Rapid software testing and conformance with static code analysis

October 2016

Walter Capitani

Product Management, Rogue Wave Software

What we do

Rogue Wave helps organizations **simplify** complex software development, **improve** code quality, and **shorten** cycle times

RogueWave SOFTWARE
Accelerating Great Code

GENIVI®

Company snapshot

We are the largest independent provider of cross-platform software development tools and embedded components

Founded:	Headquarters:	Employees:	Offices Worldwide:
1989	Louisville, CO	350	11

Our capabilities cover different languages, code bases, and platforms. We meet development where – and how – it happens.

GENIVI®

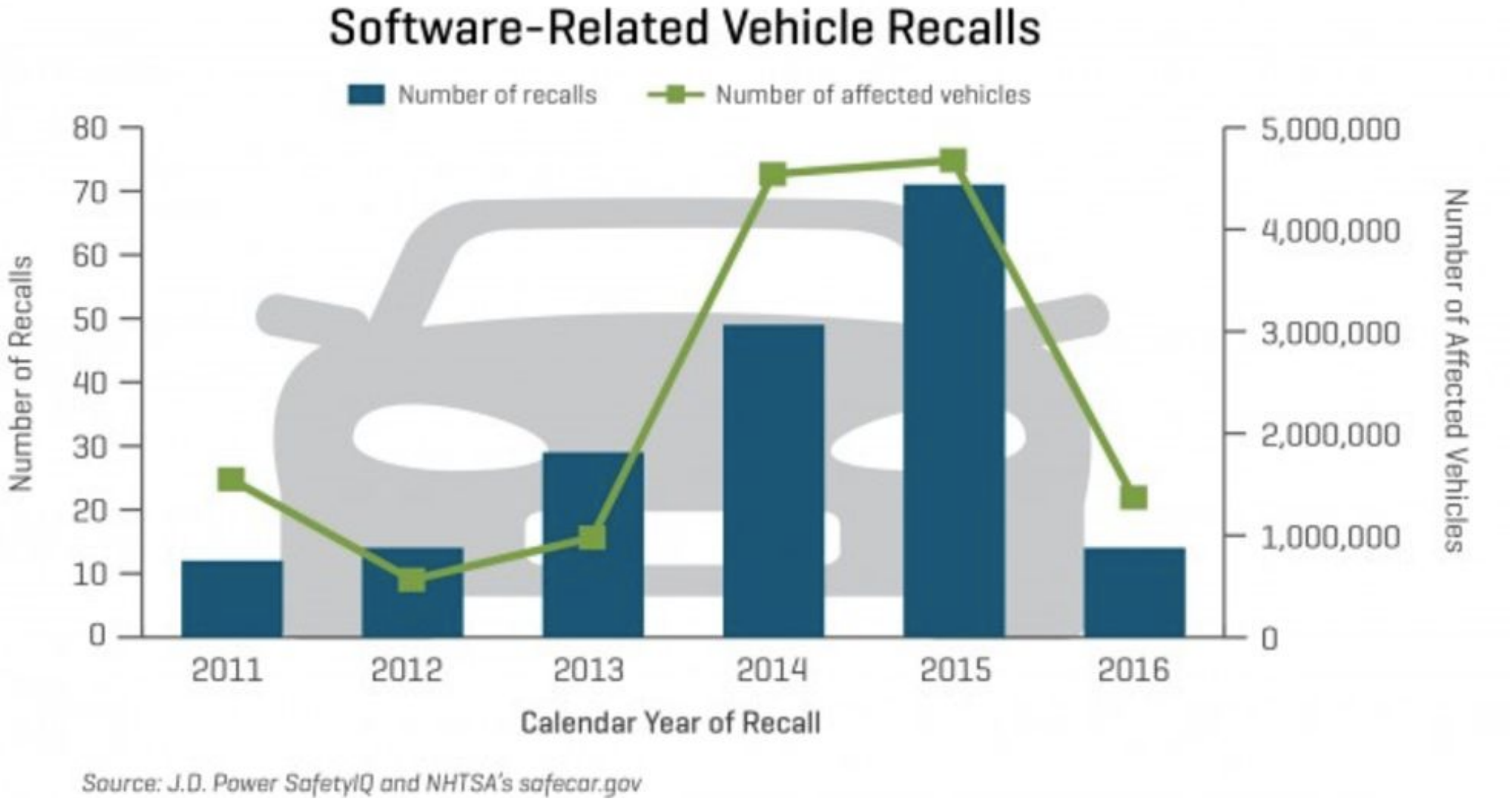
We enable mission-critical workloads

Used by 3,000 customers in over 57 countries across diverse industries to develop mission-critical applications and software

GENIVI®

Rapid software testing and conformance with static code analysis

SOFTWARE NOW TO BLAME FOR 15 PERCENT OF CAR RECALLS



How can static code analysis improve software quality?

What are the factors affecting software quality, complexity, and security?

- Greater use of software in vehicles
- Pressure to release on time (or as soon as possible!)
- Market demand for new features
- Greater use of third-party libraries

GENIVI®

How can static code analysis improve software quality?

- Find common issues in code
