1. Summary and reflection of Successful Software Management Style: Steering and Balance

Metrics and measures for software products 沒有一個明確的度量單位，而且是非常的主觀，然而作者認為 software management 跟 movie producers 的製片過程''部分''有異曲同工之妙，在文章結尾作者希望能用不同的角度來看專案管理的管理技術，並且認為專案管理對專案經理的重要性，並對此提出了佐證。一開始先是對 iterative approach 進行解釋，認為傳統的 project management approaches 在 problem space, solution space and planning space 去做調整會帶來許多的不確定性而不鼓勵去做協調，而現代的 iterative approach 通稱為 iterative development methods，在計畫和預算內成功交付 software product，並且產品需要持續的發展、生產、評估等等，以 IBM Rational Unified Process 為例，分為四個階段 Inception, Elaboration, Construction 和 Transition，每個階段代表 project 的狀態而非一個連續的過程，如同電影行業拍一部電影一樣 ，將許多人的貢獻綜合成一個有凝聚力的綜合知識產權，例如:programmer 在 building a model, sketching a flowchart 等等時，本身就已經知道是以一個猜測且合成的抽象解決方法，如同電影製作人以同樣的方式查看腳本、故事、設置模型和服裝設計，承諾拍攝場景，使演示文稿足夠有形，以判斷其整體綜合效果。再來是 Precision 和 accuracy 的比較，在 software management 裡，precision 和 accuracy 並不像看起來沒什麼差別，反之 software management 充滿灰色地帶、situation dependencies 和 ambiguous trade-offs，因此 software manager 更需要精準預測和評估風險的影響。第三是 Four patterns for successful steering，應為在 iterative approach 的過程中，需要動態的控制和設置中間的檢查點幫助 programer 更好管理，所以設置 four pattern 助於創建檢查點，分別是 Scope management、Process rigor、Progress honesty 和 Quality control，然而大多數的 project management 都不太能接受，與傳統的概念不符且說起來容易做起來難，最重要的是管理一個 project 是需要一定的領導能力，因此我們更應該珍惜具有一定

領導能力的 project manager 如同電影製片人一樣不僅可以創造出好的產品，還可以作為缺乏經驗的團隊成員的導師。接下來分別會對四個 pattern 作分析。Scope management 在傳統的情形中主要以投資者或得利益者接受完成的 project，反之現代可以看到許多協商和變更的要求，而這些通常會影響利益相關者之間的合同，目前有兩種主要的用戶規範，第一個是 vision statement 開發人員應以用戶可以理解的格式讓買家或用戶理解，第二種是 evaluation criteria 是源自於第一種和不同的資訊而組成的，對早期提供較好的 framework。Process rigor 由於在 agile camps 和 processmaturity camps 都沒有對於解決方案有一個明確的規範，然而大多數的 project management 都同意在 teams are distributed、project is large、stakeholders are involved 等等能有 process rigor 幫忙，然而不適當的強調 process rigor 可能會導致 project 的失敗，因此 project managers 必須把領導的重心從管理團隊開始到整個生命週期以便平衡各個團隊。Progress honesty 在程式發展的過程中，由於利益相關者關係常導致互相不信任，一個好的專案應該展示誠實進步和進展，在專案初期 Stakeholders 不應該對初期錯誤、離題或不成熟的設計反應過度，而開發組必須對連續增量的實際改進負責，並證明其有效改進，這樣才不易導致雙方的不信任。Quality control 大多數的測試在整個生命週期都會出現，在早期測試有助於在生命週期的早期解決架構上重要的問題，此外還提供了一個不斷發展的測試平台，用於持續評估系統的進度和性能，為了鼓勵在生命週期的早期進行集成測試，會用 iteration 來進行測試，利用 inception iterations、elaboration iterations、construction iterations 和 transition iterations 有助於產品展示給使用者。最後則是 The economic benefits of a leadership style，傳統的管理專案在整合和測試活動中佔用總資源的大約 40%，其中大部分工作都是在過多的 scrap and rework 中消耗的，然而在 iterative process and steering leadership style 的現代專案可以讓產品減少消耗約 25%的預算。

2. Please explain What is software project management? What is its significance among the whole software development?

從一個 project 的初期(規劃、計畫等等)到產品的完成與後續品質控管與維護，project 的整個生命週期裡，人事管理、進度追蹤、與使用者溝通、中途的產品維護等等都屬於 software project management，在整個 software development 的過程中，好的管理可以在初期擬訂良好的計畫，有利於後續產品的開發，在產品開發的過程中，良好的與使用者溝通，不易出現與使用者不符的產品，也可以有效的監督專案的進度和品質，在問題發生時，也可以快速的進入狀況，並適當的解決，在專案的結尾也能夠有效的掌握產品的品質和控管，最後一個良好且優秀的管理，甚至可以節省原先所預期的預算。

3.According to your own developing experience, answer the questions corresponding to your choice. Have you ever bumped into the challenges mentioned in the essay? Have you ever used the methods mentioned in the essay?

在文章裡我最常遇見的是人事管理和程式整合的部分，由於每個人都有不同的事情，大家也不是都待在一起工作，所以有時要開會時，不是每個人都會到齊，導致有些人的進度根本無從而知，也無法確認進度是否有如期進行，有時可能要等到期限到了才會看到人，即便每次都可以如期開會，在做整合時，時常會遇到非預期內的問題，例如:每個人的 code 在繳交時，都認為已經沒有 bug，也經過很多的測試，但當整合在一起時有時就會出現在測試時沒注意到的 bug，或是整合時才會出現的 bug，這時就要再花大量時間和人力去解決，當時並沒有使用文章裡的方法，也許下次可以試著用文章裡的方式去時做一個專案，雖然多少都會有問題，但也許會比漫無目標的管理有更好的效果。

4.What suggestions would you give for Taiwanese software industry after reading the essay? Imagine you are leading a software development team. What would you pay

most attention to?

也許可以嘗試用不同的管理方式，去看看哪種管理方式更適合現在的團隊，雖然可能需要一段時間的磨合，但時間長，我認為可能會帶出比想像中更好的效益，若我能夠領導一個軟體專案團隊，我會希望能夠在團隊裡的溝通和磨合會更多加注意，應為發展一個專案，團員之間的溝通我認為相當的重要，良好的溝通會有助於減少整合時出現的問題，而且團隊裡的氣氛也相當的重要，我認為良好的氣氛更帶動整個團隊的進步。