

110 年公務人員高等考試三級考試試題

類 科：資訊處理
科 目：系統專案管理

曹勝老師解題

一、請解釋 DevOps 說明在軟體開發生命期 (Software Development Life Cycle) 中該如何執行 DevOps(25 分)

1. 《考題難易》：★★★★(最難 5 顆★)
2. 《解題關鍵》：本題為 DevOps 應用題，掌握 DevOps 概念搭配 SDLC 過程即可得到解答。

【擬答】：

(一)DevOps 模式是一種重視「軟體開發人員 (Dev)」和「IT 運維技術人員 (Ops)」之間溝通合作的文化、運動或慣例。透過自動化「軟體交付」和「架構變更」的流程，來使得構建、測試、發布軟體能夠更加地快捷、頻繁和可靠。

(二)DevOps 開發維運搭配 SDLC 的實務做法

DevOps 開發維運實務做法反映了持續改善與自動化的理念。許多實務做法著重於一或多個開發週期階段。這些實務做法包括：

1. 持續開發：這項實務做法涵蓋了 DevOps 開發維運生命週期的規劃與編碼階段。其中可能涉及版本控制機制。
2. 持續測試：這項實務做法可以在正在編寫或更新應用程式的程式碼時納入預先排定的持續自動化程式碼測試。這類測試可以加快將程式碼交付至正式作業的速度。
3. 持續整合 (CI)：這項實務做法可以將組態管理 (CM) 工具和其他測試與開發工具結合，以追蹤正在開發的程式碼有多少已準備就緒可供正式作業使用。其中涉及在測試與開發之間快速提供意見反應，以便迅速識別並解決程式碼問題。
4. 持續交付：這項實務做法可以在測試後，自動將程式碼變更傳送至正式作業前或接移環境。然後，員工可以決定是否要將這類程式碼變更推廣至正式作業環境。
5. 持續部署 (CD)：與持續交付類似，這項實務做法可以自動將新程式碼或變更後的程式碼發佈至正式作業環境。執行持續部署的公司每天可以發佈數次程式碼或功能變更。使用 Docker 和 Kubernetes 等 Container 技術有助於在不同的部署平台和環境中維持程式碼的一致性，進而實現持續部署。
6. 持續監控：這項實務做法包括持續監控運作中的程式碼以及支援該程式碼的底層基礎架構。回報錯誤或問題的回饋循環隨後又會回到開發階段。
7. 基礎架構即程式碼：這項實務做法可在各種 DevOps 開發維運階段中使用，以自動化軟體版本所需的基礎架構資源配置。開發人員可從其現有的開發工具中新增基礎架構「程式碼」。例如，開發人員可以視需求從 Docker、Kubernetes 或 OpenShift 建立儲存 Volume。這項實務做法也可讓營運團隊監控環境組態、追蹤變更，並簡化組態的復原作業。

(三)DevOps 在 SDLC 各階段執行

1. 規劃：此階段有助於定義商業價值與需求。範例工具包括 Jira 或 Git，可協助追蹤已知問題並執行專案管理。
2. 編碼：此階段涉及軟體設計及軟體程式碼的建立。範例工具包括 GitHub、GitLab、BitBucket 或 Stash。
3. 建置：在此階段，您將管理軟體建置與版本，並使用自動化工具來協助編譯及封裝程式碼，以供將來發佈到正式作業環境。您使用的原始碼程式碼儲存庫或套件儲存庫也會「封裝」產品發佈所需的基礎架構。
4. 測試：此階段包括持續測試（手動或自動），以確保最佳的程式碼品質。
5. 部署：此階段可以包含有助於管理、協調、排程和自動將產品發佈至正式作業環境的工

具。

6. 操作：此階段可在正式作業期間管理軟體。

7. 監督：此階段包括識別及收集有關正式作業中特定軟體版本問題的資訊。

志光學儒保成 工科人專屬學習規劃

精心安排完整豐富的上榜課程

工科考試所需要的資源，我們通通幫你準備好了

<p>法科架構班</p> <p>學校沒教的，我們教給你！名師精解法科知識，結合實務例子，助你建構法科概念。</p>	<p>扎實正規班</p> <p>完整堂數規畫，循序漸進學習，讓您深度修習工科各專業學科知識。</p>	<p>作文實戰班</p> <p>作文再也不是理工人的痛！透過專業老師的輔導，快速強化您的寫作架構、邏輯概念。</p>
<p>主題題庫班</p> <p>主題式教學，搭配各類試題演練，進行考點分析及破題要點訓練，讓您短時間各科實力倍增。</p>	<p>全國全真模擬考</p> <p>檢視應考實力、訓練臨場反應、掌握最新考題趨勢，全程比照考試時程，模擬考場實戰氛圍，讓您能以平常心應考！</p>	<p>精華總複習</p> <p>考前重點總複習，精準掌握重要考點，讓您考前實力突飛猛進。</p>
<p>考前提要關懷講座</p> <p>名師考前最終提點，穩定你累積許久的實力，讓你的觀念更加清晰。</p>	<p>工科全科班</p> <p>公職+國營完善循環課程規劃，All in One課程一次到位，奠定穩固基礎、強化上榜實力。</p>	

