

# 一個支援 Scrum 的工具：利用問題追蹤及持續整合系統

## A Scrum Supporting Tool: Using Issue Tracking and Continuous Integration Systems

高子騰 陳建村 鄭有進 謝金雲

國立臺北科技大學資訊工程系

Tzu-Teng Kao, Chien-Tsun Chen, Yu Chin Cheng, and Chin-Yun Hsieh

Department of Computer Science and Information Engineering

National Taipei University of Technology

Email: {t5598026, s1669021, yccheng, hsieh}@ntut.edu.tw

### 摘要

Scrum 為眾多敏捷方法中，最為人熟知的方法之一。由於 Scrum 強調面對面的溝通，所以許多專案資訊都直接記錄在白板或牆上。本論文的目的為開發一個支援 Scrum 的開放原始碼軟體，使得與 Scrum 相關的專案資訊，可被記錄於電腦中。我們結合在敏捷式方法中常用的問題追蹤及持續整合系統來達成此目的。我們使用問題追蹤系統來協助管理 Scrum 中的 Stories 與 Tasks，並利用持續整合系統的特性，每天自動收集 Stories 與 Tasks 的狀態並產生專案進度報表。此外，我們所產生的專案進度報表可與持續整合系統中各種報表(例如：測試案例通過率與程式碼行數增長報表)互相比對，以做為專案進度佐證的參考資料。

**關鍵詞：**Scrum、持續整合、問題追蹤

### 一、前言

近年來強調以人為本，以程式碼為主的敏捷方法 (agile method) [3]，已逐漸被軟體開發團隊所接受。目前較為人所熟知的敏捷方法有 Extreme Programming (XP) [1][2]、Scrum [4][6]、Feature Driven Development (FDD) [5] 等。相較於其他敏捷方法，由於 Scrum 對於專案計畫、管理、與追蹤等課題有著比較清楚的規範，且可以同時在 Scrum 中實施其他敏捷方法的最佳實務作法 (best practices)，因此越來越多的團隊使用 Scrum 來導入敏捷方法。

在 Scrum 中，其主要的三個核心價值分別為 visibility、inspection、adaptation[7]。Visibility 表示關於影響產出軟體的各種開發流程因素都必須要看的見。例如，Scrum 利用 Task board 與 Burndown Chart 來了解專案進度。Inspection 表示開發流程應該要被頻繁地檢視，例如 Daily Scrum 活動 (每天 15 分鐘的會議) 就是一個典型的 inspection 範例。Adaptation 表示當開發流程脫離正軌時，我們應該要調整流程或是改善產出物。例如，Scrum 允許我們依據開發的進度動態增加或減少在 Sprint (一個固定時間的開發週期) 中所開發的功能。由於敏捷方法強調面對面的溝通，因此在 Scrum 中許

多關於專案的文件或圖表都是直接畫在白板或是貼在牆壁上。對於專案進度有興趣的人，必須自行到專案開發現場去了解。有鑒於此，許多支援 Scrum 的軟體工具便應運而生，讓專案進度可以直接從電腦中查閱。

本論文的目的在於開發一個支援 Scrum 的開放原始碼軟體。為了避免增加 Scrum 團隊使用工具軟體的負擔，我們並不單獨開發一個獨立的 Scrum 工具。相反地，我們從敏捷式方法經常使用的輔助工具中，選擇了**持續整合**(continuous integration)及**問題追蹤**(issue tracking)這兩種工具來支援 Scrum。我們利用問題追蹤工具來管理 Stories (在 Scrum 中對於需求的稱呼) 與 Tasks (完成 Story 所需執行的工作)。此外，我們使用持續整合軟體每天自動收集目前 Stories 與 Tasks 的狀態並匯整成報表，使得開發團隊能了解專案當前最新的狀況並加以追蹤及管理。當專案進度出現異常時，也可以自動地通知相關負責人。

本文其它章節組織如下。第二節簡介其它相關支援 Scrum 開發程序的工具。第三節說明本論文提出的方法。第四節以案例說明支援 Scrum 開發程序的工具之使用。第五節對於 Burndown Chart 的分析。最後為結論與未來工作。

