Introduction to Software Engineering

What is Engineering?

Systematic application of scientific knowledge in creating and building cost effective solutions to practical problems

Software Engg. - A Problem Solving Activity

- *Analysis:* Understand the nature of the problem and break the problem into pieces
- *Synthesis:* Put the pieces together into a large structure

Software Engg. - A Problem Solving Activity

For problem solving we use

- *Techniques (method):*
  - formal procedures for producing results using some well-defined notation
- *Methodologies*:
  - collection of techniques applied across software development and unified by a philosophical approach
- *Tools:*
  - instrument or automated systems to accomplish a technique

Software Engineering: Definition

Software Engineering is a collection of techniques, methodologies and tools that help with the production of

- a high quality software system
- with a given budget
- before a given deadline

while change occurs

Software Production has a Poor Track Record

Example: Space Shuttle Software

- Cost: $10 Billion, millions of dollars more than planned
- Time: 3 years late
- Quality: First launch of Columbia was cancelled because of a synchronization problem with the Shuttle's 5 onboard computers.
  - Error was traced back to a change made 2 years earlier when a programmer changed a delay factor in an interrupt handler from 50 to 80 milliseconds.
  - The likelihood of the error was small enough, that the error caused no harm during thousands of hours of testing.
- Substantial errors still exist
  - Astronauts are supplied with a book of known software problems "Program Notes and Waivers"

Why are software systems so complex?

- The problem domain is difficult
- The development process is very difficult to manage
- Software offers extreme flexibility

Dealing with Complexity

1. Abstraction
2. Decomposition
3. Hierarchy

1. Abstraction

- Inherent human limitation to deal with complexity
- Chunking: Group collection of objects
- Ignore unessential details => Models

Models are used to provide abstractions

- System Model
- Task Model
- Issues Model

Models are used to provide abstractions

System Model:

- *Object Model*:
  - What is the structure of the system?
  - What are the objects and how are they related?
- *Functional model*:
  - What are the functions of the system?
  - How is data flowing through the system?
- *Dynamic model*:
  - How does the system react to external events?
  - How is the event flow in the system ?

Models are used to provide abstractions

Task Model:

- *PERT* (Program Evaluation & Review Technique) Chart:
  - What are the dependencies between the tasks?
- *Schedule*:
  - How can this be done within the time limit?
- *Org Chart*:
  - What are the roles in the project or organization?

Models are used to provide abstractions

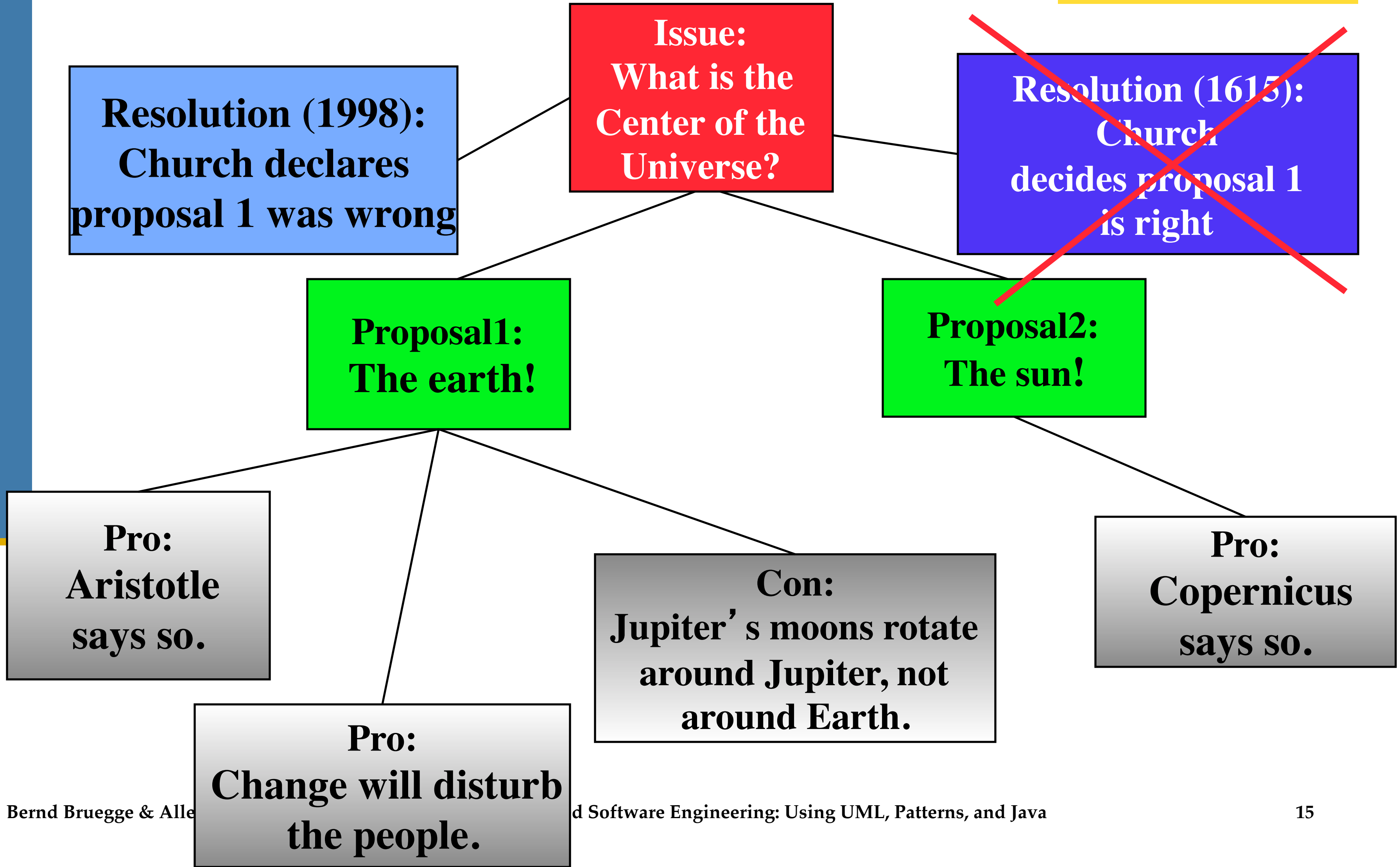
Issues Model:

