

**MICHIGAN STATE UNIVERSITY**

## 3. Software Tools I Design & Development

CSE 498, Collaborative Design




Wayne Dyksen  
Brian Loomis  
Department of Computer Science and Engineering  
Michigan State University  
Fall 2004

### S Overview

- Overview of a lifecycle or development process
- Tools used in each phase
  - Design
  - Development
  - Test/Stabilization
  - Deployment

### S Why have a process?

- PRO: Methodology tells you how to do something
  - Seems like a good idea...
  - Carries lessons learned forward so we don't repeat mistakes
  - Makes sure we're thoroughly understanding the problem
- CON: Most people do not have the experience to know which parts of the method help get them to the goal
  - Methodology can lead to "analysis paralysis"
  - Methodology cannot solve fundamental people issues like communication
- What's your methodology? How do you do team projects?



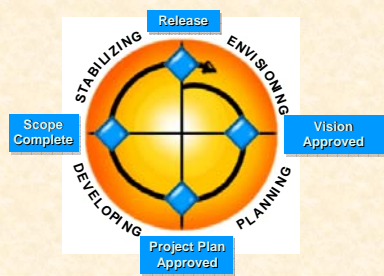
### S Lots of processes exist...

- Mil-Std 2167A, 10000 series...
- Capability Maturity Model (SEI)
- PMBOK (IEEE)
- Extreme programming (XP), agile methods
- "Corporate custom"
  - We'll look at Microsoft Solutions Framework
  - Most methods have similar steps but often call them by different names

### S Process Models

- Life cycle models establish the order for project activities
- Two models are popular
  - The waterfall model
  - The spiral (or rapid application development) model
- MSF combines both
  - Milestone-based process
  - Flexible and iterative process

### S Process Model for Application Development



## S Milestone-Driven Process

- Milestones are review and synchronization points, not freeze points
- Milestones enable the team to assess progress and make mid-course corrections
- The process model uses two sorts of milestones
  - Major milestones
  - Interim milestones
- Achieving a major milestone represents team and customer agreement to proceed
- Deliverables are physical evidence that the team has reached a milestone

0-7

## S Last time...

- We talked about envisioning
  - The vision document
  - Talking with the customer about the scope of the project
  - Requirements analysis
- Now we move into planning
  - Object analysis, modeling
  - Technology mapping
  - Schedule and risk planning

0-8

## S Project Plan Approved Milestone

Signals agreement on

- Project trade-off strategy
- Project risks
- What will be built
- When it will be built
- How it will be built
- Who will build it

0-9

## S Deliverables for Project Plan Approved

Deliverable	Purpose	Owner
Functional specification	Describes what will be built	Program management
Master project plan	Describes how it will be built	Program management
Master project schedule	Describes when it will be built	Program management
Master risk assessment document	Describes any issues in building it	Program management

0-10

## S Functional Specification

**Describing the product feature set in explicit detail**

- Translates project scope into actionable steps toward achieving the vision
- Serves as a contract between the customer and the project team
- Is the basis for building the schedules and project plan

0-11

## S Scope Complete Milestone

Signals agreement on

- The planned feature set
- Whether the planned feature set has been developed
- Baselined materials to support user performance
- The stabilization process, including betas and testing

0-12

### Team Focus During Developing

Role	Focus
Product management	Customer expectations management; communication plan execution; beta planning
Program management	Project tracking; team communication and coordination; beta planning
Development	Feature development; testing
User education	User performance support development and testing; beta planning; product usability testing
Testing	Test specifications, cases, and scripts development; testing
Logistics management	Operational support documentation; beta planning; internal team support

### Suggested Interim Milestones

### Internal Releases

Getting the product to a known state and incrementally building upon it

### Benefits of Internal Releases

- Help the team to break large, complex projects into understandable, more manageable pieces
- Help the team to better manage priorities and risks through sequencing the releases
- Allow the team to correct for variations in the plan
- Allow the team to practice releasing
- Increase the overall quality of the product
- Provide a short-term motivational goal
- Contribute to morale because the team can see progress

### Guidelines for Internal Releases

- Treat internal releases within a single project like versioned releases of a product
- Address high-priority and high-risk features in the earliest possible release
- Define a quality bar to determine when the product has met the standards for internal release
- Make each release as cohesive and yet independent as possible
- Conduct postmortem reviews of each internal release

### Deliverables for Scope Complete

Deliverable	Purpose	Owner
Revised functional specification	Describes product trade-offs and the planned feature set	Program management
Revised master risk assessment document	Describes the issues	Program management
Source code and executables	Baselines the product	Development
User performance support elements	Baselines the materials that support users	User education
Testing elements	Describes how the product was and will be tested	Testing
Revised master project plan and revised master project schedule	Describes what's been done to date and remaining plans	Program management

## S Code Reviews

**Assessing code to improve its quality and to improve the capabilities of the development team**

Some ways to conduct code reviews

- A comprehensive, formal review
- A more casual, peer-based review
- An independent, third-party review

0-19

## S Daily Build

**Building the product in an executable form on a daily basis**

A public daily build is

- A strong indicator that a team is functional
- A way to make the product and its progress visible
- The heartbeat of the development process

**Video**

0-20



Tools we use....



## S Design tools

- Word, Excel (noun analysis, contracts)
- Visio (UML and ORM), Erwin, Rational XDE
- MS Project (schedule)
- Team portal and bug tracker set up

0-22

## S Development tools

- Visual SourceSafe, PVCS, Clearcase (code mgmt)
- NAnt, Ant, Kinook (builds)
- FxCop (code reviews)
- NDoc, HTMLHelp, Robohelp (documentation)

0-23

## S Let's see some tools...

- Time to demo
  - Walkthrough a simple application lifecycle

**Demo**

0-24

## S Summary

- Design
- Development

0-25

## S BACKUP SLIDES

0-26

## S XML Comments

```
class XmlElement
{
  /// <summary>
  /// Returns the attribute with the given name and
  /// namespace</summary>
  /// <param name="name">
  /// The name of the attribute</param>
  /// <param name="ns">
  /// The namespace of the attribute, or null if
  /// the attribute has no namespace</param>
  /// <return>
  /// The attribute value, or null if the attribute
  /// does not exist</return>
  /// <seealso cref="GetAttr(string)"/>
  public string GetAttr(string name, string ns) {
    ...
  }
}
```

0-27