


MICHIGAN STATE UNIVERSITY

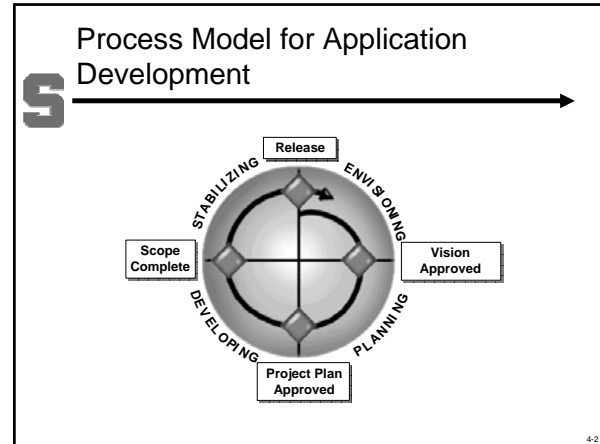
9. Software Engineering II

Process And Tools For Stabilization, Testing, & Release

CSE 498, Collaborative Design



Wayne Dyksen
Brian Loomis
Department of Computer Science and Engineering
Michigan State University
Spring 2005



2 Principles of QA


- Zero-defect mindset
- Code reviews

4-3

Zero-Defect Mindset

Committing to the highest possible level of quality within project constraints


- Team members must understand the required quality level for their work
 - Articulate the quality bar for all work performed – who does this?
- Work is not complete until it reaches that level of quality
- The zero-defect mindset is embodied in
 - Task deliverables
 - Milestones



4-4

Benefits of a Zero-Defect Mindset

- 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



4-5

Techniques for Zero-Defect Development

- Write unit test cases before debugging
- Assume the code is broken, then 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


4-6

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
- Tools plus people...
- Daily build status and source control are evidence of a working process



4-7

S Testing in development

- Coverage testing
 - Attempts to thoroughly test every feature of the product
 - Attempts to thoroughly test the code base of the product
 - Is used primarily during the developing phase
- Usage testing
 - Attempts to successfully complete usage scenarios
 - Attempts to test the product in its expected environment as users might stress it
 - Is used primarily during the stabilizing phase

4-8

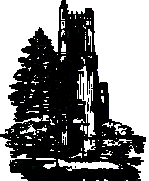
S Types of Testing

- Unit tests
- Functional tests
- Check-in tests
- Build verification tests
- Regression tests
- Configuration tests
- Compatibility tests
- Stress tests
- Performance tests
- Documentation and help file tests
- Alpha and beta tests

4-9



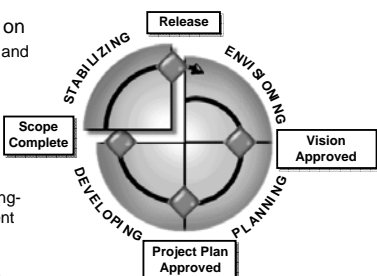
Stabilization



S Release Milestone

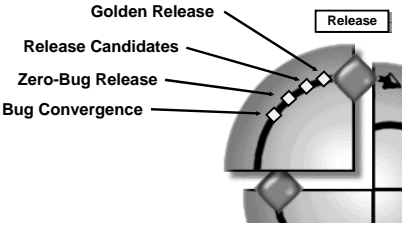
Signals agreement on

- Product stability and resolution of all known bugs
- Customer acceptance of the product
- Transfer of ownership for long-term management and support
- Change in team focus to the next release



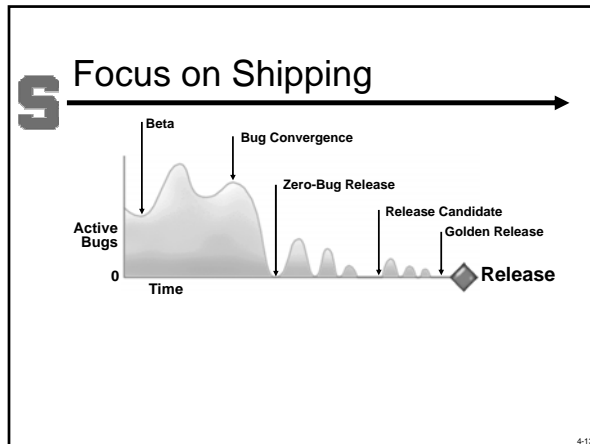
4-11

S Suggested Interim Milestones



Entering the stabilizing phase marks the transition from a schedule-driven focus to a ship-driven focus

4-12



S Beta Testing

Testing of a stabilized product by external end users

- Provides actual end-user usage testing in the expected environment
- Requires greater team effort than alpha testing
- Occurs with different frequency and size depending on a number of factors

4-14

S Bug Triaging

Evaluating and prioritizing bugs to determine their appropriate resolution

- Uses a review committee to prioritize and assign bugs
- Determines what new bug fixing, if any, will be done
- Balances stability against customer needs
- Can result in loss of features for the sake of stability

4-15

S Test and stabilization tools

- VSS, Clearcase, PVCS
- NUnit, NUnitASP, JUnit
- App Center Test
- WinRunner, XRunner
- NDoc, HTMLHelp, Robohelp
- FxCop
- MSI or InstallShield
- CLR Profiler

4-16

S Show me the code!

- Testing and packaging our customer app

4-17