- What are the open and closed issues?
- What constraints were posed by the client?
- What resolutions were made?

Example of an Issue: Galileo vs. Church

- What is the center of the Universe?
  - Church: The earth is the center of the universe. Why? Aristotle says so.
  - Galileo: The sun is the center of the universe. Why? Copernicus says so. Also, the Jupiter's moons rotate round Jupiter, not around Earth.

Issue-Modeling



Issue:
What is the Center of the Universe?

Resolution (1998):
Church declares proposal 1 was wrong

~~Resolution (1615):
Church decides proposal 1 is right~~

Proposal1:
The earth!

Proposal2:
The sun!

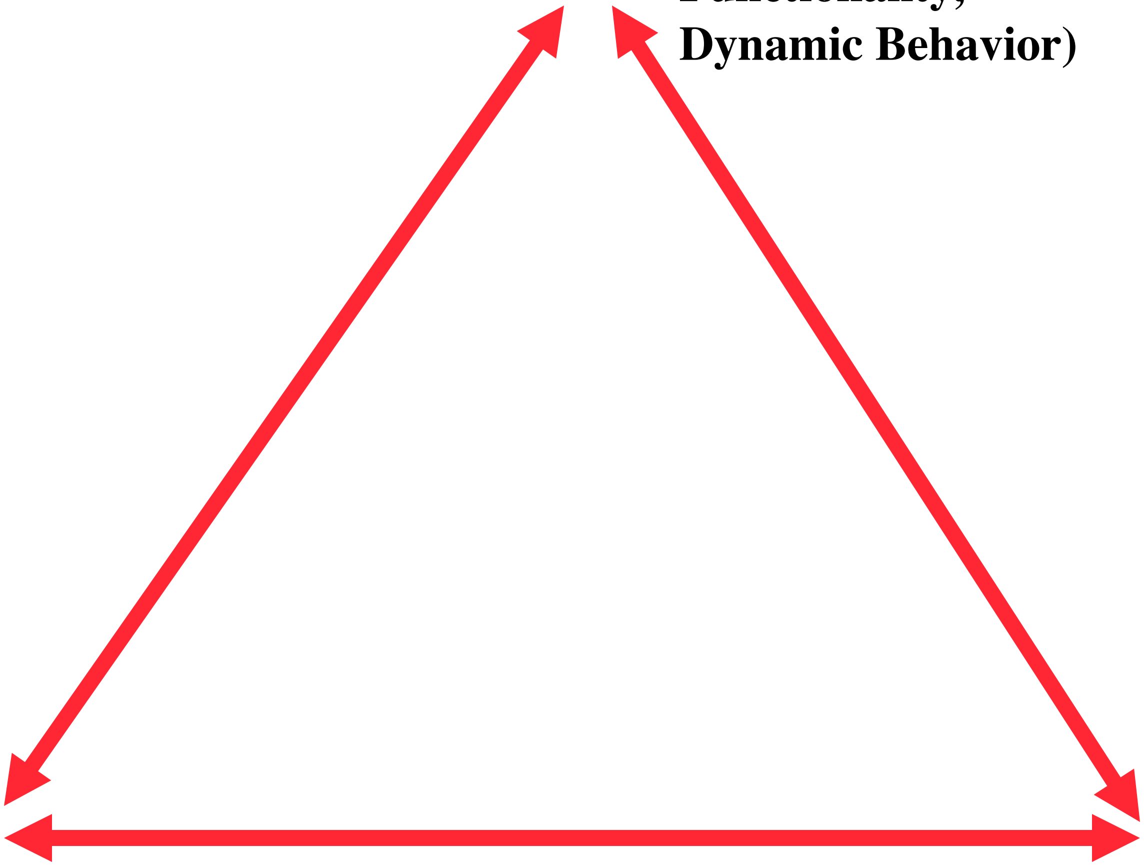
Pro:
Aristotle says so.