109普考 電子工程
曾○維
一年考取

我是工科人，我工頂啦！

由於考試的題目非常靈活，參加題庫班，除了勤做考古題外，大量實作解說，很快地強化我的考前記憶，每做一道題目馬上能判斷是在哪一章節，然後再進行解題。

■完整課程資訊詳洽全國志光·學儒·保成門市■

二、請解釋何謂完全測試(Complete Testing)?該如何執行?可行性如何?請使用 Windows 或 Mac 作業系統中常的小算盤程式(Calculator Program)來舉例說明之。(25 分)

1. 《考題難易》：★★(最難 5 顆★)
2. 《解題關鍵》：本題為測試概念題，了解測試時設計測試個案觀念即可得到解答。

【擬答】：

(一)完全測試是一種質量保證的測試技術，其中所有可能的場景或資料都要測試以完成此一過程，以確保在測試階段結束時沒有未被發現的問題。例如小算盤程式若要進行完全測試，需要將所有的數字組合全部輸入一次，如 1+1,1+2,1+3,...。即使你在這個世界上有所有的時間，你仍然沒有足夠的時間來測試所有可能的輸入組合和輸出組合。

(二)完全測試執行方式

1. 找出所有可能的輸入與對應的輸出，例如小算盤程式的輸入包括有效輸入，也就是所有的數字組合；以及無效輸入，數字與其他錯誤資料(如文字)的組合。而且每一個都要列出其輸出。
2. 考量所有的狀態，例如小算盤程式進行中需要先輸入第一個數字，再輸入要進行的運算(+, -, *, /, ...)，接著再輸入第 2 個數字，再輸入後面可能的運算，再輸入....
3. 考量所有的情境，例如小算盤程式可能在不同的溫度環境下都要能運作，就要設計從絕對零度到天曉得多少度的情境。
4. 根據上列設計測試個案，加以執行。

(三)完全測試幾乎完全不可行，原因如下：

1. 程序的可能輸入域太大而無法完全用於測試系統，包括有效輸入和無效輸入。例如小算盤程式需要將所有的可能數字組合，包括有效與無效的都設計測試個案。
2. 程序可能有大量的狀態。輸入可能存在時序約束，即輸入可能在某個時間有效，而在其他時間無效。例如小算盤程式需要考量不同時間可能出現的數字，以確保全部都考量到。
3. 可能無法創建系統的所有可能執行環境。當系統的行為取決於真實的外部世界時，例如天氣、溫度、海拔高度、壓力等，這一點變得更加重要。例如小算盤程式需要將所有的可能測試溫度狀態考慮進來，以確保全部都考量到。

三、在系統開發中,有時為了因應顧客時程需求或企業為緊抓市場商機而要求專案成員縮短開發工期(即縮短時程),在此情況下管理者通常會允諾增加可用的資源。巴里·玻姆(Bary Boehm)建構成本模型(Constructive Cost Model,通常被簡稱為 COCOMO)與勞倫斯·普特南(Lawrence H. Putnam)的軟體生命週期管理模型(Software Life Cycle Management Model,通常被簡稱為 SLIM)為常用的軟體開發時程與成本預估模型。請針對一個二年期的資訊系統開發專案,如果開發人員被要求得改在 22 個月、20 個月及 12 個月內完成該專案,請使用軟體生命週期管理模型來舉例說明上述三種不同的縮短時程要求會對專案整體工作量造成何種程度的影響?實際上是否可行?請從技術面與管理面上說明專案經理或開發人員該如何面對與處理。(25 分)

1. 《考題難易》：★★★★(最難 5 顆★)
2. 《解題關鍵》：本題為軟體生命週期管理模型應用題，需要掌握普特南公式才可得到解答。

【擬答】：

(一)Putnam 的軟體生命週期管理模型是一種參數式軟體專案成本預估模式，其公式如下：

$$S = E * K^{\frac{1}{3}} * t_d^{\frac{4}{3}}$$

S:軟體專案大小

t_d :開發時間

K:開發所需總人力

E:軟體開發環境指標,包括了硬體的限制、程式的複雜度、專案成員的經驗等眾多相關開發因素的交互作用

因此若改變開發時間(t_d 變小),在軟體專案大小與軟體開發環境指標不變的情況下,所需投入人力將大幅增加,因為 $K = (S / (E * t_d^{\frac{4}{3}}))^3$, 例如若將時程縮到 12 個月(原來的 1/2),需要投入的人力為 $(1 / (1 * 0.5^{\frac{4}{3}}))^3 = 16$, 也就是原來人力的 16 倍,且 E 勢必會改變,在管理上很難達成。

