

# Lecture 3

# Topics on

# Requirements Engineering

Some material taken from the Tropos project at U of T

Copyright © Yijun Yu, 2005

## Course information

# Let's vote ...

- Course Project/Final Exam  
50-50  
or 60-40?
- Midterm/Final Exam  
15-35  
or 0-50?
- Final Exam  
Open notes  
or Close notes  
or Aid Sheet

Last tutorial ...

# Web Services

- Why we use Web Services?
- Key standards: SOAP, WSDL and UDDI
- Tomcat/Axis implementation of a legacy OmniEditor web service
- Architectures of the OmniEditor
- *Requirements specifications*

Last lecture ...

## Software Reengineering

- Reasons to reengineering
- The *horseshoe* process model
- “Overloaded” words ...  
reengineering, reverse engineering,  
restructuring and refactoring
- There is another RE\*ING in SE:  
*Requirements engineering*

Today ...

# Topics on Requirements Engineering

1. What are Software Requirements?
2. Why are they important?
3. How to engineer the requirements of a software, ...?
4. Why shall we do goal-oriented requirements engineering? Goal models
5. Summary

# 1. What are software requirements?

- Definition from Google: [define:Software Requirements](#)
- The set of functions, performance measures, and constraints that software must satisfy.
- A more or less formal statement of what a software application should do. Sometimes business analysts create requirements and hand them to software developers. Other times software analysts interview business people in order to determine the requirements for a software application development effort. Business people invariably define requirements less formally than necessary. Business people tend to define requirements with written statements or with process diagrams. Software developers are more likely to define software requirements by means of Use Case Diagrams or Class Diagrams, which often aren't that clear to business analysts. Software Requirements constitute an important interface between business managers and IT organizations. If the handoff isn't clear and precise then the resulting system is likely to disappoint the business people who requested it.

# What are software requirements, then?

- Requirements are expectations of the system by the environment: what problem is solved?
  - Software refers to the software plus the system/platform where it is running
- Requirements to the software product
  - Functionalities: functional requirements
  - Qualities: non-functional requirements  
Reliability, Correctness, Usability, Performance, Security, Privacy, ...
- Requirements to the software development process
  - Productivity: How fast you deliver functionalities?
  - Maintainability, Understandability, Reusability, etc.:  
How good you can maintain the product qualities?

# Related Requirements

- System requirements specify the minimal demands (dependency) to the environment (hardware/software/people)

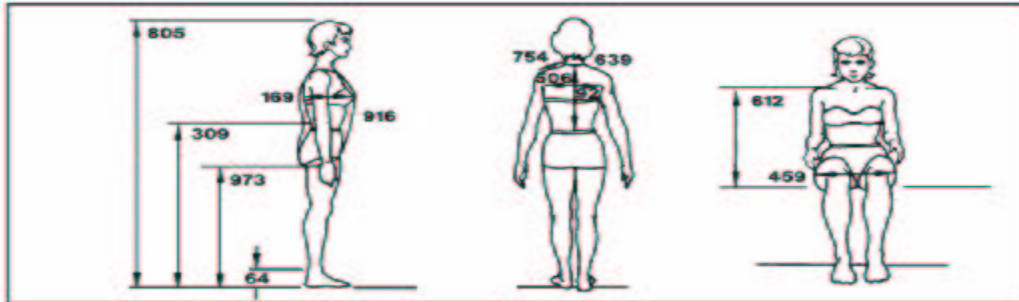
“Windows 3.1/95/98/NT/XP, 256MB, English”

“Platform independent”? ...

