VI. Synchronize & stablize model - (A version of incremental model)

(i) Requirement analysis - interview, extract a list of features with priorities set by the customer

(ii) A specification document

(iii) Work is divided into 3 or 4 builds

1st build consist of most critical features
2nd build consist of next critical features
& so on.

(iv) Each build is carried out by a no. of small team working parallel

(v) At end of each day all the team synchronize - partially completed components are tested together & debug the resulting product.

(vi) stabilization - at the end of each build and faults detected are fixed & the build is frozen

Adv. (i) synchronization - components always work together

(ii) Early insight into the operation of product & can modify the requirement during course of a build

(iii) for incomplete requirements.

iv. Extreme programming - (Based on incremental model)

(i) S/W development team determine the features that client would like the product to support

(ii) for each feature - team informs the client how long it will take to implement that features & how much it will cost.

(iii) client selects features to be included in successive build using cost benefit analysis.

(iv) The proposed build is broken down into several pico-planned tasks.

(v) Task is then integrated into current version of product. Test cases for task are retained & utilized in all further integration testing.

features of extreme programming

1. computers are setup in the centre of large room lined with small cubicles.
2. A client representative work with XP team
3. NO individuals can work overtime for 2 successive weeks
4. There is no specialization .. All work on design, code & testing
5. NO overall design phase before various builds are constructed.
6. The design is modified while the product is being build.
7. Whenever a test case will not run, code is

iv. Extreme programming – (Based on incremental model)

(i) S/W development team – determine the features that client would like the product to support

(ii) for each feature – team informs the client how long it will take to implement that features & how much it will cost.

(iii) client selects features to be included in successive build using cost benefit analysis.

(iv) The proposed build is broken down into several pieces called tasks.

(v) Task is then integrated into current version of product. Test cases for task are retained & utilized in all further integration testing.