### 二、背景與相關研究

#### 2.1 Scrum

在 Scrum 中，定義了三種主要的角色，分別為：Product Owner、Scrum Master 及 Scrum Team。Product Owner 為 Stakeholders 的代表，收集及了解 Stakeholder 對產品的需求，並確認專案團隊是否完成正確的需求。Scrum Master 為負責指引專案團隊，能正確地執行 Scrum 開發的程序。Scrum Team 即為軟體專案開發的團隊，負責軟體功能的開發及測試。

如圖 1 所示，Stakeholders 對於要開發的產品擁有一個願景，並對於這樣的願景而產生出對於產品的需求。然後 Product Owner 會收集這些的需求來描寫成 Stories，並將這些 Stories 新增至 Product Backlog(一個有優先順序的 Stories 列表)。而 Product Owner 依 Stakeholder 所期望完成的優先順序來定訂 Story 的重要性。

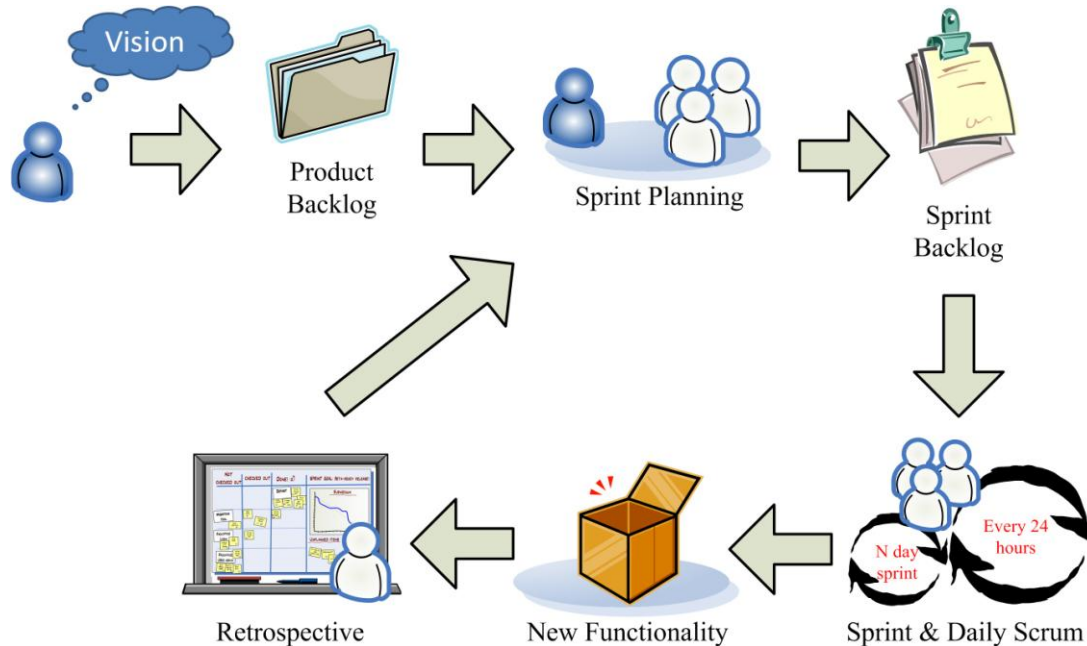


圖 1 Scrum 開發流程[15]

