**INTRODUCTION**

WHY RE?

* error propagation in lifecycle
  + erroneous design in design phase
  + program implemented according to erroneous design
  + uncorrectable/hidden errors in testing
  + Etc.
* Examples:
  + Bell labs/ IBM ⇨ 80% of defects inserted in requirements phase
  + U.S.A.F. ⇨ 36% of defects due to faulty requirements translation
  + Voyager/Galileo ⇨ 194/197 defects attributed to requirements phase

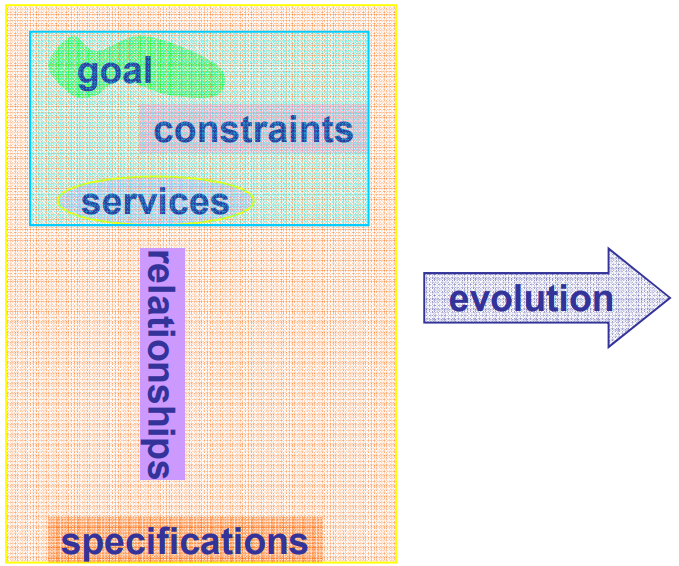
What Factors Contribute to Project Success?

* the CHAOS ten:
  + executive management support
  + user involvement
  + experienced project manager
  + clear business objectives
  + minimized scope
  + standard SW infrastructure
  + firm basic requirements
  + formal methodology
  + reliable estimates
  + other

How Costly Are Requirements Errors?

* takes 1-4 working hours to find & fix bug through inspections
* takes 15-20 working hours to find & fix a bug in a function/system test
* 200:1 cost ratio between finding errors in requirements stage vs. maintenance stage of SW lifecycle

What Is RE?

* RE is branch of systems engineering concerned with real world GOALS for, SERVICES provided by, and CONSTRAINTS on SW systems
* RE is also concerned with RELATIONSHIP of these factors to precise SPECIFICATIONS of system behavior and to their EVOLUTION over time

Role Of Requirements Engineering

* agreement regarding the requirements b/t system developers, customers, and end-users
* the basis for SW design
* support for verification & validation
* support four system evolution

Systematic Decision Making is Essential

* Requirements Engineering is about determining
  + problems with the current status (As-Is)
  + objectives to achieve
  + changes to bring about for a better future (To-Be)

What’s Essential?

* Modeling
  + “A model is a pattern, plan, representation (especially in miniature), or description designed to show the main object or workings of an object, system, or concept” [ Wikipedia]
* Systematic decision making
  + “Decision making can be regarded as an outcome of mental processes (cognitive process) leading to the selection of a course of action among several alternatives. Every decision making process produces a final choice.[1] The output can be an action or an opinion of choice” [Wikipedia]

What is RE Really about?