  - Buffer overflows (security exploit or program crashes)
  - Null pointer dereferences (your program crashes)
  - Memory leaks (processor runs out memory and locks up)
  - Uninitialized data usage (data injection)
  - Platform/OS specifics (privilege escalation, etc…)
  - Concurrency (deadlock)

How does static code analysis work?

- Automatically inspects source code to find potential defects
- Different types of analysis
  - Walks down every path of your code
  - Inter-procedural
  - Inter-file
- SCA runs the tests that your developers **don't (or won't) write**
- SCA will find defects that other testing won't

How can static code analysis find bugs my testing doesn't?

- Traditional testing tools require reproduction of the exact runtime conditions that cause the issue to occur
- This in turn requires developers to write specific tests that will exercise the code in the specific way that reveals the defect at runtime
  - This is time-consuming for developers
  - Even comprehensive testing may not trigger the specific runtime conditions that cause the defect

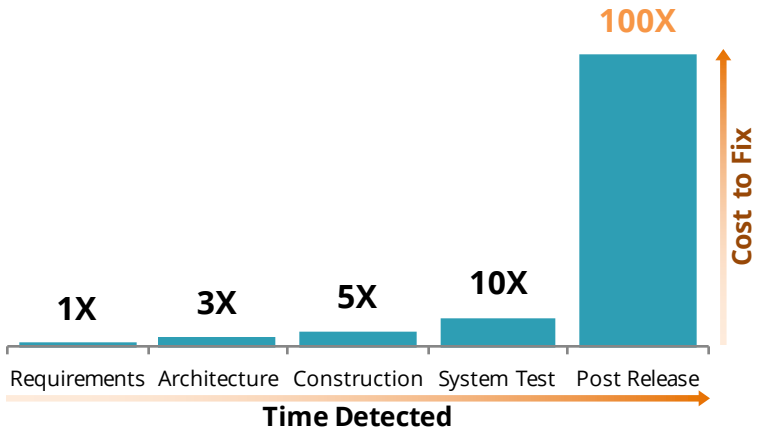
- Static code analysis helps by finding defects that are hard to find with the human eye
  - These defects are mot generally found by code review
  - Many are traditionally found with dynamic testing after a failure has occurred in testing or the field – but its too late!

