

Premessa

C'è un fenomeno molto particolare legato a Microsoft come una sorta di eco. Microsoft è un'azienda enorme e l'ecosistema dei Microsoft Certified Partner moltiplica tale dimensione di svariati ordini di grandezza. Se siete sufficientemente integrati in questo ecosistema, può essere difficile vedere oltre i suoi confini. Ogni volta che cercate una soluzione a un problema relativo a un prodotto o a una tecnologia Microsoft, è probabile che troverete una risposta che implica l'utilizzo di ulteriori prodotti Microsoft. Qualsiasi cosa urliate nella camera dell'eco, la risposta sarà sempre *Microsoft!*

Quando Microsoft mi ha assunto [Mark] nel 2003, ero già saldamente integrato in questa camera a eco, avendo lavorato per anni come Microsoft Certified Partner, che adoravo! Ben presto mi hanno spedito a una conferenza tecnica interna, a New Orleans, per conoscere l'ultima eccezionale tecnologia Microsoft.

Oggi non ricordo nessuna delle sessioni sui prodotti Microsoft cui ho partecipato, ma ricordo l'ultimo giorno. Quel giorno, non essendo riuscito a provare alcuna sessione che potesse soddisfare la mia sete di nuove tecnologie, non vedevo l'ora di tornare a casa, in Danimarca. La mia priorità era quella di trovare un posto dove sedermi, in modo da poter controllare la posta, quindi ho scelto una sessione che sembrava in qualche modo interessante e ho acceso il mio laptop.

La sessione era strutturata in modo approssimativo e ha visto la partecipazione di diversi relatori. Uno era un ragazzo barbuto di nome Martin Fowler, che ha parlato di TDD (*Test-Driven Development*) e simulazioni dinamiche. Non l'avevo mai sentito nominare e non l'ho neanche ascoltato molto attentamente, ma qualcosa deve essermi rimasto in mente. Subito dopo essere tornato in Danimarca, mi è stato affidato il compito di riscrivere da zero un grosso sistema ETL (*Extract, Transform, Load*) e ho deciso di provare il TDD (e si è rivelata un'ottima decisione). L'uso di mock dinamici era naturale, ma ha introdotto anche la necessità di gestire le dipendenze. Ho scoperto che era un problema molto difficile ma anche affascinante, e non riuscivo a smettere di pensarci.

Quello che è iniziato come un effetto collaterale del mio interesse per il TDD è diventato una passione. Ho fatto molte ricerche, ho letto molti blog sull'argomento, ho scritto io stesso parecchi post, sperimentato il codice e discusso l'argomento con chiunque volesse ascoltarmi. Sempre più spesso ho dovuto cercare ispirazione e guida fuori da Microsoft. Progressivamente, molti mi hanno sempre più associato al movimento ALT.NET, anche se non sono mai stato molto attivo. Ho commesso tutti gli errori che era possibile fare,

ma sono stato gradualmente in grado di sviluppare una comprensione coerente della Dependency Injection.

Quando Manning mi ha contattato con l'idea di un libro sulla Dependency Injection in .NET, la mia prima reazione è stata: "È davvero necessario?". Ritenevo che tutti i concetti di cui uno sviluppatore aveva bisogno per comprendere la Dependency Injection erano già descritti in numerosi blog. C'era qualcosa da aggiungere? Onestamente, pensavo che la Dependency Injection in .NET fosse un argomento già ben eviscerato.

Riflettendoci, tuttavia, mi sono reso conto che, mentre le conoscenze necessarie sono sicuramente già tutte disponibili, sono anche molto disperse e molta terminologia è conflittuale. Prima della prima edizione di questo libro, non c'erano titoli sulla Dependency Injection che tentassero di offrirne una descrizione coerente. Dopo averci ben riflettuto, mi sono reso conto che Manning mi stava offrendo una notevole sfida e una grande opportunità per raccogliere e sistematizzare tutto ciò che sapevo sulla Dependency Injection.

Il risultato è questo libro che utilizza .NET Core e il linguaggio C# per introdurre e descrivere una terminologia completa e offrire una guida alla Dependency Injection, ma spero che il valore di questo libro vada ben oltre la piattaforma. Penso che il linguaggio dei modelli qui articolato sia universale. Che siate sviluppatori .NET o utilizzate un'altra piattaforma orientata agli oggetti, spero che questo libro vi aiuti a migliorare le vostre competenze di ingegneri del software.