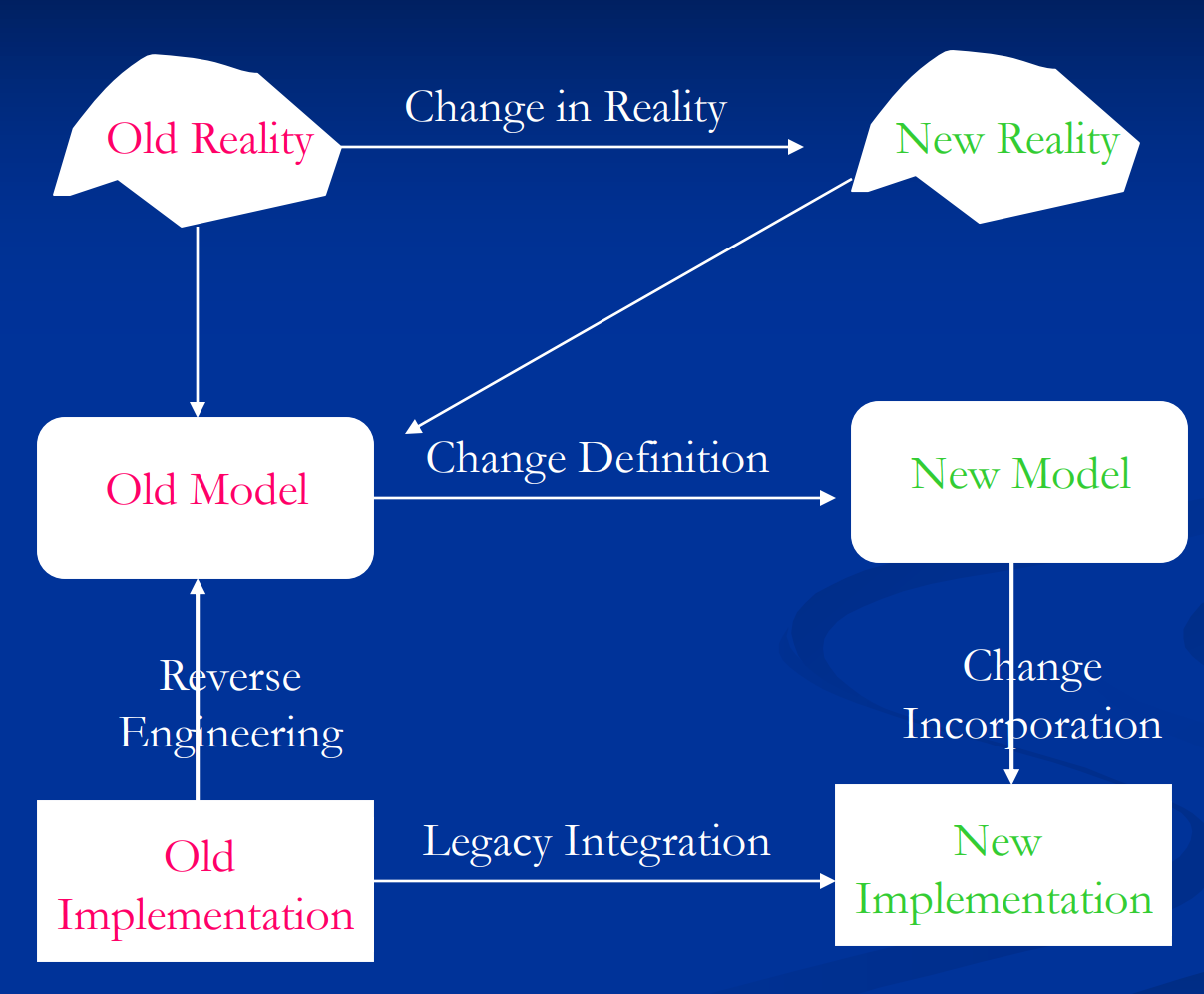
* Finding out EXACTLY what the customer wants

**RE PROCESSES**

What is a Process?

* Given input, transforms it into output
* Consist of a set of activities
* Process descriptions are also specifications

The Basic RE Evolutionary Process



* Evolution is inevitable – traceability is more than a virtue

A Basic Framework [Loucopolos]

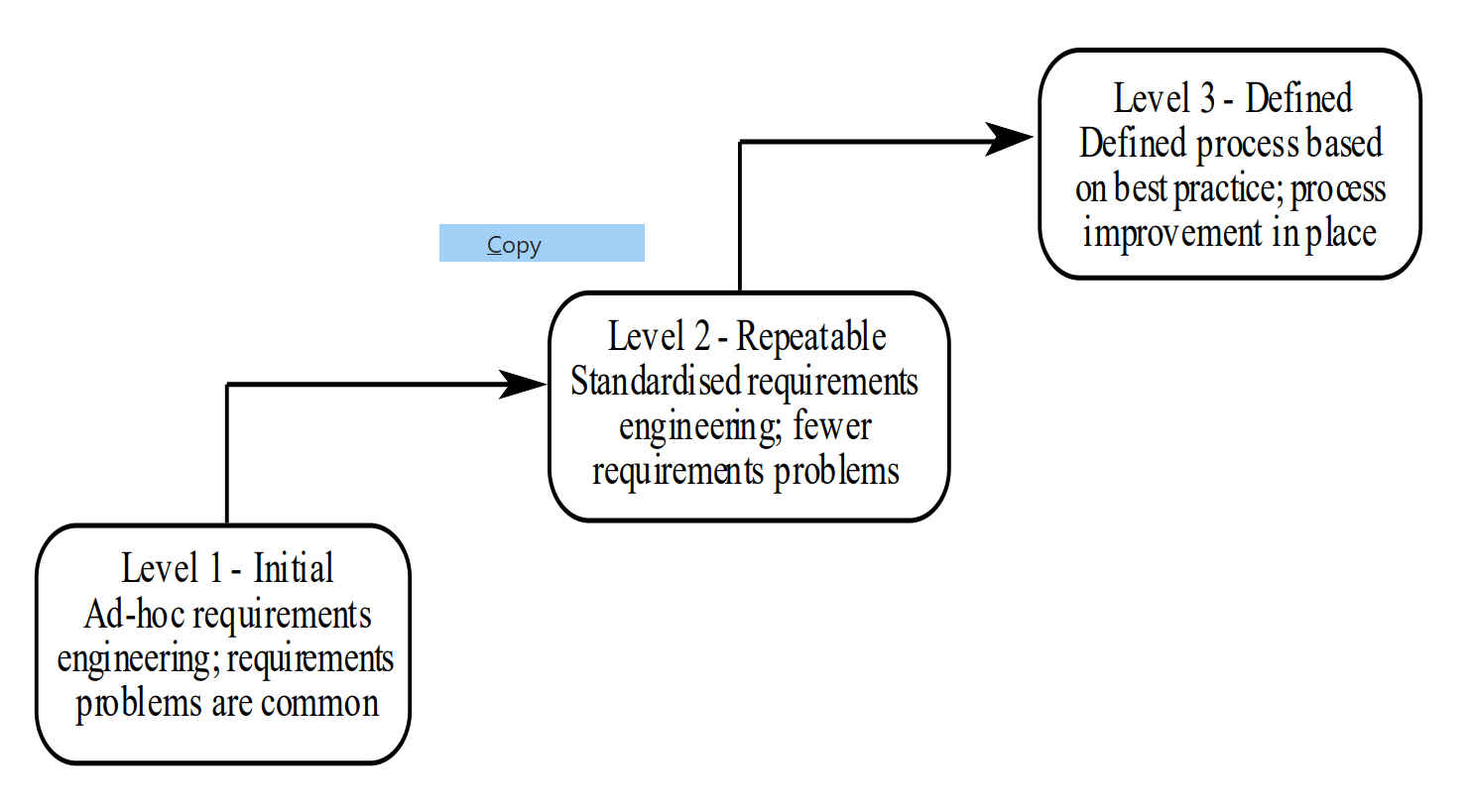
* 3 fundamental activities:
  + understand the problem
  + (formally) describe the problem
  + attain an agreement on the problem
* Elicitation:
  + determine what’s really needed, why needed, whom to talk to
* Specification:
  + produce a (formal) RS model: translate "vague" into "concrete", etc. make various decisions on what & how
* Validation:
  + assure that the RS model satisfies the users’ needs

