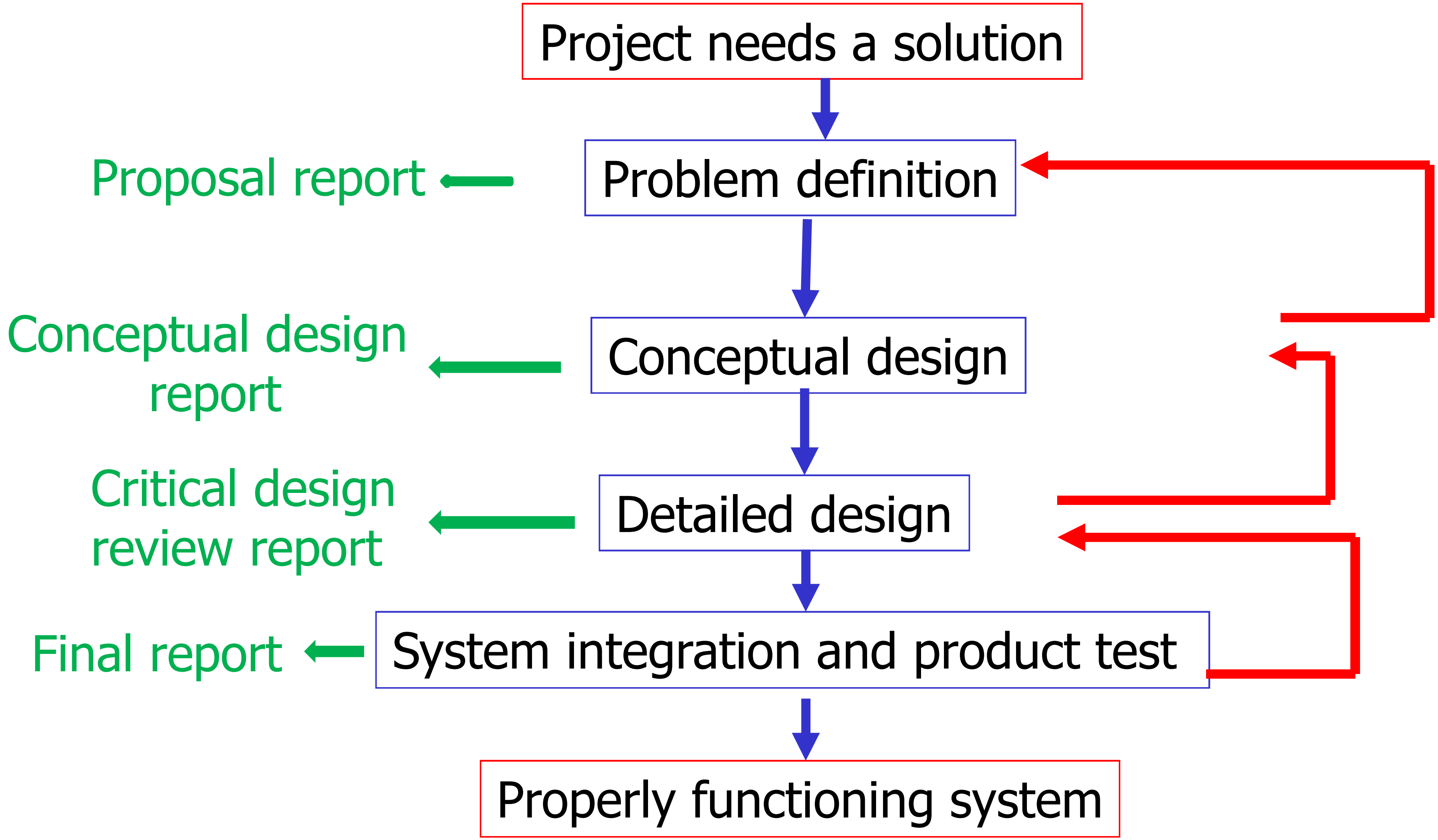
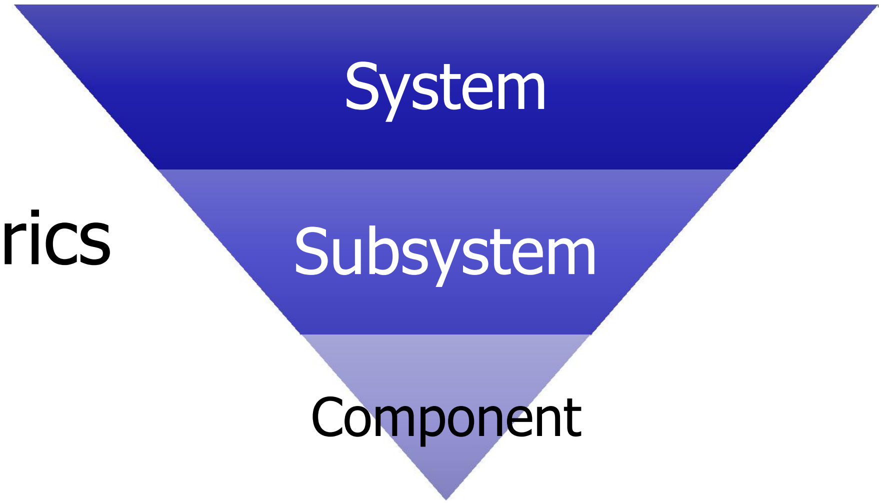
FUNDAMENTALS OF ENGINEERING DESIGN