(二)Putnam 模式更進一步的提出了可壓縮的專案時程與專案困難度(D)間的關係為

$$D = \frac{K}{t_d^2}$$

即困難度和可壓縮時間的平方成反比,與人力投入數量成正比。因此若投入人力不變,三個縮時方案的困難度為

1. 縮短到 22 個月, t_d 成為原來的 22/24, 若投入人力不變, 困難度成為原來的約 1.19 倍。
2. 縮短到 20 個月, t_d 成為原來的 20/24, 若投入人力不變, 困難度成為原來的約 1.44 倍, 大幅提高。
3. 縮短到 12 個月, t_d 成為原來的 12/24, 若投入人力不變, 困難度成為原來的約 4 倍, 基本上已經是不可能的任務。

(三)這時需要採取趕工縮程(Crashing),這是經由分析多種替代方案,將更多人力資源安置在要徑上之活動,以縮短完工工期而不改變任何活動排序的一種專案時程壓縮技術。然而,「趕工縮程」可能增加人力資源或加班費用,不利於成本績效。

四、請解釋軟體演進(Software Evolution)與軟體演進定律(Law of software Evolution),請至少列舉三條定律說明其意涵。此外倘(Development Team)採用傳統軟體開發方法(Traditional Software Development Methodologies)但演進團隊(Evolution Team)卻選擇採用敏捷方法(Agile Methodology);或是開發團隊選定敏捷方法但演進團隊卻使用傳統軟體開發方法。試問上述這二種狀況各會對軟體演進造成何種影響?請詳細說明。(25 分)

1. 《考題難易》：★★★★(最難 5 顆★)
2. 《解題關鍵》：本題為軟體演進應用題，必須掌握開發方法與演進方法的本質才能作答。

【擬答】：

(一)軟體演進(Software Evolution)是說軟體必須持續修改,以符合客戶和市場的需求變化。軟體演進定律包括下列：

公職王歷屆試題 (110 高考三級)

1. Continuing Change

系統必須不斷適應，否則它會逐漸變得不那麼令人滿意。

2. Increasing Complexity

隨著系統的發展，它的複雜性會增加，除非進行維護或減少它的工作。

3. Self Regulation

系統演化過程是自我調節的，產品和過程措施的分佈接近正常。

4. Conservation of Organizational Stability (invariant work rate)

在演化的系統中，平均有效的全球活動率在產品的整個生命週期內是不變的。

5. Conservation of Familiarity

隨著系統的演進，所有與之相關的開發人員、銷售人員和用戶，例如，必須保持對其內容和行為的掌握，才能實現令人滿意的演進。過度增長會削弱這種掌握。因此，隨著系統的發展，平均增量增長保持不變。

6. 持續增長(Continuing Growth)

系統的功能內容必須不斷增加，以在其生命週期內保持用戶滿意度。

7. 品質下降(Declining Quality)

若無法得到嚴格維護並適應運行環境的變化，系統的品質會下降。

(二)若負責系統開發的開發團隊與負責維護的演進團隊採不同的方法，則會造成維護問題，如下所述：

1. 如果系統開發採用敏捷方法，則相應的維護也只能是敏捷維護。若採傳統結構化方是維護，因為這時系統軟體配置的惟一成分是程式原始碼，一旦有系統維護的需求時，維護工作只能從艱苦的評價程式代碼開始。由於沒有完整規範的設計開發文檔，無程式內部文檔，對於軟體結構、資料結構、系統介面以及設計中的各種技巧很難弄清，如果編碼風格再差一些，則系統維護工作十分艱難，因此，有許多軟體人員寧可重新編碼，也不願維護這種系統。另一方面，由於無測試文檔，不能進行回歸測試，對於維護後的結果難以評價。
2. 如果系統開發採用了傳統結構化方法，則系統交付時有完整的軟體配置文檔，此時若採敏捷方法，將不會考慮到修改可能帶來影響的情況下，設計修正錯誤的途徑。也不會使用測試說明書中包含的測試方案進行回歸測試，如此將大大提高維護的工作量，降低軟體品質。

志光學儒保成

公職工科+國營事業

1+1 更有力

準備公職的同時，可報考國營事業考試，善用重疊考科，一次準備就能多次上榜！

上榜路徑大公開！一年內超過8次上榜機會！

初等考 1月 ● 最容易上手的公職考試	關務特考 4月 ● 考科少於同職等考試	鐵路特考 6月 (110年因疫情延至9月) ● 佐級錄取率最高	高普考 7月 (110年因疫情延至10月) ● 主流考試，缺額眾多	調查局特考 8月 (110年因疫情延至10月) ● 三等月薪76,000起
地方特考 12月 ● 考科同高普考	自來水評價人員 不定期舉辦 ● 只考選擇題	台電考試 不定期舉辦 ● 考科少、好準備	中油僱員 不定期舉辦 ● 只考2科，多為選擇題	國營事業職員級 不定期舉辦 ● 國營退休潮，缺額多，工科類科競爭者少

錄取率高

109年 工科錄取率最高達**19.42%**

電力工程 高考 19.42% 普考 17.33%	電子工程 高考 9.04% 普考 9.39%	機械工程 高考 18.27% 普考 13.70%	資訊工程 高考 12.92% 普考 10.47%
---------------------------------------	-------------------------------------	---------------------------------------	---------------------------------------