Testing and Practice of CSE

ESC108A Elements of Computer Science and Engineering

B. Tech. 2017

Course Leaders:

Roopa G.

Ami Rai E.

Chaitra S.

Objectives

- At the end of this lecture, student will be able to
  - Explain classification in testing
  - identify the practices of computer science engineers in research and prototyping domain
  - identify the practices of computer science engineers in software development and testing domain

Contents

- Software Testing

- Practice of CSE

Software Testing

- Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test

- Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use

Classification of Testing

- There are three dimensions of classification
  - By means
    - Testing by machine (DYNAMIC testing)
    - Testing by human (STATIC testing)
  - By target
    - Testing outside FUNCTION (black box)
    - Testing inside STRUCTURE (white box)
  - By granularity
    - UNIT level
    - INTEGRATION level
    - SYSTEM level

White Box Testing

- White-box testing also known as clear box testing, glass box testing, transparent box testing, and structural testing

- A method of testing software that tests internal structures or workings of an application

- The tester chooses inputs to exercise paths through the code and determine the appropriate outputs

White Box Testing: Branch Coverage

Branch Coverage (Decision Coverage)

- This measure ensures whether boolean expressions evaluated to both true and false

- *Eg:*

    if (condition1 && (condition2 || function1()))

    statement1;

    else

    statement2;

- Disadvantage: it ignores branches within boolean expressions
    – The control structure completely exercised without a call to *function1()*

Black Box Testing

- Treat code as a black box and verify whether its requirements have been met

- A method of software testing that examines the functionality of an application without peering into its internal structures or workings

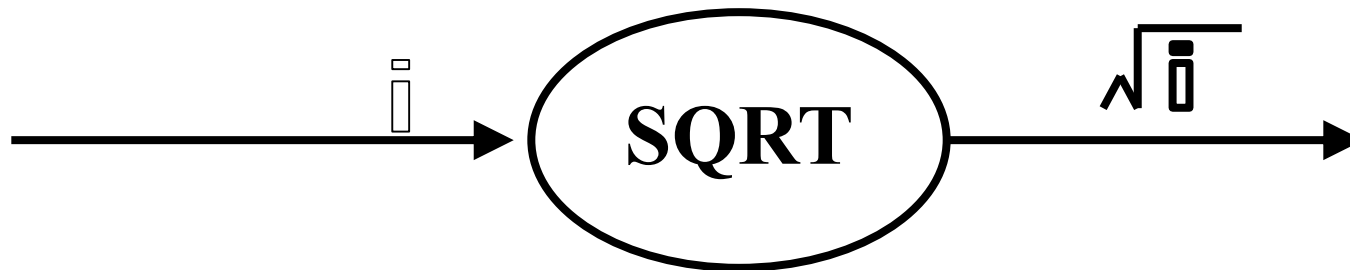# Test Case Design - Black Box Testing

Equivalence Testing

- Partition the input domain of a program in to finite number of equivalence classes such that one can reasonably assume that a test case of a representative value of each class is equivalent to a test of any other value

- **Identifying the Equivalence Classes**

External condition	Valid equivalence classes	Invalid equivalence classes
	Represent valid inputs to the program	Represent all other possible states of the condition (i.e. , erroneous input values)

Test Case Design - Black Box Testing

- Example: A program reads an input value in the range of 1 and 5000
  - computes the square root of the input number

Test Case Design - Black Box Testing

- There are three equivalence classes
  - One valid and two invalid equivalence classes are defined
  - The set of negative integers
  - Set of integers in the range of 1 and 5000
  - Integers larger than 5000
- The test suite must include
  - Representatives from each of the three equivalence classes
  - A possible test suite can be {-5,500,6000}

Invalid | **1** | **valid** | **5000** | **Invalid**

Levels of Testing

- Unit testing
  - tests that verify the functionality of a specific section of code, usually at the function level
  - these types of tests are usually written by developers as they work on code

- Integration testing
  - any type of software testing that seeks to verify the interfaces between components against a software design