Design Process



Problem Definition

- Needs assessment
- Define top-level functional requirements
- Define objectives and metrics
- Specify performance requirements
- Identify constraints



–http://www.mrc.uidaho.edu/mrc/people/jff/480/handouts/design_process/

–Lecture notes on 'Understanding & Applying The Engineering Design Process' by Mark D. Conner, The Engineering Academy at Hoover High School

–Ralph M. Ford and Chris S. Coulston, Design for Electrical and Computer Engineers: Theory, concepts and Practice, Mc Graw Hill, 2005.

Needs Assessment

- The aim is not to solve the problem but to understand what the problem is
  - What does this client want?
  - What is the problem that the design is to solve?

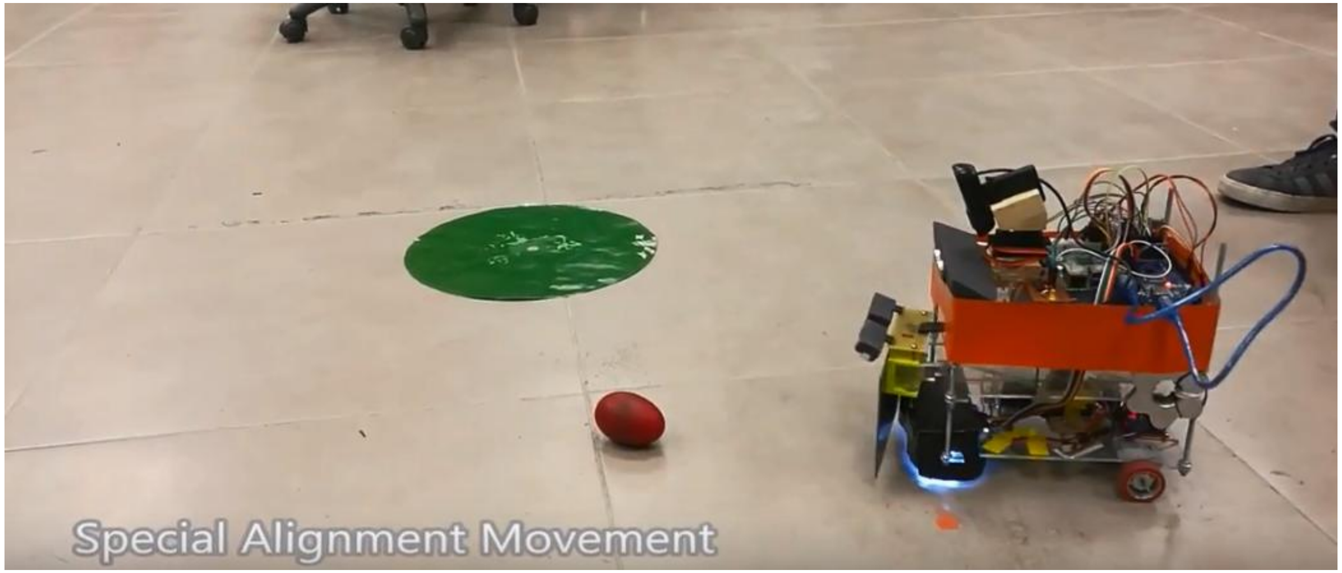
Functional Requirements

Specifies a behaviour that a system or subsystem must perform.

