SOFTWARE DEVELOPMENT PHASES

Dr. MONIKA GUPTA
RESEARCHER

IBM RESEARCH INDIA

REQUIREMENTS

Where do most problems come from?

REQUIREMENT SPECIFICATION !!

WHAT IS REQUIREMENTS SPECIFICATION?

1. PROCESS

2. PRODUCT

REQUIREMENT SPECIFICATION (PROCESS)

1

Create high level descriptions

2

Distinguish between ‘right’ and ‘wrong’ system

3

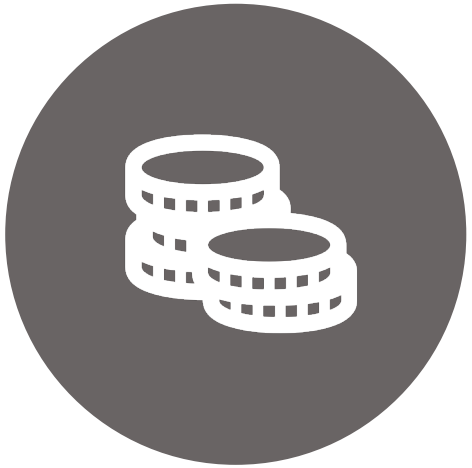
Capture ‘WHAT’ not the ‘HOW’ of solution

REQUIREMENT SPECIFICATION (PRODUCT)

- IEEE is an international organization that has a computer society that has developed a template SRS

WHY ARE REQUIREMENT SPECIFICATION IMPORTANT?

1. ENGINEERING ARGUMENT



2. ECONOMIC ARGUMENT

Cost overrun (as a percentage) vs.
Percent effort on Scope and Requirements

Spend 0% on your requirements	200% cost overrun
Spend ~3% on your requirements	100% cost overrun
Spend ~5% on your requirements	50% cost overrun
Spend 10-20% on your requirements	40% cost overrun

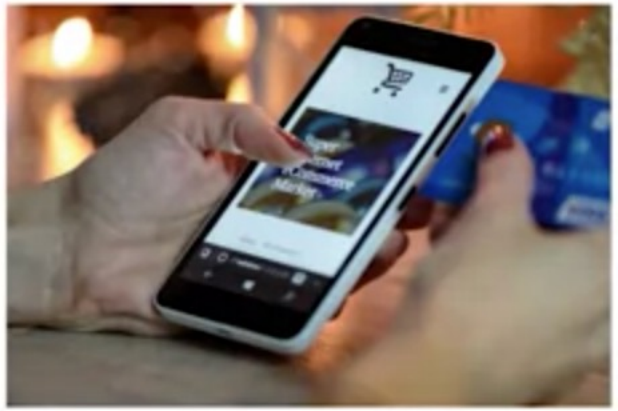
ENGINEERING REASONS

- Spending time up front saves time later

ECONOMIC REASONS

- Most problems come from this stage and the longer they fester in the product, the more costly they are to fix once we do find them.
- In fact, when the problems aren't found until the product has been deployed, you can see repair costs as high as a thousand times the cost if the problem would have been found during the requirements analysis.
- It's nearly an exponential increase in cost at every stage: design, to coding, to unit and integration testing, to system testing and again a thousand times at deployment.

REQUIREMENT VS SPECIFICATIONS



Users

Developers

An example of user requirement and system specification

User Requirement

1. One person must be able to load the boat on the car rack

System Specification

1. Boat must be lighter than 100 lb. (45.36 kg.)
2. Boat must have handles
3. Car rack must be padded (boat slides onto rack easily)
4. Etc.