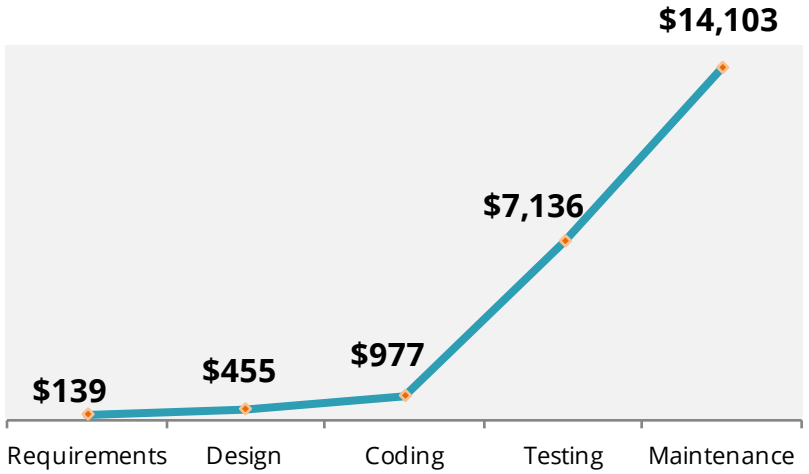
Source code analysis benefits: security & quality

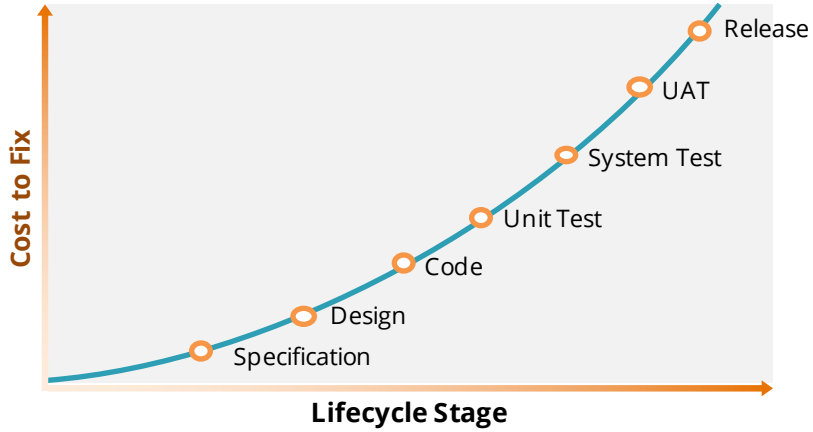
- Significantly reduces the cost of reliable, secure software
  - Complements existing testing approaches
  - Automated and repeatable analysis
- Enforces key industry standards
  - DISA STIG, CWE, MISRA
  - CERT, SAMATE
  - OWASP, DO-178B, FDA validation
  - ...and more

