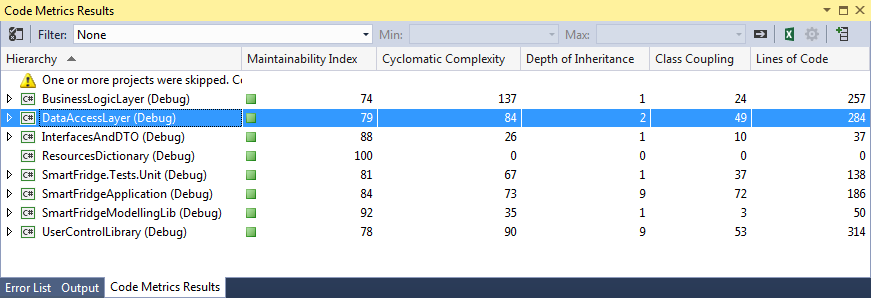
# Integrationstests

## Unittest

## Static analysis

Her ses resultaterne for en statisk analyse foretaget af Visual Studio:



Figur – Static analysis, code metrics

De følgende vurderinger af resultaterne tager udgangspunkt i MSDN’s retningslinje.r (Microsoft, 2015)

### Maintainability index

Giver indikation af hvor let det er at vedligeholde koden.

Da en værdi mellem 20 og 100 indikerer god maintainability, har samtlige projekter i vores solution en god maintainability, med et upper peak på 84, da SmartFrdigeModellingLib ikke er applikationskode.

### Cyclomatic Complexity

Giver indikation af hvor kompleks kodetraverseringen er, altså om der er mange cyklusser.

Generelt ikke enormt høj i solutionen, men BusinessLogicLayer har en semi-høj CC i forhold til DataAccessLayer. Dette kan skyldes et højt antal af hjælpefunktioner i BLL.

### Depth of Inheritance

Giver indikation af arveheraki-kompleksiteten. Generelt lav da der ikke bliver arbejdet meget med arv i vores solution, da det ikke har været meget relevant.

### Class Coupling

Giver indikation af koblingen mellem klasser. Her er resultatet højt ved SmartFridgeApplication, da det er hovedprogrammet i vores solution. Med andre ord er det den store klasse der afhænger af en del små klasser. Her kunne man have valgt at interface flere klasser.