

Installation and Maintenance of Health IT Systems

Unit 5

The Software Development Life Cycle

What We'll Cover...

- What is the Software Development Life Cycle (SDLC)?
- Why do we need SDLC?
- Phases
- The waterfall model
- The iterative model
- The spiral model
- An example
- SDLC and EHR Systems

What is SDLC?

- SDLC = Software (or Systems) Development Life Cycle
 - A detailed plan on the creation, development, implementation, and eventual phase-out of a software package
- Several different models exist. Two typical categories are:
 - Waterfall model
 - Iterative model

Why Do We Need SDLC?

- Software purchases and upgrades can be costly.
- Integration of poorly designed or untested software can be devastating to a business.
- Poorly designed software increases security risks.
- Failure to adequately plan for software integration can limit efficiency and cost in project over-runs and lost productivity.

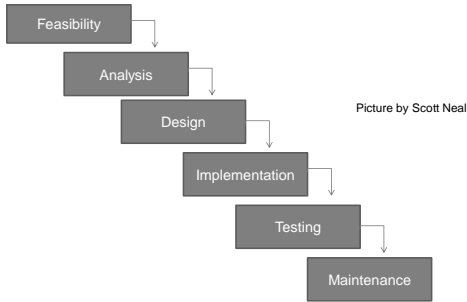
Factors for Success

1. Management support
2. Technical and business expertise
3. Determining the product focal points
4. Following a well-defined procedure
5. Proper documentation for maintenance

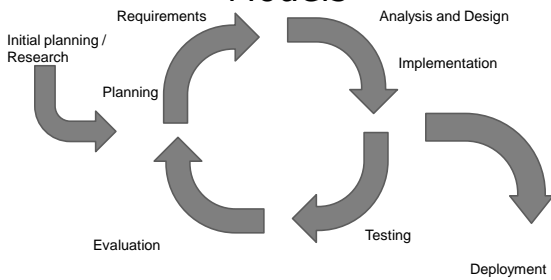
Phases of a Typical SDLC Model

- Many different models exist for developing software systems.
- All models follow some variation of these general phases:
 - **Initiation**
 - **Concept development**
 - **Planning**
 - **Requirements analysis**
 - **Design**
 - **Testing**
 - **Implementation**
 - **Maintenance**
 - **Disposition**

Waterfall SDLC Model



Iterative and Incremental Models



Initiation Phase

- Need is identified, and project manager is assigned.
- Concept Proposal, outlining the business need and justification, is generated and approved by upper management.
- Once approved, the project moves to the next phase: Concept Development.

Concept Development Phase

- Defines the scope of the development project
- Useful reports:
 - Feasibility study
 - Cost / benefit analysis
 - System boundary analysis
 - Risk management report

Planning Phase

- What are you supposed to deliver?
- What personnel will be needed?
- What external resources should you bring in, if any?
- Develop in-house or purchase software?
- What hardware constraints do you have?

Requirements Analysis Phase

- Focuses on what the system will actually do
- What specifically are the user requirements?
- What OS and interfaces?
- How proficient will the users need to be?

Design Phase

- A blueprint of the software is developed.
- Program components and workflow are established.
- Program documentation begins to take shape.

Development Phase

- The software product is coded and assembled and takes on life.

Integration and Testing Phase

- Formalized process using parameters developed during the design stage
- “Roll-Out” testing helps ensure stability in the real world environment.
- New software is tested to ensure that data can be easily migrated from the obsolete software into the new product reliably.

Implementation Phase

- User communication and training take place.
- Data is migrated from the old system and checked for integrity.
- The system is brought online. Whenever possible, the old system continues to function in case there are roll out issues.
- After a successful distribution, data is gathered to determine successful implementation.

Operations and Maintenance Phase

- Day-to-day operation of the software
- System continues to be monitored for anomalies and bugs.
- Patching and updates are deployed as needed or to improve functionality of the product.
- The product lifetime can be extended.

Disposition Phase

- A plan to close down the software once it becomes obsolete or is replaced
- Encompasses a wide range of detail including:
 - Safe, secure disposition of software and obsolete hardware components
 - Archiving of documentation
 - Secure transition, with destruction or archiving of data

Common SDLC Models

- Waterfall
- Iterative

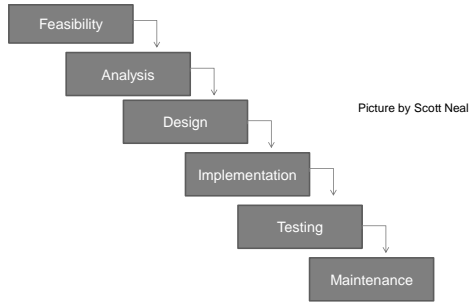
The Waterfall SDLC Model

- Employs more traditional techniques for developing software
- Introduced in 1970
- Uses a sequential development process
- Each phase is perfected before progressing forward
- Often criticized as an imperfect model

Phases of the Waterfall Model

- Requirements (or Analysis)
- Design
- Construction, Implementation, or Coding
- Integration
- Testing and debugging (or Validation)
- Installation
- Maintenance

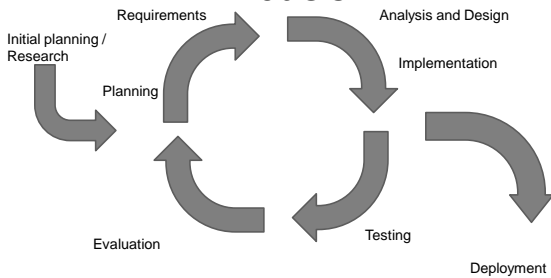
Waterfall SDLC Model



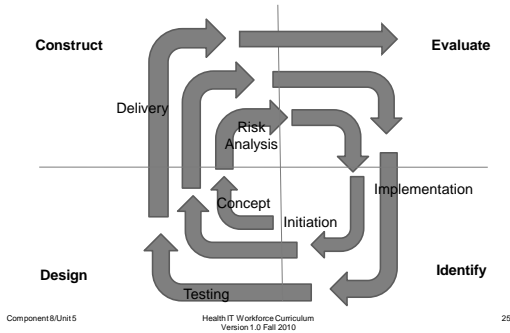
Iterative and Incremental Models

- Developed to address weaknesses in the waterfall model
- Cyclic process
- Back-tracking allowed
- Allow developers to design a system using repeated cycles (iterations)

Iterative and Incremental Models



Spiral Model



A Not-So-Real-Life Example

- Widget Inc.'s market research identifies the need: efficient square jar canning software.
- R&D devise and test a conceptual canning software system that should address the needs. A feasibility study is submitted and approved.

A Not-So-Real-Life Example

- The design team builds a blueprint and writes documentation of how the canning software should operate once completed.
- Next, the implantation team begins coding the modules. At any time, errors and problems may arise requiring intervention of the project manager. Documentation is completed to ensure the product can be adequately troubleshoot and maintained.

A Not-So-Real-Life Example

- The coded software is successfully tested, errors are corrected, and the software is retested.
- The support infrastructure is put into place that will provide customer support and upgrading as needed using the original documentation as a baseline.
- Software is finally brought into production!
- Quality Assurance team identifies issues for correction and passes off to the support team.

The SDLC and EHR Systems

- Particularly important if planning in-house design or EHR program modifications, e.g. for integration
- Needed to ensure product satisfaction and quality assurance
- Needed to mitigate risk factors and minimize downtime