當每次 Sprint 開始前，Product Owner、Scrum Master 及 Scrum Team 會一起參與 Sprint Planning Meeting。在會議中，Product Owner 向 Scrum Team 說明在 Product Backlog 中的 Stories，讓 Scrum Team 能透過這樣的介紹來了解 Stakeholders 需要的是什麼。之後，Scrum Team 利用 Story Point 的方式，來預估 Story 的難易度。而 Story Point 衡量的方式，常見的有利用人-工天或人-工時來做為衡量基準。例如利用人-工天的方式，預估這個 Story 需要花費 4 點的 Story Point 才能完成，則代表預估一個人需花費 4 天來完成這個 Story，或者是兩個人需花費 2 天來完成。經預估過後，Scrum Team 便開始描述預期將如何展示或呈現出 Stories 完成的功能。此時 Product Owner 若發現在描述中與原先的所期望的 Story 不符合時，便能立即修正 Scrum Team 對於 Stories 的理解，以防止最後完成的功能不是 Stakeholders 想要的。接著，Scrum Master 根據先前的記錄來計算出 Estimated Velocity(預估在該 Sprint 能完成的 Story Point 的數量)，做為在這 Sprint 中實際能投入時間上限的基準。然後以不超上限為原則，依重要性高的順序來將的 Stories 加入至這次的 Sprint，並記錄於 Sprint Backlog(一個將在該 Sprint 中要完成的 Stories 及 Tasks 列表)。

當 Sprint 開始時，在每天的 Scrum Team 都會進行簡短的 Daily Scrum。在會議中，Scrum Team 的每個成員各自描述在昨天完成了什麼，今天要做什麼，及是否有發生什麼問題需要大家幫忙解決的。若有 Story 偏離先前預估的時間時，則在會議的討論後，Scrum Team 的成員各自在 Task Board 上，對該 Story 的預估時間加以修正。Scrum Master 便藉由 Task Board 上的 Burndown Chart 來了解 Stories 完成的狀態，以便於管理及追蹤在這 Sprint 中目前的進度。所以當有圖上呈現出問題的徵兆時，能夠即早修正。

當 Sprint 結束時，Scrum Team 進行 Sprint Review。在這會議中，Scrum Team 將在 Sprint 中所完成的功能展示給 Product Owner 看，Product Owner 便能藉此了解功能是否符合 Stakeholders 所期望的需求，因此能立即回覆意見給 Scrum Team 來加以改善。

最後，對於這個結束的 Sprint 進行 Sprint Retrospective，來回顧在這個 Sprint 中進行上所產生的問題、時間預估錯誤等。使得在下一次的 Sprint 的進行時，能夠做得更好。重覆從 Sprint Planning Meeting 到 Sprint Retrospective 這樣的流程，直到將 Stakeholders 所期望的產品完成為止。

## 2.2 相關研究

VersionOne[16]是一個支援敏捷式專案管理的商業軟體，其提供 Scrum、XP、DSDM[10]等樣板來支援各種敏捷式的開發程序。其本身並無支援其他敏捷方法的最佳實務作法，例如：問題追蹤或迴歸測試等。所以其提供整合其它外部工具的介面來支援敏捷方法的實務作法。像是在問題追蹤上，提供與 JIRA[11]及 Bugzilla[9]的整合；在迴歸測試上，提供與 Quicktest Pro[13]的整合。但 VersionOne 為商業軟體且對於平台的相依性過大，僅能安裝及使用在 Windows 系列的平台上。

Microsoft Visual Studio Team System (VSTS)[17]是一個支援團隊開發的輔助系統。主要針對支援 Agile 與 CMMI 這兩種開發方式，也能夠藉由下載並匯入 Scrum 的範本來支援 Scrum。對於其他敏捷方法的最佳實務作法，如迴歸測試、持續整合及問題追蹤而言，都有能力來支援。但 VSTS 如同 VersionOne 一樣同為商業軟體且對於平台的相依性過大，僅能安裝及使用在 Windows 系列的平台上。相較於 VersionOne 及 VSTS，我們的工具為開放軟體且由於採用 Java 技術開發，因此可以

