05506017
Software Engineering

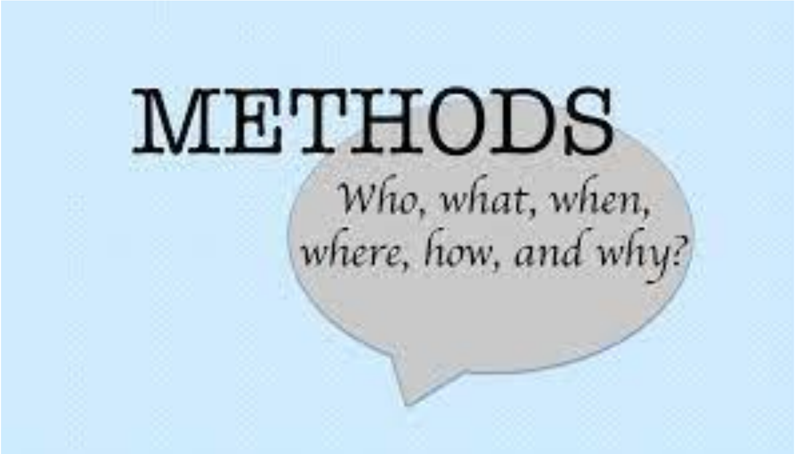
Chapter 2 :
Software Engineering Principles

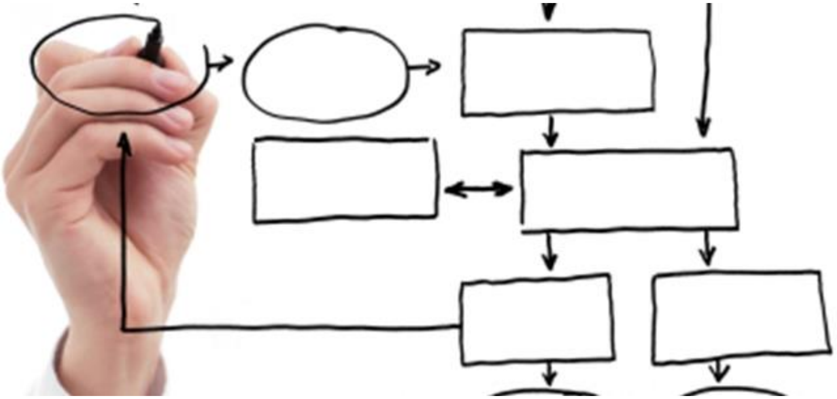
ผศ.ดร.วรางคณา กิ้มปาน

ภาควิชาวิทยาการคอมพิวเตอร์ สจล.

Software Principles

- It is important to know how to make a success to software development.
- SW principles deal with SE process and final product.
- SW engineers should be equipped with appropriate methods and specific techniques that will make processes and products.


METHODS
Who, what, when, where, how, and why?



