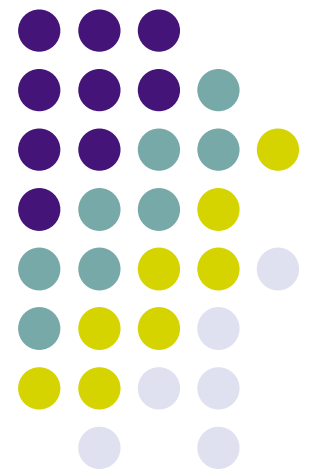


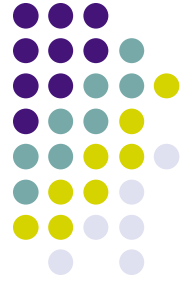
# Software Design and Architecture

---

The once and future focus of  
software engineering

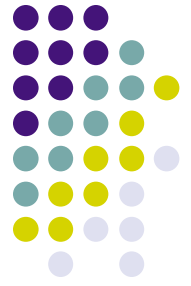
Richard N. Taylor  
André van der Hoek  
University of California, Irvine





# Structure of the Paper

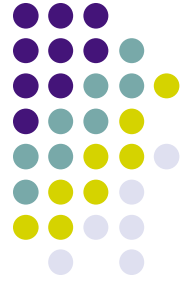
- Introduction
- Paradigms and Persuasions: How design has been approached
  - Prescriptive design methods, Notations, The wisdom of experience, HCI design, Design outside of software, Cognitive and social strategies
- Contemporary Currents
  - Agile methods, Aspect-oriented design, Design analysis, Component-based design, Software architecture
- Design, Designing, and Designers
  - The elements of design research, The community of designers
- Research Directions
  - Directions reflecting good recent progress, Directions from new capabilities, Directions from design imperatives, Directions from examining our past, Directions from looking outside of CS, Barriers to progress
- Challenges and Vision
  - Technical challenges, Community challenges, A vision for the long future



# Structure of the Talk

1. What's the deal with the subtitle?
2. Design of what?
3. Just what is Design?

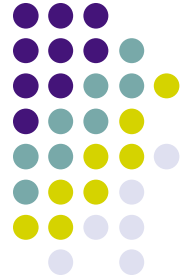
Plus a little grab-bag of thoughts at the end,  
including “the long future”



*Hic jacet Arturus, rex quondam, rexque futurus*

Geoffrey of Monmouth, 12th century

# Design is at the Center of SE



Goals and Dreams

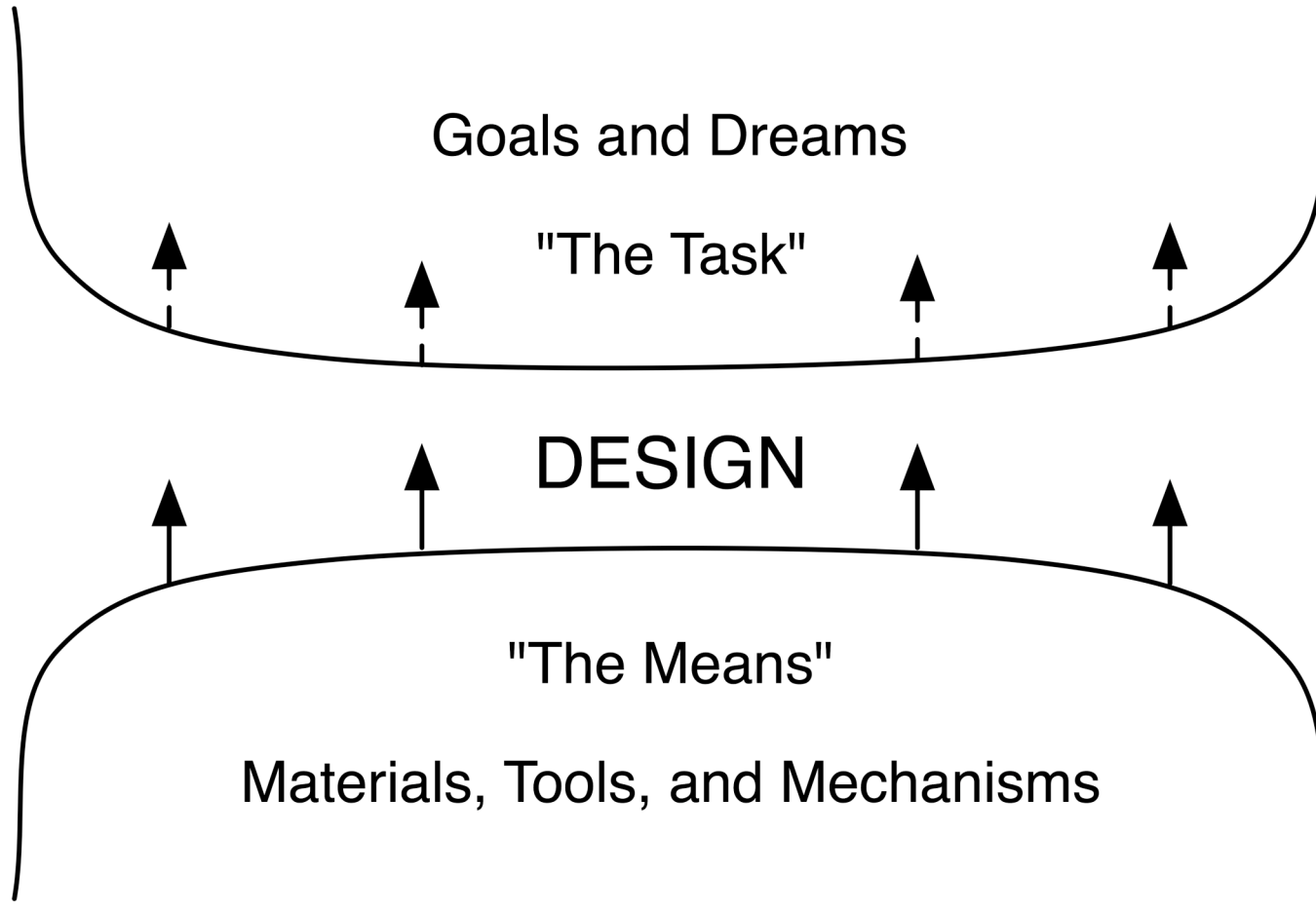
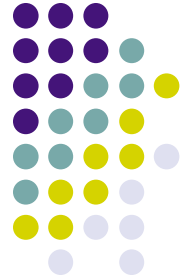
"The Task"

