

**The Software Process**

- Step and procedure 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 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

- DeliverySchedule
- Performance
- Deployment
- Configurability/Adaptability
- Portability
- Reliability
- Usability(User InterfaceFeatures)
  - UserInterface Standards
- SystemFunctions
- Maintainability
  - CodingStandards

---

---

---

---

---

---

---

---

## 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 specifications 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 and testing

---

---

---

---

---

---

---

---

## ComputerAidedSoftware Engineering(CASE)Tools

- (File)VersionManagementTools
  - Aka SourceCodeControl
  - Supportsmultipledevelopersworkingonsamecode base
  - Supportsabilitytotrackamodulethroughit's development
  - Supportsoftware releasesbytaggingfileversions
  - Mosttoolsstoredifferencesbetweenfileversions.
  - Webinterfaces

---

---

---

---

---

---

---

---

## VersionManagementTools

- Rational ClearCase
- PVCSVersionManager
- MicrosoftVisualSourceSafe
- CVS - ConcurrentVersionsSystem
- Seethislink:  
<http://www.iac.honeywell.com/Pub/Tech/CM/CMTools.html>

---

---

---

---

---

---

---

---

## SoftwareEngineeringTools

- FeatureTrackingTools
  - TrackSystemRequirements
  - TrackSystemChangeRequests(Critical Incidents, aka "bugs")
  - Canassociateissuewithsourcefiles,classes (integrationwithversionmanagementtools.)
  - Responsibilityforsoftware requirementscanbe assignedtospecificteammembers
  - Priorityofissuescanbeassignedandtracked

---

---

---

---

---

---

---

---

## Feature Tracking Tools

- Anydatabase
- Intersolv: PVCSTracker
- 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.advancedsw.com/>)
- CASEwise Systems Inc.
  - Modeler
  - (<http://www.casewise.com/>)
- Cayenne Software Inc.
  - ObjectTeam, GroundWorks, Terrain
  - (<http://www.cayennesoft.com/>)
- Evergreen Software Tools Inc.
  - EasyER, EasyOBJECT
  - (<http://www.estl.com/>)
- ICON Computing Inc.
  - Headstart, Catalysis
  - (<http://www.iconcomp.com/>)

---

---

---

---

---

---

---

## Specification/DesignTools supportingUML

- IntelliCorp Inc.
  - LiveModel SAP R/3 Edition
  - (<http://www.intellicorp.com/>)
- INTERSOLV Inc.
  - Allegris Series, AppMaster Designer
  - (<http://www.intersolv.com/>)
- LogicWorks Inc.
  - Erwin
  - (<http://www.logicworks.com/>)
- PLATINUM technology inc.
  - PLATINUM Paradigm Plus
  - (<http://www.platinum.com/>)
- Popkin Software and Systems Inc.
  - System Architect, SA/BPR and SA/Object Architect
  - (<http://www.popkin.com/>)

---

---

---

---

---

---

---

---

## Specification/DesignTools supportingUML

- Rational Software Corp.
  - Rational Rose for Visual Basic
  - (<http://www.rational.com/>)
- Riverton Software Corp.
  - HOW
  - (<http://www.riverton.com/>)
- SELECT Software Tools Inc.
  - SELECT Enterprise
  - (<http://www.selectst.com/>)
- Siemens-Nixdorf Information Systems Inc.
  - ComUnity Visual Framework
  - (<http://www.sni-usa.com/>)
- Softlab
  - Enabler
  - (<http://www.softlabna.com/>)

---

---

---

---

---

---

---

---

## Specification/DesignTools supportingUML

- Software One Ltd.
  - Software One EXCHANGE
  - (<http://www.software1.com/>)
- Sterling Software Inc.
  - KEYS Workgroup
  - (<http://www.key.sterling.com/>)
- Sybase Inc.
  - Powersoft S-Designer
  - (<http://www.powersoft.com/>)
- Texas Instruments Software
  - Composer
  - (<http://www.ti.com/software/>)
- Vision Software Tools Inc.
  - Vision Builder
  - (<http://www.vision-soft.com/>)
- Visio Corp.
  - Visio Professional
  - (<http://www.visio.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-caseview
  - Systemasperceivedbytheendusers
- Logicalview
  - Structuraldescription
- Componentview
  - Divisionofwork,dependenciesbetweencodemodules
- Concurrencyview
  - Fordescribingsynchronizationandcommunication aspectsofsystem
- Deploymentview
  - Describinghowsystemisdeployable

---

---

---

---

---

---

---

---

## Diagrams

- Use-CaseDiagram
- ClassDiagram
- ObjectDiagram
- StateDiagram
- SequenceDiagram
- CollaborationDiagram
- ActivityDiagram
- ComponentDiagram
- DeploymentDiagram

---

---

---

---

---

---

---

---

## OtherSoftwareTools

- Automatedtestingtools
  - SegueQualityWorks
- Make/Buildtools
- Metricstools

---

---

---

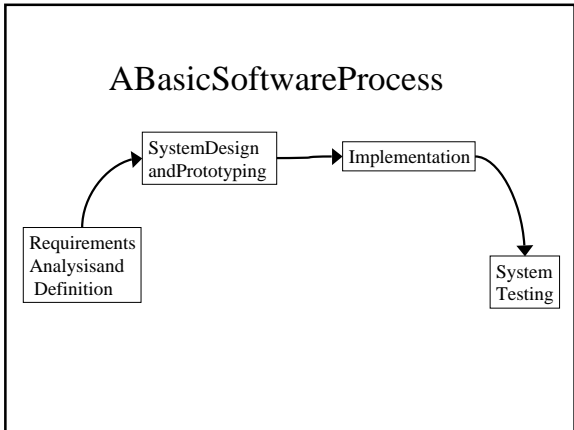
---

---

---

---

---




---

---

---

---

---

---

---

---

- ### 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
- 

---

---

---

---

---

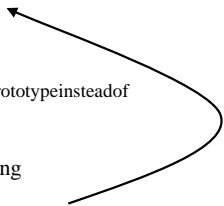
---

---

---

## 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




---

---

---

---

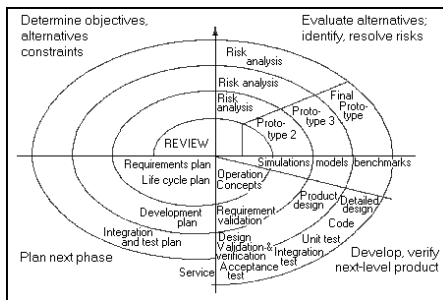
---

---

---

---

## Spiral Model




---

---

---

---

---

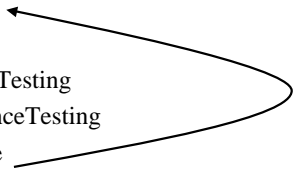
---

---

---

## 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

---

---

---

---

---

---

---

---

## Capability Maturity Model (CMM)

- 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 Level have  
you experienced?

---

---

---

---

---

---

---

---