Améliorez la qualité de vos logiciels embarqués grâce à l’analyse statique
Developing secure code - Gary McGraw touchpoints

Code review

Penetration testing

Architecture risk analysis
Developing secure code

Code review #1

Penetration testing #3

#2

Architecture risk analysis
Automating code review

Code review

with a tool
A code review tool – Polyspace Bug Finder

Identify defects

Enforce coding rules

Produce and monitor quality metrics
Automatic code review – Overall picture

Delivery
Enabling a fix-as-you-go process

Polyspace Bug Finder
Ford deploys static code analysis at enterprise level

Ford Motor Company - Powertrain Controls. Calibration and NVH (PCCN)

Responsible for developing Software for Gasoline/Engine/Driveline Controls Worldwide

200+ Software Developers Worldwide

Gasoline Engine Controls
Dearborn, MI

Transmission and Driveline Controls
Livonia, MI

Hand Code + Model Based Processes
Windows + Linux Development Environments

Diesel Engine Controls
Basildon, Essex, UK

Watch video
Solar Impulse saves 1 to 2 man-year with Polyspace

Watch video
Finding bugs

Polyspace
Bug Finder
Where does the find-and-fix process stop?
The magic box – Myth or reality?

Source code

Polyspace Code Prover
Proving code vs. finding bugs

Polyspace Code Prover
static void pointer_arithmetic (void)
{
    int array[100];
    int *p = array;
    int i;

    for (i = 0; i < 100; i++) {
        *p = 0;
        p++;
    }

    if (get_bus_status() > 0) {
        if (get_oil_pressure() > 0) {
            *p = 5;
            p = 10;
        } else {
            i++;
        }
    }

    i = get_bus_status();

    if (i >= 0) {
        *(p - i) = 10;
    }
}
Robust verification of software components

SW modules

Quality gate

Integrated SW
Contextual verification of application
Key takeaway

Automate Code Review

with Polyspace tools