05506017
Software Engineering

Chapter 4 :
Tasks of Software Development

ผศ.ดร.วรางคณา กิ้มปาน

ภาควิชาวิทยาการคอมพิวเตอร์ สจล.

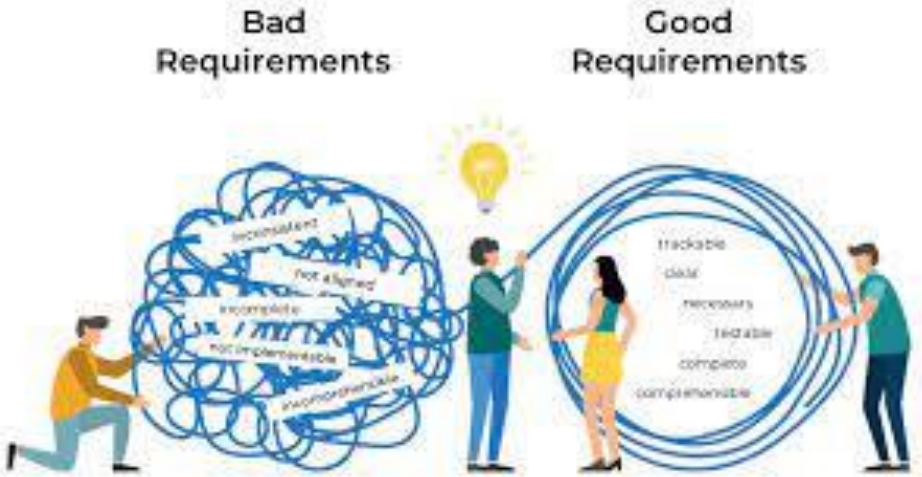
The Activities

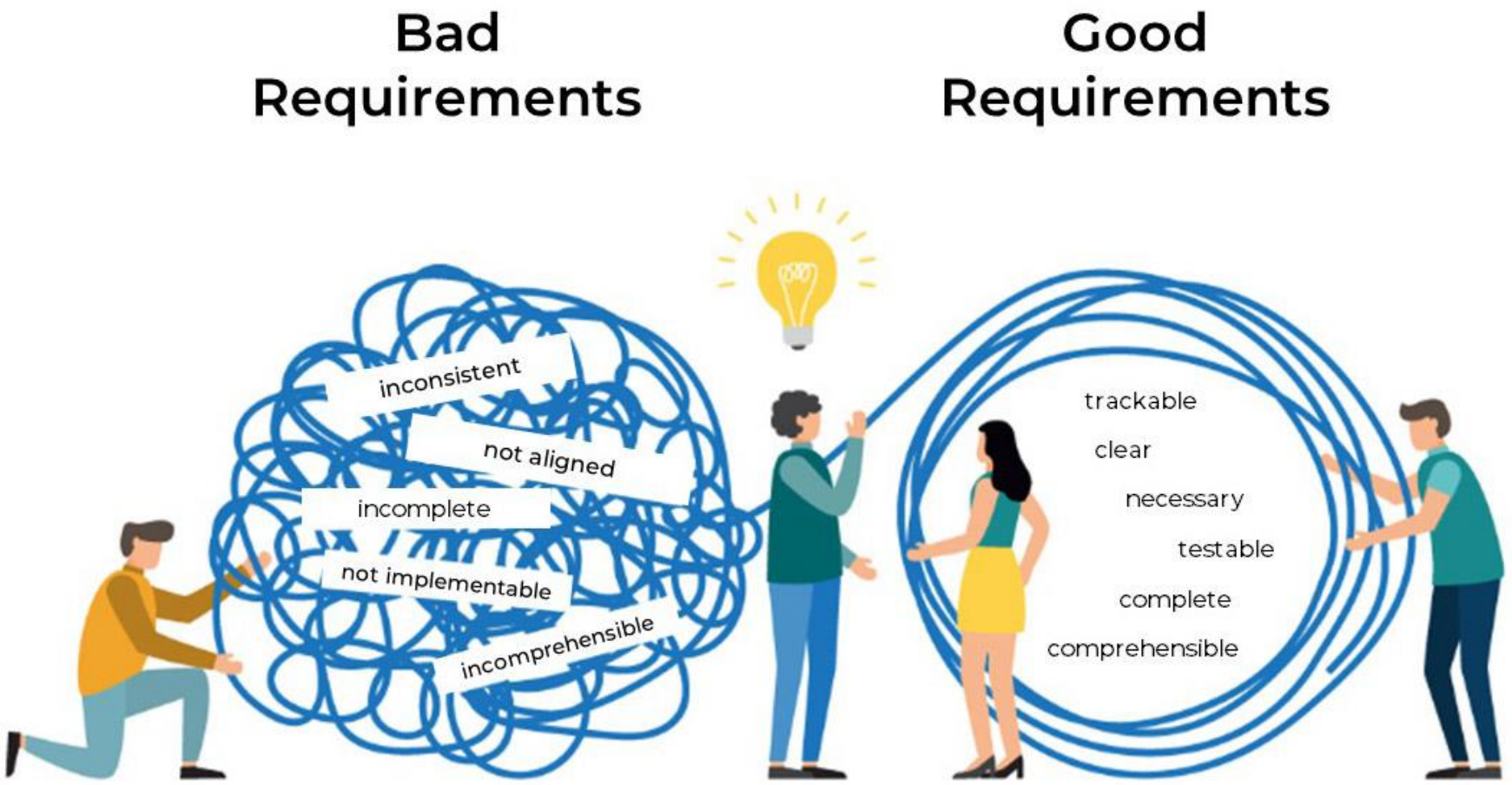
The Tasks - Feasibility study

- A feasibility study establishes whether or not the project is to proceed. It may be that the system is unnecessary, too expensive or too risky.
- One approach to a feasibility study is to perform cost-benefit analysis.
- The cost of the proposed system is estimated, which may involve new hardware as well as software, and compared with the cost of likely savings.

The Tasks - Requirements engineering (specification)

- At the start of a project, the developer finds out what the user (client or customer) wants the software to do and records the requirements as clearly as possible.
- The product of this stage is a requirements specification.