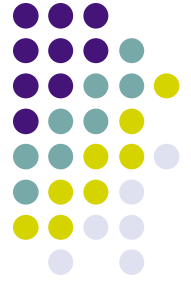
DESIGN

"The Means"

Materials, Tools, and Mechanisms

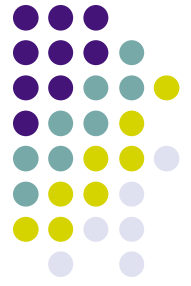
# Design Will Remain the Center





# Consequences

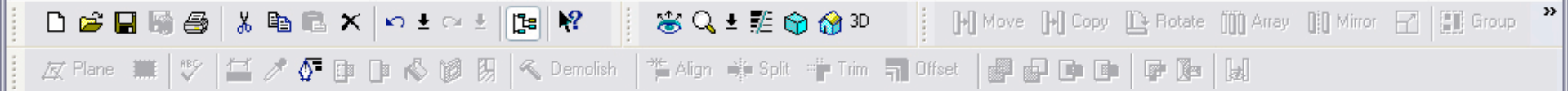
- We can all stay gainfully employed!
- Design provides a “North Star” for
  - Our educational practices
  - Our research emphases
    - Essence v. accident: as we center on the former, design becomes pre-eminent
  - Our characterization of what we do to outside parties
- Design will remain as a worthy subject of study



# Design of *What?*

- Past and current software design research
  - Prescriptive design methods, Notations, The wisdom of experience, HCI design, *Design outside of software*, *Cognitive and social strategies*
  - Agile methods, Aspect-oriented design, Design analysis, Component-based design, Software architecture
  - Point: almost all are focused on **structure** of software
- Design in other disciplines (automotive, architectural, fashion, consumer products)
  - ∃ at least as much emphasis on appearance and behavior -  
- perhaps more so.
- We need to focus on design of structure **and** appearance and behavior

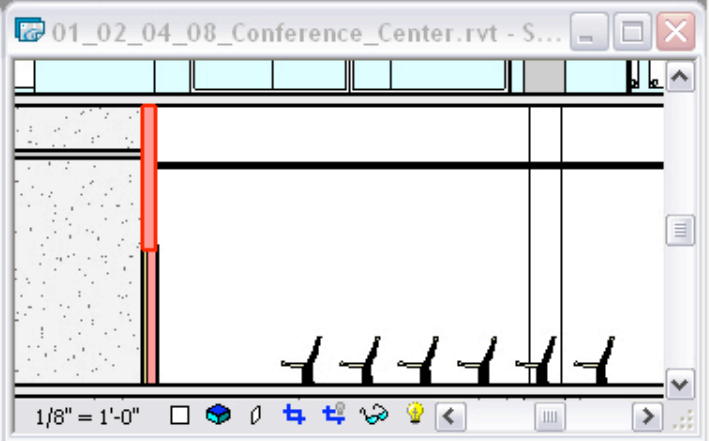
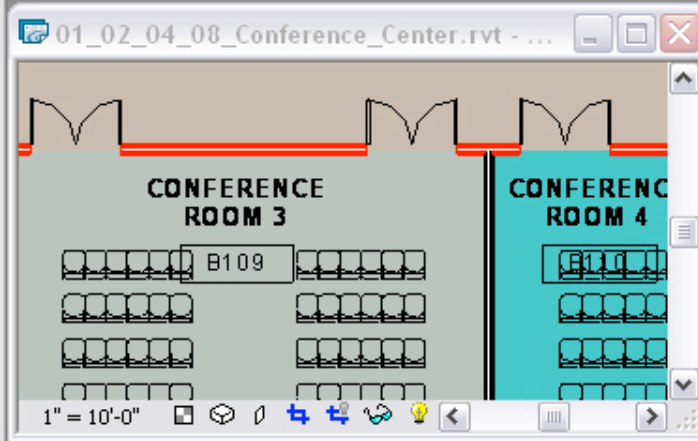
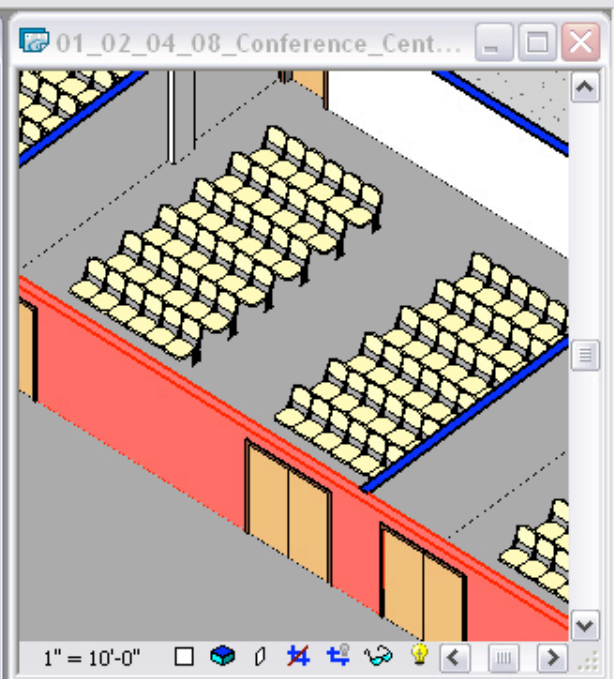
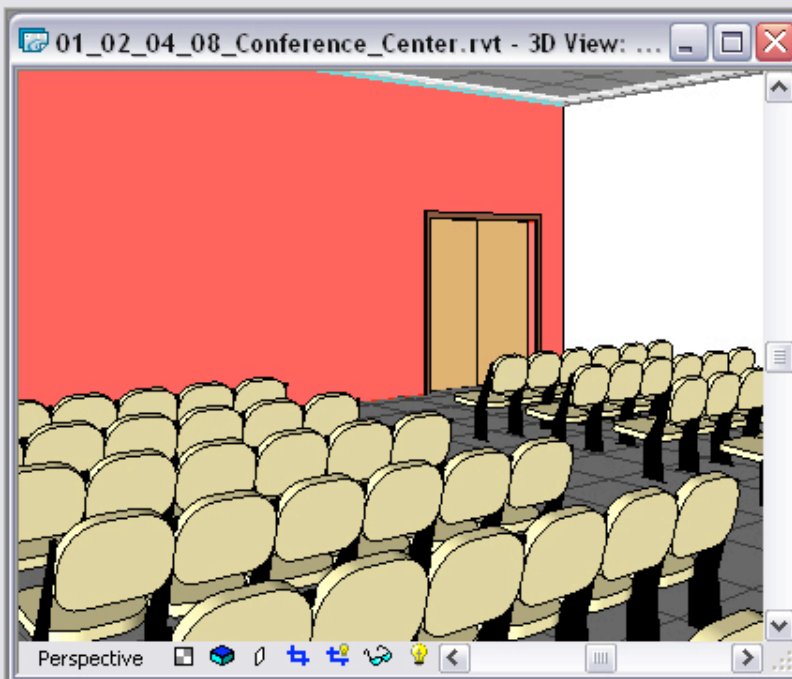


