研究動機

自1940年，電腦都還未誕生之前，人類的生活中就已經有程式的存在。在發明現代電腦之後，程式的運用就更加的精進；個人電腦以及網路的普及更讓程式在社會大眾之間被廣泛運用，撰寫程式再也不是研究人員或工程師的專利。近年來隨著APP的時代來臨，許多企業與素人皆投身軟體開發，而在撰寫程式與軟體的過程中，持續整合 (Continuous integration) 就扮演了相當重要的角色。

過去，當多人共同撰寫同一程式碼時，總有這樣的情況發生：當自已好不容易在期限之內完成任務，在自己的電腦中測試功能也沒有問題，但到了會議中與同事整合後才發現，大家的程式風格各自迥異，又或者是需求的變化無法及時發現，必須重新來過。傳統的軟體開發過程總是將測試放在後頭，發現問題之時早已來不及回頭修改；即使在期限內完成修改，讓不同開發者修改後再整合的後果，就是一大段紊亂、毫無規則與品質可言的程式碼。

軟體開發本就不是件能夠獨力完成且一蹴可幾的事，在確認方向及需求後，途中往往還有更多的變數，所以不斷修改與變更的程序在軟體開發的過程中更是需要被採用，也就是在途中頻繁地停下來檢查程式碼，如此一來，軟體開發便能應對經常性的變化。所以養成隨時與定期整合的習慣，就能夠避免未來更多的修正。但往往這份工作就是交代某個工程師去解決，但這並不是件容易的事情，一個人面對數百萬行的程式碼，不但除錯是個問題，就算找到了錯誤，要找到這段程式碼的主人又是另一個問題。

於是，為了建置一個讓軟體專案能夠持續與自動化整合的環境，完成每日的測試及檢查，並讓工程師們能夠專注於開發，又能夠讓軟體能夠靈活的變動並能夠有效地做到品質控管。綜合以上繁瑣的每日例行工作，本研究想到了以持續整合工具——Jenkins來解決以上的問題。