

Must-know Knowledge of “Software Engineering and Management”

By Prof. S.J. Huang on November 15, 2023

Department of Information Management, NTUST (Taiwan Tech)

Fundamental to SEM

1. List at least five principles of seven Engineering Principles. 列出七項工程原則中的至少五項原則。

原則一：建立標準作業程序

原則二：分解

原則三：模組化

原則四：遞增法

原則五：預視改變

原則六：焦點分離

原則七：通用性

2. What are the four natures/essences for a software development project. 軟體開發專案的四種性質/本質是什麼。

從「問題」到「答案」 從「程式」到「系統」

從「人腦」到「電腦」 從「個人」到「團隊」

3. How to define a good quality application software? 如何定義一個優質的應用軟體？

最基本的是保障軟體的**功能需完整、正確**，在設計上確保軟體性能優越不卡頓，並且使用體驗順暢，符合直觀好上手的易用性，做好安全性保護措施，確保隱私不外洩、同時保障軟體的穩定性，最後需進行軟體定期維護、持續更新，保障用戶體驗及版本相容性等等，才可稱之為優異的應用軟體。

Functional requirement / 功能性需求

Accuracy / 正確

Completeness / 完整

Etc.

Non-Functional Requirements / 非功能性需求

Performance / 效能

Ease of Use / 易用性

Reliability / 可靠性

Maintainability / 維護性

Security / 安全性

Portability / 可移植性

4. What is the theory of triple constraint in project management?

專案管理中的三重約束理論是什麼？

成本（符合預算（如本）、價廉）

時間（準時交付（如期）、交付或上市時間）

範疇（定義專案的範疇、確保專案的完成度）

品質（優異品質（如質）、物美）



5. What is scope creep and ripple effect in software engineering? 什麼是軟體工程中的範圍蠕變和漣漪效應？

scope creep：增加未經授權的新產品、要求或工作的附加特性或功能（即超出商定的範圍）。

ripple effect：連鎖反應指標顯示了軟體變更將對系統的其餘部分產生什麼影響，連鎖反應的計算是基於單一變數的變更對程式其餘部分的影響；可稱之為程序複雜性的衡量標準。

