Niklaus Wirth and Bill Gates –

two software entrepreneurs

by Dr Peer Ederer, Strategy Academy Rotterdam, 2004

On November 12th, 2003 a new star had risen in the sky of the ICT industry: Swiss based Esmertec AG. Esmertec is a young software company that achieved the almost impossible in these days: a third round financing of € 23 million to continue explosive growth. It was the largest software VC funding since 18 months and the industry took notice.

The money rewarded a remarkable, but long in the waiting, success story. Within less than 24 months of entering the mobile phone market, the company's software product will be featured on about 120 million mobile phones in 2004. That equals about a 25% world market share for all mobile phones produced and displaces competitors such as Japanese Aplix who have had years of a head start. But that shall only be the beginning, with the newly secured funds, already profitable Esmertec plans to score several more of such successes in the near future.

Fifty three years ago…

The story begins in the year 1950, when Konrad Zuses famed Z4 computing machine was lent for 5 years to Eduard Stiefel from the Swiss Federal Institute of Technology in Zurich, better known as "ETH". This transaction was not only Zuse's first commercial success with his computing technology, it also made Switzerland the only one of two countries with a computer at a university. It thus marked the beginning of the long and difficult joint march of computer technology trying to become a science as well as trying to become a commercial success.

Zuse was not only the world's first computer hardware inventor, he had also pioneered what was later to become known as "structured computer programming" by conceiving "Plankalkül", a mathematically driven algorithmic language that was an only partially finished set of instructions for operating his Z4. It was this scientific mathematical approach that underpinned the conception for the ALGOL 60 programming language that a team of international professors had produced in the course of the 1950ies. ALGOL 60 was

the scientists' alternative answer to FORTRAN, developed since 1953 by and for IBM for commercial purposes to operate its 704 line of computers. FORTRAN (*Form*ula *Trans*lating) has from its very beginnings to this day been an immensely difficult and therefore mistake-prone programming language to learn and work with, but it was ready for shipment in 1957. From then on FORTRAN accompanied IBM's commercial domination of the worldwide computing industry until well into the 1980ies.

The rise of FORTRAN and COBOL

The scientists' invention of ALGOL 60 (*Algo*rithmic *L*anguage) on the other hand came out in 1960. Not only did it come late, it also had a number of serious shortcomings severely limiting its practical usage. While a dedicated scientific user community emerged in Europe for ALGOL 60, it remained virtually unknown in the all-important American market. This was galling, especially since the more engineering-oriented FORTRAN did leave an opening for another programming language to flourish, which was the fast growing market of business data processing. The gap was soon filled by COBOL (*Co*mmon *B*usiness *O*riented *L*anguage), which came out of an American Defense Department sponsored project started in 1959. COBOL never added anything or even took much from the advanced sciences of computer programming, but just as FORTRAN, COBOL remains one of the most used languages due to installed legacy systems. And just as FORTRAN it is difficult to master and prone to mistakes. (As an historical anecdote, the key inventor of COBOL, Grace Hopper, also coined the term "bug", when she discovered that a moth had settled between one of the coils of her computer relays, thus "bugging the system").

During the years of the ALGOL development in the late 50ies, a young Swiss scientist worked towards his diploma at ETH: Niklaus Wirth. In 1959 he moves to Berkeley, where in 1963 he earns his PHD. He becomes Assistant Professor in neighbouring Stanford University and part of an international committee to develop an improved ALGOL X. In 1965, together with C.A.R. Hoare, another committee member, Wirth becomes the core creator of ALGOL-W, which had surmounted most of the shortcomings of ALGOL 60.

World of programming splits apart in 1968

But by 1966, the ALGOL X committees took a sudden and history defining different turn towards designing software. The committee members strove to build an encompassing and very powerful language that could solve all problems known to computing at the time. The result was ALGOL 68, released in 1968, which ended up being so complicated and difficult, that it was never implemented anywhere. Hoare and Wirth had distanced themselves from this release already earlier, with Hoare observing frustratedly:

"There seem to be two ways of programming: either to make the design so simple, that there will obviously be no mistakes and shortcomings, or to make the design so complicated, that no mistakes and shortcomings will be obvious."

From thereon, these two approaches to software programming had split the world of software into two camps, the simple camp and the complex camp.