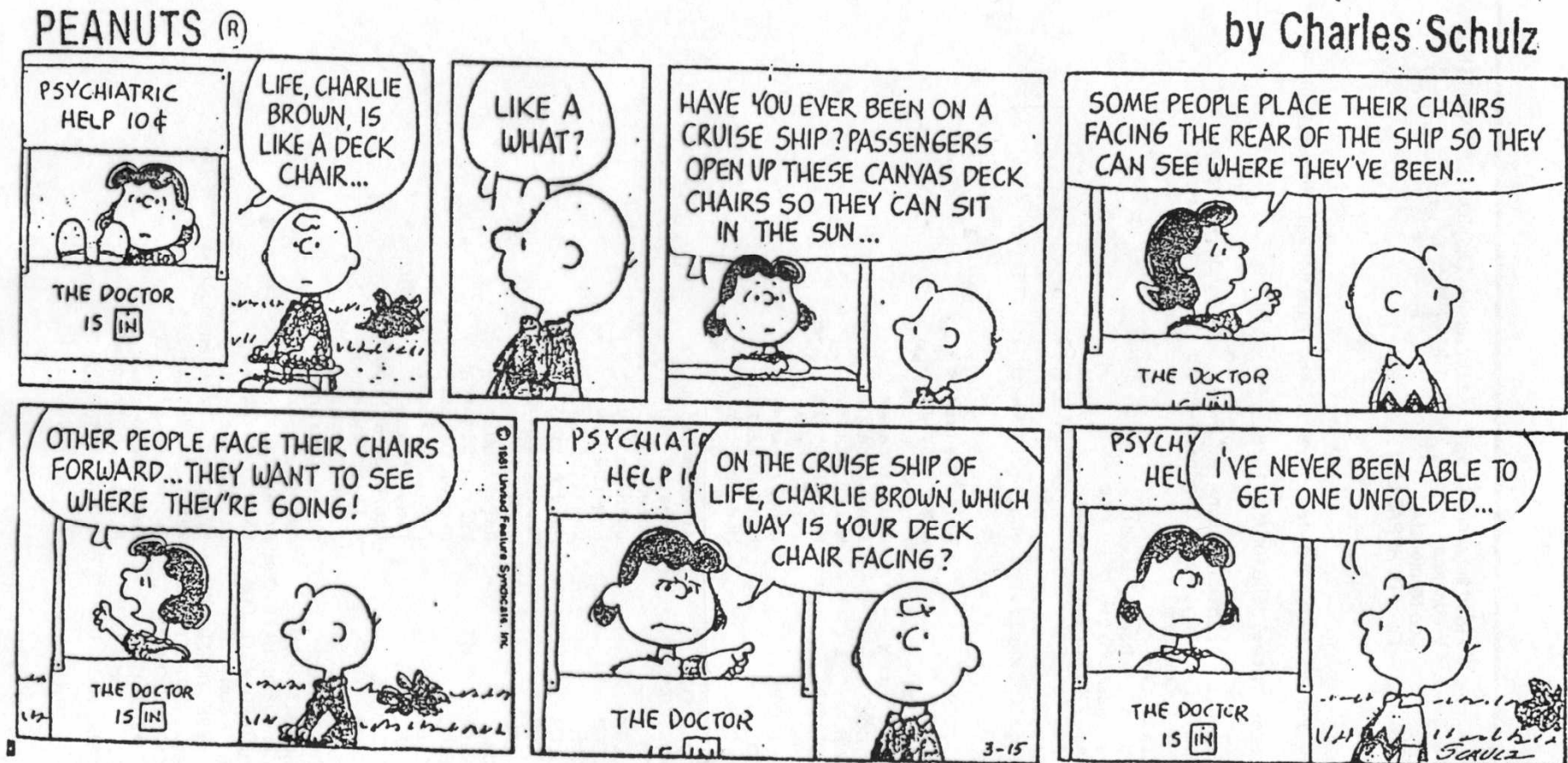
- Stakeholder Requirements specify the expectations from different agents in the world  
Domain engineers, End-Users, Developers, Managers, Testers, HCI designers, Administrators, Partners, Competitors, Lawyers, Artists, you name it ...
- Business Requirements  
Market, ROI, Profit margin, Market share, Organizations



