System Development (SDLC)

- 1. Identify the problem
  - SO
    - Type of SDLC
      - Predictive(waterfall)
        - planned in advance
        - develop according to intended plan
        - Requirements are well understood
        - Requirements are stable
        - low technical risk
      - Adaptive
        - Incremental/Development
          - Completes portions of the system in small increments
          - The complete system structure is built first
        - Walking Skeleton
    - Methodology
      - Models
      - Techniques
      - Tools
    - Every organization relies on IS
      - Reason 1: to carry out and manage their business operations
      - Reason 2: to interact with their customers and suppliers
      - Reason 3: to compete in the marketplace
    - 4 fundamentals components
      - Technical sub-system
        - Technology (IT)
        - Process
        - People
      - Social sub-system
        - Structure
- 2. Plan and monitor the project
  - Systems Development Process
    - Iterative development
      - "grower" piece-by piece
      - Through multiple iterations
      - Complete small part of system (mini- project), then repeat processes to refine and add more until done
    - Agile development
      - Features
        - flexibility to add new requirements
        - responsive to change
        - Iterative in nature
        - responding to change
      - Values
        - individuals and interactions over processes and tools
        - working software over comprehensive documentation
        - customer collaboration over contract negotiation
      - Principles
        - Develop software as the primary goal(instead of documents or models)
        - Minimize modeling activity
        - Embrace change and change incrementally
        - quick feedback from end-users and other developers
        - communicate with others
        - Maximize stakeholder ROI
      - Example
        - Unified process (UP)
          - FOUR phases
            - Inception phase(start)
            - Elaboration(understand requirements)
            - Construction
            - Transition
          - Two types of disciplines
            - System Development activities
            - Project Management activities
        - Extreme programming (XP)
          - core values
            - Simplicity
            - Feedback
            - Courage(Willing to throw away bad codes and design choices)
            - Communication (among stakeholders)
          - practices
            - Planning(Ask on user stories instead of use cases)
            - Testing(every step)
            - Pair Programming(algorithm/code, switch roles)
            - Simple Designs
            - Refactoring(Improving code)
            - Ownership of code(everyone can review and improve)
            - Continuous integration
            - ……
        - Scrum
          - PRODUCT BACKLOG
          - SPRINT BACKLOG
- 3. Discover and understand the details
  - Investigating system requirements
    - Type
      - Functional
      - Non-functional
    - Stakeholders
      - Interest
      - Power/Influence
    - Information Gathering Techniques
      - Interviewing Users and Other Stakeholders
      - Distributing and collecting questionnaires
      - Review Existing Reports, Forms, and Procedure Descriptions
      - Observing and documenting business procedures
      - Research Vendor Solutions
      - Collect Active User Comments and Suggestions
    - Models
      - use cases
        - THREE formats
          - Tabular format without details
          - Tabular format with details
          - Graphical format
            - Two popular types
              - << Include>>(should include, from origin to another)
              - <<Extend>>(may include, from extend back to origin)
      - user stories
        - Content
          - "As a <role> I want to <goal> so that <benefit>"
          - Acceptance Criteria(identify the features to satisfy user)
        - Benefits
          - Easier to communicate with users
          - Users do not need to be trained to understand
          - Real time feedback
          - Encourages collaboration and feedback
        - Avoids
          - design detail
          - technical function
      - Domain Class diagrams
        - Model
          - classes
          - attributes
          - relationships
          - multiplicities
        - Relation types
          - Generalization
          - Specialization
          - Superclasses
          - Subclasses
          - Inheritance
          - Aggregation
          - Composition
      - sequence diagrams
      - activity diagrams
- 4. Design system components
  - TWO levels
    - Architectural Design(Conceptual)
    - Detailed Design
  - FIVE MAJOR Design Activities
    - Describe the environment
      - Communications with External Systems
      - Conforming to an existing Technology Architecture
    - Design the application components
      - O: Design class
        - Symbols
          - Attributes
            - Visibility(+public, -private)
            - Attribute name(Lower case camelback notation)
            - Type expression(class,string,integer,double, date)
            - Initial value
            - Property(such as Date)
            - Class level attribute(Underline)
          - Method signatures
            - Method Visibility
            - Method name
            - Method parameter list
            - Return type expression
          - Class level method
            - Visibility
            - Entity Class
            - Boundary (View Class)