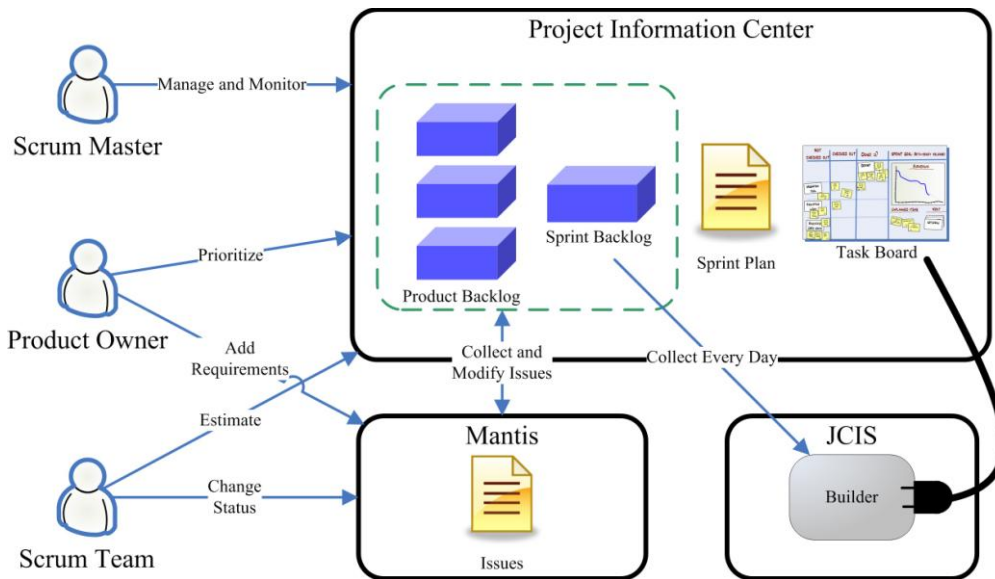


圖 2 系統環境圖

跨平台執行。

ScrumWorks[14]是一個支援 Scrum 開發程序的專案管理的工具。其有提供兩種版本，分別為 Basic 及 Pro，其中 Basic 的版本是完全免費的。Basic 版僅提供對於 Scrum 開發程序的基本管理功能；而 Pro 版有提供對於 JIRA 及 Bugzilla 這兩種問題追蹤系統做整合。但對於迴歸測試及持續整合的部份並無提供與其它軟體的整合。而 ScrumWorks 雖為桌面應用程式，但有提供可在不同平台上執行的版本，讓使用者可依現有的環境來挑選適當的版本。相較於 ScrumWorks，我們的工具因結合持續整合工具，能夠自動執行迴歸測試及整合建置並產生報表。使用者能藉由報表與專案進度的互相參照，以更加清楚地了解專案的現狀。

### 三、解決方法

為了要實現結合問題追蹤及持續整合以支援 Scrum 開發程序的工具，因此我們從問題追蹤及持續整合工具中，來挑選出合適的工具並加以整合。

所以對於問題追蹤工具而言，因現今的問題追蹤系統種類繁多，如 Mantis[12]、BugZilla、JIRA 等。但因為 Mantis 較於其它問題追蹤系統而言，擁有容易安裝、容易使用、免費、開放原始碼、支援架設於各種平台等特性，且做為問題追蹤的工具而言，其功能足以支援在 Scrum 中工作項目的管理。所以我們經上述的考量後，選擇使用 Mantis 做為整合問題追蹤工具的對象。

所以對於持續整合工具而言，我們擴充先前提出的持續整合工具—JCIS[8]。JCIS 其擁有免費、開放原始碼、容易擴充、支援異質平台的整合建置、支援架設於各種平台等特性。因此我們繼續擴充其功能以成為能夠支援 Scrum 開發程序的持續整合工具。

如圖 2 所示，本工具主要分為三個部份所構

成，分別為：Project Information Center(PIC)、Mantis 及 JCIS。PIC 為本工具主要呈現的介面，提供對於 Mantis 上 Issues 的操作，以及做為管理及提供 JCIS 收集資訊來源。Mantis 做為管理 Stories 及 Tasks 的儲存庫(Repository)，將 Stories 及 Tasks 視為在 Mantis 上的 Issues 來進行管理。JCIS 會根據 PIC 中的資訊，自動收集 Stories 及 Tasks 現狀的資訊，並定期地將資訊匯整成報表。

