



延锋伟世通
YANFENG VISTEON

如何使用 Polyspace 提升软件开发过程中的质量

张歆钰, 南京延锋伟世通



MATLAB EXPO

延锋伟世通核心产品



Cluster



CDC



HUD



Display



Infotainment




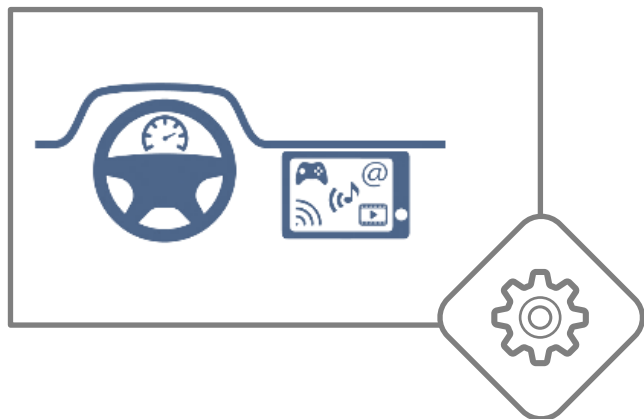
V2X




BMS



两个故事



问题：随机重启
投入：5个资深工程师
时间：3个月
代价：搞坏一辆车的换挡
改动：一个数字



问题：仪表里程丢失
投入：3个资深工程师
时间：2周
原因：一个非常基础的问题



软件开发过程中的痛点

莫名其妙的问题

这个问题好奇怪，概率好低，不知道怎么复现，跑着跑着就出来了

---- 这么低级的错误，原来是边界值没有做保护

哎呀，怎么这么多千奇百怪的问题，随机发生

---- 原来是对数组操作越界了，把随机值写入了后面定义的变量里

当然是我自己的代码写的最好

---- 所有人都认为自己的写代码最好，通俗易懂又美丽，最终导致同一个项目里的代码各有各的风格

怎么办呢？

什么是静态代码扫描

静态代码扫描，学名：静态程序分析(Static program analysis)，在不执行代码的情况下分析源代码的质量

— 代码质量左移的最佳方案

01

优点

- 不需要编译通过代码, 仅通过扫描代码发现问题, 可以跳过不同语言严苛或复杂的编译过程, 更迅捷的发现问题;
- 通过规则自动化扫描代码, 便可以直接检测出常见逻辑错误, 空指针, 内存泄漏等常见问题, 问题覆盖面广, 代码覆盖率高;
- 降低人工review代码的工时成本, 自动化的发现bug和潜在问题。

02

缺点

- 只能发现常规简单的逻辑错误, 而不能发现业务相关的逻辑问题, 仍然不能彻底解放人工Code Review;
- 扫描规则过细会导致扫描出的非高优问题过多而增加不必要的工作量, 但是此问题可以通过问题分级或问题过滤等手段进行规避和改进。

静态代码扫描规则类别

空指针检查 (Null Pointer)

数据越界 (Buf Over Run)

内存泄漏 (Mem Leak)

逻辑错误 (Logic)

可疑代码检查 (Suspicious)

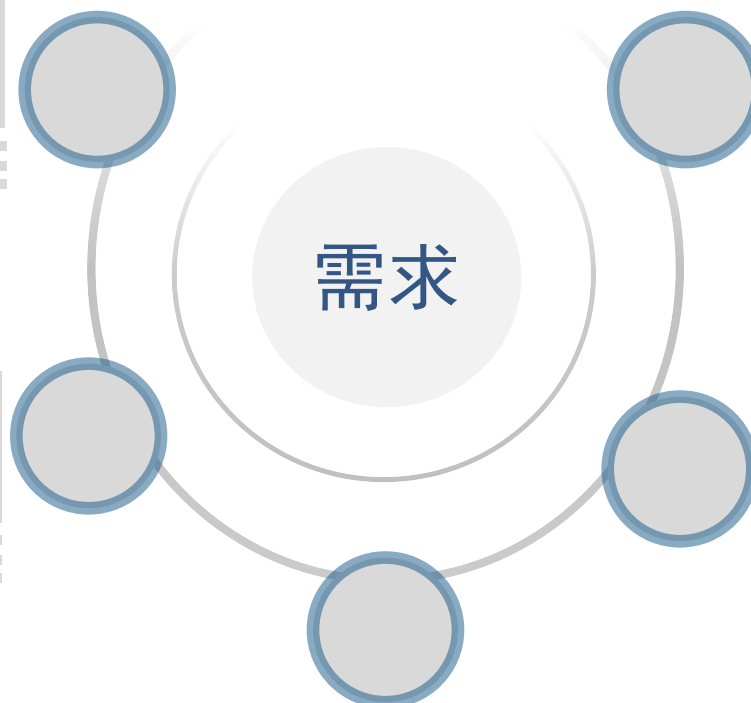
运算错误 (Compute)

