**MAKERERE UNIVERSITY**

**COLLEGE OF ENGINEERING, DESIGN, ART AND TECHNOLOGY**

**SCHOOL OF ENGINEERING**

**DEPARTMENT OF COMPUTER AND ELECTRICAL ENGINEERING**

**CMP2101: SOFTWARE ENGINEERING**

**NAME: ASINGWIRE ALVIN**

**REG No: 12 / U / 135**

**STUDENT No.: 212000531**

**SIGNATURE: ..........................**

**QUESTION**: COMPARE AND CONTRAST THE PROS ACHIEVABLE THROUGH VERTICAL STRUCTURAL PARTITIONING AND HORIZONTAL STRUCTURAL PARTITIONING.

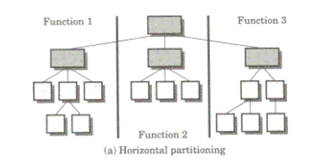
Structural Partitioning

**Horizontal Structural Partitioning**

Horizontal partitioning describes separate branches of the modular hierarchy of reach major program function. The Control modules, represented in a darker shade are used to coordinate communication among and execution of program functions. The easiest approach to horizontal partitioning describes three partitions - input, data transformation often called processing and output. Partitioning the architecture horizontally gives a number of distinct advantages:

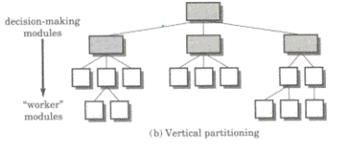
* Results in software which is easier to test
* Leads to software which is easier to maintain
* Results in propagation of fewer side effects
* Results in software which is easier to extend

Because huge functions are decoupled from one another change tends to be less extensions and complex to the system a general occurrence tend to be easier to accomplish without side effects.



**Vertical Structural Partitioning**

The vertical partitioning often called factoring suggests in which control and work should be distributed top- down in the program architecture. The Top-level modules should perform control functions and do little original processing work. Modules which reside low in the architecture should be the workers performing all computational, input, and output tasks.



Partitioning the architecture vertically has the following advantages:

* These are easy to maintain the changes
* They reduce the change impact and error propagation

The nature of modification in program architectures justifies the requirement for vertical partitioning. The change in a control module high in the architecture will have a higher possibility of propagating side effects to modules which are subordinate to it. The change to a worker module given its low level in the structure is less likely to cause the propagation of side effects. In common changes to computer programs revolve around changes to input computation or transformation and output. Overall control structure of the program example for its basic behavior is far less such as to change. For this reason vertically partitioned architectures are less likely to be susceptible to side effects when changes are made and will thus be more maintainable - a key quality factor.

*References:*

* *“Software Engineering” by Ian Sommerville, Addison-Wesley, 2001*
* [*http://msdn.microsoft.com/en-us/library/ms178148.aspx*](http://msdn.microsoft.com/en-us/library/ms178148.aspx)
* [*http://dev.mysql.com/tech-resources/articles/performance-partitioning.html*](http://dev.mysql.com/tech-resources/articles/performance-partitioning.html)
* *http://www.expertsmind.com/topic/design-concepts-and-principles-/structural-partitioning-92036.aspx*