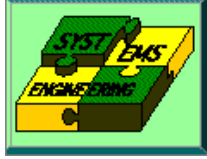
George Mason University



SYST 101: Intro to Systems

Lecture 27

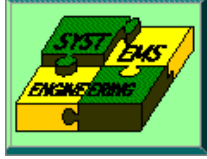
Apr 27, 2004

C. Wells, SEOR Dept.

Announcements

- Remaining Semester Schedule

  – May 1 Review for final
  – May 2 SYST 490/495 presentations
    Theater in Johnston Center, 8:30
  – May 13 Final Exam 10:30 – 1:15



Agenda

- Semester Review
- Course-Instructor Evaluations (4:00)

Overview/Summary

- All systems exist within other systems.
  - And systems have systems within them.
  - Endless nesting of systems within systems, from the universe down to subatomic particles.
- Systems interact with each other.
  - Both with systems “external” to themselves,
  - And with their “internal” subsystems.

Systems Analysis

- Systems Analysis: The understanding of the structure of a system, and its behavior
  - Structure: what composes a system, what are its subsystems, and what are its interfaces.
  - Behavior: how a system reacts to stimuli from its interfaces (external and internal)

Systems Analysis Includes

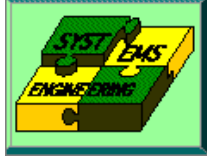
- Analysis of the need
  - Stakeholder requirements
  - House of Quality
- Analysis of the environment & context
  - Issues Analysis
  - External systems and interfaces

Systems Engineering

- Development and design of the new system or system modifications
- Clear, unambiguous representation
  - Functional Decomposition
- Defined, controlled process for bringing the system into being.

What is a System?

- Numerous definitions everywhere
- A System is:
  - A set of interacting components that together accomplish some goal or behavior; it exists within an environment, and can interact with that environment.



Key Terms

- Scope of the System
- Mission or Goals
- Requirements
- Stakeholders
- Lifecycle
- Interactions
- Behavior

Scope of the System

- What is included in your system, and what is not.
- The System's *Boundaries*

Mission or Goals

- What is the system supposed to do?
- How well does it need to do it?
  - Performance
- Criteria for success

Requirements

- Based on the Mission/Goals
- More detail
- Must be clear
- Must be *testable*
  - Someone else should be able to test whether your system satisfies the requirement or not

Stakeholders

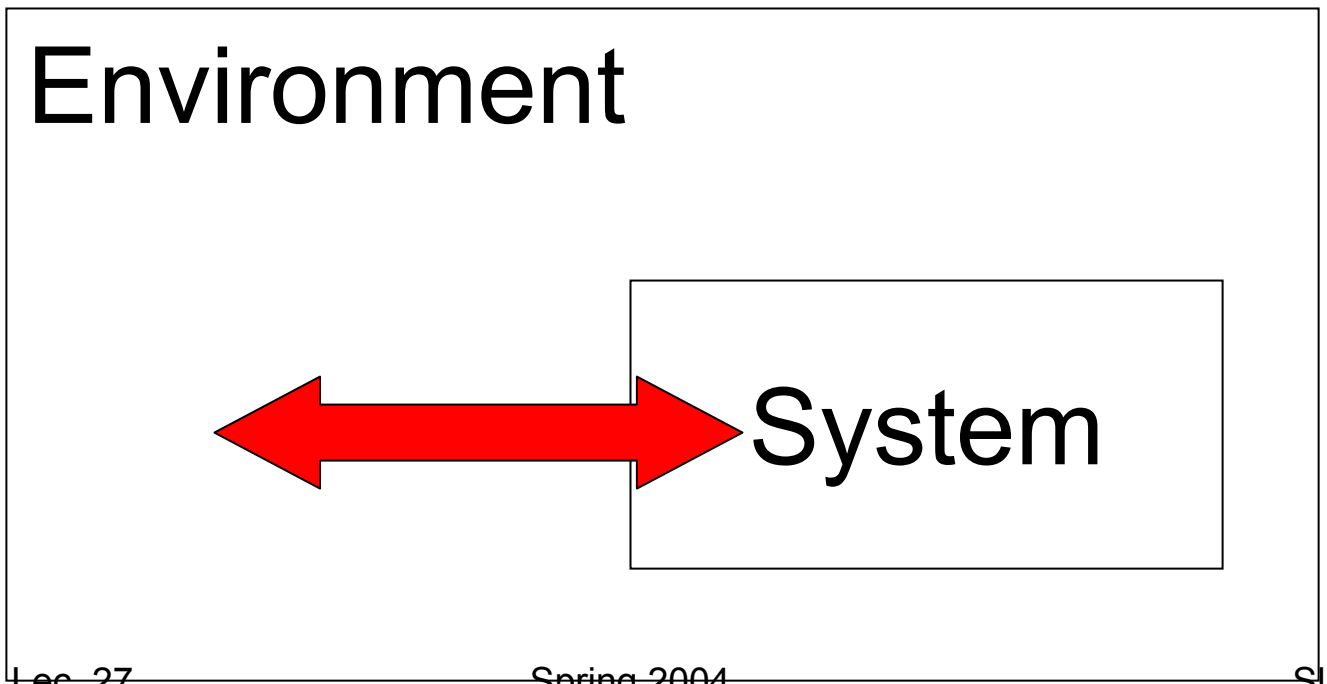
- All of the people or organizations that care about or are impacted by the system.
- Everyone who needs to have input into how the system will function or how it will be used.

Lifecycle

- Systems usually undergo a “life”
  - Starting with initial ideas and concepts
  - Through the design process,
  - Then they’re developed and tested,
  - Deployed in the field or commercial arena,
  - Maintained and operated,
  - Retired and removed from use.
- Examples where Lifecycle problems exist?

Interactions

- Every system interacts with it's environment.

