Reflections on Product Quality

On Management and Product Quality

Be very careful of what you wish for – you might get it

If you want a lower component cost, you might get it – at the expense of an increased service and guarantee cost.

Reorganization never solves anything – it just adds to the confusion

First observed in writing 400 BC by the governor of a Greek province.

Product failures due to user errors are just as real as product failures due to equipment fault

Remember to budget for understandable manuals and idiot-proof human interfaces.

The purpose of project documents is to ensure product quality, not to satisfy some administrative need.

Insist on quality in the project documents from the project staff. Purely administrative documents are your responsibility and should never be delegated.

Discourage "quick fixes".

The only "quick" thing about it is the introduction of new bugs.

Advice to a Manager in a Stressed Situation

If you can keep your head when all about you
    Are losing theirs and blaming it on you,
If you can trust yourself when all men doubt you,
    But make allowance for their doubting too;
If you can wait and not be tired by waiting,
    Or being lied about, don't deal in lies,
Or being hated, don't give way to hating,
    And yet don't look too good, nor talk too wise:

If you can dream-and not make dreams your master;
    If you can think-and not make thoughts your aim;
If you can meet with Triumph and Disaster
    And treat those two imposters just the same;
If you can bear to hear the truth you've spoken
    Twisted by knaves to make a trap for fools,
Or watch the things you gave your life to, broken,
    And stoop and build 'em up with worn-out tools:

If you can make one heap of all your winnings
            And risk it on one turn of pitch-and-toss,
And lose, and start again at your beginnings
            And never breathe a word about your loss;
If you can force your heart and nerve and sinew
            To serve your turn long after they are gone,
And so hold on when there is nothing in you
            Except the Will which says to them: "Hold on!"

If you can talk with crowds and keep your virtue,
            Or walk with Kings-nor lose the common touch,
If neither foes nor loving friends can hurt you,
            If all men count with you, but none too much;
If you can fill the unforgiving minute
            With sixty seconds' worth of distance run,
Yours is the Earth and everything that's in it,
            And-which is more-you'll be a Man, my son!

- Rudyard Kipling

On Hardware Quality

It is impossible to test quality into a product. Quality must be designed into the product.

The hardware designer is always responsible for the production yield. The purpose of production tests is to check *production* quality, not *product* quality.

When the prototype fulfils all marketing requirements, you are only 40% done

The remaining 60% is spent transforming the prototype into a producible product.

Analog Design Advice

Always treat “analog ground” as a signal.

The only connection between analog ground and power ground/digital ground should be through one resistor. Design for a 100 ohm resistor and mount a 0 ohm resistor.

Always expect that external connections will introduce ground loops if possible

Remember that a varying magnetic field will induce an electromotive force in the ground loop. This force is proportional to the area of the loop. Remember to check for ground loops internally (on PCBs, through wiring harnesses etc.).

Shielded cables are no help against electromagnetic interference.

Use balanced transmission and twisted pairs of wires to eliminate electromagnetic interference.

Digital Design Advice

Digital signals must be treated as analog signals when sent over long lines

A long line is a wire or trace that is longer than $l_c = t_r \cdot c \cdot n$, where t_r is the minimum rise time of the signal, *c* is the speed of light and *n* is the correction factor for physical wires (usually about 0.67).

Always terminate long lines

A long line is defined above. The necessary terminating resistor is usually in the range of 100 – 120 ohm.

Be extremely careful when moving between clock domains

Setup and hold times can usually not be guaranteed when doing so.

On Software Quality

*It is impossible to **test** quality into a product. Quality must be **designed** into the product.*

!

Module Design Advice

Always document all available entry points into the module in the module design document and have it reviewed by the rest of the project software staff.

The first and foremost requirement for a piece of code is that it is readable. Obscure and tricky code is impossible to maintain and often contains hard-to-find bugs.

"C" Programming Advice

Always declare functions and variables that are internal to your module as "static"

That way, nobody else can bring havoc to your internal module structure.

If at all possible, never use or publish global variables

If somebody wants to know the value of one of your internal variables, write a "member" function to return an official value of that variable.

Understand and use type qualifiers (const and volatile)

The **const** qualifier tells the compiler that the variable should not be modified. The **volatile** qualifier tells the compiler that the variable may change outside the current compiler scope. All global variables should be qualified as **volatile**.

From MSDN:

- The **const** keyword can be used to modify any fundamental or aggregate type, or a pointer to an object of any type, or a **typedef**. If an item is declared with only the **const** type qualifier, its type is taken to be **const int**. A **const** variable can be initialized or can be placed in a read-only region of storage. The **const** keyword is useful for declaring pointers to const since this requires the function not to change the pointer in any way.

- The compiler assumes that, at any point in the program, a **volatile** variable can be accessed by an unknown process that uses or modifies its value. Therefore, regardless of the optimizations specified on the command line, the code for each assignment to or reference of a **volatile** variable must be generated even if it appears to have no effect.
- If **volatile** is used alone, **int** is assumed. The **volatile** type specifier can be used to provide reliable access to special memory locations. Use **volatile** with data objects that may be accessed or altered by signal handlers, by concurrently executing programs, or by special hardware such as memory-mapped I/O control registers. You can declare a variable as **volatile** for its lifetime, or you can cast a single reference to be **volatile**.
- An item can be both **const** and **volatile**, in which case the item could not be legitimately modified by its own program, but could be modified by some asynchronous process.

All code should be passed through lint. Every remaining lint warning should be commented.

The lint code checker is an excellent tool for discovering subtle bugs in the code.

Common “C” Pitfalls and How to Avoid Them

Always test pointers for NULL before using them.

Otherwise you are likely to either mess up the interrupt vector table or cause a fatal access error trap.

Be extremely careful when decrementing unsigned types

If an unsigned variable has the value 0, decrementing it will result in a large positive value!

Always use parentheses in complex expressions

Otherwise the compiler will interpret the expression according to the precedence rules – and those are not obvious.