Andrés Roldán Aranda

ETSIIT Lecturer

Electronics Department, University of Granada - SPAIN

T.1 - Previous concepts

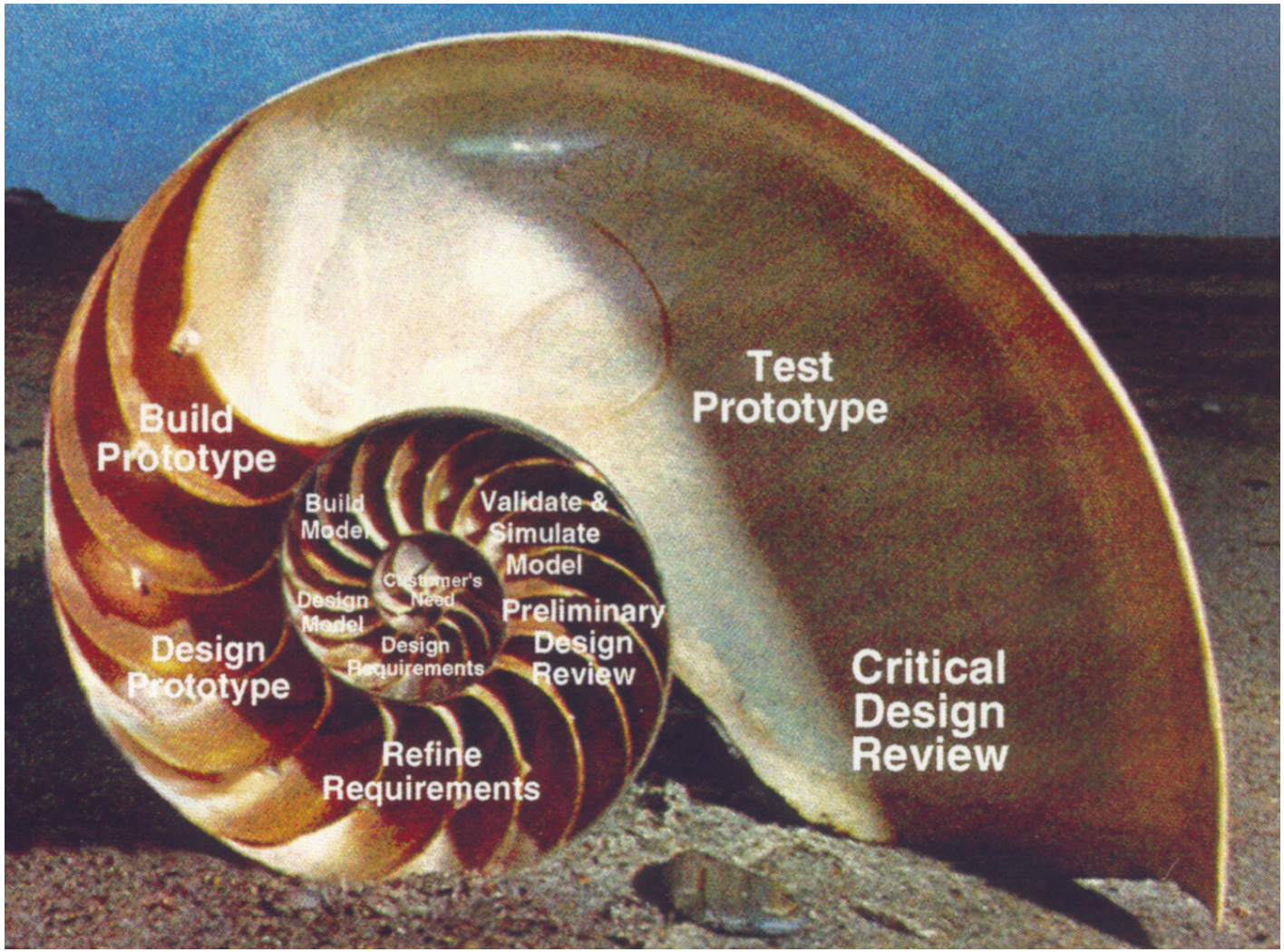
Concept, Design, Prototyping & Project Management

Printed Circuits Technologies 15

Table of contents

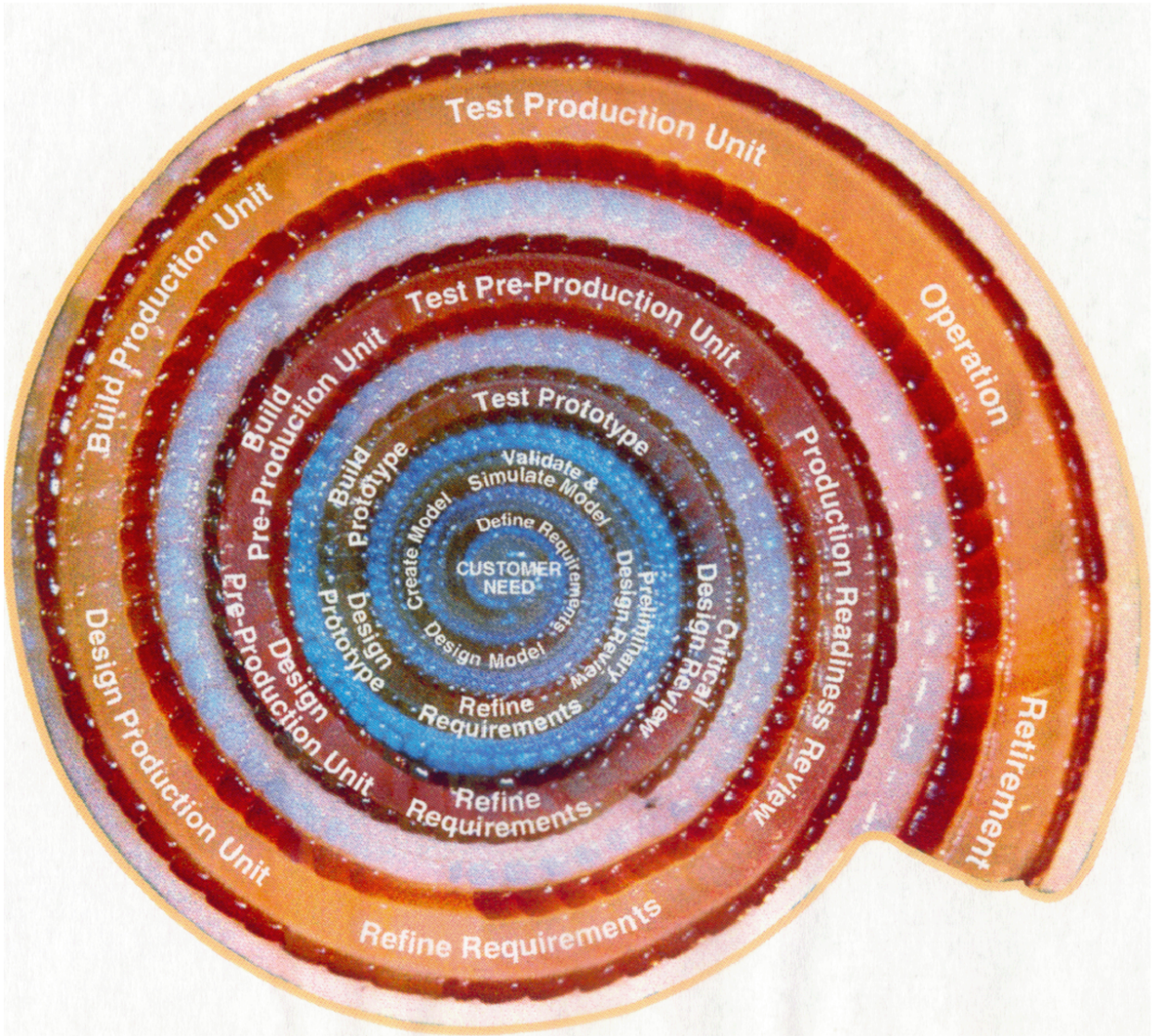
System design process (SDP)