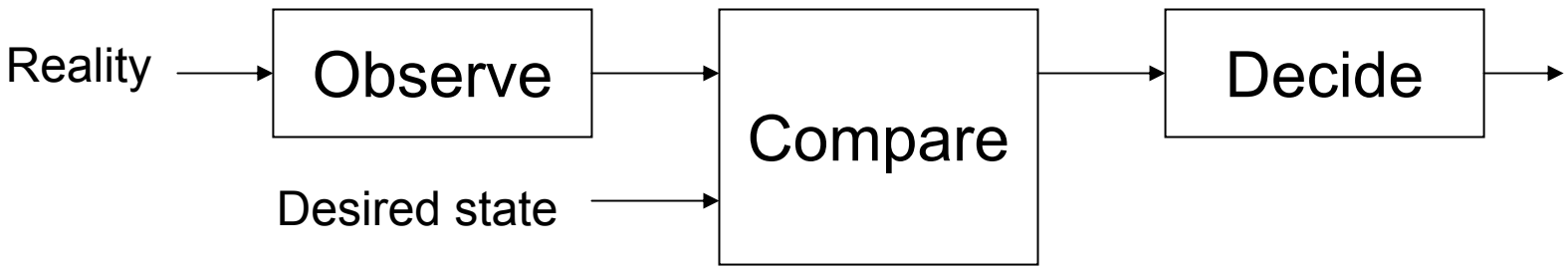


Behavior

- Defines what a system needs to do or does in response to stimuli
- *Stimuli* : Various events, conditions or occurrences that stimulate a reaction in the system.
- Systems are usually purchased for their behavior, not their appearance.

Beginnings . . .

- It starts with an evaluation
  - Something is not right or could be better



Welcome to Ambiguity

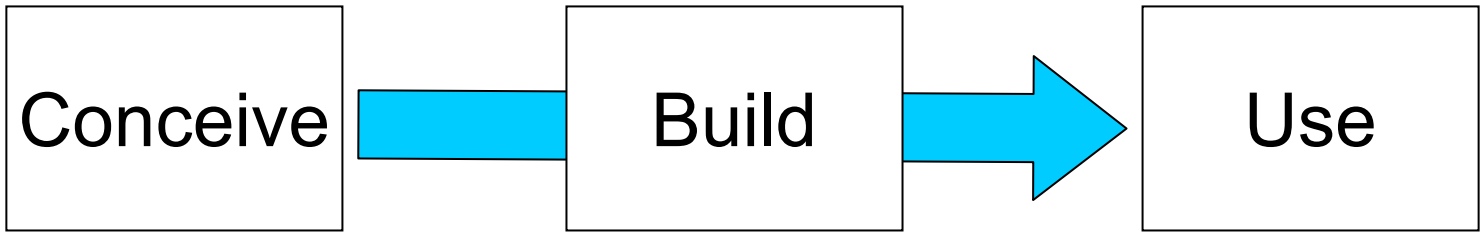
- Key concept is necessary and sufficient
  - It depends
- When we decide, how good is "good enough"
  - It depends
- When we observe, how good is "good enough"
  - It depends

Trial and Error . . .

- After you evaluate, you try something and see if it will work

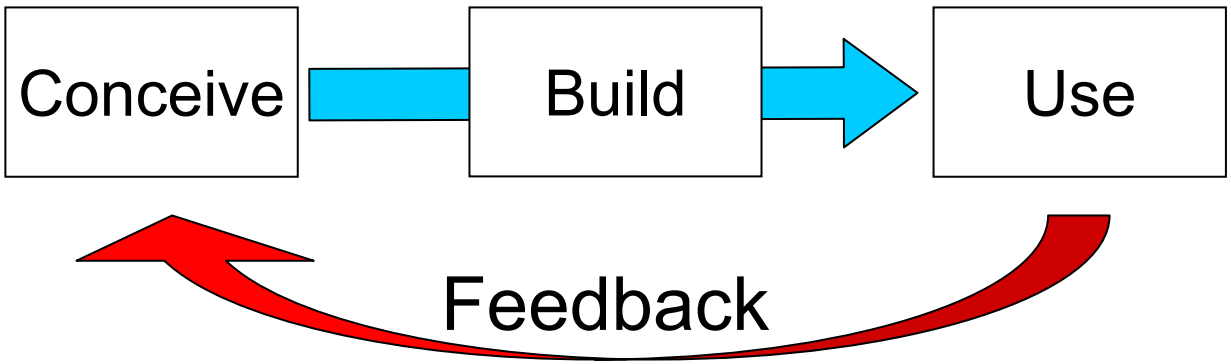
  - Where “evaluate” is the observe, compare and decide process

The Most Basic Creative Process
(the Trial Part)



- Suitable for anything from stone chisels to life in general . . .
- But how do you Conceive?
  - You make an **educated** guess at the solution!