6. List at least 6 knowledge areas (KA) in “Guide to SWEBOK”. 在“SWEBOK 指南”中列出至少 6 個知識領域（KA）。

1. **Software Requirement**軟體要求

2. **Software Design**軟體設計

3. **Software Construction**軟體構建

4. **Software Testing**軟體測試

5. **Software Maintenance**軟體維護

6. Software Configuration Management 軟體配置管理

7. Software Engineering Management 軟體工程管理

8. Software Engineering Process 軟體工程流程

9. Software Engineering Models and Methods 軟體工程模型和方法

10. **Software Quality**軟體品質

11. Software Engineering Professional Practices 軟體工程專業實踐

12. Software Engineering Economic 軟體工程經濟

13. Computing Foundations 計算基礎

14. Mathematical Foundations 數學基礎

15. Engineering Foundations 工程基礎

Software Process

7. List three generic software development process models? 列出三種通用的軟體開發流程模式？

瀑布模型 Waterfall mode

▪ 屬計畫導向，規格的獨立且不同的階段和發展。

增量開發 Incremental model

▪ 屬計畫導向或敏捷開發，系統的規格、開發和驗證是相互交織的。

整合與配置 Integration and configuration

▪ 屬計劃導向或敏捷開發，系統由現有的可設定組件組裝而成。

8. What are four fundamental activities that are common to all software processes?

所有軟體流程共有的四項基本活動是什麼？

規範／規格—定義系統應該做什麼

設計與實施—定義系統的組織並實施系統

驗證—是否符合客戶的要求

演進—持續改變系統以回應不斷變化的客戶需求

9. List the common components in the definition of process description. 列出流程描述定義中的公共元件。

工作/最終產品：過程活動的最終結果。

角色：反映參與流程的人員的職責。

前置條件和後置條件：在流程活動頒布或產品生產之前和之後的狀態。

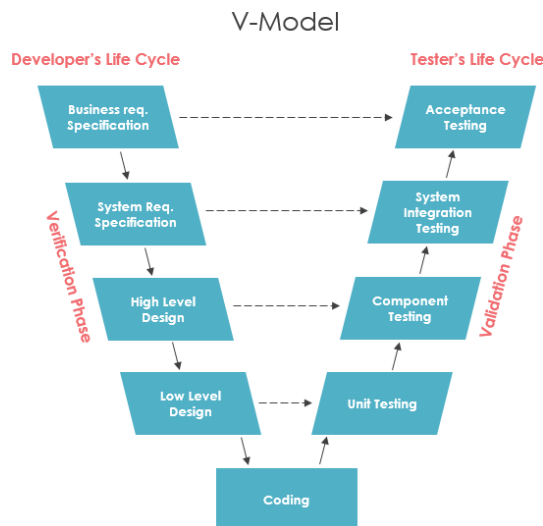
操作程式：一組書面說明，描述了正確執行安全活動所採取的分步過程。

10. What is “V model” in software engineering? 什麼是軟體工程中的「V模型」？

V 模型代表了一個軟體開發過程，可以被視為瀑布模型的擴展。

在編碼階段(coding)之後，製程步驟不再以線性方式向下移動，而是向上彎曲，形成典型的 V 形。

V 模型展示了開發生命週期(Developer's life cycle)的每個階段與其所對應的測試階段(Tester's life cycle)之間的關係。



11. List two often used approaches coping with software changes due to its inevitable nature.

由於軟體變更不可避免，請列出兩種常用的因應方法。

系統原型設計System prototyping

- 給定一版本的系統模型，以**快速確認客戶需求**，確保彈性變動，讓軟體更接近預期、符合客戶要求與確定設計的可行性。

增量交付Incremental delivery

- 以**增量的方式交付產出的結果**，透過客戶回饋和持續實驗(檢驗)，讓改變循序漸進，去避免需求變動的大更改和提升團隊對於變化的容忍度。

12. List principal factors affecting the choice of plan-based or agile software development.

列出影響選擇計劃或敏捷軟體開發的主要因素

- 專案複雜性
- 需求穩定性
- 開發團隊專業水平
- 專案規模
- 交付時間要求

Agile Software Development

13. What's Manifesto for Agile Software Development? 敏捷宣言

1. 個人與互動 重於 流程與工具 Individuals and interactions over processes and tools
2. 可用的軟體 重於 詳盡的文件 Working software over comprehensive documentation
3. 與客戶合作 重於 合約協商 Customer collaboration over contract negotiation
4. 回應變化 重於 遵循計劃(因應變動 重於 依循計畫) Responding to change over following a plan

14. List at least five principles of agile methods. 12項敏捷原則至少列出5項

1. 最優先的任務:及早且持續地交付有價值的軟體以滿足客戶需求。
2. 定期且頻繁地交付可用的軟體，週期是數週或是數月、但時間越短越好。
3. 業務人員與開發者在專案期間應每日持續互動。
4. 可用的軟體是主要衡量進度的指標。
5. 精簡，將不需要的工作項目數量最大化是重要的。

15. List at least six practices or techniques that were introduced in extreme programming (XP)?

列出極限編程 (XP) 中引入的至少六種實踐或技術？

1. **Small releases** 小量交付：做重要且有用的部分並交付給使用者確認
2. **Simple design** 精簡設計：採用最簡單容易明白的設計
3. **Refactoring** 重構：所有開發者都盡快將優化的程式碼重編，以確保程式碼簡單可維護
4. **Pair programming** 結對編程：不分彼此，個人重於專案的重要性programmer 是兩個人一組可以兩個一起討論，邊做邊check code是否有問題
5. **Continuous integration** 持續整合：每個迭代完成的結果能快速佈署到環境中，盡快且持續的交付給使用者進行確認
6. **Sustainable pace** 穩定步調：盡量不要加班
7. **On-site customer** 駐點參與的客戶：客戶onsite開發團隊中，不過實務上都是開發團隊駐點到甲方提供的環境及場所中

16. Why has the Scrum agile method been widely adopted in the practice of software development industry? 為什麼Scrum敏捷方法在軟體開發產業實務上被廣泛採用？

