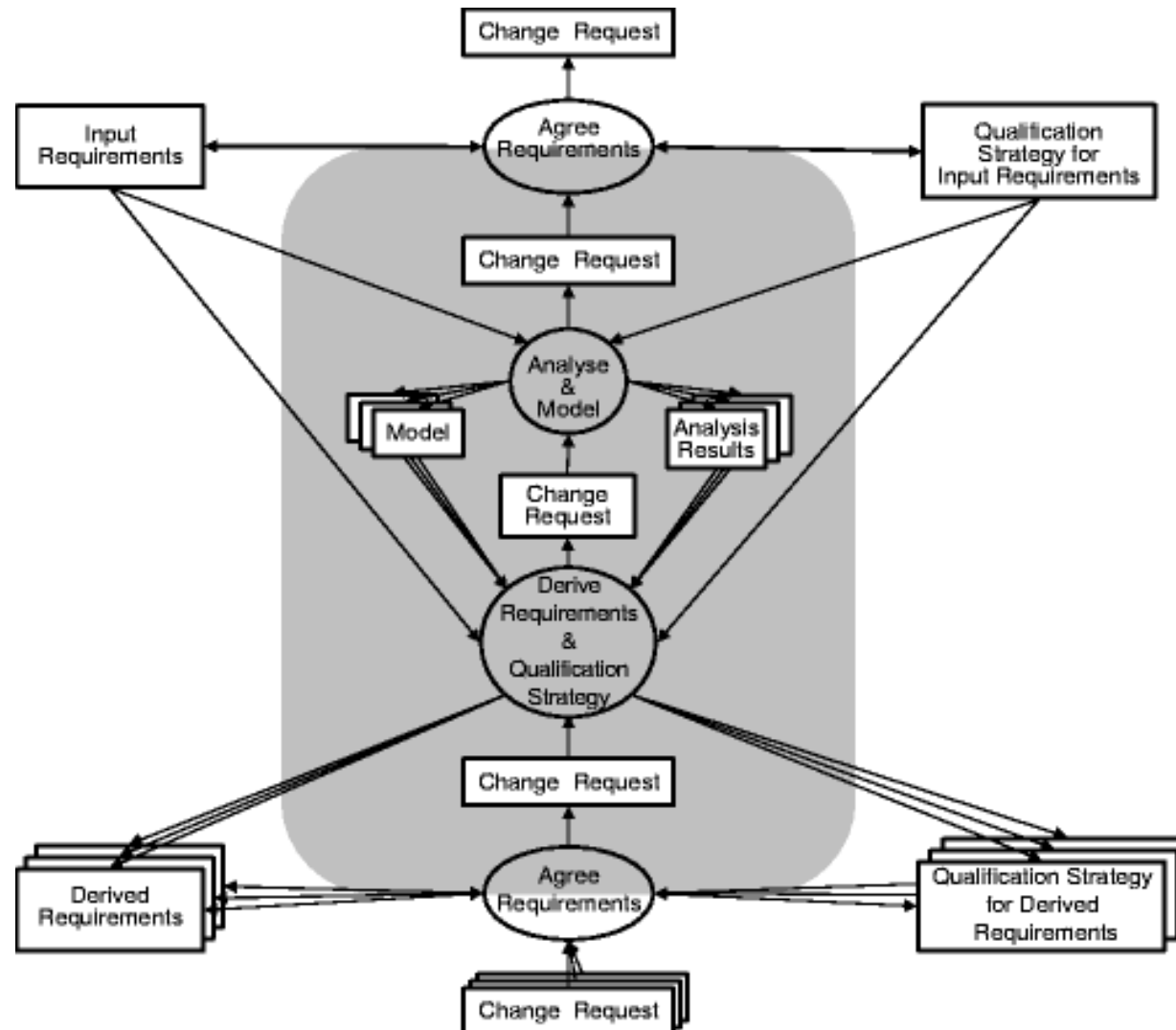
# A General Model of an RE Stage



# Ideal Situation

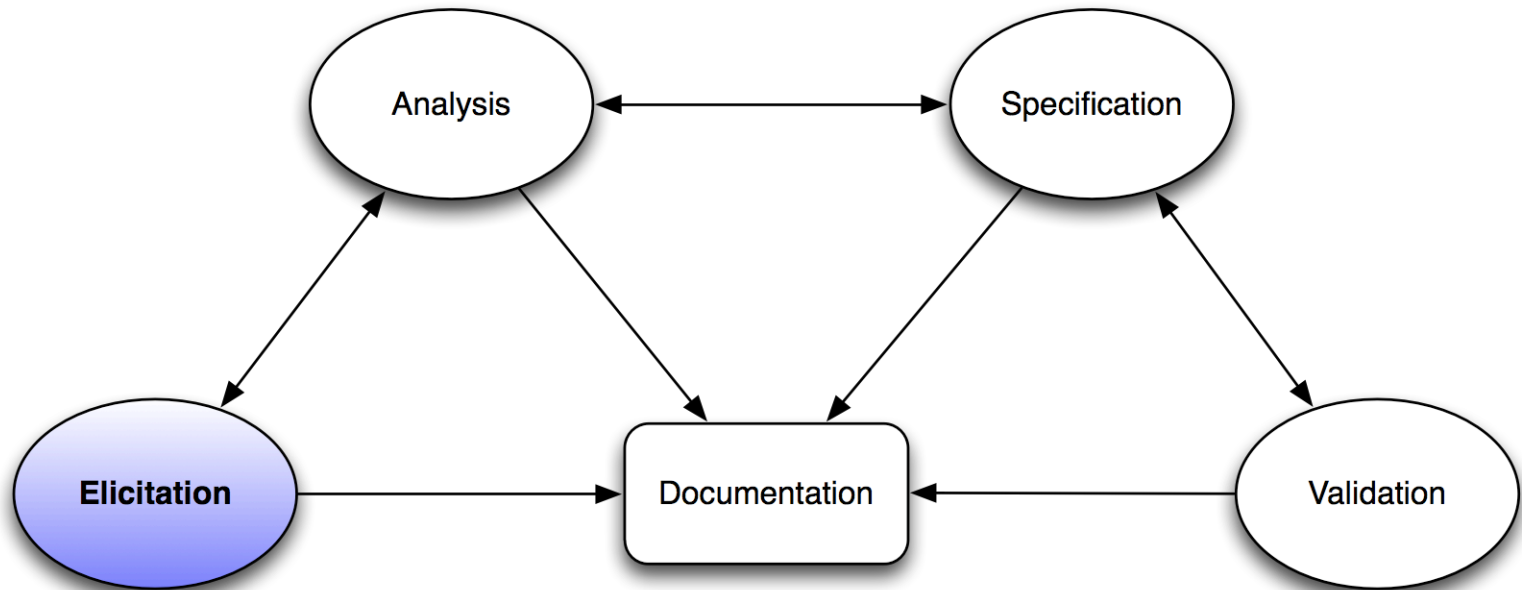


# Real World Situation





# Requirements Elicitation

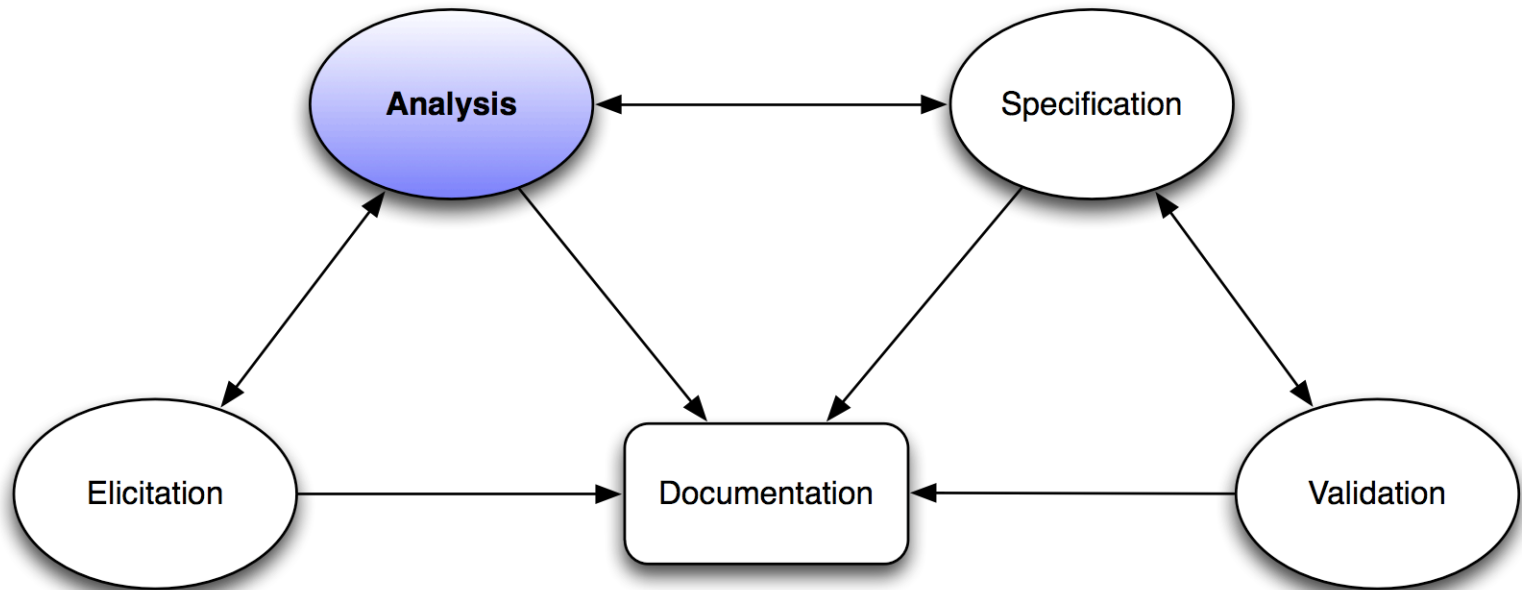


# Requirements Elicitation



- What it is:
  - Collecting information to identify problems and opportunities
  - Finding out information about the software to built, including
    - The application domain
    - The environment in which it will be used
- Why it is hard:
  - Thin spread of domain knowledge, tacit knowledge
  - Limited observability
  - Training in techniques for elicitation
  - Poor tool support for elicitation process
- What techniques are used:
  - Interviews
  - Rapid Application Development (RAD) workshops
  - Scenario-based methods

