# Comprehensive applications threat analysis

Designing security into software

**Benefits**

- Proactively anticipates and avoids the threat of thousands of vulnerabilities
- Dramatically reduces the cost of development, operation and maintenance
- Increases security

In today's heightened threat landscape, with more users, more devices and more applications, vigilance is more important than ever. Vulnerabilities, or security defects, are deficiencies that allow attackers to perform unauthorized actions  and evade protection mechanisms. The cost of these vulnerabilities can be damaging—ranging in severity from jeopardizing the mission, to data loss, or delaying deployment. Once applications are in production, undiscovered vulnerabilities can lead to serious financial and operational risks. The earlier a defect is discovered, removed or avoided, the better.

By using a consistent process to determine potential vulnerabilities in an application's design, you can have greater confidence in the security posture of your applications. This lowers the cost of development while reducing the number of vulnerabilities, which results in a decreased possibility of data breaches and lower cost of security defect remediation.

### Early detection is key

Perspecta's comprehensive applications threat analysis (CATA) service helps you identify critical, yet missing, security requirements when it's most beneficial—early in the development life cycle. Our service will also help with architectural choices that unnecessarily—and often dramatically—increase the number and severity of both known and unknown vulnerabilities.

Our service delivers security requirements gap analysis (SRGA) and architectural threat analysis (ATA) reports to enable developers to make informed, cost-saving adjustments during the software / application design phase. Together, SRGA and ATA identify the specific actions needed to address problems from both regulatory perspectives and architectural threat model views.

Unlike industry-standard methodologies, our CATA service attacks the problem from a requirements and architectural angle, eliminating entire classes of vulnerabilities rather than just individual weaknesses in software.

### How it works

Our CATA process starts with the overall application architecture to ensure we have a holistic understanding of the security requirements and threats.

Using our proprietary tools, CATA SRGA identifies missing security requirements and determines their operational impact, promoting risk-based decision making. Our CATA ATA tools model the applications architecture, highlighting areas of risk and allowing to be addressed systematically early in the process—resulting in immediate cost savings versus the traditional approach of finding and fixing vulnerabilities as they are discovered.

### Why Perspecta

Perspecta refined the SRGA and ATA processes over the last decade, remaining focused on securing applications cost effectively, allowing us to bring years of refinement and expertise to your software development life cycle.

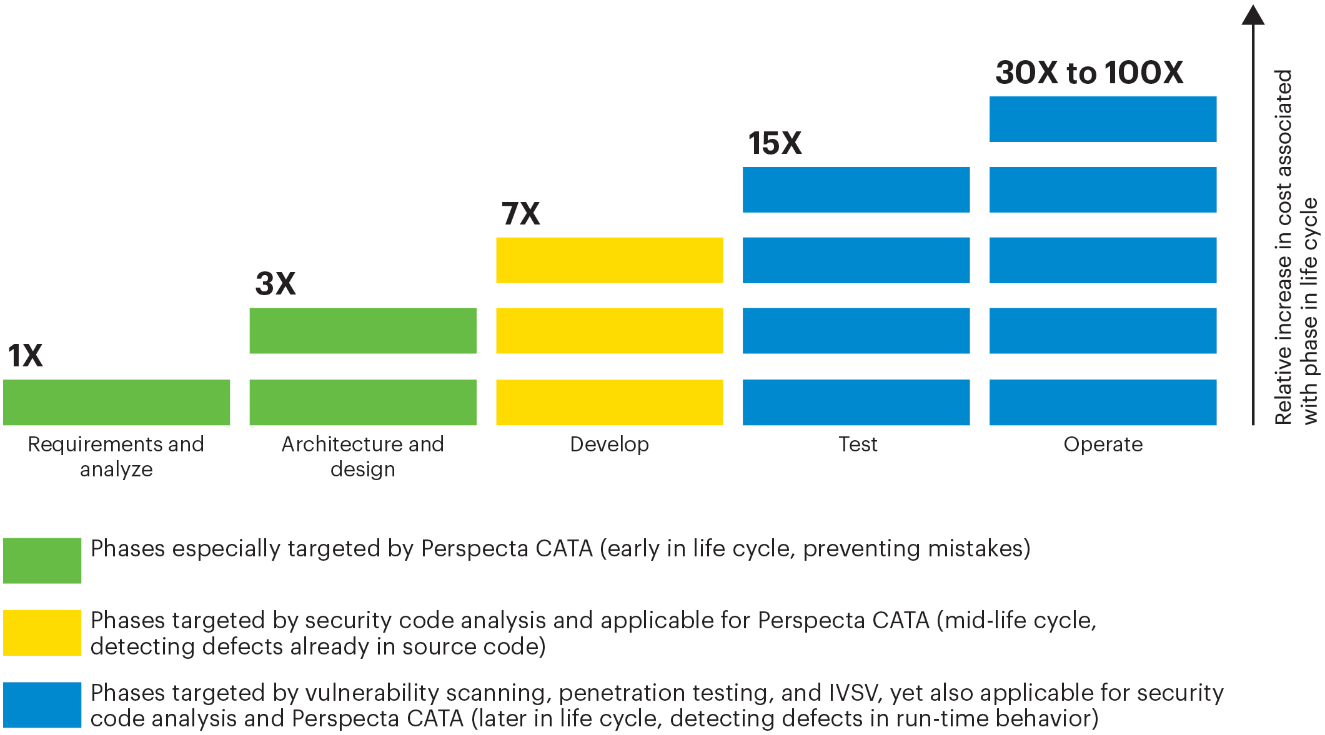
Fixing the problem early is less expensive, more effective and safer than fixing them after applications go live. Perspecta's unique knowledge base brings a tailored solution delivered by experienced professionals ensuring your applications have security built-in from the ground up.

perspecta™

# The cost of remediation

Various studies have shown that software defects found and fixed after deployment cost a significant amount less than if they had been found (or avoided) at the beginning of the life cycle. In fact, the National Institute of Standards and Technology (NIST) estimates it is 30 times or more expensive to bolt on quality after deployment.

The graphic below illustrates the relationship between the phase in which the remediation occurs and the cost of the remediation.



🟩 Phases especially targeted by Perspecta CATA (early in life cycle, preventing mistakes)

🟨 Phases targeted by security code analysis and applicable for Perspecta CATA (mid-life cycle, detecting defects already in source code)

🟦 Phases targeted by vulnerability scanning, penetration testing, and IVSV, yet also applicable for security code analysis and Perspecta CATA (later in life cycle, detecting defects in run-time behavior)