Scrum 則是實現敏捷開發的其中一種方法，也是敏捷開發中最被廣泛使用的框架，其以增量迭代式的軟體開發流程，重視高靈活性與彈性，可隨時視市場與用戶需求調整產品走向。

在敏捷開發過程中，可基於前一次的交付成果重複反饋，透過頻繁的增量交付，使軟體接近並達到想要的目標。協助企業根據執行專案或產品開發不斷變化的特性，進行調整並降低風險，創造高價值的產品。

因Scrum 敏捷方法具有上面幾項特性，可以有利於專案開發，所以Scrum被廣泛採用。

17. For what types of system particularly are agile development approaches likely to be successful?

敏捷開發方法特別可能對哪些類型的系統取得成功？

特別適用於需要持續應對變化、需求、技術方法不明確的軟體開發，由於規劃是漸進式的，更容易接納改變以反映不斷變化、或更複雜的需求。

18. List four practical problems with agile software development. 列出敏捷軟體開發的四個實際問題。

1. 由於需求不穩定，開發更需要技術熟練、經驗豐富且全心投入的團隊。
2. 無法提供明確的項目完成日期，使時間計劃較不確定。
3. 持續的變更可能會導致更多額外成本。
4. 快速迭代和需求變化，會讓項目難以保障成果品質。

19. List three questions that should be asked when deciding whether or not to adopt an agile method of software development. 列出三個問題在決定是否採用敏捷軟體開發方法時應該問的問題。

因為當需求還很模糊的時候，例如只跟團隊說「做一個用聲音操控的裝置」，在沒有產品規格和功能的情況下，若按照傳統的規畫邏輯（瀑布式開發），專案根本無法推展。

這是由於**瀑布式專案管理適合需求和工作方法明確的專案**，例如工程專案、舊有產品更新。**敏捷則適合專案複雜度、技術不確定的專案**，像軟體開發、新產品研發等。