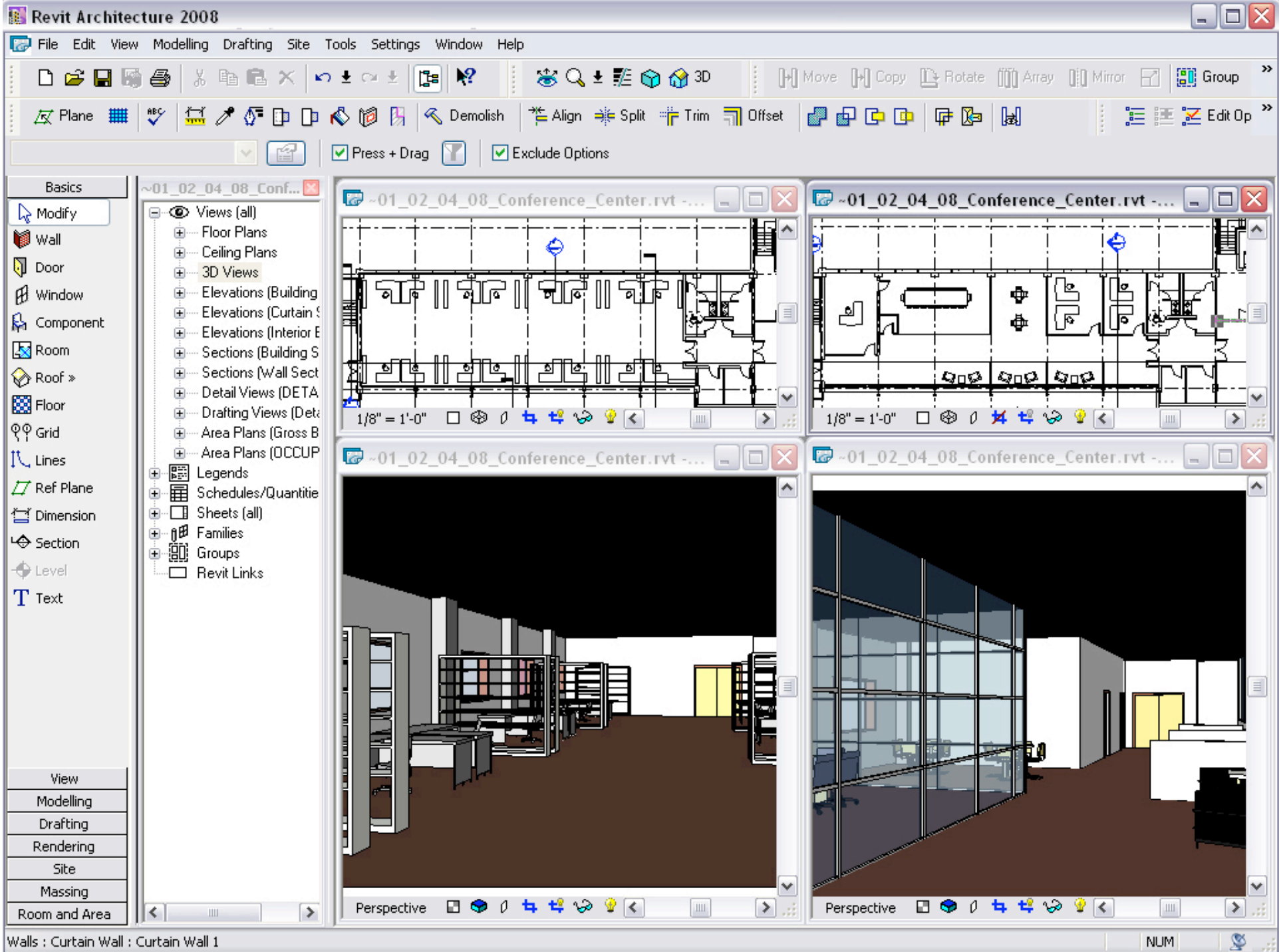


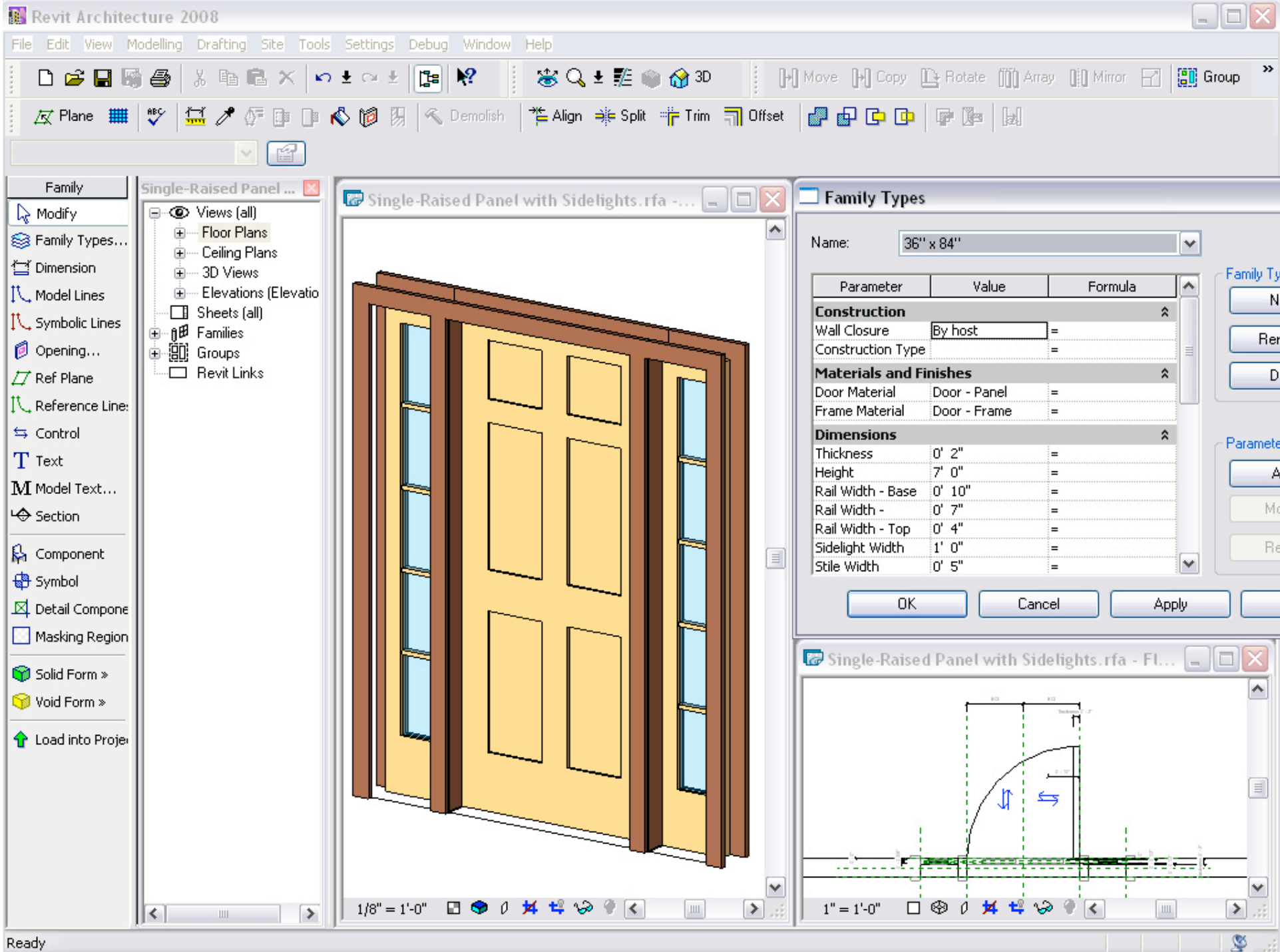
Basic Wall: Interior - Sound 7 3/4" F  Press + Drag

- Basics
  - Modify
  - Wall
  - Door
  - Window
  - Component
  - Room
  - Roof >
  - Floor
  - Grid
  - Lines
  - Ref Plane
  - Dimension
  - Section
  - Level
  - Text
- 
- View
  - Modelling
  - Drafting
  - Rendering
  - Site
  - Massing
  - Room and Area

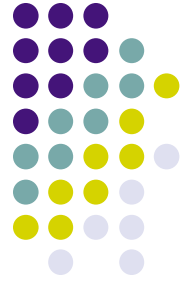
- 01\_02\_04\_08\_Confe...
  - Views (all)
    - Floor Plans
    - Ceiling Plans
    - 3D Views
    - Elevations (Building
    - Elevations (Curtain
    - Elevations (Interior E
    - Sections (Building S
    - Sections (Wall Sect
    - Detail Views (DETA
    - Drafting Views (Det
    - Area Plans (Gross B
    - Area Plans (OCCUP
  - Legends
  - Schedules/Quantitie
  - Sheets (all)
  - Families
  - Groups
