Study Objectives

• Traditional SDLC
• Waterfall Method and its Variants
• Requirement Phase
Systems Development Life-Cycle (SDLC)

• What are traditional SDLC Phases?
  A. Requirements
  B. Feasibility
  C. Risk Management
  D. Design
  E. Development
  F. Project Planning
  G. Performance Testing
  H. Implementation
  I. Post Implementation
CISA SDLC Phases and Relationship

Phase 1 – Feasibility

Phase 2 – Requirements

Phase 3A – Design

Phase 4A – Development

Phase 5 – Implementation

Phase 6 – Post Implementation

Phase 3B – Selection

Phase 4B – Configuration

Build

Buy

Reviews at the end of each phase acts as “stage gate”
## Traditional SDLC Phases

<table>
<thead>
<tr>
<th>Phase</th>
<th>Phase Name</th>
<th>Key Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feasibility</td>
<td>- Feasibility, Business Case, ROI</td>
</tr>
<tr>
<td>2</td>
<td>Requirements</td>
<td>- Business Requirements, Systems Requirements, Vendor Requirements</td>
</tr>
<tr>
<td>3A</td>
<td>Design</td>
<td>- Systems Architecture: overall system</td>
</tr>
<tr>
<td>OR</td>
<td>Selection</td>
<td>- High Level Design: overall system</td>
</tr>
<tr>
<td>3B</td>
<td></td>
<td>- Detailed Design: module, interface, sub-systems, service, database, UI etc.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Selection of Vendor and Vendor supplied system</td>
</tr>
<tr>
<td>4A</td>
<td>Development</td>
<td>- Coding</td>
</tr>
<tr>
<td>OR</td>
<td>Configuration</td>
<td>- Testing: unit, integration, systems, performance</td>
</tr>
<tr>
<td>4B</td>
<td></td>
<td>- Configuration of the Vendor supplied system</td>
</tr>
<tr>
<td>5</td>
<td>Implementation</td>
<td>- Data Conversion, Change Management, Implementation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Final UAT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Incident, Outage, and Availability Management</td>
</tr>
<tr>
<td>6</td>
<td>Post Implementation</td>
<td>- Lesson Learned</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Evaluation of Benefits vs. Business Case</td>
</tr>
</tbody>
</table>
Water-Fall SDLC Model

BOEHM’S WATERFALL MODEL

1. SYSTEMS REQUIREMENTS
2. SOFTWARE REQUIREMENTS
3. PRELIMINARY DESIGN
4. DETAILED DESIGN
5. CODING
6. TESTING
7. OPERATION & MAINTENANCE

SOURCE: Adapted from Barry Boehm’s SDLC Model, IEEE, 1975

Barry Boehm’s water-fall model
Waterfall Model

• One of the common and traditional SDLC methodology
• Barry Boehm’s modified SDLC model in 1970s
• Feedback loop allows for validation between successive phases
Waterfall Model

• When will you use?
• Come up with Examples (5 minutes)
Waterfall Model - Pros and Cons

- Structured
- Works well when requirements are well defined
- Better for relatively larger projects
- Better if there is time in hand

- Too much documentation
- Making changes becomes difficult during SDLC
- Speed to market – not quick
- Delay in implementation
Verification and Validation Model (V Model)

V-Model is a variant of water-fall model

Adapted from “CISA Review Manual” ISACA, Exhibit 3.8”
Spiral SDLC Model

Spiral Model is a variant of water-fall model
Spiral SDLC Model

• It’s iterative
• Each iteration goes through mini-waterfall
• Gives Business Team time to refine and prioritize requirements
• Helps assess Business Value before committing to entire project
• Helps course correct
• More common than traditional waterfall SDLC
Question

What’s the main reason of the reviews at the end of each phase in the SDLC?

A. Funding Approval to continue development
B. Approval by management to proceed to the next phase or possibly kill the project (stage gate)
C. Design and Code Familiarity
D. Internal Auditor compliance
Requirement Phase

• Why is Requirement Definition Important?

A. Helps Define Business Needs
B. Helps come up with systems requirements
C. Makes Business “wish list” to more concrete definition of the ask
D. Provides foundation to the rest of the SDLC phases
E. All of the above
Business/User Requirements

- Defines functional needs of the Business and how systems should solve it
- Includes operational, quality, and performance needs as well
- Many times not written concisely
- Describe from users’ perspective what they want
- Could have “must-have” as well as “nice to have” requirements that need to be further prioritized
- Example: “1. Provide ability to collect user name and address field in a web UI”
- Generally written by the Users or Business Team
Systems Requirements

- Defines functional needs of the Business and how systems should solve it, from systems perspective
- Includes operational, quality, and performance, interface, reporting, troubleshooting requirements as well
- Written concisely
- Describe from users’ perspective what they want, as well from systems perspective how would the systems handle them
- Would have “must-have” as well as “nice to have” requirements that are prioritized
- Traceability is often included
- Specificity is the goal rather than generality
- Tools such “Use Case” is used to determine systems behavior
- Written by the IT Team in collaboration with Business and Design Teams
Business vs. Systems Requirements

Example

1. Provide ability to collect user name and address field in a web UI. The customer should be able to submit name and address data before proceeding to the ordering page.