REQUIREMENT AND SPECIFICATION ARE CLOSELY RELATED CONCEPTS IN DEFINING THE SOLUTION

- Requirements for the user; Specifications for the developer
- Write the Requirements in the User language
- Write the Specifications in the system language
- Make sure that the Specifications meet the Requirements

- It is not about ‘what’ the system will do but rather how the system will perform the behaviors
- Define system properties and constraints
- Process requirements
- Often more critical than functional requirements

Capturing the ``Right'' thing

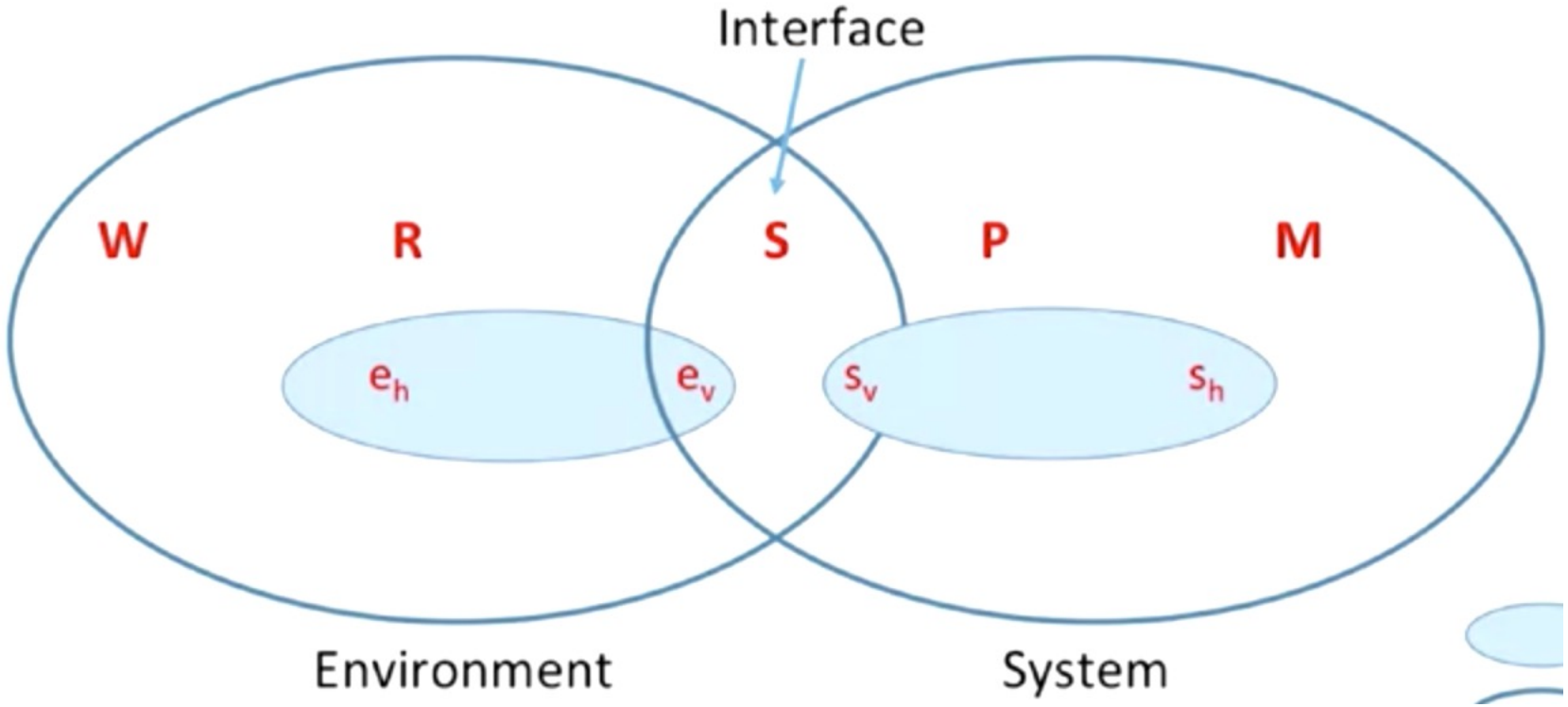
Requirements are always in the problem domain