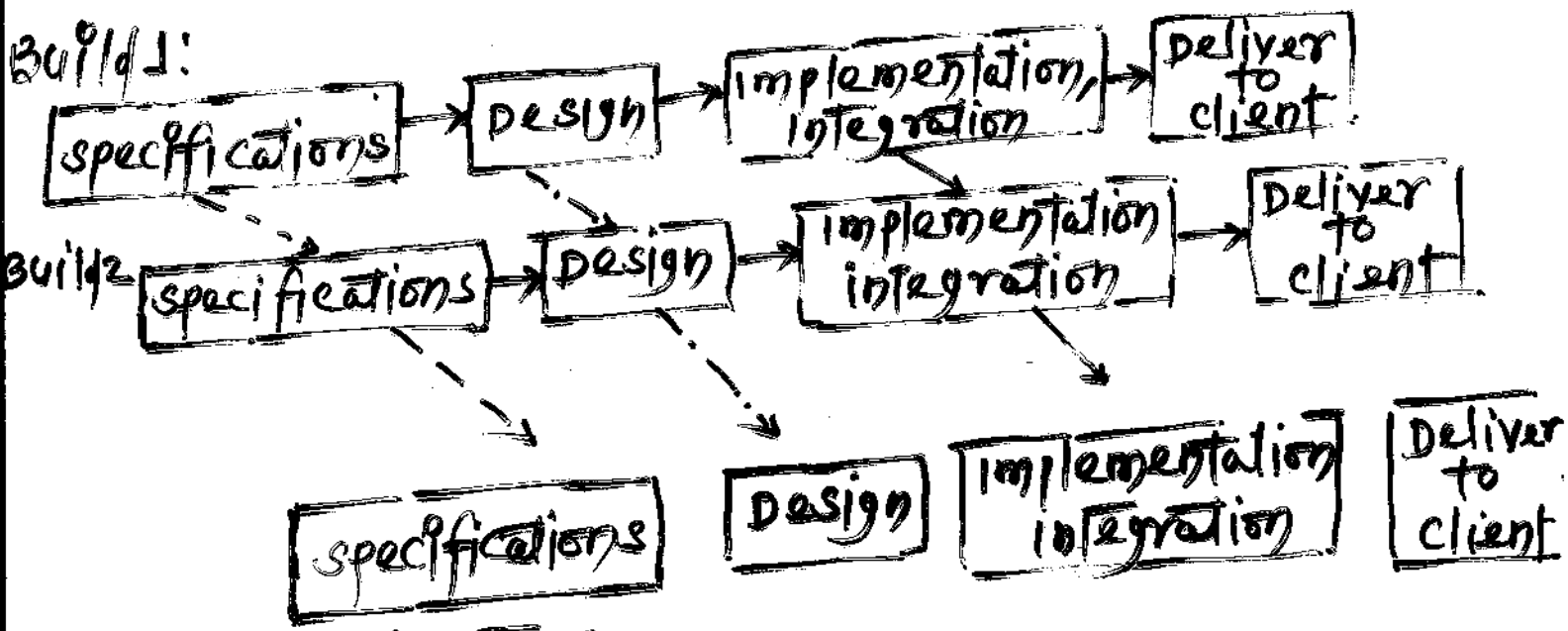
features of extreme programming

1. computers are setup in the centre of large room lined with small cubicles.
2. A client representative work with XP team
3. NO individual can work overtime for 2. successive weeks
4. There is no specialization .. All work on design, code & testing
5. NO overall design phase before various builds are constructed.
6. The design is modified while the product is being build.
7. Whenever a test case will not run, code is reorganized.

Analysis-

In waterfall & rapid prototyping model-
- deliver to client a complete product
- There is project delivery date

Incremental model - It deliver an operational quality product at each stage



- - - → specification team
— . — → Design team
——→ implementation/integration team

Adv. - Gradual introduction
- change & adoptions are natural

Difficulty- Each additional build has to be incorporated into existing structure.

Incremental model does not distinguish b/w developing a product & maintaining it.

Begin with a design that support entire product view product as a sequence of builds, each independent of next.

Advantage.

1. As compared to a waterfall model Development of the process is linear preceding from rapid prototype to delivered product. feedback loops are less likely to be (needed) in this case

Rapid-

* construct the product prototype as possible
* use of rapid prototype- determine what the client's real needs are:
* Rapid prototype implementation is discarded
* Internal structure of rapid prototype is not relevant.
* prototype is modified rapidly to reflect client need

Integrating the waterfall & Rapid prototyping Model

iv. Incremental Model

The product is designed, implemented, integrated & tested a series of incremental builds, where a build consists of code pieces from various modules interacting to provide a specific functional capability

At each stage - a new build is coded & then integrated & tested as a whole.

Break up the target product into builds subject to constraint that each build is integrated into existing s/w the resulting product must be testable.