  - Revit Links



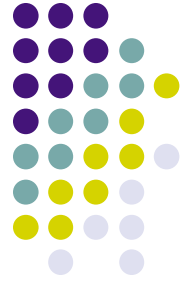




# Application Design v. Software Design

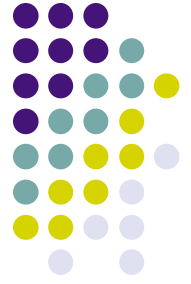


- Is this a false dichotomy?
  - The software engineering literature seems pretty one-sided...
- Is it just HCI v. software design?
  - What issue predominates in the following:
    - An airline reservation website
    - Mashups (dynamically composed web apps)
    - GoogleEarth
- If it is not our job to design applications, who's job is it?



# When Does Design Happen?

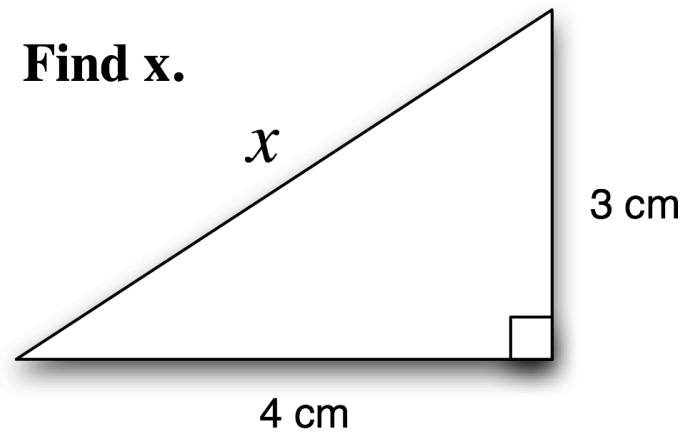
- Simple test: When is the user interface to an application designed during your favorite software process?
- A problem: we still speak of “requirements, then design”. But **design** is being done in “requirements” -- much of the time.
  - Kudos to the agile movement for the right focus here.



