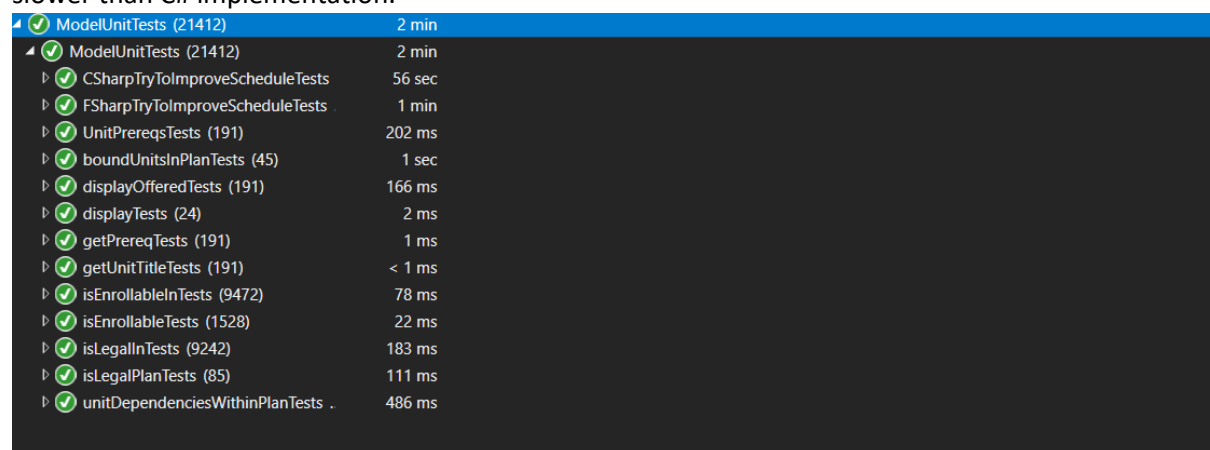


This report will compare C# (imperative programming) and F# (functional programming), regarding the execution performance, the ease of development, clarity, understandability and maintainable.

Execution Performance

F# code is generally slower than C# code. The following is one epoch of the tests, although it shows almost no difference in this epoch, in some of the cases, F# implementation may even be 20 seconds slower than C# implementation.



| | |
|-----------------------------------|--------|
| ✓ ModelUnitTests (21412) | 2 min |
| ✓ ModelUnitTests (21412) | 2 min |
| ✓ CSharpTryToImproveScheduleTests | 56 sec |
| ✓ FSharpTryToImproveScheduleTests | 1 min |
| ✓ UnitPrereqsTests (191) | 202 ms |
| ✓ boundUnitsInPlanTests (45) | 1 sec |
| ✓ displayOfferedTests (191) | 166 ms |
| ✓ displayTests (24) | 2 ms |
| ✓ getPrereqTests (191) | 1 ms |
| ✓ getUnitTitleTests (191) | < 1 ms |
| ✓ isEnrollableInTests (9472) | 78 ms |
| ✓ isEnrollableTests (1528) | 22 ms |
| ✓ isLegalInTests (9242) | 183 ms |
| ✓ isLegalPlanTests (85) | 111 ms |
| ✓ unitDependenciesWithinPlanTests | 486 ms |

Ease of development

It may be hard to learn functional programming such as F# as it has quite a steep learning curve since it is a completely different style of programming from imperative programming. However, after hours or even days of development with F#, it becomes easier. All procedures in F# are based on functions and every procedure is handled asynchronously, which is easier to track and debug. On the other hand, when dealing with complex function, C# has more choices to implement the same function (as it can directly mutates the state).

Clarity

With higher-order function and pipeline operators, the developers can easily know the effects of their code at each stage. C#, on the other hand, may give extra complexity to find out what the whole function is about. However, F# does have a downside of not providing type declarations clearly, the type declarations are not necessary for developers but still can be implemented (in a less intuitive way).

Understandability

From the perspective of understandability, let's compare them when their implementations all have high-level comments and concise variable and function names. F# program is still better in this case, since it can use higher-order functions and pipeline operators. It gives extra understandability to other developers what the code is about. For example, F# code uses List.map which is a higher function to transform every element in the list to another element if a function is given. In the same case, C# code implement it by using for loop and mutation of the state, which makes it hard to track the state and understand what the entire procedure is about.

Maintainable

It is easier to maintain the F# program than C# program. As I mentioned, F# uses a lot of higher-order functions and pipeline operators, which ensures the clarity and understandability. Sometimes changing the features in F# program only needs a line of code, while C# can take a few lines of code. As F# code is short and clear, other developers can also easily pick up with the program and maintain it.