高危函数 (Unsafe Func)

未初始化 (Uninit)

发现软件中潜在的缺陷，避免问题到集成阶段

提高集成阶段问题定位的排查的效率



26262和ASPICE认证要求：
静态分析、代码度量和形式化验证

公司内部对质量规范的要求：
C Guidelines

行业标准要求：
MISRA C:2012 、 MISRA C++:2008

Polyspace能做些什么

Polyspace Bug Finder



查找问题



代码规范

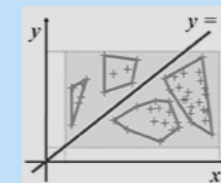


代码度量

Polyspace Code Prover



证明不存在关键运行时错误



没有漏报



所有输入、数据和控制流检查



编写代码



集成



认证



评审

Polyspace能做些什么

查找问题并强制执行编码标准



问题类型

- ❖ Numerical
- ❖ Tainted Data
- ❖ Security
- ❖ Cryptography
- ❖ Data Flow
- ❖ Concurrency
- ❖ Static Memory
- ❖ Dynamic Memory
- ❖ Good Practice
- ❖ Performance
- ❖ Resource Mgmt.
- ❖ Programming



编码标准

- ❖ MISRA C:2004
- ❖ MISRA C:2012
- ❖ CERT C
- ❖ MISRA C++:2008
- ❖ AUTOSAR C++-14
- ❖ CERT C++
- ❖ Naming Rules
- ❖ JSF AV C++
- ❖ ISO/IEC TS 17961

Polyspace能做些什么

监控质量指标和趋势



代码度量

- ❖ Complexity
- ❖ Recursions
- ❖ Language Scope
- ❖ Function Coupling
- ❖ HIS
- ❖ Paths, Inputs, Calls
- ❖ Project
- ❖ File
- ❖ Function
- ❖ Stack Usage

