



# Proud to Be a Programmer

Tatu Ylönen

Tatu Ylönen Oy Ltd

[ylo@tyoy.fi](mailto:ylo@tyoy.fi)

[ylo@ssh.com](mailto:ylo@ssh.com)

# Outline



**tectia™**

- Speaker's own career
- Start-ups and technical innovation
- programmers and Programmers (and other software professions)
- Visions for future

## Speaker's Own Career 1/4



tectia™

- Started programming as a hobby around 1982 (spend two summers at University of Helsinki computer science library reading)
- Commercial programming as a free lancer and independent author since 1984
- Graduated from high school and entered HUT in 1987
- Founded New Generation Software (NGS) Oy Ltd in 1988
  - Full text information retrieval software, for local area networks, newspaper archives, medical information, image archives etc
  - Sold to CEO in 1992 (who sold it to Novo ~1993)
- MSc from HUT in 1992 (thesis on full text retrieval)
- Compulsory military service 1992 – command and control systems

## Speaker's Own Career 2/4



- Researcher at HUT (Tekes & Academy of Finland) 1993-1995
  - Conceived the project, wrote applications, lab got funding
  - Implemented a fast database system
  - Lead to Licentiate's thesis in 1994
- Founded Applied Computing Research (ACR) Ltd 1994
  - To do project work, e.g. device drivers, a main memory database system, file system e.g. for Nokia
  - Became a holding company (with some research projects), eventually merged with SSH around 2004
- Frustrated by difficulties in getting database research results published, wrote a tool called Secure Shell (SSH) for secure remote logins over the Internet in early 1995, and published it as free software

## Speaker's Own Career 3/4



- Founded SSH Communications Security Corp late 1995 to commercialize SSH Secure Shell
- Bad distribution agreement deprived the company of most Secure Shell revenue, but the company developed world leading IPsec toolkit that grew rapidly
- Company grew >100%/year for first five years (350% in 1998)
- Profitable and self-funded for first four years; private placement in 1999 and listing on the Helsinki Exchange in 2000 (personnel 190 and sales 19Me XXXcheck)
- IPsec business hit hard by IT depression starting 2000, eventually sold the business in 2003
- Secure Shell distribution situation changed in 2000 and the company got to sell it directly; it started growing rapidly and replaced the IPsec business
- Secure Shell rebranded SSH Tectia around 2003, and completely reimplemented as a system-level product for very large corporations
- Major commercial breakthroughs in 2007, e.g. Wal-mart

## Speaker's Own Career 4/4



tectia™

- Various roles at SSH: first CEO&CTO 1995-1996, CTO 1997, CEO&CTO 1998-1999, CTO 2000-2001, CEO 2002?-2003?, CTO 2003-2004
  - Much of the CTO time hands-on involved in the most important development projects, actually writing code
- Co-founder in VDSL Systems Oy (1998?) (went bankrupt after burning 14Me of VC), invested in Portality in 1999
- Stepped down from day-to-day operations in SSH in 2004 (I still spend 1-2 days per week on strategic and board-level issues)
- Founded Tatu Ylönen Oy
  - Knowledge representation and processing, natural language understanding
  - Highly research oriented, but potential for changing the way computers are used - high risk, high reward project
  - In research&development mode, still a couple of years before products will ship
  - Currently five people, could use one or two good programmers more

# Speaker's career drivers



tectia™

- I've mostly been entrepreneur, never really worked for someone else
- Accustomed to risk and irregular income; it felt strange when I started getting regular salary from SSH after the third year
- My heart is in technology; I get pleasure from inventing new things, writing cool code, building things, learning
  - But I've also done, even enjoyed, a lot of the business side
  - I've also done technical support, technical sales support, put out fires (both literally(!) and figuratively) etc as needed
  - I also don't spit on the money; I wouldn't bother (with companies) if there wasn't potential for making big money
    - Secure Shell was initially written for political reasons though, and because it just was needed
- At some point during my career, I've used almost everything I learned at HUT
- I wish I had taken few business management and marketing courses, and gone for more than the minimum grade in mathematics/physics/chemistry

# Technology Start-Ups: The Two Tracks



**ssh tectia™**

- The cool mouse trap track
  - Someone comes up with a business idea that would help some group of customers, gets funding, hires programmers and architects to implement it
  - Usually very ordinary technology, mediocre programmers, lots of hype and marketing
- The technology track
  - Someone comes up with better or revolutionary technology, figures out where it will give benefits, gets funding, hires managers and marketing to help commercialize it
  - Often great technology, but challenged by lack of marketing, sales and business skills
- Combination of both is of course best, but unfortunately great business people and great technologists often don't know how to find each other