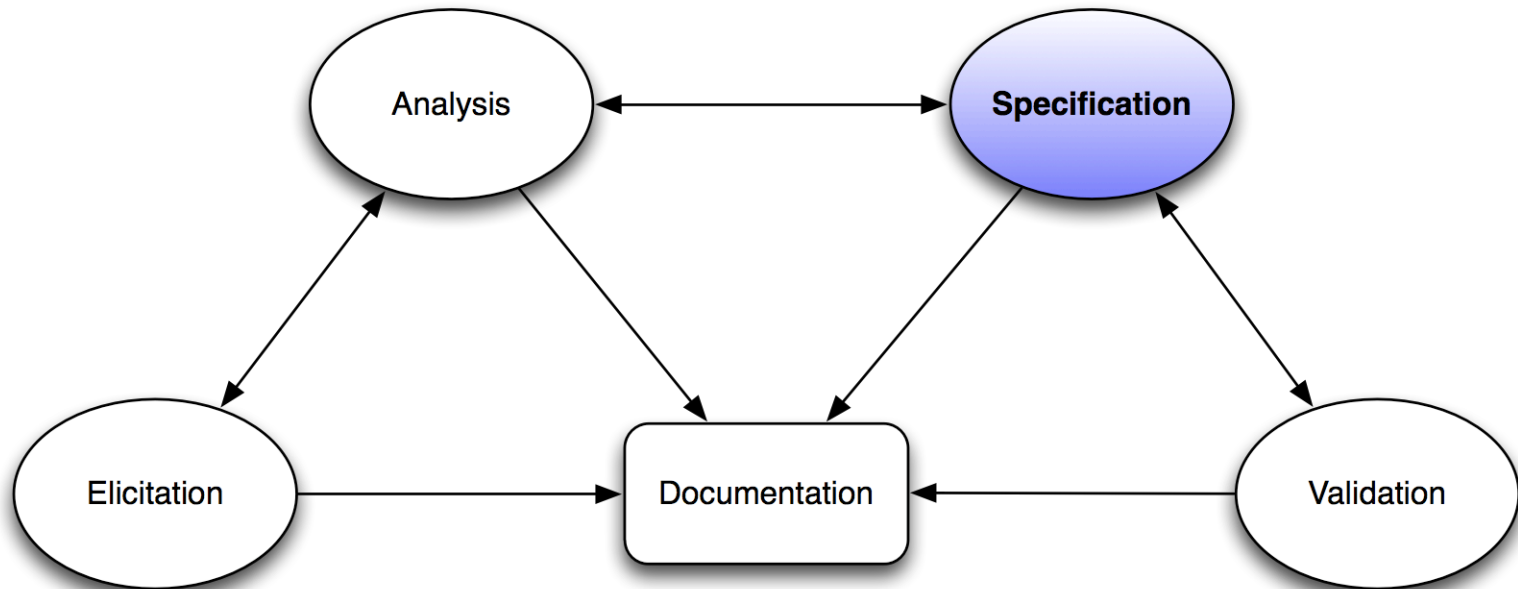
# Requirements Analysis



# Requirements Analysis

- What it is:
  - Building models of requirements that are amenable to evaluation of its properties
  - Identifying conflicts between requirements (stakeholders)
- Why it is hard:
  - Formally modelling natural language requirements is an art form
  - Many requirements conflict, especially when many stakeholders are involved
- What techniques are used:
  - (Diagrammatic) modelling languages, e.g., UML (Unified Modelling Language), SCR (Software Cost Reduction) tables, ...

