How Microsoft Builds Software

By M.A. Cusumano and R.W. Selby
Communications of the ACM, Vol. 40, No. 6, 53-61.

"Microsoft has tried to scale up a loosely structured small-team (same might say hacker) style of product development. The object is to get many small parallel teams (three to eight developers each) to work together as a single relatively large team in order to build large products relatively quickly while allowing programmers and teams freedom to evolve their designs and operate autonomously."

"These small teams evolve features and whole products incrementally."

This model might work for Microsoft but not for everybody else.

Frequent Synchronizations and Periodic Stabilizations

“Many companies, including Microsoft, now follow a process that iterates among design, building components, and testing, and also overlaps these phases and contains more interactions with customers during development. Many companies also ship preliminary versions of their products, incrementally adding features or functionality over time in various product releases. “

“In addition, many companies frequently integrate pieces of their products together. Frequent integrations help determine what does and does not work without waiting until the end of the project, which may be several year away.”

Developers have freedom, but they must check-in their code by 5pm.

Developing and Shipping Products

1. **Work in parallel teams but “synch up” and debug frequently**

2. **Always have a product you can ship, with versions for every major platform and market.**

3. **Speak a common language on a single development site.**

4. **Continuously test the product as you build it.**

5. **Use metric data to determine milestone completion and product release.**

Conclusions

1. **Microsoft is distinctive in the sense that it introduced a structured hacker-like approach to software product development that works reasonably well for both small- and large-scale products.**

2. **It is an example how culture and competitive strategy can drive product development and the innovation process.**

3. **Introduces product that are "good enough", rather than waiting until something is "perfect".**

4. **Then improve the products by incrementally evolving their features, and selling upgrades to customers around the world.**

Software Engineering at Microsoft

1. There is no “one model” that every product team at Microsoft uses to create software.

2. Each team determines the best model to achieve their goals: depend on the size and scope of product, market conditions, team size, and prior experience.

3. A new product might be driven by time to market so as to get in the game before there is a category leader.

4. An established product might need to be very innovative to unseat a leading competitor or to stay ahead of the pack.

5. Each situation requires a different approach to scoping, engineering, and shipping of the product.

6. Finally, even with the need for variation, a general practice has been adopted, while allowing for significant experimentation and innovation in the process.

Redefining the Software Life Cycle:

- **Waterfall**
- **Spiral**
- **Incremental**
- **Y Model**
- **Microsoft Model**
- **Agile (with pair programming)**

Towards CASE (Computer-Aided Software Engineering) Environment.