First steps in the SDP

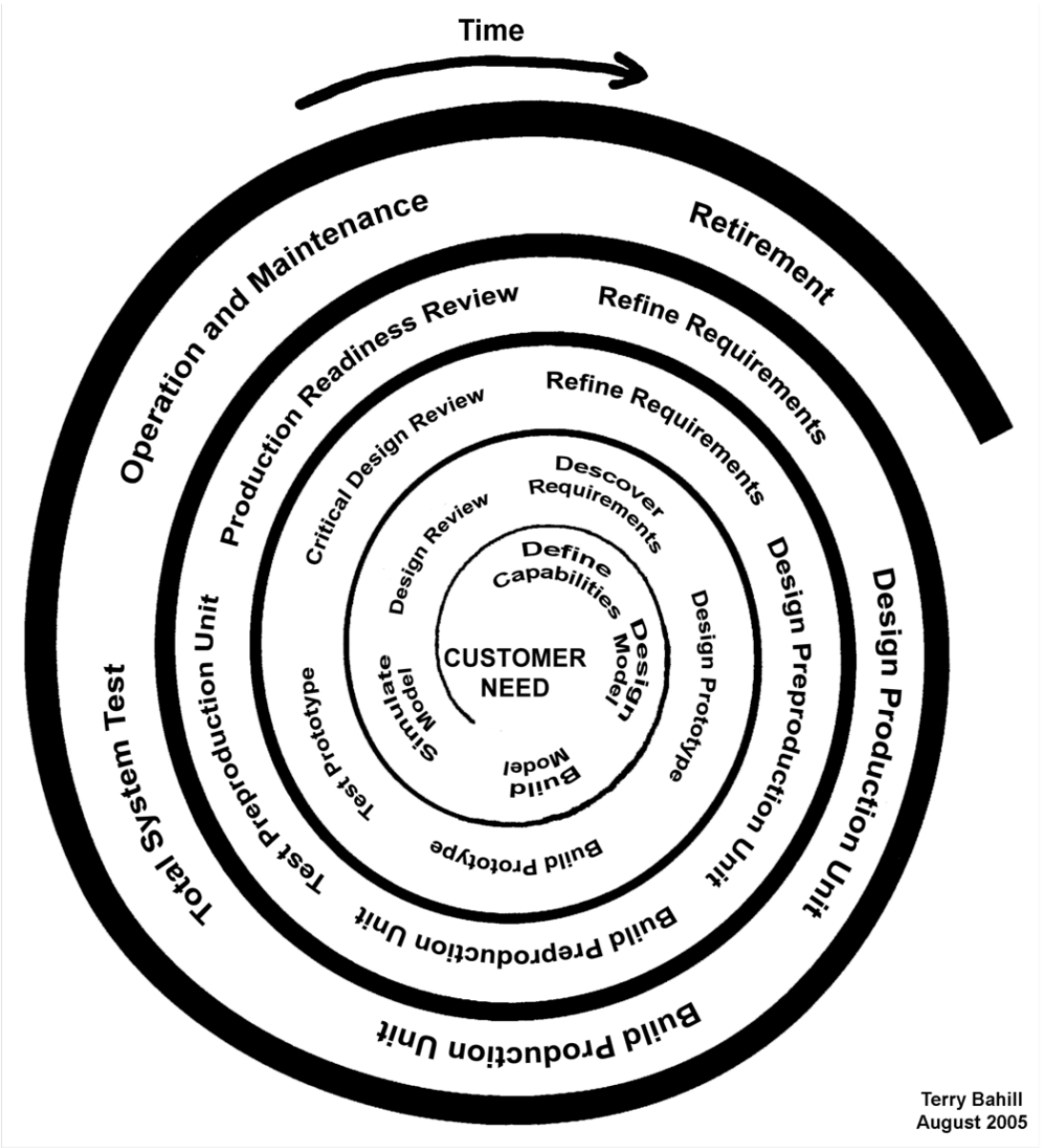


The System Design process

The hole process, from the beginning to the end



Spiral Lifecycle model



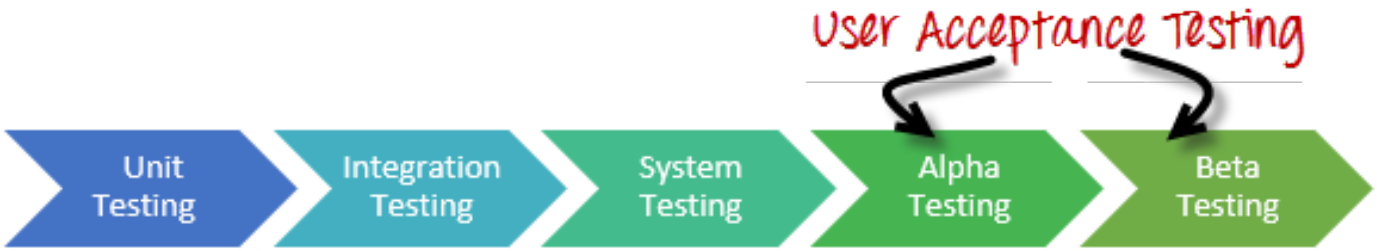
Alpha Beta Testing - DeMystified

α-test

Alpha testing is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or public. The focus of this testing is to simulate real users by using blackbox and whitebox techniques.

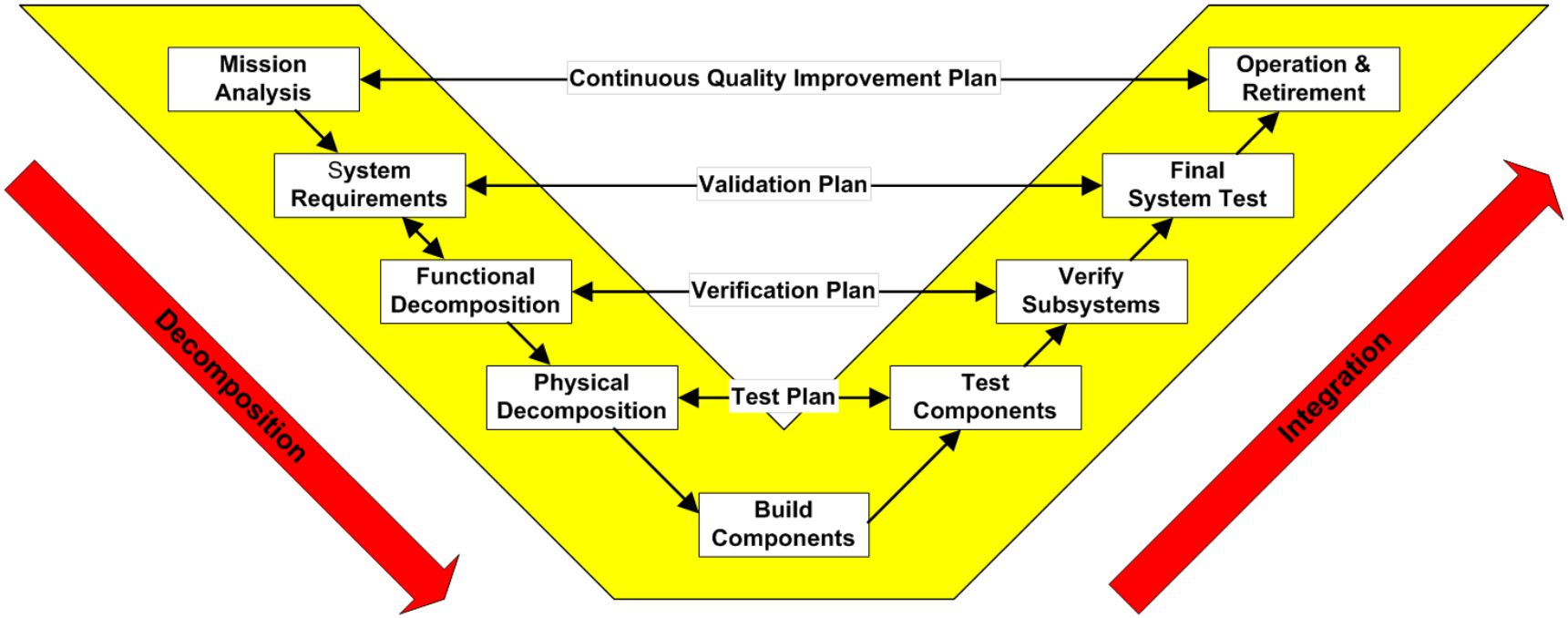
β-test

Beta Testing of a product is performed by "real users" of the software application in a "real environment" and can be considered as a form of external user acceptance testing.



