MICROSOFT SOFTWARE DEVELOPMENT

Many consider Bill Gates a techo-nerd, who got lucky. But he is one of the few CEOs of a major company, who not only understands the technology, he also understands business.

Companies often fail when the technology leaders who start to company fail to understand how to manage a business.

Others fail when they bring in management who thinks they understand business, but don't understand the technology.

Microsoft has been amazingly successful at huge growth rates yet still retaining the flexibility of a small company.

Microsoft Secrets book

Microsoft is successful because of:

- Having a very astute leader - Gates
- Apple and IBM not understanding their own technology. They gave away their leadership.
- Being at the right time at the right place - some luck involved here.
- Not being complacent in any one technology area

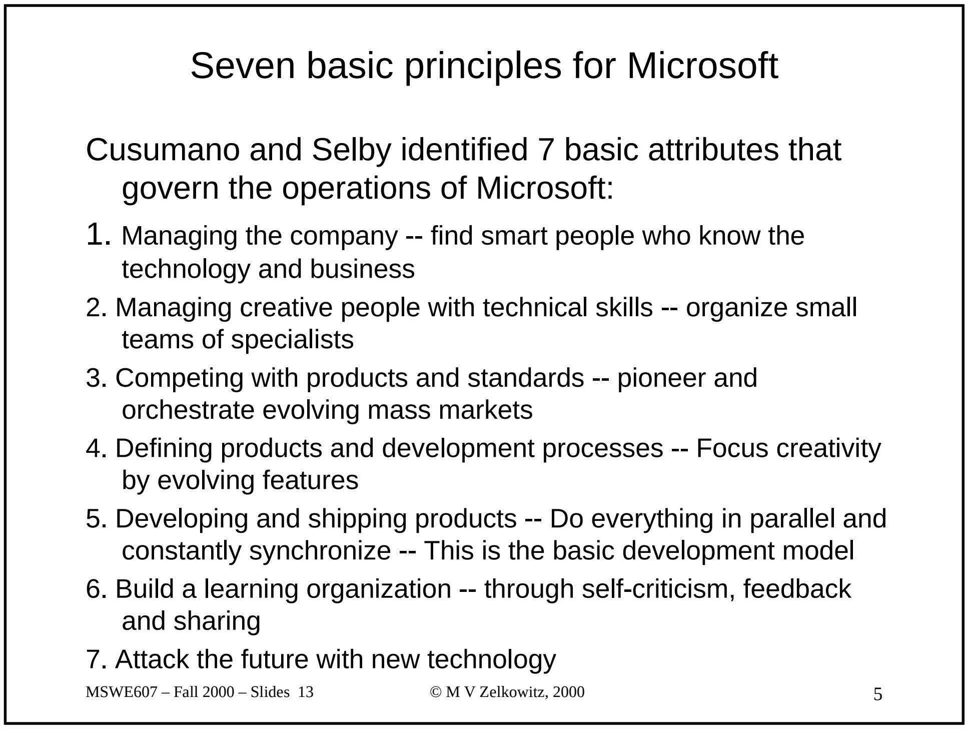
This is based on "Microsoft Secrets" by Selby and Cusumano. So it is based on only one set of opinions.

A Short History

- Bill Henry Gates born in 1955 in Seattle. Father was a lawyer and mother was a teacher.
- In 1968 at a private school he teamed up with Paul Allen to learn programming in BASIC. In 1972 formed Traf-O-Data to build a small computer to record and analyze traffic data.
- In 1973 went to Harvard. In 1974, Allen left University of Washington to work at Honeywell near Boston.
- In 1974 MITS announced the Altair computer. Gates and Allen left Boston for Albuquerque to be near MITS. Built BASIC for Altair. Formed Microsoft with Gates owning 60% - Allen 40%.
- Gates realized money was in software; hardware was changing so rapidly.