- expressed as “doing” statements
- typically involve output based on input

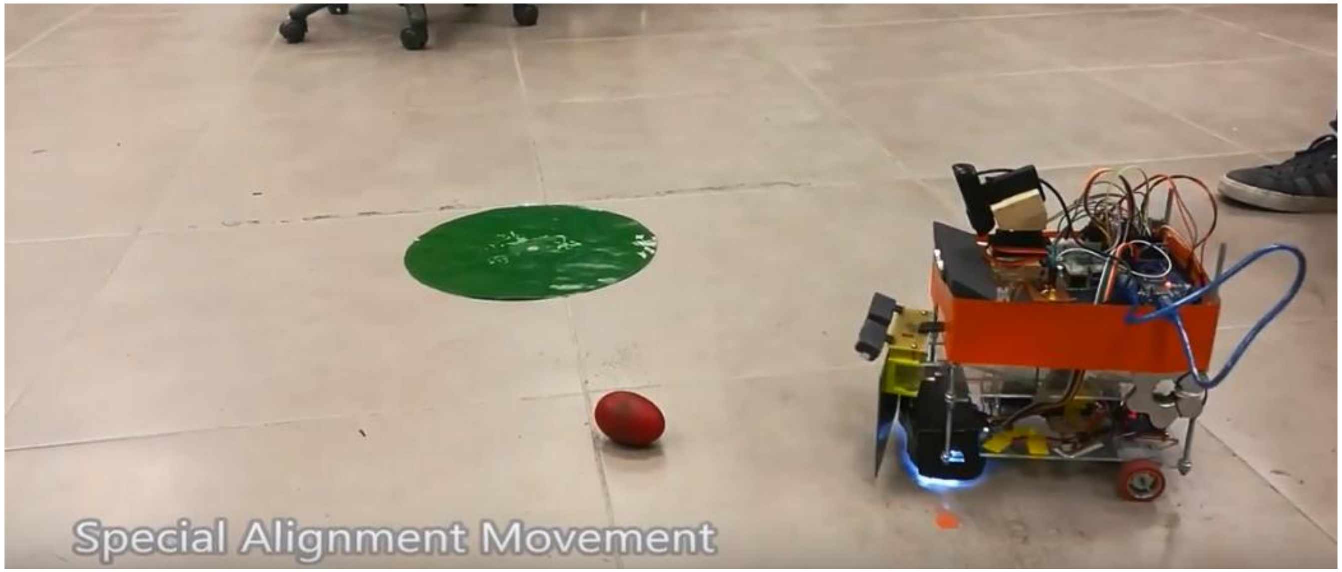
Example

- Design and construct a robot which can compete with a similar robot in pushing egg shaped “balls” along a playfield and place them in ”nests” assigned for them, before the opponent.



Example

- Detect start signal
- Detect the egg
- Detect the nest
- Align the robot, the egg and the nest
- Push the egg towards the nest by controlling it
- Place the egg into the nest



Define objectives

- Objectives, are the desired attributes of the design, what the design will "be" and what qualities it will have
- They are often adjectives/adverbs (e.g., fast, low cost)

