Traditional Development

Analysis / Design

Process models
- Usually Waterfall derivatives
- Data and Function based models
- Dataflow / Entity-Relation / Structure Diagrams (DFD ERD Structure Chart)
- RDBMs, supporting CASE tools

Common Methodologies
- Big organizations have their own
- Known methodologies:
  - Yourdon
  - Jackson
  - SSADM
  - SDM, SADT, DSSD ....

Common Approach
- Analysis:
  - DFD, ERD, Data Dictionary
- Design
  - DFD, Tables, Normalization, PSL/PDL
  - Data -> Structure -> Function
  - Transform Analysis
- ‘Procedural’ language + SQL

Analysis: DFD
- External Entity: User
- Process: Password check
- Data store: ID info
- Data: ID info
Traditional Software Development

DFD: Context Diagram

DFD: Level 1

Entity Relationship Diagrams

An Example ERD

A Data Dictionary Example

Analysis to Design
Modular Complexity

- Complexity:
  - \[ C(p1+p2) > C(p1) + C(p2) \]
- Functional Dependency:
  - Cohesion
  - Coupling

A System and its Modules

Transform: Transform Flow

Transaction Flow

First Step in Transform

Transaction flow: first step
Transaction flow: second step

Procedural Specification: PDL

Do
Read account number
If (invalid account number) go to beginning
Ask for operation type
If (Operation type == deposit) then [Deposit(); go to beginning]
Ask for password
If (invalid password) go to beginning
If (Operation type == balance) then [Balance(); go to beginning]
If (insufficient balance) then go to beginning
If (Operation type == money transfer) then [Transfer(); go to beginning]
If (Operation type == money transfer) then [Transfer(); go to beginning]
Else Withdraw()
While (continuous)

Testing
- Test Plan
- Test cases
- Testing techniques
- Testing strategies

Testing Techniques
- Black box
- White box
  - Exhaustive testing: prohibitive complexity
  - Basis path testing: manageable:
    - Cyclomatic complexity =
      - # decision points + 1
      - N_n - N_e + 2
      - # Loops in a planar flow-graph

Basis path testing

A = 2
While B <> A do
  If C > A
    D = C
  Else if D > C
    D = A
End while
If D > 2
  D = A
B = A
Cyc.Comp = 9 - 6 + 2 = 5

Testing Strategies
- Top-down
- Bottom-up
- Sandwich
Object Oriented Approaches

- Programming languages: raising abstraction
- Languages in 80s, methodologies later
- A modeling that is closer to human mind than to machines
- Very similar OO modeling languages
- Consistent development media from analysis to coding
- Meanwhile maturing of SE; incorporating non OO techniques

Object Oriented Approaches - II

- Entities should be modeled with their operations
- Supports reuse and modularity
- Complexity is attempted to be addressed by generalization and specialization mechanisms
- Details to come soon…

Newly Developing Concepts

- Component Technologies
- Design Patterns
- Architectural Frameworks
- No methodological support yet!
- Domain orientation