

# Implement Best Practices in your Software Factory to Improve DevOps Metrics

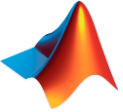
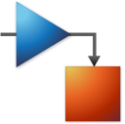

October 20, 2022 | Stuttgart

Tjorben Gross

Skanda Naglapur Ramamurthy



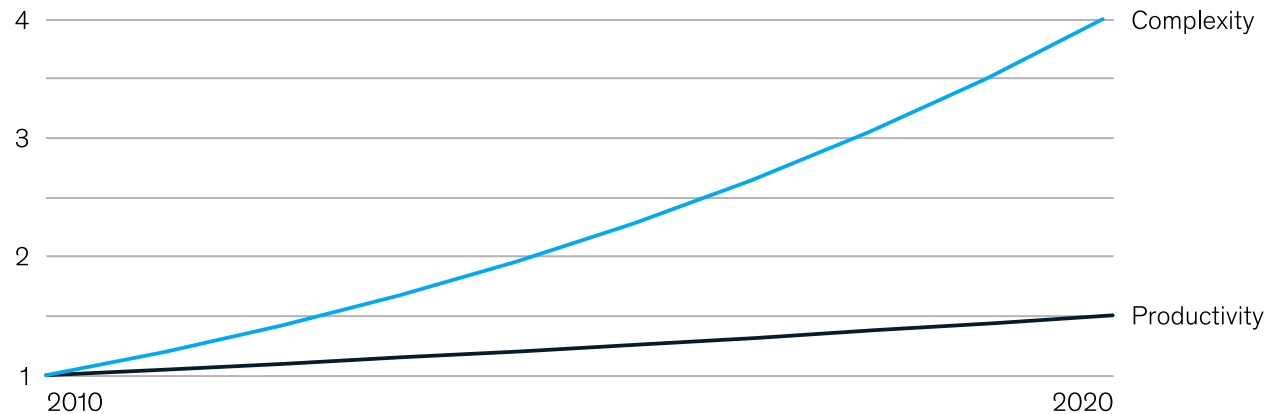
# Agenda

- ➔ Current challenges in the automotive industry
- How challenges are addressed using SW Factory and DevOps
- Establish a mature issue detection process
  -   
- Conclusion

# The Challenge: Mastering the Digital Transformation

Software complexity is increasing more quickly than productivity.

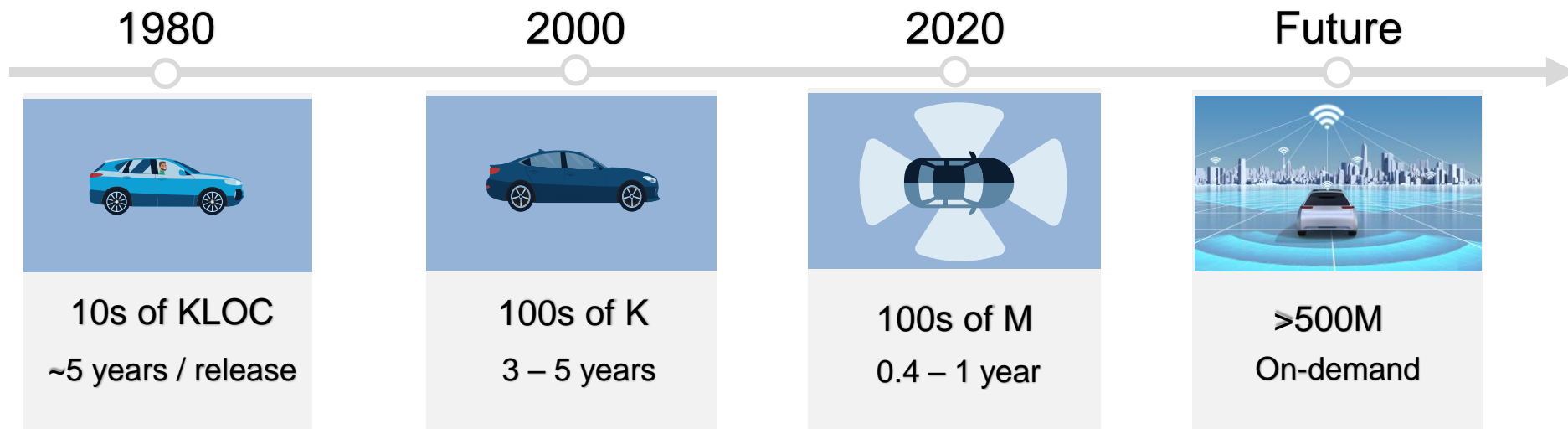
Relative growth of software complexity and productivity over time, indexed for automotive features



Source: McKinsey's SoftCoster embedded software project database

Source: <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/when-code-is-king-mastering-automotive-software-excellence>

McKinsey  
& Company



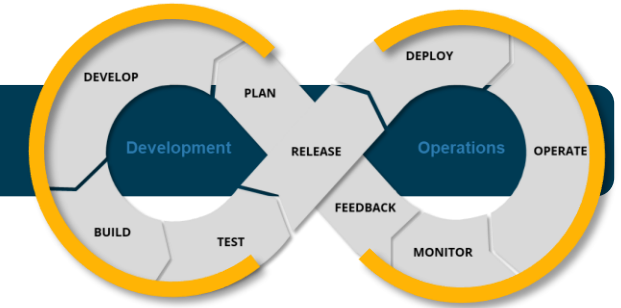
- **Accelerate development and enable early testing**
- **Agile practices, CI, and automated testing**
- **State-of-the-art development toolchain**

# Goals of DevOps and Software Factory

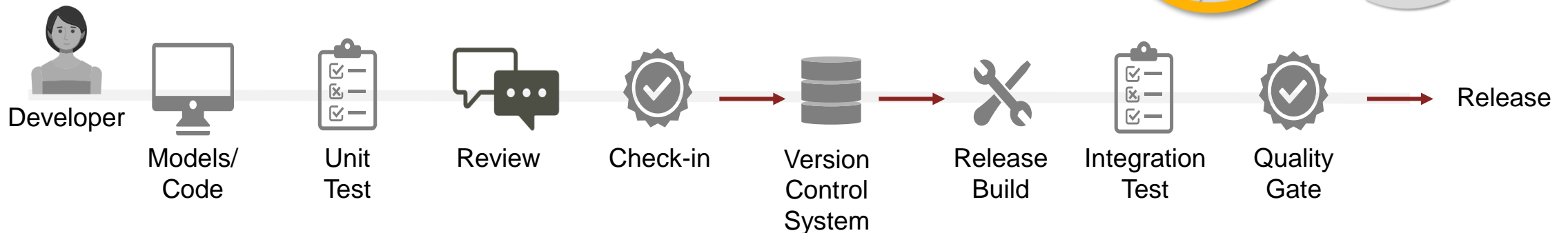
## DevOps

Unites agile development with reliable operations

Goal: **Reduce the time** between committing a change and placing it in production, **while ensuring high quality**



## Software Factory



Goal: **Repeatability, Faster delivery, Higher quality**

# Agenda

- Current challenges in the automotive industry

➔ ▪ How challenges are addressed using SW Factory and DevOps

- Establish a mature issue detection process



- Conclusion

# DevOps and Software Factory increase Quality and Productivity

## Elite Performers measured against Low Performers w.r.t. DORA Metrics

**973x  
more**

frequent code deployments

**6570x  
faster**

lead time  
from commit to deploy

**3x  
lower**

change failure rate  
(changes are 1/3 less likely to fail)

**6570x  
faster**

time to recover  
from incidents

# DevOps and Software Factory increase Quality and Productivity

## Elite Performers measured against Low Performers w.r.t. DORA Metrics

**3x**  
**lower**  
change failure rate  
(changes are 1/3 less likely to fail)

**6570x**  
**faster**  
lead time  
from commit to deploy

Typical results reported by enterprises using SAFe



<https://scaledagile.com/what-is-safe/>

**50%**  
Faster time-to-market

**50%**  
Defect reduction

**35%**  
Increase in productivity

**30%**  
Happier, more engaged employees

# Model-Based Design & Static Code Analysis enable high DevOps Performance

**Reduce the time** between committing a change and placing it in production, **while ensuring high quality and compliance**

ISO 26262

Functional Safety

ISO/SAE 21434

Cybersecurity for Road Vehicles

ISO 21448

Safety of the intended functionality (SOTIF)

**3x**  
**lower**

change failure rate  
(changes are 1/3 less likely to fail)

**6570x**  
**faster**

lead time  
from commit to deploy

**50%**

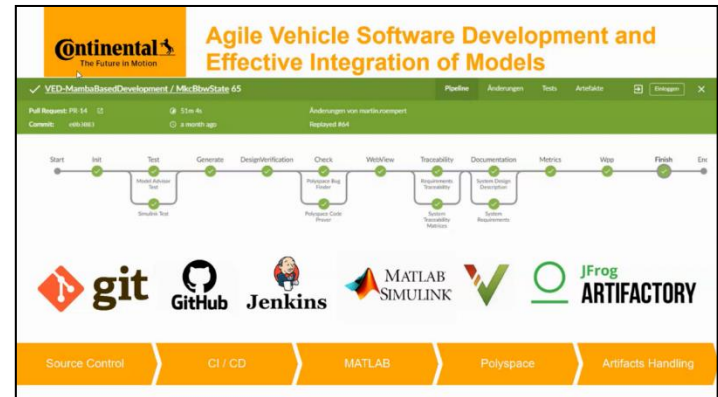
Defect reduction

**35%**

Increase in productivity

[Link to technical article](#)

Figure 1. Issue counts for software releases before and after the adoption of Model-Based Design.