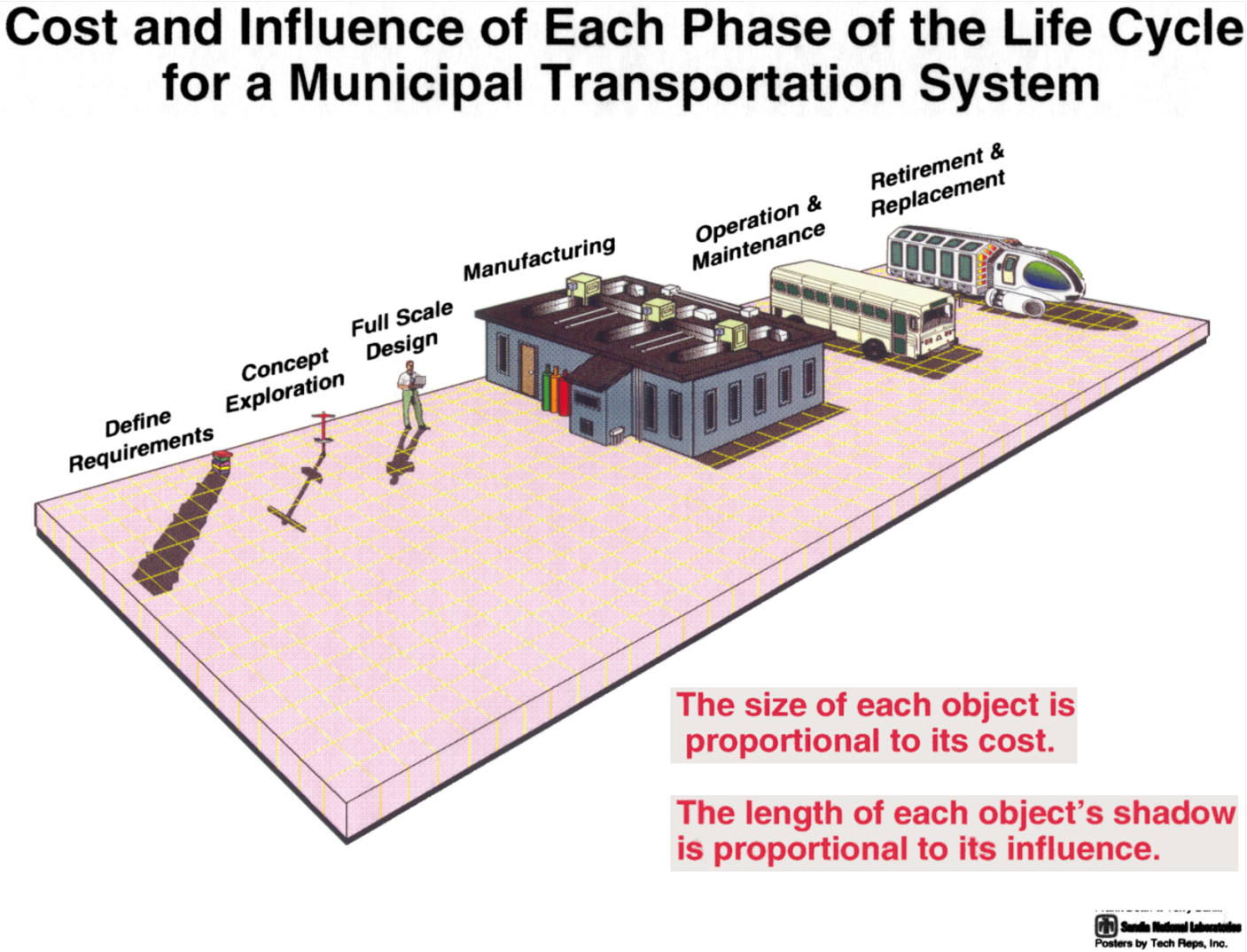
Vee Life cycle model

Other way of thinking

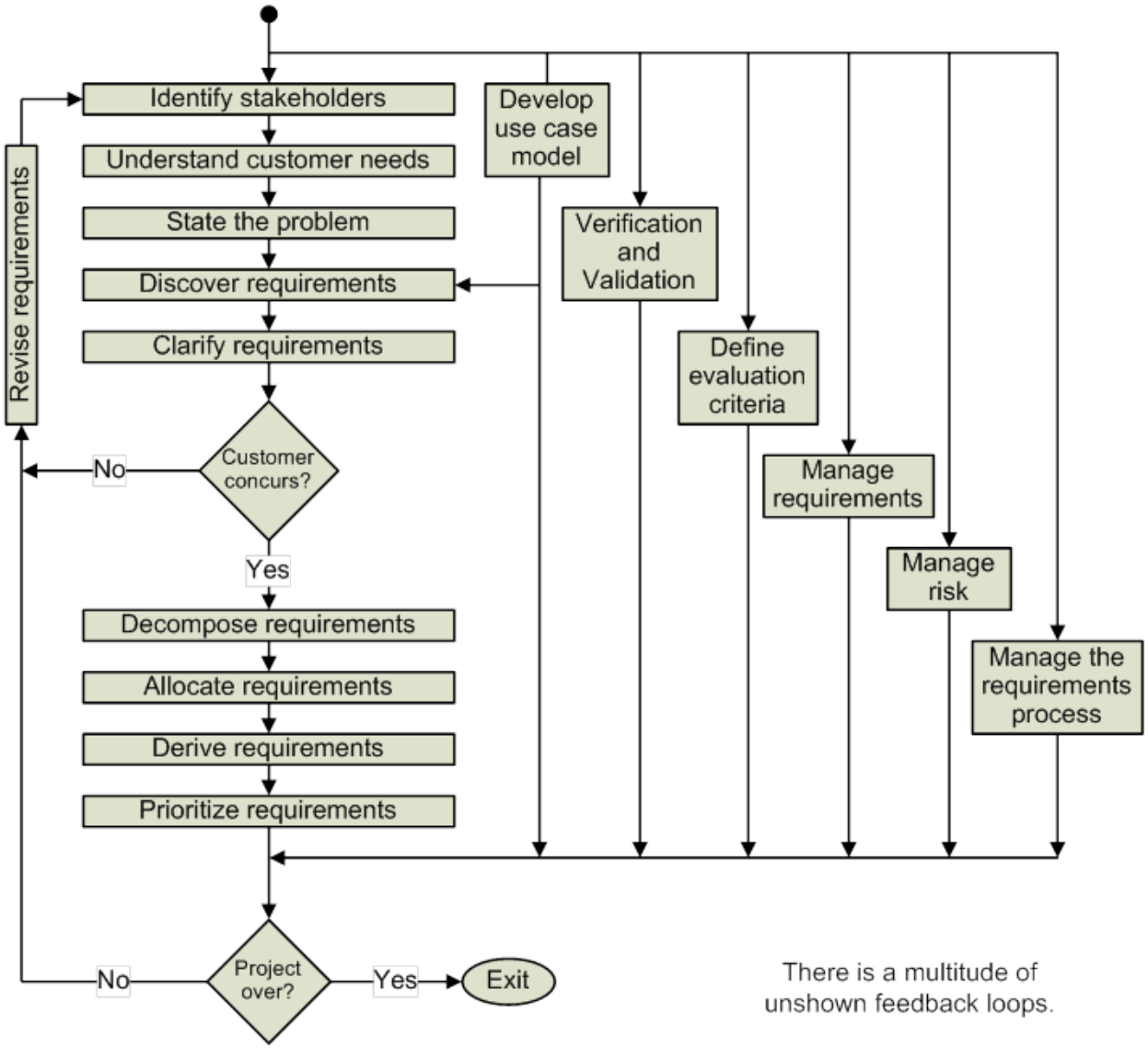


The design downstroke and the manufacturing upstroke