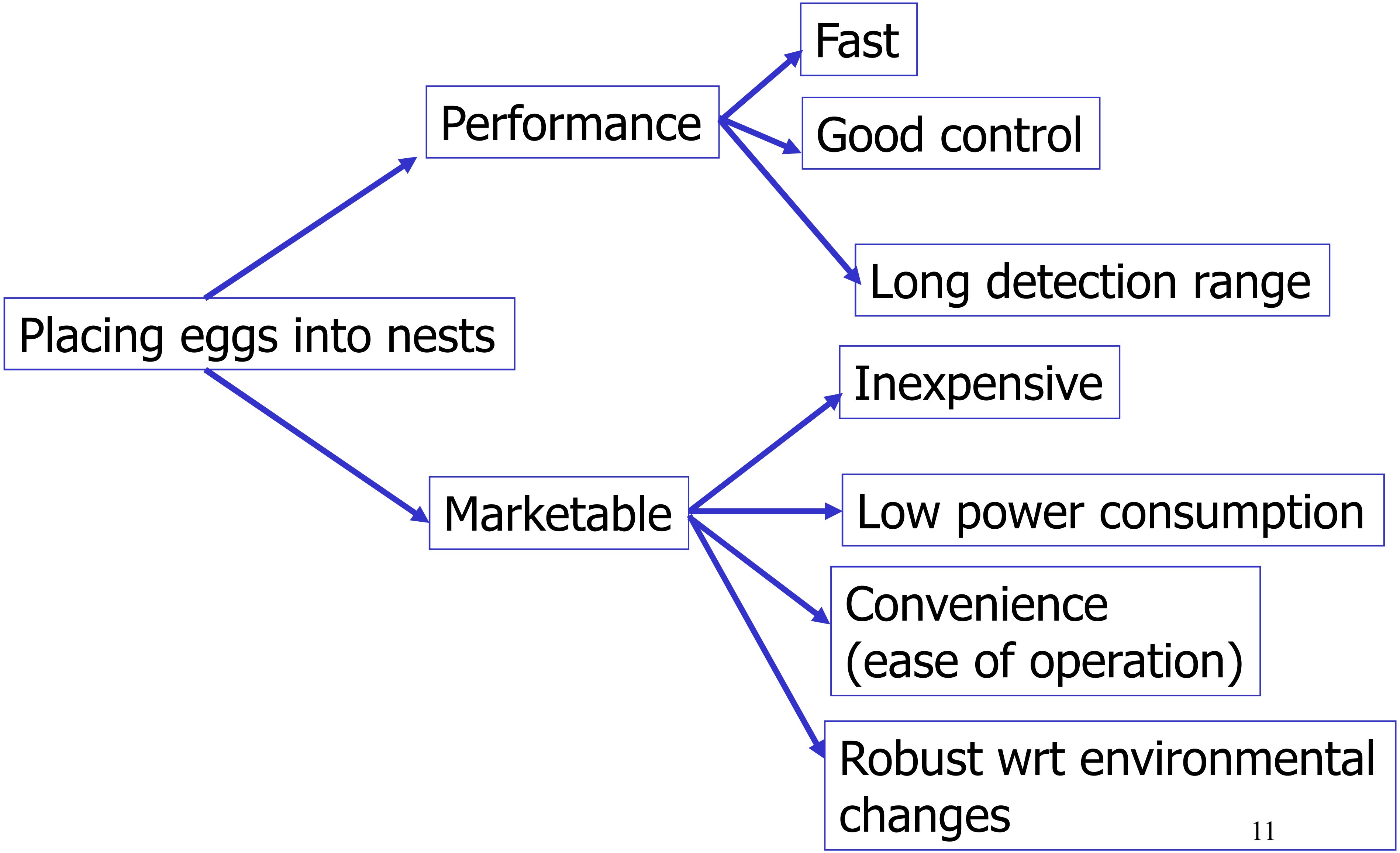
Objective examples

- Performance related
  - Speed
  - Accuracy
  - Resolution
- Cost
- Ease of use
- Reliability, durability
- Power
  - Voltage levels
  - Battery life