Online Panel Discussion:  
[Agile Vehicle Software Development and Effective Integration of Models](#)

Verification time cut by up to 50 %

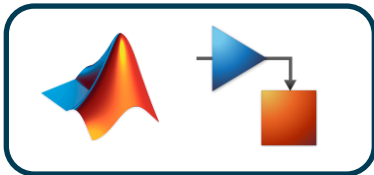
Continental Develops Electronically Controlled Air Suspension for Heavy-Duty Trucks  
[Link to user story](#)



# Agenda

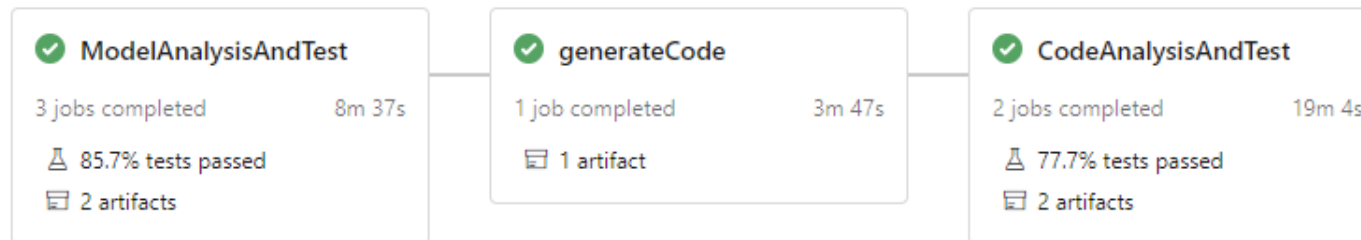
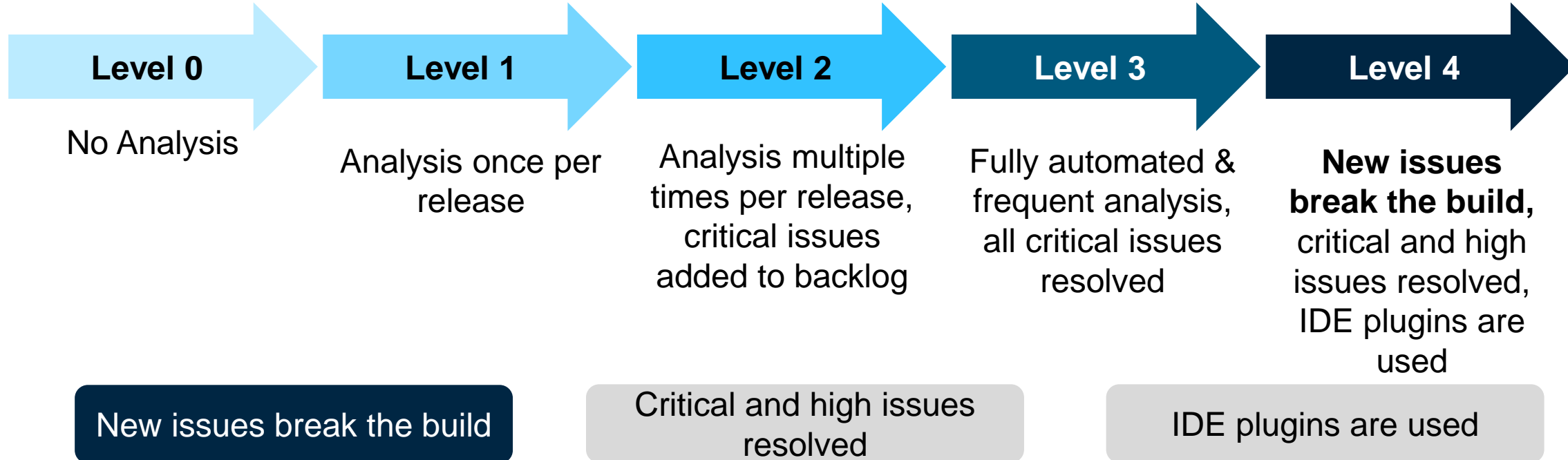
- Current challenges in the automotive industry
- How challenges are addressed using SW Factory and DevOps

- ➔ ▪ Establish a mature issue detection process

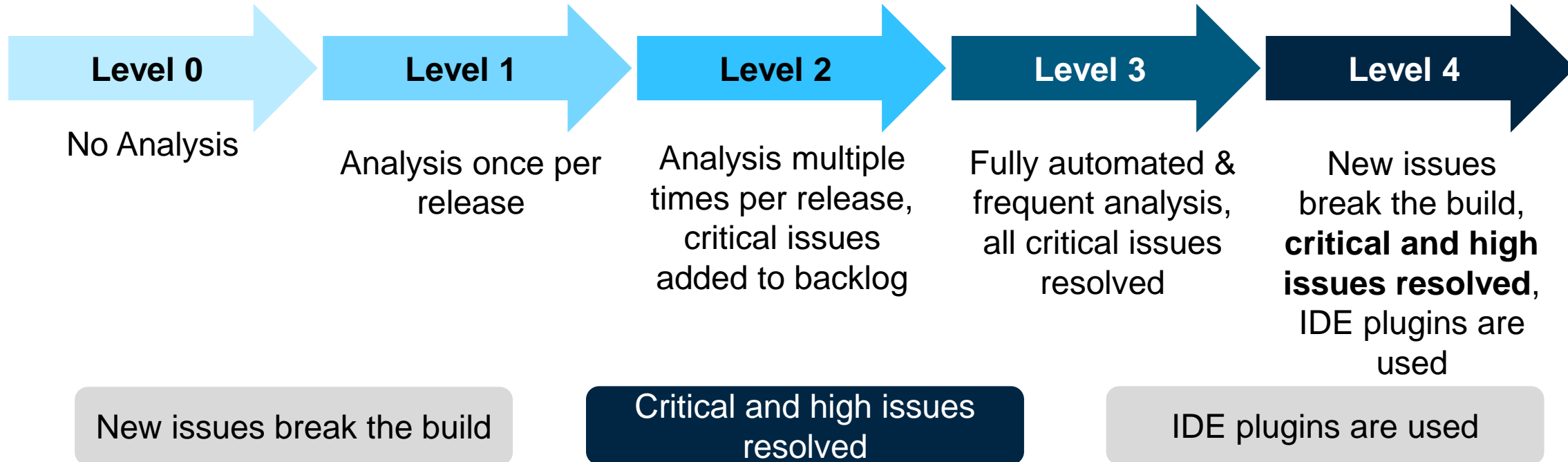


- Conclusion

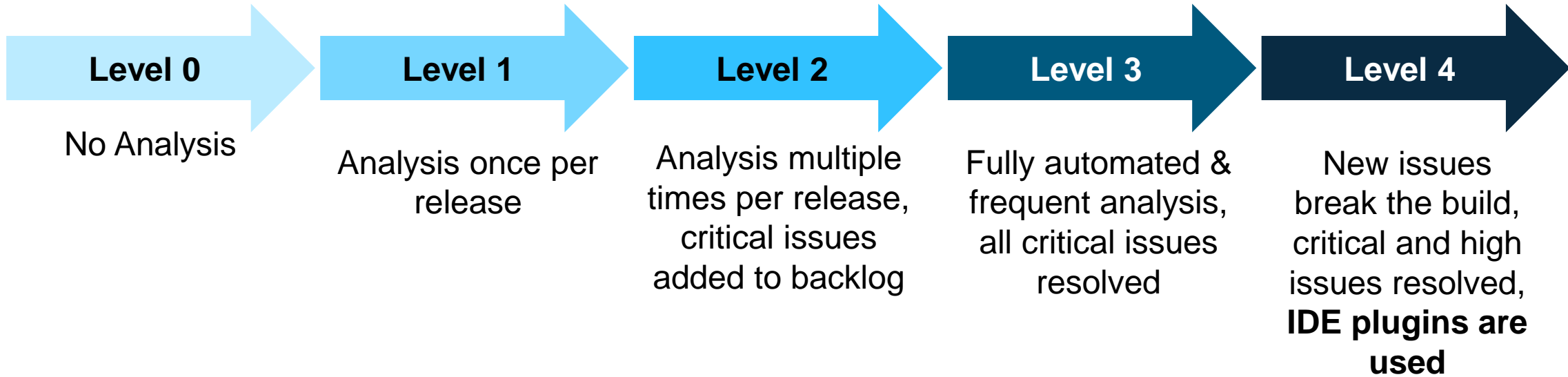
# Establish a mature Issue Detection Process



# Establish a mature Issue Detection Process



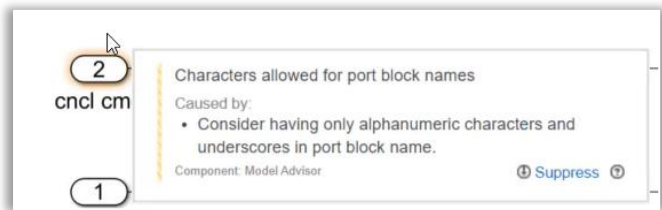
# Establish a mature Issue Detection Process