在先前 2.1 節中所描述的，在 Scrum 中參與的角色為下列三種：Product Owner、Scrum Master 及 Scrum Team。首先 Product Owner 會收集 Stakeholder 的需求，並在 Mantis 上將需求新增成 Issues。PIC 會從 Mantis 中收集該專案上的 Issues，並匯整成 Product Backlog。所以 Product Owner 可以藉由 Product Backlog 的呈現，來取得目前 Stories 的列表，並對每個 Story 設定其重要性。

在 Sprint Planning Meeting 中，Scrum Team 經討論後，藉由從 PIC 的 Product Backlog 中，來設定對 Stories 所預估完成要花費的時間。並對於幾個重要性較要高的 Stories，來描寫在完成時要如何展示。Scrum Master 在 PIC 上的 Sprint Plan 來設定這個 Sprint 的目標、開始日期、長度、參與人數及能投入專案的專注率(Focus Factor)。Scrum Master 以不超過藉由 PIC 所計算出的 Estimated Velocity 為原則，從 Product Backlog 中來挑選重要性高的 Stories 進 Sprint Backlog。

在 Sprint 的期間中，Scrum Team 的成員藉由在 Mantis 上將 Story 所對應 Issue 的狀態改變成「Assigned」或「Closed」來呈現出這個 Story 已領取或已被完成。

而對於 JCIS 而言，延伸其 Builder 來新增一個 Task Board Builder。藉由挑選該 Builder，使得當 JCIS 在執行持續整合時，能夠自動收集目前



Sprint Backlog 中 Stories 的狀態，並在產生出 Task Board Builder 的 Report 後將其發佈至 PIC。因此 Scrum Master 在 Daily Scrum Meeting 中，能夠藉由從 PIC 中最新的 Task Board Report 來了解目前 Sprint 的進度。也能在進行 Sprint Retrospective 時，指定前某日的 Task Board Report，來了解當時 Sprint 進行狀況，以協助日後改善流程的依據。

#### 四、使用範例

本節以一個小型專案為例，來展示工具的使用。在團隊中，包含 Product Owner、Scrum Master 及 Scrum Team 三種角色。

首先 Product Owner 在提出對產品的需求中，有一項為「管理使用者帳號」，其中包含新增、刪除、修改及查詢。所以將該需求分割成三個 Stories，分別為「新增/刪除使用者帳號」、「修改使用者帳號」及「查詢使用者帳號」三種。並分別在 Mantis 中新增這些 Stories 並且填寫目前必要的欄位。然後在 Product Backlog 中依順設定其重要性為 50、30、70，如圖 3 所示。

接著 Scrum Master 在 Sprint 開始前，利用 Sprint Plan 功能來設定這個 Sprint 的必要資訊，例如：長度為 4 個星期、參與人數為 3 人、開始日期為 2008/4/7。然後扣除與專案無關的大小事項後，預估能夠投入在專案的比率大約為七成，因此在這次 Sprint 的專注率(Focus Factor)中填入 70%，如圖 4 所示。

在 Sprint Planning Meeting 中，Scrum Team 的成員預估各 Stories 所需花費的時間，之後並利用簡單的句子描述各 Stories 在完成時要如何展示，例如：「新增/刪除使用者帳號」便會描寫說：「打開管理頁面，新增使用者，DB 中建立使用者，顯示新增成功；打開管理頁面，刪除使用者，DB 刪除使用者，顯示刪除成功」。然後工具根據 Sprint Plan 中所設定的資訊，計算出來 Estimated Velocity 為 44 個 Story Point，以做為這次 Sprint 的上限。接著 Scrum Master 依據重要性的優先順序來將「新增/刪除使用者帳號」及「查詢使用者帳號」加入至 Sprint Backlog，如圖 5 及 6 所示。

Scrum Master 可以藉由每日自動產生的 Task Board Builder Report 來了解團隊的工作效率是否穩定，如圖 7 所示。

在第三天的 Daily Scrum Meeting 中，發現對「查詢使用者帳號」這個 Story 而言，先前過於低估其複雜度，所以預估仍需多花費 1 個 Story Point 才可完成。因此在 Sprint Backlog 中，將「查詢使用者帳號」這個 Story 的預估值從 2 改變成 3。

