

使用既有 ezKanban 領域模型實作 ezScrum

專題編號：114-1-CSIE-S005

執行期限：113 年第 1 學期至 114 年第 1 學期

指導教授：鄭有進

專題參與人員：111820020 胡瑀涵

111820029 黃曉筠

一、摘要

在領域驅動設計中，建立領域模型是一項知識與互動密集的活動。若能將已成功建立的模型應用於其他具相似問題領域的系統，將有助於重複使用現有知識與結構，進而提升開發效率。

本專題嘗試重用已開發超過六年的看板系統 ezKanban 的領域模型，以建構一個新的 Scrum 系統——ezScrum。具體而言，我們首先透過事件風暴技術釐清 Scrum 問題領域中的核心需求，並以 ezKanban 領域模型模擬之。接著在使用案例層，透過操作 ezKanban 的領域模型來實作 ezScrum 系統。

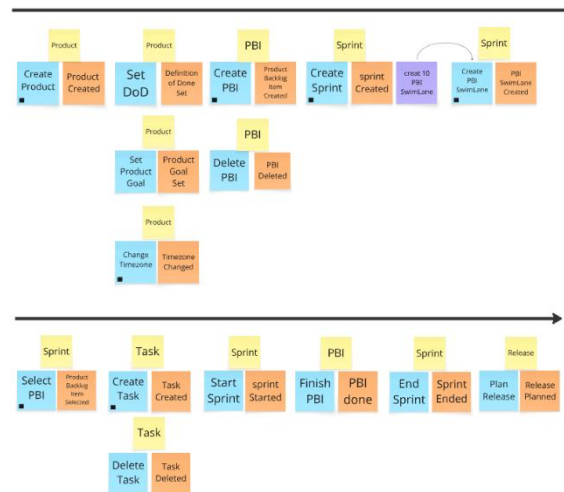
二、緣由與目的

北科資工系軟體系統實驗室所開發的 ezKanban，套用領域驅動設計、Clean Architecture、CQRS、Event Sourcing 等技術開發，其領域模型經過多年驗證已十分穩定。由於看板視覺化工作流程的特性，可以用來表達 Scrum 工作流程，因此相較於修改舊有的 ezScrum，或是從頭開發新的 ezScrum，我們選擇重用既有的 ezKanban 領域模型來實作全新的 ezScrum。

藉由此專題，我們希望探討在重複使用領域模型過程中所遭遇的挑戰及其解決策略，並分析此方法於相近領域系統間的適用性與限制。

三、重用領域模型策略分析

我們採用事件風暴分析 Scrum 業務流程，並從中找出聚合 (Aggregate)，如圖一所示。



圖一、Scrum 事件風暴分析

圖二為 ezScrum 與 ezKanban 的領域模型對照，黃色便利貼表示 ezScrum 的聚合，粉紅色便利貼表示 ezKanban 的聚合。以下說明兩者領域模型的對照關係：

(一) Scrum Team：負責交付產品的團隊，重用 Team。

(二) Product：團隊正在開發的產品，重用 Board。

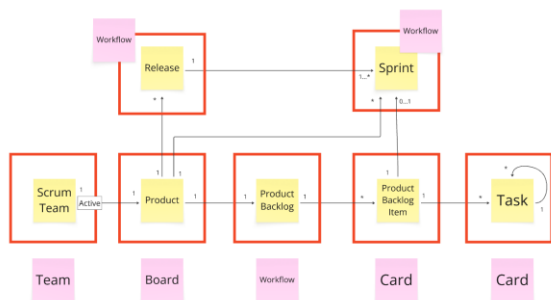
(三) Product Backlog：產品待辦清單，重用 Workflow。

(四) Product Backlog Item：Product Backlog 中的一條項目，簡稱 PBI，重用 Card。

(五) Task：PBI 被分解後的具體執行工作，重用 Card。

(六) Release：產品釋出計畫，包含若干個 Sprint 的開發成果，重用 Workflow。

(七) Sprint：固定開發週期，通常介於一至四週，重用 Workflow。



圖二、ezScrum 與 ezKanban 的領域模型對照

四、業務邏輯對照

在確定 ezScrum 與 ezKanban 的聚合的對應關係後，我們在使用案例層實作 Scrum 的業務邏輯。例如在 CreateProductUseCase 中，先創建了一個 Board，並在其中創建兩個 Workflow 分別代表 Product Backlog 及 Task Backlog。其餘使用案例皆採用上述類似方式實作。