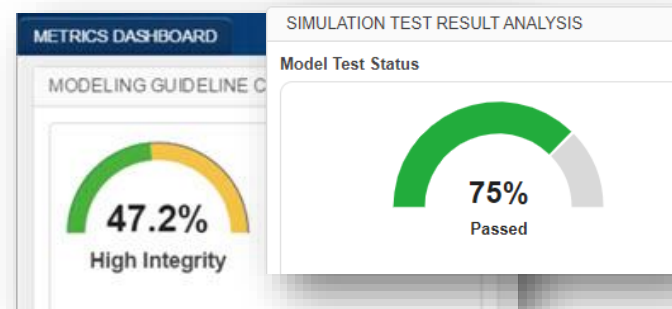
New issues break the build

Critical and high issues resolved

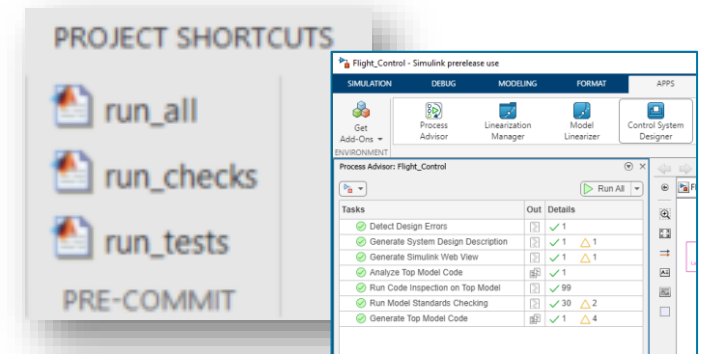
IDE plugins are used



Edit time checks

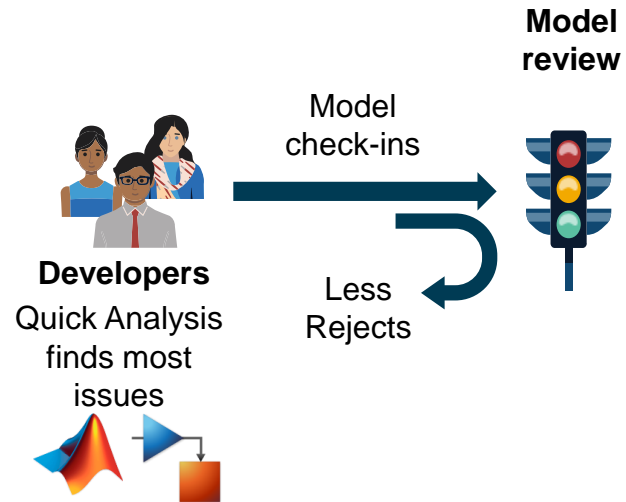


Quality Dashboards

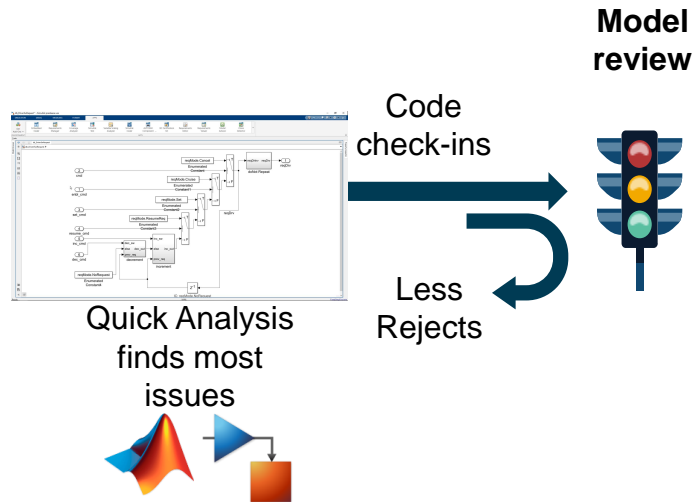


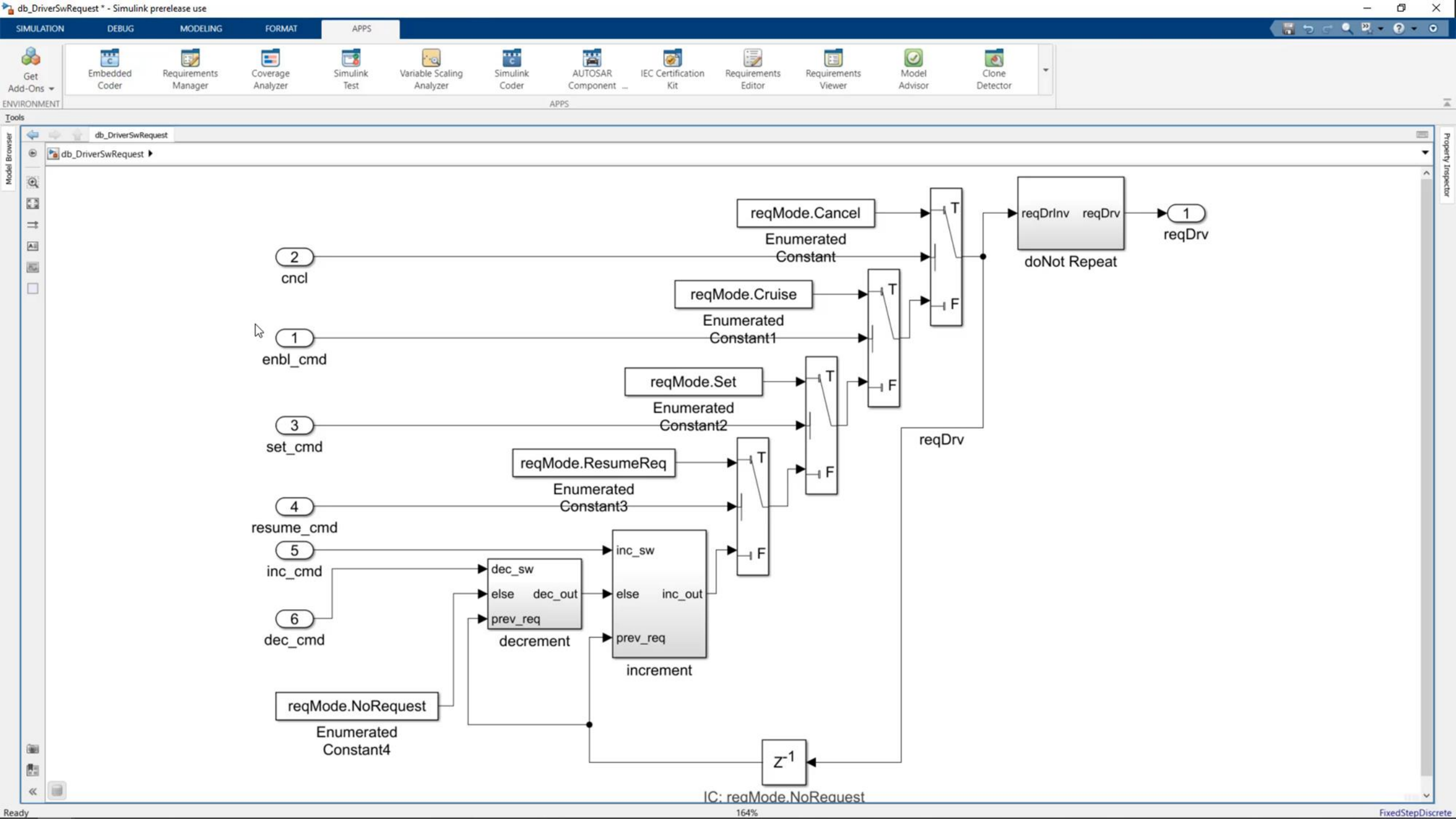
Exhaustive pre-commit gating

# Pre-Submit Workflow with Model-Based Design

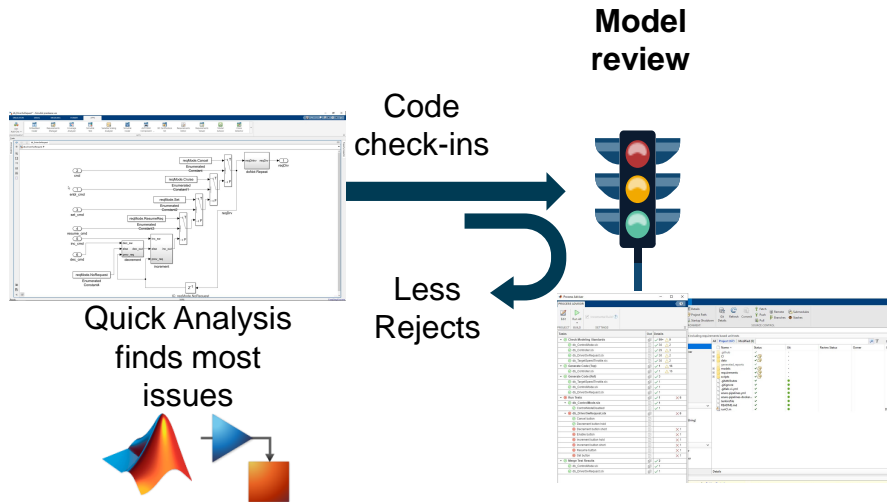


# Pre-Submit Workflow with Model-Based Design





# Pre-Submit Workflow with Model-Based Design



Direct Feedback



Learning



PROVIDED BY  
SCALED AGILE



IDE Plugins are used



PROCESS ADVISOR

Incremental Build ?

PROJECT BUILD SETTINGS

Tasks	Out	Details
<ul style="list-style-type: none"> <li>Check Modeling Standards                             <ul style="list-style-type: none"> <li>db_ControlMode.slx (30/2)</li> <li>db_Controller.slx (29/3)</li> <li>db_DriverSwRequest.slx (30/2)</li> <li>db_TargetSpeedThrottle.slx (30/2)</li> </ul> </li> <li>Generate Code (Top) (1/16)                             <ul style="list-style-type: none"> <li>db_Controller.slx (1/16)</li> </ul> </li> <li>Generate Code (Ref) (3)                             <ul style="list-style-type: none"> <li>db_TargetSpeedThrottle.slx (1)</li> <li>db_ControlMode.slx (1)</li> <li>db_DriverSwRequest.slx (1)</li> </ul> </li> <li>Run Tests (1/6)                             <ul style="list-style-type: none"> <li>db_ControlMode.slx (1)                                     <ul style="list-style-type: none"> <li>ControlModeDisabled (1)</li> </ul> </li> <li>db_DriverSwRequest.slx (6)                                     <ul style="list-style-type: none"> <li>Cancel button</li> <li>Decrement button hold</li> <li>Decrement button short (1)</li> <li>Enable button (1)</li> <li>Increment button hold (1)</li> <li>Increment button short (1)</li> <li>Resume button (1)</li> <li>Set button (1)</li> </ul> </li> </ul> </li> <li>Merge Test Results (2)                             <ul style="list-style-type: none"> <li>db_ControlMode.slx (1)</li> <li>db_DriverSwRequest.slx (1)</li> </ul> </li> </ul>		

SOURCE CONTROL

including requirements based unit tests

All | Project (67) | Modified (0)

Name	Status	Git	Review Status	Owner	Class
.github	✓	.			
CI	✓	.			
data	✓	.			
generated_reports	✓	.			
models	✓	.			
requirements	✓	.			
scripts	✓	.			
.gitattributes	✓	●			
.gitignore	✓	●			
.gitlab-ci.yml	✓	●			
azure-pipelines.yml	✓	●			
azure-pipelines-docker...	✓	●			
Jenkinsfile	✓	●			
README.md	✓	●			
runCl.m	✓	●			

Details

PROJECT SHORTCUTS

Search
 Custom Tasks
 Run Checks
 References
 Details
 Project Path
 Startup Shutdown
 Git Details
 Refresh
 Commit
 Fetch
 Push
 Pull
 Remote
 Branches
 Submodules
 Stashes

ENVIRONMENT SOURCE CONTROL

Project - Demo Project including requirements based unit tests Editor - ci\_runCodegen.m

Views

Files

Dependency Analyzer

Name	Status	Classification	Git	Review Status	Owner
models	✓		■		
db_DriverSwRequest	✓		■		
db_DriverSwRequest.slx	✓	Design	■		Tjor
db_DriverSwRequest_harnessInf...	✓	Derived	■		
resources	.		■		

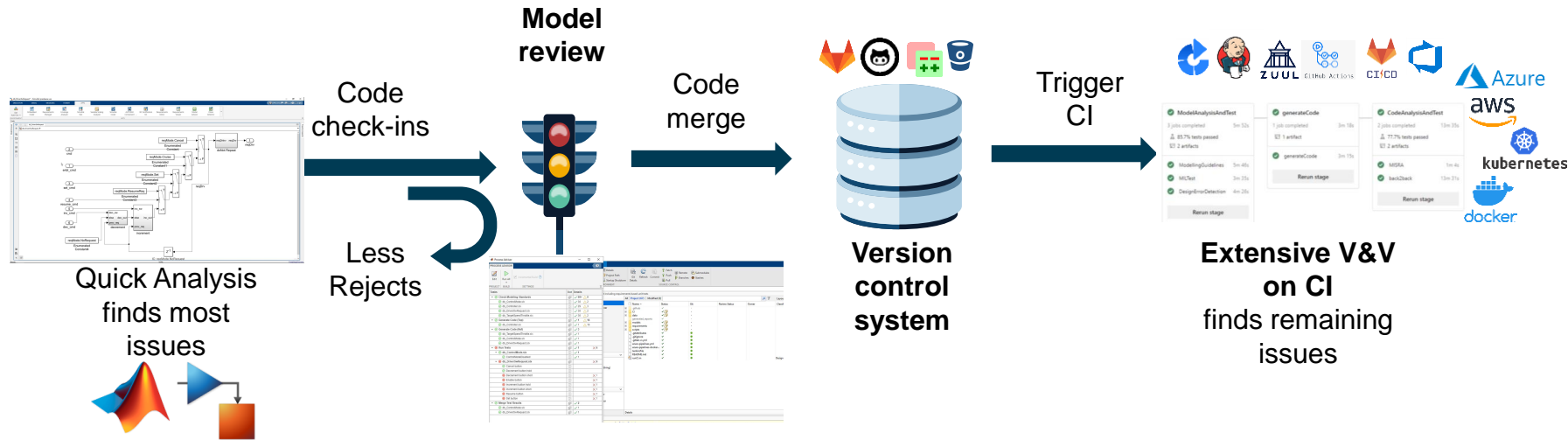
Labels ^

Git v

Current branch: myNewFeature  
Branch status: Normal  
No remote tracking branch

Details

# Post-Submit Workflow with Model-Based Design



Direct Feedback



Learning



Flow / Automation



IDE Plugins are used



New issues break the build

# #20220317.2 my new feature

MBD\_CC - Dockerised on Linux

This run is being retained as one of 3 recent runs by pipeline.