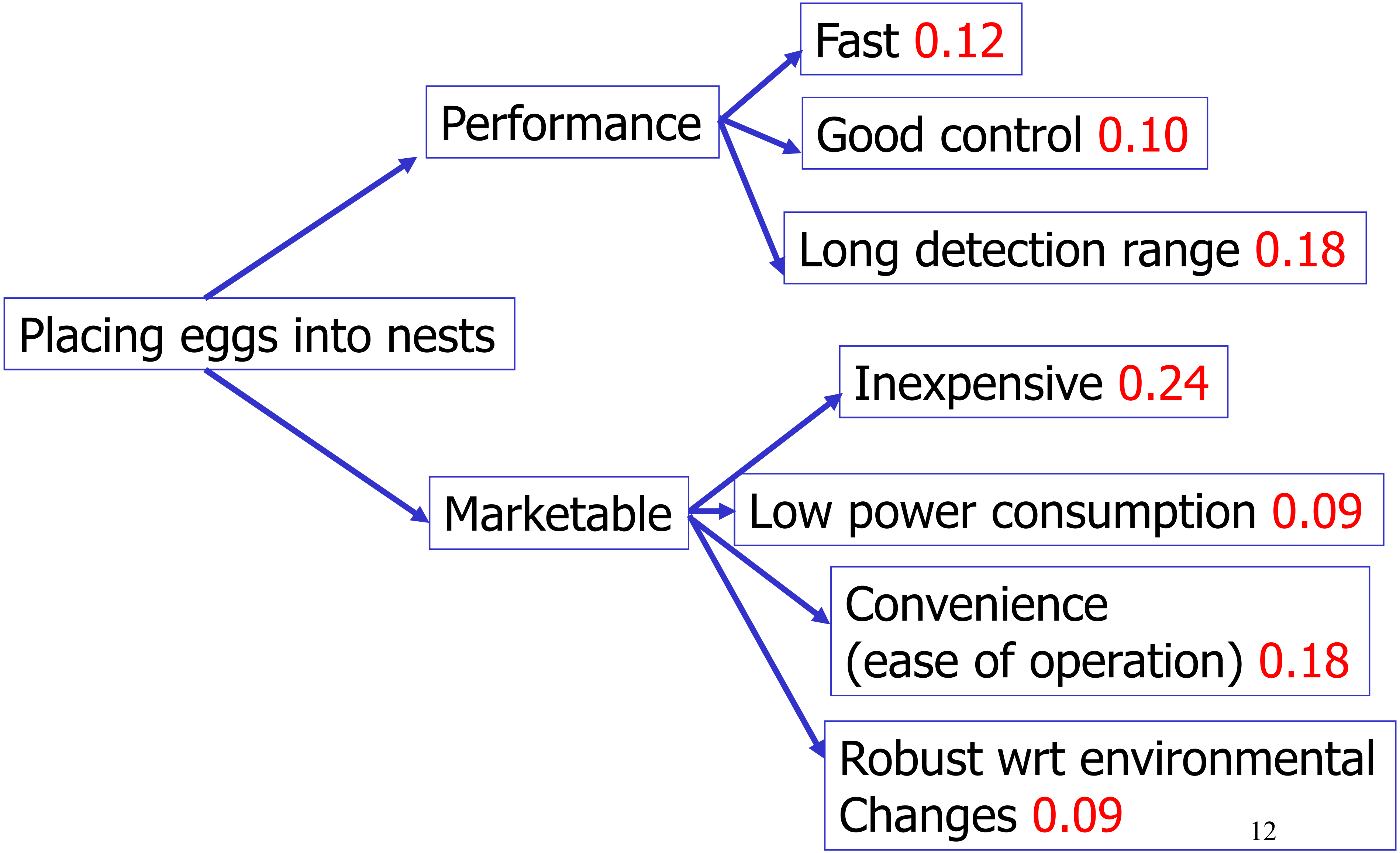
Objective trees

- Make a list of objectives
- Group the relevant objectives
- Form a hierarchical tree structure

Objective trees



Weighted Objective trees



Why do we need objectives?

- Objectives allow exploration of the design space to choose among alternative design configurations
- Three design alternatives
  - Design 1: D1
  - Design 2: D2
  - Design 3: D3
- Which one is the best choice according to my objectives?

Evaluation of design alternatives

	F 0.12	GC 0.10	LDR 0.18	I 0.24	LPC 0.09	C 0.18	R 0.09	Total
D1	8 0.96	6 0.6	10 1.8	4 0.96	2 0.18	0 0	2 0.18	4.86
D2	0 0.0	6 0.6	8 1.44	10 2.4	8 0.72	2 0.36	2 0.18	5.7
D3	2 0.24	8 0.8	0 0.0	2 0.48	6 0.54	10 1.8	4 0.36	4.22

10: Excellent,8: Good, 6: Satis., 4: Av., 2: Unacceptable, 0: Failure

Define objective metrics: Metrics measure how well the objectives are met

Objective Metrics

	Fast	Long Detection range	Robustness to changes in light conditions
10 Excellent	<5 min.	1-2m	Works in the dark and under sunlight
8 Good	5-10	80-100cm	Works in the dark and in the laboratory lighting
6 Satisfactory	10-15	60-80cm	Works under sunlight and in the laboratory lighting
4 Average	15-20 min	40-60cm	Works everywhere in the laboratory
2 Unacceptable	20-30 min.	20-40cm	Works only at some specific locations in the laboratory
0 Failure	>30 min.	0-20cm	Sometimes works at some specific locations in the laboratory

From objectives to requirements

	F 0.12	GC 0.10	LDR 0.18	I 0.24	LPC 0.09	C 0.18	R 0.09	Total
D1	8 0.96	6 0.6	10 1.8	4 0.96	2 0.18	0 0	2 0.18	4.86
D2	0 0.0	6 0.6	8 1.44	10 2.4	8 0.72	2 0.36	2 0.18	~~5.7~~
D3	2 0.24	8 0.8	0 0.0	2 0.48	6 0.54	10 1.8	4 0.36	4.22

- What happens if you don't accept a design alternative lasting longer than 30 minutes?
- Operation time <30 min becomes your performance requirement

Specify performance requirements

- A requirement specifies a capability or a condition to be satisfied.
  - Expressible as numbers and measures
  - Examples:
    - Capability: Works in the dark and under sunlight
    - Condition: Operation time<30 min.
- Translates needs into terminology that helps us to measure how well we met them
  - It turns the problem statement into a technical, quantified form

Requirement types

- Functional
- Performance : Refers to a requirement that quantitatively defines a system's or part's required capability.
  - Tells us how well the design will perform
- Physical : Specifies the physical characteristics of a system or system part.
  - Weight, size, etc.

A good requirement is:

- Abstract
  - What the system will do, not how it will be implemented
- Unambiguous
- Traceable
  - To the needs and desires of the user
- Verifiable, measurable
  - Are we building the system correctly?
  - Test plan!!!
- Achievable (realistic, feasible)
  - Research, engineering know-how, system modeling

Good requirement examples

- The robot must have an average forward speed of 0.5 feet/sec, a top speed of at least one foot/sec, and the ability to accelerate from standstill to the average speed in under one second
- The robot should place the first egg in the nest within at most 20 min.

