

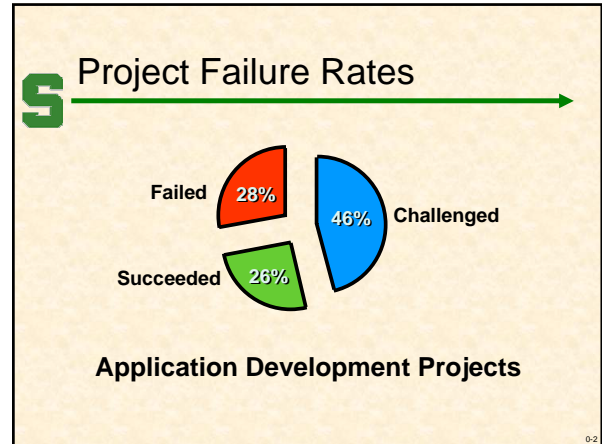
MICHIGAN STATE UNIVERSITY

4. Software Engineering I

CSE 498, Collaborative Design




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



S Root Causes of Failure

- Separation of ...
 - Goal and Function
 - Business and Technology
- Lack of Common
 - Language
 - Process
- Failure to
 - Communicate
 - Act as a Team
- Inflexible Processes



"When projects fail, it's rarely technical."
Jim Johnson, The Standish Group

0-3

S Software Engineering I

Consider Managing..

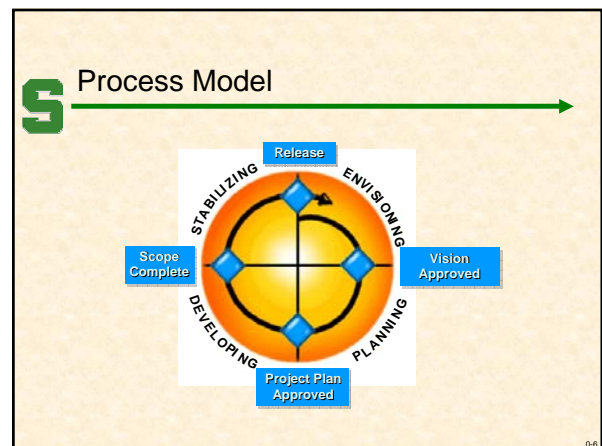
- ☒ The Process
- The Risks
- The Team

0-4

S Software Development Models

- NB: Model "Imposes" Order for Activities
- Two Popular Models
 - Waterfall Model
 - Spiral Model (Rapid Application Development Model)
- Process Model
 - Milestone-Based Process
 - Flexible and Iterative
 - Hybrid of Waterfall and Spiral

0-5



S Milestones

- Used For
 - Review Points
 - Synchronization Points
 - Not Freeze Points
- Enable
 - Assessment of Progress
 - Mid-Course Corrections




0-7

S Types of Milestones

- Major Milestone
 - Met by Tangible Deliverables
 - Acceptance by Team and Customer
 - Establishes Agreement to Proceed
- Interim Milestone
 - Met by Tangible Deliverables
 - Acceptance by Team and Possibly Customer
 - Establishes Ability to Proceed

0-8


S Principles of a Successful Process

- Creating Living Documents 
- Using Versioned Releases 
- Making Project Trade-offs 

0-9

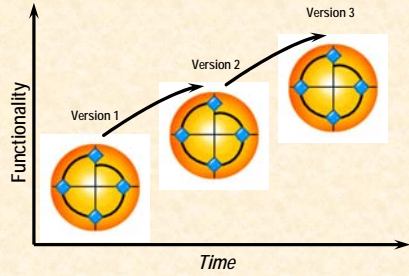
S Creating Living Documents

- How
 - Base Lining As Early As Possible
 - Freezing As Late As Possible
- Why
 - Avoids “Analysis Paralysis”
 - Establishes Structured Change Control Process



0-10

S Using Versioned Releases



0-11

S Versioned Releases: Benefits


- Force Closure On Project Issues
- Set Clear And Motivational Goals With All Team Members
- Manage Uncertainty And Change In Project Scope
- Encourage Continuous And Incremental Feature Delivery
- Enable Shorter Time To Market