當「查詢使用者帳號」這個 Story 開始處理時，Scrum Team 的成員便到 Mantis 將 Story 的狀態改變成「Assigned」。之後完成該 Story 且經測試驗證過後，便將 Story 的狀態改變成「Closed」。

在 Sprint 結束之後，進行 Sprint Retrospective 時，可檢視過去 Task Board Report 並與其它 JCIS

上的報表做對照，來做為改善下次 Sprint 的參考。

#### 五、Burndown Chart 的情境分析

在 Sprint 的期間中，Scrum Master 可以藉由在 Task Board 上所呈現的 Burndown Chart，從圖上

Product Backlog  
Sprint #1 Now.

ID	Name	Importance	Estimation	How To Demo	Notes	Execute
102	查詢使用者帳號	70	2	打開管理頁面，新增使用者，DB中建立使用者，顯示新增成功；打開管理頁面刪除使用者，DB刪除使用者，顯示刪除成功		
100	新增/刪除使用者帳號	50	3	新增/刪除使用者帳號...		
101	修改使用者帳號	30	1	修改使用者帳號...		

圖 3 Product Backlog

Sprint Plans

Sprint ID	Sprint Goal	Start Date	Interval	Members	Focus Factor	Action
1	完成管理功能	2008/04/07	4 week(s)	3 person(s)	70 %	

圖 4 Sprint Plan

Select Stories  
Select Stories into Sprint #1 Story Point : 0 / 44

ID	Name	Importance	Estimation	How To Demo	Notes
<input checked="" type="checkbox"/>	新增/刪除使用者帳號	50	3	新增/刪除使用者帳號...	
<input checked="" type="checkbox"/>	修改使用者帳號	30	1	修改使用者帳號...	

圖 5 挑選在這個 sprint 中要做的 Story

Sprint Backlog #1 - 完成管理功能  
Sprint #1 Now. Duration : 2008/04/07 ~ 2008/05/04 Story Point : 5 / 44

ID	Name	Importance	Estimation	How To Demo	Notes	Action
102	查詢使用者帳號	70	2	打開管理頁面，新增使用者，DB中建立使用者，顯示新增成功；打開管理頁面刪除使用者，DB刪除使用者，顯示刪除成功		
100	新增/刪除使用者帳號	50	3	新增/刪除使用者帳號...		

圖 6 Sprint Backlog

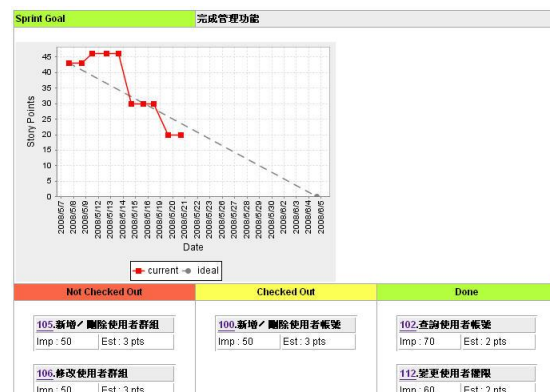


圖 7 查閱 Task Board

預估剩餘所需的 Story Point(實線)及理想剩餘的 Story Point(虛線)兩者之間的關係，來了解目前 Sprint 進行的進度並加以判斷是否如預期的進行，如圖 8。因此，我們根據其呈現出不同的結果，並歸納出四種基本可能的情境。藉由這些可能的情境，以協助 Scrum Master 判斷出問題可能的發生點。

(1) 實線無明顯偏離虛線：如圖 8 所示。Scrum Master 便可預期在 Sprint 中專案的進度符合預期，能夠如期完成。Scrum Master 除了 Burndown 圖外，可以從 JCIS 中 Compile 及 Test 的報表裡看到其建置的項目皆為通過的，並可以從 Test 報表中的看到測試的數量增加；也能從 StatSVN 的報表中看到程式碼行數在這期間內的持續地上昇。因此藉由以上報表與 Burndown Chart 之間的對照，以佐證目前專案的進度是符合預期的。

