# Disconnecting for Health

This article is about fallibility and computer systems provoked by the gently bubbling controversy over “Connecting for Health”, the grandly and completely inaccurately titled “biggest computer program in the world ever”, which simmered over once again this week.

It is in the nature of human beings that we cock everything up the first time. We also cock everything up the second time, not always on the third time and so on. Unfortunately, computer practitioners frequently do not learn from this. Building big computer systems is actually surprisingly simple if you accept one premise at the start. You will not get it right first time and almost certainly the second time either. I have been involved in software development projects big and small many times over the years, a reasonable percentage of which have succeeded and I can't recall a single occasion when we actually knew what we were doing at the start. On one or two memorable occasions, we didn't know what we were doing at the end either but that can wait for another month.

The point is that prototyping to understand a problem domain is an intrinsic part of software development, more so than any other engineering profession because we really do not understand what we are doing at the beginning. When I see student projects handed in with a requirements stage, a specification stage, an implementation stage and so on, all carefully partitioned using one or other of the numerous software process models, I honestly weep for them. The real world is almost never so clear. Its not just software development. One example I frequently use is Beethoven's Fifth symphony, the da-da-da-daaaa one, (sorry Ludwig). It is a staggering masterpiece of organisation which uses elements of mathematics in its structure, (its built on the sonata principle with sections of length deliberately very close to the 'perfect ratio' of mathematics).

So how many of you think that it fell out of Beethoven's head in this form ? I hope none. The beauty and awesome structure of the Fifth emerged by systematic refinement. We know this because Beethoven kept early versions which were frequently naïve and relatively formless. The greatness emerged from a process of prototyping. If a software developer had written Beethoven's Fifth, it would have been version 1 and on the all time awful list along with most of the efforts from the Eurovision Thong Contest. Very, very few musicians go with the first version. Software developers nearly always go with the first version even when we knew in experiments in the 1980s that it was a stupid thing to do.

So what's the problem with the NHS development ? Quite simply, we don't know what the hell they are doing and there are few signs of prototyping. The controversy is over whether it should be independently reviewed or not. How could this possibly be controversial ? All such projects should. If they were, and the dictum of 'never right the first time' was followed using open prototyping, we wouldn't have anywhere near as many disastrous projects as we do. Two of the most successful predictors of projects that are going down the pan are a) nobody wants you to see them and b) talking heads tell you there's no problem. Its your money, shout about it.

l.hatton@kingston.ac.uk