0-12

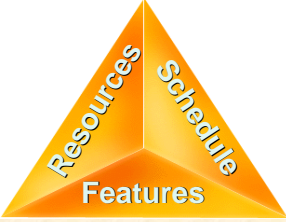
S Versioned Releases: Guidelines

- Adopt A Product Mindset
- Create A Multi-version Release Plan
- Cycle Rapidly To Make Customers And Users Confident Of Your Team's Ability To Ship
- Analyze Each Feature Request For Impact, Feasibility, And Priority
- Deliver Core Functionality First So You Can Build On It
- Stop Doing New Versions When They No Longer Add Business Value




0-13

S Making Project Trade-offs



0-14





S Project Trade-off Matrix



	Optimize	Constrain	Accept
Resources		✓	
Schedule	✓		
Features			✓

0-15


S Principles of Accurate Scheduling

- Bottom-up Estimating 
- Fixed Ship Date Mindset 
- Risk Driven Scheduling 
- Scheduling For An Uncertain Future
 - Adding Buffer Time 
 - Using Interim Milestones
 - Using Discrete Tasks

0-16

S Zero-Defect Mindset

- Team members must understand the required quality level for their work.
- Work is not complete until it reaches that level of quality.
- The zero-defect mindset is embodied in task deliverables and milestones.




Committing to the highest possible level of quality within project constraints

0-17

S Zero-Defect Mindset: Benefits


- Increases accountability for the quality of the product.
- Increases stability of the product.
- Improves schedule predictability.
- Decreases the cost of addressing issues.
- Allows testing to shift focus to quality assurance.
- Rewards quality developers.



0-18

S Zero-Defect Mindset: Guidelines

- Articulate the quality bar for all work performed.
- Reward high-quality work.
- Balance feature quality against the needs of the customers and end users.
- Reinforce the importance of quality through actions, not just words.



0-19

S Zero-Defect Techniques

- Write unit test cases before debugging.
- Assume the code is broken; prove that it isn't.
- Fix bugs before moving on.
- Use competing designs and implementations.
- Assign bugs to other developers.
- Reassess the code in light of bugs.
- Document code.
- Conduct code reviews.
- Perform daily builds.

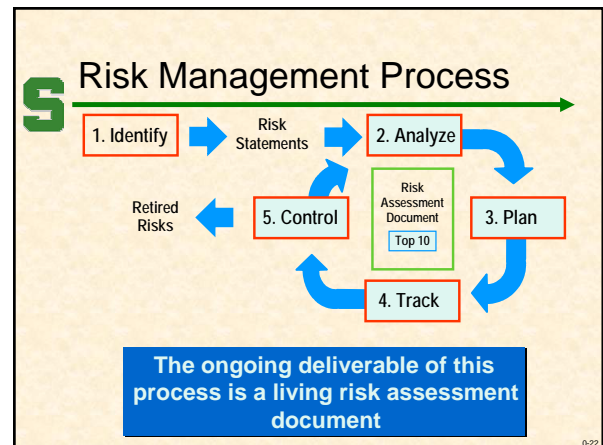
0-20

S Software Engineering I

Consider Managing..

- The Process
- The Risks
- The Team

0-21



S Software Engineering I


Consider Managing..

- The Process
- The Risks
- The Team

0-23

S Pitfalls of Hierarchical Teams

- High Communications Overhead
- Misunderstandings From Indirect Communications
- Unclear Team And Role Goals
- Disengaged Team Members
- High Process Overhead



0-24

S Team Goals for Success


- Satisfied Customers
- Delivery Within Project Constraints
- Delivery To Specifications That Are Based On User Requirements
- Release After Addressing All Known Issues
- Enhanced User Performance
- Smooth Deployment And Ongoing Management



0-25


S Team of Peers

- Is A Team Whose Members Relate As Equals
- Has Specific Roles And Responsibilities For Each Member
- Empowers Individuals In Their Roles
- Holds Members Accountable For The Success Of Their Roles
- Drives Consensus-based Decision-making
- Gives All Team Members A Stake In The Success Of The Project



0-26

S Team Model



0-27

S Product Management Role


- Acts As Customer Advocate To The Team
- Acts As Team Advocate To The Customer
- Drives Shared Project Vision
- Manages Customer Expectations
- Develops, Maintains, And Executes The Business Case
- Drives Feature Identification And Prioritization
- Develops, Maintains, And Executes The Communications Plan



