



Information Systems  
Audit and Control  
Association®

< FOCUS > < DISCOVER > < INTEGRATE > < ADVANCE >

# System Development Life Cycle

Presented By:

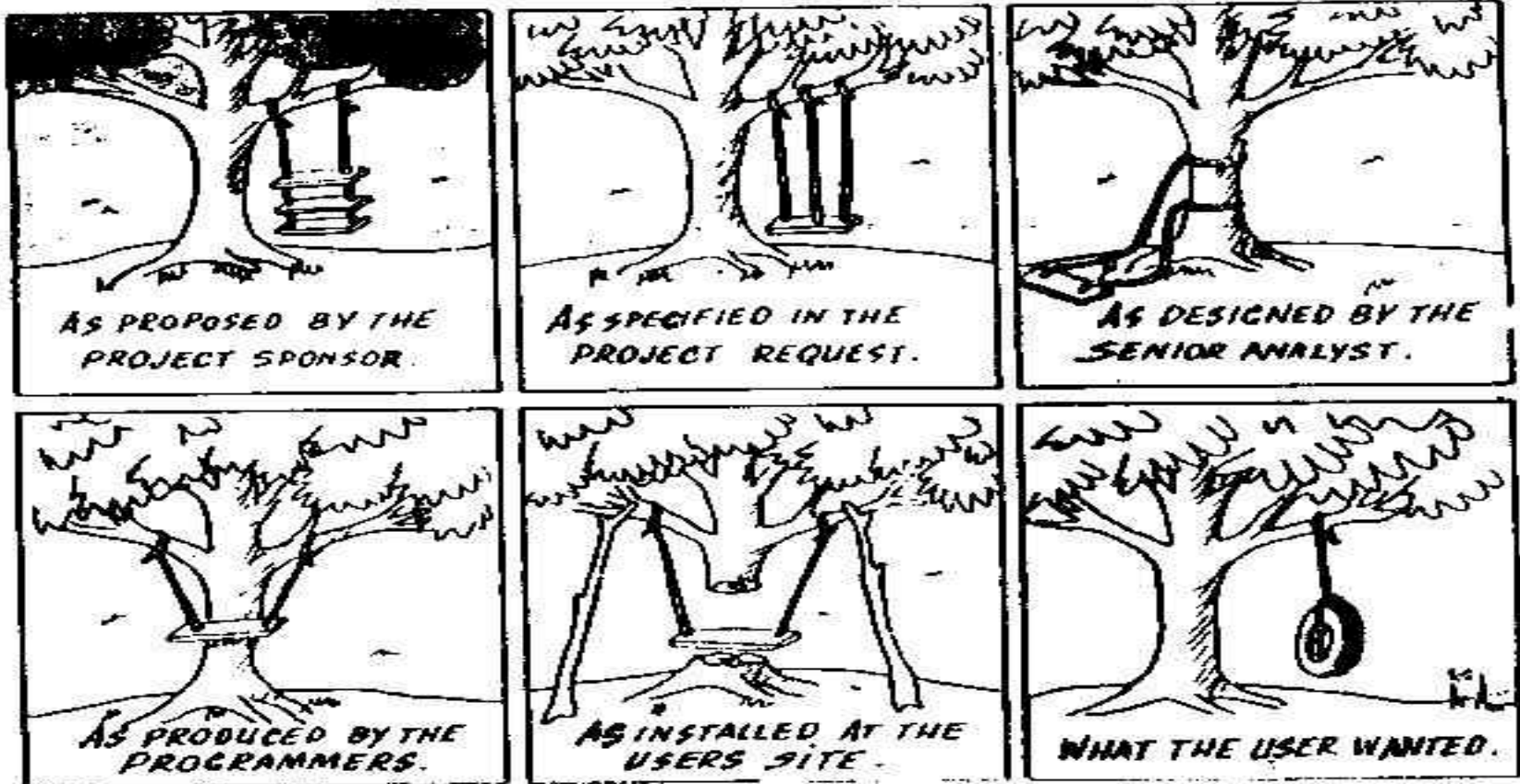
**Dave Anderson**

The San Francisco Chapter of ISACA Proudly Announces the 4th Annual

2004 SF ISACA FALL CONFERENCE

October 4-6, 2004

# SDLC Risks



# Today's Presentation

---

◆ 8:30 – 8:35      Introduction

◆ 8:35 – 9:45      Presentation

◆ 9:45 – 10:00      Q & A