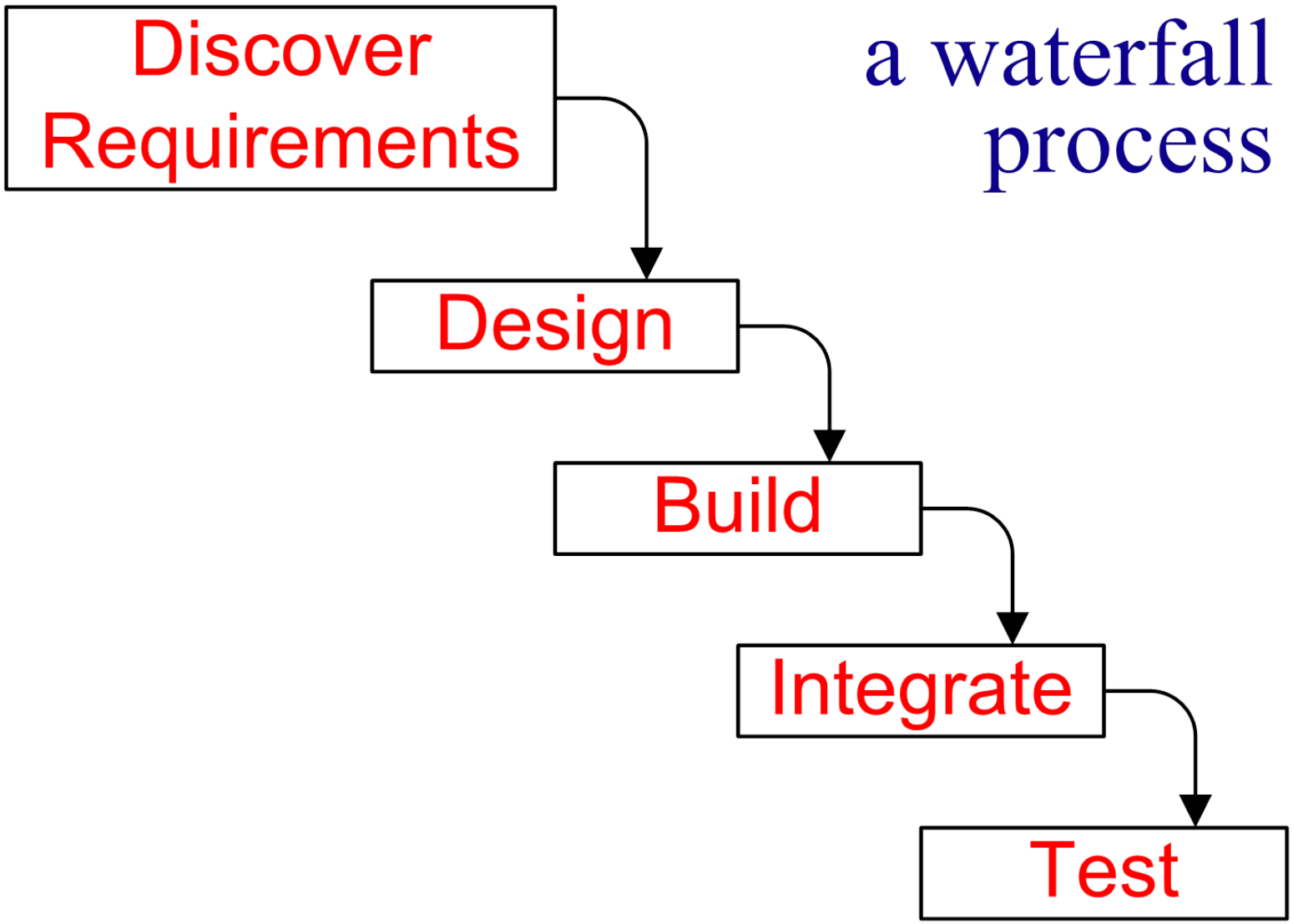
Cost & influence of each Phase of the Life Cycle



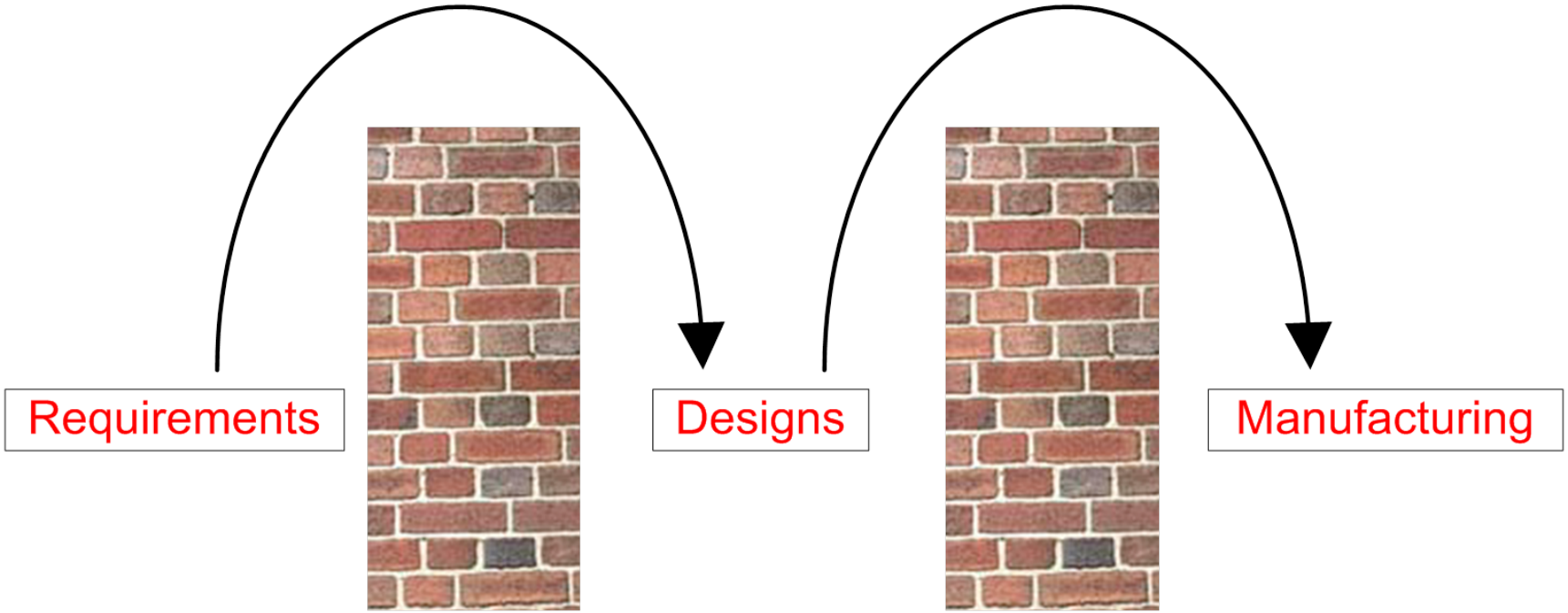
Requirements discovery process



Systems engineering is not !

