

Why Every Dev. Team Needs Static Analysis



This Presentation Will Cover:

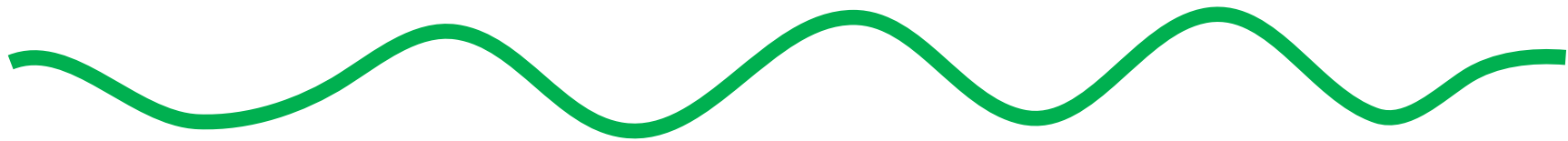
- The Cost of Bugs in Software Development
- The Advantages of Testing and Static Analysis
- Debunking Static Analysis Myths
- What Makes a Good Static Analysis Tool

The Cost of Bugs in Software Development

This is how we want development to be

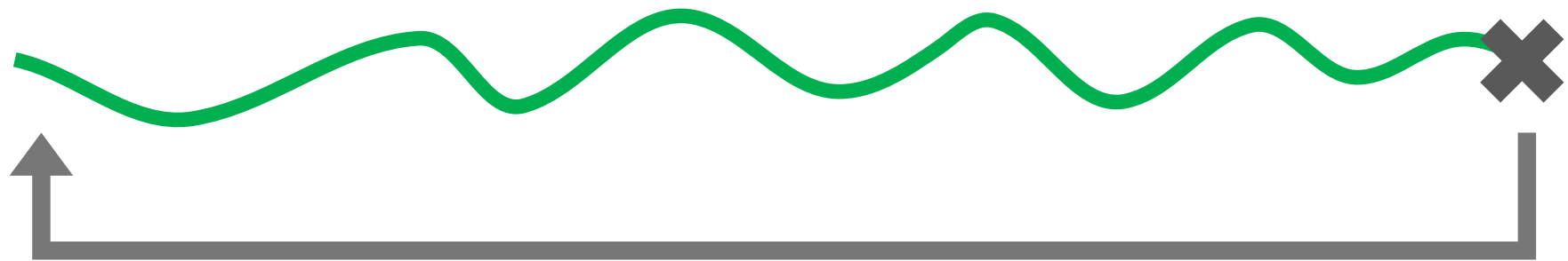
A smooth journey from beginning to end

However, development more often than not turns out to be this



A bumpy journey from beginning to end

Sometimes we even have to go back to the beginning and **start again**



Bugs and Errors
found in the **Quality Assurance Process**
make the project cost
exponentially more
time and money than
it should

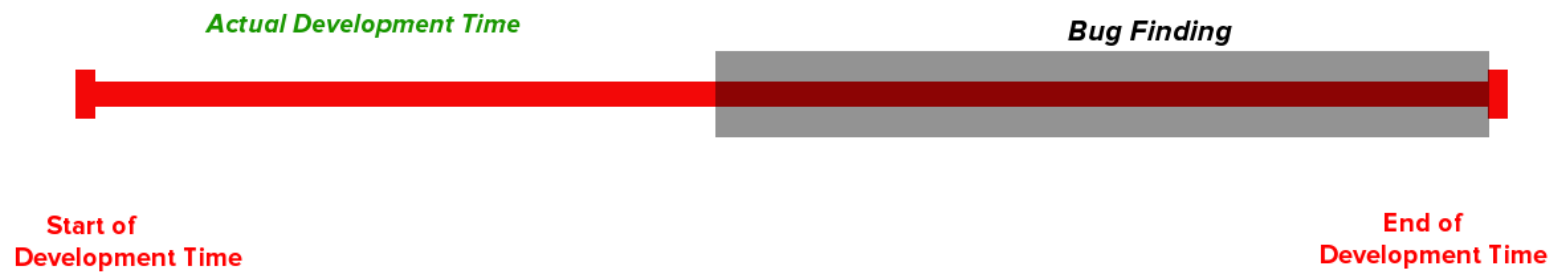


The software industry spends approximately **50%** of funds for development and maintenance on finding and fixing bugs



