Science in Software Innovation

Bert Hubert

Thoughts on the (non-)utility of science in software innovation

http://tinyurl.com/innoscience

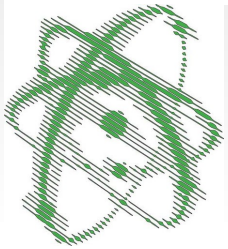
FOX-IT
FOR A MORE SECURE SOCIETY

PowerDNS

Agenda

- Who am I?
- Software innovation – what are we talking about?
- We are customers of universities, factories of the hard sciences
  - ”The secret deal”
- What we get out of universities & science
- What we'd love to get out of it
- Summary
- Drinks

Who am I?



- Applied Physics, dropped out somewhat beyond ”bachelor”
- Board member of VvTP, bit like GEWIS I think
- PowerDNS: Powers 40% of European domain names
- Research & Development
- ”Experts in IT Security – for a more secure society”

PowerDNS

- DNS converts ”www.tue.nl” into 131.155.2.83
- PowerDNS is the DNS server of around 30%-50% of all European domains, in use by the largest DNS operators in the world
  - You 'use' it every day
- First DNS server to be able to run from a database
  - ”They said it could not be done”
- First DNS server with ”easy DNSSEC”
- Powers Wikipedia with module by TU/e graduate Mark Bergsma (thanks!)

Fox-IT

- Supplies governments, financial institutions and others with IT security training, solutions and services. Around 100 ”nerds, geeks and hackers”
  - High-end cryptography, steward to the Philips Cryptosystems department
  - Audits, Forensic investigation
  - Fighting cybercrime
  - Replay: Innovative communication analysis tools
- Replay was launched in 2006 and is now one of the most advanced products on the market

Fox-IT & Universities

- We get most of our star performers from universities
- Frequent internships
- Students also graduate with us
  - As a drop out, it is highly pleasing to help someone else graduate!
- One of our core products, the Data Diode, originated as a graduation project
- In return we produce ”industry relevance” letters...

This presentation

- I've been asked to hold a ”stimulating presentation” that will provoke interaction with the audience
  - So please interrupt if you don't agree!
  - Or if you agree and have a good example, please also interrupt!
- During the entire presentation, please keep in mind that I'm a great fan of science!
- But I've been tasked to make sure you have something to talk about over the drinks that follow this presentation ;-)
  - So here goes

Software innovation: what are we talking about

- We often hear about ”ICT” or ”IT” field
- This, sadly, comprises everything from installing a mouse driver to creating a space based navigation system
- Quite a lot of IT is in fact no more exciting than assembling IKEA furniture!
- ”Large” does not mean innovative. Prime example, government payrolling system ('P2000')
- Non-innovative ICT mostly requires very good planning – it is not easy!

Software innovation

- Doing things that have not been done before
  - Not at all (example: first ”internet”)
  - Not at that scale (example: google)
  - Not under such constraints (example: iphone)
- Unsure if it can be done at all
  - First internet melted down
- It is often not even very clear what needs to be done
  - Might change during implementation
- 1% innovation, 99% perspiration
- Exciting!

I am a customer of Universities

- Thank you!
- We are grateful customers of the education you provide our future employees
  - For free too!
  - We often pay 1500 euros/day for educating people – consider a master's degree to cost 650k euros
- Example, Intel recently indicated it would close a plant if the local EE faculty would close down
  - It is immediately relevant

A customer of Universities

- Managing directors: Mathematics, Physics
- CTO Replay: Quantum Physics
- Founder Replay: Physics dropout
- Lead developer: Physics
- Lead UI designer: computer science
- Most programmers either finished a university degree or spent a lot of time @ uni
- Financial Director: Quantum Physics

What do we get out of universities & science

- **Almost no direct innovation**
  - oops
- Graduates with some relevant skills
- People with the right vocabulary & background
- **Graduates with a scientific mindset**
- **People that know the answer might not be in the book**
  - **Or that it might be wrong**

Innovations from computer science

- Direct results applicable in industry are actually quite rare (but very important)
- We asked all Fox-IT programmers, they came up with:
  - MESI protocol (1984)
  - Proven cryptography
  - Graph coloring theory
  - Halting problem
- Worryingly, this does not contain a lot of recent developments


ren en
fietsen
ijbaan
Land of
THE INNOVATOR

Indirect scientific contributions

- In short this is almost everything we do
  - WWW came out of CERN
  - GNU came out of MIT
  - Linux originated exclusively within universities
  - Bell labs (C, C++, the Transistor, need I go on?)
- **All the very words we use come from academia**
  - And a lot of our culture too
  - Case in point: Edsger Dijkstra ('Goto considered harmful' – or not!)