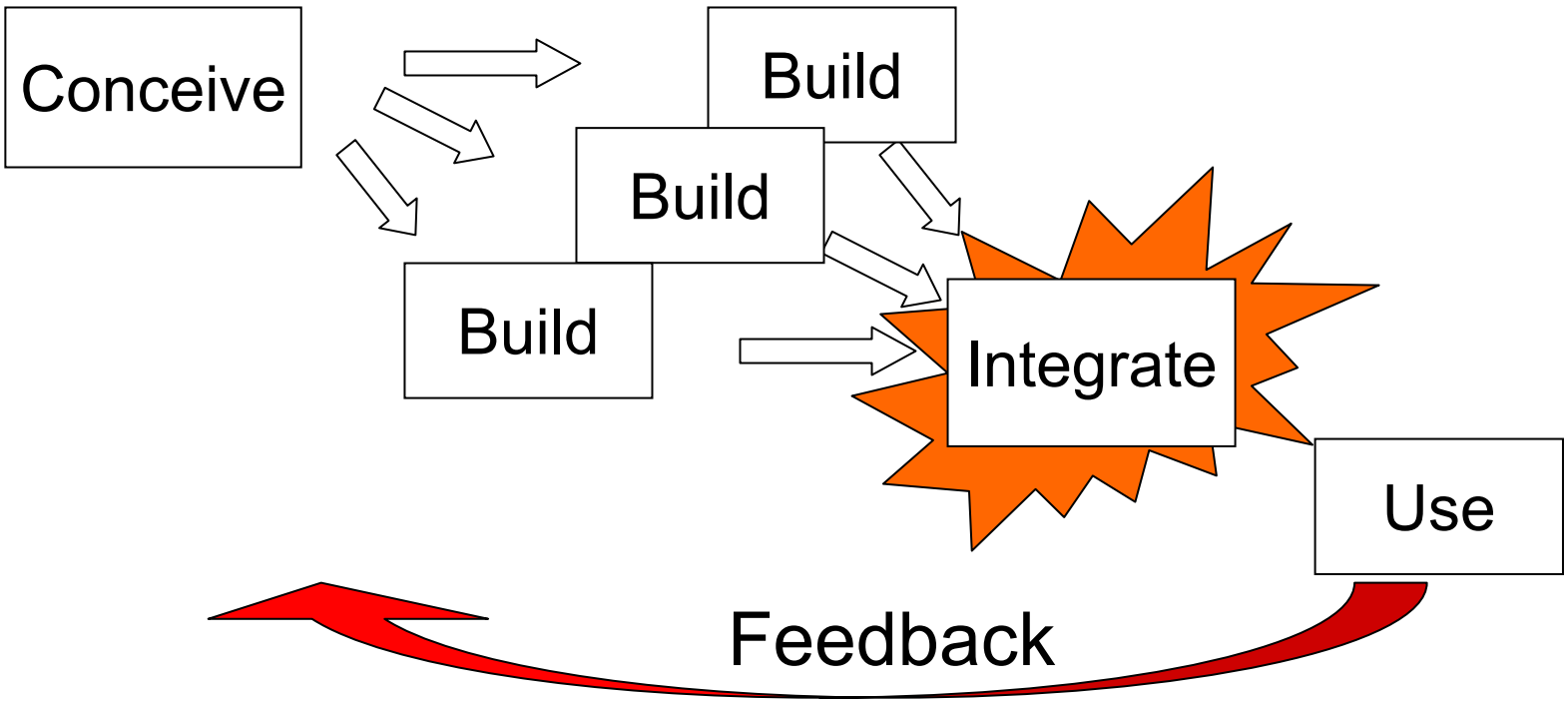
How Do Things Get Improved?



- By Feedback, where the use of the first version provides input to the second version
- Driven by evaluating the trial
- When do you stop? When is it good enough?
  - It depends

A More Complex Build Process

- Our product now has three parts, built separately, that we need to fit together...

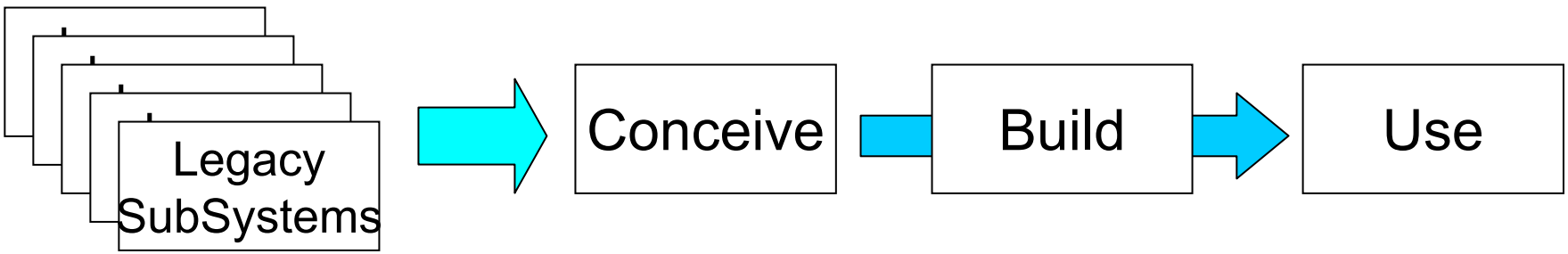


Integration Step

- If clear, unambiguous instructions haven't been given to each component builder, then integration doesn't go very well

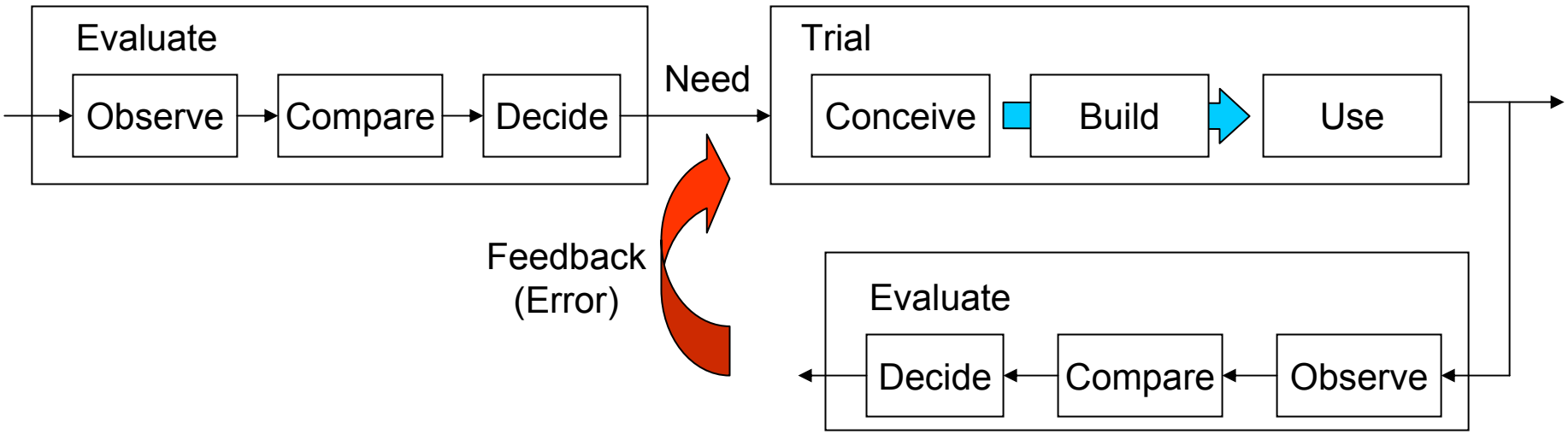
Fusion

- A New System is envisioned that joins existing, independent pieces into a new kind of system
- Somewhat like a jigsaw puzzle, seeing how these pieces can fit together



Knowledge &
Understanding

Detailed Creative Process



Invention Versus Need

- Invention Driven by need
- Needs not always conscious and perceived
  - needs may exist that you don't realize
  - but, revolutionary inventions create their own need
    - did we "need" the internet, webpages and browsers before they were invented?