The protagonist of the simple camp would become Niklaus Wirth. He had in the meanwhile been offered a professorship at ETH, which he took in 1967 and where he remained until retirement in 1999. Based on ALGOL-W, he authored the programming language PASCAL, which was released in 1971, and for which he became famous. PASCAL was an easy language to learn and was at the same time very powerful. It was quickly adopted around the world as the standard software to use for educating computer programmers. In a number of its variations, PASCAL would later power and inspire much of the Apple world, and through ADA, drive the American military.

However, due to various technical limitations inherent in the logic of the language, PASCAL had three fatal shortcomings:

a) it was virtually impossible to impose a rigorous binding standard of PASCAL, thus creating scores of dialects which were not compatible with each other
b) worse, it allowed only very limited interaction with input/output devices, thus not making enough room for the fancy hardware monitors and control devices which came onto the market in the 70ies and 80ies in the mini and micro computing arena
c) worst: it did not allow programmers to have "fun"

The last point was crucial, because programmers liked to invent things and be creative. Moreover, programmers had a penchant towards "hacking away", rather than brooding over difficult mathematical axioms. The logic of PASCAL, based on rigorous mathematical science, instead tried to prevent exactly that. It was this hacking away, being creative and being inventive which was the continuous source of all mistakes in software. These two opposing perspectives are represented by the following two quotes:

Rys McCusker: "When I was finally exposed to C, I recognized that Pascal was an annoying bondage and discipline language intended to stop you from having any fun."

Edgar Dijkstra: "The quality of a software program is reverse proportional to the number of goto statements in it."

Dijkstra himself was a renowned computer scientist, who was Professor at Eindhoven University until 1984 and then at University of Texas, and who coined in 1972 the term "structured programming". To most computer programmers working without "goto" instructions would seem quite baffling. But Dijkstra and others proved that through the use of mathematics it would be possible.

The rise of C

The complex camp of software programming was bred in the famous Bell Laboratories in New Jersey in the early 70ies and accompanied the rise of UNIX systems. The marriage of grammar between a recently developed compiler language B and the rich fountain of features embodied in ALGOL 68 led to the flourishing family of C languages. C would later be inspired by SIMULA, itself also inspired by ALGOL, to become C++. Ultimately also JAVA and LINUX sprang from this branch.

Besides hundreds of specialized programming languages such as IBM's APL, LISP, PROLOG, ABAP of SAP fame etc. the C family (incl. JAVA) would eventually become the most widespread language in use. PASCAL instead has been considered by most to be the scientifically most advanced language and the dominant teaching instrument for computer engineers, but considered useless for any real world applications of value. For young programmers equipped with UNIX machines or the Personal Computers, C became the language of choice to express themselves. Never mind, that the complexity and richness of these languages led to precisely the conditions of permanently unstable, unreliable and expensive to maintain computer systems that Hoare and Wirth foresaw when they departed in frustration from ALGOL 68. The lure of C, C++, JAVA and LINUX to do fantastic things with relatively limited programming or mathematical know-how proved too irresistible to the trade.

The rise of BASIC

But besides scientific PASCAL and fashionable C, ALGOL spawned also a third branch: BASIC and PL/1. BASIC (*B*eginners *A*ll-purpose *S*ymbolic *I*nstruction *C*ode) was conceived in 1964 as an ALGOL and FORTRAN mixed language that could be easily understood by lay people to program their computing machines. It was kept deliberately simple and therefore 10 years later became the perfect candidate for unsophisticated hobby programmers to clank away on their personal 8 bit processor computers based on Intel's 8008 and 8080 chips. In 1975 a Seattle based company called Microsoft takes the public domain BASIC program and repackages it as a *soft*ware program for the just released *Micro* Instrumentation Telemetry Systems (MITS) Altair PC, the first micro computer for the mass market.

In an unrelated development, Gary Kildall 1976 constructs an operating system concept for these hobbyist microcomputers, based on the concepts of the PL/1 language, and calls it CP/M. One part of this operating system is called BASIC Disc Operating System (B-DOS). CP/M is by far the market leader for microcomputer software. In 1979 another Seattle

based company, Seattle Computer Products (SCP), copycats the B-DOS and CP/M system and calls it Q-DOS, for Quick and Dirty DOS. SCP needed Q-DOS for running a computer it developed based on the newest Intel 16 bit Chip 8086. The chip had just come on the market and there was no CP/M version for it available yet.

