

Successful development of technology at microsoft

[Business](#), [Company](#)



The success of Microsoft can be viewed from a number of perspectives. This ranges from its managerial and organizational position to the development and processing of its products and services for its customers. This implies looking at how Microsoft invents new products with substantial utility to the customers, strategies implemented by the company to make different groups and divisions to work together in harmony to achieve mega success.

Having the best brains in the organization is not just enough to bring out the best software and the best products for the market. There must be effective managerial skills; these would harness the potentials the best brains in technology improvement, retain their autonomy just like any other highly qualified person would do, yet still work together to give the company the best. This paper evaluates the secrets of Microsoft's success in developing superb technology and remaining on top of the market.

Since it is very difficult to maintain the best brains within an organization without having them move out to practice their skills on the private basis, the paper tries to evaluate how the company maintains these best software developers and their work in harmony to bring out the best for the company. Microsoft currently is the world's largest manufacturer of PC software, and having annual revenue of \$8.7 billion.

It is probably the only PC Company that has handled more PC software projects than any other company in the world. This task is Herculean as there is a lot of complexity involved in the process, for example the development of Windows 95 that has more than 11 million lines of code and required an expansion team of more than 200 programmers as well as

testers. This is one of the greatest challenge that most of the rivals to Microsoft are unable to do and thus the difficult in competing [Reimer, 2008]

Their philosophy on product development and improvement for the last few years has been to develop its roots as a highly flexible, entrepreneurial company that does not to take up too many of the structured software-engineering practices that are commonly practiced and promoted by organizations such as the International Standards Organization and Software Engineering Institute. Microsoft strategy is developing a loosely structured small-team that some people might refer to as hacker style of product development.

Their objective is to get many small as well as parallel teams that are composed of three to eight developers each, or individual programmers who eventually work together as one team in order to build large pile of products relatively fast while still allowing individual programmers and the team members their freedom to evolve their designs and even operate nearly autonomously [Hardmeier, 2008]. The small parallel teams develop features and whole products freely while at the same time frequently introducing new technologies concepts.

However, since the developers are free to innovate while working with Microsoft, they are required to synchronize their changes frequently so that the product components work in harmony. In this regard, there are several lessons to be learned from how Microsoft develops its software products; some apply to other organizations, and some may not. Software developers as well as their managers from other organizations can discover useful ideas after careful consideration of such factors as their company's goals,
<https://assignbuster.com/successful-development-of-technology-at-microsoft/>

strategies, marketing resource constraints, and development culture and software reliability requirements [Reimer, 2008].

They can also learn to frequently synchronize and carry out periodic stabilizations of their products. Without this approach implemented by Microsoft, it would probably never have been able to design, develop, and ship the products it offers now and is planning to offer in the future [Reimer, 2008]. Teams of programmers at Microsoft and testers usually synchronize and periodically stabilize the changes they make to their products in progress, thus yielding products such as Publisher, Excel, Office, Windows 95, Word, Works, and Windows NT etc.

Since the mid-1980s, Microsoft as well as other PC software companies has recognized that they build software products in response to quality. Many of them have also found it important to have larger teams in order to come up with PC software products that consist of hundreds of thousands or even millions of lines of source code hence requiring hundreds of people to build and then test them over a certain period [Hardmeier, 2008]. The technology development approach must have a mechanism that allows developers to test their product with their customers and also refine their designs during the development process.

In most industries, many of the companies are now using prototyping and multiple cycles of concurrent design, build, and testing activities aimed at controlling interactions and also make fundamental changes in their product development. In the software industry, researchers and their managers have talked about “ interactive enhancement,” a “ spiral model” for iteration in

the phases of the project development, and the “ concurrent development” of multiple phases and the activities for the past 20 years.

Unfortunately most companies are reluctant to formally adopt these recommendations. Since users’ needs for most software are very difficult to understand and the changes in hardware and software technologies are very frequent, it is not wise to design a software system completely in advance [Lemo, 2008]. Microsoft teams start the process of product development at the point of creating a “ vision statement”, and then define the goals for the new product and the user activities that need to be supported by the new product features.

Then, the product managers or the marketing specialists take charge while the consulting program managers who specialize in writing functional specifications of the product are also incorporated. In consultation with developers, program managers, then write a functional specification that outlines the product features with organized schedules and the staffing allocations; the initial specification document rarely covers all the details of each feature. During product development, team members revise the features set and details, and learn about the new product.

