

D3: Object-Orientation and other mythologies

Les Hatton, lesh@oakcomp.co.uk

Jul 1996

I have been threatening for a little while to put the boot into OO, so Joone seems a good time to write this. This article is therefore an open challenge to the many practitioners of OO out there to prove systematically with appropriate data that OO delivers any of the benefits which have been promised for so long. Let me present my case first.

OO is intended to mirror the way we think and is therefore considered to be self-evidently better than whatever it is we were doing before. In other words, it is once again a triumph of intuition as an alternative to the scientific method. It has generated an entire dictionary of poorly understood new concepts such as polymorphism, inheritance, encapsulation and so on. Enough money has been sacrificed on the altar of this particular belief and it is time that we saw some tangible proof or disproof. Here is some disproof.

My own company happened to have two significant (both > 50,000 lines) but comparable projects, one written conventionally in C and one written using OO design and implementation methods in C++. Both projects were produced by experienced people. We have a complete change and fault history so I decided to analyse these records to measure the corrective maintenance benefits of using C++ instead of C. These benefits are supposed to include more reliable systems and ease of change. After going through over 3000 change records, I extracted the 12% relevant to corrective change to be confronted with the fact that the C++ system has a defect density some 25% MORE than the C system and that each defect took on average twice as long to correct. Even the simpler ones took longer, so the whole distribution is right shifted. In other words, using C++ instead of C has increased our corrective maintenance overhead by nearly 300%. (Note that on average 50% of all maintenance is corrective).

Shortly afterwards, I read Watts Humphrey's admirable Personal Software Process book to find that he included data showing that C++ led to significantly higher correction costs than equivalent Pascal projects. Finally, I have just received further evidence from Professor Walter Tichy at the University of Karlsruhe, an authority on software measurement, which showed exactly the same effect.

When I show this data to OO people, they usually say, "that's because you aren't doing OO right", to which I answer, "show me your data then", to which they answer "we don't have any". My current view is that the data suggests

that OO does not deliver any of its advertised promise and is simply a different rather than better way of doing things. Now I have only seen C++ data and I know that many OO people are devotees of other languages, however there appears to be no data at all for these.

So here is my challenge to OO developers. Please start acquiring data, because so far, things don't look promising for this paradigm, however much fun you're having. See you next month when I return from Japan.