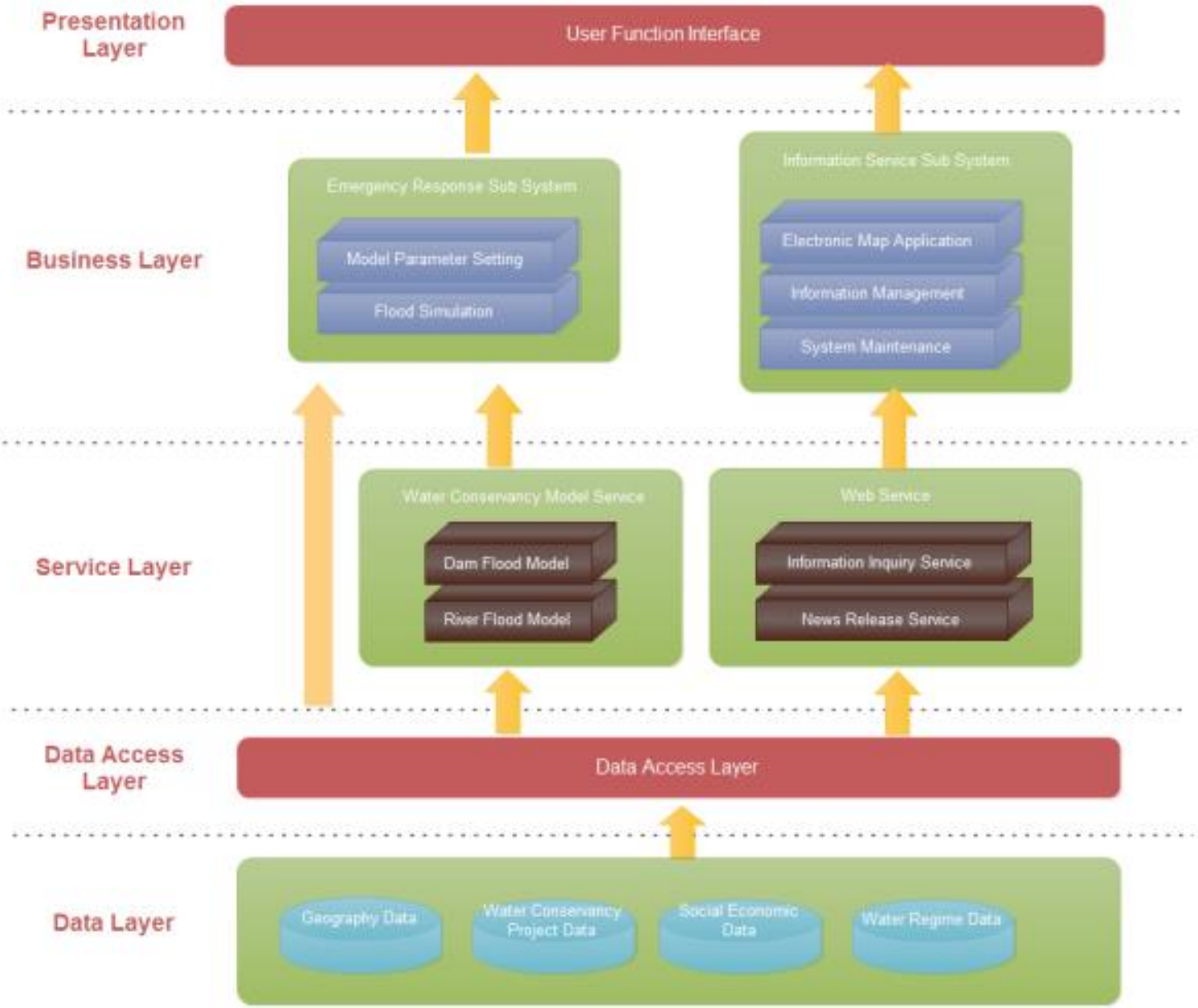

Bad Requirements
Good Requirements
inconsistent
not aligned
incomplete
not implementable
incomprehensible
trackable
clear
necessary
testable
complete
comprehensible

