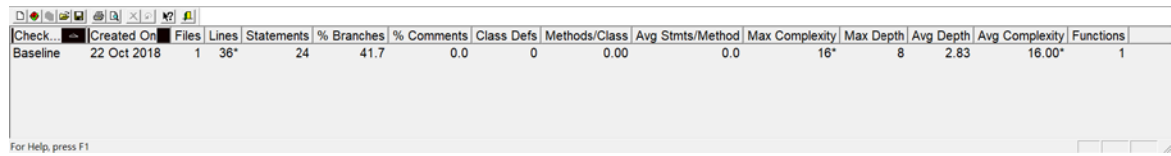


2.



The screenshot shows a software development tool interface with a table displaying project metrics. The table has the following columns: Check, Created On, Files, Lines, Statements, % Branches, % Comments, Class Defs, Methods/Class, Avg Stmt/Method, Max Complexity, Max Depth, Avg Depth, Avg Complexity, and Functions. The data row shows: Baseline, 22 Oct 2018, 1, 36*, 24, 41.7, 0.0, 0, 0.00, 0.0, 16*, 8, 2.83, 16.00*, and 1. Below the table, there is a footer that says 'For Help, press F1'.

Check	Created On	Files	Lines	Statements	% Branches	% Comments	Class Defs	Methods/Class	Avg Stmt/Method	Max Complexity	Max Depth	Avg Depth	Avg Complexity	Functions
Baseline	22 Oct 2018	1	36*	24	41.7	0.0	0	0.00	0.0	16*	8	2.83	16.00*	1

B.

1.

Project name: 網路購物

Function: 付款, 取出貨方式, 紅利點數累積, 優惠促銷活動, 會員分級, 產生報表

Scenario_1: 付款

顧客選擇想要付款方式(信用卡, 貨到付款, 郵政劃撥等)

計算該次交易總金額

輸入顧客基本資料

確認完成交易

Scenario_2: 取出貨方式,

顧客選擇想要出貨方式

輸入配送所需相關資料

確認完成交易

Scenario_3: 紅利點數累積

根據交易歷史紀錄顯示該會員累積之紅利點數

並在顧客付款時顯示可兌換之優惠

Scenario_4: 優惠促銷活動

顯示折扣商品

寄廣告信件給會員

Scenario_5: 會員分級

依會員等級發放對應獎勵

依據顧客累積下單金額為其做會員分級

Scenario_6: 產生報表

每個一段時間分析過往購買紀錄

2.

I. Please describe what is top-down and bottom-up approach:

A. top-down approach: 先決定開發方向再決定細部功能

B. bottom-up approach: 先決定細部功能再決定開發方向

II. Compare the difference between two approaches. What are their pros and cons?

A. top-down approach: 開發方向明確不易偏離主軸大方向過於攏統可能導致後續難以驗證和收尾困難

B. bottom-down approach: 細節穩定性較高但拘泥於細節未全盤考慮

3.

Q1: 如何確保個資不會外洩?

Factor: Integrity

Criteria: Access control

Metrics: 防禦惡意程式的成功率

Q2: 交易資料出錯頻率多高?

Factor: Reliability

Criteria: Error tolerance, Accuracy

Metrics: mean time between failure, Mean Time to Failure, Defect Density

Q3:如何確保系統容易操作？

Factor: Usability

Criteria: operability

Metrics: Customer Problems, Customer Satisfaction, User experience

Q4:如何維護系統功能？

Factor: Maintainability

Criteria: Simplicity, modularity

Metrics: Fix backlog and backlog management index, Fix response time, Fix responsiveness, Percent delinquent fixes, Fix quality

Q5:如何擴充系統功能？

Factor: Flexibility

Criteria: Expandability

Metrics: Expand backlog and backlog management index, Expand response time, Expand responsiveness, Percent delinquent expands, Expand quality



C.



D.

1. 比較能因應現在多變開發需求，強調和客戶間的合作關係 簡化文件的繁瑣
提高整體的效率。

2. CMM: 確認每個階段的明確待辦事項並在完成該階段後向下進行
agile development: 頻繁更新版本，強調與客戶間的溝通和互動，對與開發計畫比較有彈性。
3. 優點:簡化、減少資源浪費(工作功能規劃文件溝通)、多次少量的產出成品、因應變化、全程與客戶共同參與、團對規模較小溝通效率較高
缺點：對於已經確定流程之專案採用 waterfall 更能完整規劃，敏捷開發相對浪費資源、先決條件較多、需要客戶配，對個別成員能力要求較高，初期專案規劃有過多假設易導致大量返工成本。
4. 兩者間有相呼應但切入觀點並不一致，「學會專案管理的 12 堂課」透過與傳統開發方法之比較著重在敏捷軟體開發的優點，後者是在大家崇尚敏捷軟體開發之氛圍下提出對傳統開發方法之反思。
5. 可以。
6. 敏捷開發可採用客戶取向的 Metric，例如客戶滿意度、使用者經驗.....等等，因客戶參與度較高，容易收集他們實際回饋且透過頻繁溝通持續增進系統的完整度，可透過問卷調查來衡量。