Spiral Model [KotonyaSummerville98]

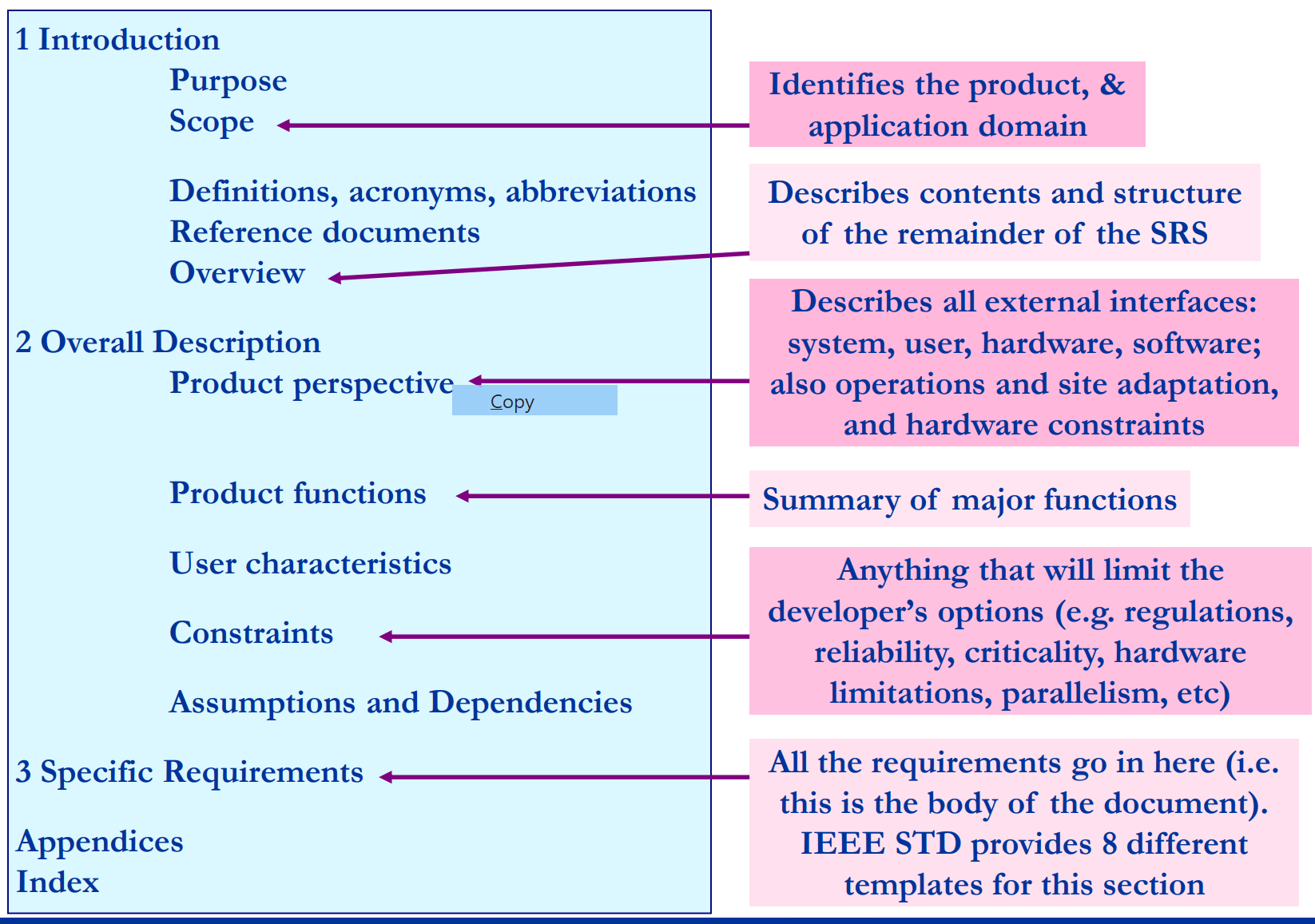
* Requirements elicitation:
  + Requirements discovered through consultation with stakeholders
* Requirements analysis and negotiation:
  + Requirements are analysed and conflicts resolved through negotiation
* Requirements documentation:
  + A requirements document is produced
* Requirements validation:
  + The requirements document is checked for consistency and completeness