# Requirements Specification

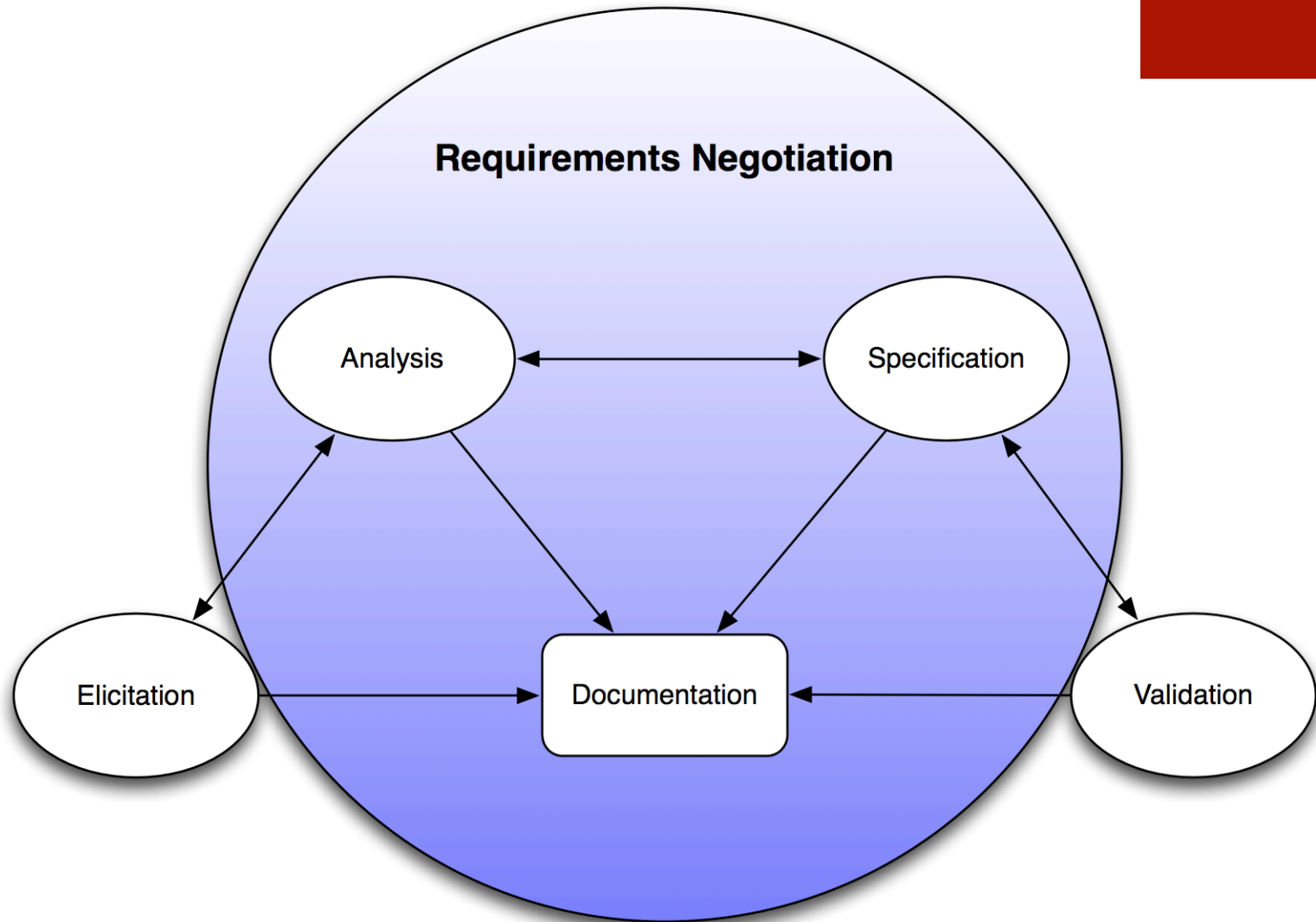


# Requirements Specification



- What it is:
  - Description of what users need to be able to do with the system
  - Description of what the system must do for the stakeholders involved.
  - Description of the qualities that the system user/system functionality must have.
- Why is it hard?
  - Correctness in specification is a hard problem.
  - Completeness is impossible to check.
- What techniques are used?
  - Requirements matrices
  - Template guided requirements