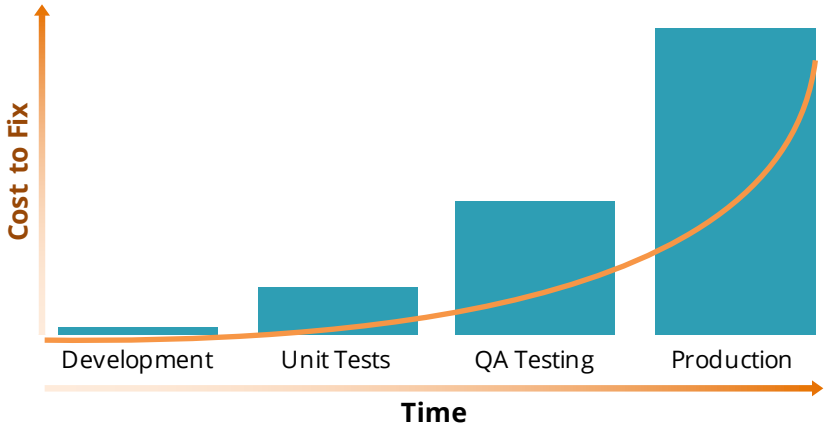

Continuous Static Code Analysis

The faster you find a defect, the less costly to fix









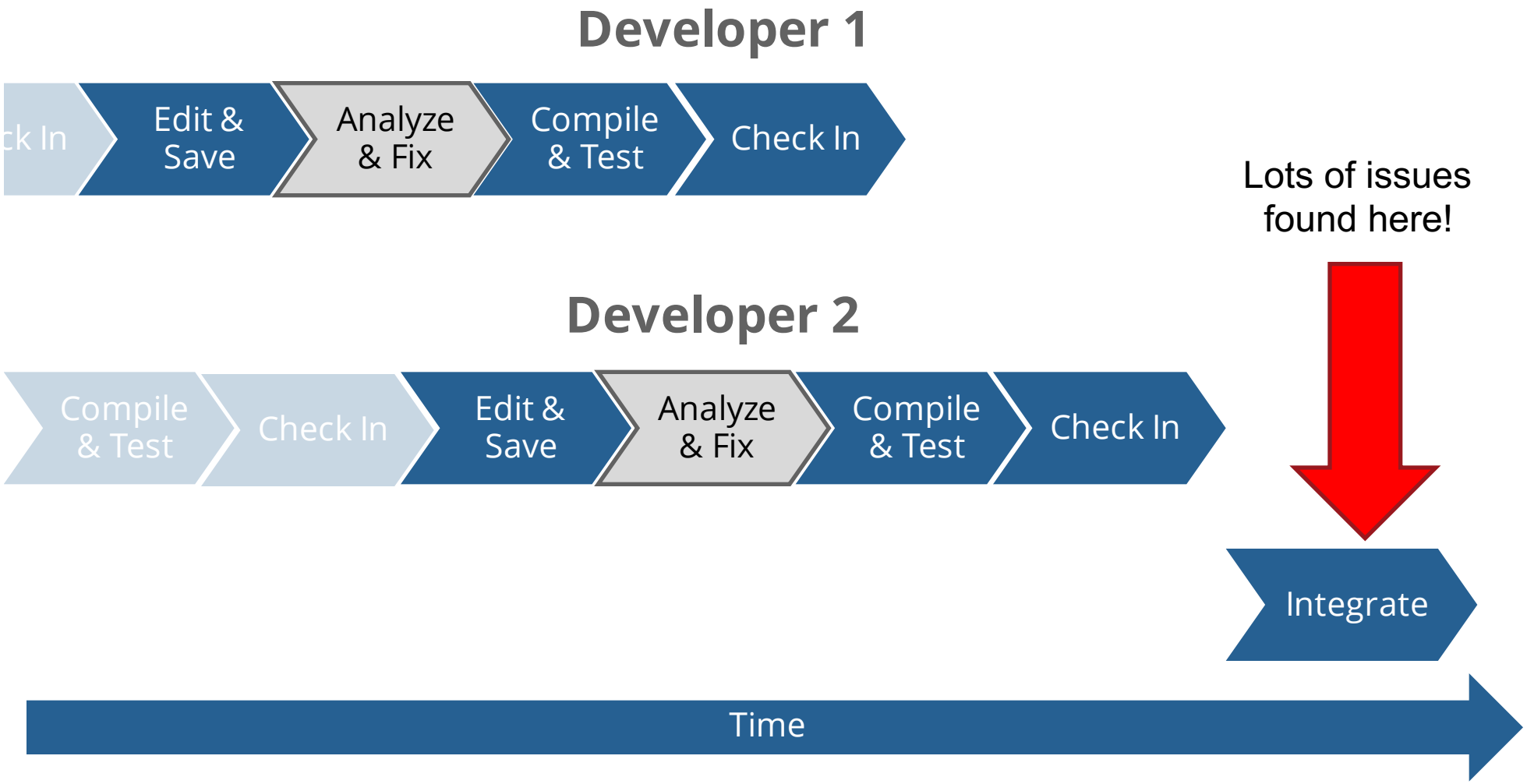
Traditional analysis done after compile/build

- Late stage “rework” reduces tool adoption
- Timelines compromised
- Issues are more expensive to fix