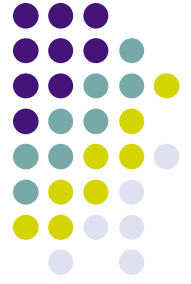
# Where Did We Go Wrong?

- First, a little test from Math class:

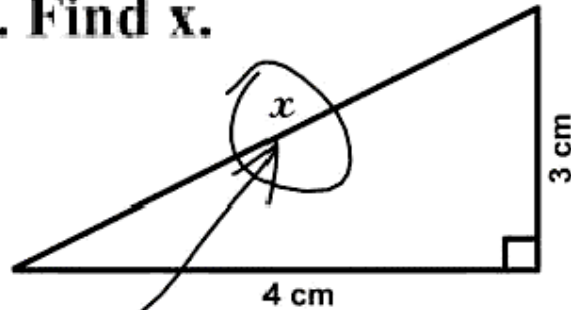
**3. Find  $x$ .**



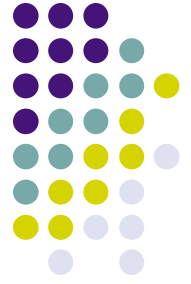
# One Student's Answer....



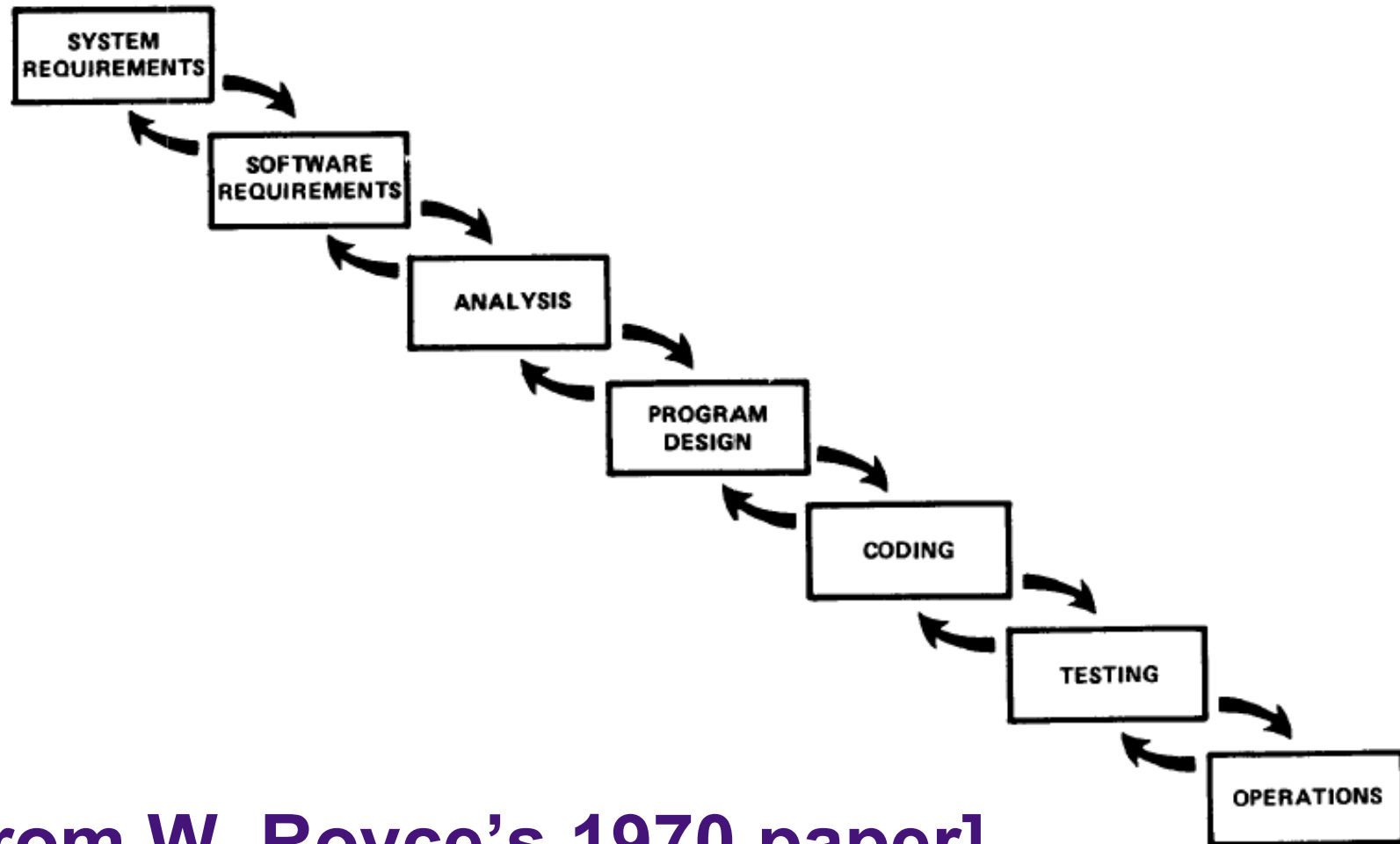
3. Find  $x$ .



