Gresham College

12 June 2018

Computers and the Future

Professor Martyn Thomas

INTRODUCTION

Three years ago, with the support of the City of London IT Livery Company *The Worshipful Company of Information Technologists*, Gresham College created the opportunity to become the first Professor of Computing in the 400 year history of the College and I applied for the post. In my letter of application, I said this:

> *Modern society is dependent on computers. Less than 70 years after the first successful program ran on the first modern computer, software-based systems are everywhere. Yet we are only at the beginning of the revolutionary changes that computers will bring to society. We are on the verge of the "internet of things", where almost everything could contain intelligence and be network connected. 2015 may see the first artificial life, where a whole organism is reproduced at molecular level as a software simulation. Several countries have started preparations to introduce driverless cars on public roads. Autonomous air vehicles with high resolution cameras and satellite navigation have already moved from military applications into toyshops.*
>
> *But all this progress is dependent on a software industry that is still at the craft stage, 45 years after the phrase "software engineering" first came into common use. Most programmers lack even a basic understanding of computer science or of the disciplines that are fundamental to engineering professions. As a result, many software projects overrun in costs and time, fail to deliver real benefits, suffer reliability and usability problems, and leave users exposed to costly cybercrime.*
>
> *My lecture programme will explore the state of software today, how we got to where we are, and what we shall need to do to shore up the foundations of a digital society that is increasingly built on sand. The lecture programme has been designed to inform, to entertain, and to stimulate balanced discussions that lead to effective actions. Information technology underpins the wealth creation by the City of London, so it is particularly appropriate that Gresham College should play a leading role in accelerating the transition of the craft of software development into a mature engineering profession.*

My first lecture was on 20 October 2015 and tonight, after 19 lectures, my lecture series will come to an end — although thanks to continuing support from the Livery Company, the IT Professorship continues strongly and my successor as Professor of IT will take over this summer and give his first lecture in October. There will also be three lectures from a new Visiting Professor of IT.

This evening, I want to draw together the central messages from my lecture series and to look to the future, because we are still only just at the beginning of the industrial and societal revolution that started in Manchester almost exactly 70 years ago on 21 June 1948 (a little after 11 am) when the first modern computer ran its first program and changed the world in ways that were certainly not imagined by pioneers Tom Kilburn, Freddie Williams, Alan Turing, Maurice Wilkes, John Pinkerton and the women and men who worked with them.

Our world will change much more in the next seventy years. It will be an exciting and profitable time to be working with software and I envy future software engineers the fun, excitement and satisfaction that they will have.

When technology is advancing rapidly, it is easy to imagine that all the challenges that you face are new and that there is little to be learnt from your predecessors. But in computing as in all other branches of engineering (and perhaps in all professions) there are very few **new** mistakes. Almost everything that you do wrong and that causes you trouble and humiliation, someone else has done before and, if it was a big mistake, they (or someone else) will have drawn lessons from it and written about it.

The great software engineer and manager Fred Brooks was called "the father of the IBM 360" (IBM's hugely influential mainframe computer series), having run the project both for the hardware development and for the operating system, OS/360. He wrote, *"It is a very humbling experience to make a multimillion-dollar mistake"* — and this was in the 1960s when a million dollars was real money. From his experience grew his famous book *The Mythical Man Month*[i]*,* where the mistake is described in detail on pp 47-48.

My lecture series *Living in a Cyber-Enabled World*[ii] contains many of the lessons that I have learnt from my first 50 years in the software industry. I hope that future engineers will find them helpful as they address the challenges thrown up by the future.

In economics, **Gresham's Law** is the principle that "bad money drives out good", from a letter that Sir Thomas Gresham wrote to Queen Elizabeth I when she came to the throne in 1558[iii]. I have found that it is also true that bad software drives out good software, because it is cheaper and quicker to write bad software than well-engineered software and so the bad software captures the market. Perhaps this should be called **Gresham's Law of software**.

It seemed appropriate to number the lessons and principles that I have expounded in each of my lectures **G1** to **G18** below.

