

# The right tool



As a software developer I use a number of different computer languages: Java to create long lived applications; C where portability and performance are crucial; JavaScript on websites; Perl if writing the code fast is more important than maintainability; and so on. Personally I would expect any programmer that claimed to be a “professional” to be able to demonstrate significant experience using more than 10 mainstream languages. They should also be able to intelligently discuss the merits and drawbacks of all the major “families” of languages and explain their choices.

Of course, on this site our topic is how best to improving the technical data held by oil companies, not how to identify good software developers. But, even when we’re improving oil company data handling there is a similar set of choices to be made. There are a number of, I’m not sure what to call them: techniques; disciplines; topics; specialisations; paradigms; or approaches. Each of them combines insights and provides an apparently consistent and comprehensive view of the way the world works, and suggests ways to solve complex issues. The most obvious ones (in no particular order) are: Six Sigma; Big Data; Business Intelligence; Data Science; Information Architecture; Enterprise Architecture; Data Management; Systems Thinking; Taxonomy; Change Management; Maturity Metrics; User Experience Design; Solution Design; Ontology; Game Theory; Chaos Theory; Knowledge Management; Programme Management; Library Science; and Document

Management; to name just a few. All of these have their evangelists, their dewy eyed prophets prepared to patiently explain why all your issues would easily be solved if you would just start applying their particular tools and methods. As an unswerving adherent to the “data management” one I know I am just as guilty of that particular psychosis as anyone.

It is undoubtedly true that every single one of these has proved valuable in understanding and fixing issues that had serious business impacts, somewhere. It is also almost certain that they have all been inappropriately applied and, because they were out of place, have failed to deliver what they initially promised. My own belief is that every specialty on this list has something useful to contribute, in the right circumstances. Any professional “business advisor” should at the very least be aware of their various benefits and limitations, they should know under what circumstances each of them will deliver insights and, more importantly, under what conditions they clearly won’t.

A developer who only knows one way to work (or even worse only one coding language) inevitably ends up trying to cram their latest coding goal into the slots their favourite tool provides, rather than trying to select the best tool for the job. These methodologies are exactly the same, none of them perfectly fit every situation. However comfortable we are in one particular framework we owe it to ourselves to stretch into other less familiar ways of thinking so that we can better identify when our preferred approach is not the best.