Satisfying a Need

- When a system satisfies a need, we say that it has “utility”.
- “Utility” - The system (or device) serves some purpose and, hopefully, does it better (in some measure) than other things. It may do it cheaper, faster, more efficiently, easier for the user, more reliably....

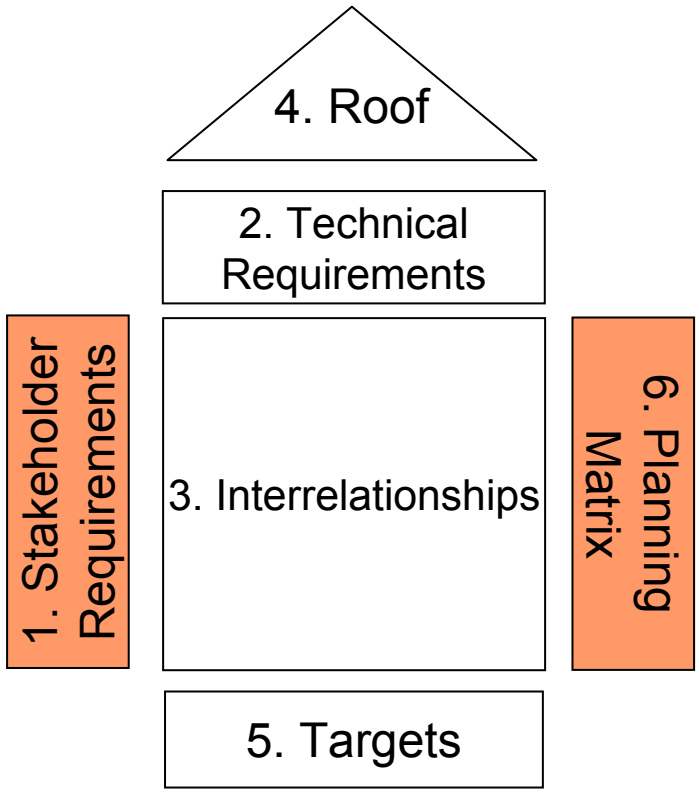
Cost/Benefit Ratio

- A measure used to evaluate systems
- All systems have an associated "cost"
  - Cost to buy, time to learn and use, maintenance costs
- All (good) systems have a benefit (utility) when used.
- Is what you get out of them worth what you put in?

Issues Analysis

- Requires logical and careful thinking about the desired end result, and how you plan to get there.
- May require re-thinking your concepts and plans as you proceed.
- May require mathematical analysis or computer simulation.
  - Calculus, Analytical geometry,

House of Quality



1. what does it need to do?

2. detailed aspects that help it do what it needs to do

3. how things on list 2 achieve things on list 1

4. how things on list 2 are inter-related

5. Design alternatives

Assumptions in Analysis

- Always starts with simplifying assumptions.
  - Solve the easy problem first, then add complicating factors and issues
- Always keep in mind your assumptions
  - You not really solving the real problem, you're solving something similar (you hope)

Analysis Difficulties

- You can never be sure when you are done
  - Understanding what is important
  - Unanticipated failure mechanisms
  - Unrealistic assumptions
- Mind set when performing analysis
  - Keep an open mind
  - Second set of eyes on the problem

How to Analyze

- Actually build the system and test it
  - Costs a lot and can't test for all situations
  - Risks to life
  - One time events
- Build a model of the system and test the model.
  - Cheap(er), safer, repeatable under various situations, and usually simpler that the system
  - But are they adequate?
    - it depends

Model - Definition

- A model is a representation of some entity.
- The entity does not have to actually exist.
- The model itself does not have to have physical existence.

Why Use Models

- Because the entity –
  - does not have to exist
  - is too complicated to understand
  - is too costly to build unless it works
  - is too dangerous to use until we understand it

Models Are Scalable

- How detailed must a model be?
  - It depends on the use
- What kind of model should be used?
  - It depends on the use
- You can do a cost/benefit analysis of the information gained (benefit) versus the type/detail of the model (cost)

Model Verification

- Models may need to be validated if their accuracy is questionable
  - Too simplified
  - Design far from the existing practice
  - Design close to failure
- The design may require Qualification if the accuracy of the model is questionable

WARNING!

- Models are not reality
  - They represent reality
  - They are simplistic
  - They are erroneous (but may be good enough)
- The problems we encounter in system engineering are really problems in the adequacy of our models

Two Flavors of Models

- The “pre-reality” model
  - exists before an entity
  - used in creating an entity
  - used in understanding a hypothetical entity
- The “post-reality” model
  - exists after the entity exists
  - used in understanding a real entity

Adequacy

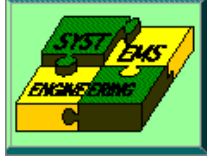
- The problem of adequacy is exposed when we attempt to make the “entity”
  - The realized model does not match expectations
- The problem of adequacy is exposed when we attempt to model reality
  - The model’s behavior does not match reality

Risk

- There is risk whenever we go between a modeled universe and the real universe
  - Errors caused by
    - Lack of experience
    - Errors in assumptions
  - Impact of errors
    - cost
    - opportunity

Risk Reduction

- Improve models
  - Improved fidelity
  - Improved detail
  - Validate models
- Improve analysis