If too many or too few builds

(ii) RAPID PROTOTYPING MODEL

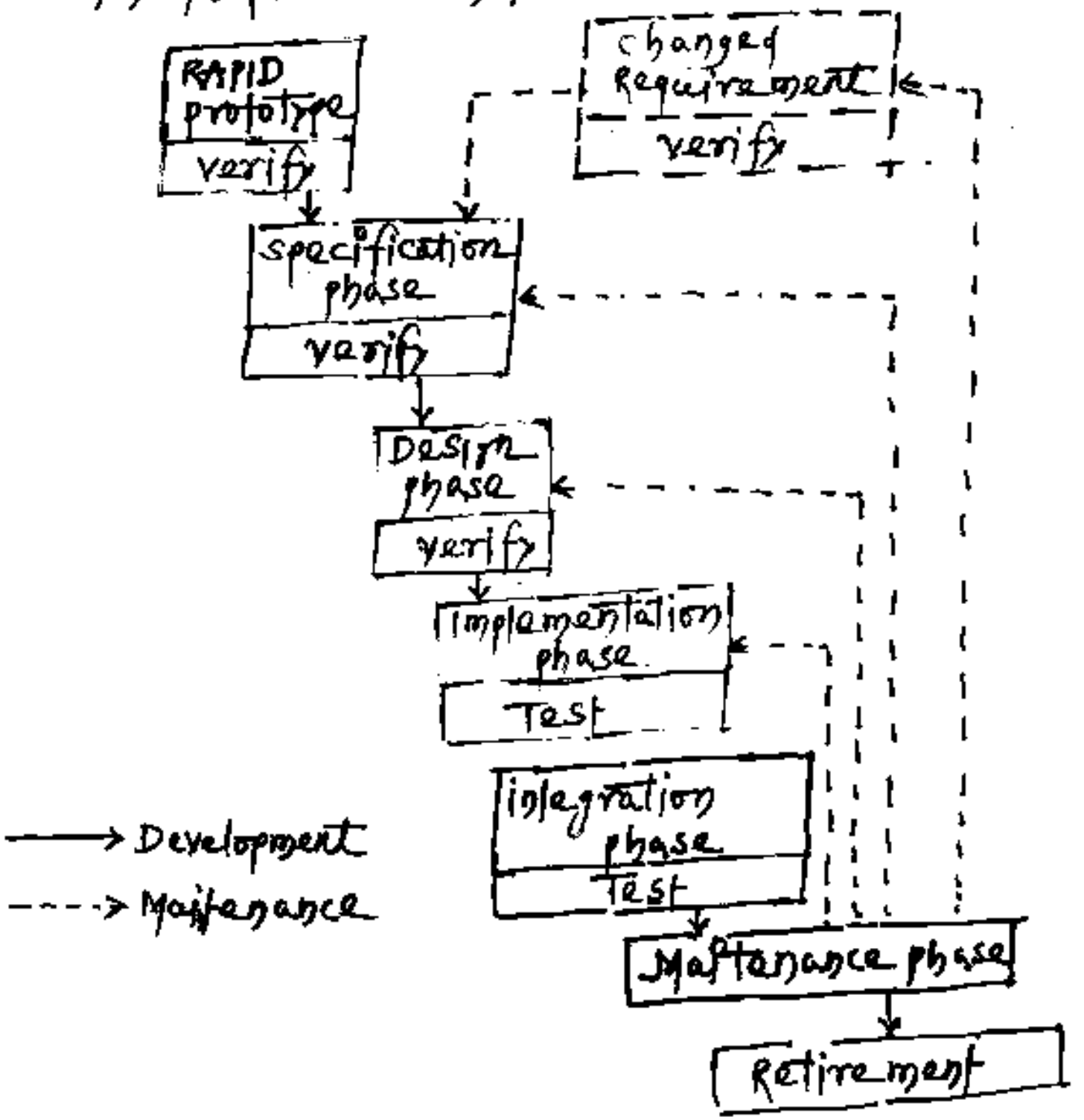
RAPID PROTOTYPING IS A WORKING MODEL FUCTIONALLY EQUIVALENT TO A SUBSET OF THE PRODUCT