Polyspace能做些什么

证明不存在Critical Run-time Errors和安全漏洞



Examples of
Critical Run-time
Errors

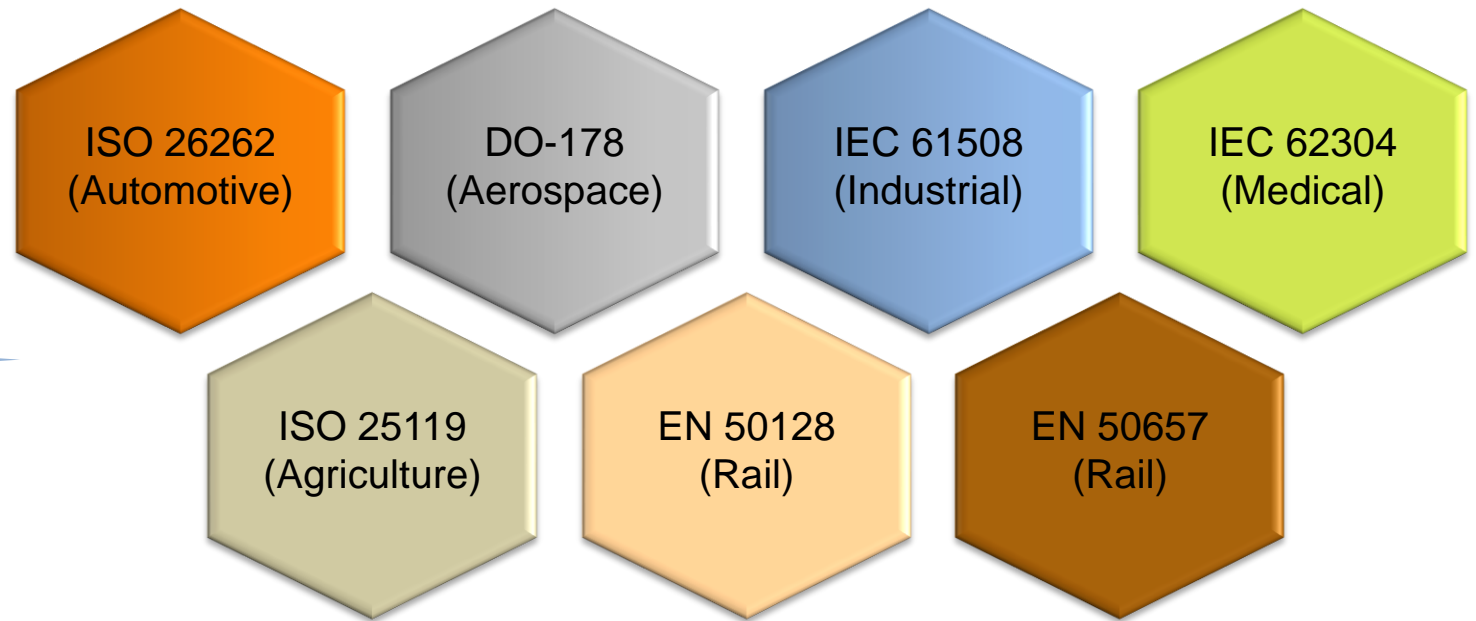
- ❖ Divide by Zero
- ❖ Buffer Overrun
- ❖ Overflow,
Underflow
- ❖ Dead Code
- ❖ Illegal Pointer
Dereference
- ❖ Assert
- ❖ Uninitialized variable
- ❖ Concurrent
access