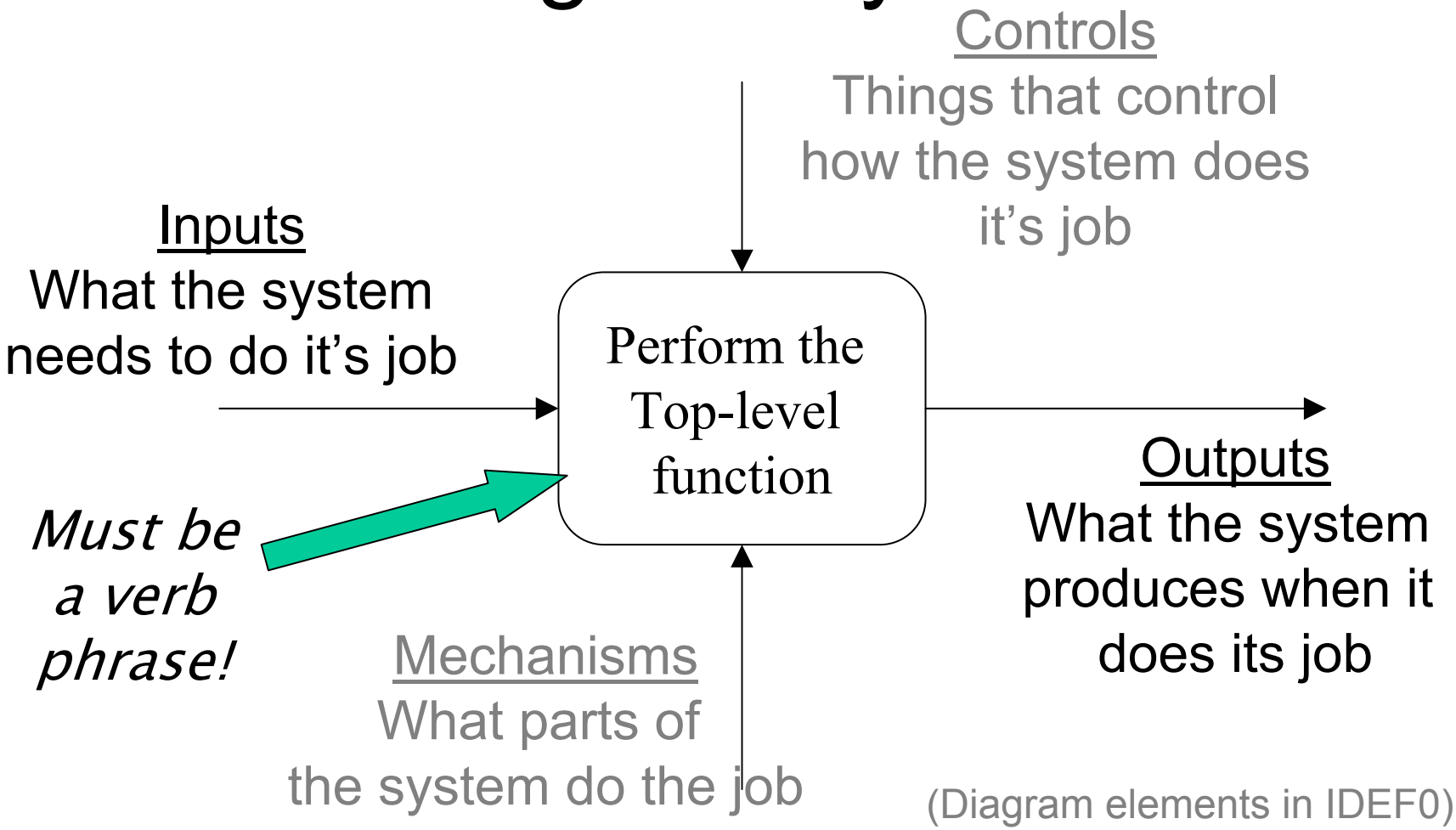
Step Back and Get an Overview

- We do things by trial and error
  - Trial
    - Conceive (educated guess)
    - Build (a model or the real thing)
    - Use (or analyze)
  - Error
    - Observe
    - Compare
    - Decide
- We iterate until the solution is “good enough”
  - We use models until satisfied with the solution
  - Then build the real thing

Pair Wise Comparison

YOU	Barney	Fred	Wilma	Jane
Barney		B > F?	B > W?	B > J?
Fred	F>B?		F > W?	F > J?
Wilma	W>B?	W>F?		W > J?
Jane	J>B?	J>F?	J>W?	

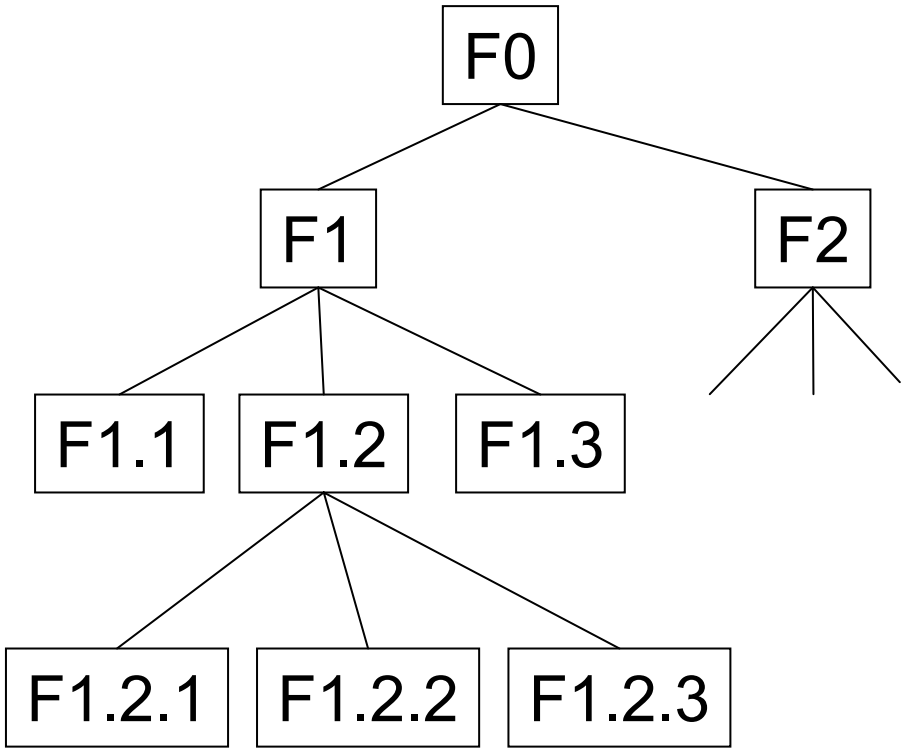
Diagram Syntax



(Diagram elements in IDEF0)

