

平臺工程

Platform Engineering

為何是企業IT現代化新關鍵

iThome 電腦報周刊副總編輯 王宏仁

ray@mail.ithome.com.tw

2021年 台灣SRE人 需求起飛

Senior Site Reliability Engineer (SRE) 03/20更新

中磊電子股份有限公司 本公司其他工作



軟體開發_網站可靠性工程師SRE (新竹廠)R08 03/24更新

友達光電股份有限公司 本公司其他工作

科技金融_ SRE Engineer 03/22更新

中國信託商業銀行股份有限公司 本公司其他工作

顯示技
雲端與
更體整

工作內容

1. 為新一代開放銀行核心建立
2. 建立運維流程及文件，開發
3. 以敏捷模式不斷優化、改善
4. On-Call處理線上問題和異常
5. 協助應用於容器之運行監控
6. 研究新技術，導入日常運維

職務類別 軟體專案主管、軟

Site Reliability Engineer 03/10更新

台灣積體電路製造股份有限公司(台積電) 本公司其他工作

Role

We are looking for a highly motivated and critical-thinking site reliability engineer in our Services Division. The ideal candidate should have exposure to systems in both development and production teams. This position requires software development, support, IT operations, and

Responsibility

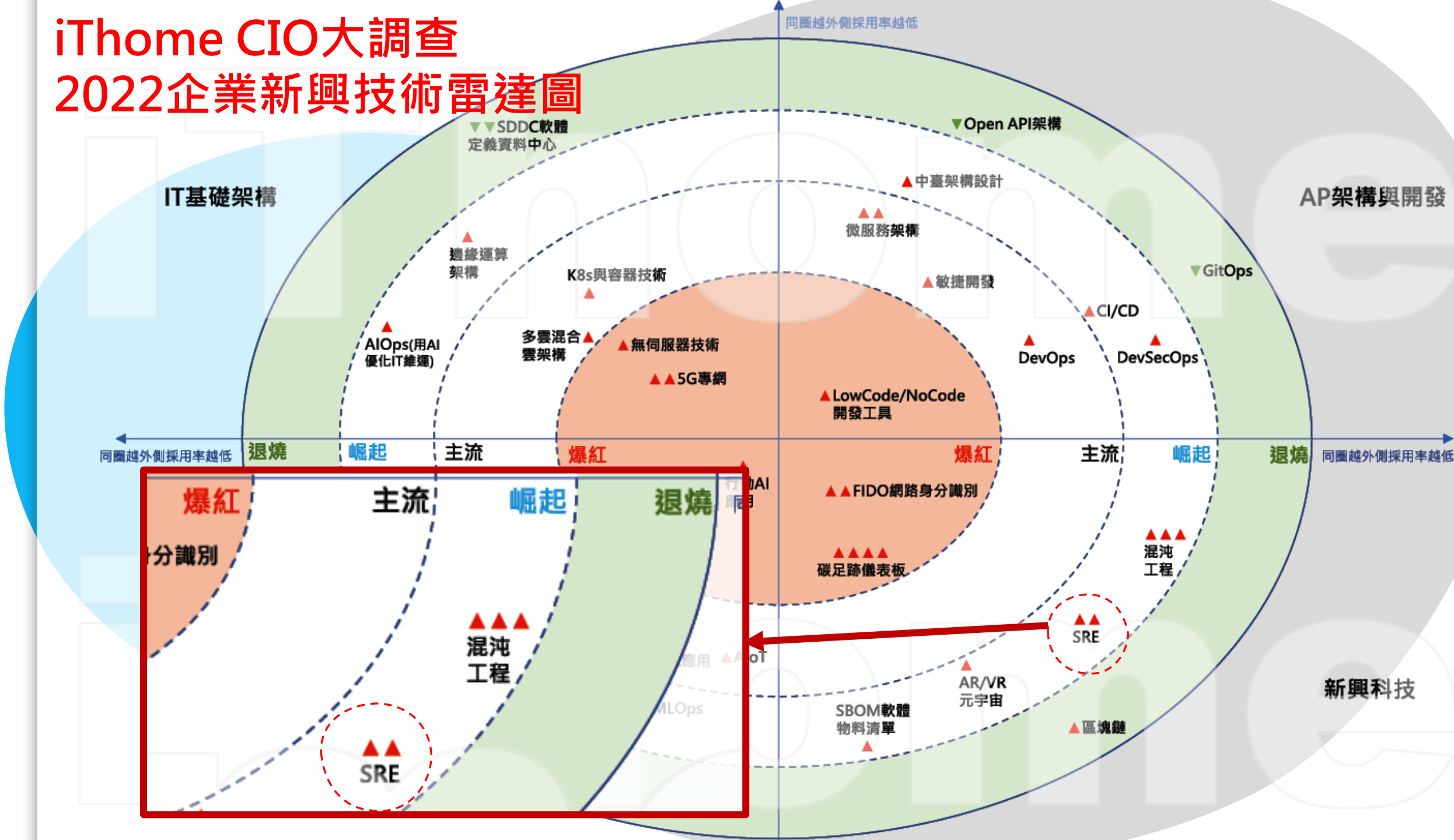
軟體類--SRE架構師(中科) 03/20更新

華邦電子股份有限公司 本公司其他工作

工作內容

iThome CIO大調查

2022企業新興技術雷達圖



Site Reliability Engineering

網站可靠性工程師

- SRE是一種維運心態，以及確保系統可靠性的一系列實作、指標和慣例方法
- 網站是SRE主要守護對象，面對外部顧客的網站
- 大規模使用者、爆量需求，又要「高度可靠」
- 剛剛看到的企業SRE招募，不全是為了服務終端顧客，
- 很大比例是服務企業內部的顧客：企業員工或開發團隊。
- 「非典型SRE」

工作內容

