P2

再開始之前我先簡介一下DevOps 這張圖片上左邊的圓圈是開發dev 右邊的是維運Ops的部分

DevOps重視這兩種人員之間的溝通，減少不同單位的資訊隔閡

透過DevOps tool讓build到deploy這段盡可能的自動化，軟體能夠更加地快捷、頻繁和可靠。

從二月到現在大部分學習的內容都在test release 這段

包含了CI CT

P3

我把這段時間學習的內容依照這些技術使用的階段

排成一個devOps toolchain 我報告的順序大致也會跟著這個順序走

P4

Git可以理解成專門放code的雲端硬碟

Git的實作是要pull別人的下來修改完再上船push回去

P5

第一個專案讓我們了解整個devops流程

建立個maven專案 (念2.3點 還記得要設定JAVA\_HOME那些環境變數  
git push 到gitlab然後再用jenkin script去build  
Jenkins script是用linux寫的所以要學會前面基礎的Linux

P10 stage分段就是這裡 linux

P12 就是重新git push build看看有沒有成功

P13 從三月中開始 剛剛我們自己建立的maven換成cslaw這個專案然後開始寫UT接著透過git push CI/CD

P14Before 每次執行test之前都會先做的動作

要驗證這個Get Set 是否正確

左邊是在test中指定的字串內容  
右邊是由20.21行 使用source code set的method設定

再用Get來看看是不是相同

P16 第一行 先用getClass()getName拿到前面的名稱 後面取得hashCode後再轉成String

試過\t 空白 最後是用Line Separator

P19原本Assert的寫法只會測試53 61的return  
中間paging那個BuildOrder要另外用argumentCaptor去取得回傳值

在69行驗證是否有正確的取得數值

P20Bind這個方法沒有回傳所以不能用mock產生  
所以不能回傳bind成功的訊息回來只能透過連接Ldap server才行 因此這個是無法測試的

P18<https://www.globenewswire.com/news-release/2019/10/29/1937163/0/en/BigPanda-Expands-AIOps-Capabilities-Launches-Root-Cause-Changes-to-Fast-Track-Incident-and-Outage-Resolution.html>

1. Bigpanda能整合各個devops工具像這邊有列出的是jenkins AWS Jira

2.特點是白箱測試 讓IT人員對他可控可測

3. LØ 通知相關的開發為運人員還有console分析功能(這個深入研究

P19這個分析功能就是根本變動偵測:由於bigpanda整合了所有的平台，他能夠擷取任何的更動，像是管理規則改變log改變 configue改變都能透過ML去跑然後預測出這個變動對系統造成的潛在問題或漏洞呈現在monitor上

Xinder  
有別於學校所學:以前學的東西很分散，一下學java 一下DB但是都沒有一個實際的project不知道這些技術是要幹嘛用的，偏向工具導向，只教一個一個的工具，在中信這裡偏向任務導向，讓我們知道說要建一個Pipeline CI/CD需要jenkins testcase git然後才開始學 比較有目標  
時間管理:好，三月初開始寫UT的時候那時候每天都很充實，連續打了三天也打不完那個禮拜的作業量，堆積到下禮拜還要處理之前的，發現要檢視自己的效率，會不會被一個bug卡很久  
解決問題能力:大部分的問題靠google或自己去對照SAMPLE ut有沒有相似可以借看的地方，如果真的不行清楚地找出不能解決的點然後詢問主管或學姊

團隊合作:當我的問題解決了，分工是一個禮拜解決一個資料夾，組員間會互相協助幫忙DEBUG  
工作氛圍優質:位子做起來很舒服，主管也很用心指導對我居家辦公報告給了不少的建議，有一個額外學習的機會，