# Boot and Power Management Processor Safety

Goutham Deva, Robert Harrison, and Tejas Rao
27 February, 2019
NVIDIA - Tegra Group
Sponsor: Matt Longnecker

# Background on the BPMP firmware

- Tegra Chip
- Boot Power Management Processor

# Automotive Safety Standards

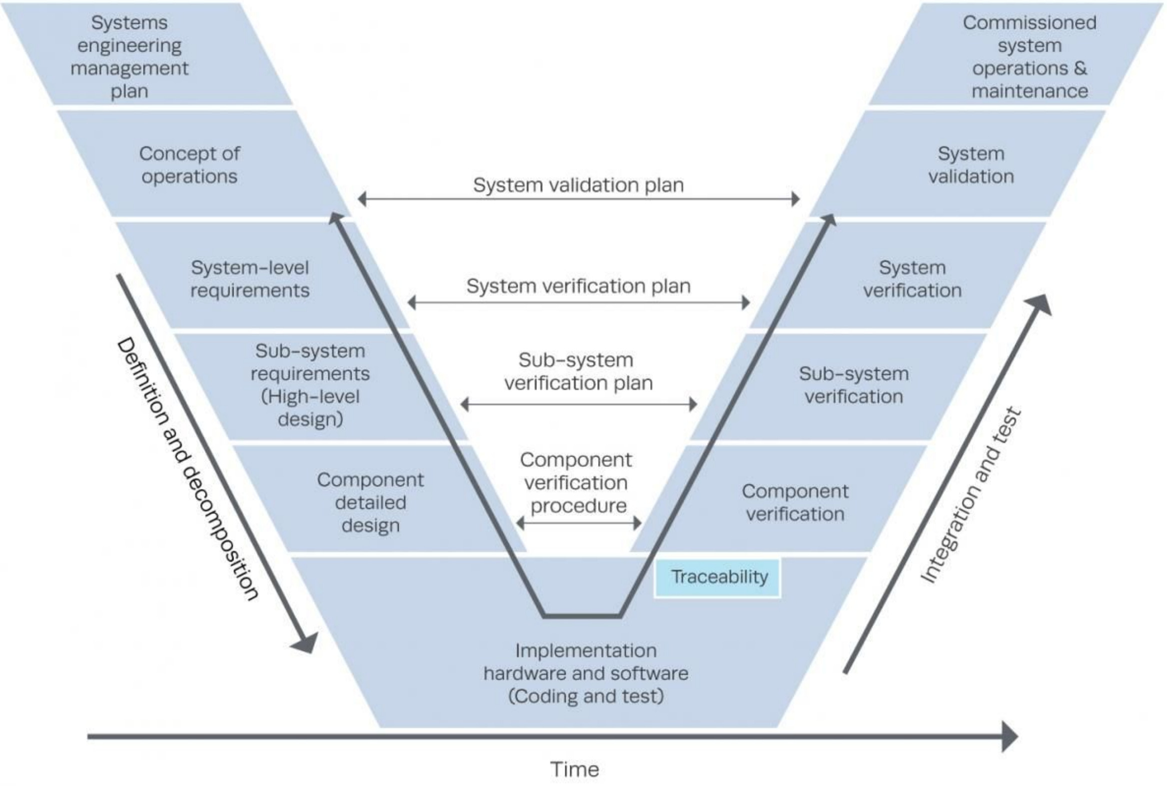
## Development Guidelines

- *ISO 26262*
- *Automotive SPICE*

## Code Safety Standards

- *MISRA C:2012*

# Verification and Validation model


Systems engineering management plan
Concept of operations
System-level requirements
Sub-system requirements (High-level design)
Component detailed design
Implementation hardware and software (Coding and test)
Component verification
Sub-system verification
System verification
System validation
Commissioned system operations & maintenance
System validation plan
System verification plan
Sub-system verification plan
Component verification procedure
Traceability
Definition and decomposition
Integration and test
Time

# Problem Statement

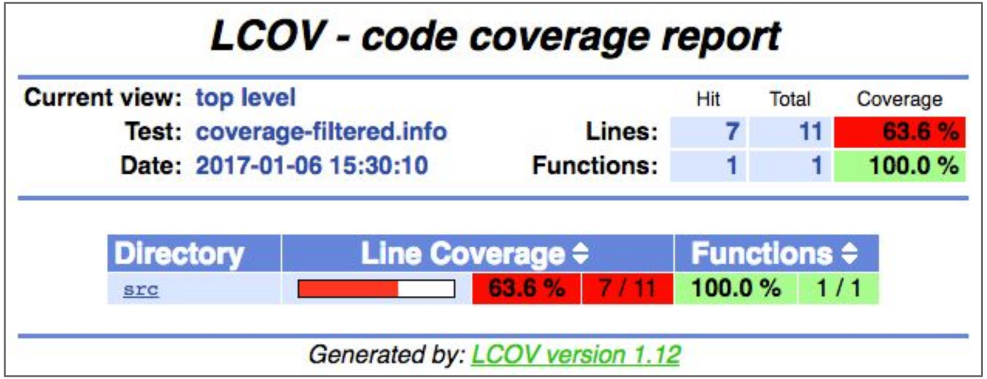
*Assist the Tegra System Software team with achieving full compliance in order to minimize the likelihood of a fault in the code causing software failure in production.*