RAD (Role Actor Diagram)

A RE Process Maturity Model (base on CMM)



IEEE Standard for SRS



RE in Agile Methods

* Basic Philosophy:
  + Reduce communication barriers
  + Programmer interacts with customer
  + Reduce document-heavy approach
    - Documentation is expensive and of limited use
  + Have faith in the people
    - Don’t need fancy process models to tell them what to do!
  + Respond to the customer
    - Rather than focussing on the contract
* Weaknesses
  + Relies on programmer’s memory
    - Code can be hard to maintain
  + Relies on oral communication
    - Misinterpretation possible
  + Assumes single customer representative
    - Multiple viewpoints not possible
  + Only short term planning
    - No longer term vision

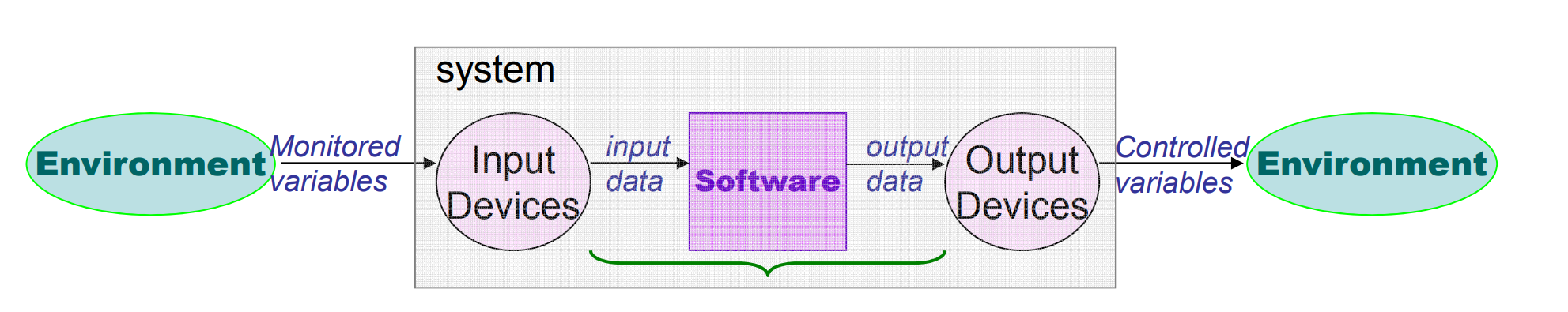
RE in V Model

**RE MODELS**

The Why-What-How model

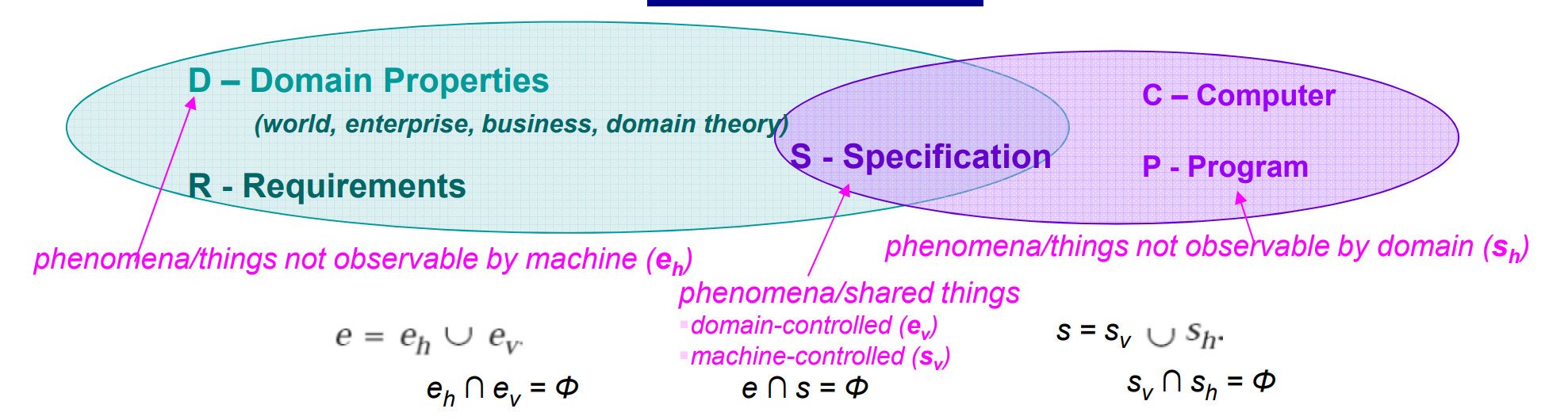
* assessment of system’s needs:
  + WHY a system is needed ⇨ enterprise requirements
  + WHAT system features will serve/satisfy this context ⇨ functional requirements
  + how the system is to be constructed (not about design) ⇨ non-functional requirements

The 4-Variable Model



* NAT(m, c):
  + describes nature without making any assumptions about the system;
* REQ(m, c):
  + describes the desired system behavior;
* IN(m, i):
  + relates the monitored real-world values to their corresponding internal representation;
* OUT(o, c):
  + relates the software-generated outputs to external system-controlled values; and
* SOF(i, o):
  + relates program inputs to program outputs.
* What are the 4 variables?
  + REQ, IN, OUT, SOF