Polyspace能做什么

支持功能安全和安全标准



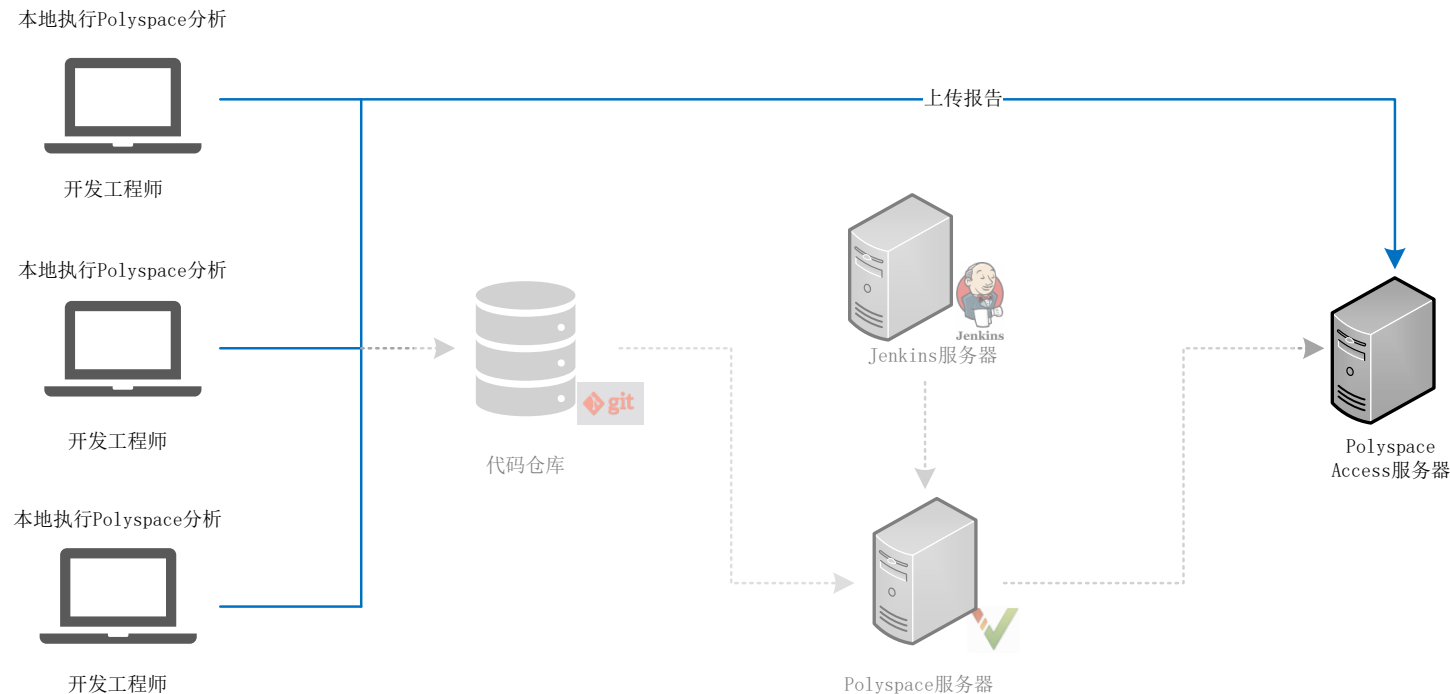