# Example system requirements not everyone can be an astronaut

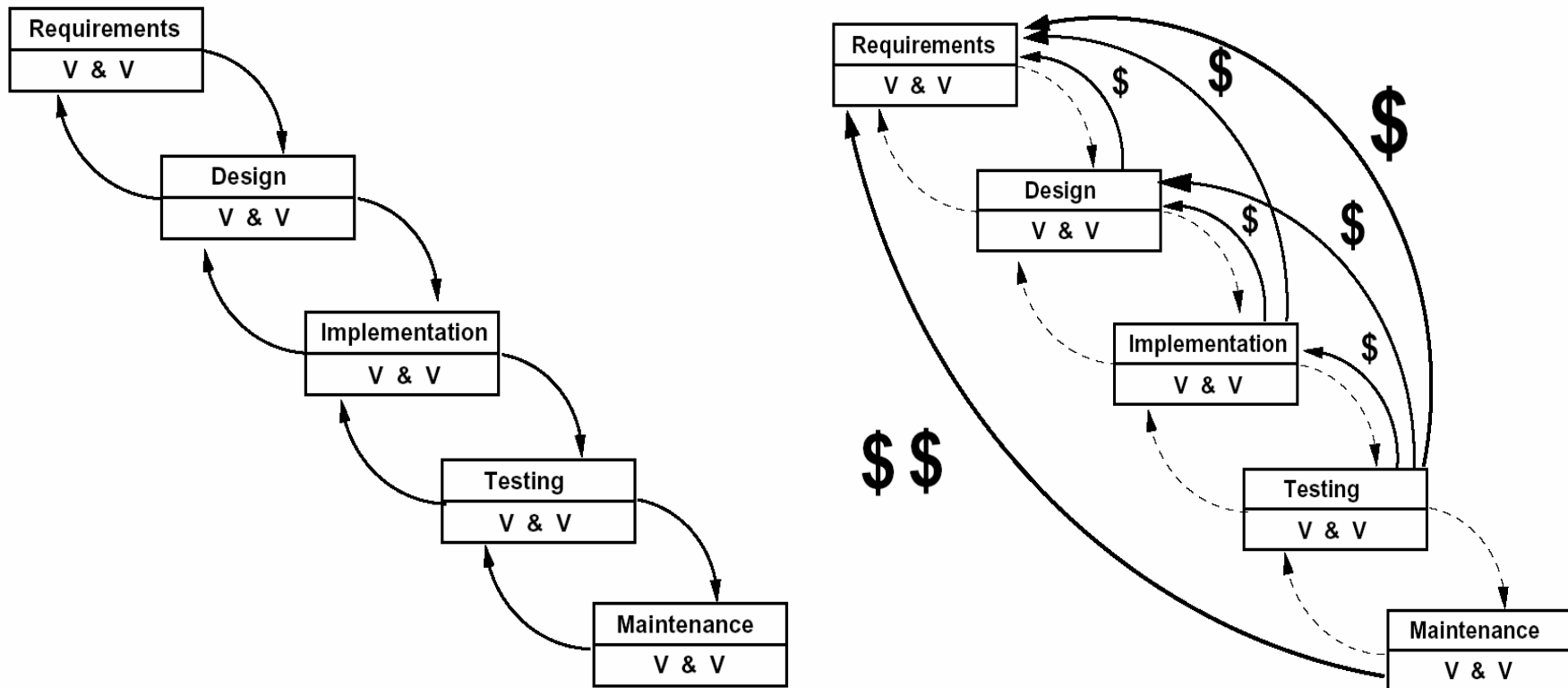
DOORSnet   Search   Sort   Display   Help	
Explore   Preferences   Print   Get entire module   Log out	
MSIS Requirements : MSIS (271 to 300 of 12243)	
3.3.1.3	■ <b>3.3.1.3 Body Size Data Design Requirements</b>
3.3.1.3	■ {A}
3.3.1.3	■ Dimensions of the year 2000, 40 year-old White or Black American male and the 40 year-old Asian Japanese female are given in Figure 3.3.1.3-1. The data in this Figure shall be used as appropriate to achieve effective integrations of the crew and space systems. The dimensions apply to 1-G conditions only.
3.3.1.3	■ Dimensional data estimates for the year 2000 White or Black American female crewmember cannot be specified at this time due to insufficient data.
3.3.1.3	■ (Refer to Reference 16, Chapter III, Appendix B, for dimensional data for the 1985 American female).
3.3.1.3	■ {A}
3.3.1.3	■ <b>Figure 3.3.1.3-1 (1 of 12) Body Size of the 40-year-Old American Male and 40-Year-Old Japanese Female for Year 2000 in One Gravity Conditions (Continued)</b>
3.3.1.3	■ 

