How Microsoft Builds Software

- By Michael A. Cusumano & Richard W. Selby
- Communications of the ACM, June 1997, vol. 40, no. 6
- Microsoft is the world's largest producer of PC software
  - In June 1996
    - 20,500 employees
    - 250 products
  - Windows 95
    - 11 million lines of code
    - 200 designers, programmers and testers
- What development process do they use?

Main Philosophy

- Does not use adopt too many of the structured software-engineering practices
- "scaled-up" a loosely structured small-team style (hacker philosophy?)
  - Small parallel teams of 3 to 8 developers each or
  - Individual programmers
  - Working together as a large team

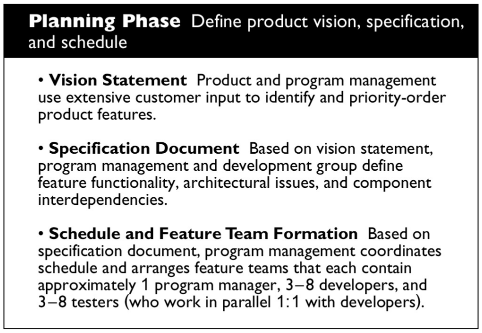
Philosophy

- Each team has the freedom to evolve their design
  - Evolve features and whole products incrementally
  - Occasionally introduce new concepts and technologies
- However
  - Since teams have so much freedom
  - There is a danger that products may become incompatible
  - They synchronize their changes frequently

Synch-and-stabilize

- Terms describing the process
  - "daily-build"
  - "nightly build"
  - "zero defect"
  - "milestone"
- Build
  - Putting together partially completed or finished pieces of the software
  - Goal
    - To determine what works and what doesn't
  - Done by completely recompiling the source code and executing automated tests

Process: Planning



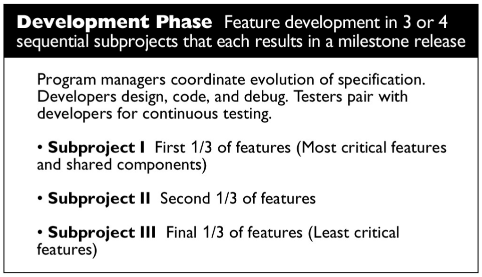
Planning Phase Define product vision, specification, and schedule

• **Vision Statement** Product and program management use extensive customer input to identify and priority-order product features.

• **Specification Document** Based on vision statement, program management and development group define feature functionality, architectural issues, and component interdependencies.

• **Schedule and Feature Team Formation** Based on specification document, program management coordinates schedule and arranges feature teams that each contain approximately 1 program manager, 3–8 developers, and 3–8 testers (who work in parallel 1:1 with developers).

Process: Development



Development Phase Feature development in 3 or 4 sequential subprojects that each results in a milestone release

Program managers coordinate evolution of specification. Developers design, code, and debug. Testers pair with developers for continuous testing.

• **Subproject I** First 1/3 of features (Most critical features and shared components)

• **Subproject II** Second 1/3 of features

• **Subproject III** Final 1/3 of features (Least critical features)

Process: Stabilization

Stabilization Phase Comprehensive internal and external testing, final product stabilization, and ship

Program managers coordinate OEMs and ISVs and monitor customer feedback. Developers perform final debugging and code stabilization. Testers recreate and isolate errors.

- **Internal Testing** Thorough testing of complete product within the company

- **External Testing** Thorough testing of complete product outside the company by "beta" sites, such as OEMs, ISVs, and end users

- **Release preparation** Prepare final release of "golden master" disks and documentation for manufacturing

Milestones in sync-and-stabilize (2-4 months)

Milestone 1 *(first 1/3 features)*
Development (design, coding, prototyping)
Usability Lab
Private Release Testing
Daily Builds
Feature Debugging
Feature Integration
Code Stabilization (no severe bugs)
Buffer time (20%–50%)

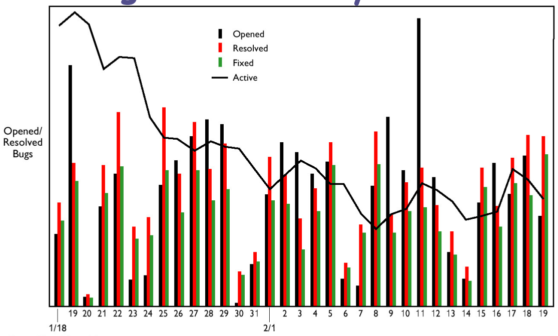
Milestones in sync-and-stabilize (2-4 months)

Milestone 2 *(next 1/3)*
Development
Usability Lab
Private Release Testing
Daily Builds
Feature Debugging
Feature Intergration
Code Stabilization
Buffer Time

Milestones in sync-and-stabilize (2-4 months)

Milestone 3 *(last set)*
Development
Usability Lab
Private Release Testing
Daily Builds
Feature Debugging
Feature Intergration
Feature Complete
Code Complete
Code Stabilization
Buffer Time
Zero Bug Release
Release to Manufacturing

Example: Excel/Graph Bug Data & Daily Builds



Source: Microsoft internal document

Comparing Processes

Synch-and-Stablize	Sequential Development
Product development and testing done in parallel	Separate phases done in sequence
Vision statement and evolving specification	Complete "frozen" specification and detailed design before building the product
Features prioritized and built in 3 or 4 milestone subprojects	Trying to build all pieces of a product simultaneously
Frequent synchronizations (daily builds) and intermediate stabilizations (milestones)	One late and large integration and system test phase at the project's end
"Fixed" release and ship dates and multiple release cycles	Aiming for feature and product "perfection" in each project cycle
Customer feedback continuous in the development process	Feedback primarily after development as inputs for future projects
Product and process design so large teams work like small teams	Working primarily as a large group of individuals in a separate functional department