View

Summary Tests Associated pipelines

Triggered by  Tjorben Gross

Repository and version

MBD\_CC

myNewFeature c2720fcd

Time started and elapsed

Today at 15:25

22m 59s

Related

0 work items

4 published

Tests and coverage

82.6% passed

[Setup code coverage](#)

Stages Jobs

## ModelAnalysisAndTest

3 jobs completed 5m 52s

85.7% tests passed

2 artifacts

ModellingGuidelines 5m 46s

MILTest 3m 35s

DesignErrorDetection 4m 26s

Rerun stage

## generateCode

1 job completed 3m 18s

1 artifact

generateCcode 3m 15s

Rerun stage

## CodeAnalysisAndTest

2 jobs completed 13m 35s

77.7% tests passed

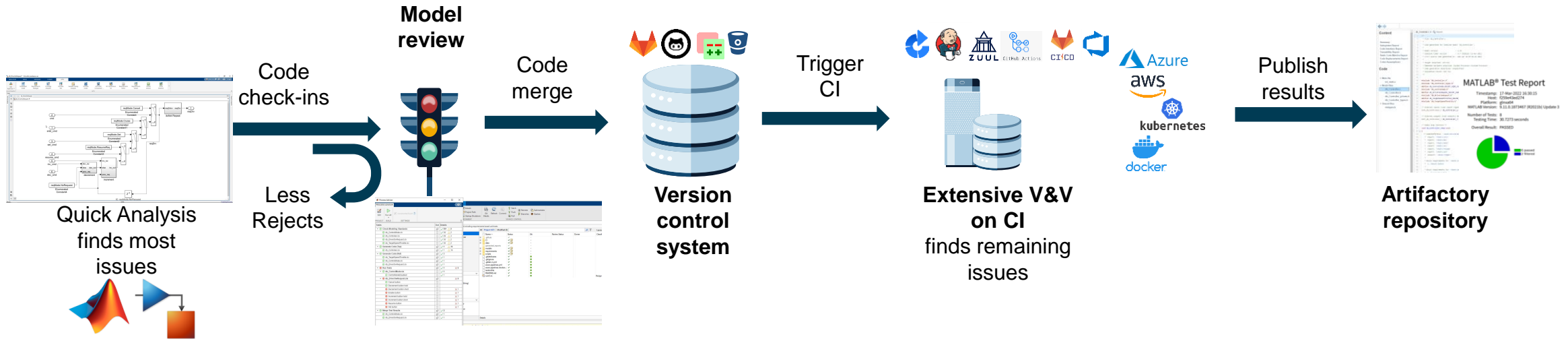
2 artifacts

MISRA 1m 4s

back2back 13m 31s

Rerun stage

# Post-Submit Workflow with Model-Based Design



✓ Direct Feedback

✓ Learning

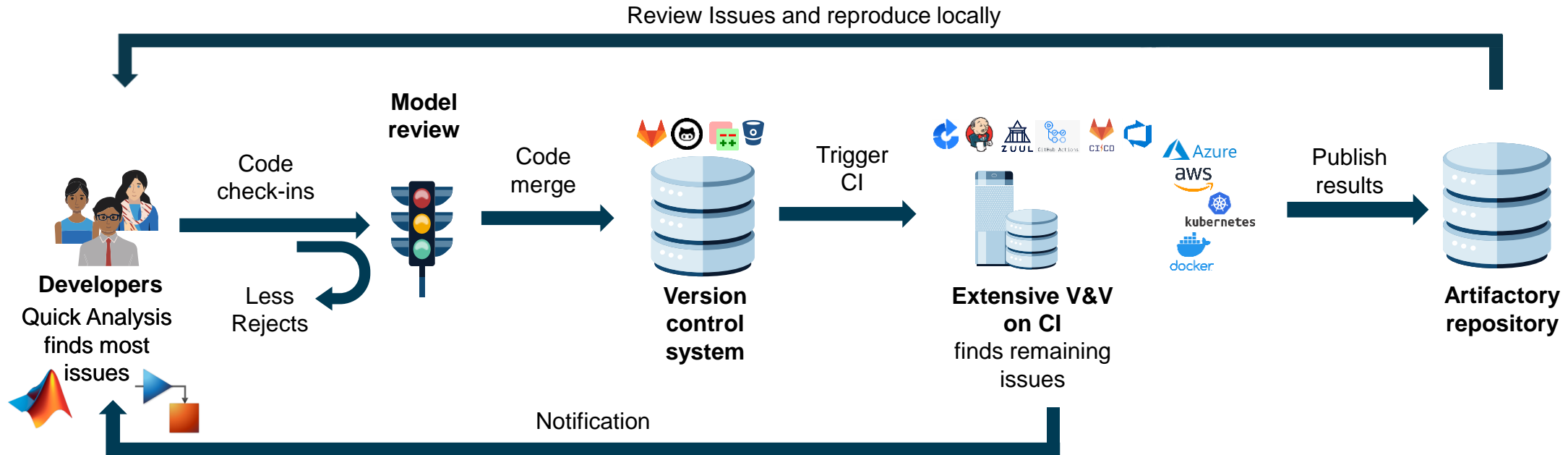
✓ Flow / Automation



✓ IDE Plugins are used

✓ New issues break the build

# Post-Submit Workflow with Model-Based Design



✓ Direct Feedback

✓ Learning

✓ Flow / Automation

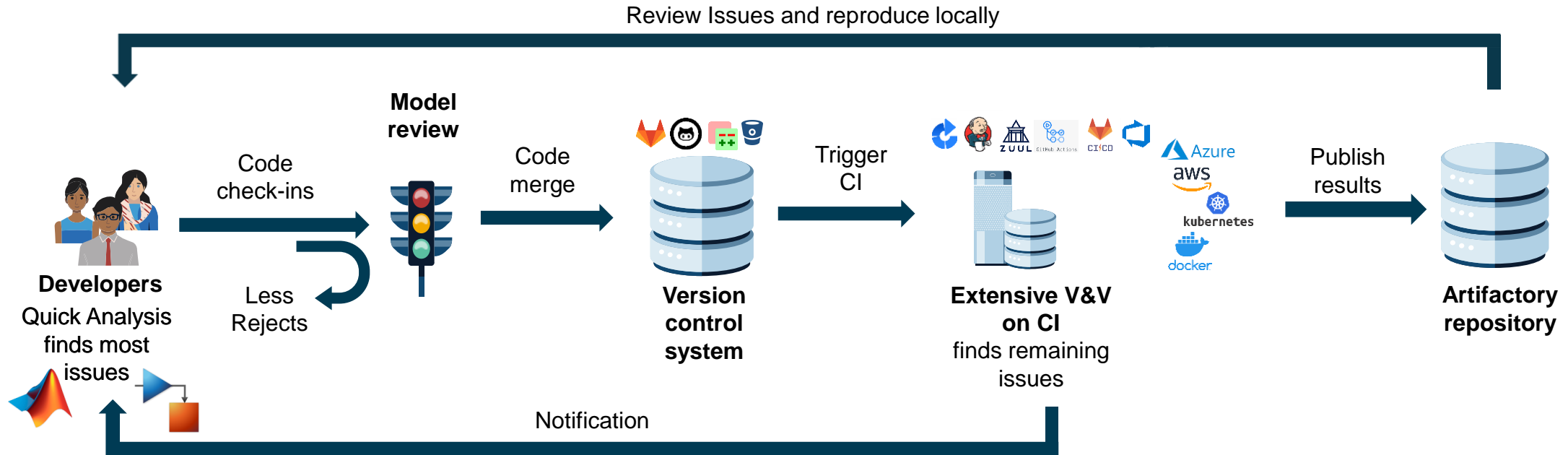


✓ IDE Plugins are used

✓ New issues break the build

✓ Critical and high issues resolved

# Workflow with Model-Based Design



✓ Direct Feedback

✓ Learning

✓ Flow / Automation



✓ IDE Plugins are used

✓ New issues break the build

✓ Critical and high issues resolved

# Agenda

- Current challenges in the automotive industry
- How challenges are addressed using SW Factory and DevOps

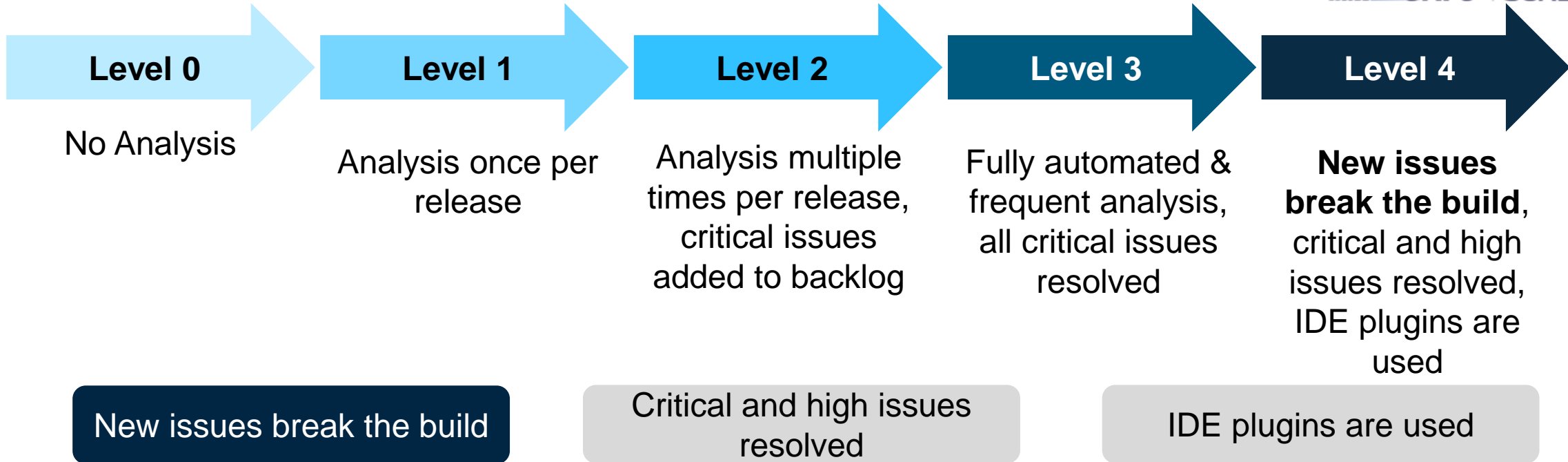
- ➔ ▪ Establish a mature issue detection process