A short history - II

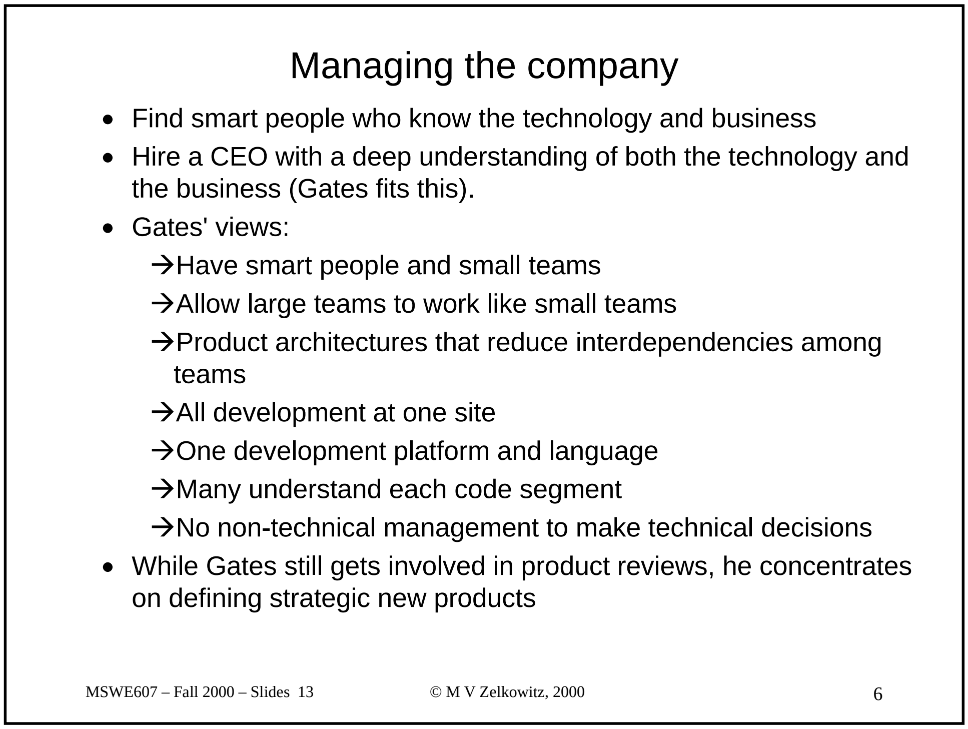
- Moved to Bellevue, WA in 1979.
- In 1980 IBM wanted BASIC for PC. Also wanted an operating system. Gates promised them one without having it. Obtained CP/M and modifies it as PC-DOS 1.0
- In 1982 Microsoft introduces GW-BASIC, Multiplan, and Word in 1983.
- In 1983 Windows announced
- Microsoft was off and running: Windows 3.0 in 1990, Windows 95 and NT in 1995, Constant upgrades to Office suite (Powerpoint, Word, Excel). The rest, as they say, is history.



Seven basic principles for Microsoft

Cusumano and Selby identified 7 basic attributes that govern the operations of Microsoft:

1. Managing the company -- find smart people who know the technology and business
2. Managing creative people with technical skills -- organize small teams of specialists
3. Competing with products and standards -- pioneer and orchestrate evolving mass markets
4. Defining products and development processes -- Focus creativity by evolving features
5. Developing and shipping products -- Do everything in parallel and constantly synchronize -- This is the basic development model
6. Build a learning organization -- through self-criticism, feedback and sharing
7. Attack the future with new technology



Managing the company

- Find smart people who know the technology and business
- Hire a CEO with a deep understanding of both the technology and the business (Gates fits this).
- Gates' views:
  - →Have smart people and small teams
  - →Allow large teams to work like small teams
  - →Product architectures that reduce interdependencies among teams
  - →All development at one site
  - →One development platform and language
  - →Many understand each code segment
  - →No non-technical management to make technical decisions
- While Gates still gets involved in product reviews, he concentrates on defining strategic new products