LECTURES, LESSONS AND PRINCIPLES

In my first lecture on 20 October 2015, *Should we trust computers?*[iv] we saw that software is a critical component in almost everything that happens in society and that the amount of software has grown dramatically, so that a modern car now contains 100 million lines of software, more than five times as much code as the flight control software of the Boeing 787 aircraft. Even the best programmers make a lot of errors and many of these errors could never be discovered by testing the software but would remain until an unusual series of circumstances cause the software to fail.

G1: Only trust software-based systems when you have good evidence.

On 12 Jan 2016, the next lecture, *A very brief history of computing, 1948-2015*[v] showed the parallel histories of computer hardware and software and that the incentives for hardware manufacturers has been to invest heavily in making their designs as correct and reliable as possible so that they could mass-produce and sell millions of identical, reliable low-cost chips, whereas software manufacturers compete by being first in the market with new applications and new features, not on reliability. The consequence has been three software crises: in the 1960s, in the 1980s and our current cybersecurity crisis.

G2: Hardware development has become professional engineering. Software remains a craft industry.

On 9 February 2016 I asked, rhetorically, *How can software be so hard?*[vi], and showed that developing software is particularly difficult because it is complex and creative, because the important properties are emergent, because the costs and timescales must be predictable, because there must be evidence that the quality is good enough (which is the hardest thing to achieve) and most importantly because the industry is still very immature as an engineering profession.

G3: Software is very complex. It contains the complexity of the application and often more.

The following lecture was on 5 April 2016. *Computers, people and the real world*[vii] used the example of the failures of the London Ambulance Service Computer-aided dispatch system and of the National Programme for IT in the NHS to illustrate how projects go wrong. These were *sociotechnical systems* designed to support the way that people carry out important and time-critical roles, but they were seen as technology projects more than as business change projects. The key message was that there is no such thing as an IT project because the ways people work is more important than technology and you need to invest more time and money in getting the changes to your business processes right than you invest in the technology.

G4: There is no such thing as an "IT Project".

On 3 May 2016, my fifth lecture, *Cybersecurity*[viii] explored the mysteries of buffer overflows, cross-site scripting and other common cyberattacks, explaining how they are carried out and how they can be detected and prevented. It is unreasonable to blame users when they are damaged as a consequence of using the tools they have been given in the way they were designed to be used. Much of the liability must remain with the manufacturers of badly flawed software.

G5: Cybersecurity is mainly a problem of badly designed software.

The final lecture of year one on 14 June 2016 looked at *Big Data and the broken promise of anonymisation*[ix], explaining that it takes remarkably little data to uniquely identify an individual and showing that it is practically impossible to anonymise most personal data records.

G6: It is usually impossible to anonymise data about individuals.

My second year started on 18 October 2016 by asking *Are you the customer or the product?*[x], a question that has been asked many times following the exposure of the ways that personal data collected by Facebook and other commercial companies has been used to profile and influence individual voters. Personal data reveals an extraordinary amount of information - much more than it appears to. It has considerable value to anyone who wants to influence the way that people behave - especially advertisers and politicians - so many companies will collect, analyse and retain as much personal data as they can. From May 2018, the General Data Protection Regulation may change the balance of power between individuals and companies because the fines for breaches of GDPR can be very large.

G7: The data you reveal will be used to make hidden decisions that affect you.

On 10 January 2017, *Safety-critical systems*[xi] asked "how safe is safe enough?" and found that the value of a life varies remarkably between countries and between different parts of society. We examined whether we could rely on software that had been "proven in use" and showed that a program that has run for a year without failing still has at best a 50% probability of failing in the following year. The international standards for safety-critical software were shown to be deficient in many ways, especially because they were written before cybersecurity was recognised as a major problem that undermines the notion of independent failures of safety functions. Standardisation bodies have been far too slow to update their standards.

G8: International standards for safety-critical software need urgent revision.

The dilemmas of privacy and surveillance[xii] on 7 February 2017 considered how the need to police cyberspace should be balanced against the internationally recognised legal right to personal privacy. Edward Snowden's publication of hundreds of top secret documents shows the extraordinary extent to which security services intercept and analyse everything in cyberspace and raises the question of whether government surveillance of citizens might threaten democracy. Strong encryption was shown to be essential for society even if this means that there will be files and messages that the police cannot read. The tension between privacy and security has led to a continuing arms race between the spooks and the geeks.

G9: Policing of cyberspace must be effective but shown always to be proportionate.