- Conclusion



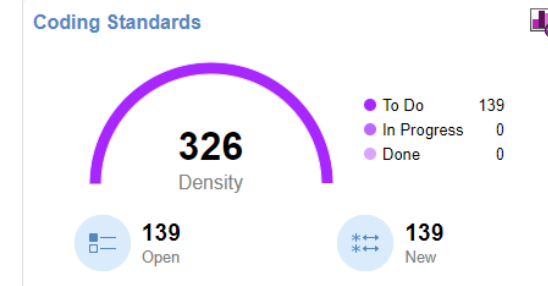
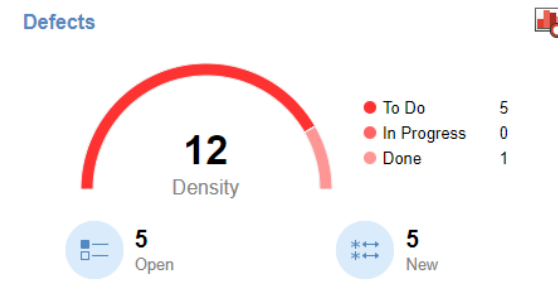
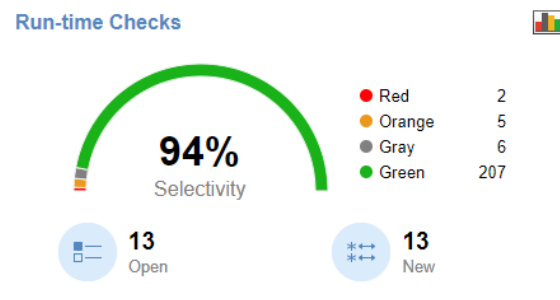
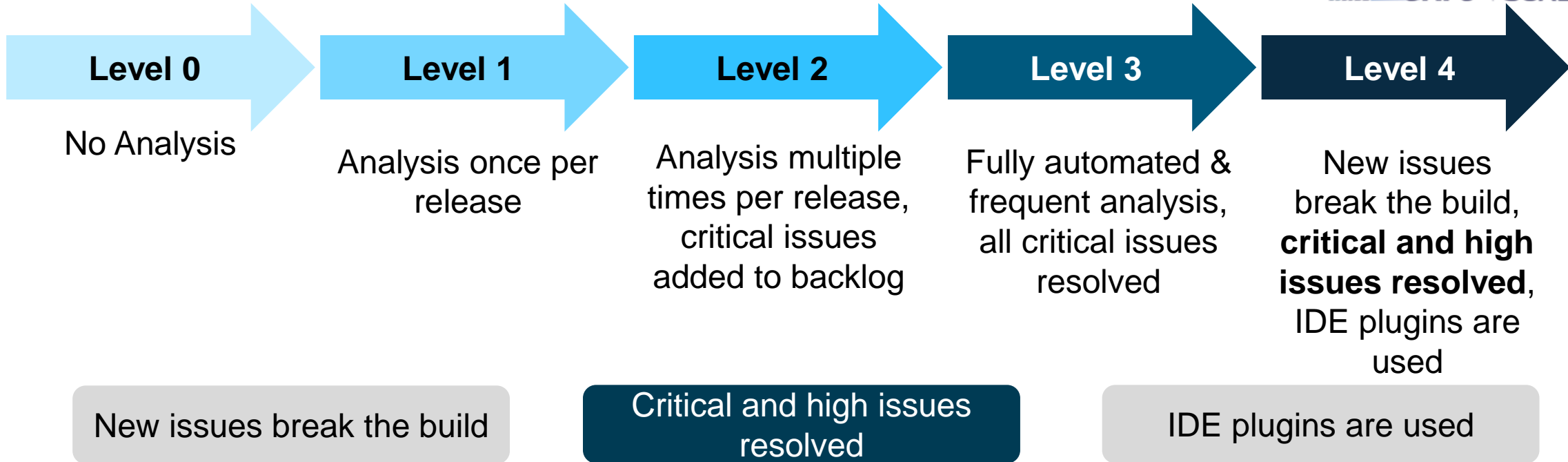
# Establish a Mature Issue Detection Process with Polyspace



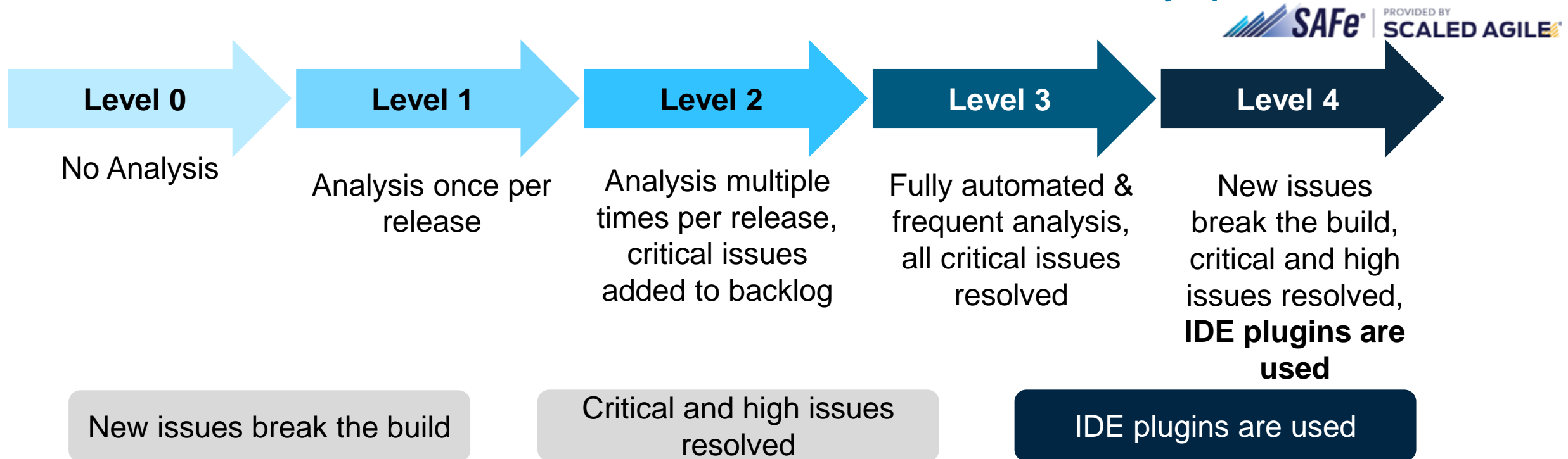
S	W	Name	Last Duration
⊗	🔗	DeclPipeline_Example	1 min 3 sec
⊗	🔗	PSBF_Example_C_Jenkins_Agent	1 min 56 sec
✅	🔗	PSBF_Example_C_offload2MDCS	25 min



# Establish a Mature Issue Detection Process with Polyspace



# Establish a Mature Issue Detection Process with Polyspace

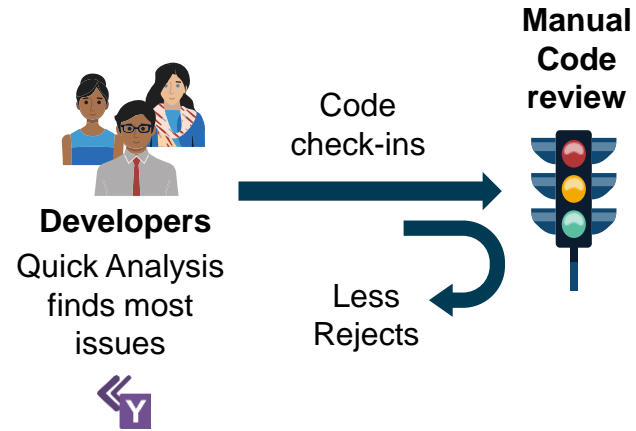


```

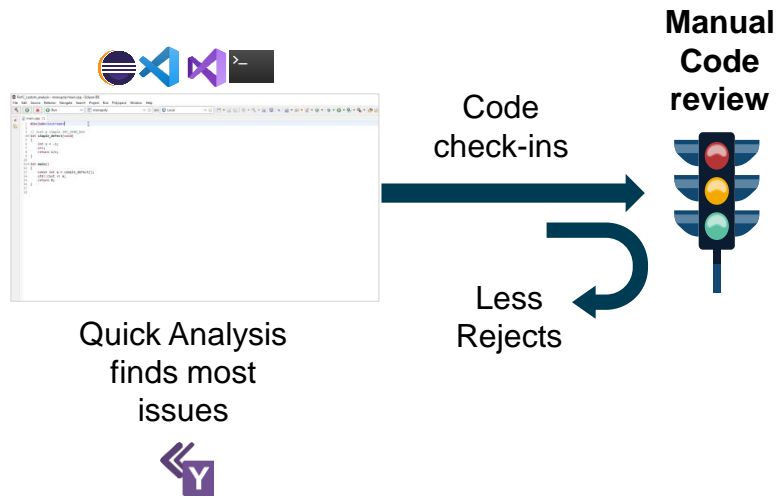
File Edit Selection View Go Run
POLYSPACE
QUALITY MONITORING
board.cpp
✘ 18-0-3 The library functions abort, exit, getenv and system from library <cstdlib> shall not be used. polyspace(MISRA C++:2008:18-0-3) [161, 14]
✘ 3-9-2 Typedefs that indicate size and signedness should be used in place of the basic numerical types. polyspace(MISRA C++:2008:3-9-2) [167, 24]
✘ 7-1-1 A variable which is not modified shall be const qualified. polyspace(MISRA C++:2008:7-1-1) [167, 44]
✘ 5-0-2 Limited dependence should be placed on C++ operator precedence rules in expressions. polyspace(MISRA C++:2008:5-0-2) [169, 15]
✘ 5-2-1 Each operand of a logical && or || shall be a postfix-expression. polyspace(MISRA C++:2008:5-2-1) [169, 19]
✘ 5-0-2 Limited dependence should be placed on C++ operator precedence rules in expressions. polyspace(MISRA C++:2008:5-0-2) [169, 28]