Relation between requirements and test plans

- The robot should detect 5kHz sine wave generated by a mobile phone

- What is the test plan?
  - How far will be the mobile phone?
  - What will be the environmental conditions?

- The robot should detect 5kHz sine wave generated by a mobile phone placed 1m from the robot at a signal to noise ratio of 20 dB.

A poor requirement

- The robot must employ IR sensors to sense its external environment and navigate autonomously with a battery life of one hour.

- Better one: The robot must navigate autonomously, with the aid of only landmarks in the specified environment, for a period of at least one hour.

Examples of Poor Requirements

– The computer shall process & display the radar information instantly.
– The ship shall carry enough short range missiles.
– The power supply output shall be 28 volts.
– The aircraft shall use stainless steel rivets.
– The power supply unit shall provide 12 V DC with a load regulation of 1% while the line voltage variation is 220 +/- 20 V AC under all load current regimes and vibration and shock profiles within the temperature range.

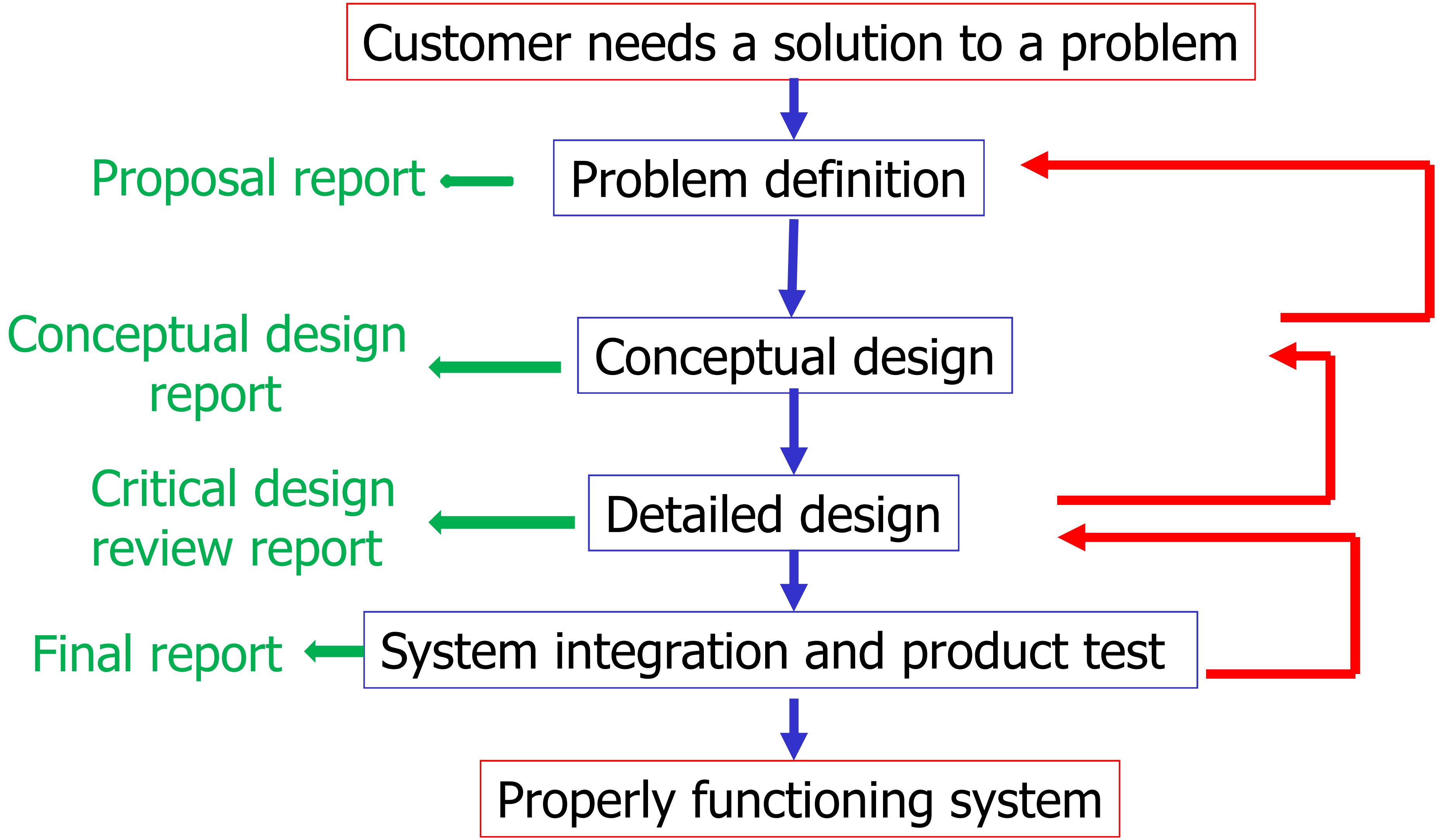
Identify constraints

- Restrictions or limitations on a behavior, a value, or some other aspect of performance
- Stated as clearly defined limits
- Often the result of guidelines and standards