In 1980 IBM wants to enter the PC market and needs BASIC programs to run on their computers. After failing to reach an agreement with Gary Kildall on CP/M (which would have been the only viable product in the market) IBM turns to Microsoft with the question of whether they could supply an operating system. Microsoft said yes and signed the contract. After signing the contract with IBM, Microsoft bought Q-DOS from Seattle Computers and handed it through to IBM. These early foundations of MS-DOS and BASIC, (later developed into Visual BASIC), became the cornerstones of the vast Microsoft world of personal computing.

Despite their various shortcomings from a scientific point of view, both the C family and the BASIC world were very successful in commercialised software products. The shortcoming of C is that by conception it is a complex language that is highly prone to errors. Yet programmers have always loved C languages for their power and freedom to hack with it. The shortcoming of BASIC was the opposite. It was not simple, but simplistic, meant only for untrained beginners, not for supporting multi-trillion industries.

Yet, what helped both of these languages was the extremely rapid development of hardware capacity throughout the 80ies and 90ies. Sloppy, undisciplined or faulty programming could be compensated to some degree by ever more capable hardware. Whether a program allocated too much memory, misused disc storage or burdened the processor, the problem could be solved with the next hardware release. There was no real pressure to economize with these resources. Both language families allowed scores of self-taught or even educated programmers to enter the PC frenzy and try their luck. Soon enough, the sheer momentum of usage meant that these two languages would dominate software development.

The rise of Microsoft

One entrepreneur understood better than anybody else of how to ride that popularity for his company's success: Bill Gates. Microsoft onsold Q-DOS on a non-esclusive basis to IBM for $186.000, even though it was an unfinished, copyright-infringing and error-prone product, that for years would lag behind the performance of Gary Kildalls CP/M. Microsoft paid at first $10,000 to SCP for its Q-DOS plus a license fee per customer, later to be changed into a one-time lumpsum of just $50.000. IBM later paid $800.000 to Gary Kildall, after they noticed that the Microsoft product was only a copycat version.

With the fourth version of MS DOS 3.1 a few years later, the product became technically on par with the original and useable. But technical performance was not the most important purchase criterium. Far more important, Gates had convinced the producer of Lotus 1-2-3, to develop its spreadsheet calculation program only for MS DOS, not for CP/M. Since 1-2-3 was the first commercial killer application for the PC, MS DOS became market leader.

In its product development, Microsoft copied a long-successful strategy from its symbiotic partner IBM, by preempting rival technology through premature product announcements. Already in early 1983, Microsoft presented its "Interface" manager to mimic a graphical user interface. It was a fake product with no functioning code behind it, but Gates knew that the GUI would be necessary to spread PC's wide and broad. In late 1983 "Windows" was announced, to be followed by real shipment of Windows 1.0 whole two years later in 1985. However, it again took until shipment of version Windows 3.1 no earlier than 1992, that the product became useable and be technically somewhat on par with the rivals (Apple at the time). Also in 1992, Microsoft releases its first real network product, Windows NT 3.1, which is based on IBMs hapless OS/2. The first release of NT is called 3.1. in order "to avoid confusion".

In 1995, Microsoft reveals Windows 95 amid sensational hype, because it would be a 32-bit software finally leaving the clumsy world of DOS behind. When Intel then shipped actual 32 chips later in the year, it was revealed that Windows 95 was in truth still a 16 bit program, that could not work well with 32 bit chips. By 1998, Intel developed its Pentium II line which was specifically designed around Windows 16 bit shortcomings. The real reason for introducing Windows 95 was to integrate DOS with the graphical interface, so that "Wintel" computers would not be compatible anymore with any of the rivalling DOS or even older MS DOS machines, thus forcing the entire software industry into the Windows programming environment. A monopoly was henceforth secured.

That strategy was repeated a few more times. Windows 98 was released amidst even more fanfare than 95, again announced to now be the true 32 bit software. That claim still did not measure up, but in Win 98, the Internet Explorer was so deeply embedded into the operating system, that it became impossible to separate these two products, causing the end of Netscape. In the year 2000, Windows ME (millenium edition) is shipped. Its real purpose was to bundle multimedia capabilities into the operating system, thus killing the business prospects of Real Media and other multimedia vendors.

In numerous court rulings Microsoft has been found guilty of copyright infringements and monopoly abuses, and has been forced to pay several hundred million dollars of fines and damages. However, that pales in light of the approximately 50 billion of dollars of cash that the company has accumulated until 2003.