【邀請您將104履歷同步上傳至華邦官方網站，將使您的履歷優先被主管看見】

投遞網址:<https://reurl.cc/b7E3bM>

1. 設計和構建高可用、高擴展性和高效能的基礎設施系統
2. 負責系統的監控、故障排除和容錯設計
- ~~3. 協助開發團隊將 GitOps 流程實現到 CI/CD 流程中，以實~~
4. 建置並維護 DevOps 平台工具以支援開發團隊，如Kuber

Senior Site Reliability Engineer (SRE) 03/20更新

中磊電子股份有限公司 本公司其他工作

工作內容

Sercomm has been committed to the development of broadband networking software and fir comprehensive IoT solutions. To help us build functional IoT cloud-native application, we're looking for a SRE engineer who can be responsible for deploying product updates, ide implementing integrations that meet our customers' needs.

1. Building and implementing new development tools and infrastructure
2. Understanding the needs of stakeholders and conveying them to developers
3. Working on ways to automate and improve development and release processes
4. Testing and examining code written by others and analyzing results

高科技製造業 平台團隊 實例

- 新竹一家超大科技製造業，在2021年開始大舉招募SRE能力的人才，後來，更發展出服務內部開發的專責平臺團隊，像是在數位業務部門下新設立了一個平臺團隊，負責基礎服務和設施，提供數位商務需要的共用服務，例如資安、商務應用平臺、資料平臺解決方案、共用服務等中臺服務，讓雲原生架構打造的基礎架構，就算面臨頻繁、控制系統的變更，也能提供強健又便利的運作環境，來支援開發團隊所需。
- 這正是一種非典型SRE的做法