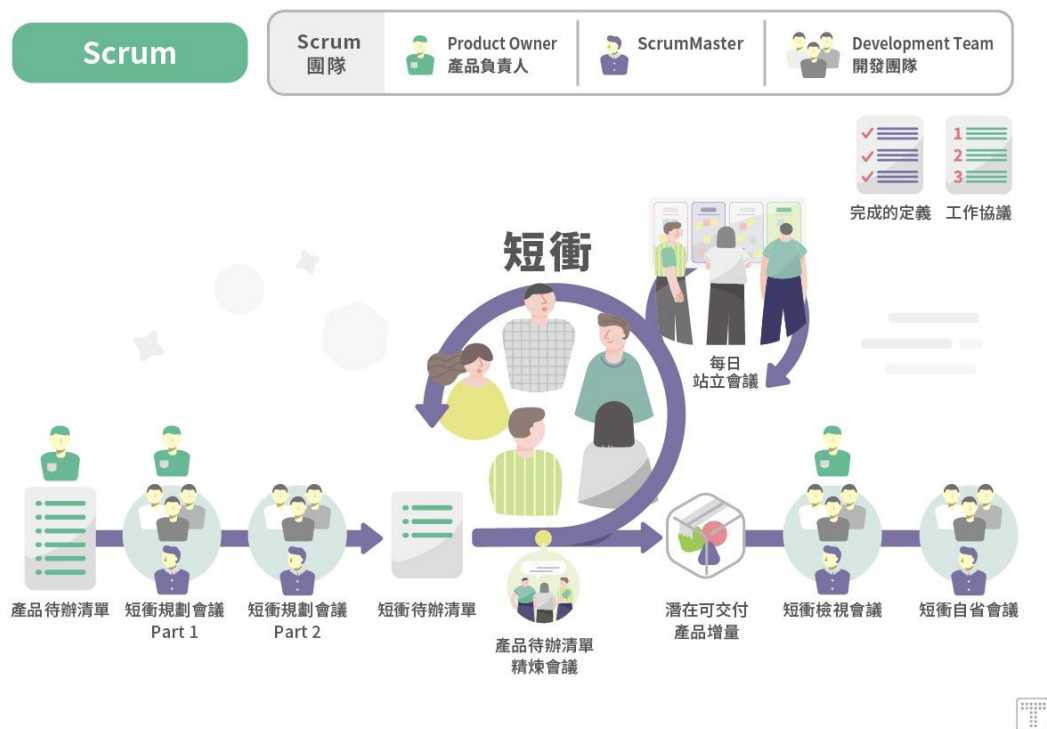
故下面幾個問題是在決定是否採用敏捷軟體開發時應該問的問題：

Q1 專案需求是否明確？

Q2 專案工作方法、使用技術是否明確？

Q3 專案複雜度如何？

20. Drawing the SCRUM workflow to illustrate three roles , three artifacts , and three ceremonies.
繪製 SCRUM 工作流程以說明三個角色、三個工件和三個儀式。



21. Why is ScrumBan widely adopted in APP development? 為何ScrumBan 被App開發廣泛應用、接納

因為它結合了Scrum和Kanban的優勢，既有Scrum的結構和規範，又具備Kanban的靈活性和流程優化，適應性強，能更靈活地應對需求變化和提高生產效率。

Capability Maturity Model Integration (CMMI)

22. What are the three models of SEI Capability Maturity Model Integration (CMMI) V1.3?

SEI能力成熟度模型整合（CMMI）V1.3有哪三個模型？

CMMI-DEV 產品和服務開發 CMMI for Development

CMMI-ACQ 產品和服務採購 CMMI for Acquisition

CMMI-SVC 服務建立、管理和交付 CMMI for Service

23. List five Maturity Levels (ML) in the CMMI V1.3 model.

列出 CMMI V1.3 模型中的五個成熟度等級 (ML)。

ML1 Initial最初的

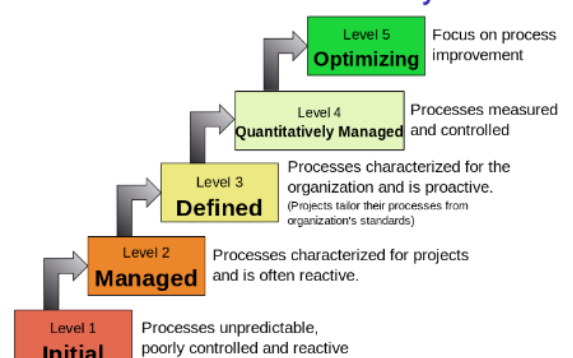
ML2 Managed管理

ML3 Defined定義

ML4 Quantitatively Managed量化管理

ML5 Optimizing最佳化

Characteristics of the Maturity levels



24. List four Capability Levels (CL) in the CMMI V1.3 model.

列出 CMMI V1.3 模式中的四個能力等級 (CL)。

CL0: 不完整或未執行(Incomplete or Not Performed)

CL1: 已執行 (Performed)

CL2: 已管理 (Managed)

CL3: 已定義 (Defined)

CMMI-DEV 流程領域 (Process Area, PA)

	專案管理	流程管理	工程	支援
成熟度等級 5 最佳化		組織績效管理 (OPM)		原因分析與解決方案 (CAR)
成熟度等級 4 已量化管理	量化專案管理 (QPM)	組織流程績效 (OPP)		
成熟度等級 3 已定義	整合專案管理 (IPM) 風險管理 (RSKM)	組織流程焦點 (OPF) 組織流程定義 (OPD) 組織訓練 (OT)	需求發展 (RD) 技術解決方案 (TS) 產品整合 (PI) 驗證 (VER) 確認 (VAL)	決策分析與解決方案 (DAR)
成熟度等級 2 已管理	專案規劃 (PP) 需求管理 專案監督與管制 (PMC) 供應商協議管理 (SAM)			度量與分析 (MA) 流程與產品品質保證 (PPQA) 構型管理 (CM)
成熟度等級 1 啟始				