*Here it is*



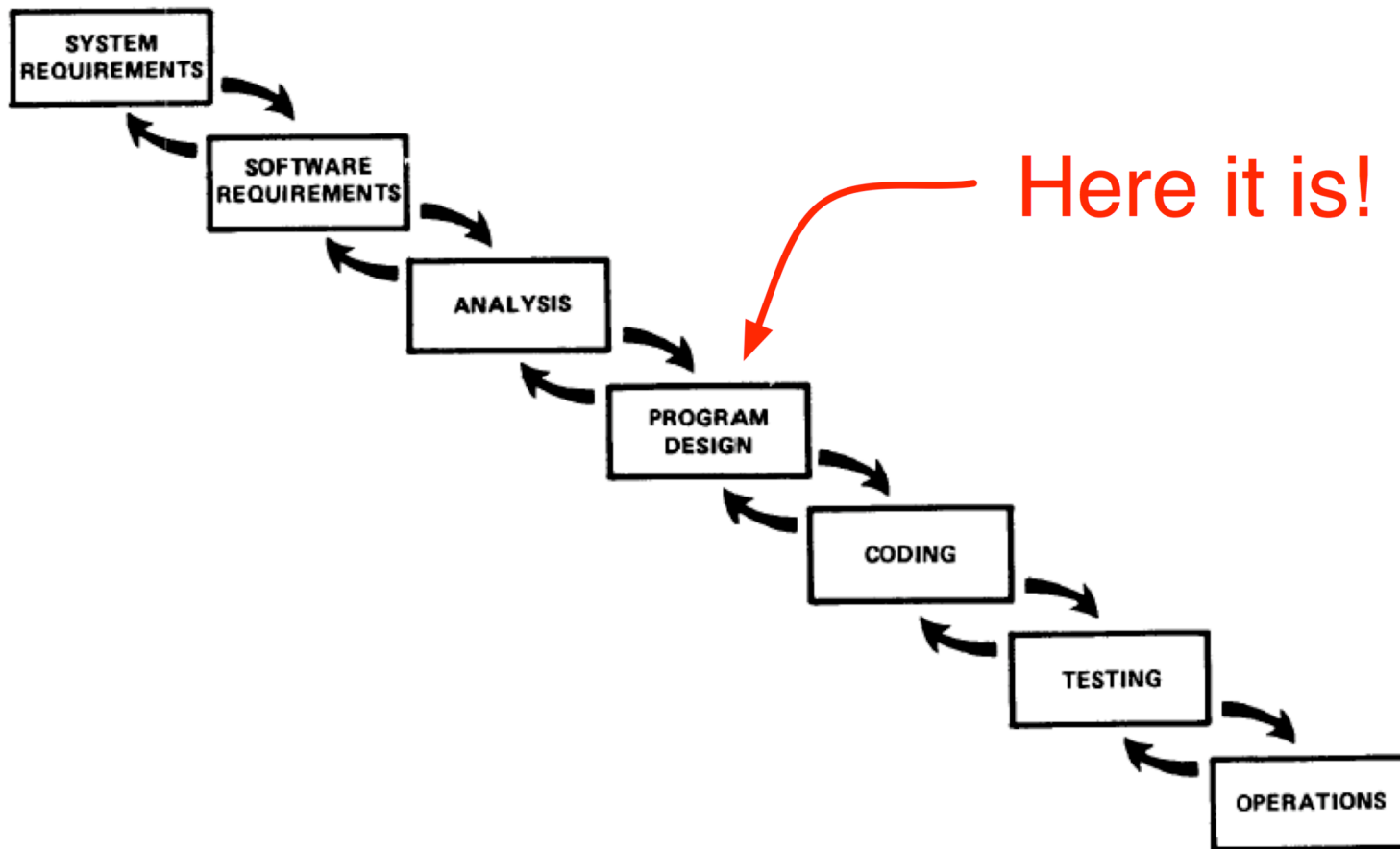
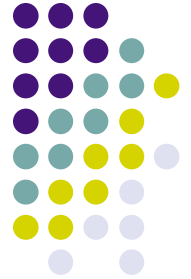
# Find Design



[From W. Royce's 1970 paper]