The WRSPM Model

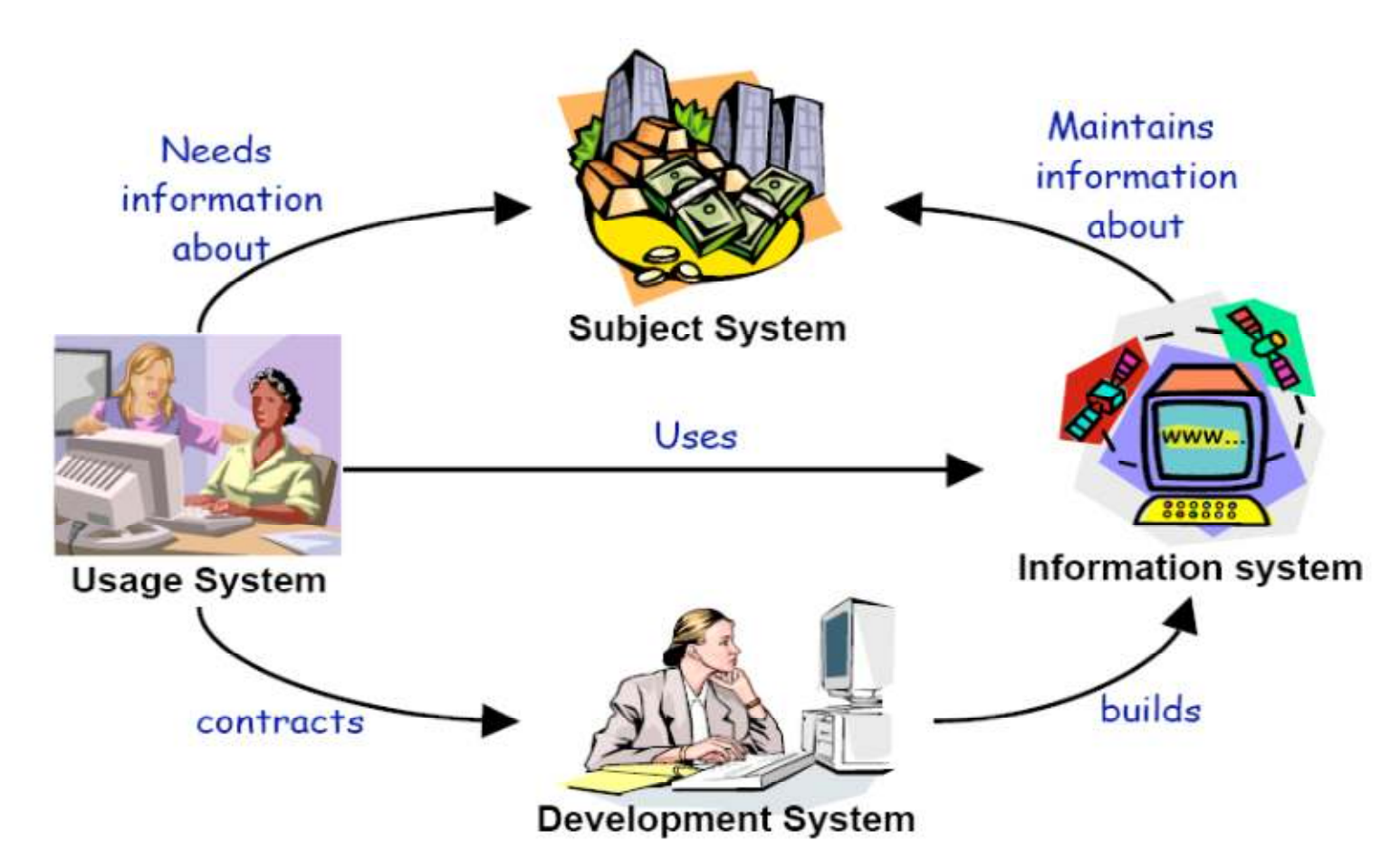


* Domain Properties: (indicative, = assumptions=domain knowledge)
  + things in the environment (application domain) that are true regardless of the proposed system
* Requirements: (optative)
  + things in the application domain that we wish to be made true through the proposed system
    - “Many phenomena not accessible by the machine”
  + Should only contain information about the ENVIRONMENT
* Specification:
  + a description of the behaviors that the program must have in order to meet the requirements
    - "Can, and should, only be written in terms of shared phenomena”
* Requirements should contain nothing but information about the environment.
* Requirements describe what is observable at the interface between the environment and the machine – hence exist only in the environment (Anything else is regarded as implementation bias)
* see coffee machine example