制
七第

25. List four **categories of process areas(PA)** in the CMMI-DEV V1.3 model.

列出 CMMI-DEV V1.3 模型中的四類流程域。

專案管理

工程

流程管理

支援

26. List at least five process areas at ML2 in the CMMI-DEV V1.3 model.

列出 CMMI-DEV V1.3 模型中 ML2 處的至少五個流程域。

專案規劃(PP) Project Planning

專案監督與管制 (PMC) Project Monitoring and Control

構型管理(CM) Configuration Management

度量與分析(MA) Measurement and Analysis

流程與產品品質保證 (PPQA) Process and Product Quality Assurance

需求管理(REQM) Requirements Management

供應商協議管理 (SAM) Supplier Agreement Management

27. List at least five process areas at ML3 in the CMMI-DEV V1.3 model.

在 CMMI-DEV V1.3 模型的 ML3 中列出至少五個流程域。

需求發展(RD) Requirements Development

技術解決方案 (TS) Technical Solution
產品整合 (PI) Product Integration
驗證 (VER) Verification
確認 (VAL) Validation
整合專案管理 (IPM) Integrated Project Management
風險管理 (RSKM) Risk Management
組織流程焦點 (OPF) Organizational Process Focus
組織流程定義 (OPD) Organizational Process Definition
決策分析與解決方案 (DAR) Decision Analysis and Resolution
組織訓練 (OT) Organizational Training

28. List at least six generic practices within Generic Goal 2. 在通用目標 2 中列出至少六種通用做法。

GG 2 將託管流程製度化 Institutionalize a Managed Process

- GP 2.1 建立組織政策 Establish an Organizational Policy
- GP 2.2 規劃流程 Plan the Process
- GP 2.3 提供資源 Provide Resources
- GP 2.4 分配責任 Assign Responsibility
- GP 2.5 訓練人員 Train People
- GP 2.6 控制工作產品 Control Work Products
- GP 2.7 確定相關利害關係人並讓其參與 Identify and Involve Relevant Stakeholders
- GP 2.8 監控與控制流程 Monitor and Control the Process
- GP 2.9 客觀評估依從性 Objectively Evaluate Adherence
- GP 2.10 與高階主管一起檢視現狀 Review Status with Higher Level Management

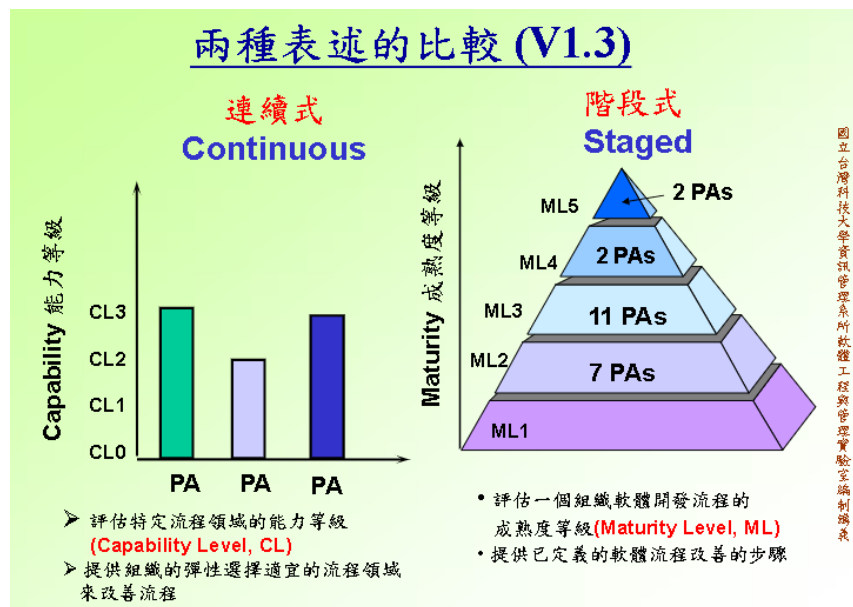
29. List two generic practices within Generic Goal 3. (2%) 列出通用目標 3 中的兩種一般做法。(2%)

GG 3 將定義的流程製度化

- GP 3.1 建立明確的流程 Establish a Defined Process
- GP 3.2 收集流程相關經驗 Collect Process Related Experiences