Pro:
Change will disturb the people.

Con:
Jupiter’s moons rotate around Jupiter, not around Earth.

Pro:
Copernicus says so.

Interdependencies of the Models

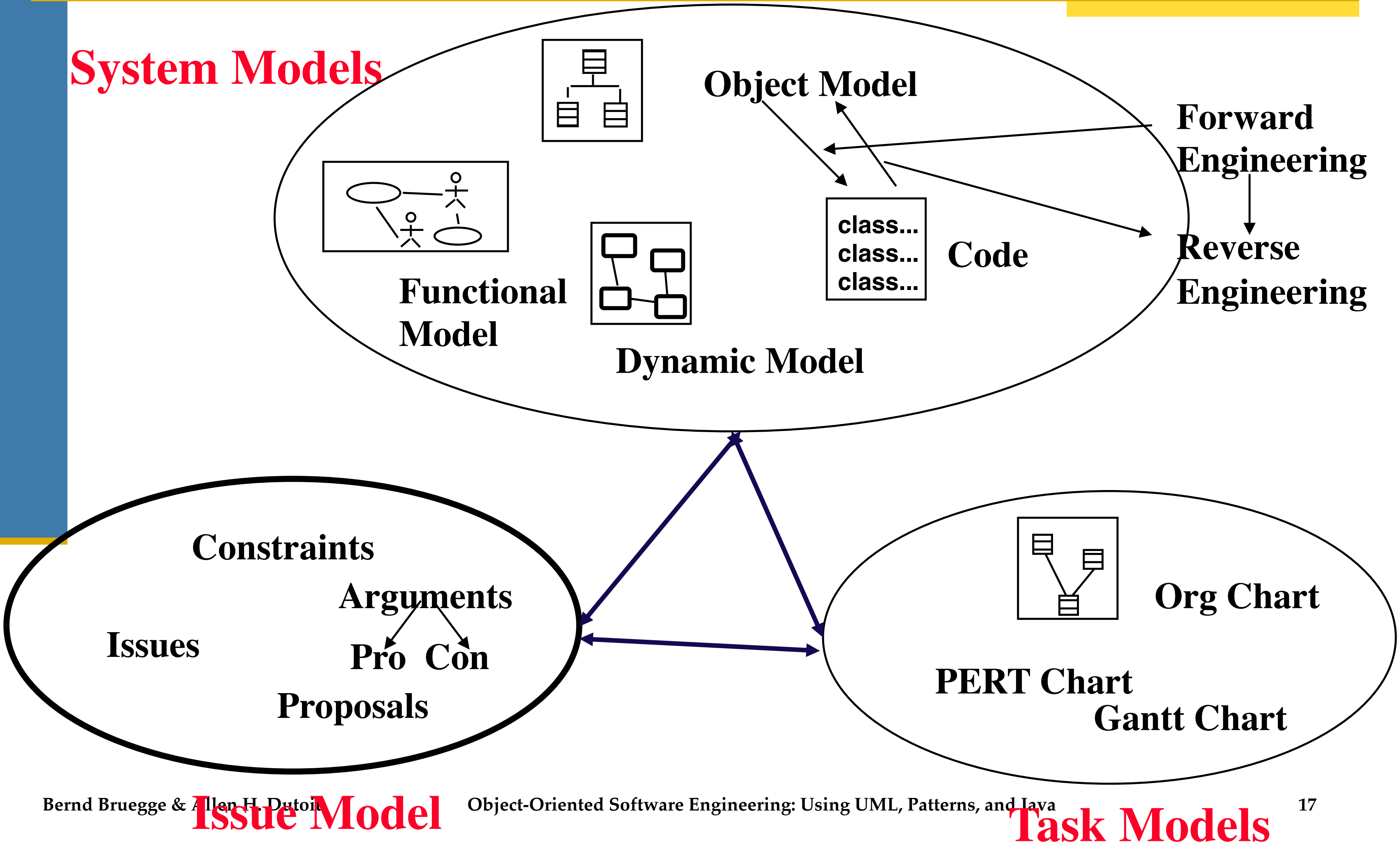


System Model (Structure, Functionality, Dynamic Behavior)

Issue Model (Proposals, Arguments, Resolutions)

Task Model (Organization, Activities, Schedule)

The “Bermuda Triangle” of Modeling



System Models

Object Model

Forward Engineering

Reverse Engineering

Functional Model

Dynamic Model

class...
class...
class...

Code

Constraints

Arguments

Issues

Pro Con

Proposals

Issue Model

Org Chart

PERT Chart

Gantt Chart

Task Models