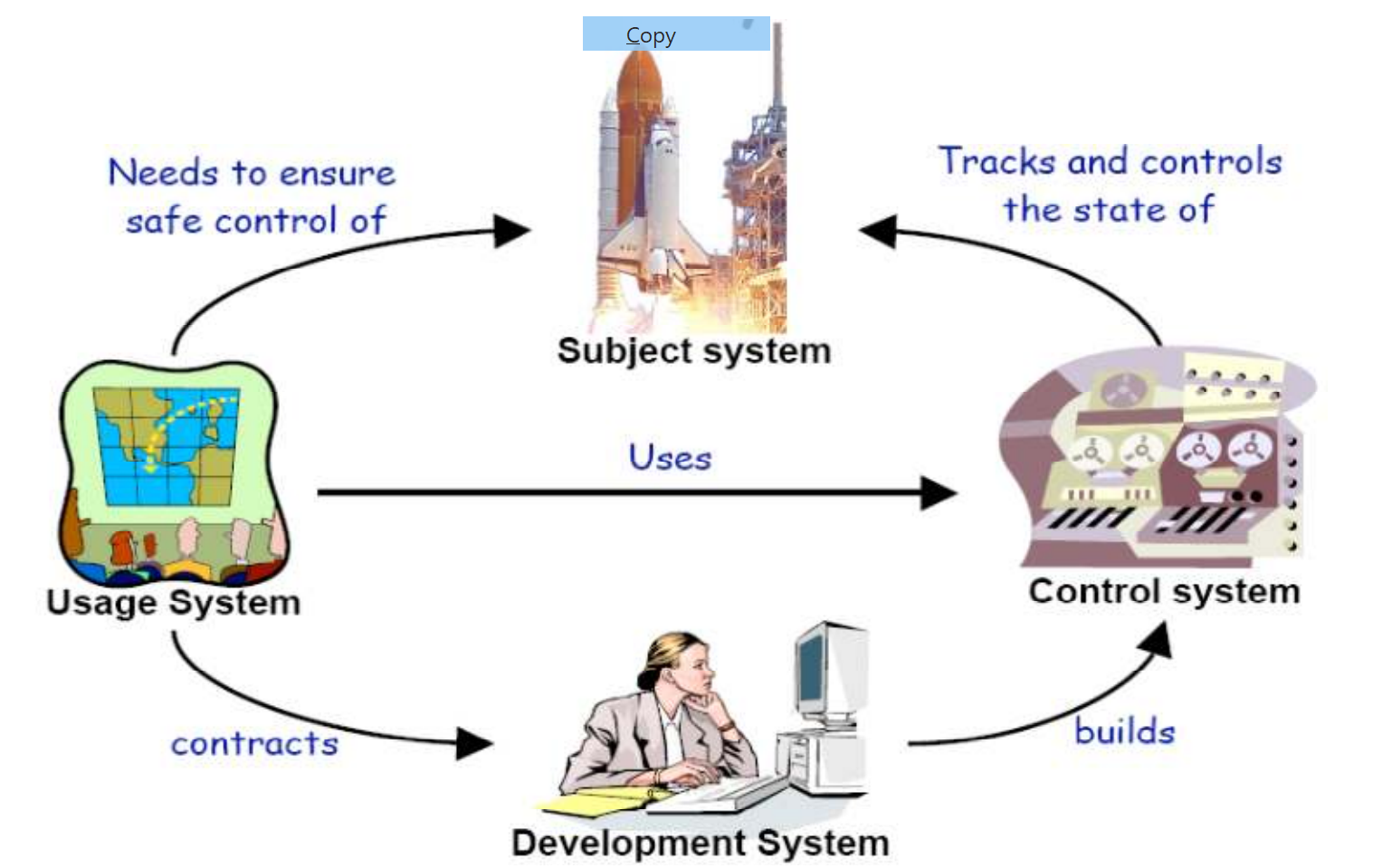
Four Worlds Of RE

* subject world
* user world
* developer world
* system world

Four Worlds of RE for Information Systems

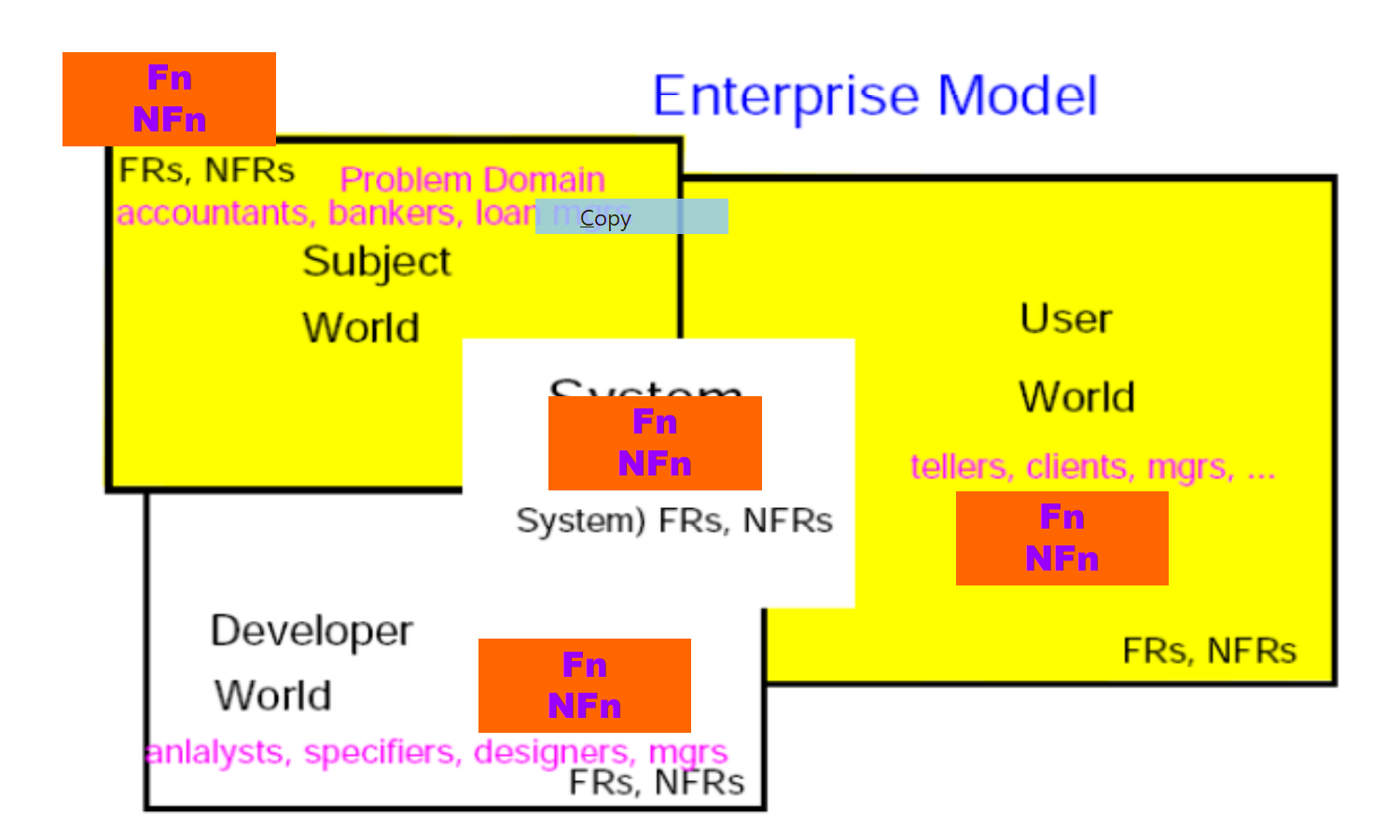


Four Worlds of RE for Control Systems



The goal-service-constraint model:

* Does the reference model capture all the above?
* Where are goals, services and constraints? [Zave94]
* Which is about S, D |= R?
* Which is about technical feasibility, component reuse, etc.?
* Where is traceability?