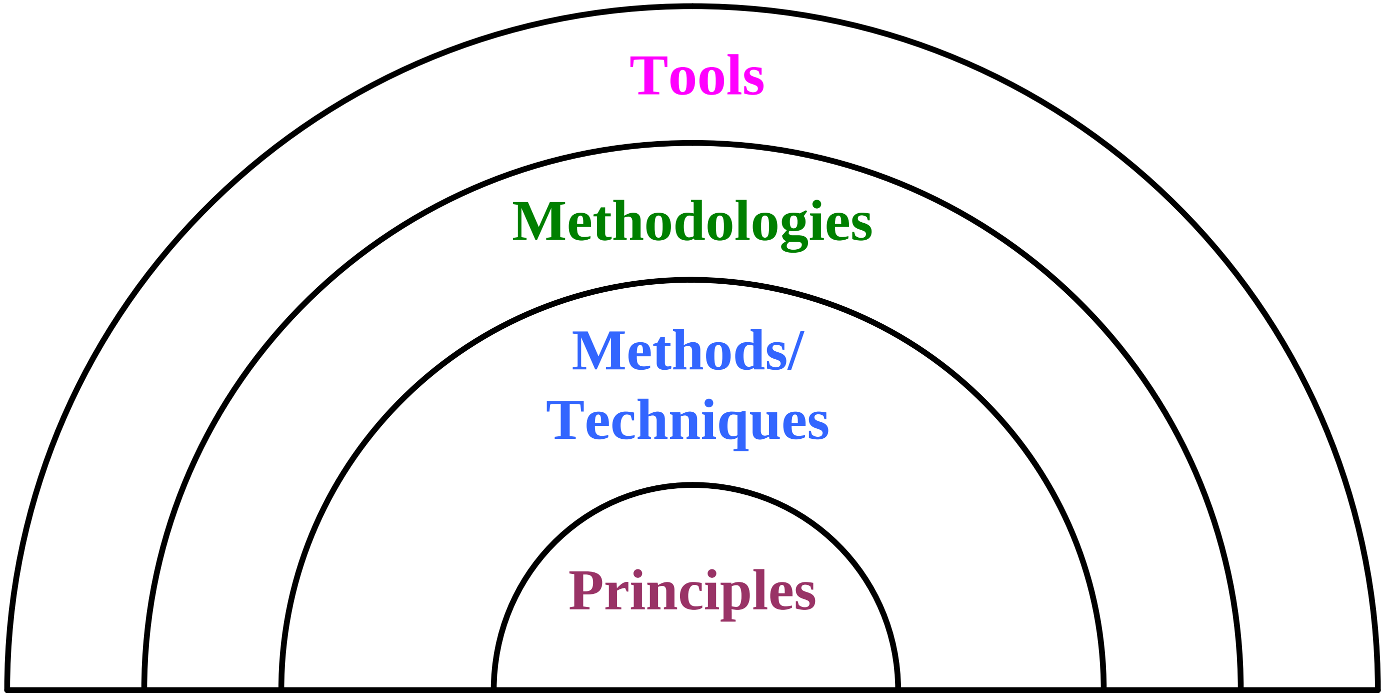
TECHNIQUE
shutterstock.com · 1831072243

Method? Technique?

- Method is general guidelines for some activity.
- Technique is more technical and mechanical.
- Methods and techniques are formed as methodology.

Tool is developed to support the application of techniques, methods, and methodologies.

Relationship among 4 layers

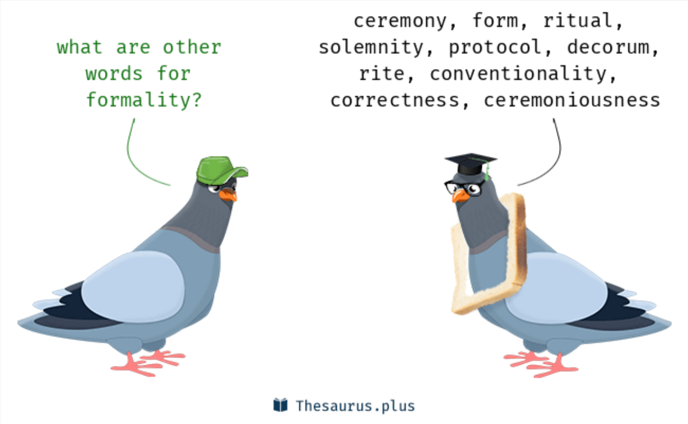


** Principles are the basis of all methods/techniques, methodologies and tools. **

Important Principles

- To discuss about 7 important principles that apply throughout the SW development process as;

  1. Rigor and Formality
  2. Separation of concerns
  3. Modularity
  4. Abstraction
  5. Anticipation of change
  6. Generality
  7. Incrementality



what are other words for formality?

ceremony, form, ritual, solemnity, protocol, decorum, rite, conventionality, correctness, ceremoniousness

