EXAMENSARBETE Static analysis on a large codebase
STUDENT Ella Eriksson, Jashar Vahid
HANDLEDARE Sardar Muhammad Sulaman (LTH), Antonio Vicent (Ericsson)
EXAMINATOR Per Runeson (LTH)

Statisk analys i produktionen

POPULÄRVETENSKAPLIG SAMMANFATTNING **Ella Eriksson, Jashar Vahid**

Statisk analys är ett sätt att hitta buggar i programkod utan att köra koden. Det här arbetet testar befintliga verktyg i en riktig produktionsmiljö för att avgöra hur användbara de är.

Statiska analysverktyg kan vara användbart för att hitta fel i programkod. Ett stort problem med sådana verktyg är att de riskerar att varna för onödiga saker och därmed sänka förtroendet hos utvecklarna som använder verktyget. I vårt examensarbete letar vi upp och utvärderar två statiska analysverktyg, Clang Static Analyzer och Cppcheck. Dessa verktyg har så pass bra träffsäkerhet i sina felrapporter att de fungerar att använda även på en stor kodbas. Verktygen lider dock av problemet att de är relativt långsamma. Analysen på den kodbas vi testade tog 49 respektive 14 minuter, för de två verktygen. Genom att köra verktygen automatiskt nattetid kan man dock få fördelarna av analysen utan att behöva offra värdefull tid på att vänta på resultaten. Om man begränsar analysen till den kodfil man för tillfället arbetar med kan man dock söka igenom koden på under en sekund. Detta gör att verktygen kan användas på olika sätt beroende på önskan och behov.

Många företag har kodbaser på hundratusentals eller miljontals rader kod. Att leta efter buggar i så stora kodbaser är ett tidskrävande och svårt arbete. De utvecklare vi har pratat med upplever att de lägger en betydande del av sin arbetstid på att granska och rätta kod. Många fel i programkod är desvärre svåra att se genom en titt på koden och att skriva testfall som täcker alla möjliga scenarier är kanske ännu svårare.

Genom att automatiskt söka igenom koden för buggar kan programmerarnas tid användas mer effektivt. Statisk analys är ingen ersättning för manuell kodgranskning men ett mycket bra komplement. Programmerarna kan koncentrera sig på att förbättra lösningar och försäkra sig om att koden skrivs på bästa sätt, istället för att lägga tid på att hitta småfel som kan identifieras automatiskt.

Att statisk analys fungerar även på större kodbaser i riktig produktionsmiljö innebär att många företag kan effektivisera sin utvecklingsprocess. Det faktum att de verktyg vi har testad är gratis innebär också att tröskeln för att prova dem i den egna miljön blir lägre. Genom intervjuer med utvecklare har vi kommit fram till att felen som hittas av verktygen i hög grad är sådant som skulle rättats till direkt om en sådan felrapport hade funnits tillgänglig när koden skrevs.

I det här arbetet valde vi att titta på flertalet statiska analysverktyg, både gratisverktyg och de som kräver licens. Under arbetets gång föll många bort genom strul med licenser, dålig funktionalitet eller andra skäl. Det slutade med att två gratisverktyg, Clang Static Analyzer och Cppcheck, gick hela vägen och testades både i den befintliga automatiska testmiljön, och manuellt på flera kodbaser.