            - Controller Class
            - Data Access Class
          - Type
        - OO principles
          - High-Cohesion
            - Class focus on one function, achieve one aim
          - Low Coupling
            - Not too close to other classes, should independent
          - Navigation Visibility
        - Steps of Object-Oriented Design
          - Choose a single use case
          - Choose any one of THREE paths
            - CRC Cards(simple)
            - Communication Diagram(mid)
              - components
                - Actor
                - Object
                - Link
                - Message
              - Message Syntax
                - Iteration condition([])
                - sequence number(1.1, 1.2, 2)
                - return value(:=)
                - message name
                - parameter list
                - Multiply occurring message(*)
            - Sequence Diagram(complex)
              - System Sequence Diagram
          - Package diagram construction
          - First-Cut Design Class Diagram
    - Design the user interface
      - Six key principles
        - Human-Interface Objects
          - Affordance(the appearance of the object suggests its function)
          - Visible with Feedback
        - Consistency
        - Discoverability
        - Closure
          - Users should know when a task is completed
          - UNDO to reverse actions
        - Readability and Navigation
          - target audience
          - Readable text for all users
          - Clear navigation
        - Usability and efficiency
          - SHORTCUT KEYS
          - Simplicity
            - Follow KISS (Keep It Simple Stupid) principle
          - Meaningful error
            - Contents of error messages:
              - problem
              - reason
              - solution
      - COMPLEX reason
        - multiple devices, device has constraints (e.g. resolution, screen size, memory)
        - using different OS
        - purpose (input, dialog box, report) varies
        - user characteristics differ(users with disability, novice/experienced)
      - Component
        - Physical aspects
          - Device
        - Perceptual aspects
          - everything else (e.g. colours, sizes, sound, font)
        - Conceptual aspects
      - Metaphors
        - Direct manipulation metaphor
          - display are manipulated to look like physical objects
        - Desktop metaphor
        - Document metaphor
        - Dialog metaphor
      - UI issues for Smartphones
        - Small screen size
        - Layout and formatting
          - Rotating view
          - Resizing screen
          - Visible navigation
          - Scrolling
        - Small keyboards, Touch screens
          - Data entry and user actions
            - Fat finger
        - Limited network capacity
    - Design the database
    - Design software classes and methods(Detailed Design)
      - Design Class Diagram
      - Sequence Diagrams
      - State Machine Diagrams
    - Controls and Security
      - Integrity Controls
        - Inputs
          - Prevent invalid or erroneous data from entering the system
        - outputs
          - To ensure that output arrives
            - at proper destination
            - in accurate
            - current
            - complete
        - data and programs
      - Security Controls
        - Fraud prevention
          - internal
        - Encryption
          - external
    - Internal deployment environment
      - O: Three-layer (tier) client server architecture
        - Layers
          - View (Client)layer
          - Domain layer (Application logic)
          - Data layer
        - MIDDLE LAYER translates client requests into data access commands that can be understood by data servers
        - Reduces the workload of data servers. Relieves clients from complex processing tasks
        - each server is specialized with a certain task
      - 2-tier architecture(client-server)
    - Designing Reports
      - Detailed reports
      - Summary reports
      - Exception reports
      - Executive reports
- 5. Build, test, integrate system components
  - Test
    - aims
      - determine its operational characteristics (e.g. performance)
      - Whether it contains bugs
      - uncover design flaws and limitations
      - verify that it is 'FIT FOR PURPOSE'
      - reduce the risk of failure
    - Types
      - Unit test
        - Advantage
          - reducing the cost of bug fixes
          - issues are found at early stage
      - Integration Testing
        - Parameter values
        - Run-time exceptions
        - Unexpected state interactions
      - Stress Testing(Performance)
        - Response time
        - Throughput
      - User Acceptance Testing (UAT)
    - Benefits
      - Better quality of software
      - More value to customers/users
      - satisfied customers
      - Improved user experience
- 6. Complete and deploy the solution
  - Converting
    - Convert data into new format
    - Identify relevant data from various systems
    - Identify and capture data from paper sources and manual data entry
  - Training Users
    - hands-on training, including practice exercises, questions and answers, and one-on-one tutorials
    - enablers
      - system operators
        - less formal, self-study
  - Documentations
    - System Documentation
    - User Documentation
  - O: Deployment approaches
    - Direct deployment
    - Parallel deployment
    - Pilot deployment
    - Phased deployment

XMind: ZEN | Trial Mode