QinetiQ use of Java

- Java is not used for real time embedded software.
  - C or C++ is used instead.
- Java used for Web applications and Desktop applications.
- QinetiQ Java Coding Standard exists.
  - Essentially a style guide for manually developed code.
  - Is not machine checkable, enforced by process and audit.
- Consists of **Must** and **Must not**, **should** and **should not**, and **may**
  - **Must** is mandatory, but can be deviated from with agreement of QinetiQ's Software Advice Service
  - **Should** is a recommendation, reasonable exceptions can occur in practice
  - **May** expresses optional guidelines.

Examples

- **Must**
  - Naming conventions
  - Floating point equality /inequality tests must not be used
  - Must be a default clause in a switch statement
  - Parameters must not be modified within a method
  - A return statement must not be used in a try or finally block
- Mixture of rules for clarity, predictability and defensive measures
- **Should**
  - Conditional expressions should not produce any side effects
  - A method returning non-void should have a single return statement
  - Methods should not return handles to mutable internal data
  - All read and write accesses to double and long variables should be declared as synchronized ...

Conclusions

- Coding Standard is useful but rationale between musts and shoulds seems a little arbitrary.
- Predictability and verifiability is not the rationale for the Coding Standard.
- Rules are mainly restrictions on the use of Java rather than a coherent subset.
- Real time requirements are only partially addressed.
- Machine checking, and consistent policy is a weakness.
- However no clear demand for a high integrity subset simply because of the nature of QinetiQ’s business at present.
- There is a demand elsewhere in the world.