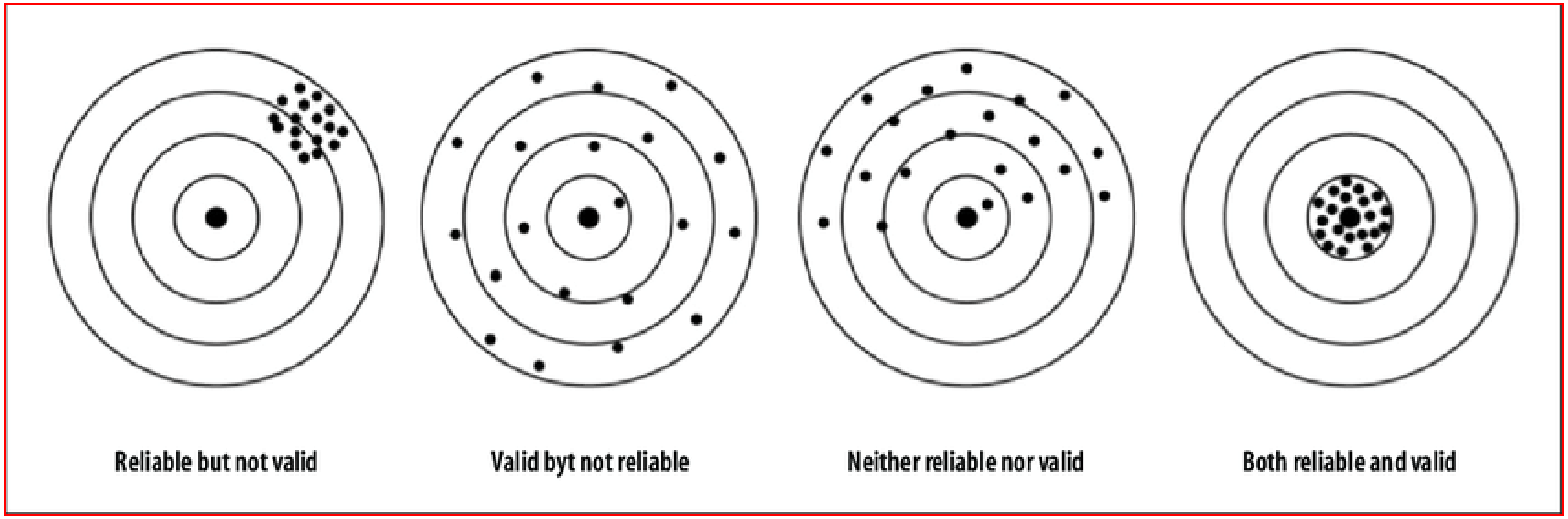
Rigor and Formality

- Within a rigorous approach, we can produce more reliable products, control their cost, and increase the confidence in their reliability.
- It enhances creativity by improving the engineer's confidence in creative results.
- Formality is a stronger requirement than rigor, it requires the SW process to be driven and evaluated by mathematical laws.

Rigor and Formality (2)

- The engineer (mathematician) must be able to understand the level of rigor and formality that should be achieved, depending on the conceptual difficulty of the task and its criticality.
- Rigor and formality are not restricted to programming.
- Influence of rigor and formality will be on the reliability and verifiability of SW product. And have also beneficial effects on maintainability, reusability, portability, understandability, and interoperability.
- Rigorous documentation of the SW process may help maintain an existing product.

Reliability



https://www.researchgate.net/figure/Possible-combinations-of-validity-and-reliability-of-measurement-instruments_fig1_319070974

Verifiability

https://medium.com/fordham-business-law-association/smart-contract-basics-a-legal-contract-perspective-part-ii-verifiability-4bc5c4ea8b88