Something business would never do

- Two wonderful academic areas of research:
  - Quantum computing ('there is no quantum computer yet')
  - Post-quantum cryptography ('for when we get one')
- In Eindhoven terms: Schnorr versus Tanja Lange & Dan Berstein
- When this is done, and eventually the physicists give us the quantum computer, we'll be ready for it
- Science will have provided the infrastructure without being a flashy 'innovation'

The scientific mindset

- Large computing environments are complex systems with dynamic behaviour
  - Ask Gödel
- Errors can occur at compile time (good), during tests (good) or in full operation (bad)
- When studying a misbehaving system, the full 'scientific method' needs to be employed
- Hypothesis, **experiments**, no interest in theories that can't be falsified etc
- Physical scientists are actually closer to this world – we actually talk about 'instrumenting' code

The ”edge of science”

- When doing new things.. you are doing new things
  - Sounds so simple
- Graduates of universities have had that experience too
  - The answer of the experiment is not known
  - The goal of the research is to learn **new** things
  - No one told you the ”how”
- In science, it is clear that while the answer will be there, it might not be in the book
  - You are **writing** the book
  - (the tools may be in the book)

Overall, it is working

- We have no other source of critical thinking employees!
  - Polytechnic graduates typically reach for the book when asked to innovate
- The scientific method works very well on any complex system
  - And you have to believe in it → witch docter otherwise
- Without academia, we wouldn't even have words to talk about what we are doing
- We also get some directly useful skills & innovation

So what IS a university?

- Secret pact (in descending order of loftiness)
  - (Fundamental) Science
  - Satisfy student's curiosity (& need for beer)
  - Get students marketable skills (& a job)
    - Conversely, get us companies useful employees
  - Keep everybody at university employed
- There is an interchange between these four
- ”Universities get funding because society finds it worthwhile to do so”
  - Lighter phones, cure for cancer, environmentally safe energy, cars that run 300kph etc
  - **Needs to get people jobs & industry employees too!**

Mathematics & CS

- (Pure) mathematics has long had a difficult relation with industry
  - Rarely a business need to prove Fermats last theorem
  - Cryptography has (slightly) wider practical applications
  - However, mathematicians are almost guaranteed to be so smart you take the risk ;-)
- CS sits at a very difficult cross roads
- ”Too theoretical to be practical, too practical to guarantee the brilliance that makes up for that”

Some Dijkstra quotes

- Google for ”Dijkstra quotes computer science”
- **'Computer Science is no more about computers than astronomy is about telescopes.'**
- 'I mean, if 10 years from now, when you are doing something quick and dirty, you suddenly visualize that I am looking over your shoulders and say to yourself "Dijkstra would not have liked this", well, that would be enough immortality for me.' - he got that.
- 'The competent programmer is fully aware of the strictly limited size of his own skull; therefore he approaches the programming task in full humility, and among other things he avoids clever tricks like the plague'

Perspective on CS in Software Innovation

- University CS education offers us students with a mix of: useful skills, scientific knowledge, ”knowing the answer is not in the book”, scientific method
- This mix is not optimized for having an innovative software industry
  - **And indeed, we almost have none in The Netherlands**
  - Polytechnic education offers us another, equally unoptimal mix
- We do have a large ”IT Industry”..

The problem

- We need a mix of scientifically curious people who can think in a disciplined fashion about complicated systems ('the scientific method')
- We also need people with a vast amount of skills!
  - Actual programmers! That know about real hardware! (not 'MMIX')
- **In our experience, there is not a single school nearby that educates people to become actual non-IKEA programmers**

The explanation

- CS departments here educate people to **design** software
  - Teach theory on how to do this
  - Do some implementation in order to further this design ability
    - Often on research platforms
- The actual implementation is subservient to the design
  - May be outsourced, or performed by less educated personnel
- ”The rest is an implementation detail”

The real world

- In the real world.. innovative software is not designed first by designers
  - And then implemented by ”programmers”
- As far as I know this has never worked for doing anything new
  - It has worked for ”IKEA Programming”
  - Oddly enough, the IKEA Programming at IKEA was a mess!
- We can't deal with designers from university that aren't actually skilled at implementing!