# What is SDLC?

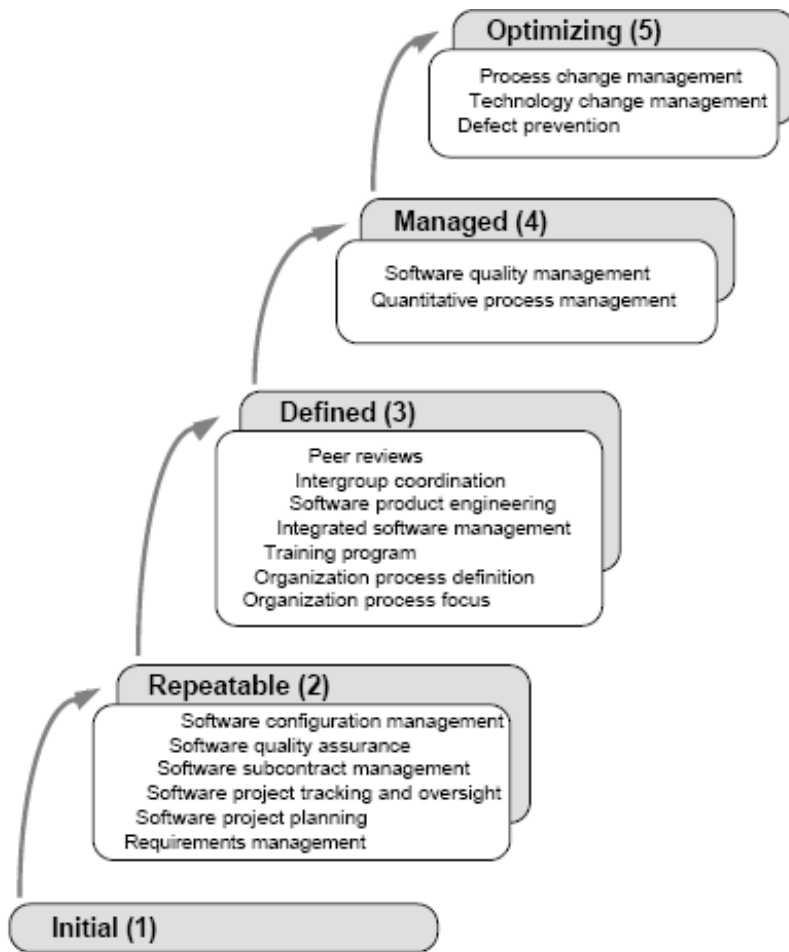
- ◆ **Phased approach to developing custom software**
- ◆ **Each phase builds on the activities of the prior stage**
- ◆ **Each phase has its own set of tasks and deliverables**

# SDLC Phases

---

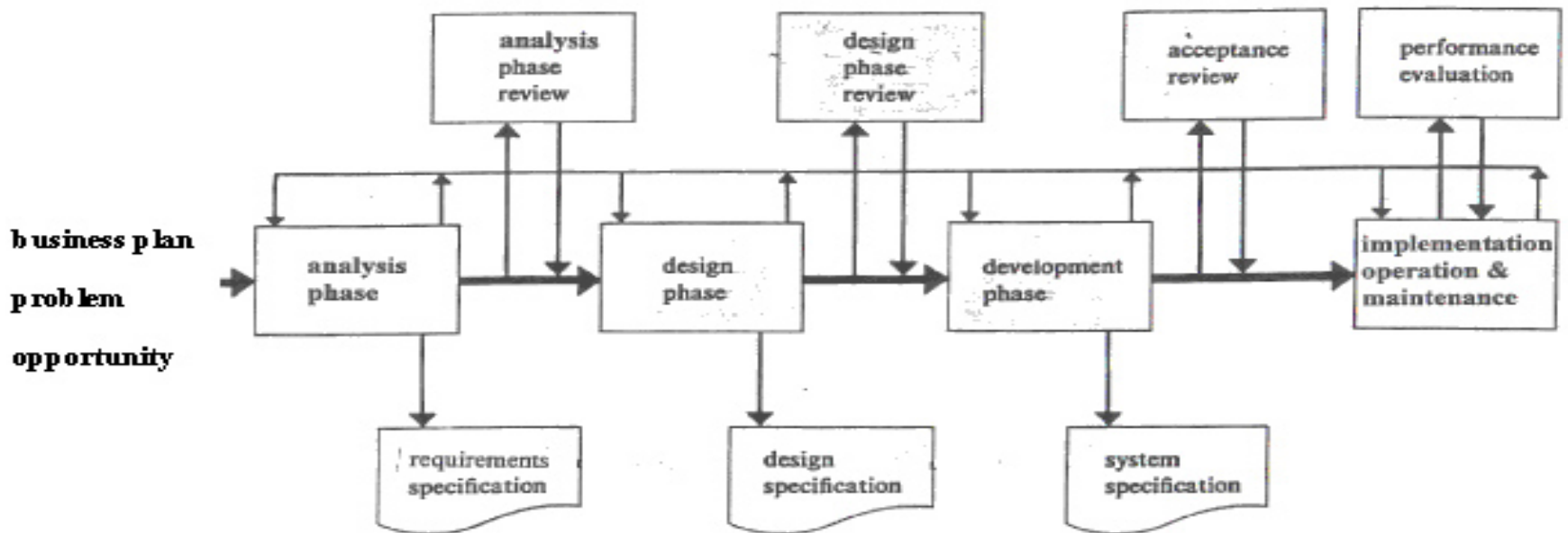
- ◆ **Requirements / Analysis**
- ◆ **Design**
- ◆ **Development / Construction**
- ◆ **Testing / Quality Assurance**
- ◆ **Implementation\***
- ◆ **Operation & Maintenance**