安全标准



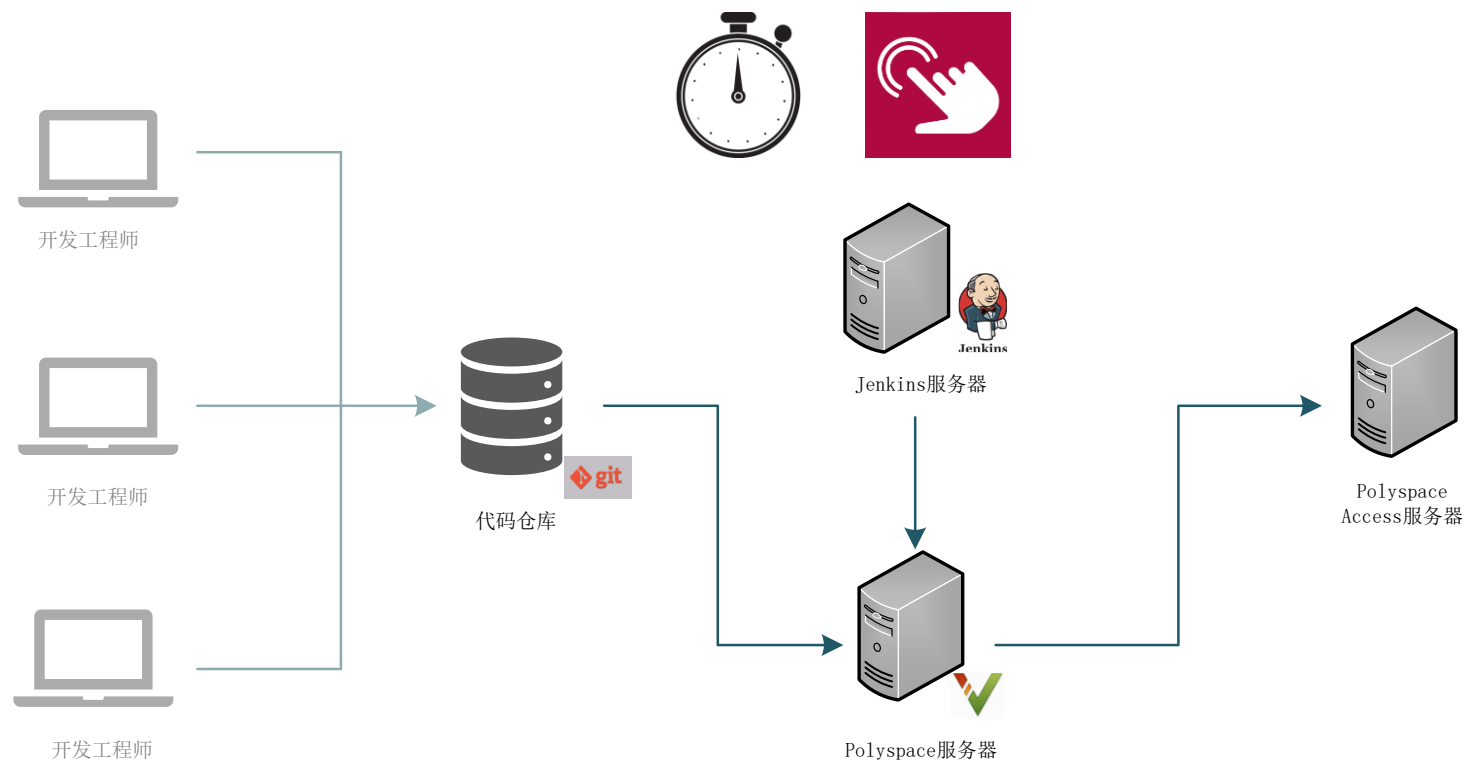
and more ...