```

# Pre-Submit Workflow and Quality Gate with Polyspace



# Pre-Submit Workflow and Quality Gate with Polyspace

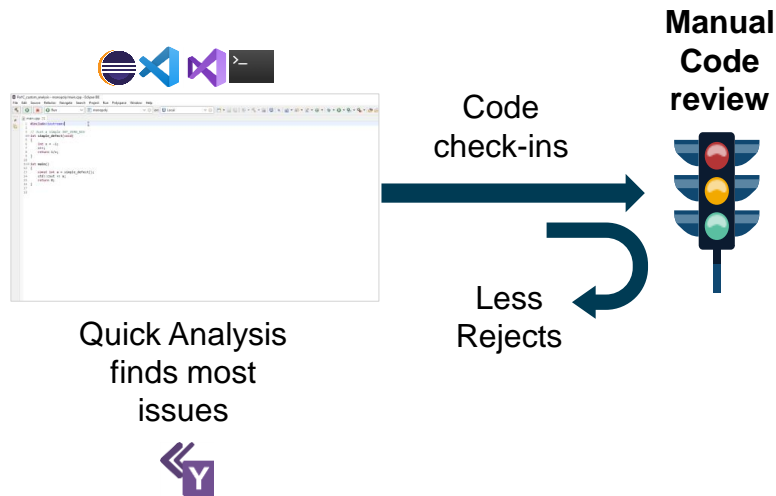




main.cpp

```
1 #include<iostream>
2
3 // Just a simple INT_ZERO_DIV
4 int simple_defect(void)
5 {
6     int x = -1;
7     x++;
8     return 1/x;
9 }
10
11 int main()
12 {
13     const int a = simple_defect();
14     std::cout << a;
15     return 0;
16 }
17
18
```

# Pre-Submit Workflow and Quality Gate with Polyspace



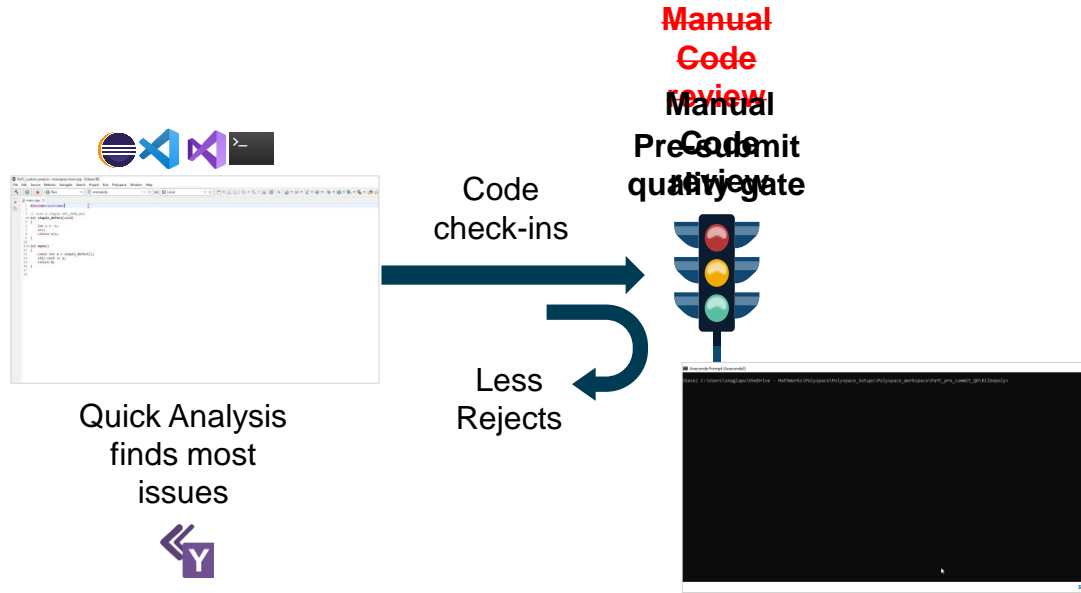
✓ Direct Feedback

✓ Learning



✓ IDE Plugins are used

# Pre-Submit Workflow and Quality Gate with Polyspace



✓ Direct Feedback

✓ Learning

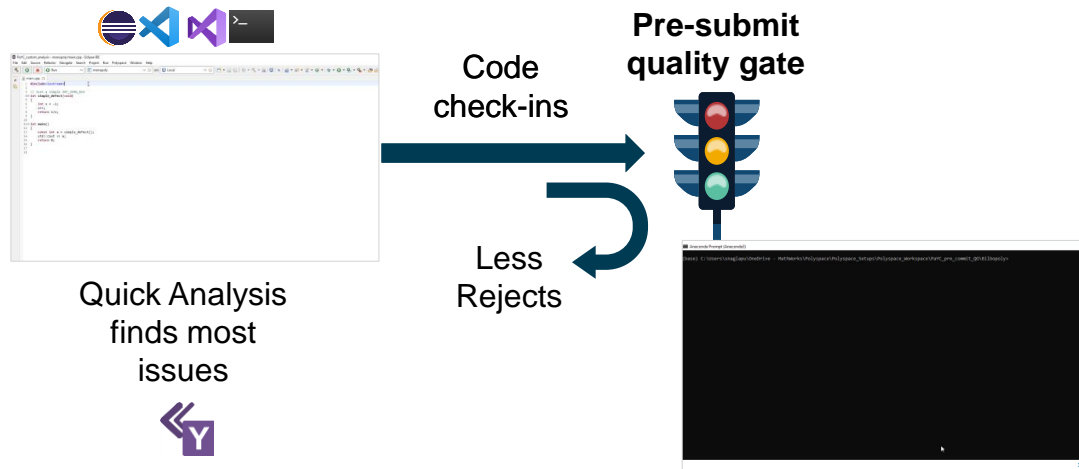


✓ IDE Plugins are used



```
(base) C:\Users\snaglapu\OneDrive - MathWorks\Polyspace\Polyspace_Setups\Polyspace_Workspace\PaYC_pre_commit_Q0\Bilbopoly>
```

# Pre-Submit Workflow and Quality Gate with Polyspace



✓ Direct Feedback

✓ Learning

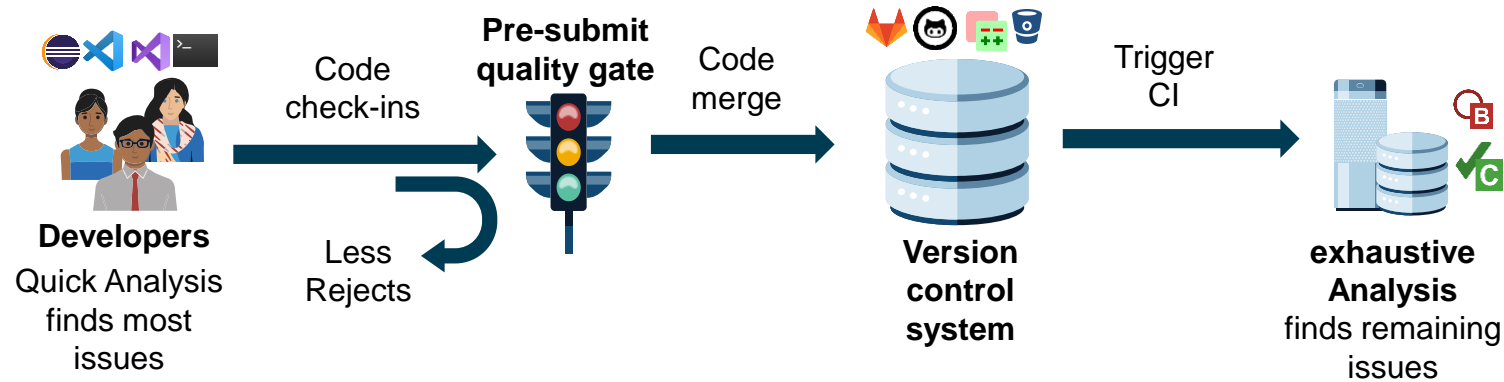
✓ Flow / Automation



✓ IDE Plugins are used

✓ New issues break the build

# Post-Submit Verification Workflow and Quality Gate with Polyspace



✓ Direct Feedback

✓ Learning

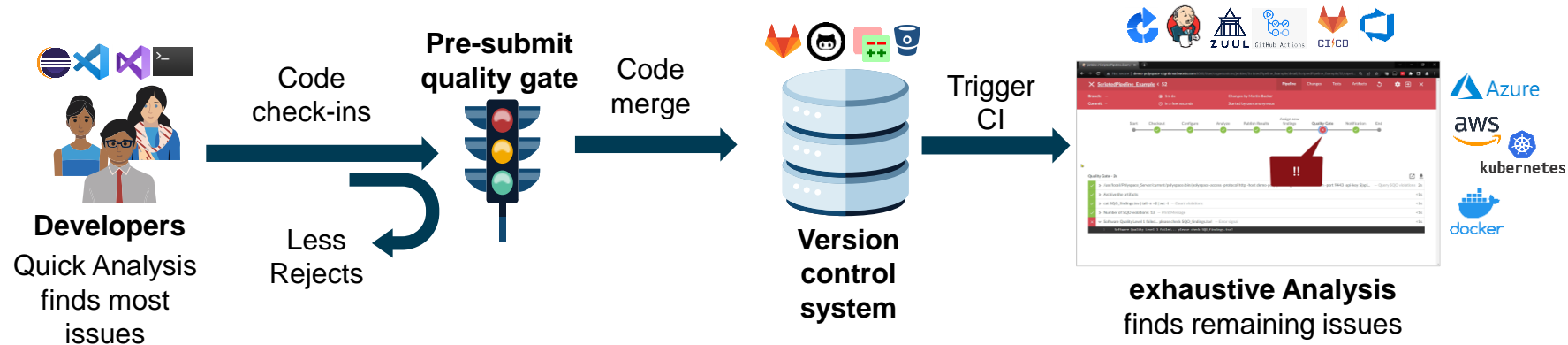
✓ Flow / Automation



✓ IDE Plugins are used

✓ New issues break the build

# Post-Submit Verification Workflow and Quality Gate with Polyspace



- ✓ Direct Feedback
- ✓ Learning
- ✓ Flow / Automation



- ✓ IDE Plugins are used
- ✓ New issues break the build



**Post-submit quality gate**

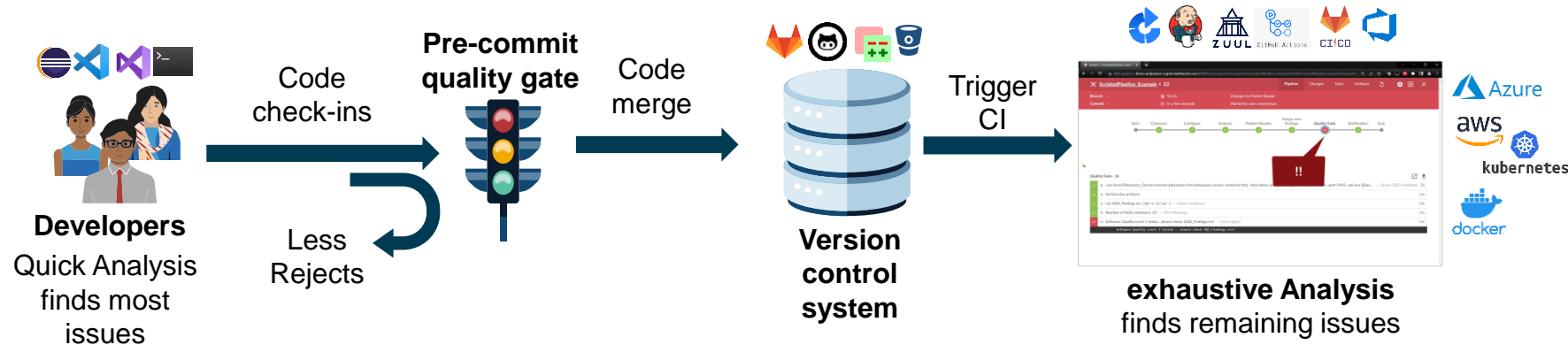
Branch: — 1m 6s Changes by Martin Becker  
Commit: — in a few seconds Started by user anonymous



