# Theory 1



## Waterfall model (classical):

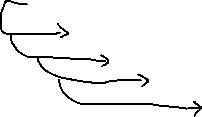


Objective Findings



Requirement Specifications

Analysis structural/Behavioural

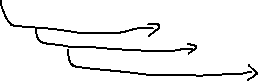


Design Er diagram/DFD

Implementation

Testing

Risk Analysis



Maintenance

* Interations/loops, Software development lifecycle

## Prototyping Model

Objective Finding

Requirement Specifications

Quick Deciosions

Queries from Provide Prototype

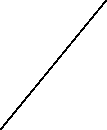
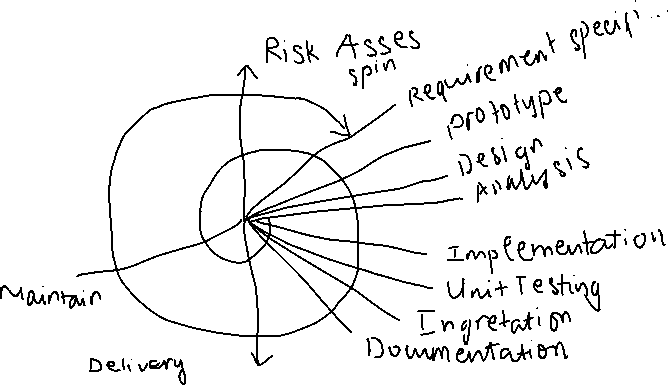
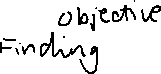
the Admin by Customer

Difinding Quries on Prototype Matching

Design -> Implement -> Testing ->

Documentation-> Risk Assessment ->Delivery Maintenance

## Spiral Model



## Compare

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Classical waterfall | Interative watefall | Prototyping | Spiral |
| Benifits | - Clear structure and milestones  - Thorough documentation  - Straightforward for small, stable projects  - Easy to plan | - More flexibility than classical  - Better error correction via feedback loops  - Incremental refinement  - Controlled iteration | - Early user involvement  - Reduced miscommunication  - Faster validation of tricky features  - Improved requirement definition | - Built-in risk analysis  - Highly adaptive  - Frequent feedback from users  - Good for large or complex projects |
| Chanlenges | - Inflexible to scope changes  - Late feedback/testing  - High risk of rework  - Minimal early customer involvement | - Still somewhat rigid  - Can complicate management  - Partial risk of late discovery  - Requires disciplined change control | - Scope creep  - Risky if prototype code becomes final  - Unrealistic user expectations  - Potential time/cost overheads | - High complexity and cost  - Demands expert risk management  - Potential scope overrun  - Hard to estimate completion time |
|  | T (1) | O (n) | O (S^d) | O (S ^ dt) |
|  | S (1) | O (m, n) 7 x 8  = O (56) Unit | 2^5 = 32 | T (S) = t(1) + O(n) + o(s^d) |