0-28

S Program Management Role

- Drives The Overall Process
- Manages Resource Allocation
- Manages The Project Schedule And Reports Project Status
- Manages The Product Scope And Specification
- Facilitates Team Communication And Negotiation
- Drives Overall Critical Trade-off Decisions



0-29

S Development Role

- Builds And Tests Features To Meet The Specification And Customer Expectations
- Participates In Design
- Estimates Time And Effort To Complete Each Feature
- Serves The Team As A Technology Consultant



0-30

S Testing Role

- Develops Testing Strategy, Plans, And Scripts
- Manages The Build Process
- Conducts Tests To Accurately Determine The Status Of Product Development
- Participates In Setting The Quality Bar

0-31

S User Education Role

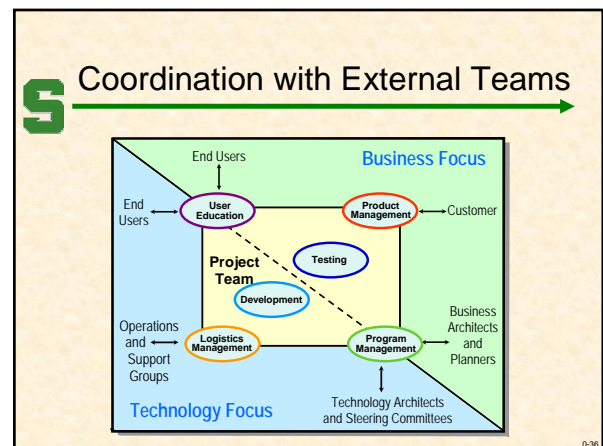
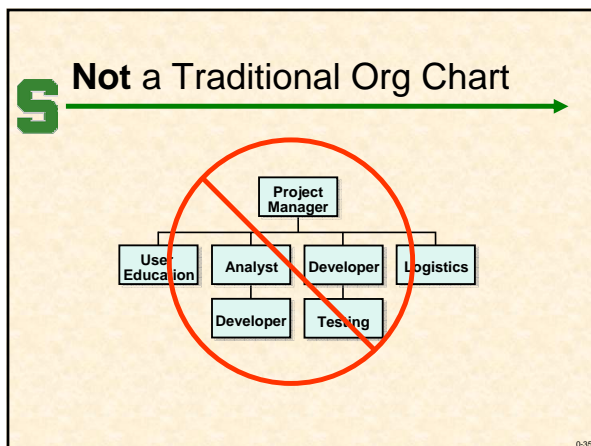
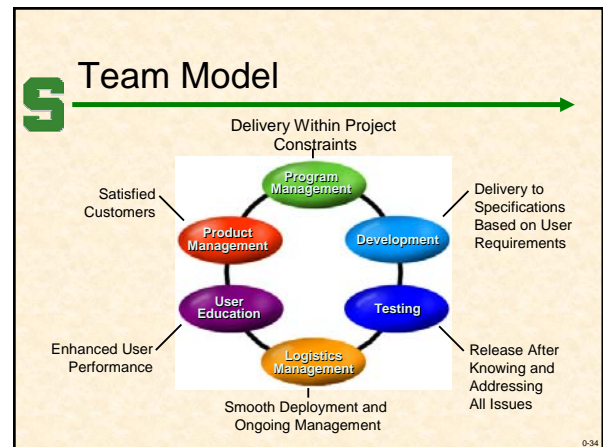
- Acts As Team Advocate To The End User
- Acts As End-user Advocate To The Team
- Participates In Defining User Requirements
- Participates In Designing Features
- Designs And Develops User Support Systems
- Drives The Usability Process

0-32






S Logistics Management Role

- Acts As Team Advocate To Operations
- Acts As Operations Advocate To The Team
- Plans And Manages Product Deployment
- Participates In Design, Focusing On Manageability, Supportability, And Deployability
- Supports The Product During Beta Testing
- Trains Operations And Help Desk Personnel For Product Release

0-33



S Principles of a Successful Team

- Shared Project Vision 
- Product Mindset 
- Zero-defect Mindset 
- Customer-focused Mindset 
- Willingness To Learn 

0-37

S Software Engineering I

Consider Managing..

- The Process
- The Risks
- The Team

0-38

MICHIGAN STATE UNIVERSITY

Software Engineering I

The Big Process Model

CSE 498, Collaborative Design

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