# Find Design



# Or Here?

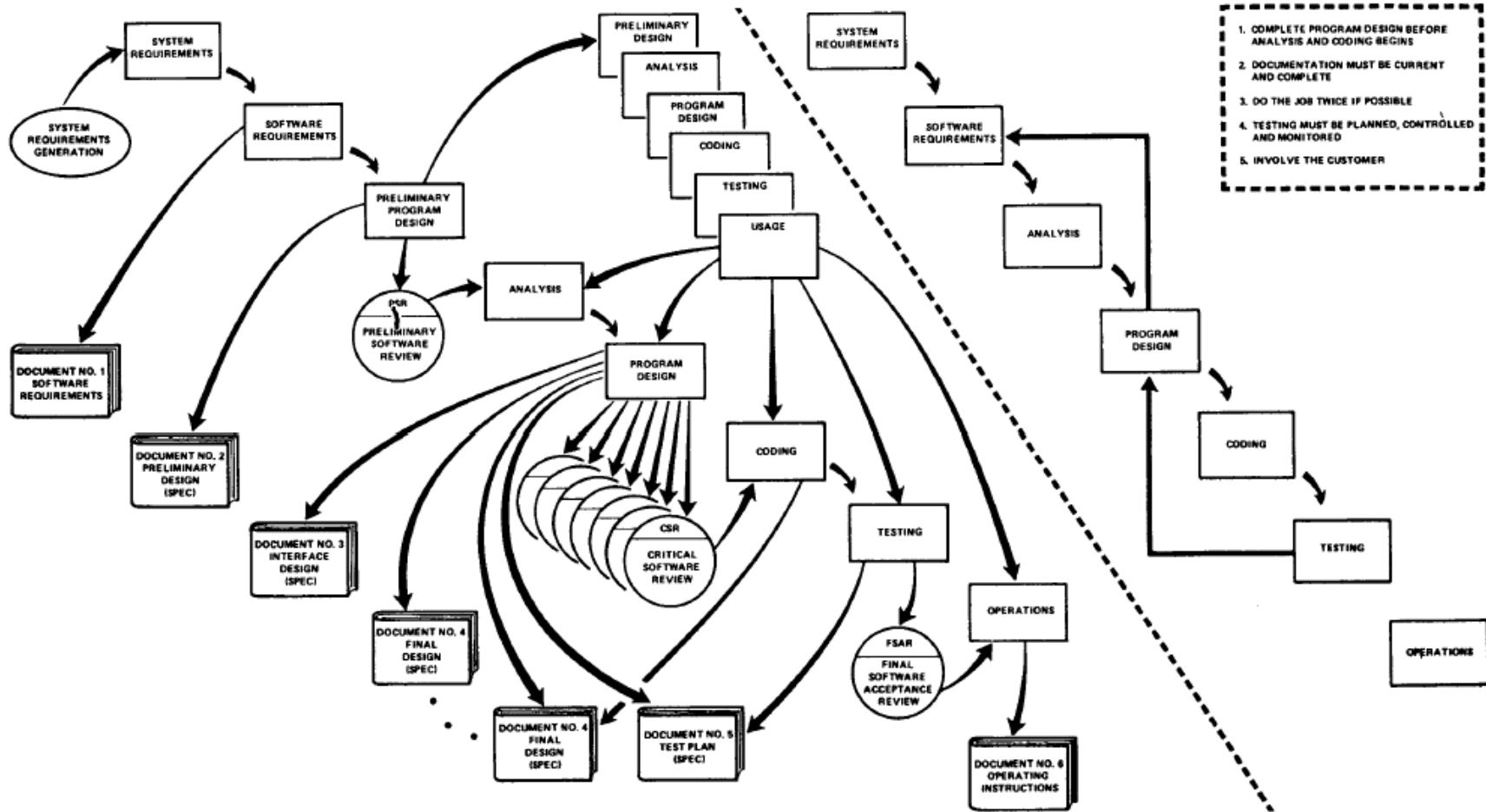
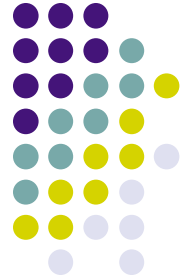
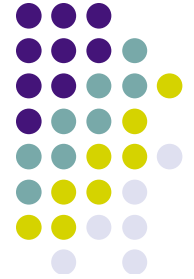
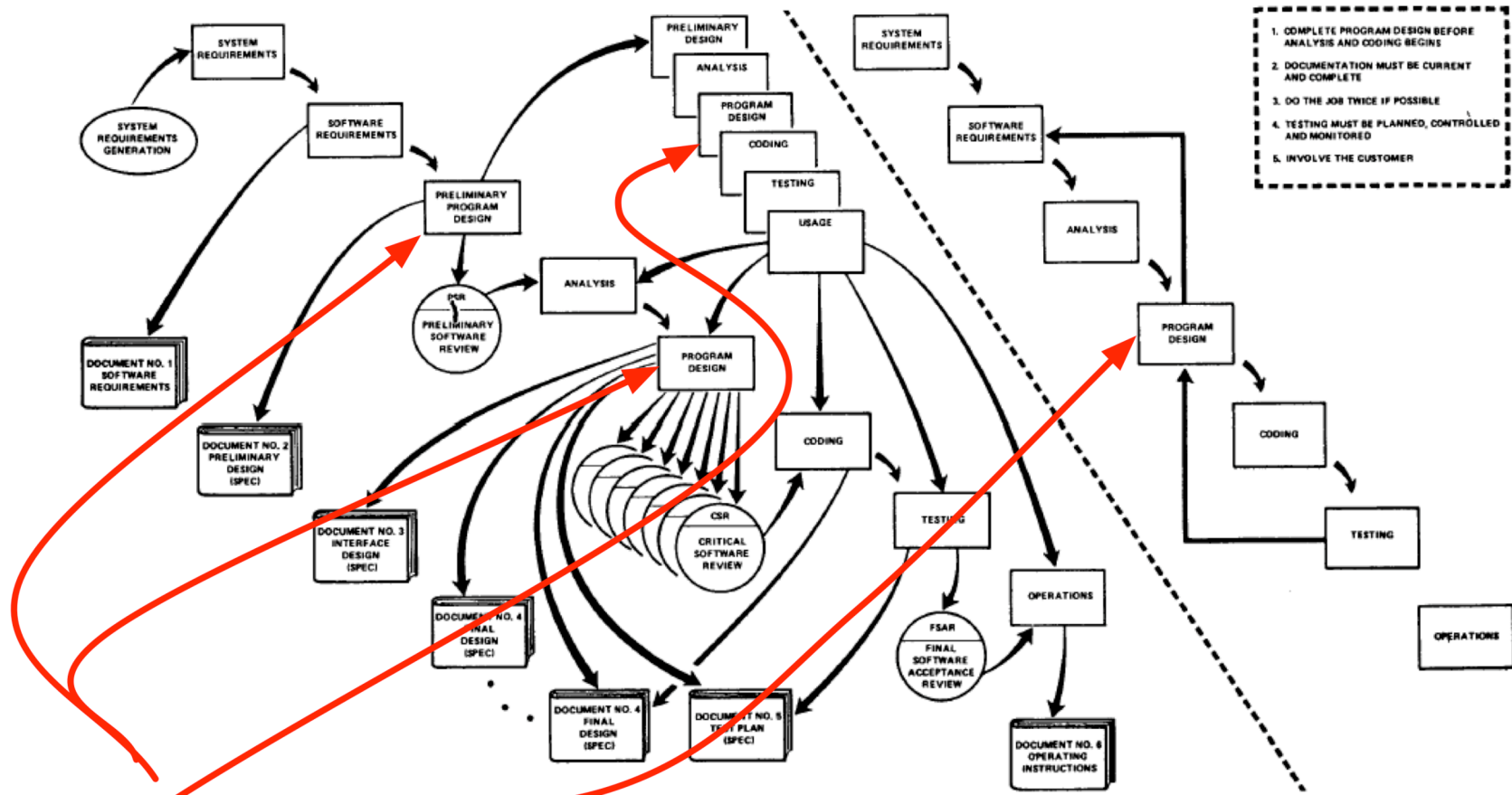


Figure 10. Summary

[From W. Royce's 1970 paper]

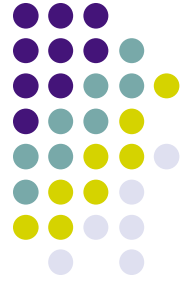


# Or Here?



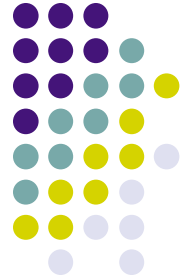
Here it is!

