# Statikus analízis

# Statikus analizís

- automatizált
- programkódban lévő defektusok megtalalása
- forráskód futtatása nélkül

# Ellenőrzések

- hibakeresés
- biztonsági felülvizsgálat
- program érthetőség
- stílus

# Program érthetőség

```
// Ternary operators should not be nested (RSPEC-3358)
public String getReadableStatus(Job j) {
  return j.isRunning() ? "Running" : j.hasErrors() ? "Failed" : "Succeeded";  // Noncompliant
}
```

# Hibakersés

```
// Loops should not be infinite (RSPEC-2189)
for (;;) {  // Noncompliant; end condition omitted
// ...
}

// Loop conditions should be true at least once (RSPEC-2252)
for (int i = 0; i < 0; i++) {  // Noncompliant: the condition is always false
    // ...
}
```

# Biztonsági felülvizsgálat

- SQL injection
- plaintext password
- weak hashing algorithms
- loose POSIX file permissions

# Statikus analizís előnyei

- forráskód futtatása nélkül
- a fejlesztési folyamat korai fázisában
  - alacsony a javítás költség
- kód karbantarthatóság

# Statikus analizís hátrányai

- hamis riasztások
- hamis biztonságérzet

# Népszerű eszközök

- Java: PMD, CheckStyle, SpotBugs
- JavaScript: ESLint
- SonarQube, Code Climate