

The logo consists of five white squares arranged horizontally.

Stanford Systems Engineering

The Stanford Systems Engineering Program

December 13, 2005

Systems Engineering at Stanford

A New Focus

Built on Strong Legacies and a
Stanford and Center for Systems
Management Partnership

Insertion Approaches

- Indirect

 - Product Development course

 - Aero/Astro and Management
Science and Engineering

 - Information Systems course

 - Aero/Astro and EE

- Direct

 - Systems Engineering Certificate Program

 - Core SE courses

The Product and System Development Curriculum

A systems engineering approach

1. Concepts of value and stakeholders
2. From value to requirements - QFD
3. Functional Analysis; Interfaces
4. Concepts and architectures/trades
5. Failure and Risk issues
Six sigma; Variance
6. V & V
7. IPPD and DFX
8. Lean initiatives
9. Value stream analysis
10. Product development mgt.
11. Cost analysis and benefits
12. Student oral reports

Information Systems in Aerospace Vehicles

- A course on information collection, processing and transfer
- Emphasis on architectures and networks
- Seamless information flow - Design for fault tolerance a feature

SE Certificate Program

- **On Campus and On Line**
 - Stanford and CSM Faculty
- **Three Core Courses - 20 Hours Each**
 - Systems Engineering Principles and Strategies
 - Value/stakeholder concepts
 - System Development: From Desire to Design
 - Reqs to Architecture/trade criteria include productivity
 - System Integration: From Design to Proven Value
 - V&V/life cycle flow

SE Certificate Program #2

- **Three Electives – 20 Hours Each**

- Decision Analysis (March and September)
- Lean Principles and Practices (March and September)
- Risk Analysis (September)

- **Potential Electives**

- Design for Manufacturing and Assembly
- Product and System Development

SEPS Course Outline

- **Session #1 – Introduction**
- **Session #2 – Systems Thinking**
- **Session #3 – Decision Analysis**
- **Session #4 – Framing and Valuation**
- **Session #5 – Managing The Value Proposition**
- **Session #6 – Baseline Management and Decision Gates**
- **Session #7 – Project Cycles and Systems Engineering Models**
- **Session #8 – Technical Development Strategies and Tactics**
- **Session #9 – Artifacts and Artifact Management**
- **Session #10 – Summary**

What Is A System?

A combination of any or all of hardware, software, facilities, personnel, data, and services to perform a designated function with specified results. The highest member of the example system decomposition hierarchy.

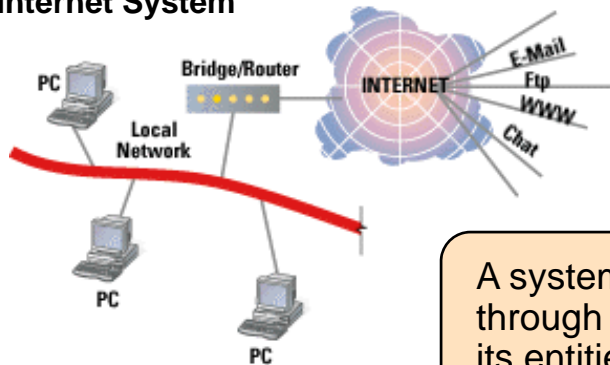
How we define the levels is a matter of perspective and purpose. Since each level could be a system, most systems are a "system of systems."

Example Hierarchy
 system
 segment
 element
 subsystem
 assembly
 subassembly
 part



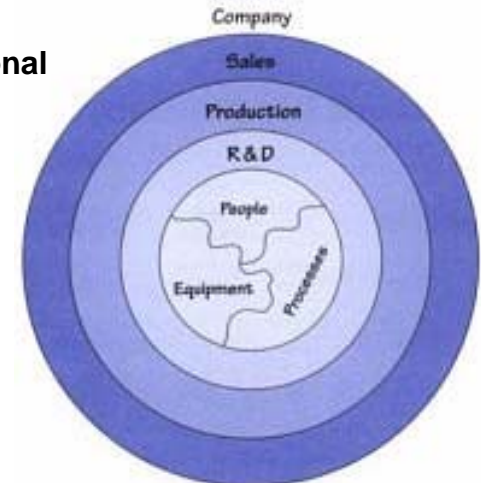
Reference:
Communicating Project Management
 Mooz, Forsberg, Cotterman
 John Wiley & Sons, 2002

Internet System



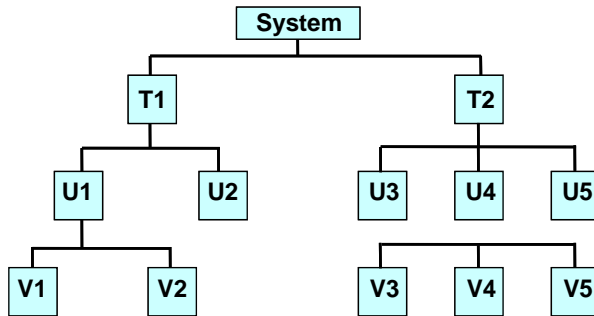
A system maintains its existence through the mutual interaction of its entities.

