The Software Process

- Step and procedures executed by a software organization during the development of software
- Not a linear sequence of steps
- Many activities occur simultaneously or in random order
- Organizations often continually improve and change their software processes

Software Process Activities

- Requirements Analysis and Definition
- System Design
- Prototyping
- Implementation and Coding
- Unit Testing
- Integration Testing
- System Testing (Alpha)
- Acceptance Testing (Beta)
- Help and Manual Generation
- Release
- Maintenance

Functional Specification

- Formal document generated after a requirements analysis.
- Contains a detailed list of the functionality required for the system being built.
- Is a contract between the client and developers which states what functionality the end system will have.
- Only technical with respect to domain knowledge associated with the software product’s purpose.
Sample Requirements

- Delivery Schedule
- Performance
- Deployment
- Configurability/Adaptability
- Portability
- Reliability

- Usability (User Interface Features)
  - User Interface Standards
- System Functions
- Maintainability
  - Coding Standards

Functional Specification

- There should be no coupling between the functional specification and the design specification.
  - The functional spec should be concerned with describing the functionality.
  - Functional specification should decouple details of how the system is to be implemented.
  - In the functional specification phase the customer should not think in “design” terms.

Design Specification

- Formal document which describes how the features described in the functional specification document are to be implemented.
- Design Phases
  - High level architectural design
  - Detailed Design (Decomposition)
    - object or module based
  - Interleaved Design testing
Computer Aided Software Engineering (CASE) Tools

- (File) Version Management Tools
  - Aka Source Code Control
  - Supports multiple developers working on same code base
  - Supports ability to track a module through it’s development
  - Supports software releases by tagging file versions
  - Most tools store differences between file versions.
  - Web interfaces

Version Management Tools

- Rational ClearCase
- PVCS Version Manager
- Microsoft Visual SourceSafe
- CVS - Concurrent Versions System
- See this link: http://www.iac.honeywell.com/Pub/Tech/CM/CMTools.html

Software Engineering Tools

- Feature Tracking Tools
  - Track System Requirements
  - Track System Change Requests (Critical Incidents, aka “bugs”)
  - Can associate issue with source files, classes (integration with version management tools.)
  - Responsibility for software requirements can be assigned to specific team members
  - Priority of issues can be assigned and tracked
Feature Tracking Tools

• Any database
• Intersolv: PVCS Tracker
• Rational: ClearQuest
• Elsinore: Visual Intercept
  – Integrates with Microsoft SourceSafe

Specification/Design Tools

• Allow for system modeling and design
• Support diagramming in a standard notation such as UML
• Support code generation from models

Specification/Design Tools supporting UML

• Advanced Software Technologies Inc.
  – Graphical Designer
  – [http://www.advancedsoftware.com/]
• CASElife Systems Inc.
  – Modeller
  – [http://www.caselife.com/]
• Cayenne Software Inc.
  – ObjectTeam, GroundWorks, Terrain
  – [http://www.cayennesoftware.com/]
• Evergreen Software Tools Inc.
  – EasyER EasyER/EASY
  – [http://www.evergreen.com/]
• ICION Computing Inc.
  – NetStart, Catalyst
  – [http://www.icion.com/]
Specification/Design Tools supporting UML

- Intellipad Corp.
  - Intellipad SAP &/ Edition
    - (http://www.intellipad.com/)
- INTERSOL Inc.
  - Allpike Series, AppMaster Designer
    - (http://www.intersolv.com/)
- LogicWorks Inc.
  - Envisio
    - (http://www.logicworks.com/)
- PLATINUM Technology Inc.
  - PLATINUM Paradigm Plus
    - (http://www.platinum.com/)
- Popkin Software and Systems Inc.
  - System Architect, SA/BPM and
    SA/Object Architect
    - (http://www.popkin.com/)