# Requirements Negotiation



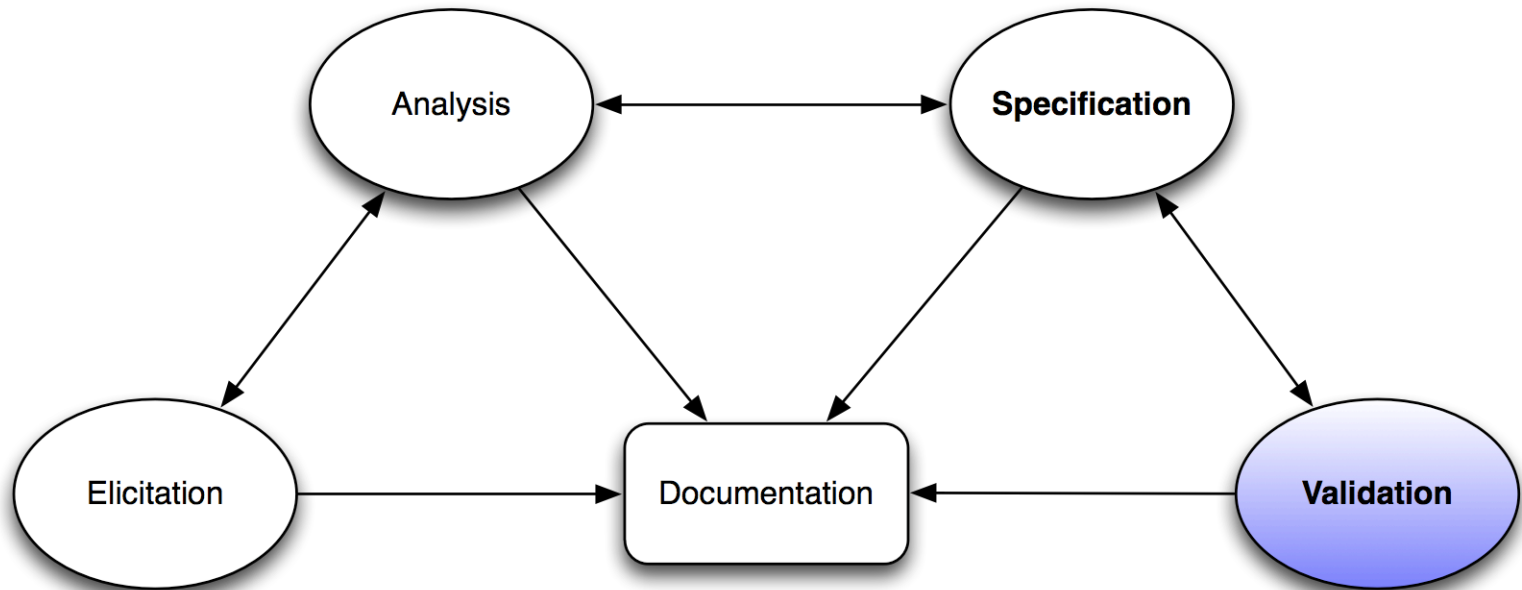
# Requirements Negotiation



- What it is:
  - When stakeholders have conflicting requirements, “getting to yes” can be a challenge
- Why is it hard?
  - Negotiations require deep understanding of the domain and the stakeholders.
  - Negotiations require identification of real requirements vs. “fake requirements”.
  - Some conflicts may never be resolved.
- What techniques are used?
  - Consensus building
  - Majority rule
  - Appeal to Authority