- System testing
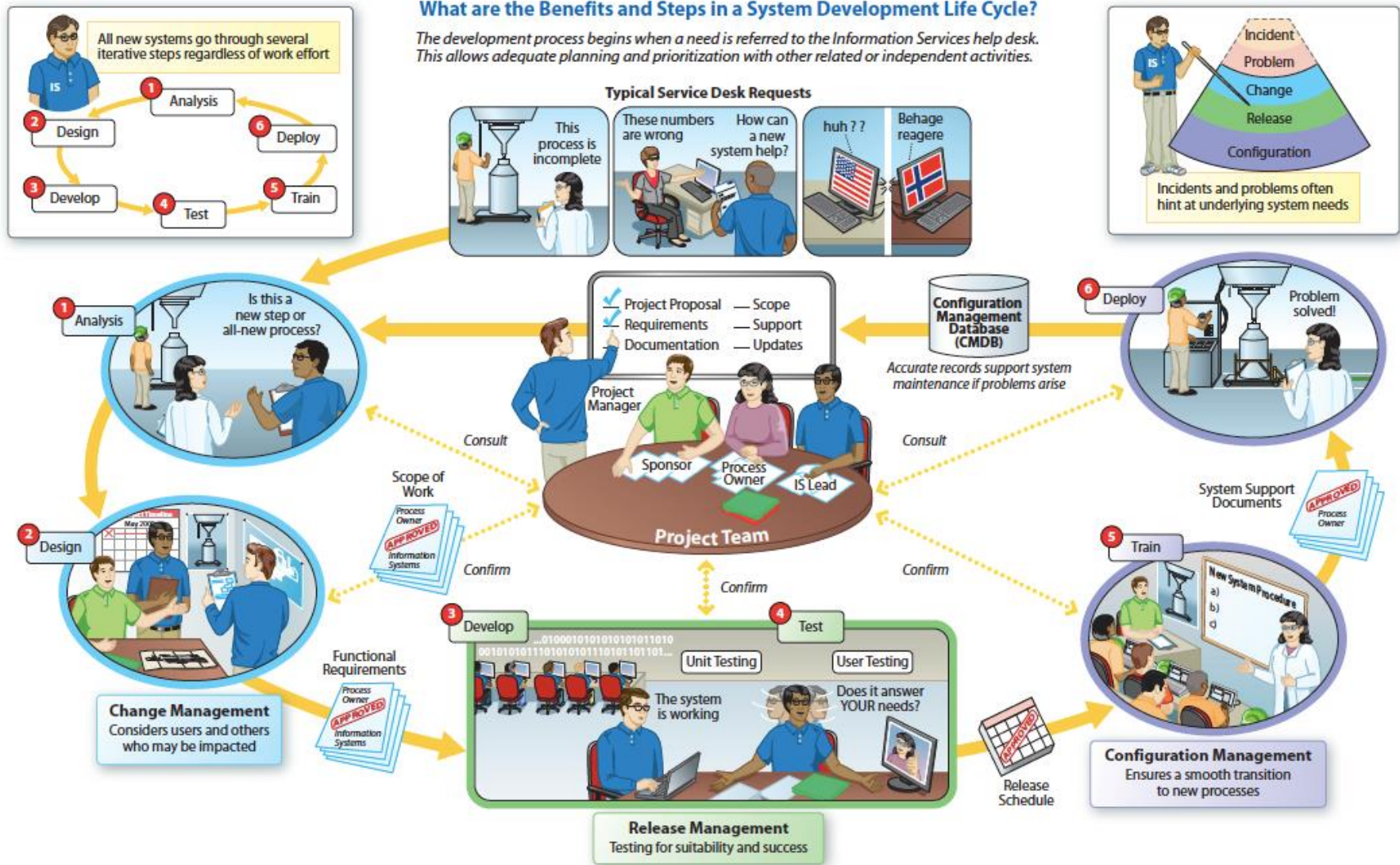 - tests a completely integrated system to verify that the system meets its requirements