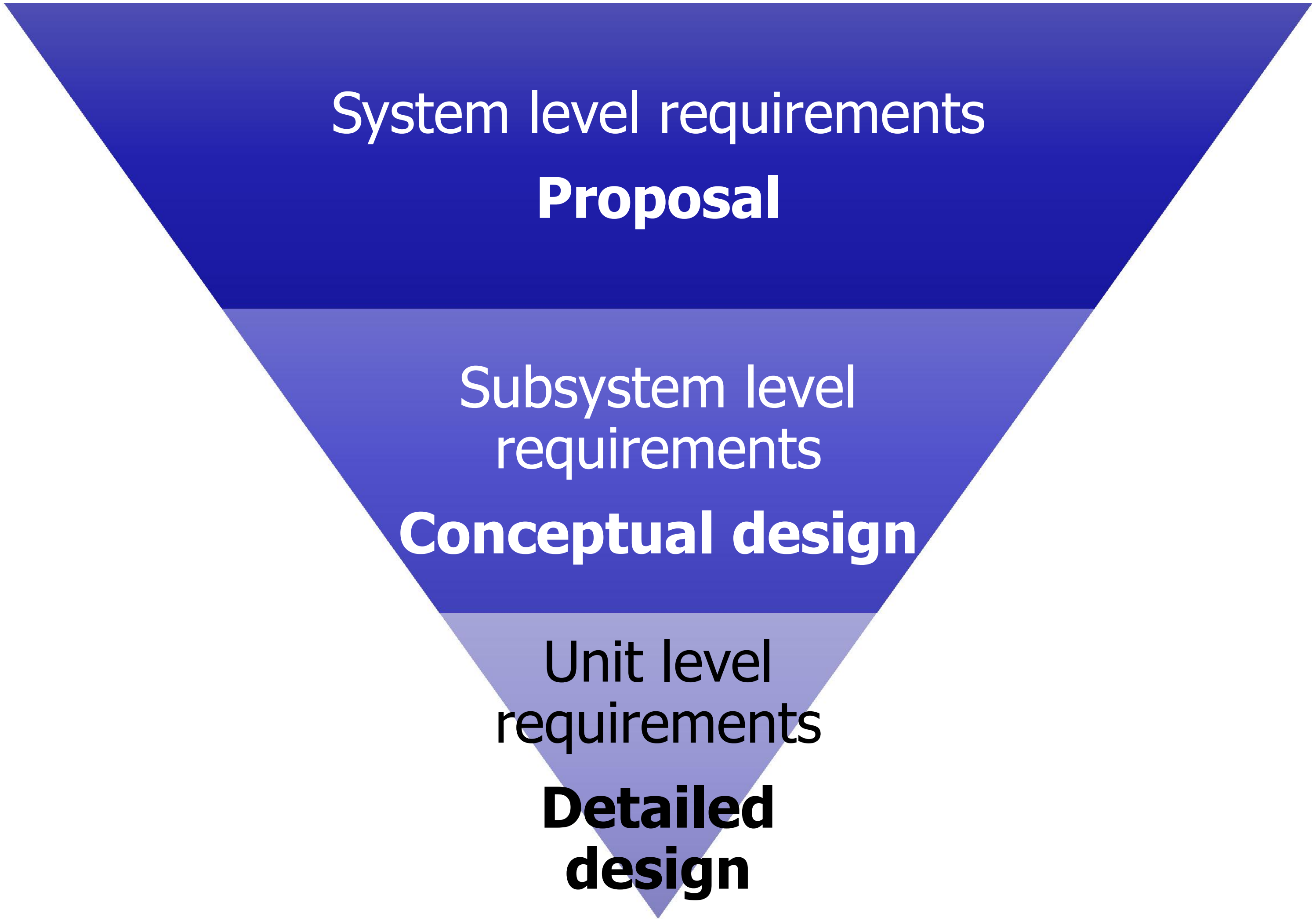
Example constraints of egg placing project

- Size of robot, pushing plate, nest
- Markers to detect robot and nest
- Start signal: 5kHz sine wave

Design Process



System level and subsystem level requirements



System level and subsystem level requirements

- System level requirement

  The robot should place the first egg in the nest within at most 20 min.