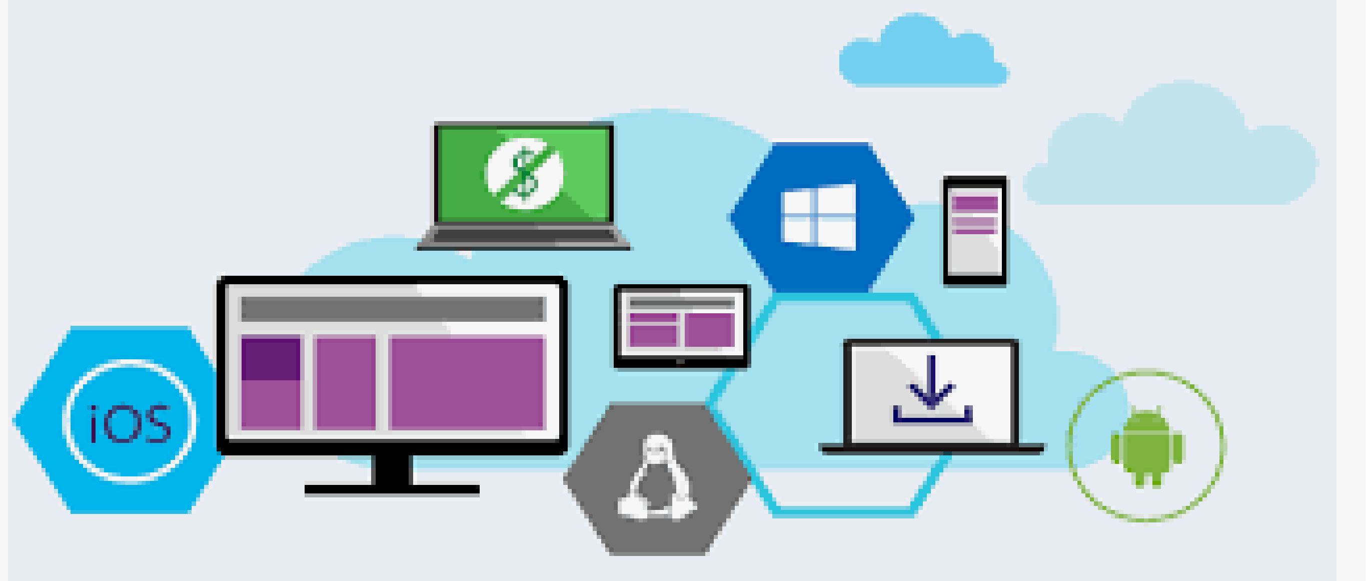
Maintainability

https://www.onupkeep.com/maintenance-glossary/maintainability

Reusability

https://www.innofied.com/7-ways-make-code-reusable/

Portability

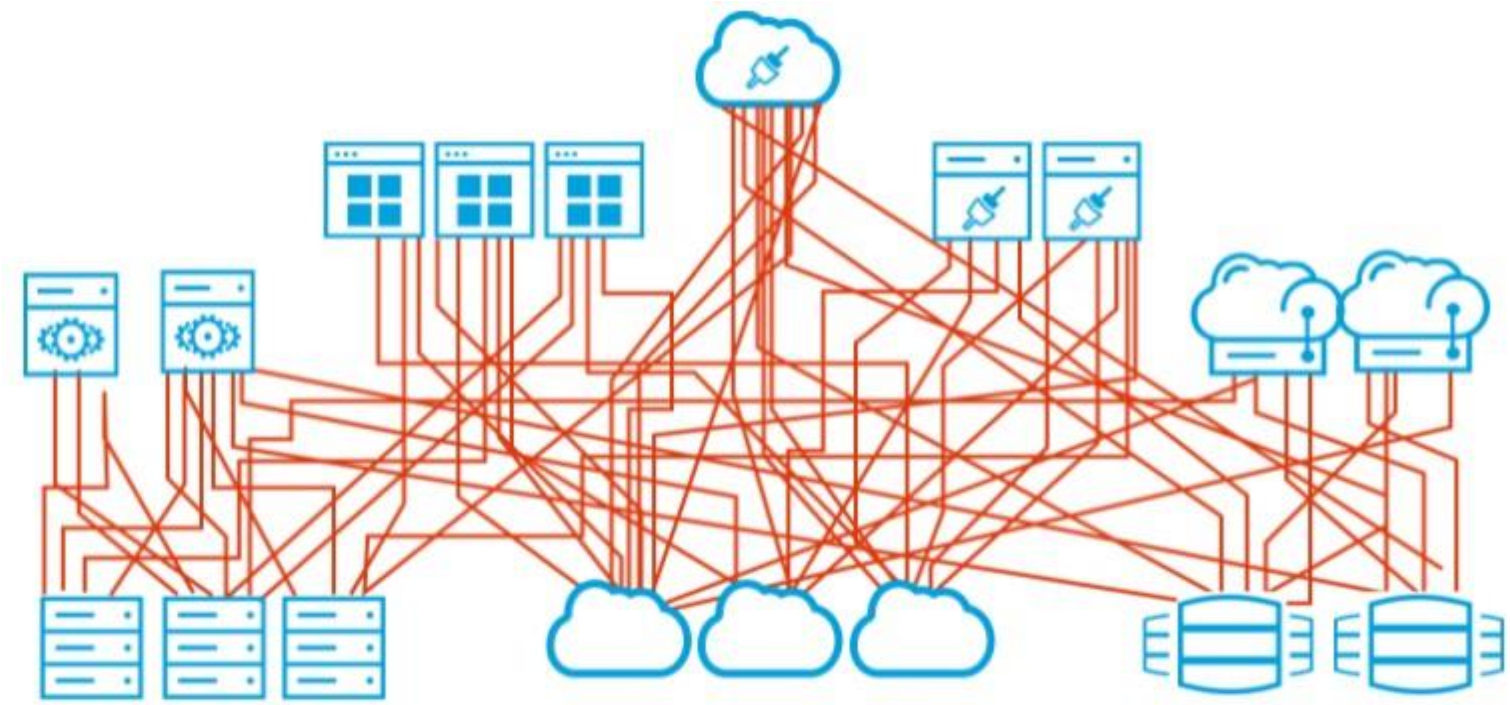