# “CMM”

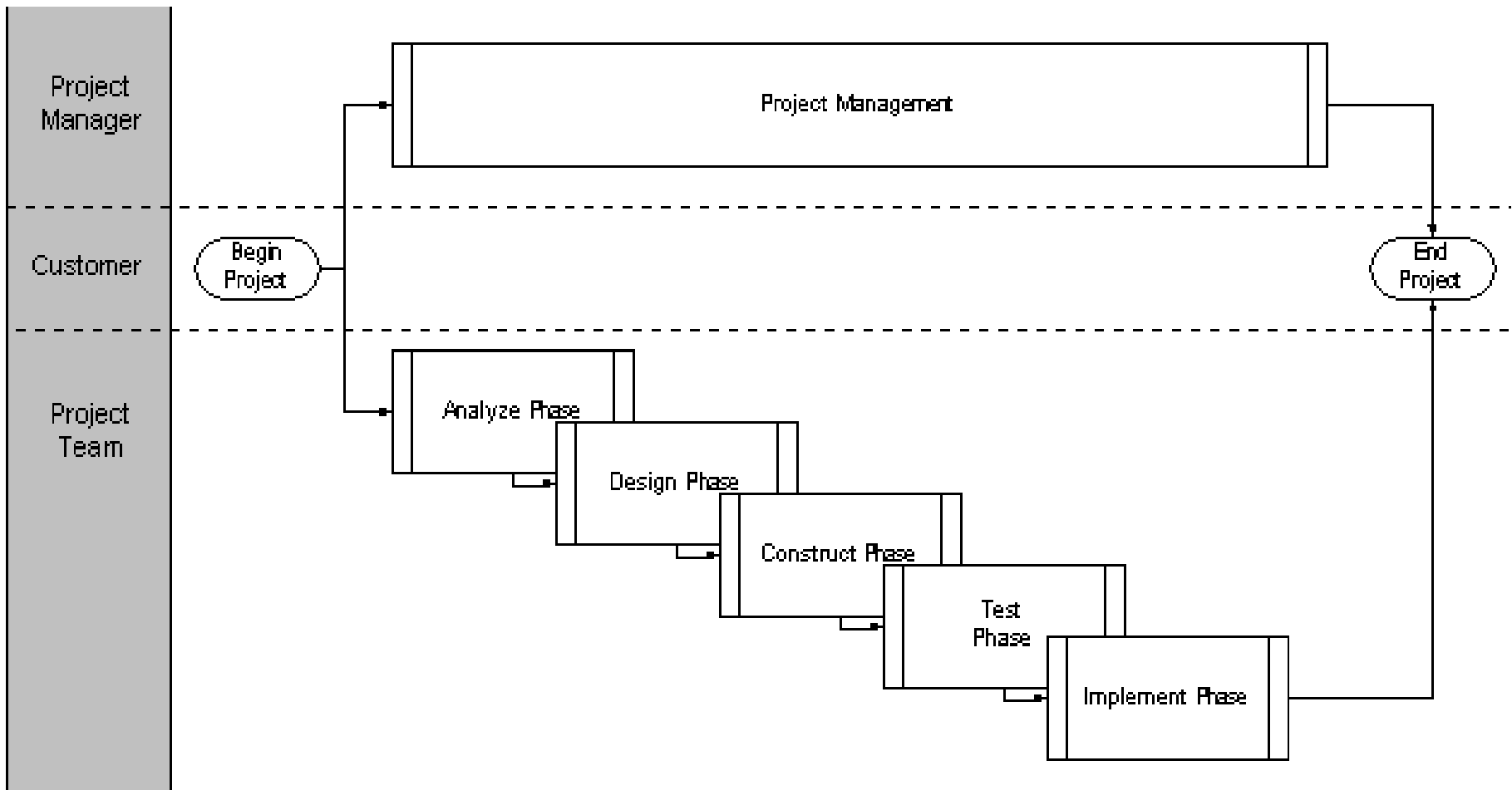


**The Software Engineering Institute (SEI) at Carnegie Mellon has developed this “Capability Maturity Model”, measuring an organization’s level of formalization of software development practices**