- Conceptual Design ⟹ Subsystems are defined

Motion subsystem	Control subsystem	Detection subsystem
• At the start of the game, the robot should move to the egg in 10 sec. • The speed of the robot while pushing the egg should be at least 5cm/sec.	The robot should push the egg without losing control at least 20 cm	• The robot should find the egg within 10 sec after losing control of it. • After detecting the egg and the nest, the robot should align with the egg and the nest within at most 30 sec.

Subsystem level and component level requirements

- Subsystem level requirement

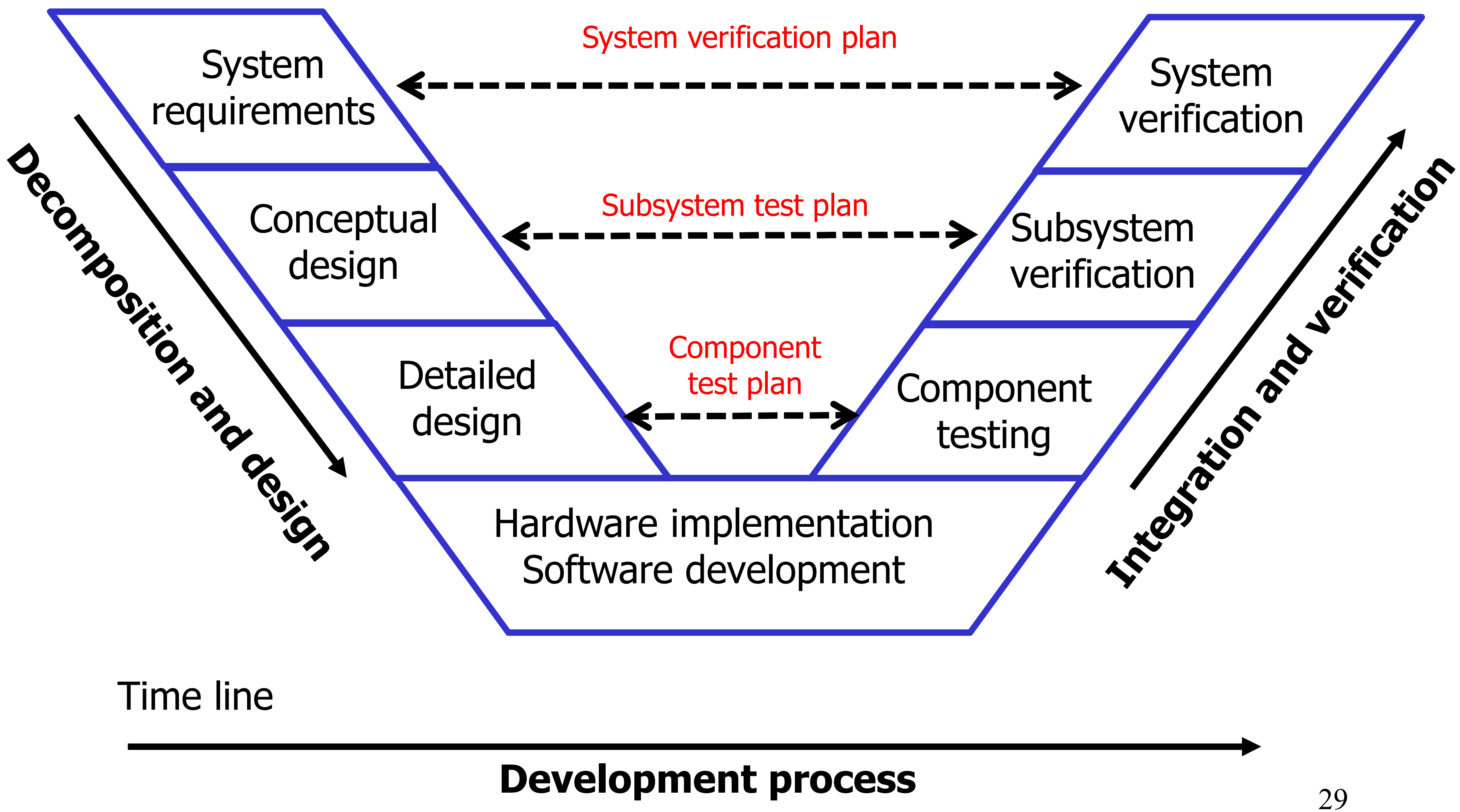
  Detection subsystem

  The robot should find the egg within 10 sec after losing control of it

- Detailed design ⟶ Components are defined

Camera	Microprocessor
The camera should be able to capture 30 frames per second	The microprocessor should be able to process 15 frames per second

V Diagram



System requirements
Conceptual design
Detailed design
Hardware implementation
Software development
Component testing
Subsystem verification
System verification
System verification plan
Subsystem test plan
Component test plan
Decomposition and design
Integration and verification
Time line
Development process