## Not All Cool New Technology Is New



ssh tectia™

- Many great products and companies and technological waves are based on good implementations of old ideas. For example, WWW was actually invented in 1965 – including the essence of HTML, hyperlinks, embedded graphics, dynamic forms with code to perform computations, e-commerce, users authoring their own pages – 20 years before Berners-Lee – just read the article on the visions for the MIT information system in Kochen: The Growth of Knowledge: Readings on Organization and Retrieval of Information, Wiley, 1967.
  - Just think of companies like Amazon and Google, whose main product is based on great implementations of old ideas (with some new stuff added by smart people, of course).
  - Often evolution of computer, storage or network technology has been the enabler (possibility of implementation that scales and existence of a potential customer base)

# Who Builds These Radically Innovative New Products



tectia™

- As is well known, the productivity of programmers varies greatly. A great programmer will produce 100 or more times more code, with much higher quality, solving much more difficult problems, than a mediocre programmer.
  - Truly mediocre programmers are rarely seen in start-ups, so the actual variance is more like a factor of 10 in productivity, and as much in quality
  - Great programmers also get paid many times more than poor programmers
- The difference comes from many factors, many of have to do with environment, motivation, form of specifications, etc.

# People Who Invent New Products Are Often Not Maintainers



tectia™

- Many innovative products are created by surprisingly small groups
  - Often, the core of the product is created by 1-5 people
  - Before the product goes to market, the team has often increased significantly
- Mature global products tend to have much larger teams developing and maintaining them
  - The high-productivity innovators tend to move on after a few years as the project matures
    - Often, the innovators, or founders, cash out and start the next company
  - Mature products need more predictable roadmaps, face much more complexity from customer requirements, have much more rigid processes
    - This radically reduces productivity
    - This is a necessity to make schedules and outcomes predictable

# How Does One Become a Great Programmer and Innovator



**tectia™**

- Good knowledge of fundamentals (algorithms, languages, operating systems, hardware, networks)
- The right amount of ambition and perfectionism, and healthy self-esteem
- Practice and experience
- More and more varied experience
- Experience from building new complex systems
- Experience from working in great teams and managing them
- Working with customers to understand their need and producing products that really do what they need
- Seeing hard times, failures, good times, successes

# Who Can Become a Great Programmer



tectia™

- You must be interested in technology (broadly) and learn it easily
- You must like learning new things, building new things, programming
- You must be able to work systematically even when facing difficulties and when the outcome won't be known for months/years
- You must be able to cope with stress to avoid burnout (most people need also other activities than work)
- Ready to do many things when needed, not just narrowly in programming
  - I think some customer interface, marketing, even management experience does very good for a programmer – even if one intends to be a programmer
- 10-20 years of experience
- Please understand that the title “Programmer” is a misnomer, because it can also mean the lowest level worker in a large software engineering organization – but I don't know a better term. Who cares about titles anyway (well, they can look nice on a CV, but they are less important than most young people seem to think)

# Some Thoughts on Technologies That Computer/Software Experts Will Need



**tectia™**

- Programming: C, some language with more advanced concepts (garbage collection, first-class functions, closures, ...), some object-oriented language (Java/C#/C++/...); data structures and algorithms broadly
- Database programming (at least SQL), network programming (TCP/IP), basics of security, localization issues & unicode, kernels
- Operating systems: Unix/Linux, Windows, kernel, preferably something embedded, preferably a microcontroller
- Hardware (memory architecture, modern multiprocessor architectures, communications, DSP, threads, assembler, compilers)
- Ability to transcend abstraction layers, all the way to hardware, as necessary

# Some Thoughts on Other Skills



**tectia™**

- **Basic work skills (how to be a good employee):**
  - Do your work as well as reasonably possible; focus on the essential and don't spend excessive amounts of thinking about things that aren't your business; motivate and help others (within reason – don't let it ruin your own work); maintains focus and direction even though (reasonable) difficulties – all projects and companies face difficulties at some stage
  - Bring up opinions and ideas (on relevant matters), but also accept decisions made
  - Understand that a company is not a democracy and not all decisions and their reasons can be told to everyone (especially in a public company)
- **Learning throughout professional life**
  - I think one should do something that teaches new things one day a week on the average (not necessarily every week and every year, but it must be made up later)
- **A love for technology and learning**
- **Ability to manage stress under pressure to avoid burn-outs**

# The Technical Track vs. the Management Track (vs. the Marketing Track)



tectia™

- Many companies recognize separate career paths for the management track (people who want to become managers and eventually lose touch with technology) and the technical track (people who still keep doing technology – often including hands-on programming – after 20-40 years on the job)
- Both careers tracks can offer very competitive salary development, opportunities for self-development and interesting work
- Transition from the technical track to management or other positions is always possible, but much more difficult in the other direction

# The CTO



- I am mostly thinking here about CTOs of fast growing technology-driven startups (whether founder-CTO or hired CTO)
- Role can be more technical or more managerial (may need other people to complement skills)
- Needs broad and deep technical expertise and experience (preferably 10-20 years, but sometimes young founders become great CTOs)
- Needs deep understanding of the field in which the company operates, on all levels (technology, standards, IPRs, products, customers, competition)
- People skills, leadership skills, ability to gather a great team, etc
- Role is fairly different in fast growing start-ups and in more established companies
- Especially in start-ups and as an entrepreneur in the initial years, work weeks can be very long and pressures high, so ability to cope with stress and good health is important