30. What's the distinction between stage and continuous representations in the SEI's CMMI model?

SEI 的 CMMI 模型中的階段表示和連續表示之間有什麼區別？



連續continuous：評估特定領域的能力等級（capability level，CL），提供給組織的選擇適宜的流程領域PA來改善流程

階段stage：評估一個組織軟體開發流程的成熟度等級（Maturity level，ML），提供以定義的軟體流程改善的步驟

31. What's the definition of "Defined Process" in the CMMI model?

CMMI模型中「定義過程」的定義是什麼？

指組織明確制定一套具體、有文件的標準流程，確保成員清楚了解並按照流程執行工作，以確保作業的一致性和可評估性。

能力成熟度模型整合 (CMMI) 將此定義為「一個領域中的一組實踐，當共同實施時，可以滿足一組被認為對於該領域的改進很重要的目標」。

32. What are the practices to be done for REQM PA to certify CL 1?

REQM PA 需要採取哪些做法來認證 CL 1？

All specific practices (CL1)

SP1.1: 了解需求 Obtain an Understanding of Requirements

SP1.2: 獲得對要求的承諾 Obtain Commitment to Requirements

SP1.3: 管理需求變更 Manage Requirements Changes

SP1.4: 維護雙向可追溯性 Maintain Bi-directional Traceability of Requirements

SP1.5: 辨識專案工作與需求之間的不一致 Identify Inconsistencies Between Project Work and Requirement

GG1 Generic Practice:

GP1.1: 執行具體實踐 Perform Specific Practices

33. What's the distinction between the generic goal and specific goal in the SEI's CMMI model?

SEI CMMI 模型中的一般目標和特定目標有什麼不同？

一般目標

就是相同的目標說明可適用於多個流程領域，必須能夠呈現執行流程領域流程制度化的特徵，一般被使用在評鑑中判斷一個流程領域是否滿足。

一般目標的例子是「流程制度化為已定義流程」。

特定目標

特定目標描述必須用來滿足該流程領域唯一的特徵，被使用在評鑑中判斷某個流程領域是否滿足。

例如，建構管理流程領域的其中一個特定目標是「建立與維護基準的完整性」。

34. What is the meaning of "Process Institutionalization" in the CMMI model?

CMMI模型中的「流程制度化」是什麼意思？

提及一般執行目標和一般執行方法時，制度化意味著這個流程已經是根深蒂固的工作方式，並有決心和一致性地執行這個流程。

然而，當和流程的需求及目標發生改變，流程的執行也可能需要改變以確保流程仍然有效。

35. Explain the following English abbreviations "SCAMPI, IDEAL, PAL, PIID". What are they used for?

解釋下列英文縮寫「SCAMPI、IDEAL、PAL、PIID」。它們有什麼用？

SCAMPI

CMMI流程改善標準評鑑方法：是目前唯一符合ARC等級A的評鑑方法。

普遍接受用來執行CMMI模式ARC等級A評鑑方法，SCAMPI方法定義文件(MDD)定義確保SCAMPI A評鑑等級一致性的規則。評鑑確保一致性等級，滿足特定成熟度等級或符合某一流程領域，對於不同被評鑑組織而言，SCAMPI都能確保評鑑標準相同，SCAMPI評鑑包含CLASS A、CLASS B與CLASS C評鑑方法，其中SCAMPI A被正式認可為最嚴謹的方法，也是唯一產出品質基準評鑑等級的方法。

IDEAL

一種流程改進方法、引導組織實現和維持CMMI模型中的高能力水平

Initiating 改革的刺激 設定工作內容 獲得高層的支持 建立基礎架構

Diagnosing 瞭解目前的狀況及預期目標 發展建議方案

Establishing 設定優先順序 研究可行方案 規劃行動方案

Acting 提出解決方案 試行可能的解決方案 修正解決方案 推展解決方案

Learning 分析與驗證 提出未來行動建議

PAL

流程資產庫 PAL(Process Assets Library)

流程資產的蒐集，可用於組織、專案或工作團隊。

是軟體流程相關文件的儲存庫，用以實施支援組織流程的技術基礎架構。

PIID

流程執行結果檢核表 PIID (Practice Implementation Indicator Documents) 收集相關資訊

填入PIID表主要是收集相關資訊說明此計畫達到了CMMI要求的特定目標和共性目標的直接或間接證據，如文檔，交流的郵件，人證等，需要注意的問題是PIID表中列到的文檔不能互相矛盾，專案組能夠對問題的理解要統一，否則在評估是對人的訪談會出問題。