(2) 實線向上偏離虛線：如圖 9 所示。因此 Scrum Master 便可預期進度上將會延誤，而無法如期完成。以下三者為造成延誤的可能原因：

A. 過於低估先前的 Story，經重新預估該 Story 之後，而調高原先 Story Point 的值。Scrum Master 可以藉由查閱 Story 的歷史記錄，發現 Story 所預估的 Story Point 高於原先預估的值，並且是在 Sprint 的期間中所修改的。

B. 先前完成的 Story 可能因為發生整合失敗、測試失敗或者是對有相依該 Story 的開發人員有發生問題。因此將完成的 Story 重新開啓(Reopen)來處理該問題。Scrum Master 可以藉由查閱 Story 在 Mantis 上的狀態是否由「關閉」而改變成其它的狀態。而從改變狀態的日期，來查閱當日 Compile 或 Test 的建置報表，是否建置失敗發生而造成的。或者是由其它相依該 Story 的開發人員，發現該 Story 的重大問題並回報至該 Story 中。

C. 僅在 Scrum Master 為了要修正向下偏移紅線的前提之下(參照(4)的情境)，而增加在 Sprint 中的 Story，來避免人力資源的浪費。所以 Scrum Master 可以參考 Story 的歷史記錄，發現紅線突然上升的當天中，剛好有 Story 加入至該 Sprint 中。

(3) 實線維持水平並向右偏離虛線：如圖 10 所示。因此如同前一個情節相似，便可預期進度上將會延誤。以下三者為造成延誤的可能原因：

A. 遇到難度較高的 Story，而使得紅線無法沿著灰色虛線一樣立即下降。但會維持水平直到 Story 完成後，而大幅度的

下降。Scrum Master 可藉由查閱該 Story 中其預估的時間是否是相當的大，才會造成無法立即完成而造成明顯的偏離。

B. 團隊因忙於其它的事情，而沒有投入參與專案的時間，使得專案處於停滯的狀態。可藉由從 Compile 及 Test 報表的結果中，發現與先前幾次的結果相同且不變。也可從對照 Task Board 上 Story 的狀態與先前的狀態，發現仍為相同。

C. 開發品質過低，測試一直失敗。雖然開發人員完成了 Story，但因無法通過測試的驗證，而造成無法將 Story 關閉。所以 Scrum Master 能從測試的報表中發現測試一直失敗，並且在 StatSVN 的報表中發現程式碼行數並無明顯成長。