Quality Gate - 2s

- ✓ > /usr/local/Polyspace\_Server/current/polyspace/bin/polyspace-access -protocol http -host demo-polyspace-ci.gnb.mathworks.com -port 9443 -api-key \${api... — Query SQO violations 2s
  - ✓ > Archive the artifacts <1s
  - ✓ > cat SQO\_findings.tsv | tail -n +2 | wc -l — Count violations <1s
  - ✓ > Number of SQO violations: 13 — Print Message <1s
  - ✗ ✓ Software Quality Level 1 failed... please check SQO\_findings.tsv! — Error signal <1s
- ```
1 Software Quality Level 1 failed... please check SQO_findings.tsv!
```

# Post-Submit Verification Workflow and Quality Gate with Polyspace



✓ Direct Feedback

✓ Learning

✓ Flow / Automation



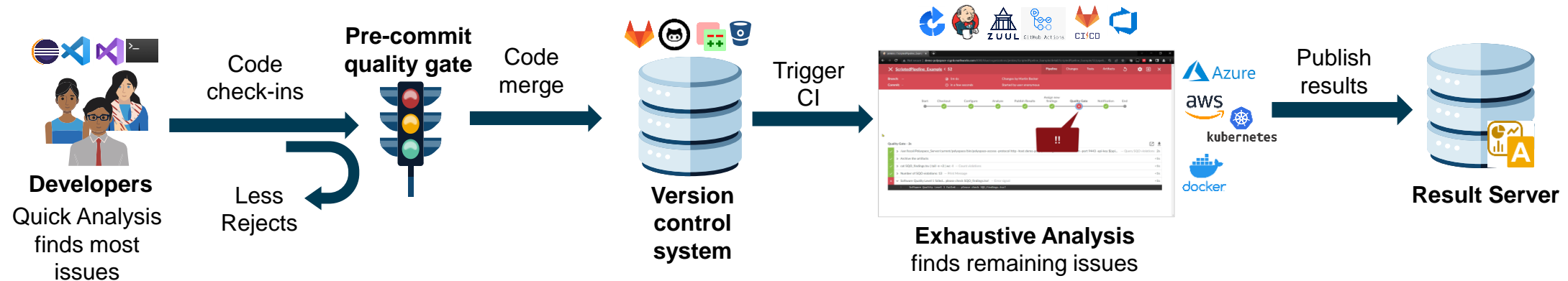
✓ IDE Plugins are used

✓ New issues break the build



Post-submit quality gate

# Result Review Workflow with Polyspace



✓ Direct Feedback

✓ Learning

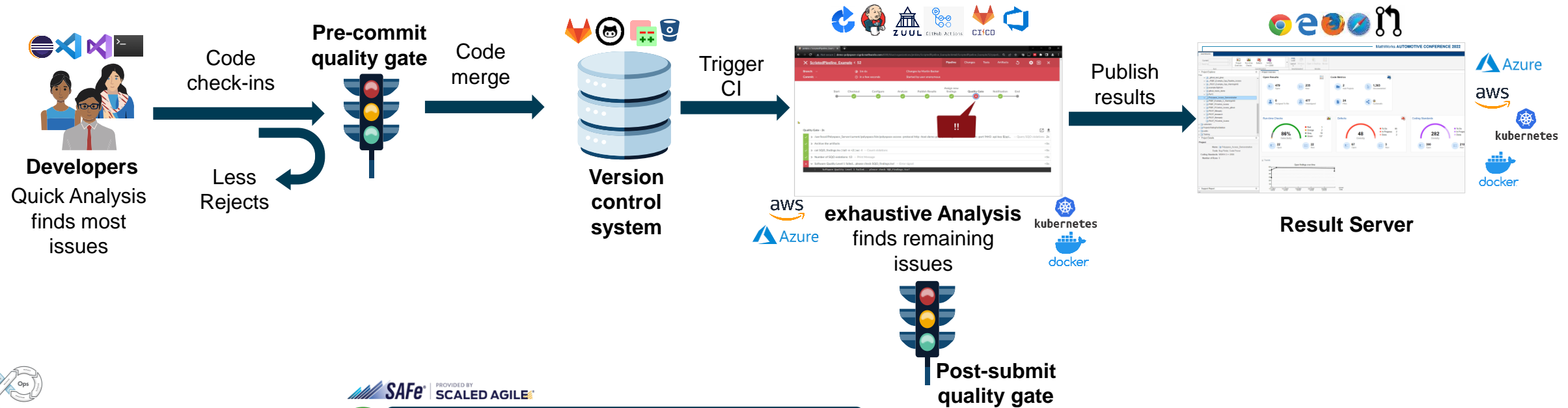
✓ Flow / Automation



✓ IDE Plugins are used

✓ New issues break the build

# Result Review Workflow with Polyspace



✓ Direct Feedback

✓ Learning

✓ Flow / Automation



✓ IDE Plugins are used

✓ New issues break the build



DASHBOARD

Current [dropdown]  
Baseline [dropdown]

Project Overview | Run-time Checks | Defects | MISRA C++:2008

Layout | Window | Open in Desktop | Review

ENVIRONMENT | REVIEW

- Project Explorer
- Filter
- ▶ \_github\_test\_ghes
  - ▶ \_PSBF\_Example\_Cpp\_Pipeline\_Access
  - ▶ \_PSCP\_Example\_Cpp\_WarningsNG
  - ▶ example-flightsim
  - ▶ github\_hooks\_demo
  - ▶ PaYC
  - ▶ Polyspace\_Access\_Demonstration
  - ▶ PSBF\_Example\_C\_WarningsNG
  - ▶ PSBF\_PIControl\_Access
  - ▶ PSBF\_PIControl\_Access\_github
  - ▶ PSCP\_Bilbopoly
  - ▶ PSCP\_binsearch
  - ▶ PSCP\_Monopoly
  - ▶ PSCP\_PIControl\_Access
  - ▶ customers
  - ▶ ProjectsWaitingForDeletion
  - ▶ public
  - ▶ Training

Project Details

Project

Name Polyspace\_Access\_Demonstration

Tools Bug Finder, Code Prover

Coding Standards MISRA C++:2008

Number of Runs 6

### Open Results

479 Open

235 New

0 Assigned To Me

477 Unassigned

### Code Metrics

2 Sub Projects

1,383 Uncommented

24 Files

Cyclomatic

### Run-time Checks

86% Selectivity

|        |     |
|--------|-----|
| Red    | 1   |
| Orange | 2   |
| Gray   | 19  |
| Green  | 137 |

22 Open

22 New

### Defects

48 Density

|             |    |
|-------------|----|
| To Do       | 65 |
| In Progress | 2  |
| Done        | 2  |

67 Open

3 New

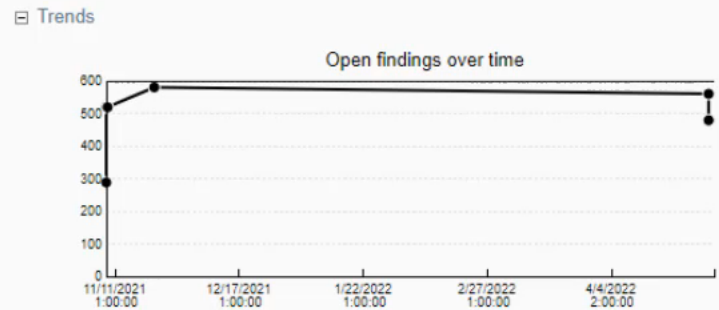
### Coding Standards

282 Density

|             |  |
|-------------|--|
| To Do       |  |
| In Progress |  |
| Done        |  |