Functional Decomposition – Representation Techniques

Graphical Representation



Outline Form

- F0
  - F1
    - F1.1
    - F1.2
      - F1.2.1
      - F1.2.2
      - F1.2.3
    - F1.3
  - F2

Deciding on the Subfunctions

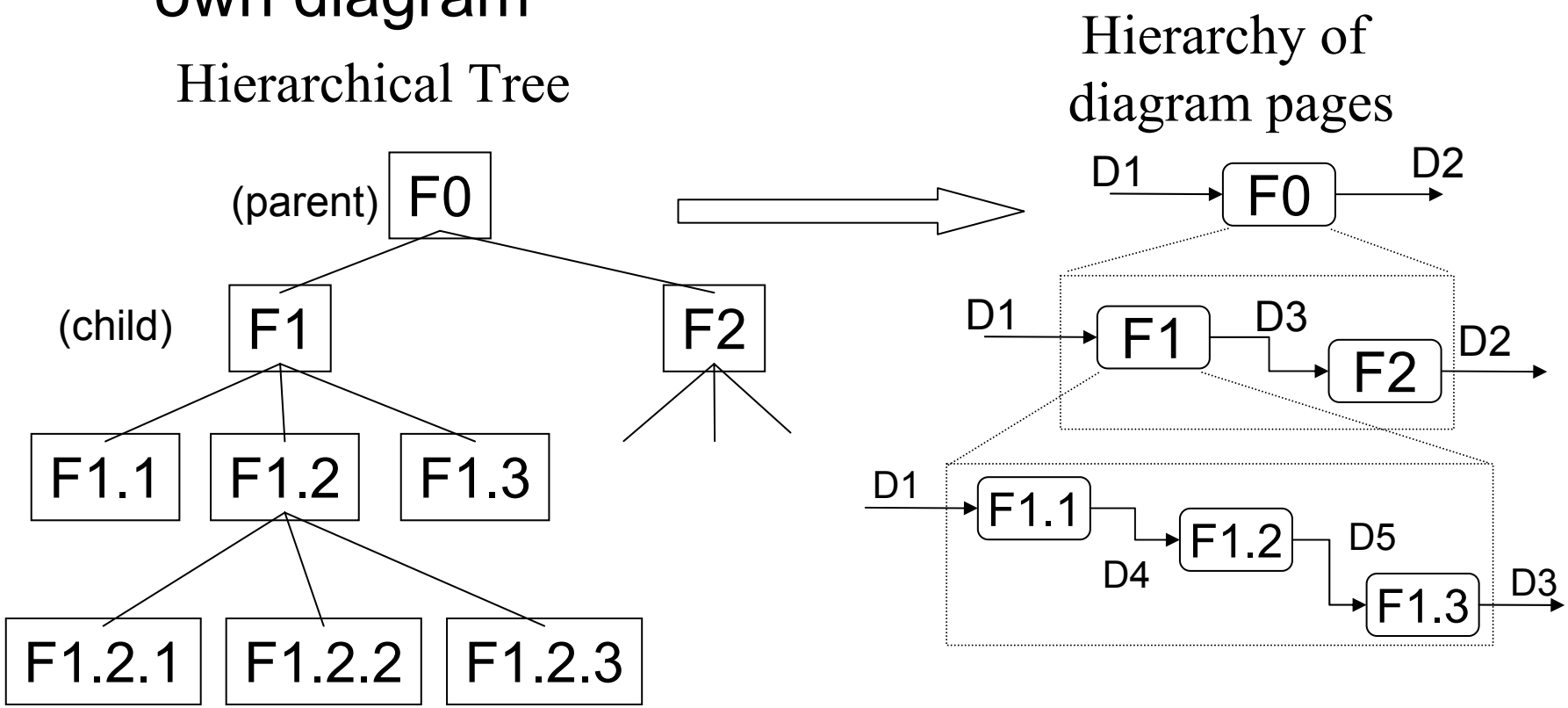
- Does involve some degree of experience, practice, creativity
  - There are no instructions on how to come up with the “right” subfunctions
  - “Bad” or poor decompositions will have certain “symptoms” – return to this later
- Can be iterative with the following analysis steps (dataflows, entity-relationships)

Rules on Functional Decomposition

- No function can be repeated.
- Function names must be verb phrases – connecting flows must be entities (data, things)
- Functions should have clear boundaries between them
  - This will be reinforced when you define the input-output entities.
- How many levels of decomposition are necessary?
  - No single answer – depends on the scope and intended audience of your design project
- How can one tell what's a "good" decomposition?
  - Clear subfunctions, easy connections between them, don't violate the above rules

Functional Decompositions

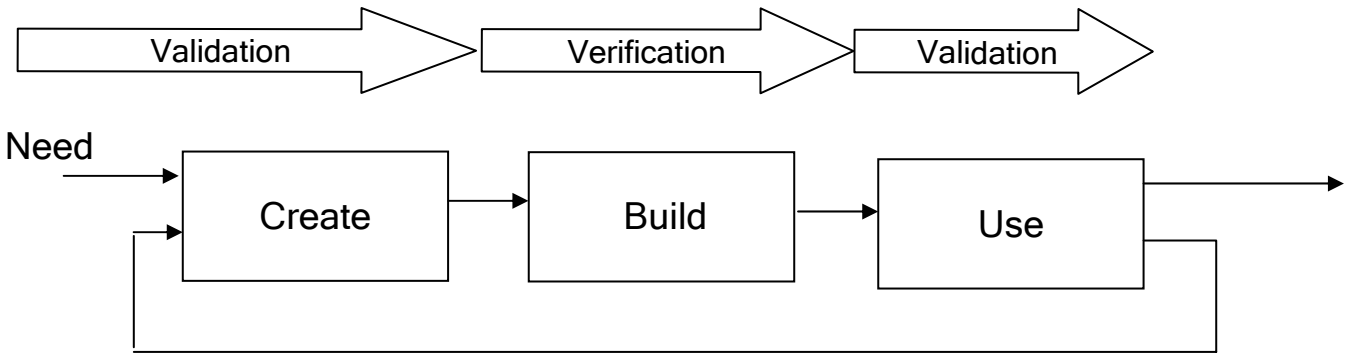
- Every function that is decomposed gets it's own diagram