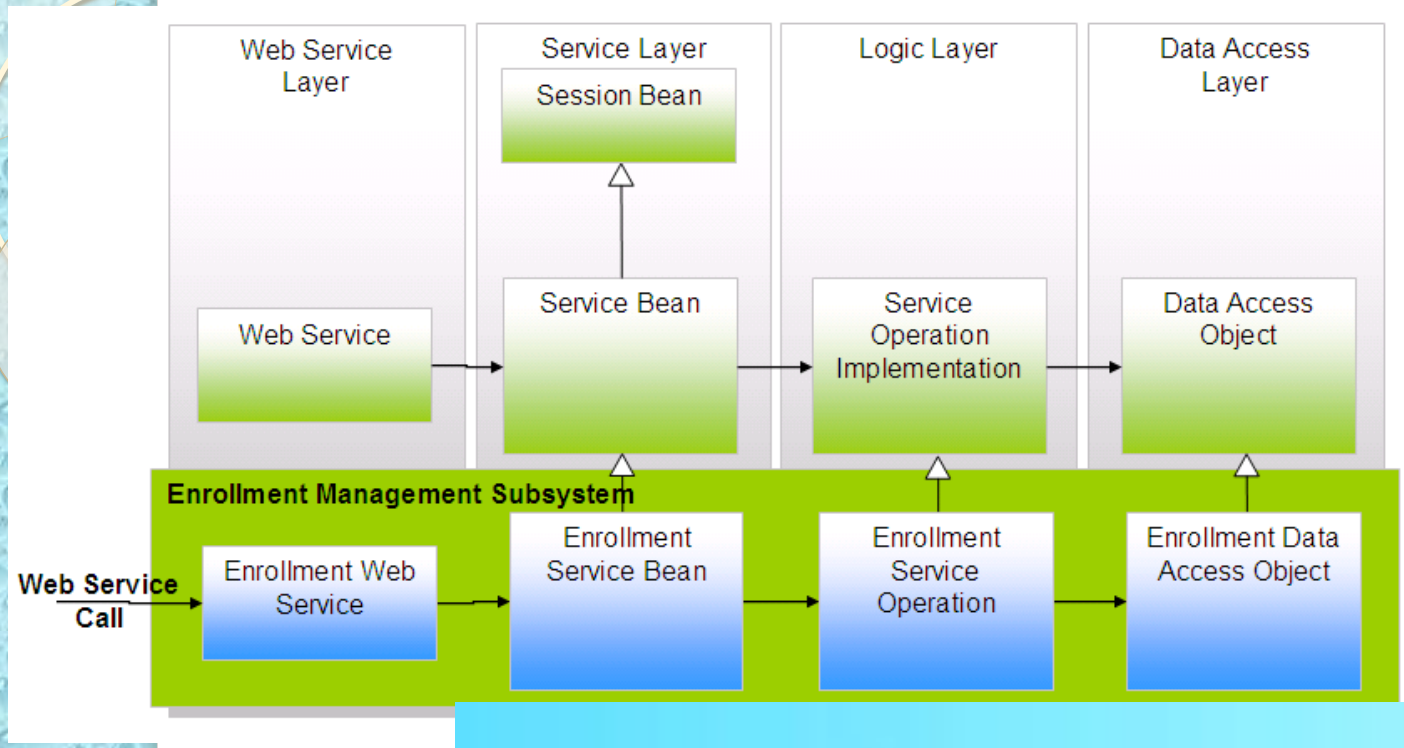
The Tasks – User interface design

- Most software has a graphical user interface, which must be carefully designed so that it is easy to use

The Tasks - Architectural (large-scale) design

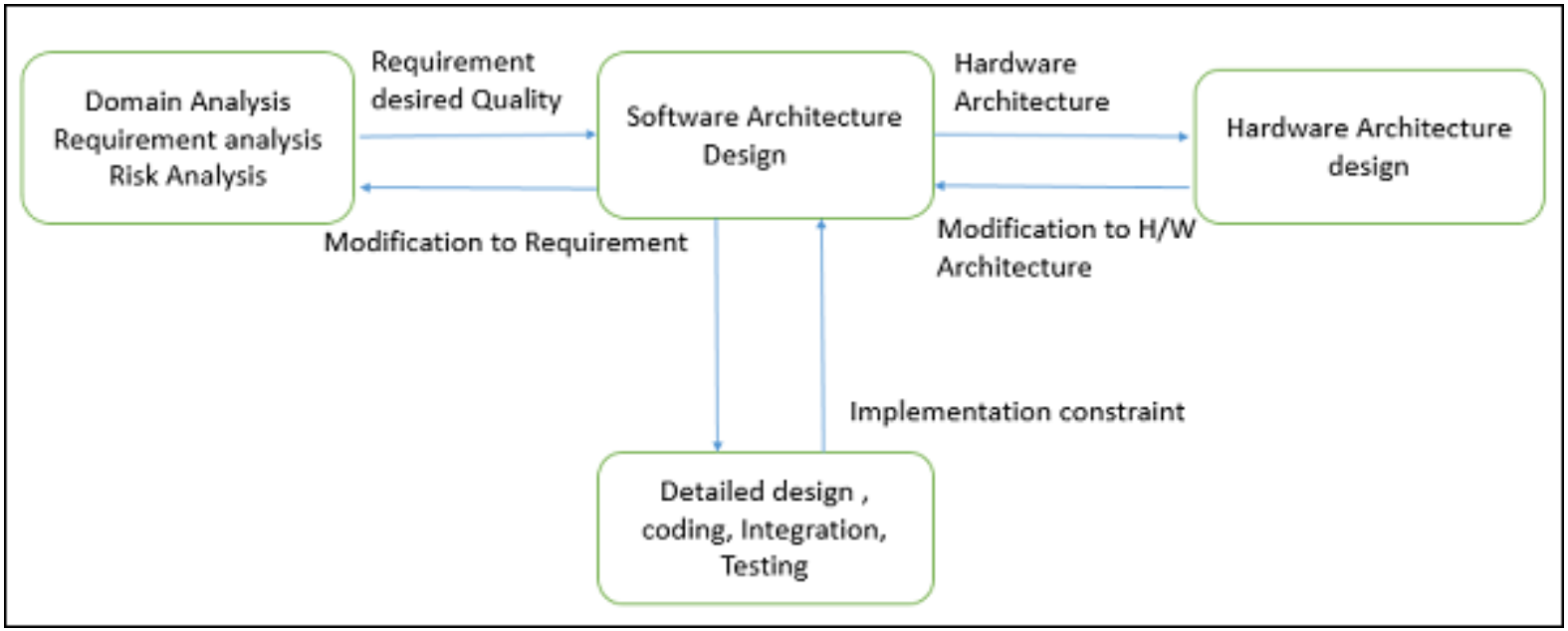
- A software system may be large and complex. It is sometimes too large to be written as one single program.
- The software must be constructed from modules or components.
- Architectural, or large-scale design breaks the overall system down into a number of simpler modules.
- The products of this activity are an architectural design and module specifications.