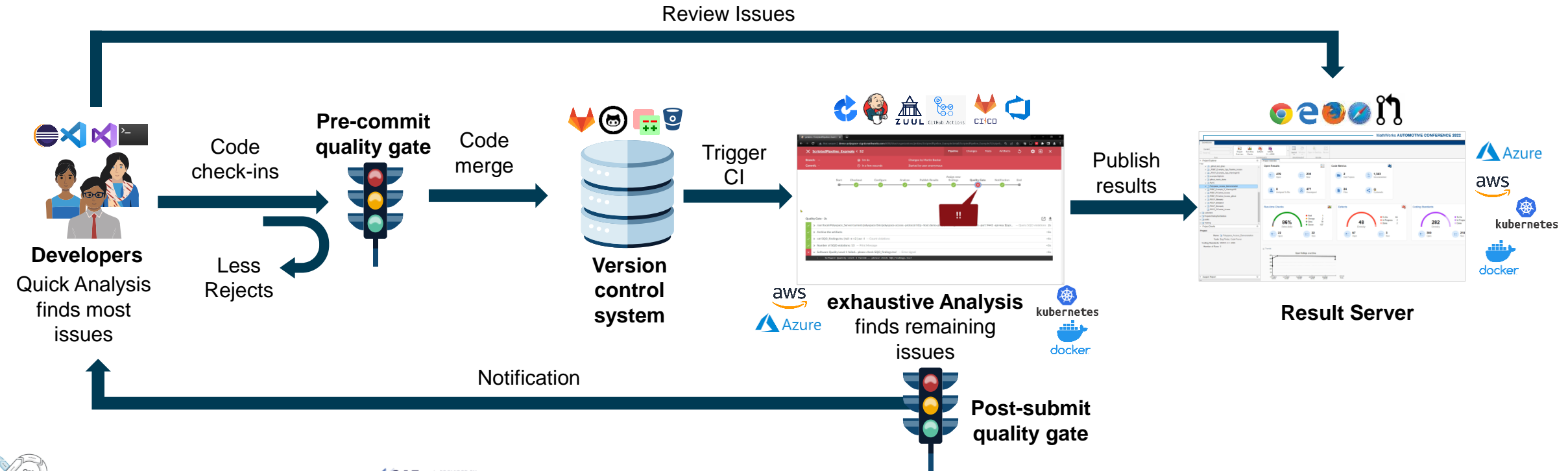
390 Open

210 New



Support Report

# Result Review Workflow with direct Feedback to Developers



✓ Direct Feedback

✓ Learning

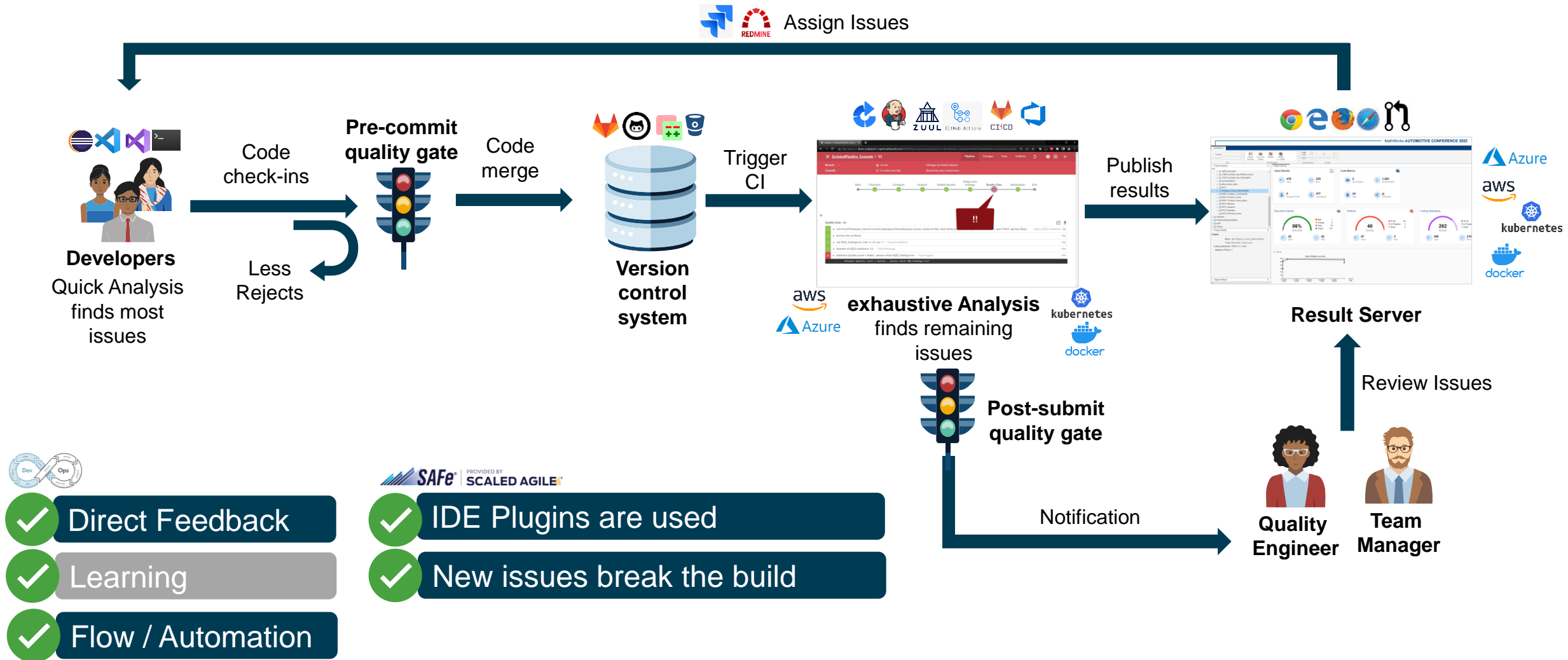
✓ Flow / Automation



✓ IDE Plugins are used

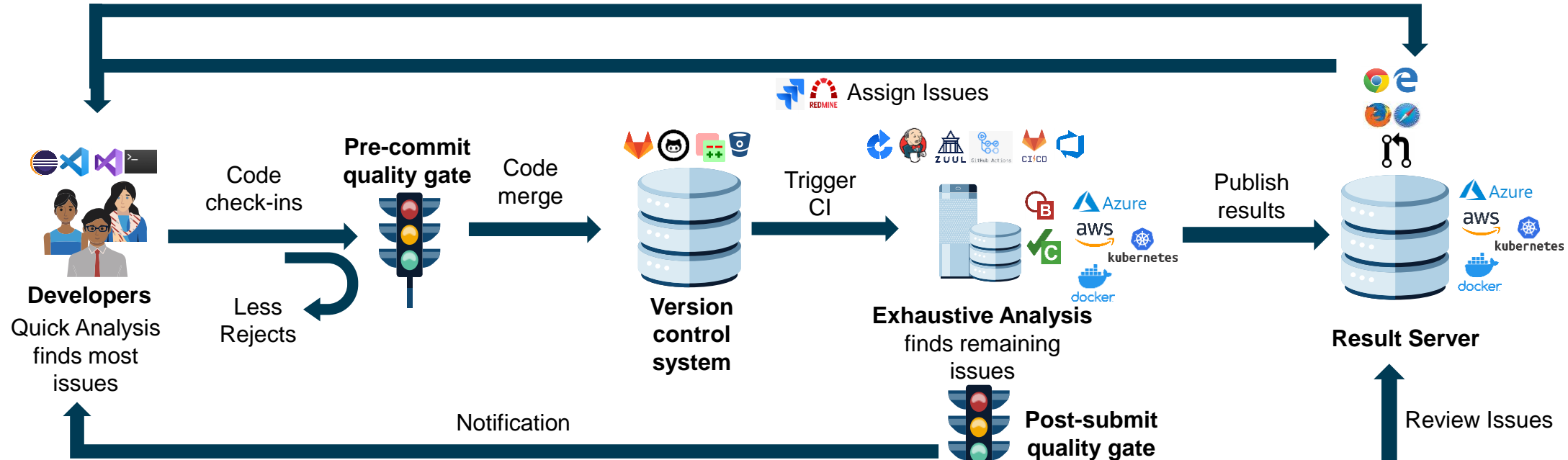
✓ New issues break the build

# Result Review Workflow with direct Feedback to QA



# Verification and Result Review Workflow with Polyspace

Review Issues



✓ Direct Feedback

✓ Learning

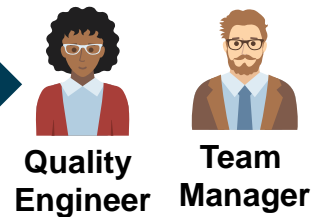
✓ Flow / Automation



✓ IDE Plugins are used

✓ New issues break the build

✓ Critical and high issues resolved



# Agenda

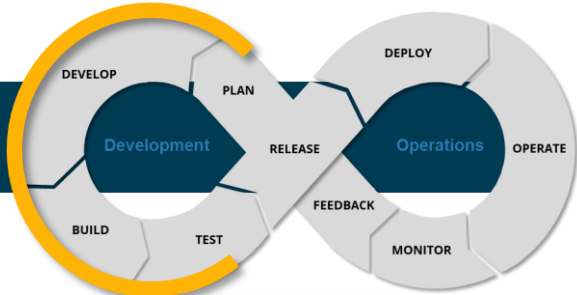
- Current challenges in the automotive industry
- How challenges are addressed using SW Factory and DevOps
- Establish a mature issue detection process



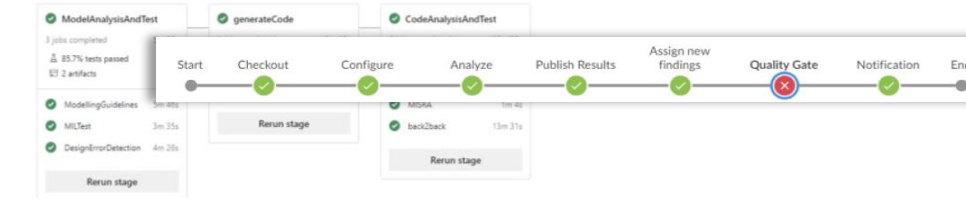
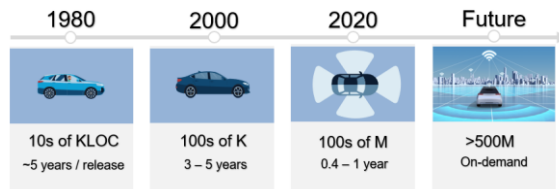
- ➔ ▪ Conclusion

# Key Takeaways

## Today's focus



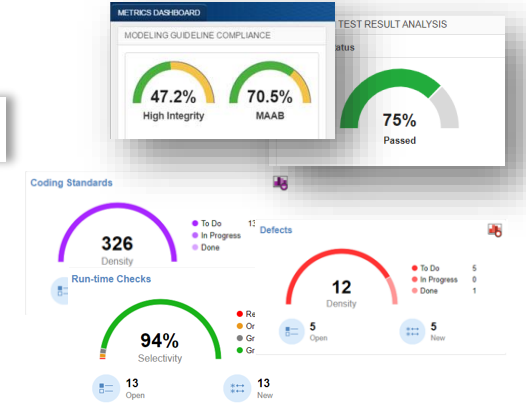
- Improve the software quality iteratively
- Improve the production speed of software deliveries



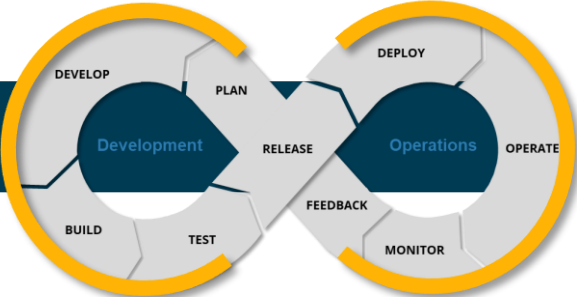
18-0-3 The library functions abort, exit, getenv and system from library <stdlib> s  
 3-9-2 Typedefs that indicate size and signedness should be used in place of the ba  
 7-1-1 A variable which is not modified shall be const qualified. polyspace(MISRA C  
 5-0-2 Limited dependence should be placed on C++ operator precedence rules in  
 5-2-1 Each operand of a logical && or || shall be a postfix-expression. polyspace(M  
 5-0-2 Limited dependence should be placed on C++ operator precedence rules in expressions. polyspace(MISRA C++-2008-5-0-2) [169, 28]

2 Characters allowed for port block names  
 Caused by:  
 • Consider having only alphanumeric characters and underscores in port block name.  
 Component: Model Advisor

1



## There is more



[Link to video](#)  
**John Deere Improving Predictive Maintenance Performance using Transient Data and Cloud-Based Algorithms**



[snaglapu@mathworks.com](mailto:snaglapu@mathworks.com)  
[tgross@mathworks.com](mailto:tgross@mathworks.com)