

1 Hva er statisk analyse?

Statisk analyse er analyse af kode uden dette bliver eksekveret. Det er altså analyse af kodens kompleksitet mere end funktionalitet.

2 Værktøjer

2.1 FxCop

FxCop er et statisk analyseværktøj udviklet af Microsoft til Visual Studio. Værktøjet er lavet med formålet at lave med det formål at øge performance, sikkerhed og design.

2.2 Compilers

Dette kan fx være en moderne compiler, som løbende holder øje med koden om der er "unused variables" eller mistænksomme konstruktioner i koden.

2.2.1 Maintainability

Maintainability index er et index, som fortæller hvor holdbar og effektiv ens kode er og hvor nemt det er at vedligeholde ens kode.

Indexet udregnes på følgende måde:

$$MAX(0, (171 - 5.2 * \ln(HalsteadVolume) - 0.23 * (CyclomaticComplexity) - 16.2 * \ln(LinesofCode)) * 100 / 171) \quad (1)$$

Tidligere var der ikke en begrænsning på 0-100, men 171 til minus ubestemt tal. Men da man indså, at når tallet nærmede sig 0, var koden meget svært at vedligeholde. Derfor ændrede man indexet til 0-100.

Fordelingen af indexet er:

	20 - 100
	10 - 19
	0 - 9

2.2.2 Cyclomatic Complexity

Vi anvender cyclomatic complexity til at få et billede af hvor svær koden er at teste, vedligeholde eller fejlfinde på, samt en indikation af hvor sandsynligt det er at koden vil lave fejl.

Den mindst mulige kompleksitet vil altid være to (2).

if	plus 1 i complexity
while	plus 1 i complexity
switch case	plus 1 i complexity for hver case

Figure 1: dennis' gøgl

Når vi skal illustrere kompleksiteten i en metode eller et system, kan vi lave en Cyclomatic Complexity graf. den skal se ud som på figur 2 herunder

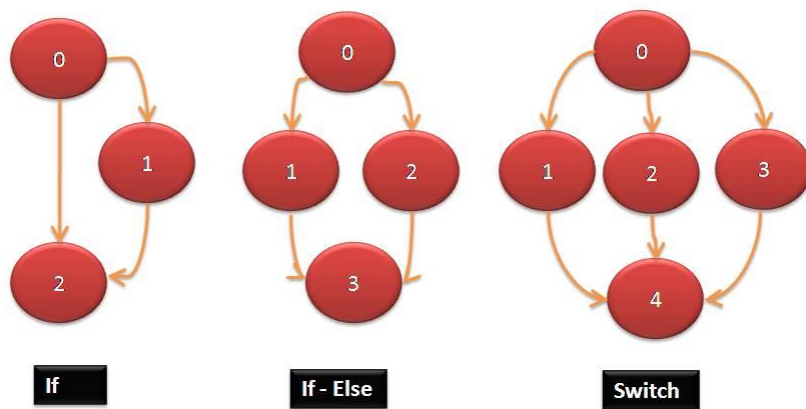


Figure 2: Graf for tre statements

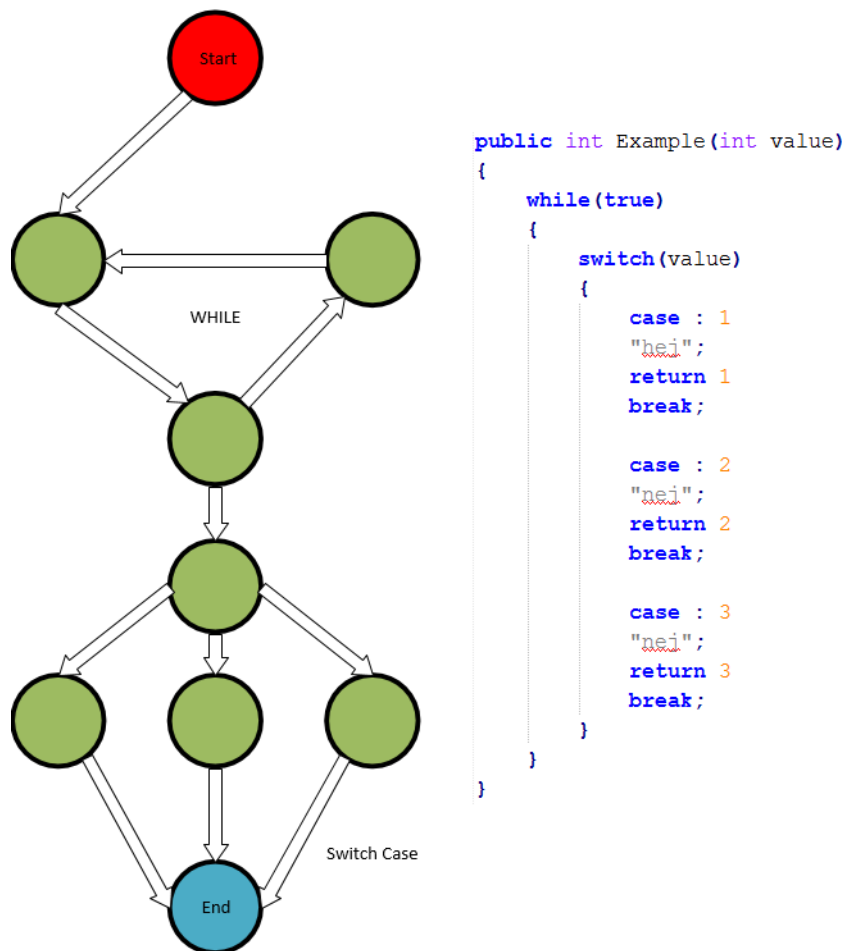


Figure 3: Eksempel på Cyclomatic complexity

2.2.3 Lines of code

Det angiver antallet af intermediate language(sprog mellem C og maskinkode) linjer i koden og IKKE source koden. Dette kan anvendes til at indikere om en metode laver for meget og evt splittes i 2. En lang metode kan være svær at vedligeholde.

2.2.4 Class coupling

Class coupling måler koblingen til unikke klasser bl.a. igennem parameter, retur typer, base classes, interface implementation mm.

Man skal altid sigte efter stor samhørighed og lav kopling i et godt software design. Høj kobling kan indikere at systemet er svært at vedligeholde og/eller genbruge.

2.2.5 Depth of inheritance

Et tal som angiver antallet af klasser defineret som hænger sammen med roden af klasse hierarkiet. Jo dybere i hierarkiet, des sværere er det at finde ud af hvor klasserne er defineret.