On 4 April 2017, *What really happened in Y2K?*[xiii], revisited the Millennium Bug to see whether the right lessons had been learnt. There is a pervasive myth that the thousands of person years of effort and billions of dollars spent worldwide were a waste of money because the forecast calamity did not occur, whereas the evidence shows that widespread failures were only averted through enormous international effort coordinated by the UN. The Millennium Bug was a near miss and the key lessons are to avoid having multiple systems that depend on a single point of failure (such as GPS) and that loosely coupled systems are more resilient than just-in-time supply chains. Redundancy is a valuable investment in resilience.

G10: Y2K was a genuine threat and a very near-miss. The lessons have not been learnt.

Having examined multiple examples of the problems that result from a test-and-fix approach to software development, on 2 May 2017 we considered the alternative. *Making software correct by construction*[xiv]*,* considered strong software engineering methods including mathematically formal proof of important system properties. The example of the *Tokeneer* experiment by Altran/Praxis for the US National Security Agency[xv] showed that such methods are practical, teachable and cost-effective; the use of these methods[xvi] to develop the iFACTS air traffic management system[xvii] showed that they could be used successfully on large scale safety critical software.

G11: Mathematically formal software development is practical and cost effective: *better can also be cheaper.*

On 13 June 2017, we discussed *Artificial Intelligence*[xviii]*,* which has been an objective since the earliest days of computing and has been given a major boost by the recent successes of machine learning systems and especially of deep neural networks. These successes are all in tightly constrained areas and generally-applicable artificial intelligence seems as far in the future as ever. The controversy over whether there can ever be a "singularity", where AI systems become more intelligent than humans and increasingly develop systems that humans cannot understand or control, echoes the objections that Alan Turing answered in 1951. The singularity will happen but not soon.

G12: An AI system will one day be able to do everything that a human can — but not in my lifetime.

Is Society ready for driverless cars?[xix] on 24 October 2017 investigated what evidence would be necessary to show that a driverless car would have fewer accidents than a human driver so that it could be safely used for all types of journeys. There are many more barriers to achieving safe "Level 5" driverless cars than most politicians and other commentators seem to realise. In particular, the way in which driverless cars might affect the behaviour of other road users seems to be under-researched.

G13: Society is not ready — and neither are the technology and regulation.

On 9 January 2018 *Will Bitcoin and the blockchain change the way we live and work?*[xx] examined public key cryptography, digital signatures and the other technologies that enable Bitcoin and other cryptocurrencies. The lecture was delivered against a background that was wrongly said to match the tulip mania of the 1630s[xxi]. The value of a Bitcoin was almost $2,000 at the time of the lecture, though I doubt that I was responsible for the halving of its value that followed almost immediately.

G14: Distributed ledger technologies are more significant than cryptocurrencies.

Hospital systems are an important group of safety-critical socio-technical systems and in *Computer bugs in hospitals - a new killer*[xxii] on 26 February 2018, Professor Harold Thimbleby and I explored the ways in which badly designed human computer interfaces (HCI) could lead medical staff to make serious errors. We concluded that it was possible that HCI faults are implicated in hundreds of annual deaths and injuries to patients and that urgent research is needed to establish the facts and to address the common causes.

G15: The safety assurance and regulation of medical devices and systems is not fit for purpose.

Despite the Russian interference in the US presidential election and the UK Brexit referendum, there is still active interest in the introduction of online voting, which politicians assume (against the evidence) would increase voter participation in elections. On 13 February 2018 *Should we vote online?*[xxiii] considered the evidence because voting is a uniquely difficult question for computer science: the system must verify your eligibility to vote; know whether you have already voted; and allow for audits and recounts. Yet it must always preserve your anonymity and privacy.

G16: Voting is a uniquely difficult question for computer science and currently unsolved.

Computer systems are already pervasive throughout society and on 20 March 2018 *The internet of things*[xxiv] explained why billions of new networked systems will be introduced in the next decade and lead to most internet traffic being between computer systems with relatively little human involvement. These new systems represent a great opportunity for improved services and greater efficiency, but they will also be a very large addition to national critical infrastructure, with little assurance of security, safety or resilience.

G17: Our cyber-enabled society needs rigorously engineered foundations, or it will fail.