https://www.pinterest.com/pin/202521314478771019/

Understandability

https://dlpng.com/png/6440273

Interoperability



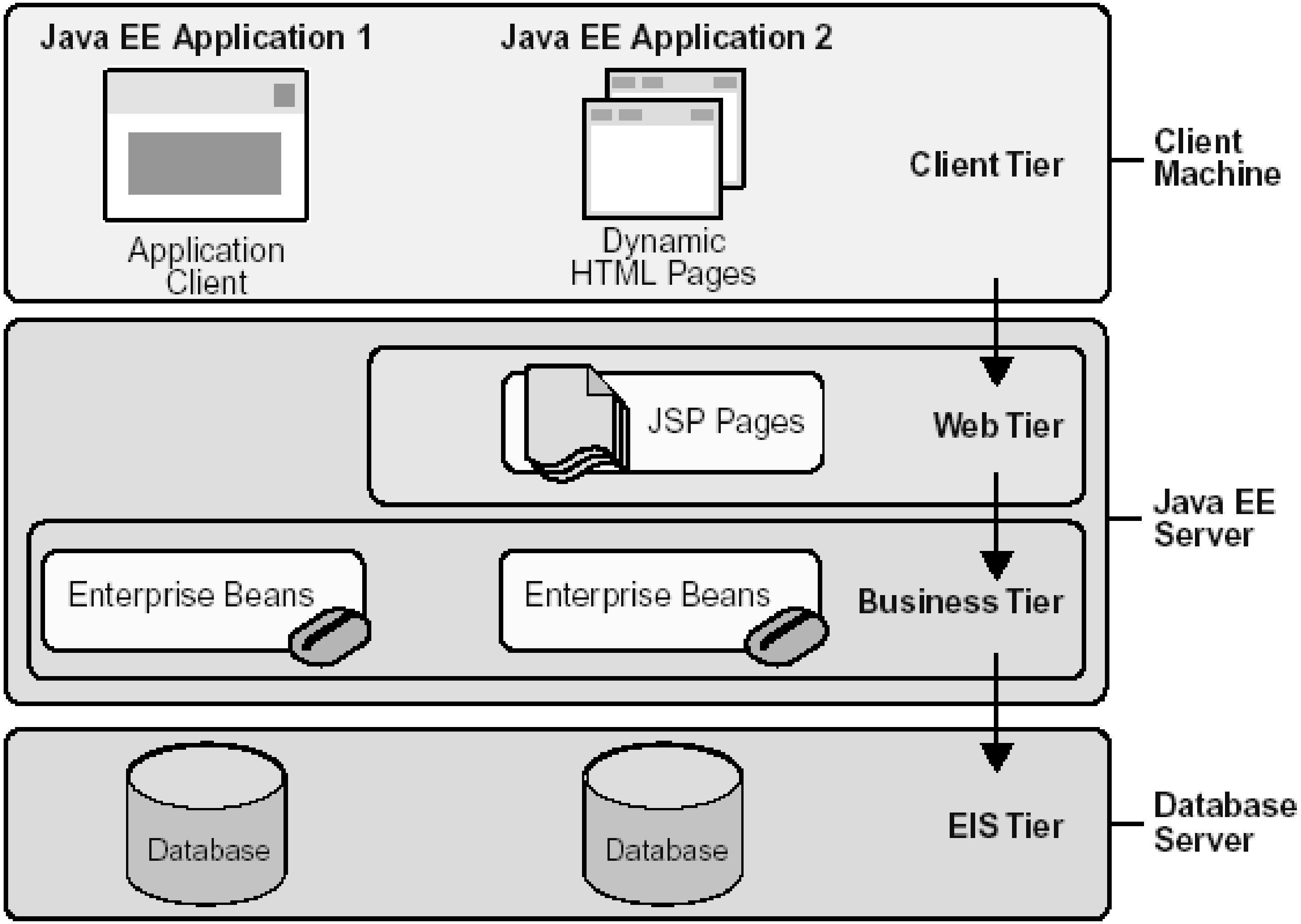
https://www.omnisci.com/technical-glossary/interoperability

