

Software Objectives

Keeping Focussed

I've often found that by writing down those fundamental truths which you want to adhere to, helps re-focus you when you get led astray by the technical details of a particularly knotty problem... the sort that absorbs you on a daily basis.

Over the years I've written a few, but a number of them keep recurring. The ones listed below try to look at business, skills, product architecture, process that are important to those of you defining software components of embedded systems (typically in embedded controls). I hope you find them useful... even if only in starting an appropriate train of thought.

Key Objectives

1. To have a well-defined, and highly visible (internal) strategy for partitioning that allows growth and maintenance for current and future capabilities
2. To achieve a highly modular architecture and implementation
3. To protect our corporate Intellectual Property (Knowledge) with respect to system and software
4. To provide the customer benefits that :-
 - allow a 'tailored' response to customer needs
 - provide a consistently high reliability software product
 - provide a rapid time-to-market
 - are seen to be forward-thinking to accommodate future industry trends
 - create confidence in us as a true 'world-class' software capability
5. To provide the corporate and divisional benefits of :-
 - maximised re-use in both system design and software implementation
 - easily and economically able to accommodate variant system requirements
 - easily and economically able to accommodate variant system implementations
 - economically transportable across different target platforms
 - minimise the software development and whole-life software costs
 - maximise independence from hardware platform / supplier
 - maximum understanding within the system and software teams
6. To have a recognisably disciplined software process that enables low-risk highly-repeatable software development of a developing product and its product variants
7. To have a common policy that enables us to give an excellent corporate software quality image across its product range for maximum marketing opportunities
8. To provide easily estimated and deterministic software performance for all components of the target system or product
9. To maximise the opportunities for 'value-engineering' product without attendant software cost and risk
10. To enable maximum advantage from supplier partnerships/relationships