# Requirements have dependencies and Reengineering needs to know about the Requirements



2. Why are requirements important?

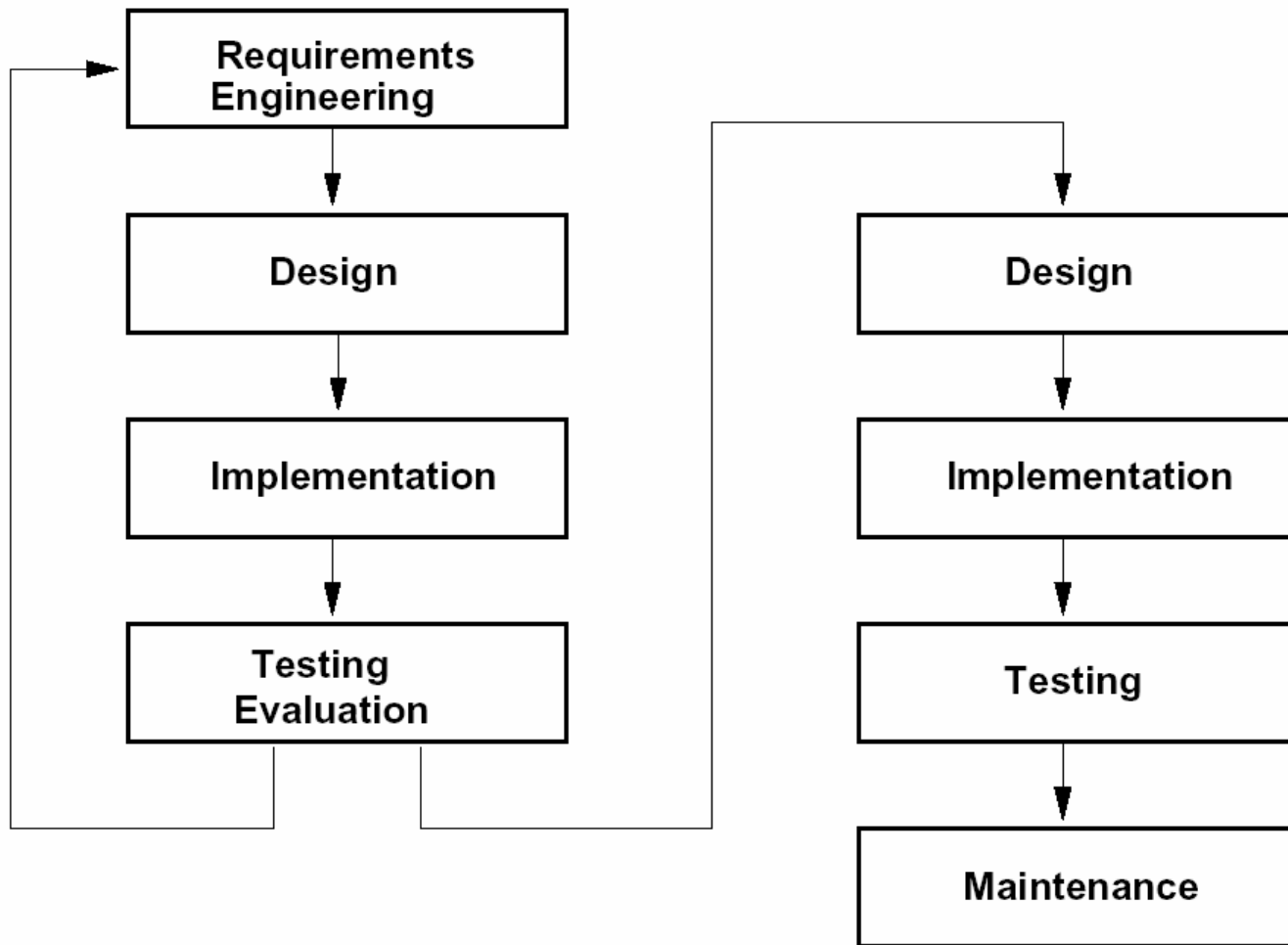
# The Waterfall process model



IT SAVES DOLLARS, IT SAVES LIVES

### 3. How to obtain requirements?

# Rapid Prototyping process



# 3. How to specify FR?

- A functional requirement
  - Goal: query [stock quote]
  - Inputs: stock quote [string]
  - Outputs: stock price [float]
  - Precondition: stock quote is not empty
  - Postcondition: stock price  $\geq 0$  if the stock quote is found, otherwise stock price = -1
- Relation to other requirements
  - To make profit of investment (why?)
  - To invoke an XMETHODS web service (how?)
  - Investor (brokers), Stock analysts (who?)
  - 9am – 5pm EST (when?)
  - Stock portfolio (what?)
  - Sometimes Helps, sometimes Hurts the profit goal (how much?)

# An alternative requirement

- An alternative functional requirement
  - Goal: query [stock name]
  - Inputs: stock name [string]
  - Outputs: stock price [float]
  - Precondition: stock name is not empty
  - Postcondition: stock price  $\geq 0$  if the stock name is found and unique, otherwise stock price = -1 if the stock doesn't exist, or stock price = -2 if more than one stock is found
- Relate to other requirements
  - To make profit of investment (why?)
  - Do not need to remember the stock quote (why?)
  - To invoke another XMETHODS web service (how?)

