Using Static Analysis in Medical Device Development

Chao Wang
Medtronic CRDM

July 23rd, 2009

Facts of Defects

(Applied Software Measurement by Capers Jones)
Capabilities and Value of Static Analysis

- Examining the code (100% path coverage) without executing the program—a white box approach. Catches problems that test suites may miss.

- Revealing errors that do not manifest themselves in testing process until an unusual set of conditions met (often after release). Catches bugs early, when they are less expensive to fix.

- Automation of code review, coding standard (Power of 10, MISRA, etc.) enforcement and reliability metrics. Pinpoints defects automatically, improving productivity.

Medical Device Industry Landscape – Static Analysis

- Medtronic
- GE Healthcare
- Cardinal Health
- Johnson & Johnson
- Abbott Laboratories
- Stryker Medical
- Philips Medical
- Schiller
- Cerner
- Baxter Healthcare
- Hoana Medical
- Zoll
- ...
Is White-box Approach Necessary

• Black-box testing is effective

• Black-box testing is NOT sufficient
  – Limited test cases
  – Effort / time constraints
  – Hard to exhaust all negative cases
  – Unintended uses

Static Analysis Methodologies vs. Tools

• Foundation
  – Data flow (symbolic execution)
  – Pattern match

• Practice
  – Code review
  – Test readiness review
  – ...

• Tool maturity
  – Evolving vendor
  – Technology progression
  – Growing needs
Static Analysis Tool Effectiveness

• False negative
  Out of 29 pre-seeded bugs

• Overall robustness:
  90–95%

Static Analysis Tool Effectiveness (cont.)

• Worst cases:
  – TestSpec - 01 - Expression always evaluates to False
  – TestSpec - 12 - macro not parenthesized
  – TestSpec - 15 - suspicious constant
  – TestSpec - 42 - fail to leave Critical Section
  – TestSpec - 43 - fail to enter Critical Section
Tool Efficiency

- False positive
  - Ability to strike the right “balance” point is important

- IDE integration (work flow)

- How to deal with inline assembly code

Usability

- What to look for
  - Reporting
  - Tracking
    - Code relationship
    - Coding history
  - Admin
  - Customizing
    - Enable/disabling checkers – group or individual
    - Additional checkers
  - Others (will discuss later)

- Take it seriously
  - The third factor after false positive/negative rates
  - When false positive/negative rates are all in acceptable range, usability becomes critical to consider.
Integrating into Dev Environment

- A seamless component of development
  - Desirable use model
  - Bug tracking system
  - Code depository system
  - ...

- Overkill?
  - Maybe
  - In integration test (before release) only? So no "disturbance" in code development at all.
  - Another approach: Nightly build / Batch mode

Roles in Design Review?

- It is arguable, but we will not intentionally use “core” static analysis in our software design review.
- While static analysis is good at finding logical errors, it is not “sensitive” to design mistakes.
- Extended capabilities may help in the design review
Roles in Verification / Validation

- Supplement not replacement of verification test
- Fit well in Test-driven Development (TDD)
- Not used in validation

Roles in Cost Cutting

- Code inspection
  - Take the burden away from developers of tedious and repentant code checking.
- Code Review
  - Reduce/eliminating coding error so code review can focus on correctness of implementing specifications.
- Regression test in incremental releases
  - Speed up the regression test
Example

```c
// clear bit 7
#define BIT_MASK 0x80
unsigned int my_val = 0xffff;

// incorrect code using logical negation operator
my_val &= !BIT_MASK;

// correct code using bit-wise complement operator
my_val &= ~BIT_MASK;
```

(Timmerman)

Roles in helping development

- Reliability metrics
  - Code complexity
  - "Dead" code
  - RCA
- Unit test
- Architecture analysis
Roles in helping development (cont.)

- **IEC 62304 Compliance (Medical Device Software – Software Life-cycle Process)**

  (Tansey, Madan)

- **Promote best practices in software development**

- **Raise the bar of acceptable coding habits**
Design for Reliability and Manufacturability (DRM) in SW

Past

VT is separated as ‘end of cycle’ activity

Current

Coding and VT processes start to integrate

Future

Minimize/eliminate code defects before VT

Special Thanks

- VJ Jagannathan
- Ken Timmerman
- Randy Wells
- Andrey Madan
- Kevin Tansey
- Robert Schlafmann
- Kevin Climisch

- Coverity
- GrammaTech (CodeSonar)
- Klocwork
- Parasoft
Questions?