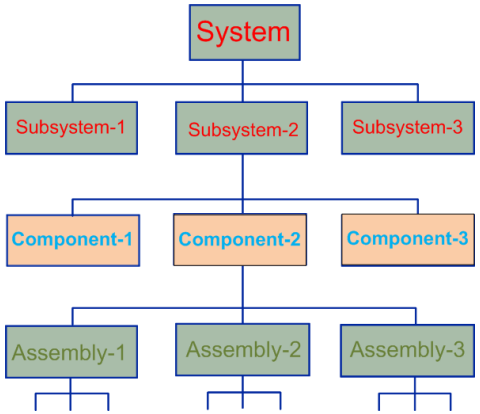


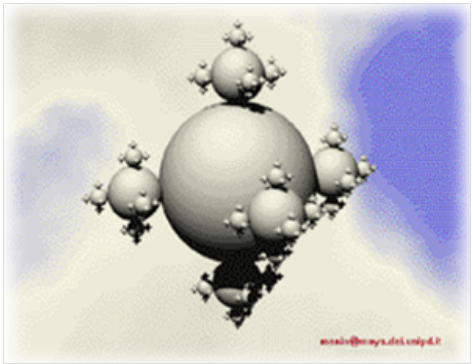
Systems engineering is not a ....!



a throw it over the wall process

Systems engineering is a fractal process





The systems engineering process is applied at levels of greater and greater detail.

It is applied to the system, then to the subsystems, then to the components, etc.
Similarly for the *fractal pattern* above, the same algorithm was applied at the large structural level, then at the medium-scale level, then at the fine-detail level, etc.

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities
- Design the interfaces

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities
- Design the interfaces
- Produce satisfying designs

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities
- Design the interfaces
- Produce satisfying designs
- Do not optimize early

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities
- Design the interfaces
- Produce satisfying designs
- Do not optimize early
- Maintain an updated model of the system

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities
- Design the interfaces
- Produce satisfying designs
- Do not optimize early
- Maintain an updated model of the system
- Develop stable intermediates

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities
- Design the interfaces
- Produce satisfying designs
- Do not optimize early
- Maintain an updated model of the system
- Develop stable intermediates
- Use evolutionary development

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities
- Design the interfaces
- Produce satisfying designs
- Do not optimize early
- Maintain an updated model of the system
- Develop stable intermediates
- Use evolutionary development
- Understand your enterprise

Principles of good design
Part I

Terry Bahill and Rick Botta, Fundamental Principles of Good System Design, *Engineering Management Journal, 20(4), 9-17, December 2008*.

- Use models to design systems
- Use hierarchical, top-down design
- Work on high-risk entities first
- Prioritize
- Control the level of interacting entities
- Design the interfaces
- Produce satisfying designs
- Do not optimize early
- Maintain an updated model of the system
- Develop stable intermediates
- Use evolutionary development
- Understand your enterprise

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states
- Rapid prototyping

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states
- Rapid prototyping
- Develop iteratively and test immediately

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states
- Rapid prototyping
- Develop iteratively and test immediately
- Create modules

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states
- Rapid prototyping
- Develop iteratively and test immediately
- Create modules
- Create libraries of reusable entities

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states
- Rapid prototyping
- Develop iteratively and test immediately
- Create modules
- Create libraries of reusable entities
- Use open standards

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states
- Rapid prototyping
- Develop iteratively and test immediately
- Create modules
- Create libraries of reusable entities
- Use open standards
- Identify things that are likely to change

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states
- Rapid prototyping
- Develop iteratively and test immediately
- Create modules
- Create libraries of reusable entities
- Use open standards
- Identify things that are likely to change
- Write extension points

Principles of good design
Part II

- State what not how (polymorphism)
- List functional requirements in the use cases
- Allocate each function to only one component
- Do not allow undocumented functions
- Provide observable states
- Rapid prototyping
- Develop iteratively and test immediately
- Create modules
- Create libraries of reusable entities
- Use open standards
- Identify things that are likely to change
- Write extension points

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins
- Design for testability

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins
- Design for testability
- Design for evolvability

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins
- Design for testability
- Design for evolvability
- Build in preparation for buying

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins
- Design for testability
- Design for evolvability
- Build in preparation for buying
- Do a sensitivity analysis

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins
- Design for testability
- Design for evolvability
- Build in preparation for buying
- Do a sensitivity analysis
- Create a new design process

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins
- Design for testability
- Design for evolvability
- Build in preparation for buying
- Do a sensitivity analysis
- Create a new design process
- Search for unintended consequences

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins
- Design for testability
- Design for evolvability
- Build in preparation for buying
- Do a sensitivity analysis
- Create a new design process
- Search for unintended consequences
- Change the behavior of people

Principles of good design
Part III

- Group data and behavior
- Use ▸ data hiding
- Write a glossary of relevant terms
- Envelope requirements
- Create design margins
- Design for testability
- Design for evolvability
- Build in preparation for buying
- Do a sensitivity analysis
- Create a new design process
- Search for unintended consequences
- Change the behavior of people

Purpose of the principles

- Using these principles will increase the probability of producing good designs.
- These design principles will help make an item reusable in a new system.

Purpose of the principles

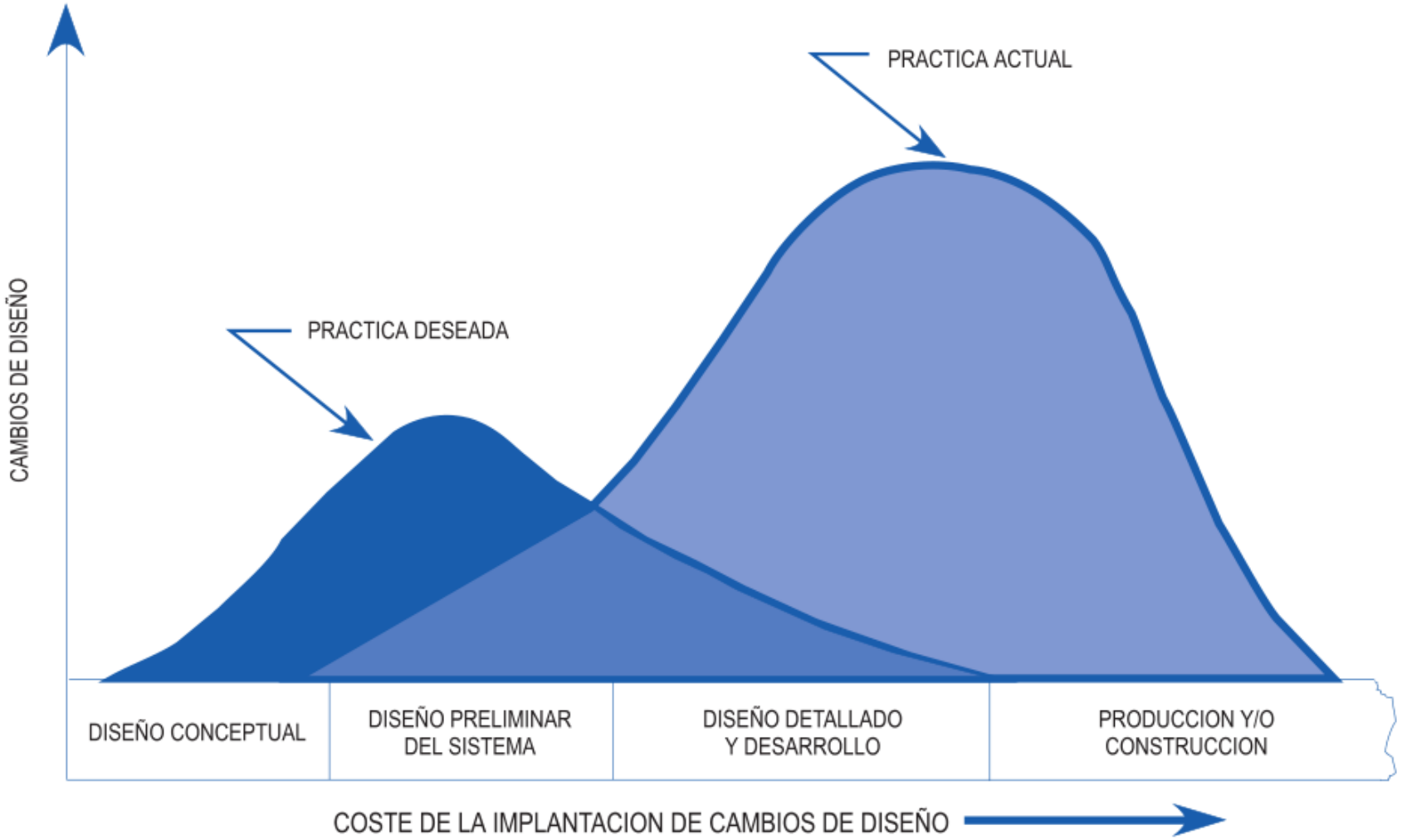
- Using these principles will increase the probability of producing good designs.
- These design principles will help make an item reusable in a new system.
- Not surprisingly, these same principles can help reduce redesign costs when requirements change.