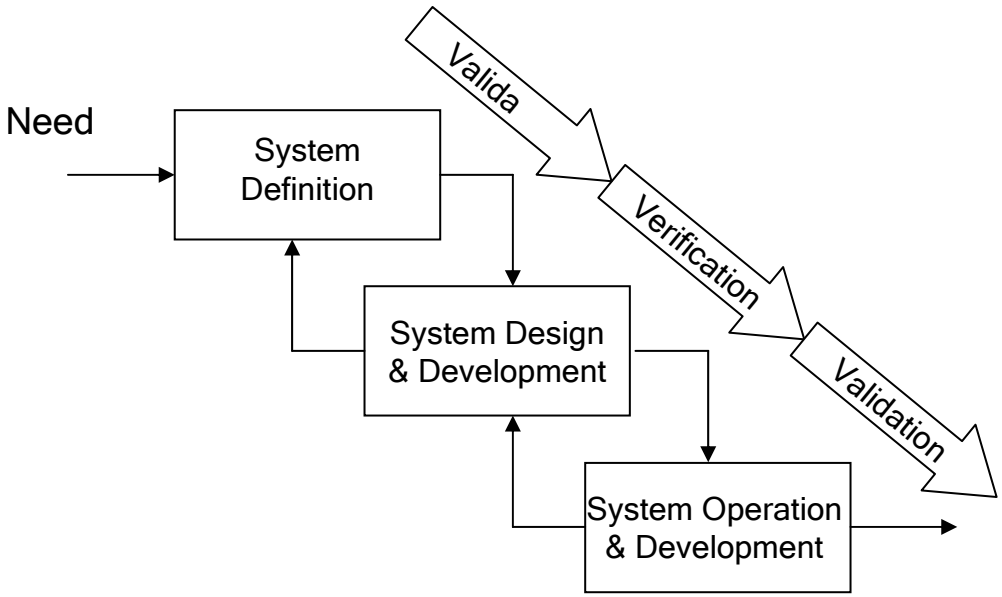
Validation and Verification

- Validation is insuring what you are building satisfies the needs
  - Associated with establishing requirements
  - Associated with the use of the system
- Verification is insuring that what you built is what you thought you build
  - Associated with building and testing the system

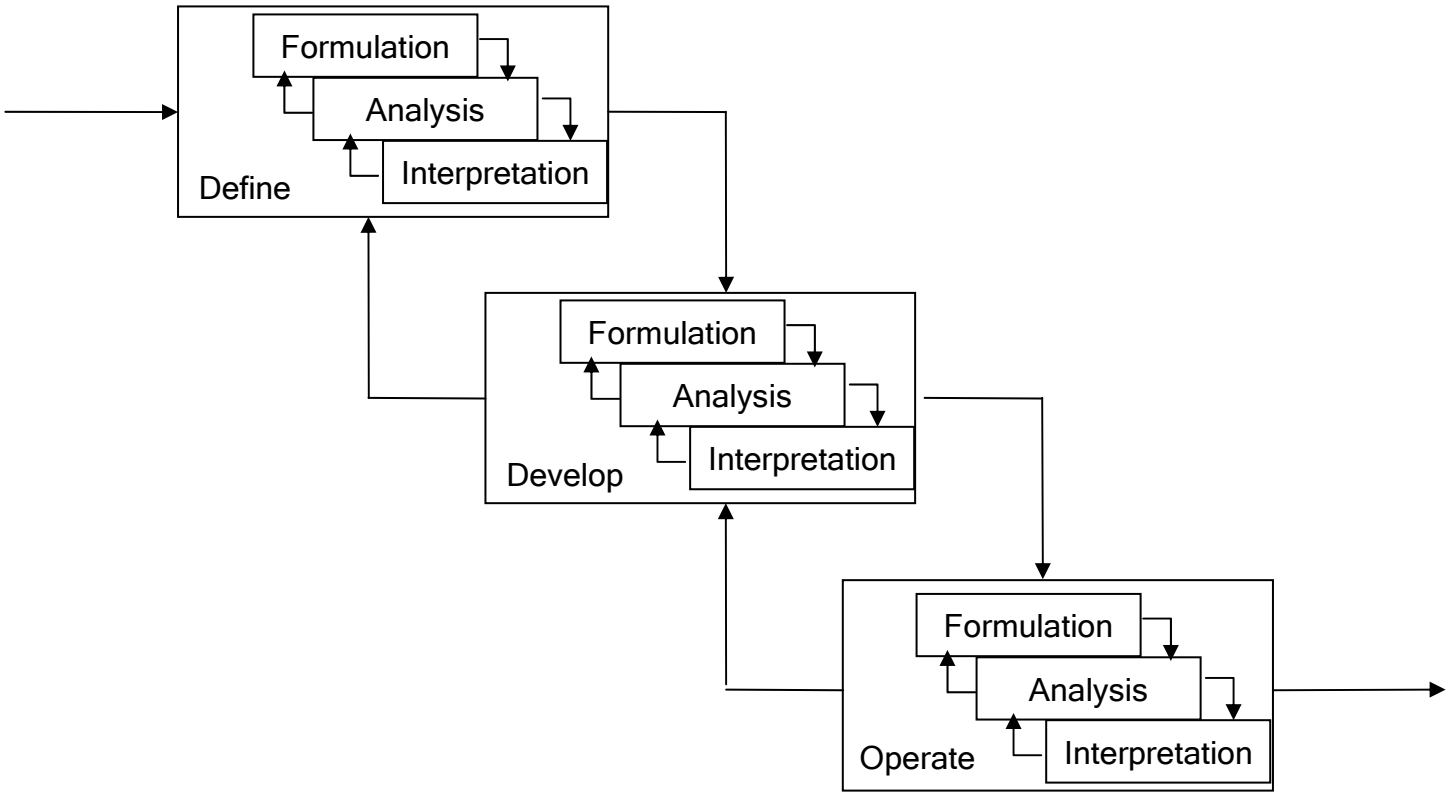
A Three Step Method with Validation & Verification