Bug finding costs about **\$312 billion** a year

It takes up **50%** of a developer's programming time



Most forms of testing only find about **1 bug** out of every **3** And all tests together barely remove **85%** of bugs during testing

Even the best companies and organizations have released products with expensive (but sometimes simple to fix) bugs



The Advantages of Testing and Static Analysis

“ *A synergistic combination of defect prevention, pre-test defect removal, and formal testing by certified personnel can top 99% in defect removal efficiency while simultaneously lowering costs and shortening schedules.*

”

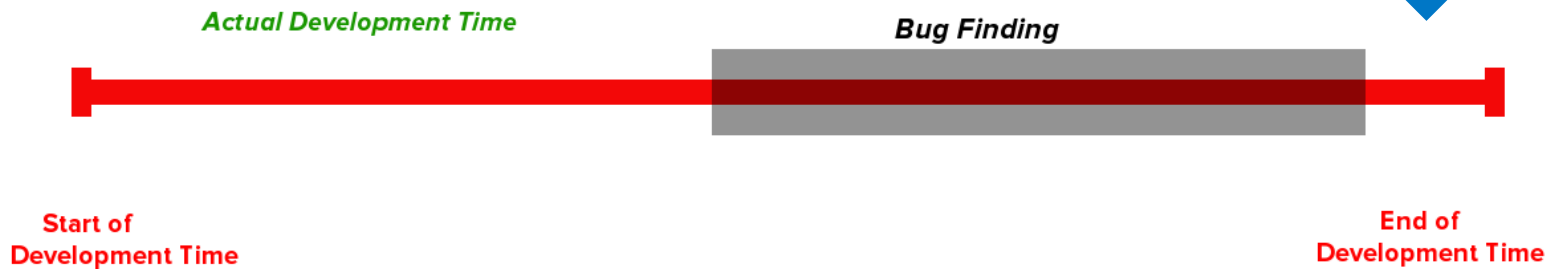
- Capers Jones, [Software Defect Origins and Removal Methods](#)

Using static analysis, unit testing, code inspections, peer review, QA, pre-test defect removal and prevention can reduce costs by as much as **50%**

Using static analysis and dynamic analysis can save a team up to **500%** more time.

Static analysis tools, specifically, have been shown to push defect detection and removal rates above 65%

Extra Time!



Debunking Static Analysis Myths

Myth #1

**Static Analysis Tools Return
Too Many False Positives**

Strong static analysis tools let you customize code rules and metrics in order to fit your project's needs and your coding style

A lot of false positives might mean:

- The tool hasn't been configured specifically for your project
- Developers don't understand how the tool works
- The tool is not being used properly in the development cycle

Myth #2

**Static Analysis Tools Are Not
Affordable or Cost-Effective**

Static analysis tools are an investment in your company. They help...

- Better communication between teams
- Lessen time spent finding and fixing bugs
- Meet deadlines more consistently
- Cut costs in the QA process
- Cut costs in post-delivery
- Create happier customers and end-users



What Makes a Good Static Analysis Tool

A good static analysis tool should be:

- Small
- Reliable
- Customisable
- Dependable
- Powerful

And it should help you:

Save Time

Save Money

Make your team happy

Make a better product

And help turn this



Into this



Presented by



CppDepend is a static analysis tool for C/C++. CppDepend supports a large number of code metrics, allows for visualization of dependencies using directed graphs, and dependency matrices. It also performs code base snapshots comparison, and validation of architectural and quality rules.

Sources:

- Capers Jones, [Software Defect Origins and Removal Methods](#)
- Andrey Karpov, [200 Open Source Projects Later: Source Code Static Analysis Experience](#)
- William B. Oliver Lawrence , Lawrence Livermore National Laboratory ,Quantifying the Value of Static Analysis, Date 5/19/2011
- [Challenging SCA Myths](#), published by Rogue Wave Software