Presentation Layer
User Function Interface
Business Layer
Emergency Response Sub System
Model Parameter Setting
Flood Simulation
Information Service Sub System
Electronic Map Application
Information Management
System Maintenance
Service Layer
Water Conservancy Model Service
Dam Flood Model
River Flood Model
Web Service
Information Inquiry Service
News Release Service
Data Access Layer
Data Access Layer
Data Layer
Geography Data
Water Conservancy Project Data
Social Economic Data
Water Regime Data


Web Service Layer
Service Layer
Session Bean
Logic Layer
Data Access Layer
Web Service
Service Bean
Service Operation Implementation
Data Access Object
Enrollment Management Subsystem
Web Service Call
Enrollment Web Service
Enrollment Service Bean
Enrollment Service Operation
Enrollment Data Access Object

The Tasks – Detailed design

- The design of each module or component is carried out.
- The products are detailed designs of each module.

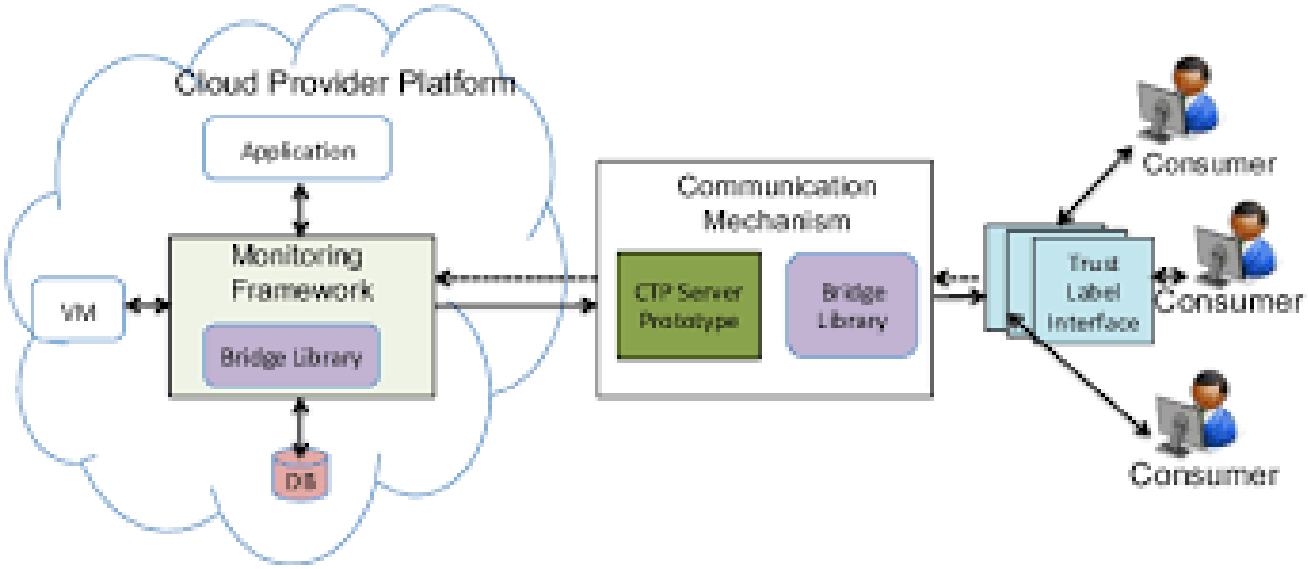


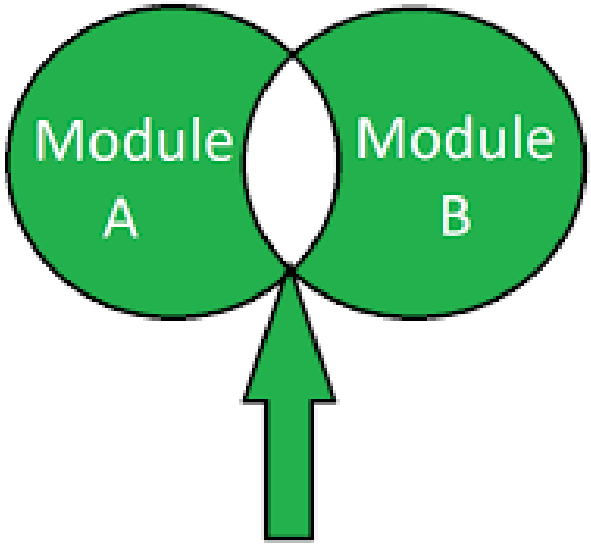
The Tasks – Programming (coding)

- The detailed designs are converted into instructions written in the programming language.
- There may be a choice of programming languages, from which one must be selected.
- The product is the code.

The Tasks - System integration

- The individual components of the software are combined together, which is sometimes called the build.
- The product is the complete system.





System Integration Testing

The Tasks - Verification

- This seeks to ensure that the software is reliable.
- According to Barry Boehm (one of the all-time greats of software engineering), verification answers the question:

  Are we building the product right? A piece of software that meets its specification is of limited use if it crashes frequently.
- Verification is concerned with the developer's view – the internal implementation of the system.

The Tasks – Verification (2)

- Two types of verification are *unit testing* and *system testing*.
- In unit testing, each module of the software is tested in isolation.

The inputs to unit testing are:

- 1. the unit specification
- 2. the unit code
- 3. a list of expected test results.

The Tasks – Verification (3)

- The products of unit testing are the test results.
- Unit testing verifies that the behavior of the coding conforms to its unit specification.
- In system testing or integration testing, the modules are linked together and the complete system tested.
- The inputs to system testing are the system specification and the code for the complete system.
- The outcome of system testing is the completed, tested software, verifying that the system meets its specification.