So where DO programmers come from?

- There are ”natural born programmers” that soak up the things they did not yet know from university
  - Is a good place for it
- In physical sciences, the student is often tasked with programming the experimental software, and in the course of doing that becomes a programmer
- If you pick the right combination of courses & specializations, you can come a long way

Even more Dijkstra

- ”Nowadays machines are so fast and stores are so huge that in a very true sense the computations we can evoke defy our imagination. Machine capacities now give us room galore for making a mess of it.
  (...)
  Developing the austere intellectual discipline of keeping things sufficiently simple is in this environment a formidable challenge, both technically and educationally.” - 1984, EWD 898
- This kind of education is vital!
- Actually, in EWD 898, Dijkstra said almost anything that I've wanted to say, except better
  - And from another direction

The 'austere discipline' best theory school

- In one vision, we get universities that teach Computer Science, or even better, ”Informatics”, in such a rigorous way that the graduates are so infused with theory & knowledge that they will have fully satisfied their curiosity, and are so well educated that adapting to the ”real world” is merely a special case
- These graduates would be hired merely because they had proven to be able to survive such a thorough education
  - This has happened with Physics for example

The ”silicon valley” school

- In the second vision, the education shifts towards practical tools, supported by the theory needed to fully understand what is going on
  - An education would in that case contain a large amount of 'actual implementation'
  - Based on technologies that can actually be put on a resume to get a job
- This would for example include things like 'proper use of revision control, bug trackers, regression tests, agile development and interfacing with Open Source projects'

Visions compared

- The ”theory school” is only for the best students
  - It stands on the fact that graduates are theoretically so well educated that they can be expected to deal with everything
- The ”trade school” however is not easy either. It includes relevant theory too. Being a ”master craftsman” is only possible with sufficient talent
  - Specifically, the trade school is not polytechnic
  - As an example: high performance concurrent code needs to know about MESI! (DJB)

Where we are today

- Most departments in The Netherlands sit unconfortably between the two visions
  - Partially due to a misunderstanding what the relation is between ICT & Innovation
    - ”IKEA IT” versus ”ASML IT”
- Eindhoven may in fact have the best fit with its local industry
  - Seen from The Hague and Amsterdam, this is quite enviable!
- The innovative software industry may have been happy with the deal so far
  - Non-innovative definitely

It may also be a Dutch thing

- We struggle with innovation
- Our government thinks innovation can be stimulated with tax breaks
- In fact, innovation has a lot to do with mountain climbing
  - Why do we climb the mountain? Because it is there!
    - Not because someone made it tax free!
- Should a university be a tool of industry?
  - Almost a dirty question!
  - It is the case though..

It is also an industry thing

- The innovative software industry is mostly small companies (and small anyhow)
  - And grouped together with the non-innovative ICT industry (where all the money is – and they get their Java programmers!)
- If we want universities to deliver more ”trade skills”, we should be supplying more input **and probably even courses & professors!**
  - Oh, and we don't want to pay for it ourselves
  - The university was free already (or, more precisely, we feel we are already paying for it)

Summarising

- Summarising, on the (in)utility of science in software innovation
- I love the highly educated people that challenge existing wisdoms and realize that new things require new thinking!
- I love the scientific method
- I could not even **talk** abour our industry without using the vast body of knowdledge generated by science
    - And there are some specific academic highlights
- I also appreciate the 'ready to use' skills of graduates
- I do think the mix is not optimized though
    - And we'd all have to work on that if we want to improve things
- **But please keep doing what you were doing → it works ;-)**

Questions?

Science in Software Innovation

Bert Hubert

Thoughts on the (in)utility of science in software innovation

http://tinyurl.com/innoscience

hubert@fox-it.com / bert.hubert@netherlabs.nl
+31622440095

FOX-IT
FOR A MORE SECURE SOCIETY

PowerDNS