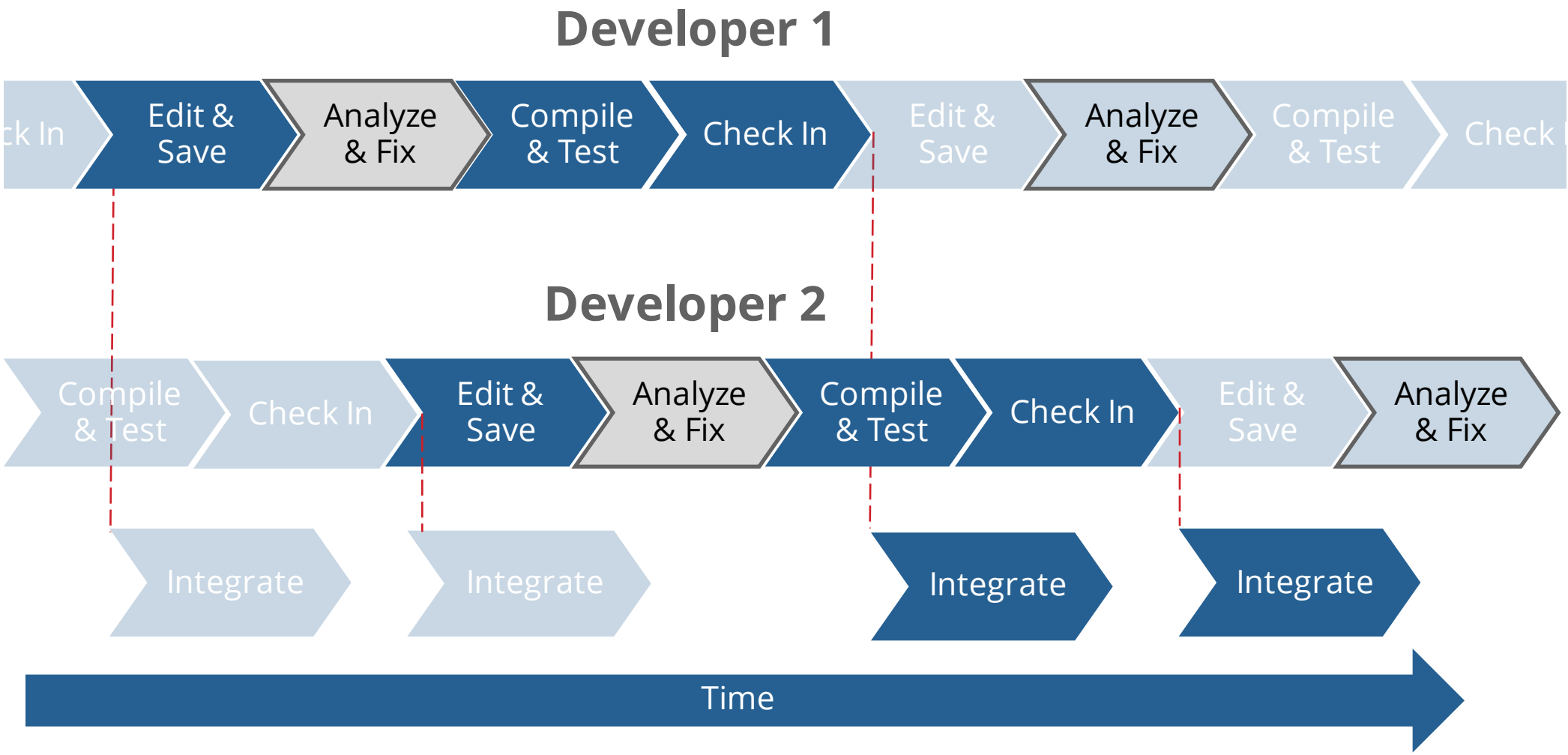
Why not perform analysis earlier in the cycle on the desktop?