实际应用场景——手动扫描

- **提交前分析。**开发工程师修改代码后，在本地桌面执行Bug Finder/Code Prover，避免自己负责的模块的问题带入到版本库。
- **结果基线。**开发工程师把本地分析结果上传至报告服务器。

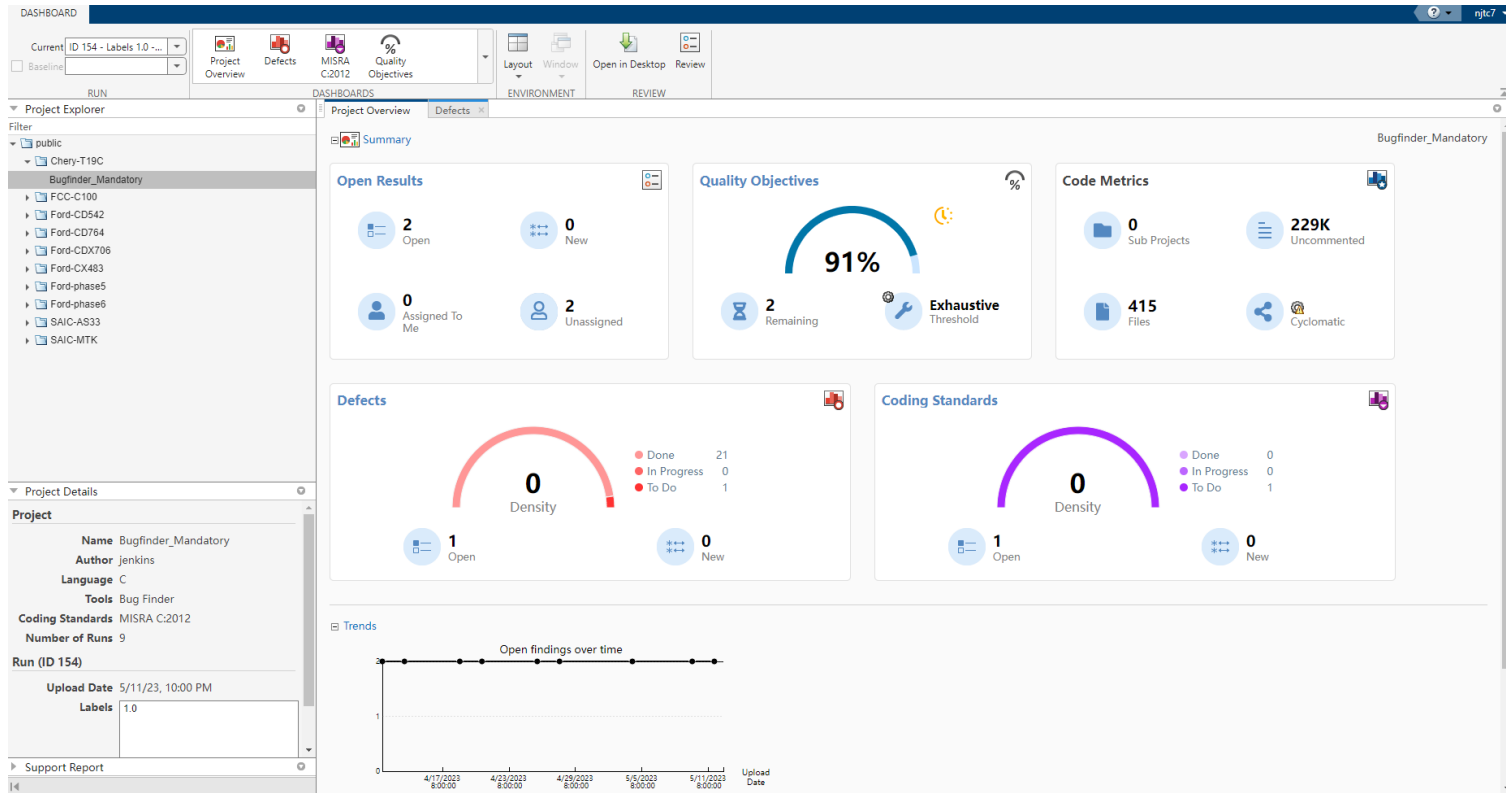


实际应用场景——自动扫描和持续集成



- 开发工程师把代码上传到代码仓库。
- 周期触发：每天凌晨自动执行。
- 按需分析：里程碑阶段和版本发布前执行
 - 软件工程师多，版本变化频繁，集中分析降低分析负载
- 构建过程：Jenkins 从代码仓库克隆整个项目的代码，发起编译。触发 Polyspace 服务器执行项目配置，而后执行 Bug Finder 和 Code Prover 分析。
- 成功执行，生成 Word 版报告，上传结果至 Polyspace Access，导出 CSV 问题列表

实际应用场景——获取并查阅报告



- **Access浏览器结果**——交付经理/质量工程师查阅每个项目的状态。
- **Word版报告**——版本发布时作为存档的报告。
- **CSV问题列表**——软件经理按文件分配问题修复任务。

案例1 MCU看门狗复位

```

208
209 if (AES_set_decrypt_key(key, 128, &aes) <= 0)
210 {
211     return 0;
212 }
213
214 CRYPTO_cbc128_decrypt(in, buf, len, &aes, iv, (block128_f)AES_decrypt);
215 // uart_printf("buf[len] = %d\t len = %d\n", buf[len-1], len);
216 memcpy(out, &buf[0], len-buf[len-1]);
217 return len-buf[len-1];

```

Overflow
 Unproven: operation [-] on scalar may overflow (result strictly lower than MIN_INT32)
 This check may be an issue related to unbounded input values
 If appropriate, applying DRS to len (argument number 4 of function aes_cbc_decrypt, defined in aes_cbc.c line 198 and called by me
 operator - on type int 32
 left: full-range [-2³¹ .. 2³¹-1]
 right: 1
 result: [-2³¹ .. 2147483646 (0x7FFFFFFE)]
 (result is truncated)

Event	File	Scope
1 Initialization of len by the main generator (argument number 4 of aes_cbc_decrypt)	aes_cbc.c	_main_gen_call_aes_cbc_c
2 Not entering if statement (if-condition false)	aes_cbc.c	aes_cbc_decrypt()
3 Not entering if statement (if-condition false)	aes_cbc.c	aes_cbc_decrypt()
4 Unproven: operation [-] on scalar may overflow (result strictly lower than MIN_INT32)	aes_cbc.c	aes_cbc_decrypt()

```

208
209 if (AES_set_decrypt_key(key, 128, &aes) <= 0)
210 {
211     return 0;
212 }
213
214 CRYPTO_cbc128_decrypt(in, buf, len, &aes, iv, (block128_f)AES_decrypt);
215 // uart_printf("buf[len] = %d\t len = %d\n", buf[len-1], len);
216 memcpy(out, &buf[0], len-buf[len-1]);
217 return len-buf[len-1];
218 }