Separation of Concerns

- It is a commonsense practice that we try to follow in our everyday life to master the difficulties we encounter.
- Many decisions may concern features of the product ; functions to offer, expected reliability, space and time efficiency, user interface.
- Others concern with; the development process, development environment, team organization and structure, scheduling, control procedures, design strategies, error recovery mechanism, or economic and financial matters.

Separation of Concerns (2)

- There are many ways that concerns may be separated.
  - Time : SW life cycle, the sequence of activities that should be followed in SW production.
  - Qualities : The efficiency and the correctness of a given program.
  - Views : Different views of the SW to be analyzed separately.
  - Parts : In term of size.
- It may result in separation of responsibilities in dealing with separate issues.

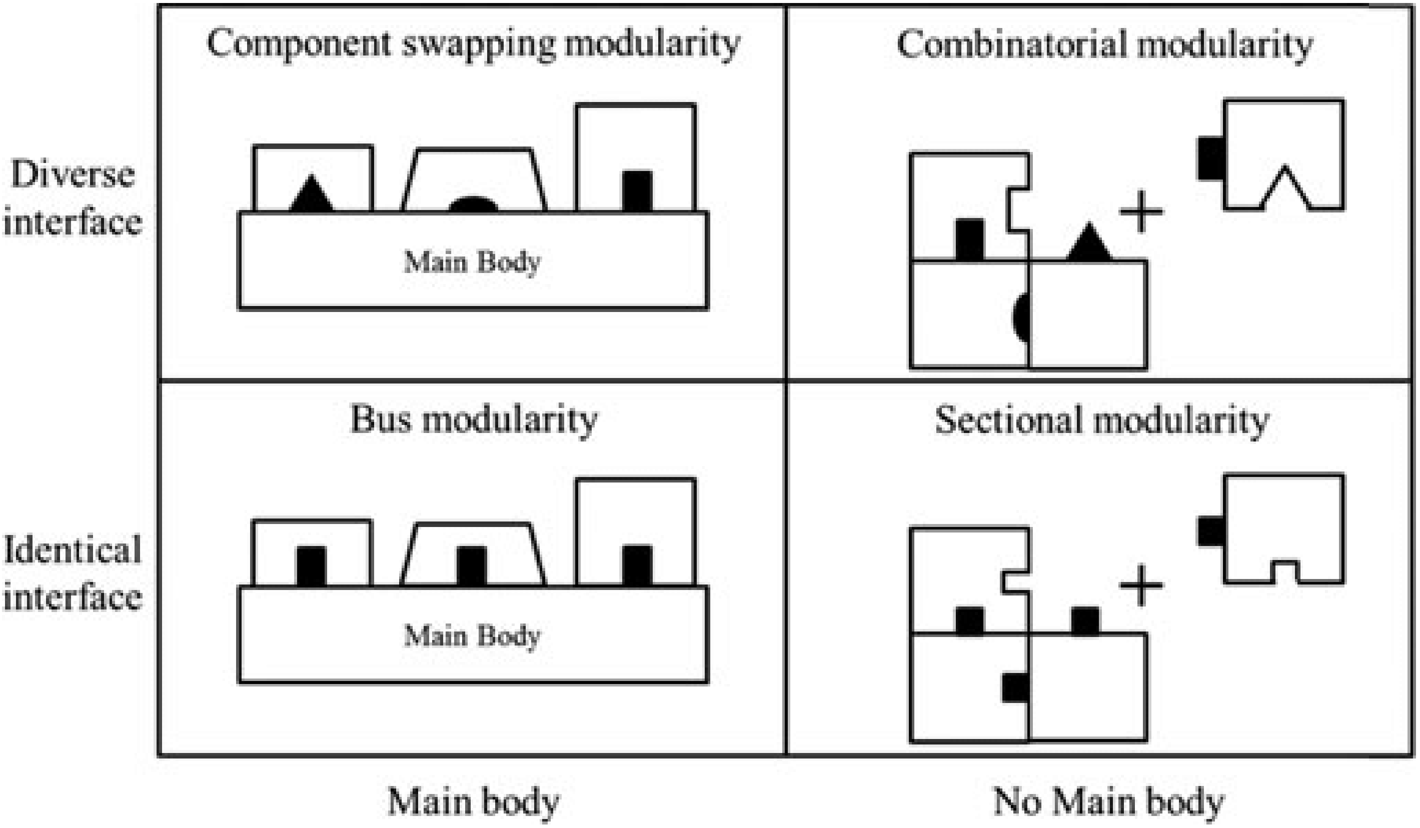

Java EE Application 1
Java EE Application 2
Application Client
Dynamic HTML Pages
Client Tier
Client Machine
JSP Pages
Web Tier
Java EE Server
Enterprise Beans
Enterprise Beans
Business Tier
Database
Database
EIS Tier
Database Server

Modularity

- A complex system may be divided into simpler pieces which is called modules.
- The main benefit is that it allows the principle of separation of concerns to be applied in 2 phases;
  - Dealing with details of each module (isolation)
  - Dealing with overall characteristics of all modules and relationships to integrate them.
- The design are bottom up and top-down design.

Modularity (2)

- There are three goals that modularity tries to achieve in practice;
  - Decomposability : dividing the original problem top down into subproblems and then decomposition recursively to each subproblem. (Divide and conquer; divide and isolate them first and conquer them individually)
  - Composability : starting from bottom up from elementary components and proceeding to the finished system.


Component swapping modularity
Combinatorial modularity
Diverse interface
Main Body
Bus modularity
