

To design or not to design

Design is an essential element in the production of a reasonable engineering system. This may be self-evident perhaps but one of the problems with software systems is that you can't actually see them unless you see the source code. Even then, relatively few people can interpret what they see with any sense of design and yet experienced software developers can recognise good design when they see it.

In contrast, a building can be seen by all and good design has a universality which most people can judge easily. For example, few people would look at St. Paul's cathedral and be disappointed. Moreover, good design extends to all engineering structures as a glance through Brunel's portfolio will reveal. In an engineering structure, good design manifests itself in aesthetic appeal, functional value and longevity. All must be present as there is little value in something pretty which falls down in a couple of weeks. In software systems, good design manifests itself in functional value, reliability, longevity and ease of adaptation to changing demands. A good example is Unix which possesses all of these traits to a greater or lesser extent.

Good design is very difficult to teach to budding software developers because we don't have many ground rules as software development generally is gripped by unconstrained creativity and is driven by fashion rather than engineering. This really shows when you compare almost any flavour of Unix against almost any flavour of Windows. Unix shows all the elements of good design and has aged very well indeed with its latest incarnation, Linux, achieving very wide-spread use from humble web server to powering the world's fastest computers. In contrast, Windows amongst other things, suffers from its success in opening up the world of desktop computing. Locking it down as malicious software has proliferated proves a Herculean task and even patching it successfully is very hit and miss.

Let me expand on this. Two of my children are full-time students. One, an architect, chose a Macintosh, nowadays a Unix derivative, on the advice of his elder brother. The other, a musician, was strongly recommended to get a Windows machine by her Faculty for compatibility with her course. The user experience could not be more different. For the last three years, my son, the Macintosh user, has had an experience very close to my ideal definition of "quiet ownership", a concept deriving from Contract Law. It just works. In contrast, my daughter had been using a Windows XP system for just three weeks when it failed mysteriously "'svchost.exe has failed because of a microscopic black hole near '0xf5ed0ca', spooky, eh ?" - , you know the kind of thing. This little gotcha repeatedly occurred and then actually hung XP when you tried to switch off. My daughter said quite reasonably, "What does this mean ?". After spending a whole morning on the telephone with the support people trying to sort it out, it turns out that an automatic patched patch on a patch had screwed up a central part of the operating system, an essential symptom of poor design. Well, really.

Physicists call this the inexorable growth of entropy. You can slow it but you can't stop it. Will Windows still be a force to be reckoned with in five years ? Laugh if you must but I wouldn't bet on it.

l.hatton@kingston.ac.uk