Computers and warfare[xxv] considered the implications of war in cyberspace on 29 May 2018. The increasing emergence of cyberspace as a dimension of warfare changes everything, because warfare is traditionally focused on geography whereas cyberspace crosses borders, penetrates walls and changes the balance between major and lesser powers.

G18: The possibility of cyberwar needs a much stronger and more strategic response.

COMPUTING IN THE FUTURE

The applications of computing that we have considered over the past three years will continue to be important and to throw up new opportunities and new challenges. Legacy systems, cybersecurity, robotics, big data and artificial intelligence would be enough on their own to keep a generation of software engineers busy for a decade, but other innovations will create both technical and ethical challenges. As we have seen with driverless cars, the greatest enthusiasm for unproven technologies will come from those who least understand them and from those promoting their own commercial interests.

The author William Gibson famously said "the future is already here - it's just unevenly distributed"[xxvi]. Some recent developments seem to threaten history and perhaps even the foundations of society. How will we retain control over the truth when it is already possible[xxvii] to create seemingly authentic audio recordings of people giving speeches that they never delivered, constructed from samples of real recordings of their voice and matched to any chosen text? Today even lip-synchronised video allows history to be rewritten with convincing evidence[xxviii].

Usable quantum computers that utilise quantum properties superposition and entanglement are starting to seem a practical possibility[xxix] and quantum computers can run algorithms that can solve *some* problems[xxx] exponentially faster than classical computers. IBM has already built a 50-qubit machine, with chips cooled to 15 thousandths of a degree Kelvin above absolute zero, but it only remains stable for around 90 microseconds. It has been said that quantum computing has already moved from being a scientific dream to being an engineering nightmare! One class of problems that already has a quantum algorithm is integer factorisation - Peter[xxxi] Shor's quantum factorisation algorithm[xxxii]. Once it becomes practical to run this algorithm for 256-bit and larger integers on a quantum computer, it will break[xxxiii] the encryption that secures https websites and all online banking and e-commerce, as well as a lot of other commercial and military security. This would be a serious problem even if companies were able to upgrade to post-quantum cryptography[xxxiv] overnight, because there would still exist a lot of their previously encrypted files that have been copied by competitors and others and that contain sensitive data.

Large scale quantum computing - say 3000 to 4000 qubits - may be only a few decades away, though there is the possibility that representing even 400 qubits would exceed the maximum information that the universe can hold[xxxv].

A range of ethical problems are being created by the increasing use of computer systems (often in the cloud) to curate personal memories (such as family records and photographs) and to support the elderly (for example). Who should have access to the data when the original owner loses mental capacity or dies, and how should this be managed? Should a deceased person's instructions to destroy the data be a legal duty on their executors or on the cloud service provider?

Looking further ahead, there is the possibility of simulating the complete biology of entire living organisms, leading to life "in silico". Caenorhabditis elegans (c. elegans) is a roundworm, about 1mm in length; its entire genome has been mapped and so has much of its proteome[xxxvi] and it is a standard model organism for many biologists[xxxvii]. The OpenWorm project[xxxviii] aims to use the data to create the first digital life form. When someone succeeds, could they run an accelerated evolution? Where might that lead? Is it too soon to start considering the implications and how society should respond?

Recent developments in technology such as the rapidly growing adoption[xxxix] of facial recognition despite a false positive rate of 90%[xl] are creating the usual tensions between opportunities for new applications and legitimate concerns about privacy. Augmented reality and brain-computer interfaces present other challenges. There will be no shortage of topics for future Gresham IT Professors to describe and discuss.

CONCLUSION

The modern computer was born in 1948 and so was I. I have spent my whole career in the computer industry trying to ensure that important software projects and systems did not fail and, when they did fail, helping to identify the reasons.

I have learnt to be sceptical about computer-based systems: about estimates of future development costs and timescales, about reliability, safety and security, and about usability. The solutions to these problems exist in the professional discipline that we call *software engineering,* but they have not been widely adopted during the past 50 years and I see little prospect of the widespread adoption of software engineering methods in the next few years.

Digital systems have enormous potential to improve our prosperity, our leisure, our work, our healthcare and our overall quality of life but these benefits are threatened by the poor quality of so much software development combined with the growing cybersecurity threat. Strong software engineering is practical and cost-effective, using science-based methods and tools. We should expect - and accept - nothing less because our wellbeing, our prosperity and even our survival depend on it.