# 3. How to specify a NFR?

- A non-functional requirement
  - Quality attribute: responsiveness [query]
  - Metric: elapsed time to get response
  - Satisfaction criteria: elapsed time < 1 second
- Another non-functional requirement
  - Quality attribute: usability [query]
  - Metric: time to memorizing the name
  - Satisfaction criteria: memorizing the name < 1 second
- Quality attribute, metrics, satisficing criteria

### 3. How to obtain requirements?

## Goal-oriented requirements engineering

- What is a goal? *Desired* state of the system.  
Captures *intentions* or *objectives*
  - Either true (satisfied) or false (denied)
  - *Partially/Fully* satisfied/denied? Soft-goal: *Satisficed*
- Reveal the rationale behind the requirements, called “early requirements”
  - Goal-oriented requirements elicitation (asking why, how, who, what, when, where and how much ...)
  - Goal-oriented requirements specification: goal modelling to define the inter-dependencies among requirements

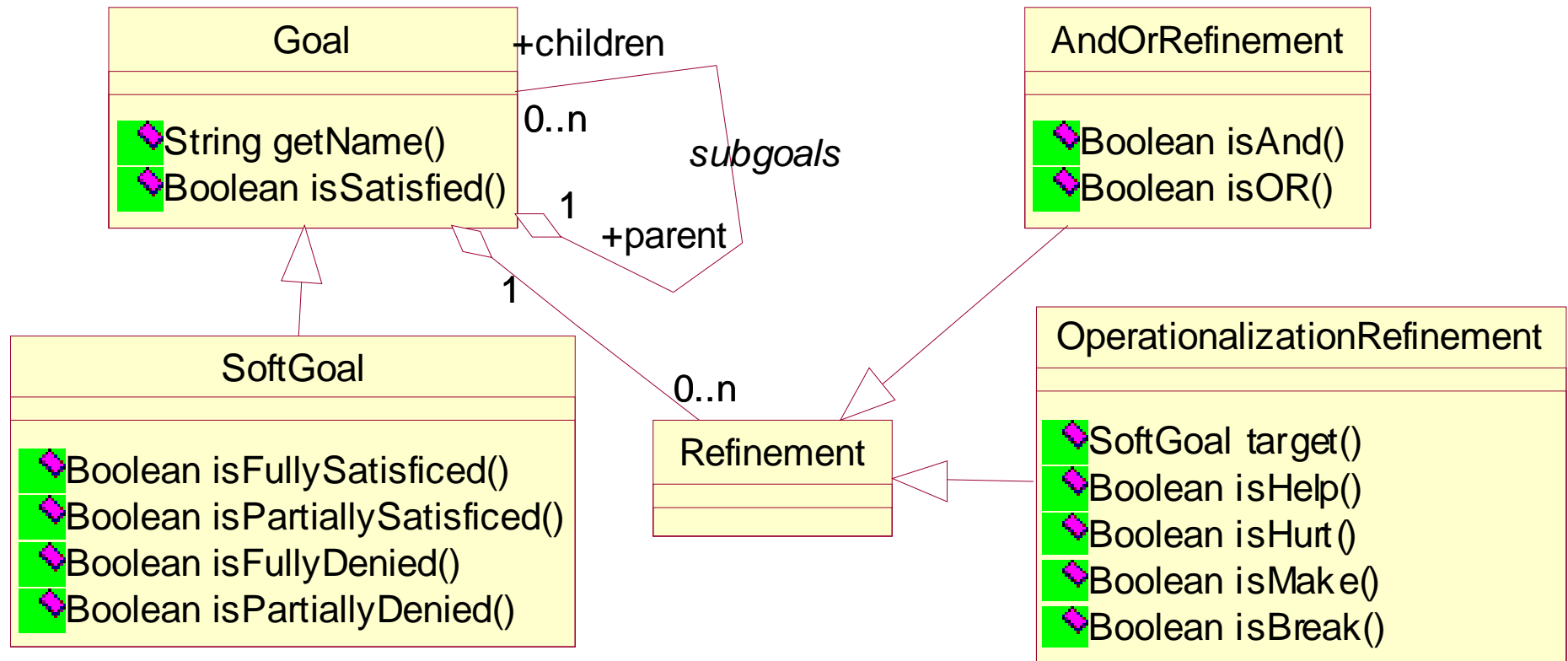


## 4. Representation issues:

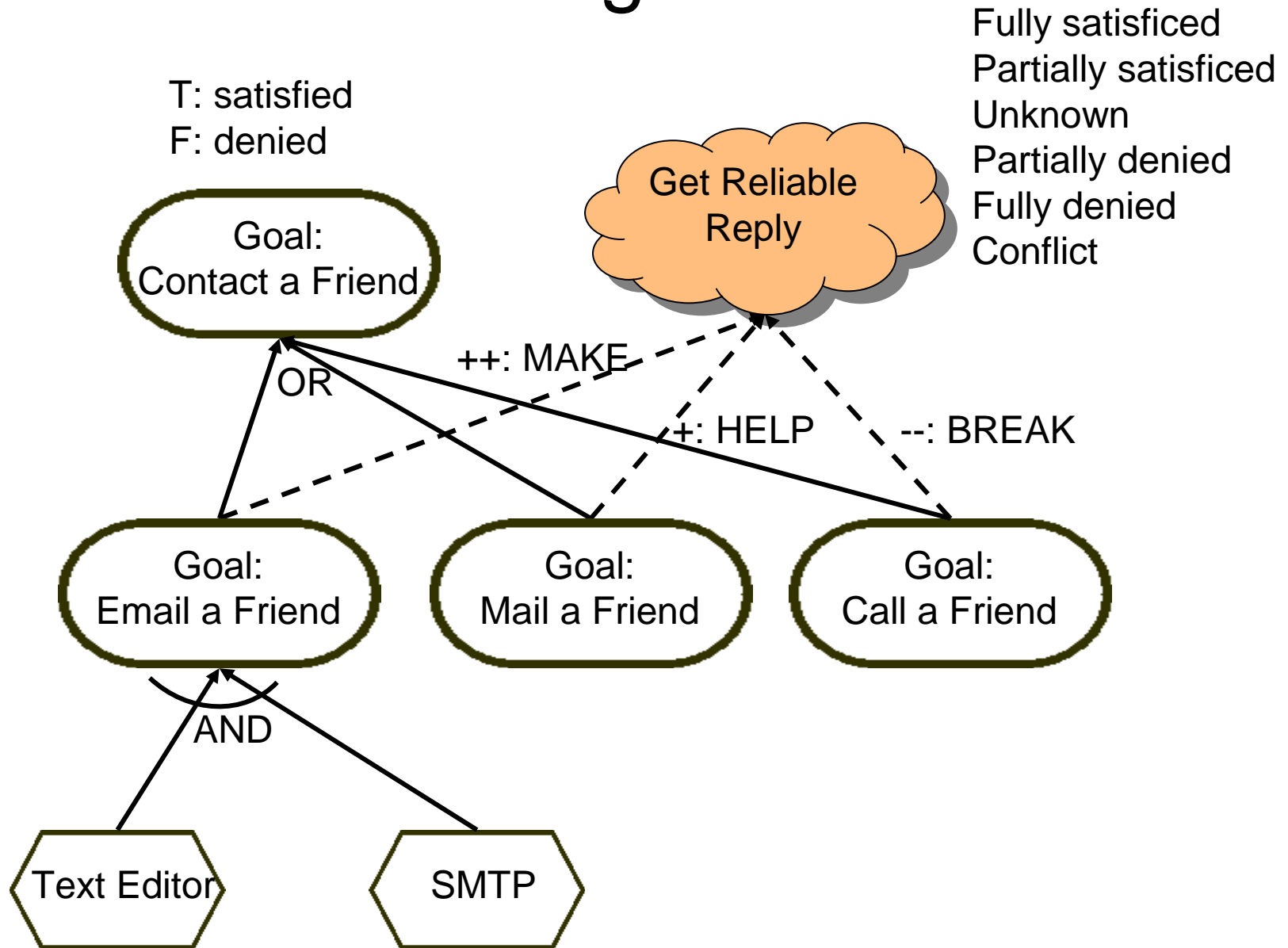
### Conceptual modelling

- Each functional requirement has an associated goal, like the “*@purpose*” statement in the Javadoc, which defines the function: What is the acceptable input and what is the exceptional input? What is the expected output?
- Each non-functional requirements has an associated softgoal, and the contribution to the satisficing of the softgoal through a criteria on a threshold of the metric: *Operationalization*
- *Dependencies among them  
(AND/OR contributions and  
HELPS/HURTS/MAKES/BREAKS correlations)*