- ☑ Eliminates new defects from being checked back into the team level build
- ☑ No extra work for developers
- ☑ In-context checking and fixes
- ☑ Continuity of development flow

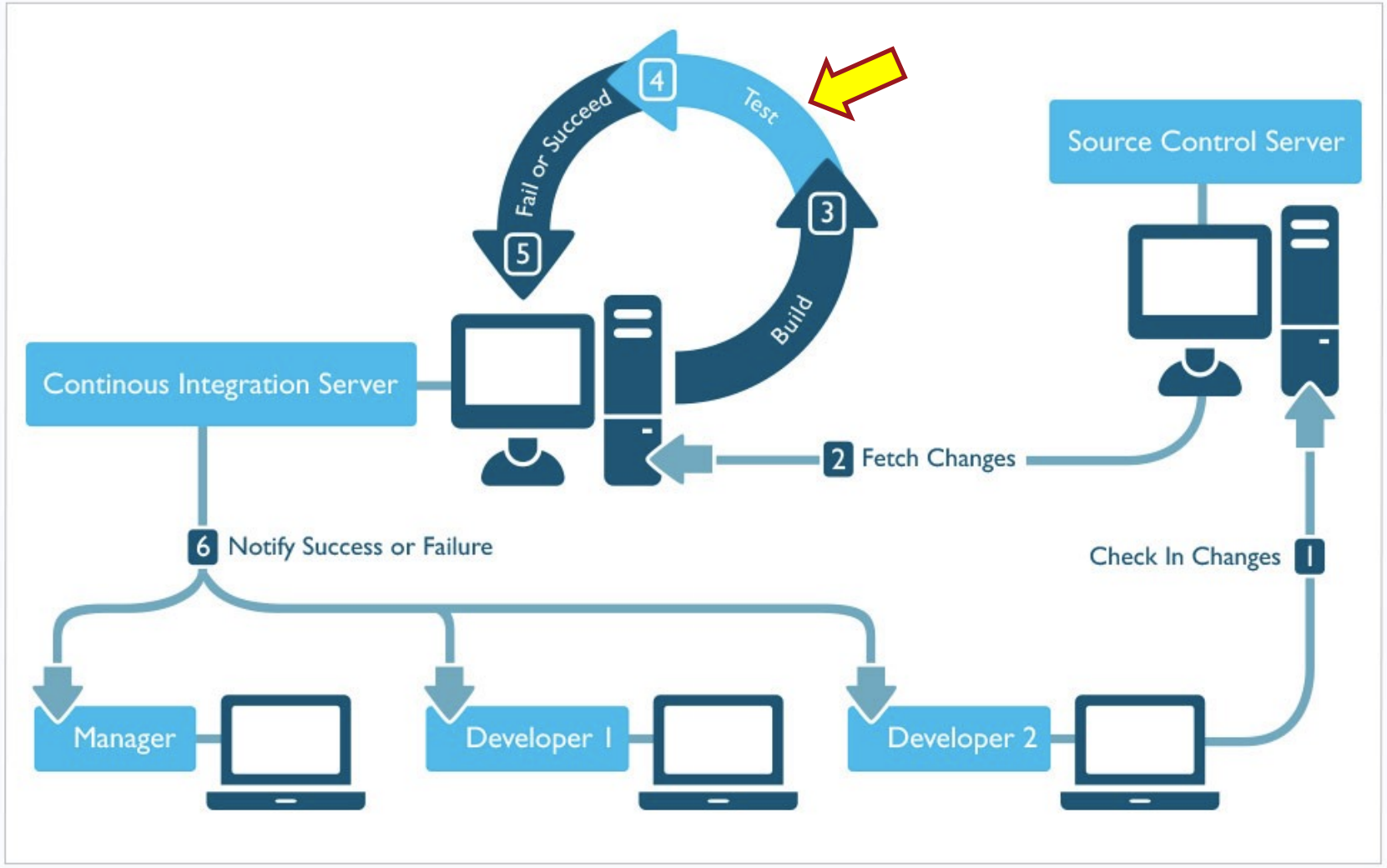
What about defects found during integration?



Continuous Static Code Analysis



Continuous Static Code Analysis



Continuous Static Code Analysis

- Improves the predictability of software release schedules
- Improves the quality and security of release software
- Reduces the cost of finding and fixing software defects

Thank you!

Walter Capitani,
Product Manager, Klocwork
Rogue Wave Software

GENIVI®