Figure 10. Summary



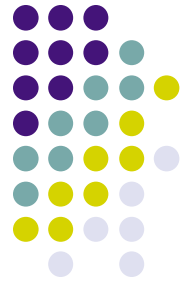
# This Early Work Misled Us

- When something is labeled “Design” or “Preliminary Design” that implies that those things that are NOT so labeled are NOT design
- Especially when some of those other things are indicated as predecessors of Design



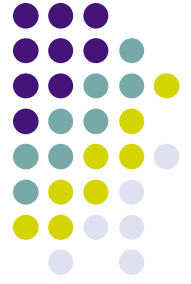
# Requirements Engineering Involves Designing

# Just What is Design?

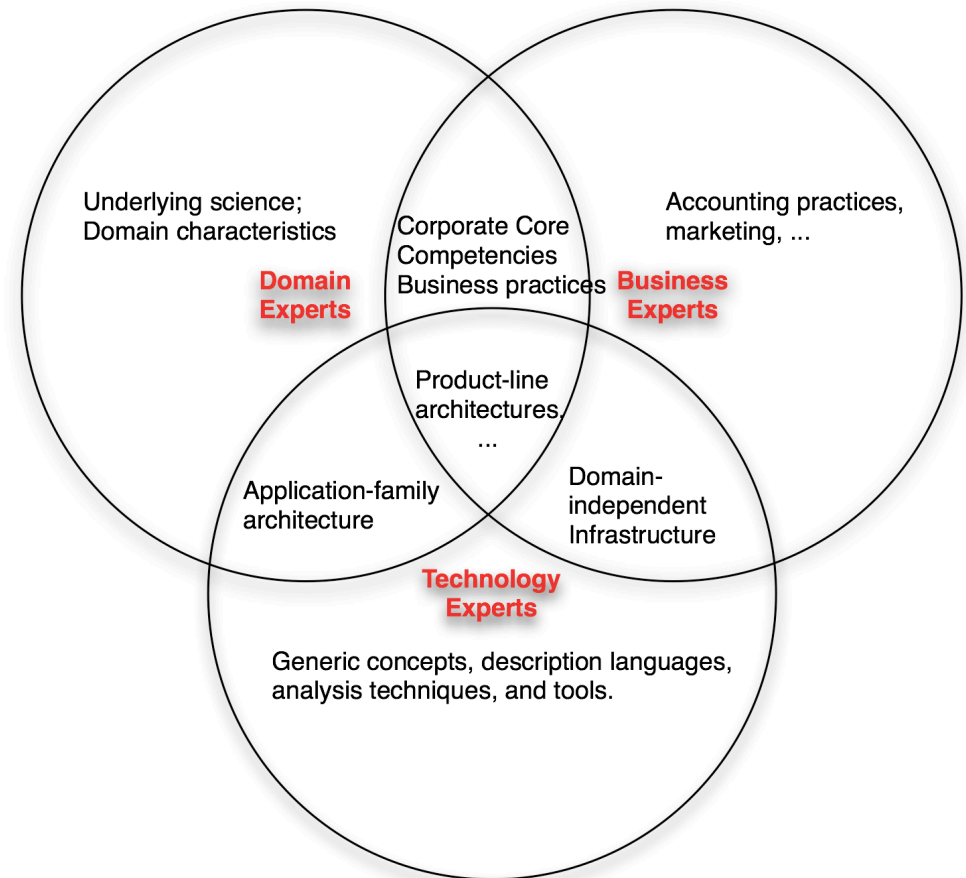


- Design as **Decision Making**
  - Decisions of how to accomplish something; how to represent something
- A solid approach to design should offer a solid approach to decision making
  - What are they?
  - Capturing them, and their rationale
  - Recognizing them when they are made, in the context
  - Analyzing, contrasting, and managing alternatives
  - Reflecting upon them to improve future designs (decision making)
  - Distilling them into practical, reusable design knowledge

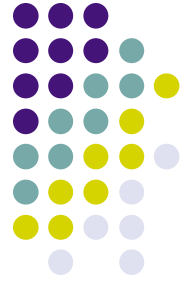
# Decision Makers: Who are the Designers?



- The software engineers
  - The “software designers”
  - The “requirements engineers”
  - The “implementers”
- The product/business strategist/analyst
- Domain experts
- Example: product-line architectures



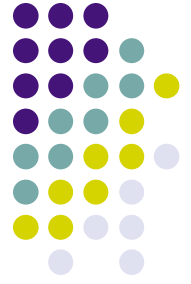
# Decision Making: When Does it Happen?



- You guessed it:
  - Throughout a system's lifetime
    - Conception
    - Detailed architecture
    - Implementation (think of the impact of choice of implementation framework)
    - Adaptation
- Why does this matter? Because these decisions matter -- and should be treated with equivalent care and be part of the design documentary

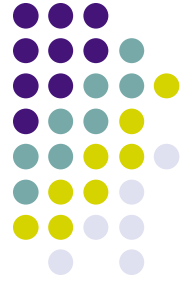


# Research Directions Grab Bag



- Never underestimate the power of brute force
  - “GoogleDesign”
- Leverage technology
  - Networking: the collaboration enabler
  - Displays

The monitor on your desk  
should be at least as large as the  
flat panel HDTV in your home



# A Design Imperative

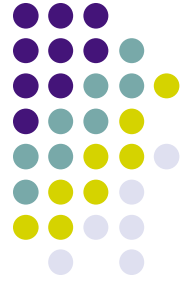
- What's key in making explicit software structural design essential, credible -- and not busy work: a full, critical tie to implementation.
  - Works in low-level (structural) design (UML)
  - Works in generative approaches to system design/implementation
  - We must make it work more broadly.



# The Long Future

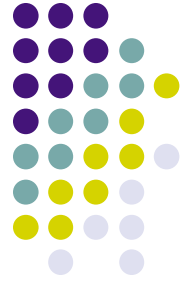
- Co-design of software *and* organizations
- Software designers as the skilled designers of other intangible products
  - The driving experience
  - Controlling
  - Commerce
  - Law





# Conclusions

1. What's the deal with the subtitle?
  - | Design is the central focus of software engineering
2. Design of what?
  - | We design structure *and* appearance *and* behavior, *and* do it throughout
3. Just what is Design?
  - | Designing involves decision-making by more than just software engineers -- and requires support



# Acknowledgments

- National Science Foundation Grants 0438996 and 0536203
- Nenad Medvidovic and Eric Dashofy
- Lee Osterweil, Peter Freeman, Alex Baker, Michael Gorlick, Peyman Oreizy, David Redmiles, Alex Wolf
- Chuck Pelly, DesignWorks USA