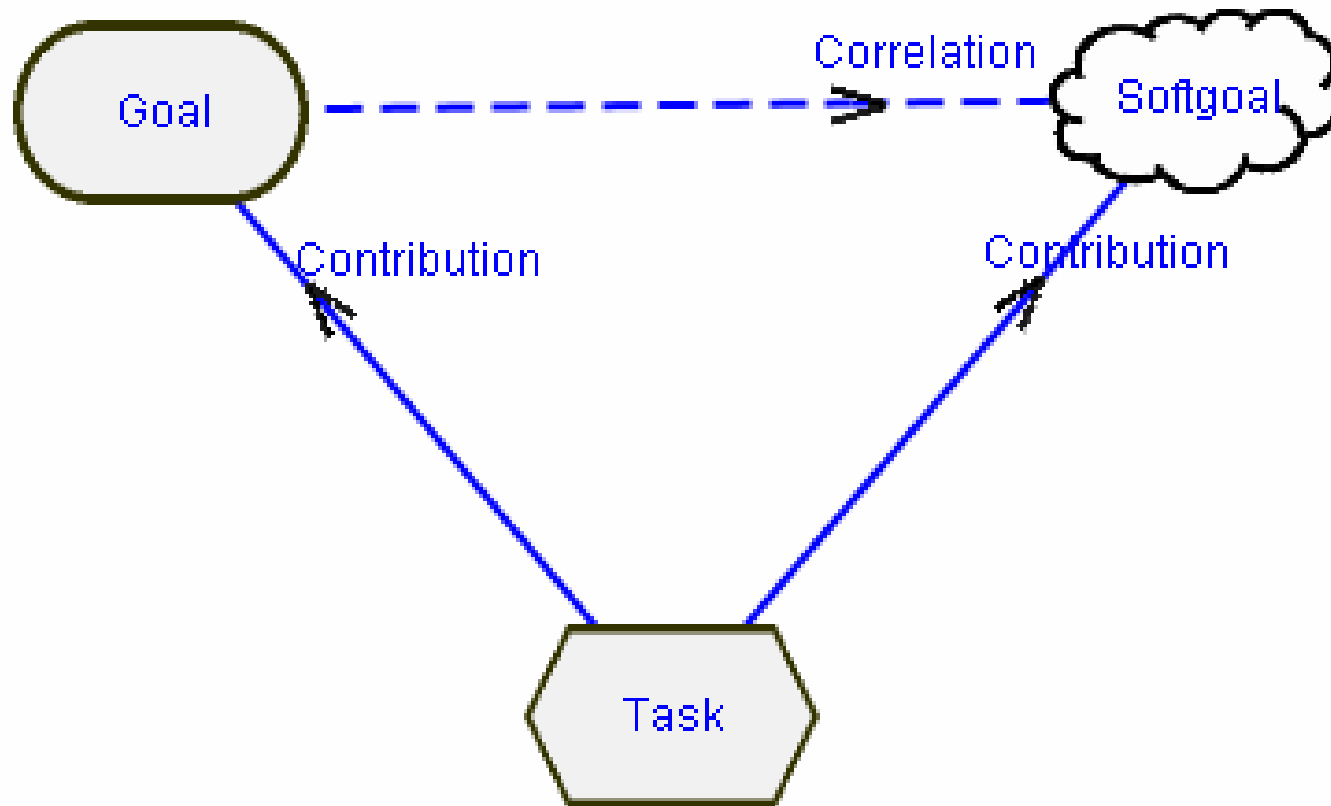
## 4. The goal model: a syntax

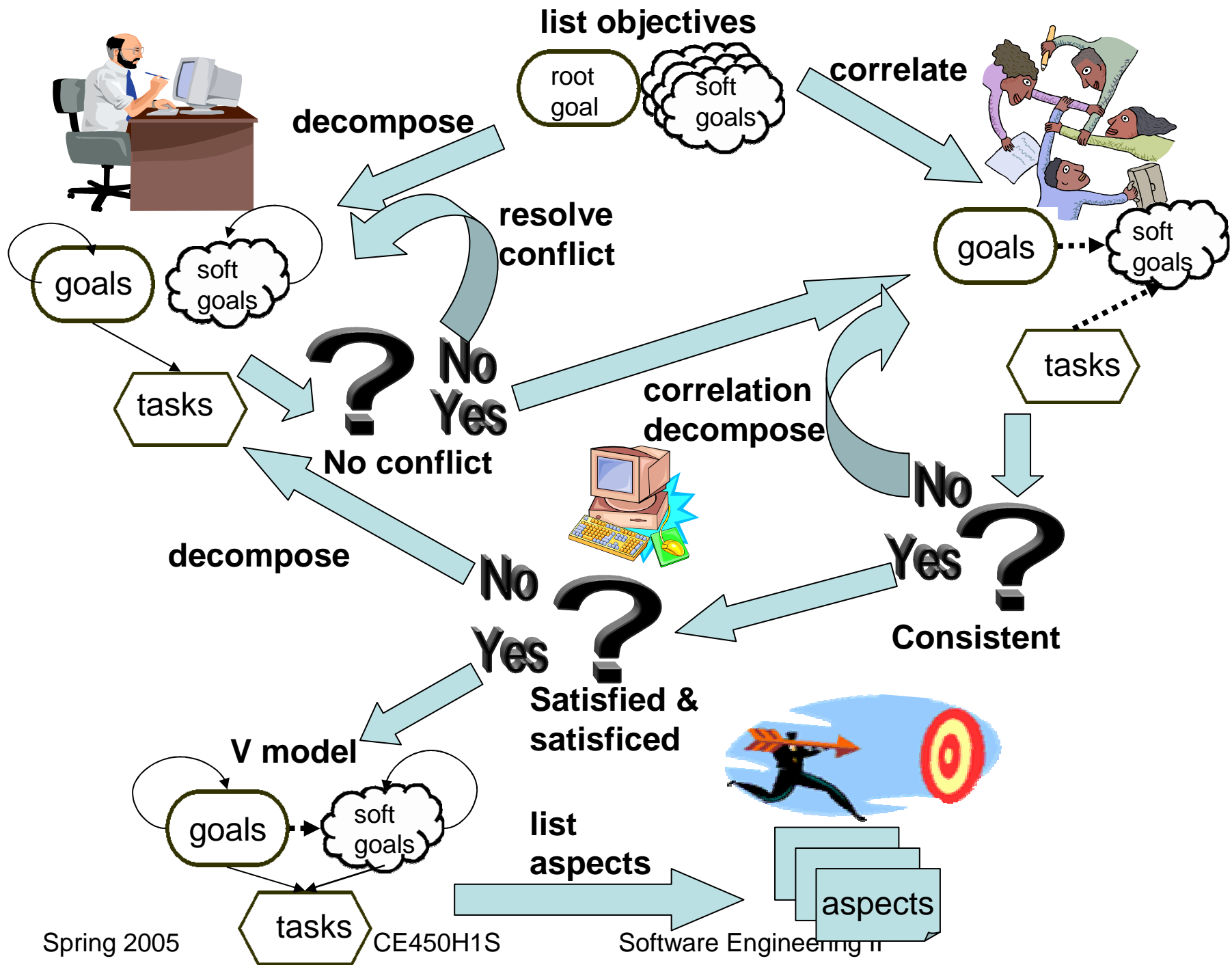


# 4. Goal reasoning: the semantics



## 4. V-graph: the pragmatics of a goal model





# 5. Summary

- RE is getting more important
- FR and NFR are explained
- Goal models are used to model early requirements, followed by software architectures, UML class diagrams, design patterns, refactoring, etc.
- The syntax/semantics/pragmatics of a goal model are explained, also with a process for goal oriented requirements engineering
- Three related tutorials will further explore the topic:
  - The OpenOME requirements engineering tool
  - Aspect-oriented programming (AOP) and the use of goal model to find aspects in the early requirements
  - Quality metrics and software measuring tools

# Further readings

- A. Dardenne, A. van Lamsweerde, S. Fickas: Goal-Directed Requirements Acquisition. *Science of Computer Programming*, 20(1-2): 3-50 (1993).
- A. van Lamsweerde, L. Willemet. “Goal-Oriented Requirements Engineering, A Guided Tour”, *International Conference on Requirements Engineering*, 2001.
- L. Chung, B. A. Nixon, E. Yu and J. Mylopoulos. “*Non-Functional Requirements in Software Engineering*”, Kluwer Academic Publishing, 1999.
- J. Mylopoulos, L. Chung, E. Yu. “From Object-Oriented to Goal-Oriented Requirement Analysis”. *Communications of the ACM*. 42(1):31-37, January 1999.
- P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani. “Reasoning with goal models”. *LNCS*, 2503:167--181, 2002.
- L. Liu, E. Yu, and J. Mylopoulos. “Security and privacy requirements analysis within a social setting”. In *RE 2003*, pp. 151--161, 2003.
- Y. Yu, J.C. Leite, J. Mylopoulos. “From goals to aspects: discovering aspects from goal models”. *RE'04*, 2004.

# What's next ...

- A Tutorial on a Requirements Engineering tool: OpenOME
- Next lectures will explain design patterns and refactoring techniques