# One view of SDLC



# Another view of SDLC



# Why use SDLC?

- ◆ **Ensure standard approach is used**  
**Duplicable + Repeatable**  
**= High Software Quality**
- ◆ **More important now, with SOX requirements for internal controls**

# Typical Problem Areas

- ◆ **Disconnect between Biz & IT on requirements**
- ◆ **Users not sufficiently involved**
- ◆ **Inadequate testing**
- ◆ **Poor change management**
- ◆ **Security/controls not addressed in system**

# Caution! SDLC needs...

◆ **Success also depends upon:**

– **Project risk management**

– **Testing**

– **Change management**

– **Data conversion**

– **Quality of documentation**

# What Info Do I Need?

- ◆ **Policies & procedures**
- ◆ **Project documentation**
- ◆ **Interviews with key stakeholders**
- ◆ **Testing documentation**
- ◆ **Samples to test for compliance**
- ◆ **Results of previous SDLC audits**

# Typical Policies

- ◆ **Procedures and standards for custom software development shall be followed for projects**
- ◆ **Decision whether to proceed with custom development or to acquire a package solution is made at the end of the requirements phase**

# Typical Policies

- ◆ **System development mgr and user(s) review, approve, and sign off all deliverables at each phase**
  - **Requirements Document**
  - **Conceptual Design Document**
  - **User Documentation**
  - **User Training Materials**
  - **Implementation Plan**

# Requirements Phase

---

- ◆ **Kick off project**
- ◆ **Collect & validate requirements**
- ◆ **Analyze data**
- ◆ **Evaluate alternatives**
- ◆ **Make recommendations**

# Evaluating Alternatives

---

## ◆ Make

- Modify an existing system
- Develop a custom solution
  - Using on- or off-shore development teams

# Evaluating Alternatives

## ◆ Buy

– Purchase packaged software

• use “vanilla” with no/minimal changes, OR

• modify/customize to meet your needs

– Use a 3d party service provider

# Design Phase

---

- ◆ **Conceptual design**
  - **Design functionality**
- ◆ **Detailed design**
  - **Select infrastructure**
  - **Design data structure**
  - **Design programs**

# Development Phase

---

## ◆ Coding

- System logic
- Output/Reports
- Interfaces

# Various Approaches

---

## ◆ Linear (traditional) Development

- Each phase occurs once and must finish before next can start

## ◆ Iterative Development

- Each phase occurs many times

# Testing Phase

---

- ◆ **Testing or “debugging”**
  - **Unit test**
  - **Integration test**
- ◆ **User acceptance test**
  - **Performance testing**

# Implementation Phase

---

- ◆ **Implementation strategy**
  - **Parallel**
  - **Big Bang**
- ◆ **Conversion plan**
- ◆ **Cut-over plan**

# Post-Implementation

- ◆ **Assess the implementation**
  - **What went well?**
  - **What went poorly?**
  - **How should SDLC change/evolve to improve?**

# Operation Phase

## ◆ Documentation

- **Functional: user manuals, work instructions, FAQ's, 'cheat sheets'**
- **Technical: Sys Admin manual, documentation of system design and functionality, controls, security**

# Maintenance Phase

---

- ◆ **Daily operations**
  - **Boot up/shut down**
  - **Backup/restore**
- ◆ **Disaster Recovery Planning**
- ◆ **Training**

# Resources

- ◆ **Capability Maturity Model**  
[www.sei.cmu.edu/cmm/cmms/cmms.html](http://www.sei.cmu.edu/cmm/cmms/cmms.html)
- ◆ **ISACA Audit Guideline-SDLC:  
Document # 060.020.100**
- ◆ [www.all.net/books/audit/kits/sdlc/pgm.html](http://www.all.net/books/audit/kits/sdlc/pgm.html)
- ◆ [www.auditserve.com/ap/sdlc.htm](http://www.auditserve.com/ap/sdlc.htm)

# Resources

- ◆ **SDLC “Bookshelf”**

**[www.technosphere.net/reading.htm](http://www.technosphere.net/reading.htm)**

- ◆ **SDLC downloads from Maryland State Dept of Education (free)**

**[docushare.msde.state.md.us/docushare/dsweb/View/Collection-3755](http://docushare.msde.state.md.us/docushare/dsweb/View/Collection-3755)**