Experience at Microsoft has shown that the feature set in a specification document may sometime change by 30% or even more. Project managers are obliged to divide the product and the entire project into different parts; they also divide the project schedule into three or four milestone junctures called sequential sub projects meant to represent the completion points for major portions of the new product in each of the milestone subproject, all the

feature teams are expected to go through a complete cycle of development, feature integration, testing, as well as fixing problems.

Throughout an entire project, they (feature teams) are required to synchronize their work in the process of building the product as well as finding and fixing errors on a daily, weekly basis. The program developer is expected to fix detectable errors at the end of a milestone subproject. These error correction is a way of stabilizing the product and thus enable the product developing team to have a clear understanding of the parts of the product that have been successfully completed. After this achievement, the development team usually proceeds to the next milestone, [Hardmeier, 2008].

One of the most important things in high-tech companies is employing creative people, but directing their creativity is supreme. This is the task of the managers. Microsoft has a strategy for tackling this problem; it structures projects into subprojects that contain priority-ordered features; and the buffer time that is within each subproject gives the developers adequate time to respond to the changes, unexpected delays and difficulties. Microsoft projects usually use the vision statements as well as outline specifications rather than a detailed design and the complete product specifications before coding.

This is done because the teams have realized that they cannot determine, in advance, what the developers need to build a good product. This approach gives the developers and program managers enough room to adapt to change, the unforeseen competitive opportunities and even threats. This is particularly related to the applications products, since development teams

<https://assignbuster.com/successful-development-of-technology-at-microsoft/>

try in many instances to devise features that copy directly to the activities that typical customers perform. Thus the teams need to have continual observation and testing with the users during development [Lemos, 2008].

Most of the product designs have modular architectures that allow team members to add or combine features in a predictable straightforward manner. In addition, the presence of managers allows team members to have their own schedules, but this is only after the developers have critically analyzed tasks in detail. Managers are then mandated to “fix” project resources by limiting the number of experts they allocate to each of the project; they also limit the time spent on projects, especially for the applications like Office and multimedia products, so that team members can delete features if they are far from reaching deadlines.

But it should be noted that cutting features in order to save schedule, time has not always been possible with the operating systems projects. In these kinds of projects, reliability is of more importance than the features. Most of the features in the operating system are closely coupled and thus cannot be easily deleted individually. The daily building of processes at Microsoft has several steps. First, to develop a feature for a product, developers check out private copies of source code files which are located at a centralized master version of the source code.

They then implement their individual features by making necessary changes to their private copies of the source code files. The developers are then mandated to create a private build of the product that contains the new features and test it. Changes from the developers' private copies are then

<https://assignbuster.com/successful-development-of-technology-at-microsoft/>

checked from the source code files to the main master version of the source code. This check-in process has an automated regression test to ensure that changes to the source code files do not create errors in other parts of the product.

This check-in by the developers back to the master copy is done at least twice a week but in some instances they may check it daily [Lemos, 2008]. Irrespective of the way the individual developers check in their changes to the main source code, a designated developer, often called the project build master, generates a complete build of the product on a daily basis by using the master version of the source code. Product teams at Microsoft also test features as they continuously build them from various perspectives. This may include bringing in customers from the street to try prototype in a Microsoft usability lab.

Also most of the Microsoft teams work from a single physical site with common development languages (primarily C and C++), standardized common coding styles and development tools. This common language and common site and tools help team members, debate, communicate, resolve problems face to face and design ideas. A small set of quantitative metrics is used in the project teams to help guide decisions in order to complete a project and also when to ship a new or upgraded product to market [Lemos, 2008]. Conclusion There are important lessons to be learnt from Microsoft especially for organizations and managers in various industries.

The synch-and-stabilize approach that is used at Microsoft is particularly suited to fast-evolving markets that have complex systems products, competition around fast-evolving product features as well as defacto high

<https://assignbuster.com/successful-development-of-technology-at-microsoft/>

technical standards and the short life cycles. Of particular interest and important lesson is how Microsoft coordinates the work of a large team of professionals thus successfully building many interdependent parts although continually changing and requiring a constant high level of communication, coordination is still maintained.

Indeed, one should appreciate that it is difficult to ensure such communication and the effective coordination there, and at the same time still allow engineers, designers and marketing people enjoy the freedom to be creative. Achieving this sought of balance is inevitably the central dilemma that the managers of product development usually face—in PC software and in many other industries.

REFERENCE:

Jeremy Reimer. Microsoft set to launch new marketing campaign. Ars Technica. Retrieved on 28th March 2008.

Sandi Hardmeier. Microsoft - The History of Internet Explorer. Microsoft. Retrieved on 28th March 2008

Robert Lemos. Trust or treachery? Security technologies could backfire against consumers. CNet news. com. Retrieved on 28th March 2008