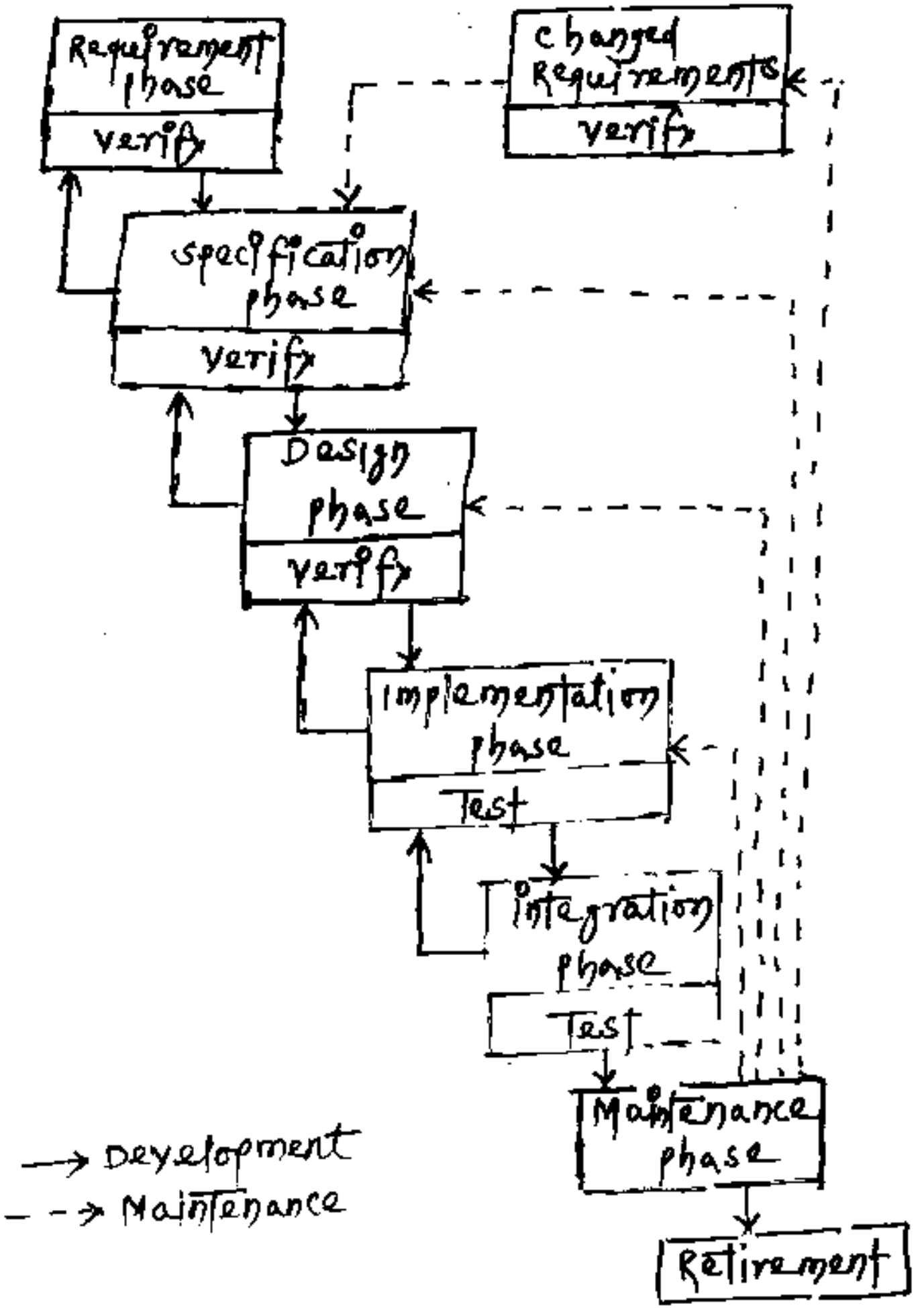
(i) Build a RAPID prototype

(ii) The client and future interact & experiment with it.

(iii) once the client is satisfied, the developer can draw up Specs. document

Then s/w process continues.





Waterfall model

(vi) Design - "how to do". If some problems in specification (incomplete, contradictory & ambiguous) planning and design documents are adjusted

(vii) Implementation:

If flaws --

feedback loops permits modification to be made to design document, specification document & requirement if needed.

(viii) Documentation:

No phase without documentation

Also approved by SQA group

(ix) Acceptance testing

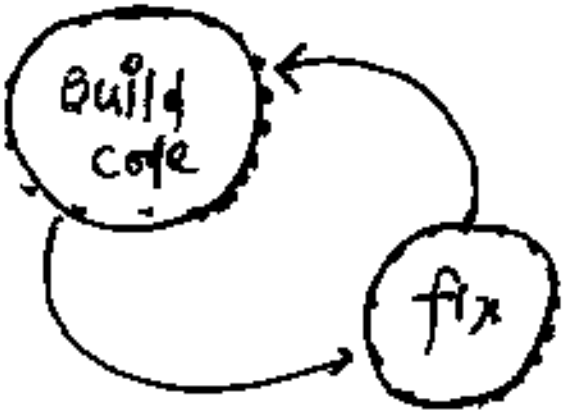
(x) Installation

(xi) Maintenance

Not only implementation changes but also design & specification changes.

Adv. (i) enforce the documentation after every phase
(ii) product of each phase check by SQA
(iii) Testing is inherent in every phase
(iv) Dynamic model - feedback loops

Dis (i) specification documents are long, detailed & very boring to read, client is inexperienced because of its style
(ii) The 1st time the client see a working product after entire coding



Disadvantages.

