**Will software that writes code alter tech’s script?**

It is an axiom of the internet age that almost nothing done by human hand cannot be done better by or with software. But what of the business of writing software itself? Much of it is repetitious, bug-riddled grunt work. Cannot this be made more efficient by software? In fact, cannot code write code? To do this would be to turn the unofficial slogan of Silicon Valley in on itself: software will not only eat the world but software will eat software. This is the direction of a team of University College London computer scientists, whose work does not just aim to take the inefficient slog out of programming but also to transform the role of most programmers. “I predict that the next computing language will not be computer language but natural language, human language,” says Guillaume Bouchard, a member of the UCL team. “If you say to your computer ‘write me a computer game in which a shark chases a man’ the computer should know what you want and create the game before your eyes. Then maybe you say ‘make the shark fiercer and faster’ and the computer will revise the code.” It sounds far off but later this year a spin-off company called Bloomsbury.ai — co-founded by Mr Bouchard — intends to release a public demo that enables people with no programming background to perform complex data analytics in a matter of minutes. With training, the same virtual agent could be used in everything from art creation to consumer products, the company claims. In such a world, says Mr Bouchard, most programmers would be more accurately described as “trainers”. You would teach a computer to write code and to understand your intent by chatting to it. Another company at the frontier of code-that-writes-code is Queue Software in Raleigh, North Carolina. Aidan Cunniffe, CEO and co-founder, says: “In six to 10 years you won’t have mass market apps like Apple’s iOS email as we do today. For example, if you’re at the theme park and you want to keep track of your kids, you’ll tell your device what you want and it will write the code for you in a second.” Today, however, Queue Software is focused on more near-term goals, specifically the January release of Dropsource, its automated code-writing platform. Dropsource writes applications based on the function and intent a user inputs. With this information, the system selects the optimum design and development approach from the same options a developer would normally consider when programming an application to, say, keep score, move between pages or gather account details. Whereas Dropsource will write the code in tenths of a second, the normal process requires a designer and/or project manager to work with a developer and write the code from scratch, even if almost identical work had been done many times over in the past. Dropsource has already been used to build iOS apps by software development houses that were granted early access. Another company on the path to code-written-by-code is New York-based Bubble.is, which offers a drag and drop visual interface for building apps. Again the promise is that no coding skills are required. Mr Bouchard estimates that there are 11m programmers in the world. “These people have been trained to translate and transfer someone else’s idea into another language. Much of this programming work is laborious — they could use their smartness in other domains.”

The common intent of such pioneers is to demystify and open up the realm of technological innovation, creativity and self-empowerment to both the developer class, who understand the computer language that powers the invisible workings of the modern world, and the vast majority of people who do not.

But there are other powerful imperatives that are pragmatic and economic. The demand for developers today far outstrips supply, and this holds back innovation and raises costs. In Silicon Valley developers rotate from company to company, collecting big inducements each time. Meanwhile, accelerator programmes for start-ups are surrounded by people with ideas looking for a tech co-founder without whom they cannot even apply. The non-technical founder usually has no idea whether their technical co-founder is a programming racehorse or a cart horse.

Unsurprisingly, some are sceptical about the potential of automatic coding. Software tools have always been evolving to harness ever-greater resources with ever-simpler instructions, after all. And there is nothing truly revolutionary out there yet in this nascent field.

Andrew Ng, chief scientist at the Silicon Valley research arm of Baidu, the Chinese internet group, and co-founder of the online learning company Coursera, is one of the sceptics.

He says: “There’s been a lot of work over the past many decades to create better developer tools. On one hand there’s been a lot of progress. On the other, the really hard part about programming is really thinking through what you’d like to express in your software, and this is the part that’s really time-consuming and really hard to automate.”

Not all coding requires such creative expression of ideas, however, and this is where automation is expected to make efficiency gains.

“Yes, there’s a lot about programming that is expressing new and novel concepts, but there’s a lot of very well-paid menial work too,” says [Scott Barstow](http://scottbarstow.com/" \t "_blank), a software development veteran who is researching automated coding. “Where I see the most erosion of value is in the rote tasks that daily occupy junior-level developers.”

Mr Cunniffe concurs. Queue Software began as a software development house building applications for clients ranging from Wall Street traders to start-ups. “We learnt that we were building the same thing over and over, and realised that the more interesting thing to do was to cut out the wasted effort and cost, and work out how to automate 90 per cent of writing code.”

If Queue, Bloomsbury, Bubble and others achieve their goals huge gains ought to follow. Top-flight programmers would still write code but, for many, their role might become one of designing a solution rather than writing the underlying code.

“There’s huge demand for programmers who get paid $160-$200 per hour to write pretty much the same code each time,” says Mr Barstow. “I see so many companies paying outside firms or even in-house teams to start from scratch on apps, which have so much overlap. If you pay a ‘dev house’ typically $75,000-$100,000 or more to build an app, I’d say at times 50 per cent of the money is wasted because the same stuff is being written again and again.”