Organizational System

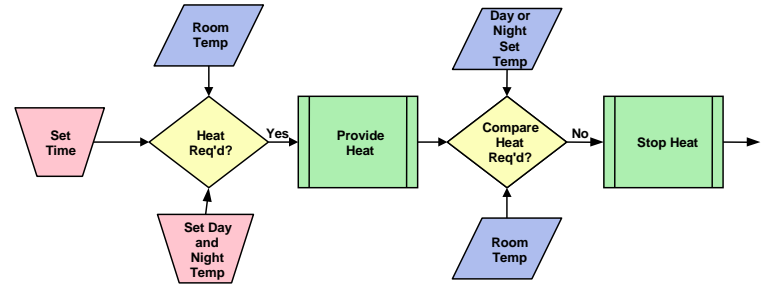


Systems – Many Types

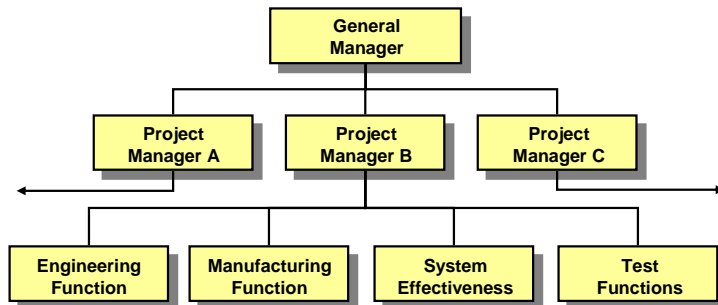
Product



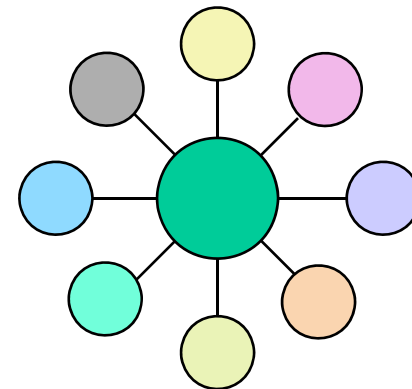
Process



Organization



Enterprise



Systems Engineering Challenges

- Translate user needs and expectations and stakeholder constraints into efficient and effective solutions
- Orchestrate the technical community and technical disciplines
- Ensure the compatibility of all interfaces
- Identify, characterize, and pursue opportunities
- Identify, characterize, and mitigate risks
- Provide information for management decisions

Certify the system

Systems Development Course Outline

- **Session #1 – Introduction**
- **Session #2 – Requirements Development**
- **Session #3 – Requirements Expression**
- **Session #4 – Conceiving Concepts and Architectures**
- **Session #5 – Selecting Concepts and Selecting Architectures**
- **Session #6 – Making Difficult Decisions**
- **Session #7 – Requirements Flowdown**
- **Session #8 – Communicating Concepts and Architectures**
- **Session #9 – Summary**

Course Plan

Day	Session	Session	Target time (in hours)	Start	Finish	Target number of Charts. Time for any exercise must be taken out of the total.
14-Sep	1	Introduction	1.5	13:00	14:30	30
		Break	0.5	14:30	15:00	
	2	Requirements Development and Prioritization	1.5	15:00	16:30	30
	2	Requirements Development and Prioritization (cont)	1	8:30	9:30	20
		Break	0.5	9:30	10:00	
	3	Requirements Expression	1.5	10:00	11:30	30
	4	Conceiving Concepts and Architectures	0.5	11:30	12:00	10
		Lunch	1	12:00	13:00	
15-Sep	4	Conceiving Concepts and Architectures(cont)	1.5	13:00	14:30	30
		Break	0.5	14:30	15:00	
	5	Making Difficult Decisions	1.5	15:00	16:30	30
	6	Selecting Concepts and Selecting Architectures	2	8:30	10:30	40
		Break	0.5	10:30	11:00	
	7	Requirements Flowdown	1	11:00	12:00	20
		Lunch	1	12:00	13:00	
16-Sep	7	Requirements Flowdown (cont)	0.5	13:00	13:30	10
	8	Communicating Concepts and Architectures	1	13:30	14:30	20
		Break	0.5	14:30	15:00	
	8	Communicating Concepts and Architectures (cont)	1	15:00	16:00	20
	9	Course Summary	0.5	16:00	16:30	