Specification/Design Tools supporting UML

- Rational Software Corp.
  - Rational Rose for Visual Basic
    - (http://www.rational.com/)
- Riverine Software Corp.
  - RIVi
    - (http://www.riverine.com/)
- SELECT Software Tools Inc.
  - SELECT Enterprise
    - (http://www.select.com/)
- Siemens-Nixdorf Information Systems Inc.
  - Conunit Visual Framework
    - (http://www.siemens.com/)
- SoMAs
  - Taquiler
    - (http://www.so-masinc.com/)

Specification/Design Tools supporting UML

- Software One Ltd.
  - Software One OCHMAGE
    - (http://www.softwareone.com/)
- Sterling Software Inc.
  - KEYWare
    - (http://www.sterlingware.com/)
- Sybase Inc.
  - PowerBuilder Designer
    - (http://www.sybase.com/)
- Texas Instruments Software
  - Conquering
    - (http://www.ti.com/software/)
- Vision Software Tools Inc.
  - Vision Builder
    - (http://www.vision-soft.com/)
- Vistia Corp.
  - Vistia Preference
    - (http://www.vistia.com/)
Popular Design Tools

- Support UML notation
  - Rational Rose
  - Microsoft Visual Modeler
  - Microsoft Visio

UML Overview

- A standard notation for
  - Views
    - Diagrams
      - Model elements
        » General mechanisms

Views

- Used to describe a system
  - A single graph can’t describe all the different aspects.
    - Functional
      - Structure, interaction between components
    - Nonfunctional
      - Performance, reliability, deployment
    - Organizational
      - Work organization
Views

- Use-case view
  - System as perceived by the end users
- Logical view
  - Structural description
- Component view
  - Division of work, dependencies between code modules
- Concurrency view
  - For describing synchronization and communication aspects of a system
- Deployment view
  - Describing how a system is deployable

Diagrams

- Use-Case Diagram
- Class Diagram
- Object Diagram
- State Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

Other Software Tools

- Automated testing tools
  - Segue Quality Works
- Make/Build tools
- Metrics tools
A Basic Software Process

- Requirements Analysis and Definition
- System Design and Prototyping
- Implementation
- System Testing

Software Process Models
- Waterfall Model
- Rapid Prototyping Model
- Spiral Model
- Build and Fix Model
- Incremental Model

Waterfall Model
- Requirements Gathering
- Functional Specification
- Design Specification
- Implementation (coding)
- Unit/Integration Testing
- System and Acceptance Testing
- Delivery and Maintenance
**Rapid Prototyping Model**

- Develop Prototype
- Functional Specification
- Design Specification
- Implementation (coding)
  - Often involves extending the prototype instead of recoding
- Unit/Integration Testing
- System and Acceptance Testing
- Delivery and Maintenance

**Spiral Model**

**Incremental Model**

- Requirements Gathering
- Functional Specification
- Design
- Implementation
- Unit/Integration Testing
- System Acceptance Testing
- Software Release
Software Processes and your Career

• As good Software Engineers you should strive to:
  – Seek employment at organizations with higher process maturity
    • Gaining experience at such organizations can help you understand how to build a quality process
  – Seek to improve the maturity of the software process employed by your organization

Software Processes and your Career

• Organizations with poor Software Processes will fail to deliver quality software on time.
• Example: Internet Startups
  – Just because these companies are working in “internet time”, doesn’t mean that they can ignore the rules of software development. If startups fail to have a mature software process they will fail.

Capability Maturity Model (CMM)

• Software Engineering Institute (SEI), Carnegie-Mellon University
• Method for categorizing the maturity and quality of a software process used by software development organizations
**Capability Maturity Model (CMM)**

- **Level 1: Initial Level**
  - Ad hoc process

- **Level 2: Repeatable Level**
  - There is an informal process
  - Code Versioning
  - Feature Tracking

- **Level 3: Defined Level**
  - Process is clearly defined, and documented.
  - Efforts made to improve process
  - Specification and Design Tools Used

- **Level 4: Managed Level**
  - Process is clearly defined and documented and measured.
  - Metrics used
  - Dr. Henry’s expertise
  - Productivity and Quality Goals set, and adhered to.

- **Level 5: Optimizing Level**
  - Process is defined, measured and optimized after each project iteration

---

**What CMM Levels have you experienced?**