Question Format

10 T/F questions 2 points each

Essay type questions

Draw fundamental process for RE

elicication

specification

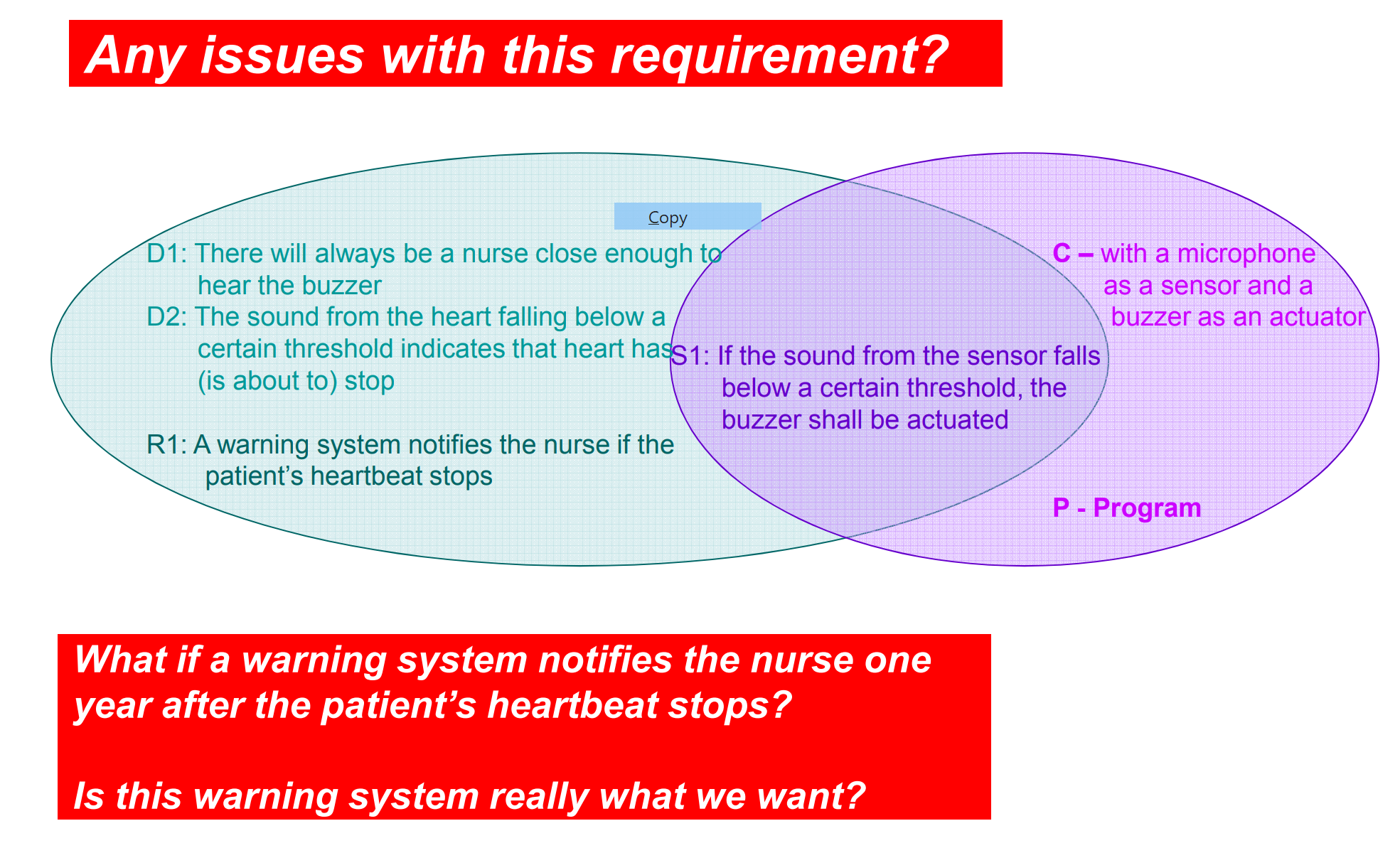
validation

2 key differences between

drawing above

and spiral model

Questions To Think About:



* see examples



Articles

<https://www.techdirt.com/articles/20060818/1613226.shtml>

FBI

* Hired government contractor to build a new system
* Let the contractor define the requirements

SAIC

* a government contractor given a multi-million computer project
* little oversight
* loosely defined objectives

<http://www.crosstalkonline.org/storage/issue-archives/2002/200204/200204-Florence.pdf>

Complete:

Requirements should be as complete as possible

* system objectives
* relationship between the software subsystems.

Traceable:

* connection to underlying source

such as a system-level requirement

* unique identifier

allows other processes to map to it

design, code, and test procedures tracing

Testable:

* All requirements must be testable
* Vague, general statements must be

avoided.

Consistent:

* no requirement should conflict with any other

Feasible:

* If they can not be implemented they should be

eliminated.

Uniquely Identified:

* essential for traceability

Design Free:

* describe "what" the system will do
* later, the detail level will handle "how"