The Tasks – Validation

- This seeks to ensure that the software meets its users' needs.
- According to Boehm, validation answers the question:

  Are we building the right product?

- Validation is to do with the client's view of the system, the external view of the system. It is no use creating a piece of software that works perfectly (that is tested to perfection) if it doesn't do what its users want.

The Tasks – Validation (2)

- An important example of a validation activity is *acceptance testing*. This happens at the end of the project when the software is deemed complete, is demonstrated to its client and accepted by them as satisfactory.
- The inputs to acceptance testing are the client and the apparently complete software. The products are either a sign-off document and an accepted system or a list of faults.
- The outcome is that the system complies with the requirements of the client or it does not.

My Definition of Validation

VERIFICATION

- 2 sleeves?
- Is it size L?
- Is it blue?
- Are any buttons missing?

VALIDATION

- Does it fit?
- Is it comfortable to drive in?
- Does the colour match my eyes?
- Can I afford it?
- Is it good quality?
- Will my date like it?

The Tasks – Production

- The system is put into use. (This is sometimes, confusingly, termed implementation.)
- The users may need training.

Maintenance

- When the software is in use, sooner or later it will almost certainly need fixing or enhancing.
- Making these changes constitutes maintenance.
- Software maintenance often goes on for years after the software is first constructed.
- The product of this activity is the modified software.

The Tasks – Documentation

- Documentation is required for two types of people – users and the developers.
- Users need information about how to install the software, how to re-install the software and how to use it. Even in the computer age, paper manuals are still welcome.
- For general purpose software, such as a word processor, a help system is often provided.
- User documentation concentrates on the "what" (the external view) of the software, not the "how" (the internal workings).
- Developers need documentation in order to continue development and to carry out maintenance.
- This typically comprises the specification, the architectural design, the detailed design, the code, annotation within the code (termed comments), test schedules, test results and the project plan.

The Tasks – Project management

- Someone needs to create and maintain plans, resolve problems, allocate work to people and check that it has been completed.