Purpose of the principles

- Using these principles will increase the probability of producing good designs.
- These design principles will help make an item reusable in a new system.
- Not surprisingly, these same principles can help reduce redesign costs when requirements change.

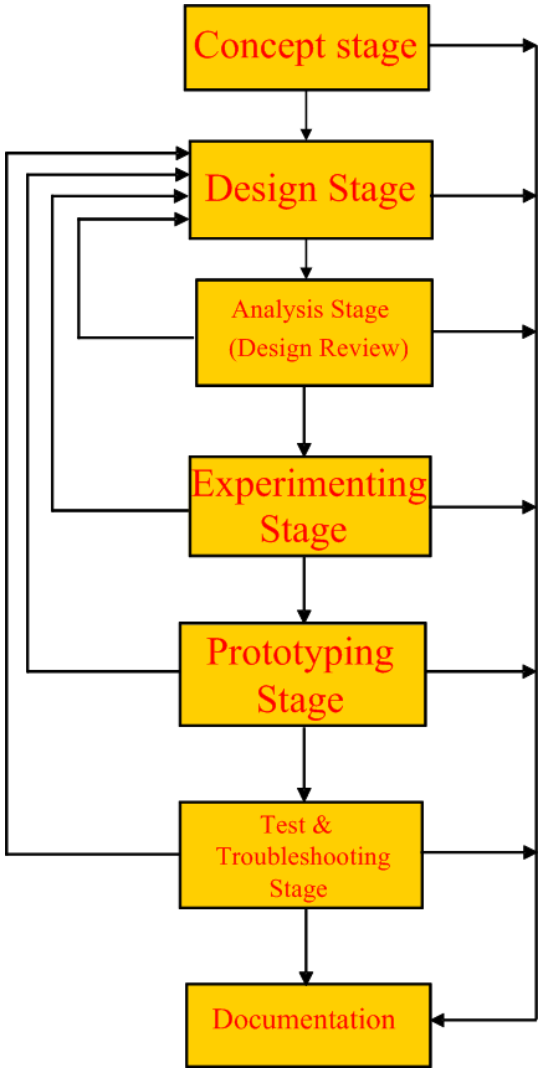
Impact of the changes introduced in the desing!

Why are we doing wrong?



Design Phases!

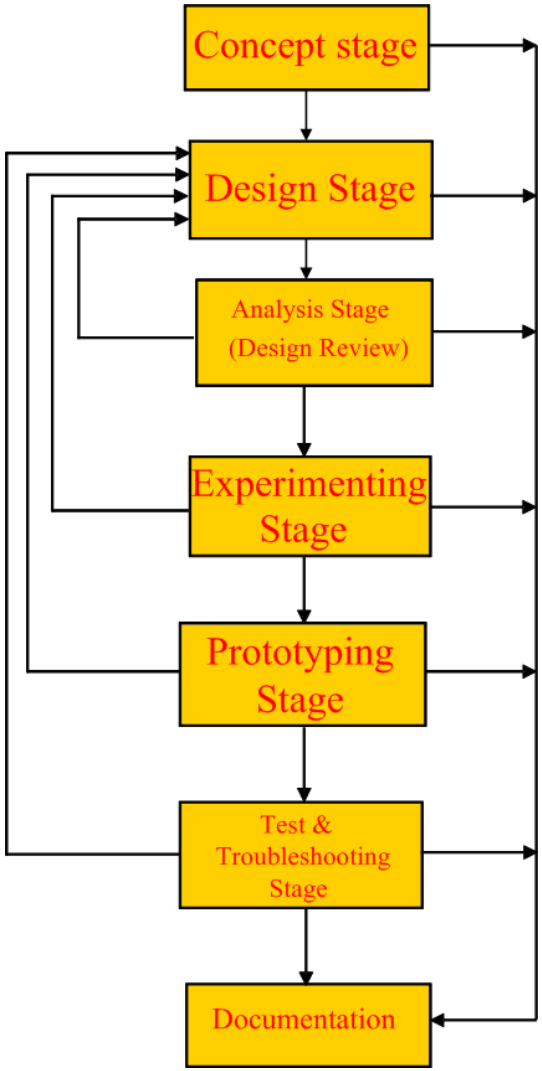
Designing algorithm



- Concept
- Design

Design Phases!

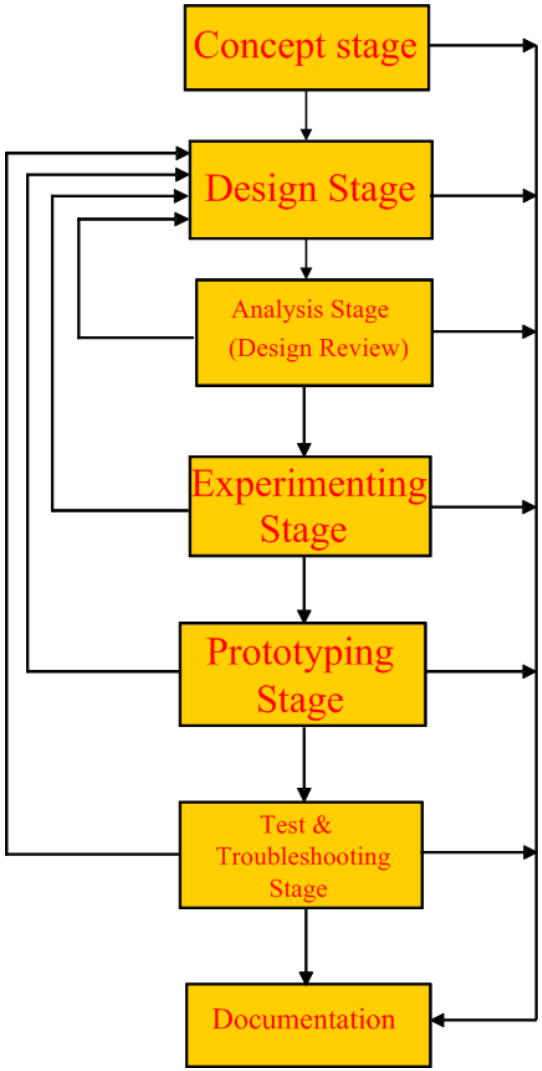
Designing algorithm



- Concept
- Design
- Analysis

Design Phases!