```

Invalid use of standard library routine
 Warning: function 'memcpy' is called with possibly invalid argument(s)
 • Checks on first argument (destination):
 ✓ Not null.
 ? May not be a memory area that is accessible within the boundary given by the third argument.
 Actual value of first argument (pointer to void): points at offset 2²⁶-1 in buffer of 134217726 bytes.
 Actual value of third argument (unsigned int 32): [0 .. 1111] or [4294967042 (0xFFFFFFFF) .. 2³²-1]
 • Checks on second argument (source):
 ✓ Not null.
 ? May not be a memory area that is accessible within the boundary given by the third argument.
 Actual value of second argument (pointer to const void): points at offset 0 in buffer of 1111 bytes.
 Actual value of third argument (unsigned int 32): [0 .. 1111] or [4294967042 (0xFFFFFFFF) .. 2³²-1]
 • Other checks:
 ✓ First argument and second argument do not have overlapping memory.

- 问题代码中访问了buf[len-1]，如果len=0，此时会访问到buf[-1]的值。
- 问题代码中使用到了标准库函数memcpy，在使用此库函数时未对三个形参进行保护，存在访问空指针，或者copy越界的情况，结合len-buf[len-1]的写法，概率性的会导致值异常，例如出现赋值的情况，结合memcpy的源码实现，源码中使用While(len -- !=0)的方式进行字节赋值，所以理论存在最大赋值长度为2³²-1，此时造成了MCU的栈(4KB)越界。

案例2 又见数组越界

The screenshot displays a code editor window for `i2c_queue.c` with the following code snippet:

```

427
428 gpio_i2c_req.dev_id = I2C_DEV_NUM;
429 gpio_i2c_req.p_buf = NULL;
430 gpio_i2c_req.tx_size = 0;
431 gpio_i2c_req.rx_size = 0;
432 gpio_i2c_req.rd_wr = I2C_WR_REQ;
433 gpio_i2c_req.p_next = NULL;
434
435 for(i = 0; i < I2C_MAX_REQ_SIZE; i++)
436     for(i = 0; i < (0x0C); i++)
437     {
438         (void)memcpy(&gpio_i2c_request_buf[i], &gpio_i2c_req, s
439     }
440
441 // (void)hal_i2c_open(&iic_handle, HAL_I2C_QUEUE);
442
443
444
445 static i2c_drv_data_st* gpio_i2c_get_queue_request(void)
446 {
447     i2c_drv_data_st *i2c_req;
448
449     if (gpio_i2c_req_num > 0)
450     {
451         i2c_req = gpio_i2c_q_head;
452         gpio_i2c_q_head = gpio_i2c_q_head->p_next;
453         gpio_i2c_req_num--;
454     }
455     else
456     {
457         i2c_req = NULL;
458     }
459
460     return (i2c_req);
461

```

The error message in the Result Details pane reads:

Non-terminating loop
 The loop is infinite or contains a run-time error.
 Loop fails due to a run-time error (maximum number of iterations: 6).

Event	File	Scope
1	Iterating on loop	i2c_queue.c
2	Function 'memcpy' is c...	i2c_queue.c
3	Non-terminating lo...	i2c_queue.c

- for循环中循环次数为I2C_MAX_REQ_SIZE(0x0C 12) 而memcpy的数组gpio_i2c_requetset_buf[]定义的大小为GPIO_I2C_MAX_REQ_SIZE(0x05 5)。
- 数组越界会导致该BUFF定义在RAM中的地址后续变量的值被篡改，影响较轻则会影响功能异常，更重的情况是导致上层某些服务crash，最为严重可能会导致程序跑飞等。

案例3 一个 break 引发的问题

The screenshot displays the MATLAB IDE interface with a 'Results List' on the left and 'Result Details' on the right. The 'Results List' shows a table of errors, with the selected error being 'Missing break of switch case 5' in the file 'hal_gpio.c'. The 'Result Details' pane shows the error message: 'Missing break of switch case (Impact: Low) Switch case terminates without a break statement or comment about the intentional fall through.' Below this, a table lists the event details:

Event	File	Scope	Line
1	Missing break of switch case	hal_gpio.c	hal_gpio_init() 163

The 'Source' window shows the code for 'hal_gpio.c' with a red box highlighting the following code block:

```
163 case HAL_IO_4G_USB_B001_EN:  
    IO_4G_USB_B001_EN_INIT();  
164  
165  
166 case HAL_IO_VA33_VIO33_3V3_EN:  
    IO_VA33_VIO33_3V3_EN_INIT();  
167  
168     break;  
169  
170 case HAL_IO_VCORE_1V0_EN:  
    IO_VCORE_1V0_EN_INIT();  
171  
172     break;  
173
```

- 由于缺失了一个break，导致一个供电逻辑异常，最终导致部分外设启动后无法正常工作。

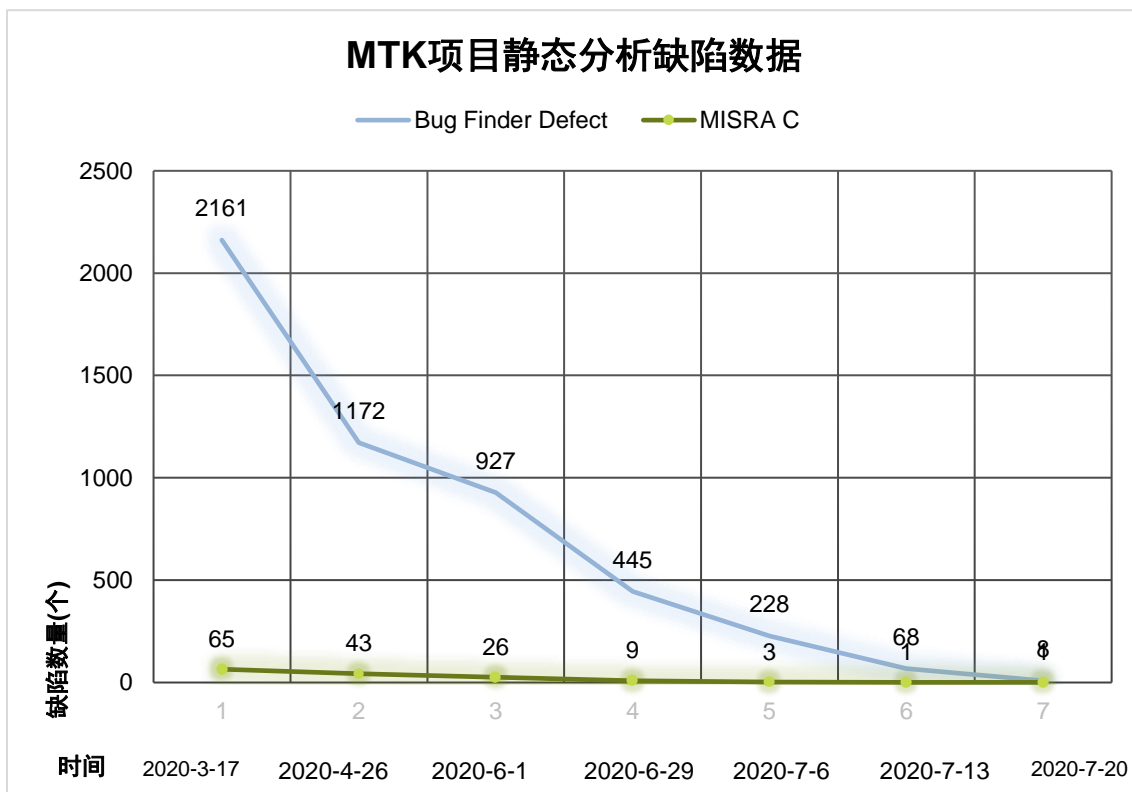
成效显著

➤ 提升了调查问题的效率

不含第三方软件的逻辑，调查不会再出现超过1周的情况（之前5个工程师，3个月时间）

➤ 大大改善了软件质量

➤ 支持通过了 ASPICE CL3 和 ISO 26262（ASIL D）



MATLAB EXPO

Thank You



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.