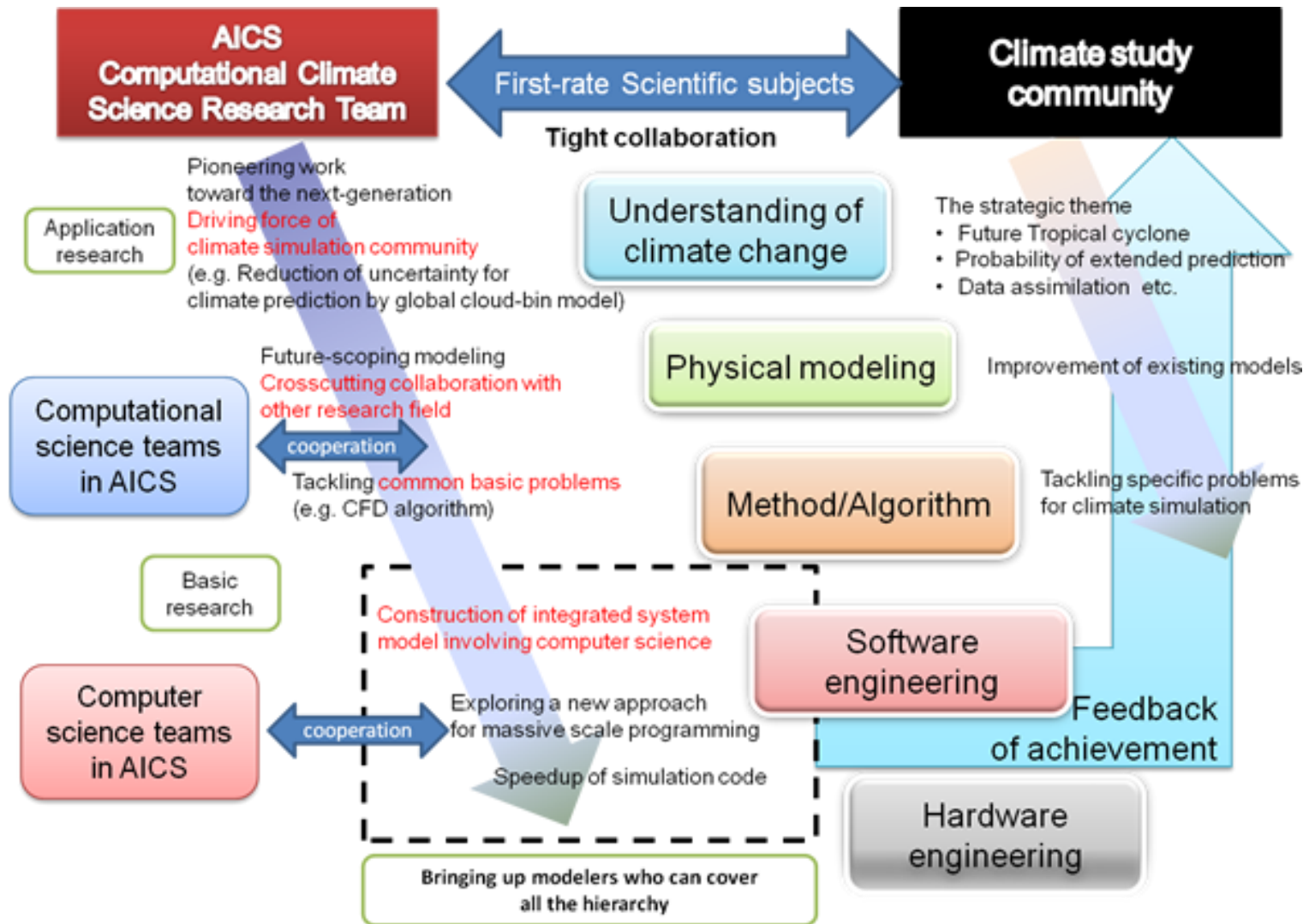# Practice of CSE

Development and Testing
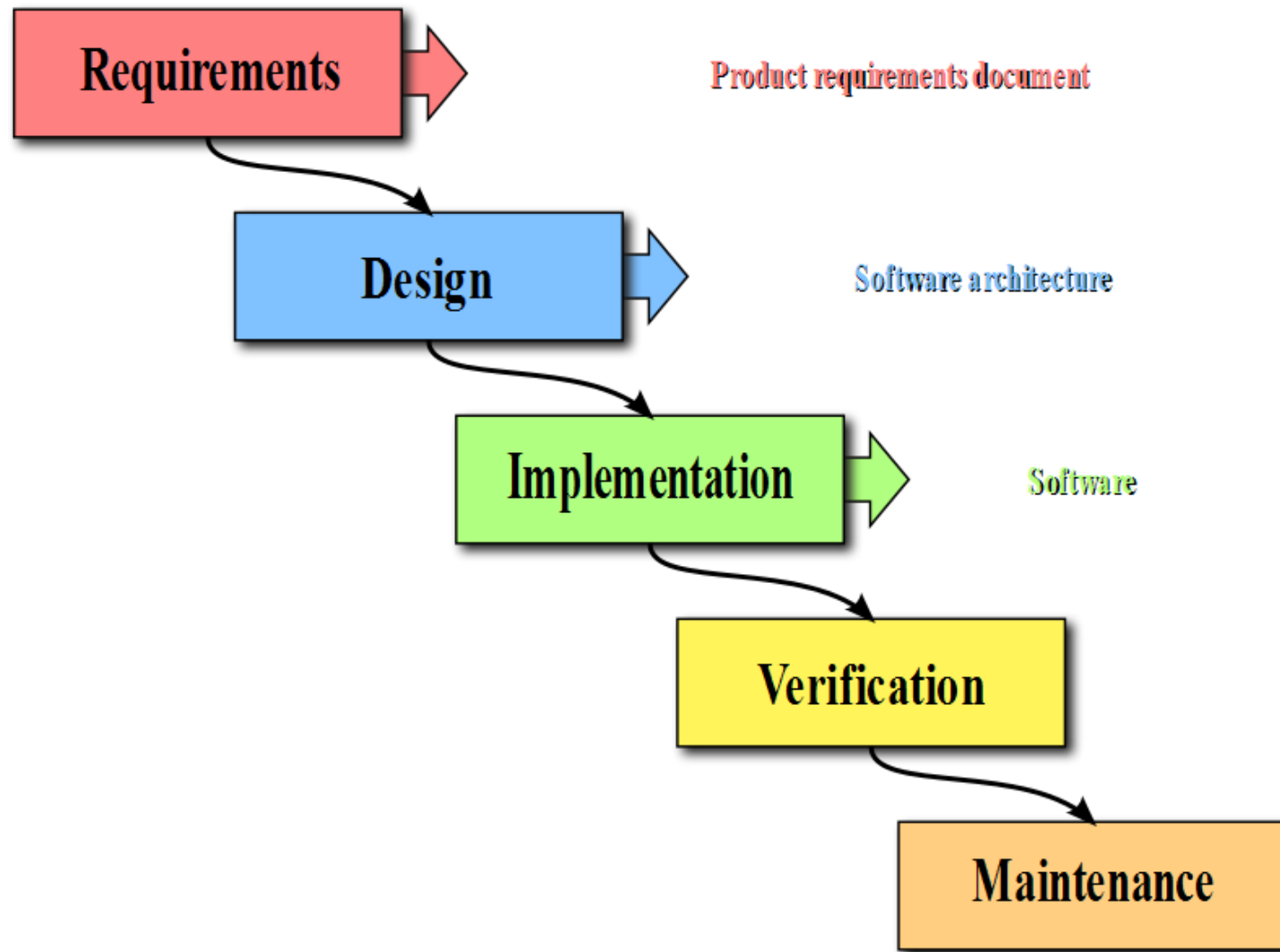
Complete Practice of CSE

Research and Development

Example: Climate Modeling and Simulation
Source : http://aics-research.riken.jp/img/tomita_e.png

Summary

- Testing is the process of executing a program or system with the intent of finding errors

- Computer Science and engineering is used in two types of practices: Namely: Research and Development and Computerized Solutions

- Computerized Solutions are produced by following software engineering principles, supported with production and maintenance practices

- Research and Development practices demand domain specific integration of computer engineering practices