# Procedure

1. Write Unit Tests
2. Refactor Files
3. Clean MISRA Violations
4. Create Software Design Documents

# 1. Write Unit Tests

- Internal Unit Testing Framework
- Device tree source files
  - Stores expected info about hardware relevant to Operating System
- Tested both coverage and requirements

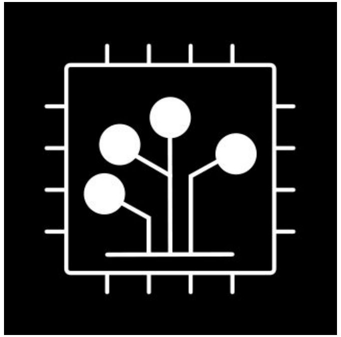


**LCOV - code coverage report**

|  |  |  | Hit | Total | Coverage |
|---|---|---|---|---|---|
| Current view: | top level |  |  |  |  |
| Test: | coverage-filtered.info | Lines: | 7 | 11 | 63.6 % |
| Date: | 2017-01-06 15:30:10 | Functions: | 1 | 1 | 100.0 % |

| Directory | Line Coverage ⇕ |  |  | Functions ⇕ |  |
|---|---|---|---|---|---|
| src |  | 63.6 % | 7 / 11 | 100.0 % | 1 / 1 |

Generated by: LCOV version 1.12



# 2. Refactor Files

- Code not in production doesn't need to be marked safe
- Separate debugging interface so it is not mixed with production code
  - Create submodules for the interface
  - Move submodules into a separate project
  - Remove all references to the debugging interface in the driver code

# 3. Clean MISRA Violations

- Eliminate Undefined Behavior
- BPMP originally not compliant

Violation:

```
uint x = 0;
if(!x)
    printf(“x is 0\n”);
```

Cleaned code:

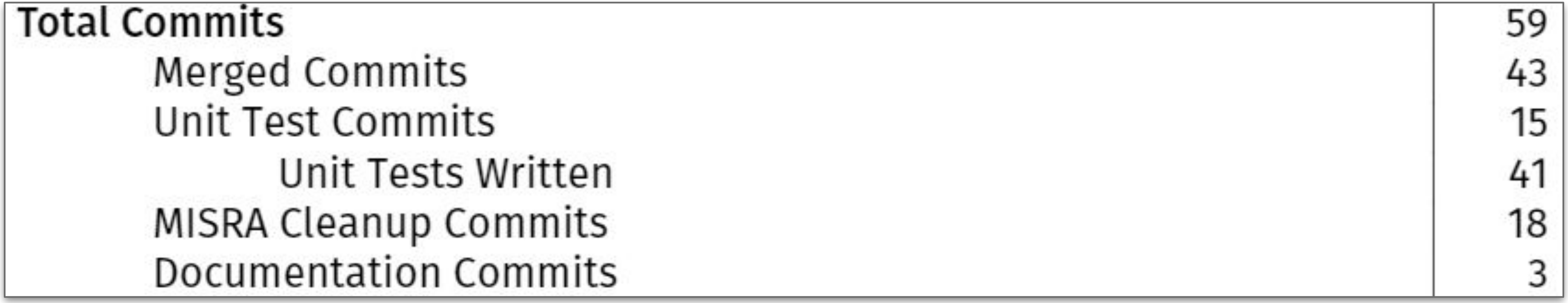
```
uint32_t x = 0U;

if(x == 0U) {
    printf(“x is 0\n”);
}
```

# 4. Write Software Design Documents

- Formated in Asciidoc
- Documentation
  - **Architecture Design**
    - *High Level Design*
  - **Module Design**
    - *Low Level Design*

# Cumulative data on commits merged into the BPMP Code Base



| | |
|---|---|
| Total Commits | 59 |
| Merged Commits | 43 |
| Unit Test Commits | 15 |
| Unit Tests Written | 41 |
| MISRA Cleanup Commits | 18 |
| Documentation Commits | 3 |

# Conclusion and Final thoughts

- Ensuring safety in software systems is hard.
- Our team has contributed a lot to making BPMP firmware safer

# Future work to be done

- Write design documentation
- Clean more MISRA files
- Write more unit tests

# Questions?

# References

"Asciidoc-Preview." Atom, Asciidoctor, atom.io/packages/asciidoc-preview.

"Introducing NVIDIA® Tegra® 4, The World's Fastest Mobile Processor." NVIDIA, www.nvidia.com/object/tegra-4-processor.html.

Jones, Tim. "Future Trends Presentation Tim Jones Ppt 97." LinkedIn SlideShare, 18 Dec. 2008, www.slideshare.net/timjones72/future-trends-presentation-tim-jones-ppt-97-presentation.

Mitre (2014). Systems Engineering Guide. McLean, VA, The Mitre Corporation.

"Self-Driving Cars Technology & Solutions from NVIDIA Automotive." NVIDIA, NVIDIA, www.nvidia.com/en-us/self-driving-cars/.