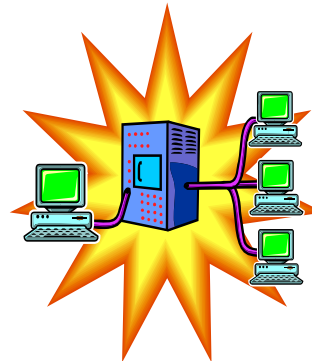
Requirements Development in Context

- Requirements drive the project
- Both business and technical
- From all stakeholders
- Many types of stakeholders and requirements
- Difficult to elicit
- Must be managed
- Change – Embrace or resist?



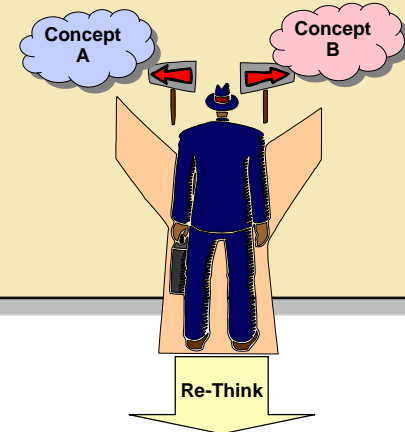
Conceiving Concepts in Context

- **Concept creation requires imaginative thinking**
- **Concept creation is the first step in selecting the correct solution**
- **If the requirements are not correct faulty concepts will result**
- **Concepts lead to the implementing architecture which also must be correct**
- **Concept and architecture selection lays the technical keel of the project and pre determines the success probability**



Selecting Concepts and Architectures in Context

- **Concept selection is key to successful projects**
- **Concepts can have several implementing architectures**
- **Architecture selection is key to successful deployment and maintenance**
- **Concept and architecture selection criteria are considerably different**
- **Quantitative approaches include Kepner Tregoe, weighted Pugh, and others**
- **Other approaches are not as rigorous**