(4) 實線向下偏離虛線：如圖 11 所示。因此 Scrum

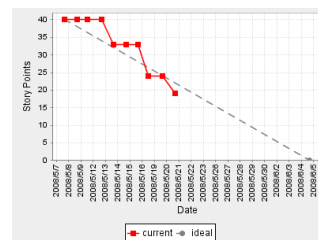


圖 8 實線沿著虛線

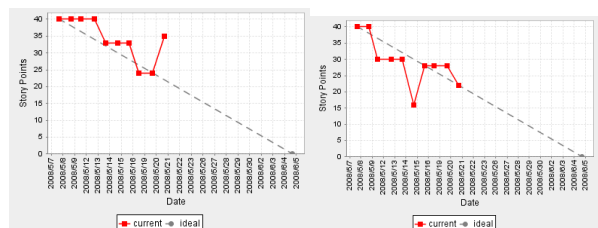


圖 9 實線向上偏離虛線及實線先向下偏離後再向上修正

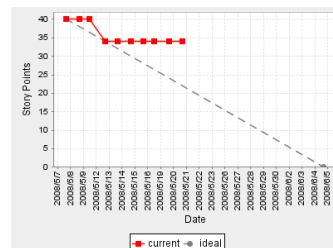


圖 10 實線向右(水平)偏離虛線

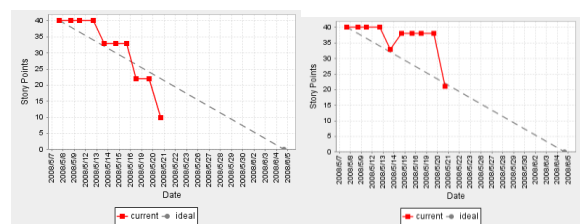


圖 11 實線向下偏離虛線及實線先向上偏離後再向下修正

Master 便可預期進度上將會提早完成，而使得在剩餘時間中的人力資源浪費。以下兩者為造成提前的可能原因：

- A. 過於高估先前的 Story，而造成 Story 在少於預估的時間內提早完成。在 Compile 及 Test 的報表中，都是呈現成功的結果，並可以藉由查閱 Story 歷史狀態的資料，了解 Story 從「Assigned」到「Closed」的期間是否小於預估的時間，而做為 Story 能提前完成的參考依據。
- B. 僅在 Scrum Master 為了要修正向上或向右偏移紅線的前提之下(參照(2)(3)的情境)，而將 Sprint 中不重要且無法良好呈現的 Story 移出，避免主要需完成的 Story 因此被延誤而導致無法完成。所以 Scrum Master 可以參考 Story 的歷史記錄，發現紅線突然下降的當天，從該 Sprint 中剛好有 Story 被移出。

藉由上述分析的情境中可以了解到，除了能從 Burndown Chart 中來了解專案進度的現狀之外，我們還能夠藉由從整合建置中各種的報表與 Burndown Chart 之間互相的對照，使得問題的範圍能夠縮小，並且迅速地找出問題發生的來源。

## 六、結論與未來展望

本論文介紹一個結合問題追蹤及持續整合以支援 Scrum 的開放原始碼工具，並對於 Burndown Chart 中各種造成專案進度落後與超前的可能性加以分析與說明。藉由我們的分析以及參考持續整合系統所提供的佐證資料，我們期望當專案進度發生異常時，專案團隊成員能夠迅速找出問題可能發生的來源並加以解決。

目前我們的系統只支援對 Sprint 的規劃，未來將提供規劃專案發佈計劃(Release Plan)的功能，並考慮支援其它持續整合及問題追蹤系統。

## 參考文獻

- [1] S.W. Ambler and R. Jeffries, *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, Wiley, 2002.
- [2] K. Beck, *Extreme Programming Explained*, Addison-Wesley, 1999
- [3] A. Cockburn, *Agile Software Development*, Addison-Wesley, 2002
- [4] H. Kniberg, *Scrum and XP from the Trenches*, InfoQ, 2007
- [5] S. Palmer and J. Felsing, *a Practical Guide to Feature Driven Development*, Prentice-Hall, 2002
- [6] K. Schwaber, *SCRUM Development Process*, <http://www.controlchaos.com/>
- [7] K. Schwaber, *Agile Project Management with Scrum*, Microsoft press, 2004.
- [8] 吳家豪, “支援JAVA應用程式發展的持續整

合系統-JCIS”, 國立台北科技大學資訊工程系碩士班碩士學位論文, June 2006.

- [9] Bugzilla, <http://www.bugzilla.org>
- [10] DSDM, <http://www.dsdm.org/>
- [11] JIRA, <http://www.atlassian.com/software/jira/>
- [12] Mantis, <http://www.mantisbt.org>
- [13] Quicktest Pro, [https://h10078.www1.hp.com/cda/hpms/display/main/hpms\\_content.jsp?zn=bto&cp=1-11-127-24%5E1352\\_4000\\_100\\_\\_](https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-127-24%5E1352_4000_100__)
- [14] ScrumWorks, <http://www.danube.com/scrumworks>
- [15] SPRiNT It, <http://www.Sprint-it.com/>
- [16] VersionOne, <http://www.versionone.com/>
- [17] VSTS, <http://msdn2.microsoft.com/en-us/teamssystem/default.aspx>