Specifications are in the solution domain

Several layers of abstraction can exist in between

WRSPM

WRSPM



Capturing the ``Right'' thing

Requirements are always in the problem domain

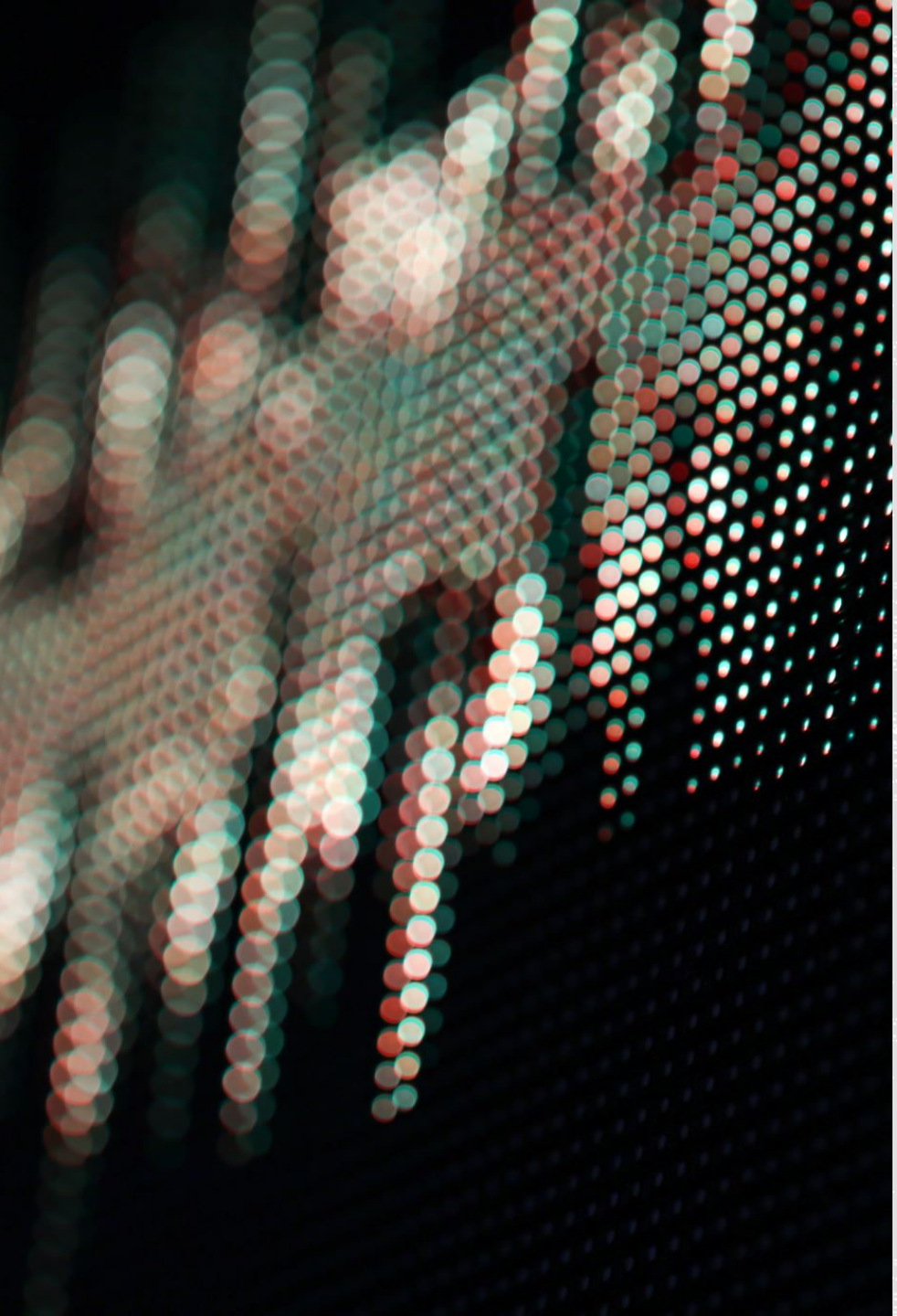
Specifications are in the solution domain

Several layers of abstraction can exist in between

WRSPM – It Is Reference Model For How We Understand Problems In The Real World

WRSPM – REAL WORLD EXAMPLE

- PATIENT MONITOR



PATIENT MONITOR

- System Design

A computer that can be programmed to use a microphone as a sensor and a buzzer as an actuator

- Requirement Specification

If the sound from the sensor falls below a certain threshold, the buzzer shall be actuated

WHAT ABOUT WORLD ASSUMPTIONS?

- If the W, S imply R, and M, P imply S, your solution is "right"