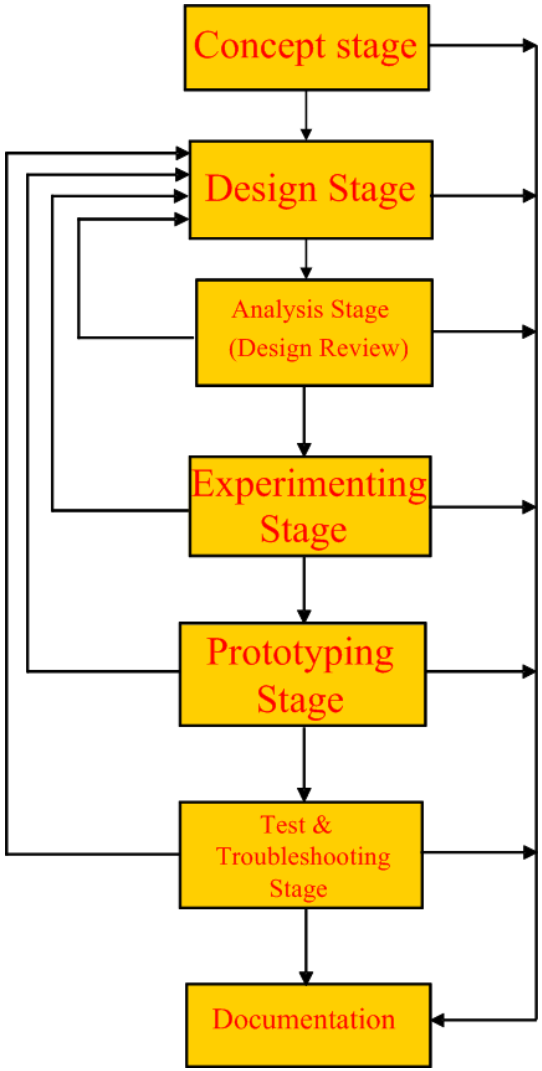
Designing algorithm



- Concept
- Design
- Analysis
- Experiment

Design Phases!

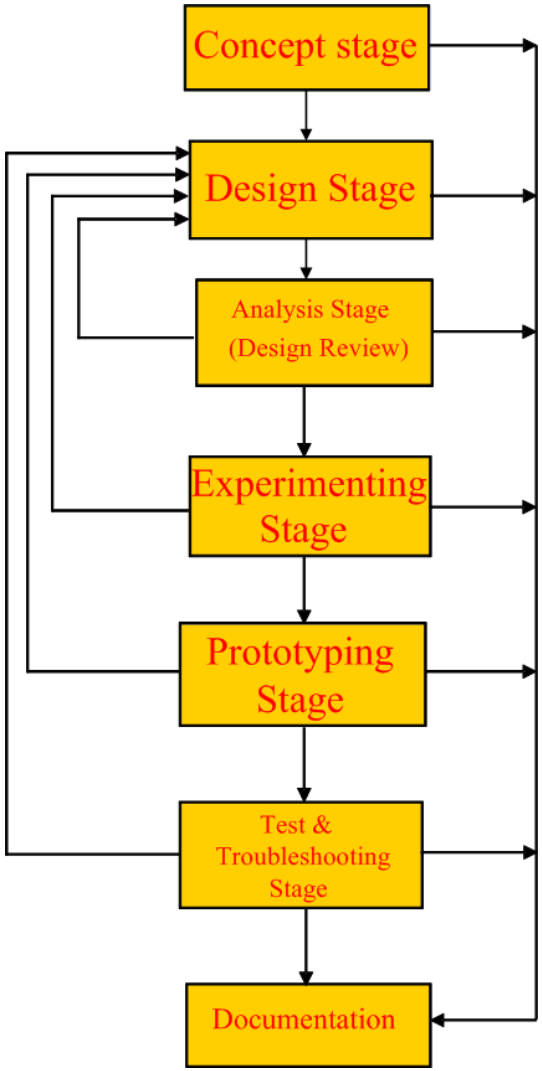
Designing algorithm



- Concept
- Design
- Analysis
- Experiment
- Prototype

Design Phases!

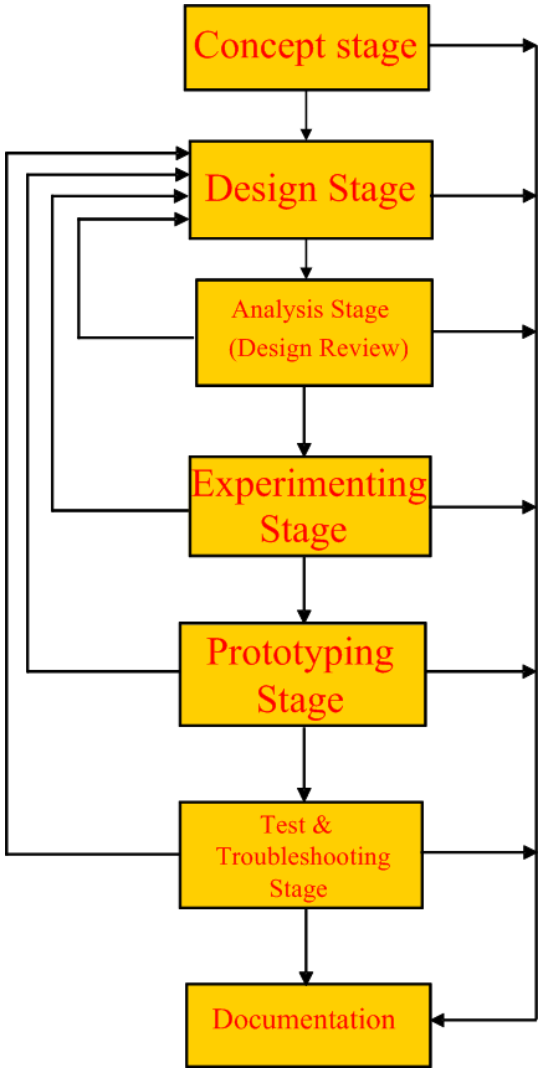
Designing algorithm



- Concept
- Design
- Analysis
- Experiment
- Prototype
- Test & Troubleshoot

Design Phases!

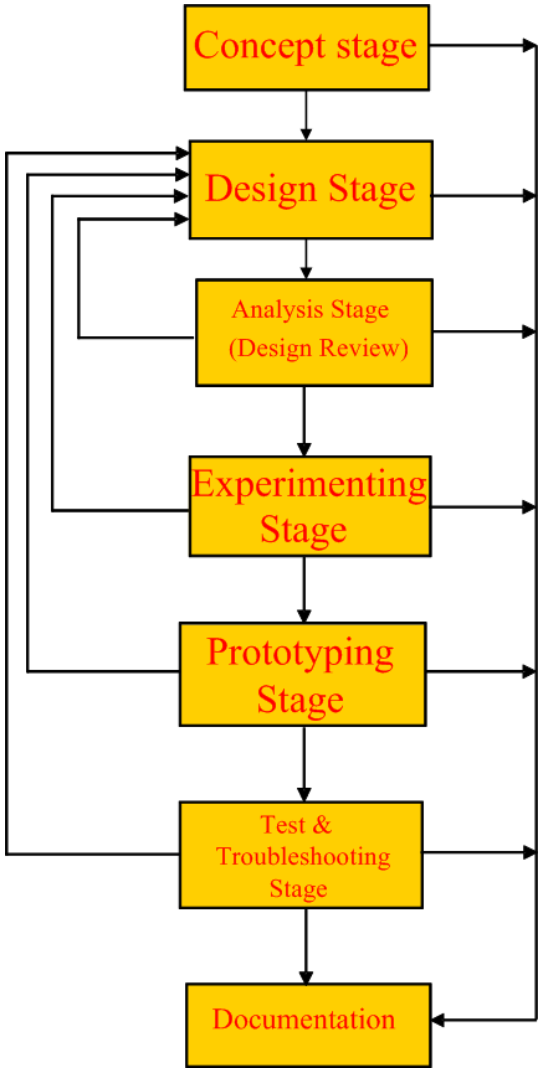
Designing algorithm



- Concept
- Design
- Analysis
- Experiment
- Prototype
- Test & Troubleshoot
- Document

Design Phases!