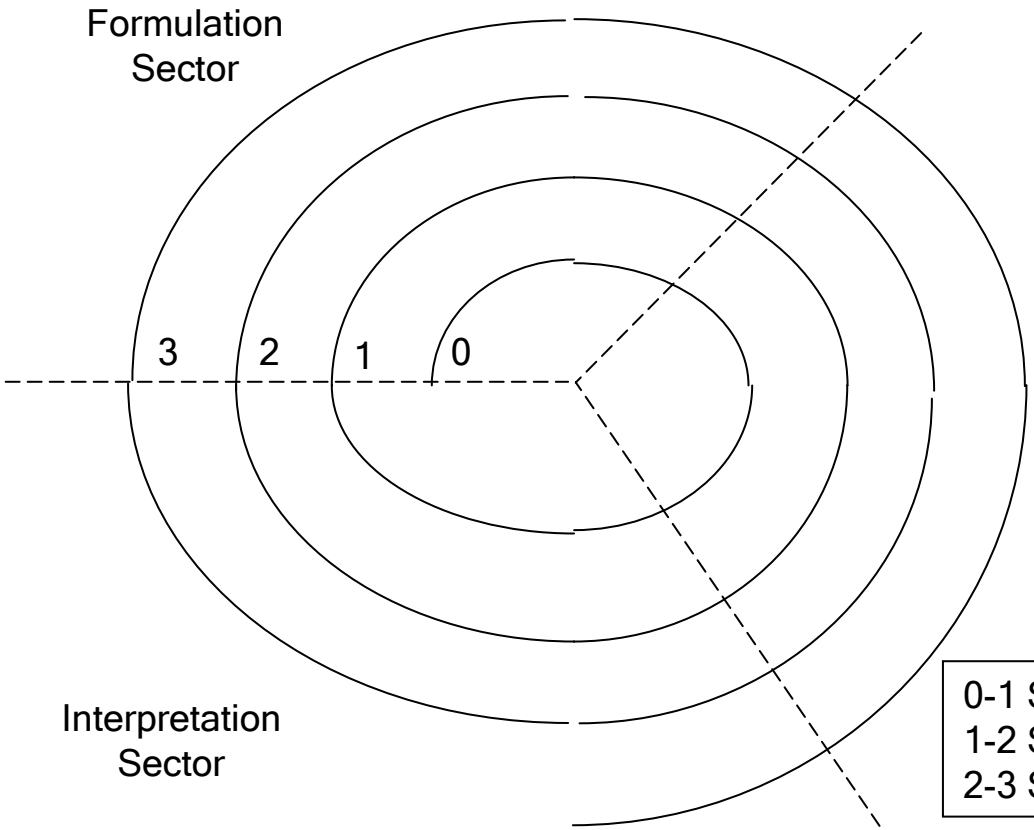
Same Method, Different Words



Expanded 3 Element Model

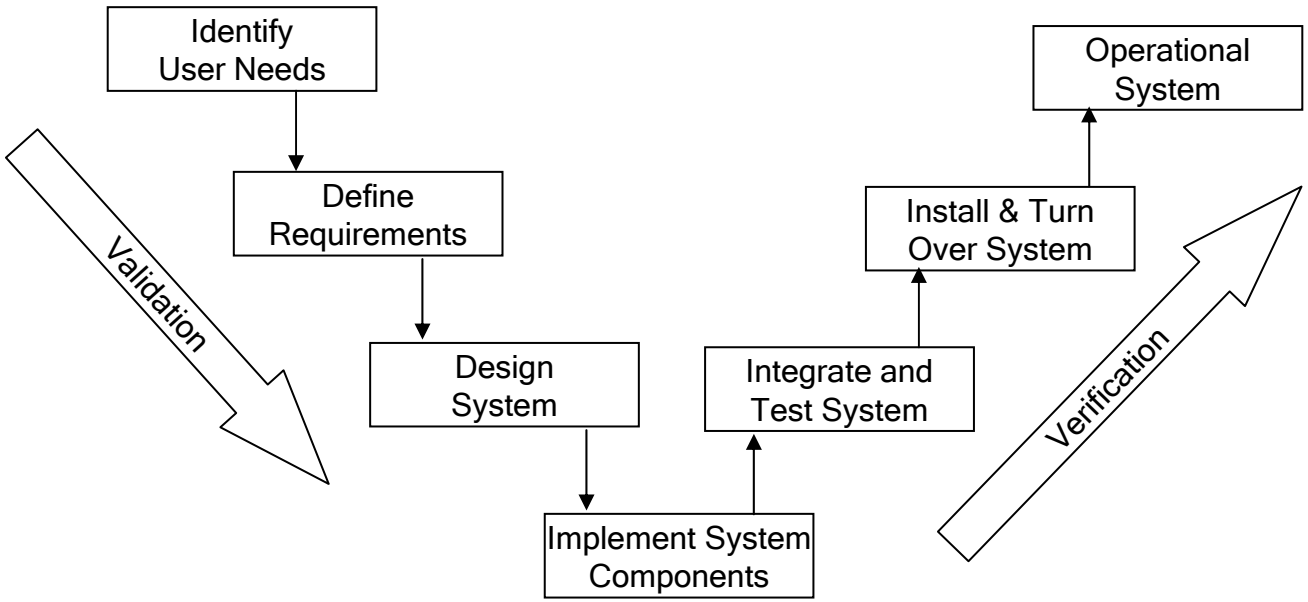


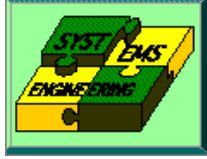
Spiral Model



0-1 System Definition
1-2 System Design & Development
2-3 System Operation & Development

The V System Engineering Methodology





Observation

- They all say about the same thing
  - Different nuances for different situations
  - NIH syndrome between companies?
- Represent common sense approach
- Provides gates to ensure work is done
- Framework for documentation