五、實驗結果

此系統後端採用 Java 搭配 SpringBoot 框架；前端使用 React.js；資料庫則支援 PostgreSQL (Message Store) 和 EventStoreDB。

目前已完成領域模型設計和事件風暴流程，並實作完所有使用案例，從 Sprint Planning、Daily Scrum、Retrospective 等完整 Scrum 使用流程皆有概括。另外，我們也探討了重用領域模型之優點與限制：

(一) 優點

重複使用核心業務邏輯層的物件不但可降低錯誤的發生機率，還能省去撰寫測試的繁瑣流程；此外，除了核心業務邏輯層，不少使用案例也可進行重用。

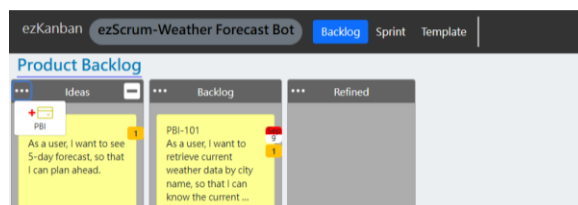
(二) 限制

跨交易邊界的使用案例：原本 ezKanban 使用案例操作單一聚合，使用案例的交易邊界與聚合一致。ezScrum 有些使用案例重複呼叫多個 ezKanban 使用案例，導致兩者交易邊界不一致，會造成以下問題。

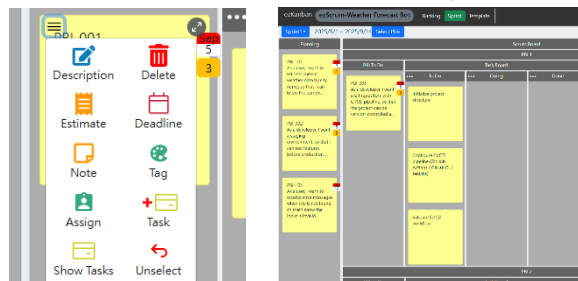
1. 效能問題：若一次大量使用原有之使用案例，會產生巨量業務 (Transaction) 操作資料庫，並造

成系統卡頓，進而影響使用者的操作體驗。

2. 資料一致性問題：若部分使用案例失敗會導致系統狀態不一致。需額外撰寫補償機制。
3. 在使用案例層增加額外的業務邏輯：業務邏輯應該在實體層實作，但現在卻在使用案例層。



圖三、產品待辦列表 (Backlog 頁面)



圖四 (左)、新增和查看所有 Task

圖五 (右)、Sprint 整體頁面

六、結論

經過這次的實作經驗，我們了解到，在決定重用既有模型之前，應先分析兩系統間之應用領域相似性，若需求差異過大或模型重疊度不高，則無法發揮重用的優勢；此外，若新系統高度依賴事件驅動且實作新的領域事件時會有困難時，在使用案例層就會需要透過更多複雜的業務邏輯來達成功能。

而本次實作的 ezScrum 在架構及領域模型上皆和 ezKanban 有很高的相似之處，雖然有些業務邏輯需要在使用案例層實作，但其成本並不高於重用領域模型所節省之成本；加上本實驗室為行為驅動開發，重用領域模型節省了大量的測試撰寫時間。因此，我們認為此次重用模型為我們帶來的利大於弊，為一有效率之重用。

有鑒於近年來 AI Coding 興起，我們希望未來能比較本方法與讓 AI 重新撰寫特定領域模型兩者之間的優缺點。