

Technical Debt in Software Development: A Multi-Perspective Investigation

Abstract

I prodotti software devono essere costantemente mantenuti e aggiornati per sopravvivere e continuare a soddisfare le esigenze di aziende e utenti. Agli sviluppatori viene spesso richiesto di eseguire attività di manutenzione ed evoluzione del software nel più breve tempo possibile al fine di rendere disponibili le modifiche al più presto. Di conseguenza, questi non hanno la possibilità di applicare le buone pratiche di sviluppo, introducendo così il cosiddetto debito tecnico, ovvero l'applicazione di una soluzione rapida e di bassa qualità invece di una migliore che richiederebbe più tempo. Ciò causerà una diminuzione della qualità del software e richiederà un notevole sforzo di manutenzione in futuro. Per questo motivo, per le aziende di software è di fondamentale importanza individuare in anticipo i sintomi del debito tecnico. Tuttavia, tali sintomi potrebbero manifestarsi in forme diverse e in diversi stadi di sviluppo, rendendo più difficile la loro identificazione. Nel contesto di questa tesi, affrontiamo questa sfida da diverse prospettive.

In primo luogo, ci concentriamo sui bad code smell, cattive scelte di progettazione o implementazione applicate nel codice sorgente dagli sviluppatori che possono portare a una scarsa manutenibilità e a una diminuzione della comprensibilità del codice. Negli ultimi anni, diversi ricercatori hanno ideato strumenti e tecniche per il rilevamento automatico di questi difetti di progettazione. Tuttavia, purtroppo, tutte le tecniche proposte sembrano essere ancora troppo limitate e inadeguate per essere applicate in contesti reali. La prima parte di questa tesi si concentra sulla sperimentazione dell'idoneità delle tecniche di rilevamento di code smell basate sul machine learning. I risultati preliminari dimostrano che le tecniche basate sul machine learning hanno scarse prestazioni per il rilevamento automatico dei code smell, a causa di diverse limitazioni come (i) la natura fortemente sbilanciata del problema, (ii) la soggettività dei risultati e (iii) l'insieme ridotto di metriche considerate finora. Questa tesi indaga separatamente questi tre aspetti, proponendo soluzioni specifiche per superarli. Sebbene siano stati prodotti alcuni avanzamenti, le tecniche di apprendimento automatico richiedono ancora ulteriori miglioramenti per fornire un rilevamento affidabile degli odori del codice.

Oltre a studiare il debito tecnico nel codice di produzione, consideriamo anche la sua presenza, oltre che la sua nocività, nel codice di test. Le attività di test sembrano ricevere un'attenzione molto minore durante lo sviluppo del software: i test vengono spesso sviluppati senza applicare principi di programmazione adeguati o generati automaticamente con il supporto di strumenti specifici. Pertanto, le suite di test risultanti sono spesso caratterizzate da una bassa qualità che potrebbe anche ridurre l'efficacia nella scoperta dei bug. Questa tesi affronta questa sfida presentando un'analisi su larga scala della qualità e dell'efficacia del codice di test sia nei sistemi tradizionali che nelle applicazioni mobili al fine di comprendere i fattori relativi al test che sono più legati ai problemi tecnici nel codice di produzione. I principali risultati confermano che le suite di test sono caratterizzate da una qualità ed efficacia del codice molto basse, in particolare per quanto riguarda le applicazioni mobili. Inoltre, a differenza di quanto precedentemente affermato in letteratura, alcuni degli aspetti qualitativi considerati (e.g., dimensione del codice, test smell) hanno dimostrato di avere una correlazione più forte con i difetti del codice di produzione rispetto alle metriche di copertura tradizionali ampiamente adottate.

Infine, includiamo anche una discussione sulle principali lezioni apprese e questioni aperte insieme ad alcune indicazioni su ulteriori direzioni di ricerca.