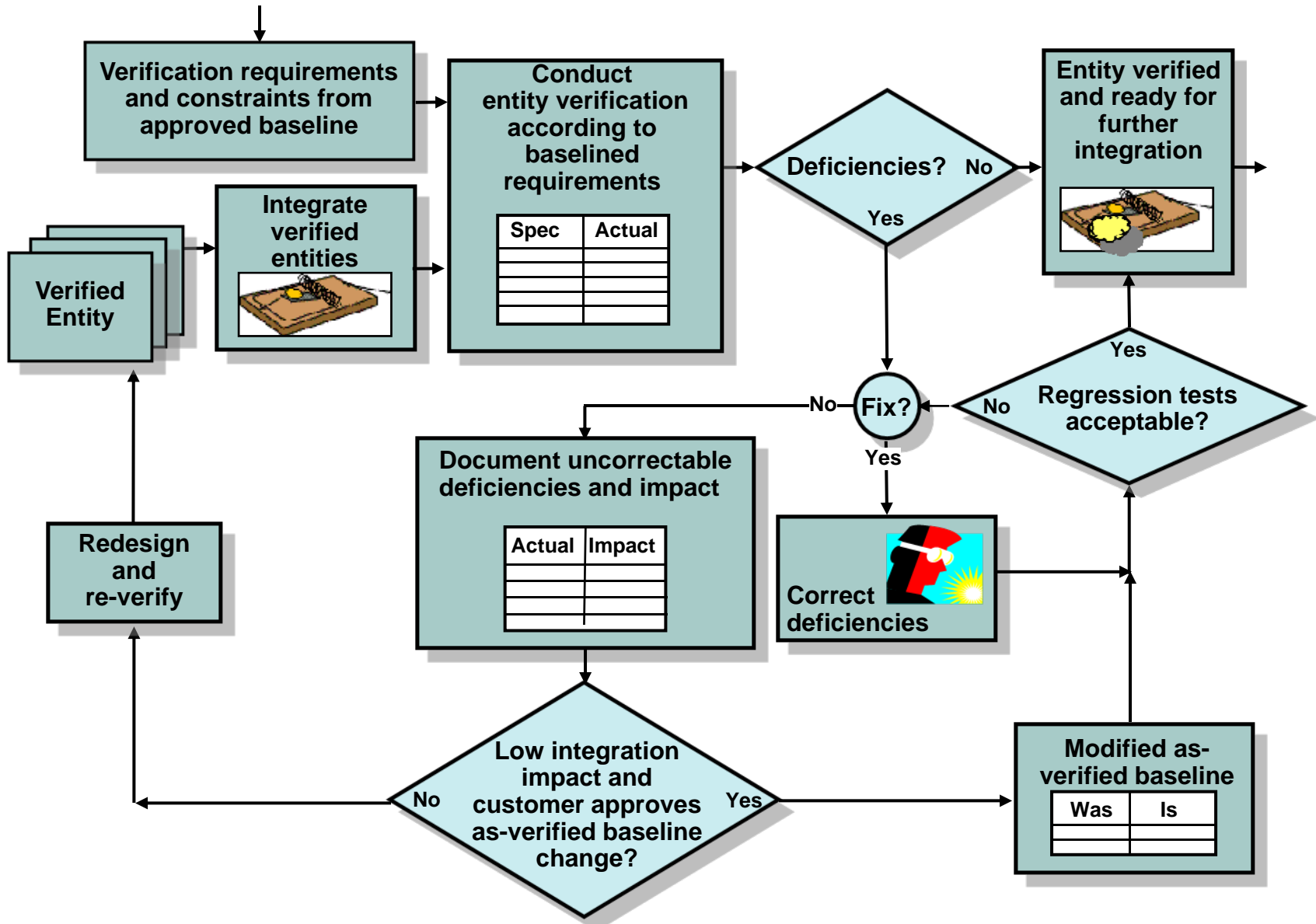
Architecture Development Process

- **Understand the selected concept and strategic objectives**
- **Develop architecture selection criteria**
 - Easy to upgraded
 - Easy to integrate
 - Easy to modify
 - Easy to maintain
 - Easy to manage
 - Maximum use of COTS
 - Etc.
- **Conceive ways of combining entities to achieve the criteria**
 - Hardware, software, people, facilities, etc.
 - Define the interactions
 - Define interfaces
- **Conduct trade study**
 - Select the best and get stakeholders' agreement
- **Conducted at all levels of decomposition**

Systems Integration Course Outline

- **Session #1 – Introduction**
- **Session #2 – Facing Uncertainty in IV&V Decisions**
- **Session #3 – Risk, Design, and IV&V**
- **Session #4 – Planning for IV&V**
- **Session #5 – Integration**
- **Session #6 – Validation**
- **Session #7– Verification**
- **Session #8 – IV&V Anomaly Management**
- **Session #9 – Summary**

Verification Analysis and Resolution Process (VAR)



Software Functional Testing vs. Structural Testing

- **Functional Testing Identifies:**
 - Missing and incorrect functions
 - Interface anomalies
 - Anomalies in data structures or external database access
 - Performance anomalies
 - Initiation and termination anomalies
- **Structural Testing verifies**
 - Independent paths
 - Logical decisions and loops
 - Internal data structures

The Six Steps For Software Walkthrough

Step	Inspection Team Participants	Objectives
1. Planning	Moderator	Scheduling/Distributing Materials
2. Overview	Designer/Coder	Education
3. Preparation	Individual Participants	Self-Study
4. Software Inspection	Entire Team	Find and Record Issues
5. Corrective Actions	Designer/Coder	Improvement
6. Follow Up	Moderator/Designer/Coder	Corrective action assessment

* Adapted from Bell Laboratory

Design Verification Issues

- **Requirements Flaws**
 - Not well enough understood
 - Conflicting
 - Too demanding
- **Design Flaws**
 - Doesn't do the job
- **Design Producibility**
 - Results unachievable with specified technology
 - Software cannot be coded
 - Hardware not be producible
 - Solution too expensive
- **Performance not verifiable**



Lean Principles and Practices

(20 hour elective: Text-"Lean Enterprise Value")

- History/background
- The Lean agenda, from mfg. to enterprise
- The Value Stream
- Waste and adaptability
- The LEM
- Flow as a maxim
- IPPD and SE
- In class exercises
- Supplier relations
- Quality issues
- Lean Systems Engineering
- Lean product devlpt.
- Enterprise implications
- Student report - Apply Lean at "home"

Decision Analysis

Learn a coherent approach to decision making with emphasis on the creation of distinctions, representation of uncertainty by probability, development of alternatives, specification of preference, and the role of these elements in creating a normative approach to decisions.

Learning Objectives:

- Analyze the personal and professional decisions that shape lives and organizations
- Follow a decision approach that scales to treat decisions at all levels of importance
- Use the concepts that clarify thinking and choice
- Avoid the pitfalls of intuitive decision making

Summary

- Stanford has defined a path toward leadership in systems engineering
- Internal and external partnerships are the key
- The Stanford Center for Professional Development is delivering the SE certificate program integrating the capabilities of Stanford faculty and CSM
- Three core and three elective courses
- First sessions in March 2006