# Requirements Validation

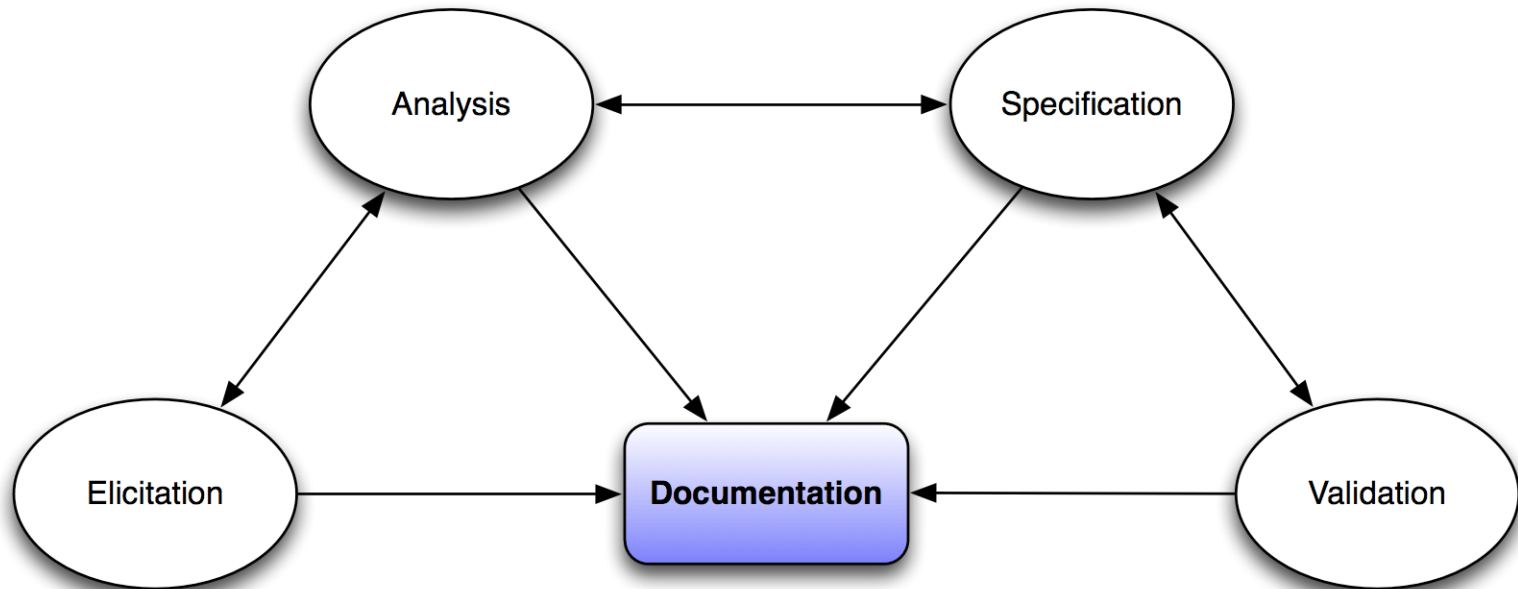


# Requirements Validation

- What it is:
  - Checking the requirements documents to make sure that **the right system is built**
- Why it is hard:
  - There is no other formal document to check requirements documents against
  - Getting the requirements wrong can be very costly
- What techniques are used:
  - *Informal techniques:* Reviews, inspections, walkthroughs
  - *Semi-formal techniques:* Prototyping, animation
  - *Formal techniques:* Formal Methods [see FMS module]



# Requirements Documentation



# Requirements Documentation



- What it is:
  - Requirements Documentation is often the one time to **record valuable information** (frequently as the basis of a legal contract)
  - Requirements documents are the key for **communicating requirements**
- Why it is hard:
  - **Requirements documents need to be communicated across (contractual & language) boundaries**
- What techniques are used:
  - Employing a **requirements engineer** to facilitate the process and to avoid common errors
  - Controlled natural language
  - Automatic generation

# Process Maturity

- Maturity of a company's RE process can be classified on the basis of the **Capability Maturity Model**
- Levels of maturity:
  1. Initial
    - Ad hoc
    - Requirements problems are common
  2. Repeatable
    - Standardised requirements engineering
    - Fewer requirements problems
  3. Defined
    - Process defined based on best practice
    - Process improvement in place

[Sommerville and Sawyer 1997]