Managing the company- II

- Organize flexibly around and across product markets and business functions
- Hire the smartest managers you can find that understand the technology
  - → Microsoft sometimes compared to Xerox PARC of 1970s
  - → Xerox PARC good research; but part of a copier company. Never understood what they had.
- Hire the best and smartest technical staff
  - → Advantage: HIGH productivity of lowest staff; But staff burnout
  - → Only hire a few per cent of applicants
  - → Downside: "they know everything." Often "reinvent the wheel," not very knowledgeable about software engineering, Microsoft has a low profile in the software engineering world

Managing creative people

- Organize small teams of specialists
- Separate technical and management career ladders.
- Prefer to hire developers right out of college to teach them the "Microsoft way."
  - →Hire about half the people they think they need
  - →Go for "Type A."
  - →Result: Long hours and burnout. About 10% leave each year for the first 5 years, then seem to stay a long time
- During early days, technical development was often chaotic,
- By early 1980s developed distinct functional skills with overlapping boundaries.
  - →Minimize bureaucracy.
  - →Lower management has enough independence from top management to define new skill sets and hire new people.
  - →Create small groups that share tasks.
  - →Learn from more experienced staff; not a large dependence on training programs.

Managing creative people - II

- Major culture change -- Why addition of testing staff:
  - →Developers do not produce perfect code
  - →Need someone detached from code and specs to provide unbiased perspective
  - →Cheaper to fix code in development then let users find it in field
- Testing perspectives (Remember PBR?):
  - User -- How will users react to product?
  - International -- Will formats work everywhere?
  - Hardware -- Will it work on all platforms?
  - Software -- Is product compatible with others?
  - Compliance with specifications?
  - Product stability? -- Measurable metrics (e.g., bug rates, bug severity); qualitative ("are we ready?")

Pioneer and orchestrate mass markets

- Enter every PC mass market:
  - → Languages (originally)- BASIC, FORTRAN, C, C++, Java, …
  - → Systems - MS-DOS, Windows 3.0, Windows 95, NT, CE …
  - → Applications - Word, Excel, Powerpoint, Access, …
  - → Internet - MSN, Internet Explorer, …
  - → Home products - MS Money, Quicken, Works, Encarta, Bob ☹, …
- Have products initially "good enough"
- To preempt competition, improve products incrementally; replace products with more advanced products.
  - → Add to revenue stream as upgrade revenue
  - → Keep the competition off balance with new features
  - → Make older products obsolete

Pioneer and orchestrate mass markets - II

- Increase sales with OEM vendors: Install system at extremely low cost in every machine built
  - → Huge initial market for operating systems software
  - → Natural base to sell application software
  - → Application developers knew APIs months before competition
  - → Basis for original Dept. of Justice investigation of Microsoft

Products and development processes

- Focus creativity by evolving features
- Have creative people working for company, but direct their creativity-- Think of new features to add but limit resources to build those features
- Projects structured via "milestones" and "vision statements" to guide teams, but not set everything in advance:
  - → Developers generally set their own estimated time (and try to adhere to it); but this can lead to shipping date disasters
  - → 12 months development consists of 3-4 milestone product releases
  - → 2-4 months of buffer time for each 3-4 major milestone
  - → 3-8 months of stabilization and 6 weeks of final preparation for release
  - → About 20% of schedule set aside for reworking bad code
  - → About 2/3 time in development and 1/3 time in stabilization
  - → Stabilization - Testing with no introduction of new features

Developing and shipping products

- Product architecture
  - → Flexible design to allow for changes
  - → Layered architecture
  - → Little documentation; code is the one document
- Do everything in parallel and constantly synchronize -- This is the basic development model: "synch and stabilize" model
  - → Waterfall model too inflexible for rapid development
  - → Allow talented staff to "play," but within the confines of a defined structured and concurrent development model
  - → Grow rather than design products
- All code developed on same platform ("WinTel") as target machines
  - → Use standard structure (called "Hungarian") - standard naming conventions
  - → Use own tools for development

