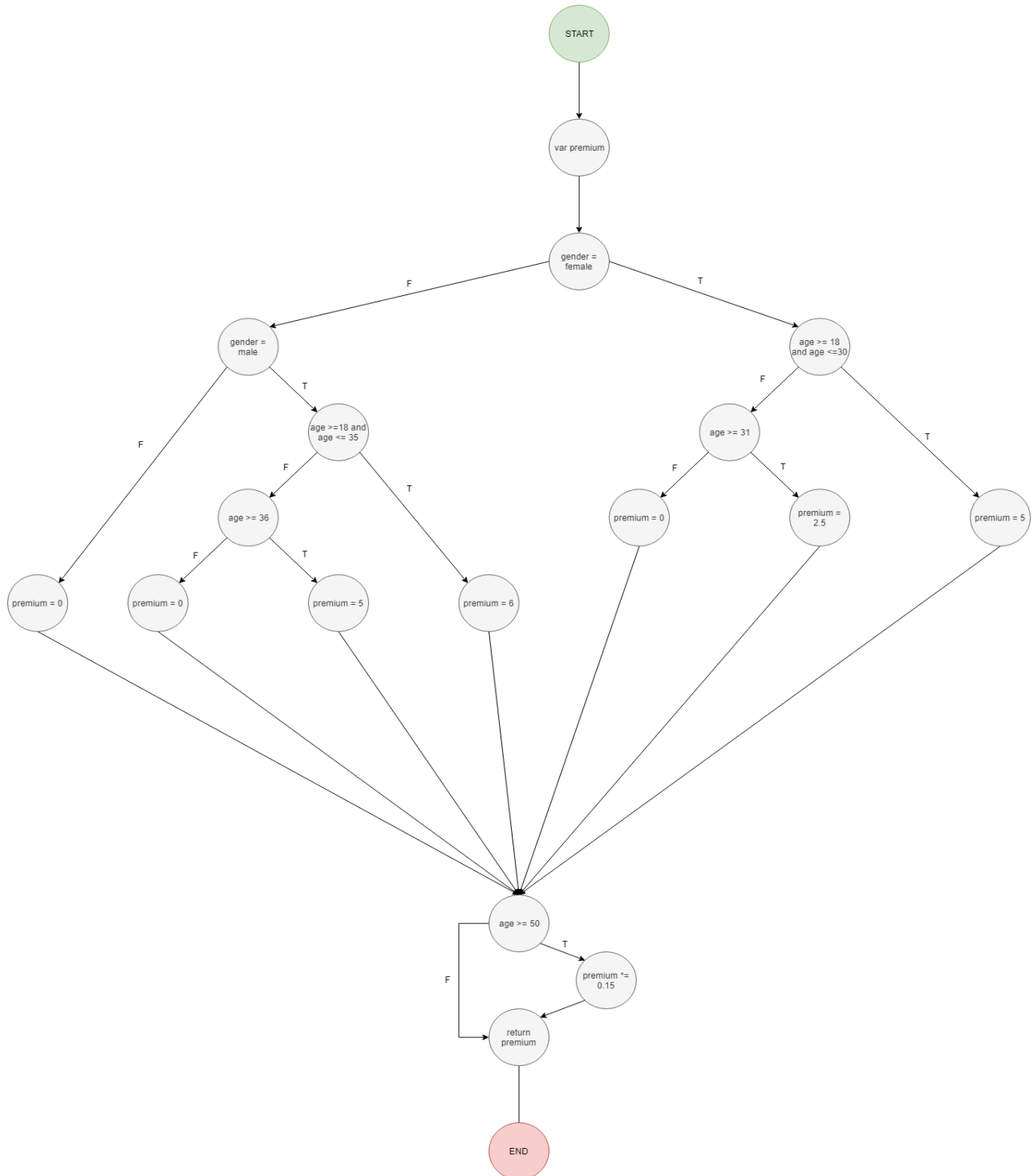# Software QA & T Assignment

Flow Graph

Figure 1 Flow Graph. For the uncompressed version, see GitHub Repo

Calculating Cyclomatic Complexity

$$Cyclomatic\ Complexity\ =\ \#Edges - \#Nodes + 2(\#Exit\ Points)$$

$$Cyclomatic\ Complexity\ =\ 25 - 17 + 2(1)$$

$$Cyclomatic\ Complexity\ =\ 10$$

Hierarchy ▲	Maintainability Index	Cyclomatic Complexity
▷ {} insurance_premium_calculator.Controllers	■ 96	2
▲ {} insurance_premium_calculator.lib	■ 64	11
▲ InsuranceService	■ 64	11
CalcPremium(int, string) : double	■ 55	10
InsuranceService()	■ 100	1

Figure 1 Visual Studio Confirms that the Cyclomatic Complexity is 10

Black Box Testing with Fitnesse

Test System: fit:fitsharp\build\debug\Runner.exe

variable defined: COMMAND_PATTERN=%m -r fitnesse.fitserver.FitServer,fitsharp\build\debug\fit.dll %p
variable defined: TEST_RUNNER=fitsharp\build\debug\Runner.exe
classpath: D:\repos\insurance-premium-calculator\insurance-premium-calculator.Tests\bin\Debug\insurance-premium-calculator.Tests.dll

insurance_premium_calculator.Tests.CalcPremiumFixture		
age	gender	premium?
17	male	0
18	female	5.0
49	male	5
50	female	1.75 *expected* 0.375 *actual*
60	apple	0

Figure 3 Acceptance test results of black box test cases

Jenkins CI

Figure 4 Jenkins build results