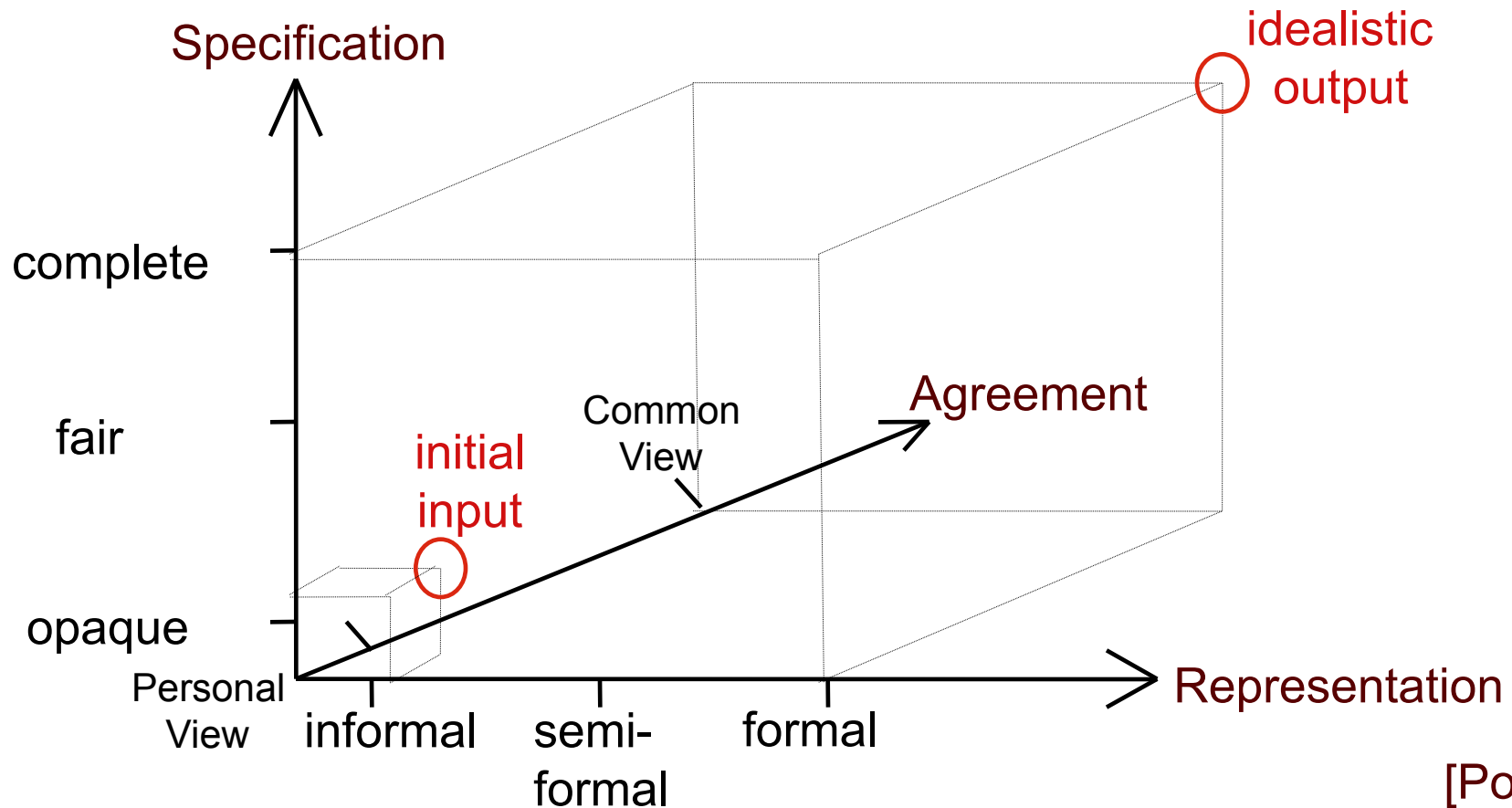


# Recognise the Diversity of Software Projects



- Requirements for large embedded systems are quite different from those for small information systems
  - Complex/Simple, long time-scale, budget
- Requirements for interactive technologies are different than those for non-interactive systems
  - More/less knowledge of user goals required; less/more autonomy in the system;
- Since the problems are diverse, so will be the solutions
  - Different approaches to requirements elicitation
  - Different approaches to requirements analysis
    - Formal methods, MDA, UCD
  - Different distances between requirements and software

# An Idealistic RE Domain Model



[Pohl 1994]

# Types of RE Projects

Where to move within the domain model in practice depends on:

- Source of requirements
  - Customer driven – specific software for a specific customer type
  - Market driven – software to be sold in the market
  - Hybrid – specific customer first, market software eventually
  - User driven – specific software for a group of users
- Nature of the product
  - One-off (‘bespoke’) vs. packaged (‘shrink wrapped’)
  - Single system vs. product family (‘product line’)
  - New system vs. upgrade from existing system





# Some Clarification on Terminology



- A **notation** is a representation scheme/language for expressing things, e.g., dataflow diagrams, the UML
- A **technique** prescribes how to perform a particular activity, and how to describe a product of that activity in a particular notation, e.g., use case diagramming
- A **method** provides a technical description for how to perform a collection of activities, e.g., the Rational Unified Process™
- A **process model** is an abstract, organised description for how to conduct a collection of activities, focusing on resource usage and dependencies between activities
- A **process** is an enactment of a process model, describing the behaviour of one or more agents and their management of resources; a process transforms inputs to outputs

Be aware that different textbooks may employ different terminology!

# Summary

- Requirements are the basis for ensuring that we are constructing the “right” (desired) system
- Requirements documents/specifications are needed to obtain agreement about what a system should do, and to which quality
- Requirements engineering is the task of creating agreed requirements documents/specifications
- Requirements engineering has many aspects: to elicit, represent, record, analyse, manage, ...



# Reading

- Hull et al. (2011) Chapter 2