If men could learn from history, what lessons it might teach us! But passion and party blind our eyes, and the light which experience gives is a lantern on the stern, which shines only on the waves behind us! - Samuel Taylor Coleridge.

© Martyn Thomas CBE FREng, 2018

References

[i] Frederick P Brooks Jr, *The Mythical Man Month,* Addison Wesley, 1975.
[ii] https://www.gresham.ac.uk/series/living-in-a-cyber-enabled-world/
[iii] http://eh.net/encyclopedia/greshams-law/
[iv] https://www.gresham.ac.uk/lectures-and-events/should-we-trust-computers
[v] https://www.gresham.ac.uk/lectures-and-events/a-very-brief-history-of-computing-1948-2015
[vi] https://www.gresham.ac.uk/lectures-and-events/how-can-software-be-so-hard
[vii] https://www.gresham.ac.uk/lectures-and-events/computers-people-and-the-real-world
[viii] https://www.gresham.ac.uk/lectures-and-events/cybersecurity
[ix] https://www.gresham.ac.uk/lectures-and-events/big-data-the-broken-promise-of-anonymisation
[x] https://www.gresham.ac.uk/lectures-and-events/are-you-customer-or-the-product
[xi] https://www.gresham.ac.uk/lectures-and-events/safety-critical-systems
[xii] https://www.gresham.ac.uk/lectures-and-events/the-dilemmas-of-privacy-and-surveillance
[xiii] https://www.gresham.ac.uk/lectures-and-events/what-really-happened-in-y2k
[xiv] https://www.gresham.ac.uk/lectures-and-events/making-software-correct-by-construction
[xv] https://www.adacore.com/tokeneer
[xvi] https://www.slideshare.net/AdaCore/white-open-do
[xvii] https://nats.aero/blog/2013/07/how-technology-is-transforming-air-traffic-management/
[xviii] https://www.gresham.ac.uk/lectures-and-events/artificial-intelligence
[xix] https://www.gresham.ac.uk/lectures-and-events/is-society-ready-for-driverless-cars
[xx] https://www.gresham.ac.uk/lectures-and-events/will-bitcoin-and-the-block-chain-change-the-way-we-live-and-work
[xxi] http://theconversation.com/tulip-mania-the-classic-story-of-a-dutch-financial-bubble-is-mostly-wrong-91413
[xxii] https://www.gresham.ac.uk/lectures-and-events/computer-bugs-in-hospitals-a-new-killer
[xxiii] https://www.gresham.ac.uk/lectures-and-events/should-we-vote-online
[xxiv] https://www.gresham.ac.uk/lectures-and-events/the-internet-of-things
[xxv] https://www.gresham.ac.uk/lectures-and-events/computers-and-warfare
[xxvi] quoted in The Economist, December 4, 2003
[xxvii] https://www.bbc.co.uk/news/av/uk-scotland-43436361/john-f-kennedy-s-lost-last-speech-recreated
[xxviii] https://www.youtube.com/watch?v=9Yq67CjDqvw
[xxix] https://www.technologyreview.com/s/610250/hello-quantum-world/
[xxx] https://www.youtube.com/watch?v=-ysVGWtAjio
[xxxi] http://www-math.mit.edu/~shor/
[xxxii] https://arxiv.org/pdf/quant-ph/9508027.pdf
[xxxiii] https://physicsworld.com/a/what-is-shors-factoring-algorithm/

[xxxiv] https://pqcrypto.org/
[xxxv] https://arxiv.org/ftp/quant-ph/papers/0703/0703041.pdf
[xxxvi] https://www.ncbi.nlm.nih.gov/pubmed/27453442
[xxxvii] https://www.sciencedirect.com/topics/neuroscience/caenorhabditis-elegans
[xxxviii] http://openworm.org/getting_started.html
[xxxix] https://nakedsecurity.sophos.com/2018/04/16/facial-recognition-cameras-on-lamp-posts-to-be-tested-in-singapore/
[xl] https://www.telegraph.co.uk/news/2018/05/05/police-defend-facial-recognition-technology-wrongly-identified/