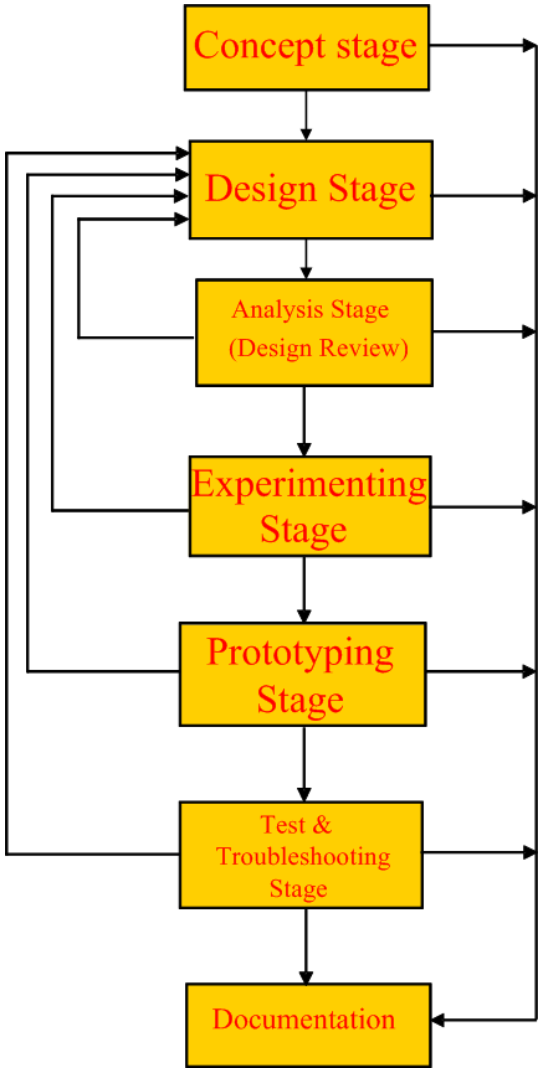
Designing algorithm



- Concept
- Design
- Analysis
- Experiment
- Prototype
- Test & Troubleshoot
- Document

Design Phases!

Designing algorithm



- Concept
- Design
- Analysis
- Experiment
- Prototype
- Test & Troubleshoot
- Document

Design

- System approach
  - Electronic Circuitry
  - Role of Input & Output Transducers
  - Product Packaging
  - User Needs
  - TQM- ▸ Total Commitment to Quality ▸ Vishay Tantalum Capacitor - Total quality Example

Design Review

- To check whether:
  - the requirements are met
  - the design is optimum
  - right components are selected
  - Quality aspects are taken in to
  - it is practical to go in for production
  - It is a time bound proposal

Experimentation

- To check whether the the circuit functions - Design Validation
- No concern with project lay out and packaging

Experimentation

- To check whether the the circuit functions - Design Validation
- No concern with project lay out and packaging
- It is a quick and easy method of assembling the components in to functioning unit

Experimentation

- To check whether the the circuit functions - Design Validation
- No concern with project lay out and packaging
- It is a quick and easy method of assembling the components in to functioning unit
- Minor & major modifications can be carried out in this stage

Experimentation

- To check whether the the circuit functions - Design Validation
- No concern with project lay out and packaging
- It is a quick and easy method of assembling the components in to functioning unit
- Minor & major modifications can be carried out in this stage

Experimentation

- To check whether the the circuit functions - Design Validation
- No concern with project lay out and packaging
- It is a quick and easy method of assembling the components in to functioning unit
- Minor & major modifications can be carried out in this stage

Prototyping

- PCB Design
- PCB Fabrication

Prototyping

- PCB Design
- PCB Fabrication
- PCB Assembly

Prototyping

- PCB Design
- PCB Fabrication
- PCB Assembly
- Product Packaging

Prototyping

- PCB Design
- PCB Fabrication
- PCB Assembly
- Product Packaging

Testing & Troubleshooting a Prototype Project

- Preliminary testing
- Operational Testing

Testing & Troubleshooting a Prototype Project

- Preliminary testing
- Operational Testing
- Troubleshooting

Testing & Troubleshooting a Prototype Project

- Preliminary testing
- Operational Testing
- Troubleshooting
- Performance testing

Testing & Troubleshooting a Prototype Project

- Preliminary testing
- Operational Testing
- Troubleshooting
- Performance testing

Final Documentation

1. Test Results Documentation
2. Summary & Recommendations Document

Final Documentation

1. Test Results Documentation
2. Summary & Recommendations Document

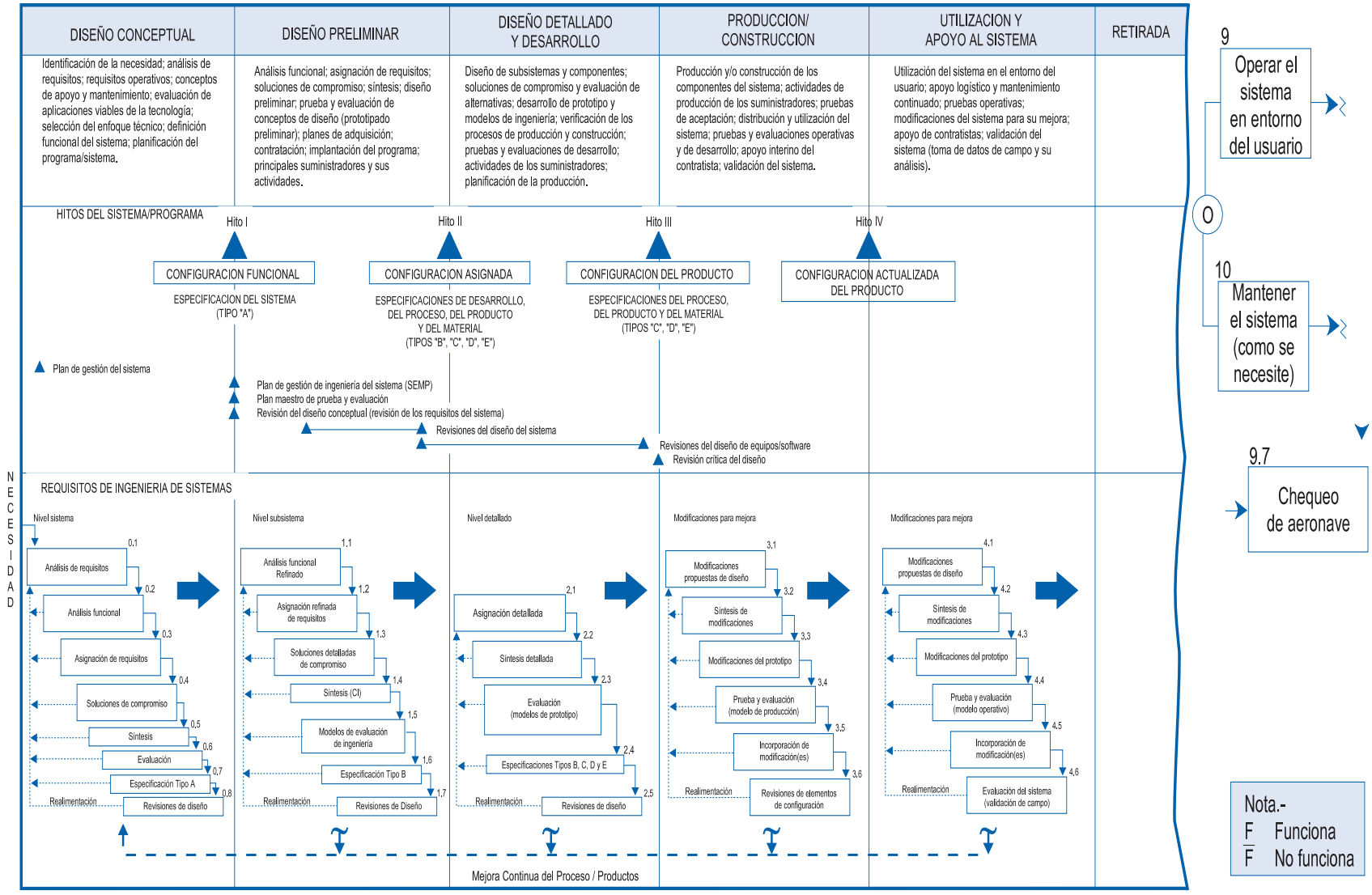


Figura 7. - EL CICLO DE VIDA DEL SISTEMA ("CONFIGURACION") -