Niklaus Wirth develops more science

Meanwhile back in Switzerland, Niklaus Wirth took a sabbatical around the time when Microsoft was started. He spends 1976/77 in Palo Alto at the Xerox labs, where he gets to know the Alto workstation plus all the other fantastic goodies in the famed labs of Xerox. Impressed with Alto's performance, but disappointed he could not buy one, he decides to build a workstation himself upon his return to Zurich. The resulting computer Lilith already featured a graphical interface with windows, mouse and all in 1978, six years ahead of Apple's Macintosh. Yet, all attempts to commercialise Lilith failed. Another Swiss businessman, however, Daniel Borel, who had also spent time in the States in the 70ies, heard of Swiss mice being built in his home country. Based on designs and technologies of Wirths group at ETH, in 1981 Borel starts his company called Logitech, and is today the world's largest Computer Input Device maker.

Besides working on Lilith, Wirth also got finally tired of all the complaining about PASCAL. Trying to preserve PASCAL's mathematical and scientific integrity while updating it to the needs of the new world of computing, Wirth develops the programming language MODULA. Like PASCAL, MODULA was praised by insiders to be a fantastic and highly advanced language – however it never gained any acceptance in the commercial world. In a final push towards achieving the ultimate programming language, Wirth then developed the OBERON language in 1992. With it, he tried to introduce component-object programming, extending the notion from MODULA that a computer should be run on the same language from operating systems all the way to applications, all the while being fully portable to any hardware system. With Oberon, Wirth realized an operating system including a document and graphics editor, network and windows system with as little as 200 Kilobytes of memory space, at a time when the standard in the industry counted its applications already in the tens and hundreds of Megabytes. But to same effect: insiders knew OBERON to be elegant and simple to use, at least as powerful as any C code but much more reliable – still usage of the language was restricted to tiny niches.

Commercialisation via Oberon Microtechnologies

A group of young software engineers trained under Wirth founded the company Oberon Microtechnologies with Wirth as head of the supervisory board. The plan and hope was to commercialise the excellent software development environment of Oberon. Though the company was showered with prices and praises, they only just survived commercially. Eventually Oberon's staff developed a Java embedded systems module, called Jbed, for use

in embedded microsystems controlling industrial applications, such as brakes in locomotives and the like. The inspiration for that came from a different department at ETH, the Institute for Robotics, headed by Professor Gerhard Schweitzer, who had developed a miniaturized real time operating system for his machines. Merging the know-how of the Oberon's programming skills with the mechanical engineering know how of the robotics department led to Jbed.

In order to place more focus on the application of embedded systems, the founders left Oberon Microtechnologies and started Esmertec AG in 1999. In October 2001, Esmertec acquires a second round financing of 10 m Swiss Francs. Until that time, the telecommunications market is considered to be only one of several market opportunities, whereas the emphasis was on industrial applications. But the new venture capital partners, Earlybird Venture Capital of Munich and French Sofinnova focussed Esmertec's strategy on the mobile phone market and recruited a seasoned CEO who had a solid track record and network in the industry. Another 24 months later, Esmertec had achieved the said 25% market share of embedded Java systems on mobile phones, making it not only the worldwide technology, but also the market leader. Within that time frame, Esmertec also established offices in California, UK, Singapore, Korea, Shanghai and Japan. By now it hired about 50 highly qualified engineers at its Swiss headquarters, and another 50 staff are working in the offices abroad. The money from the venture funding will be partially used to acquire companies in the Far East.

Esmertec's technology is claimed to run applications on a mobile phone either as much as ten times faster or with the tenth of the hardware requirements compared to rival products. All rivals based their technology on the JAVA and C family type technologies – Esmertec instead used the programming philosophy they learned from Niklaus Wirth. Thus finally, Wirth's approach to programming software had found a significant application where it made enough of a difference to be important. The computers in mobile phones place a very high premium on speed, energy efficiency, spatial dimensions and reliability – in short everything that the PC and workstation world did not need to bother so much about in the past three decades. A software environment that could now economize with minimal hardware requirements, and did not encounter deeply entrenched legacy systems, could excel in this application – if only it acted fast and decisively. This fit exactly the profile of Esmertec in 2002 – and thus ultimately rewarded with an unquestioned worldwide success Niklaus Wirth's 40 years of efforts to impress upon the world the importance and value of simple and elegant programming.