(i) cost --> very high development cost
(ii) Maintenance difficult → Without specification and document
(iii) unsatisfactory for S/W of any reasonable size
(iv) code soon becomes unfixable and unenhanceable

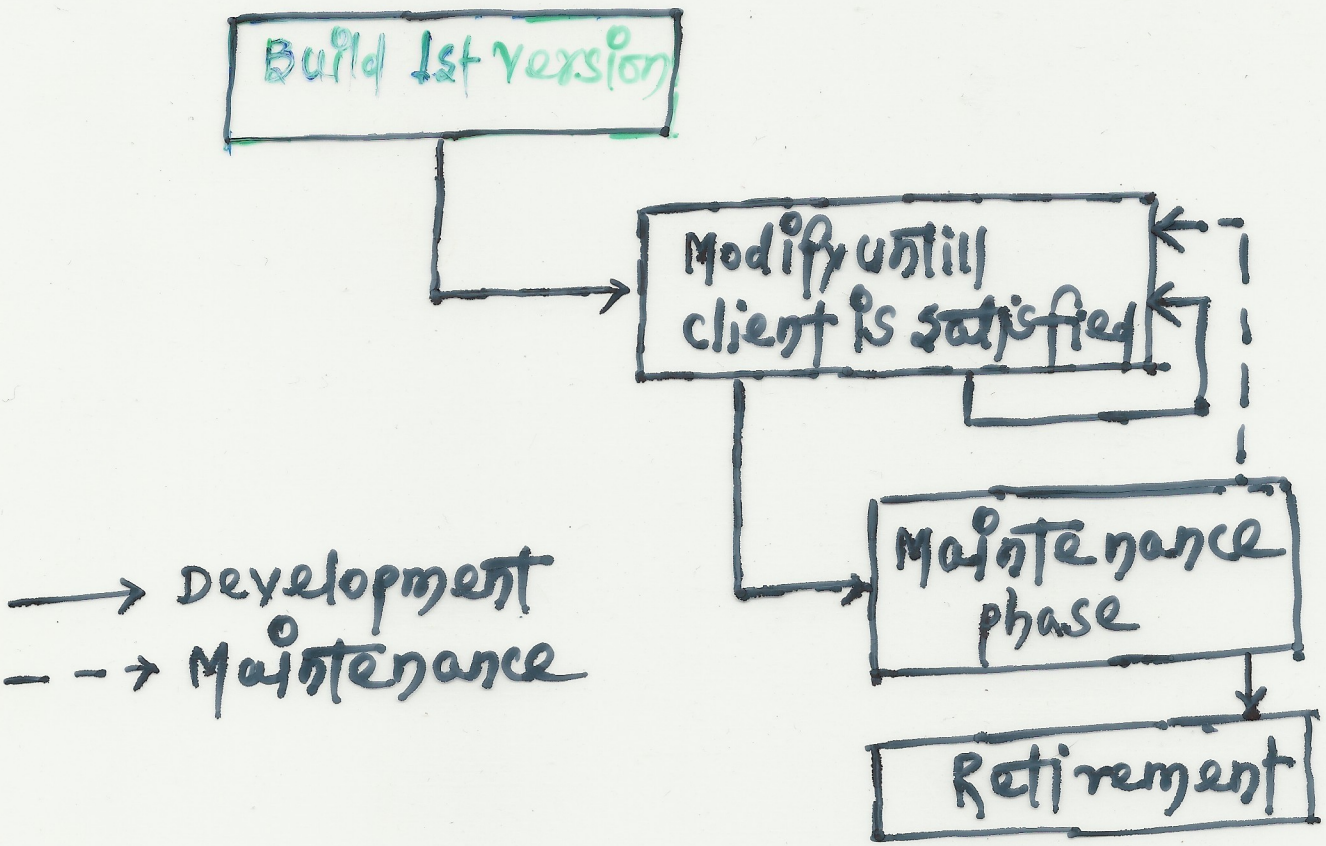
II. Waterfall Model

(i) Requirements determined & checked by the client & SQA group
(ii) Specification drawn
(iii) Specification checked by the SQA group & shown to the client.
(iv) Sign off the specification document
(v) S/W project management plan - checked by SQA group

S/W LIFE cycle Model

1. Build and fix model:

(i) product is constructed with no specifications.
(ii) Rework as many times as necessary.
(iii) This is an adhoc approach and not well defined
(iv) Basically it is a simple two phase model.
(v) The first phase is to write code and the next phase is to fix it.



(vi) fixing in this context may be error correction and/or addition of further functionality.
(vii) Although this approach may work well on small programming exercise 100 or 200 lines long.