Sectional modularity
Identical interface
Main Body
Main body
No Main body

Abstraction

- To identify the important aspects of a phenomenon and ignore the details which is a special case of separation of concerns. (Separate important aspects with unimportant details.)
- Ex. Of watch, the abstraction is a box that can be opened to replace the battery.
- For the cost estimation, consists of identifying some key factors of new system and extrapolating from the cost profiles of previous similar systems.

I'M THE PERFECT ABSTRACTION OF A DUCK!
MAYBE ARE YOU SLIGHTLY TOO ABSTRACT?

Anticipation of Change

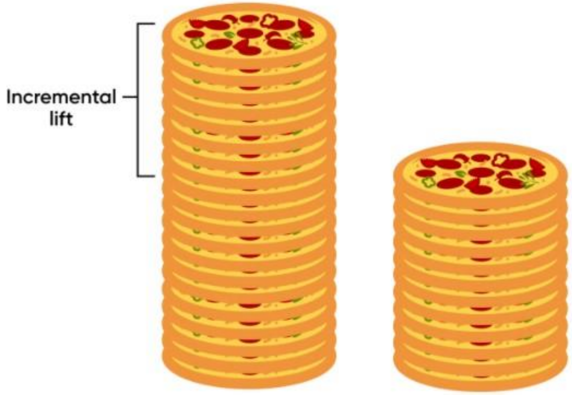
- The changes covers; the need for repairing the SW as;
  - Eliminating errors that were not detected before releasing the application.
  - The need for supporting evolution of the application as new requirements arise, or old requirements have been changed. So that the maintainability as a major SW quality.
- It maybe the one principle that distinguishes SW the most from other types of industrial productions.
- It is a principle that we can use to achieve evolvability.
- Reusability strongly affected by anticipation of change.

Generality

- Generalized solution may be more costly, in terms of speed of execution, memory requirements or development time than the specialized solution that is tailored to the original problem.

- It is a fundamental principle if the SWEng goal is to develop general tools or general packages for the market which will provide standard solution to common problems are increasingly available.

Incrementality



- It consists of identifying useful "early subsets" of the application that may be developed and delivered to customers, to get "early feedback". (Because there is no way of getting all the requirements right before an application is developed.)
- Evolutionary SW development requires special care in the management of documents, programs, test data, and etc.
- Each meaningful increment step must be recorded, documentation must be easily retrieved, changes must be applied in the controlled way… so on…