   1.1 Provide ability to enter person’s first name and last name in separate input fields
   1.2 Provide ability to enter address, which would have Street number, Street name, city, and zip code in separate input fields
      1.2.1 An address must have street name and city
      1.2.2 Zip code must be 5 character numbers
   1.3 If a user clicks on “submit” button without entering name and address fields, prompt the user with “you must enter your name and address to continue”
   1.4 If a user is a returning user, pre-populate the existing user name and address
   1.5 Provide ability to store the name and address fields as part of the customer records
   1.6 Provide ability to submit the name and address page in less than 1 second
   1.7 Provide ability to report on for manual re-entry by Support team, when the customer submits the name and address field and data is not successfully written in customer records
   1.8 The systems should be able to handle 1 million concurrent transaction with no degradation in performance
Characteristics of Good Requirements

• Complete and Comprehensive
• Consistent
• Traceable, verifiable, testable
• Clear and unambiguous
• Modifiable
• Prioritized
<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement</th>
<th>Type</th>
<th>Priority</th>
<th>Must Have?</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Provide ability to collect name and address from the end users</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Provide ability to enter person’s first name and last name in separate input fields</td>
<td>Functional</td>
<td>High</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>Provide ability to enter address, which would have Street number, Street name, city, and zip code in separate input fields</td>
<td>Functional</td>
<td>High</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.2.1</td>
<td>An address must have street name and city</td>
<td>Functional</td>
<td>Medium</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.2.2</td>
<td>Zip code must be 5 character numbers</td>
<td>Functional</td>
<td>Low</td>
<td>Nice-to-have</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>If user clicks on “submit” button without entering name and address fields, prompt user with “you must enter your name and address to continue”</td>
<td>Functional</td>
<td>Medium</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>If a user is a returning user, pre-populate the existing user name and address</td>
<td>Functional</td>
<td>Medium</td>
<td>Nice-to-have</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>Provide ability to store the name and address fields as part of the customer records</td>
<td>Functional</td>
<td>High</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>Provide ability to submit the name and address page in less than 1 seconds</td>
<td>Performance</td>
<td>Medium</td>
<td>Nice-to-have</td>
<td></td>
</tr>
<tr>
<td>1.7</td>
<td>Provide ability to report on for manual re-entry by Support team, if the customer submits the name and address field and data is not successfully written in customer records</td>
<td>Operational</td>
<td>High</td>
<td>Must-have</td>
<td>Address part of reporting</td>
</tr>
<tr>
<td>1.8</td>
<td>The systems should be able to handle 2 million concurrent transaction with no degradation in performance</td>
<td>Performance</td>
<td>Medium</td>
<td>Must-have</td>
<td>At least 1 Million</td>
</tr>
</tbody>
</table>
Requirements Gathering

• BRD from the Business and Users
• SRD includes implicit IT requirements including those related to interface, reporting, availability, reliability, etc.
• Gap Analysis
  – Considering Top Down (strategic) and Bottom up (functional) analysis of the system, function, people etc.
• Security Requirements related to
  – Data, transmission of data (PKI, HTTPS, SSL for example), authentication, authorization, access
  – Access to the systems, Fire walls etc.
  – Information protection
• Vulnerability Management
Requirement Phase Best Practices

• Involve stakeholders (IT, User, Business)
• Avoid conflicting or not-achievable requirements
• Prioritize
• Use structured format
• Leverage Good Requirements characteristics
• Include systems implicit requirements (operational, interface, quality), which are typically not easily comprehended by the users/Business
Use Case

• What does Use Case do?

• Helps in defining Requirements by how users would interact with the system

• Provides a practical way to decompose and come up with the requirements

• Often helpful on a complex project
# Use Case Example

## Name: Entering Name and Address

### Identifier: Use Case #10

### Description: This use case captures how a user would enter name and address as part of the web ordering for a pre-registered user

### Pre-condition: User must have clicked on “order” button on the home page and user is not a pre-registered user

### Steps:
1. User comes to the Personal information screen
2. The system validates the user is not a pre-registered user
3. Systems paints the name and address fields
4. User would enter first and last name
5. User would enter street name and city
6. User would enter street number and zip-code (optional fields)
7. User clicks submits
8. System stores name and address data
9. Systems shows the name and address was completed and paints “product” screen

### Post-condition: The user name and address data is now stored

### Alternate Cases:
- **10.1 (Edits fail):**
  1. Systems prompts user to enter incomplete field

- **10.2 (Name and Address data could not be stored):**
  1. Systems prompts user to contact customer service representatives with phone number

### Notes: this use is part of the ordering flow

---

### Additional Use Case Reference:

[http://www.agilemodeling.com/artifacts/systemUseCase.htm](http://www.agilemodeling.com/artifacts/systemUseCase.htm)
Auditing Requirements Definition

- Obtain the requirements artifacts such as BRD (Business Requirements Document) and SRD (Systems Requirements Document)
- Adequate participation from the key stakeholders: Business, IT (Requirements and Design teams specifically), and Users
- Ensure BRD and SRD meet quality guidelines. The best Systems Requirements are detailed, clear (unambiguous), consistent, traceable, verifiable, modifiable, and prioritized.
- Review high-level design and architecture to ensure they meet the Business needs and “reasonably” correspond to the BRD/SRD
- Determine if application is a candidate for the use of an embedded audit routine
Question

Requirements can be gathered by all except the following

A. Developing a mock system or prototype
B. Interviewing users, business, and IT teams
C. Speaking to Vendors to understand which software is selling well in last two years
D. Getting an understanding on what other companies did in a similar situation
Upcoming Assignments/Tests

1. Quiz 1 (Business Case Development, Project Management Practice, Develop Project Control, etc. Materials covered in the first 4 Classes – 1/14, 1/21, 1/28, 2/4). Multiple Choice CISA Exam type questions: Thu 2/11

2. Group Project -1 (Business Case, RFP): Thu 2/18 before the class

Questions?
Summary of Today’s Class

• Traditional SDLC
• SDLC Phases
• Variants
• Requirement Phase
• Focus of the Next Class and Reading
• Questions