TDD 在大多數時候都是一個好的開發方法，可以達到很好的模組化，也不會產生太多多餘的程式碼，debug也方便得多。但是TDD會花費大量的時間，如果開發的時間並不充裕，那TDD可能就不是一個好點子。

而且，TDD因為都是在寫CODE前就先針對單一功能寫測試，所以如果該系統有些功能要求整體性的測試才能判斷該功能的成效時就會遇到麻煩(例:需要使用者互動的功能、需要大量資料庫的功能、需要特定的網路組態的功能)。

如果你的Project需要很謹慎完善和不易變更的Design，則TDD因為其性質的關係常常是由測試來驅動整個開發流程，有時會不那麼遵循原先的Design來進行，則在這種project中也可能造成問題。

另外，TDD假設絕大部分的CODE都是由開發團隊從頭開始寫的，如果你的Project需要使用大量Legacy code的話，則可能會讓TDD的流程比較沒辦法那麼順利地執行。

最後，TDD開發方法其實也是需要開發人員的訓練的，如果開發團隊的人沒有受過專業的TDD訓練的話，直接使用TDD來開發可能會造成許多難以預期的麻煩。