# The Product Manager Dilemma



tectia™

- Finding great product managers for technically oriented software products is extremely difficult
  - People with a marketing background without good technical expertise tend not to understand the product and usage scenarios well enough
  - Some of the best product managers I have seen have been very good engineers who wanted a change
- I'm not really encouraging engineers to decide to become product managers at young age, but it never hurts to take a marketing course or two, and to be honest, there is a shortage of good people in this area

# But What Do We Need All Those Project Managers for??



**tectia™**

- Sometimes it seems that every computer science student wants to be a project manager nowadays
  - Perhaps this is because in salary statistics say that project managers are better paid than programmers on the average
- Even some professors at HUT seem to think that programming will be done by peons in India or China and that only exciting design, specification and project management will be done in Finland
  - Only many of those peons in India and China are a lot smarter than you are, work harder and longer hours than you do, and will in a few years be much more experienced than you are. They'll be more than happy to lift that burden of design, specification and project management work from as well, not to mention the marketing, sales, manufacturing, top management, ownership and the remaining roles as well (well, perhaps those former project managers can switch sides and start doing purchasing for customers here – if there still are customers here with money to buy).
- I think a great project manager usually has many years of experience doing the actual work first
- All this said, a good experienced product manager is a very important asset in a software project

# Hype Warning



- Throughout my career, I've always made it an issue to try to see through hype
- The amount of hype generated by marketing departments of large companies and even well-funded start-ups is amazing
- It is not necessary to now every TLA, FLA or ELA (e.g. SAX, SOAP, DNA, DEN, XOOA, EJB, COMN, YSI, WEB2, XEED, ABEA, WESW, XMA, ETEMEA, JBAA, BOLLA, SUNE, JCLA, JOM, J9, PYJAMA)
- Most of these "technologies" are nothing spectacular, and can be quickly learned as needed in a matter of hours to days, provided the person knows the proper fundamentals
- Try to think yourself why something would be important or good, and to whom; don't trust what magazines and discussion forums say (most of the articles and authors are influenced by marketing departments anyway)

# We NEED Engineers Who Are Capable of Developing Radically New Technologies



ssh<sup>®</sup> tectia™

- Expert technologists with experience, breath, ideas, inspiration, motives, financing
- Expert marketing and sales people as well as top managers to turn them into businesses
- Environment that encourages success and risk taking and rewards success
- More than anything, innovation is constrained by the lack of people who are technically capable enough to see and implement great new ideas in reasonable time
  - My feeling is that most start-ups fail because of technology (even though it is not always attributed to that)
- We need a few dozen people in Finland to become great programmers every year
- If they don't come from HUT, where will they come from?

## And for the Rest of Us...



tectia™

- There is still a pressing need for competent development and maintenance engineers
  - In later stages of products and companies, work pace and range of change are more reasonable, and more people are able to cope with it
  - In fact, I think the people who create new products and companies are largely a different people from the people who work at and run established businesses
  - A surprising fraction of competent engineers on the job market now are foreigners – where have all the competent Finns disappeared?
- People with good technical skills are also needed in management, marketing, sales, project management, etc
- Remember that there are 60+ “universities” in Finland alone, and most of them educate “IT professionals”. The times are over when knowing some Cobol or Java guaranteed great job opportunities.

# A Word on the Future



- It would be premature to say that the Internet and telecommunications supercycles are over, but they are well past their peak
- In a few years teramachines will be commonplace in offices – provided there are applications – a teramachine has a teraflop of computing power, a terabyte of ram and many many terabytes of disk, and is the size of a fractional rack. Today it costs about 100-200ke, but in a few years the price will be only 20ke or so – cheap enough to get dedicated teramachines for specific applications
  - I have seen very few visions, and very little research, on what can be done with such machines in an enterprise environment outside the internet/telecommunications service provision and traditional supercomputing contexts
  - I think they have the potential for changing how computers are used and what can be done with them
- Even petamachines are not all that far away, and already exist in some of the largest corporations and research institutions; in 10-20 years every division or even department could have one

# My Vision



tectia™

- I predict artificial intelligence (in various forms) will make a comeback
  - Truly usable natural language interfaces
  - Knowledge representation for natural language, related processes
  - Intelligent agents
  - Not necessarily “thinking machines”
- A lot of relevant progress, even breakthroughs, have been made in linguistics in the last 15 or so years, now that it has fully broken free from syntax-centrism and transformational grammar
- We now have the memory and processing power to manipulate knowledge bases of realistic size in reasonable time
- For comparison, a terabyte is enough to store everything a person hears over a lifetime, and enough to take a small picture (10-20k) couple of times per minute every waking hour of a lifetime.
- I conjecture that computers now have the processing capacity for true artificial intelligence; the challenge is that we don't yet understand how the program needs to be written

# Questions?



- Questions?