Developing and shipping products- II

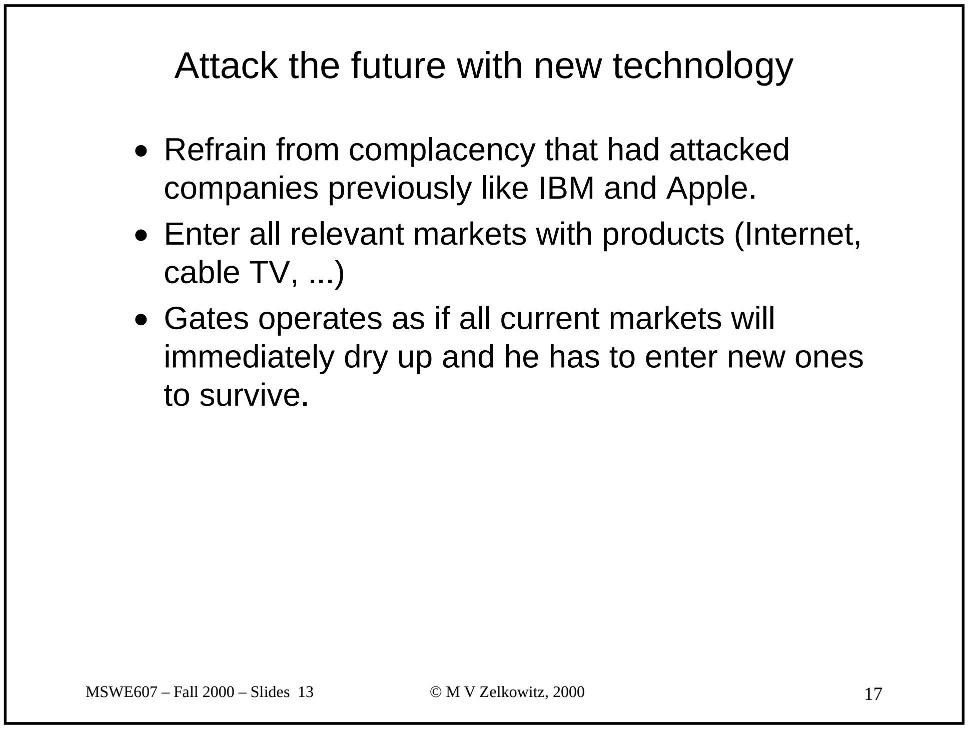
- Code must be checked in at either 2pm or 5pm.
  - → Always have a stable product that can be "shipped."
  - → Product builds done daily; Builder stays until build is done
  - → Any failure results in developer being immediately responsible for fixing problem; Also in charge of build until the next person "hoses" build -- pressure not to fail.
  - → Related to Japanese where production lines halted if any problems
  - → Works well for mature products (e.g., Word, Excel); Doesn't seem to be as effective with systems software (e.g., Windows 95, 98, NT)
- Extensive beta testing
  - → Microsoft very freely adds users to beta testing group to find errors
  - → Moving to a more rigorous testing staff (1850 developers, 1850 testers -- July, 1995)
  - → Buggy products in mid-1980s forced Microsoft to address its quality control problems

Products and development processes - III

- All development work at a single site. Makes communication among developers easier
- Project status reports:
  - → Sent monthly to Gates and other executives
  - → Short and standard format
  - → Gates looks for: deletion of features or schedule slippage
- Program Reviews:
  - → Every 3 months for each product
  - → Attended by Gates and senior executives
  - → 1-2 from program management, software development, software testing, product management, user education
- Code reviews:
  - → Limited number of formal reviews
  - → Not sure of effectiveness due to short product times and frequent changes.

Build a learning organization

- Companies filled with smart people can degenerate into a collection of arrogant and independent individuals
  - →Early view of Microsoft
  - →Company trying to get away from that model
  - →Self-criticism, feedback and sharing
- Postmortems reports now fairly standard
- Process audits - study a project to determine what staff was doing- 1 week
- Retreats - at least 1/year for key company members
- "Eating your own dog food" - use your product as the first user
- Promote sharing and linkage across product groups



Attack the future with new technology

- Refrain from complacency that had attacked companies previously like IBM and Apple.
- Enter all relevant markets with products (Internet, cable TV, ...)
- Gates operates as if all current markets will immediately dry up and he has to enter new ones to survive.