Platform Engineering

平臺工程

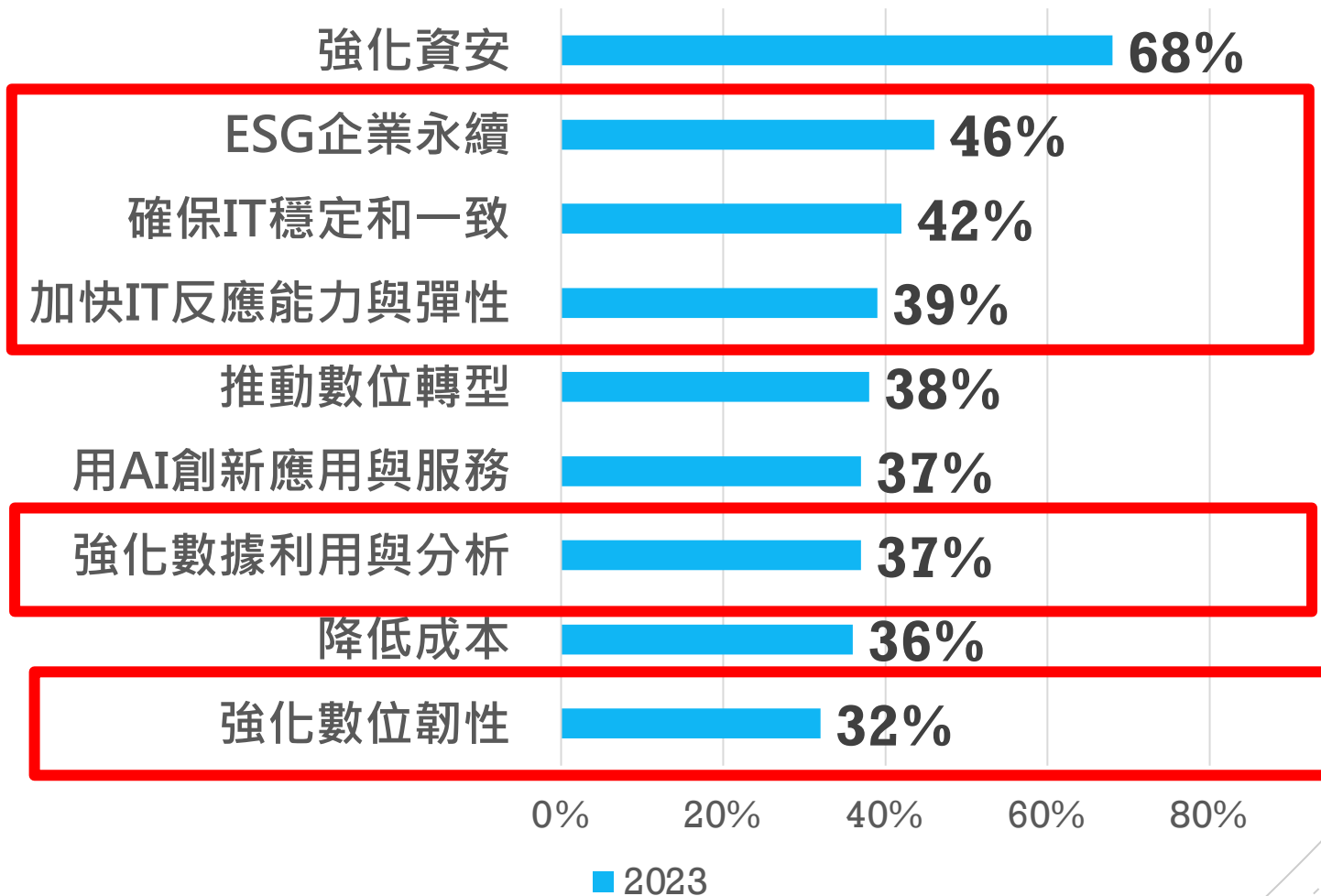
- 服務內部團隊
- 將他們常用功能、共用系統、通用模組等都服務化（或者微服務化）
- 打造成一套平臺，以產品角度來管理和維運
- 去年，國外出現了一個專有名詞，來形容這種非典型SRE的做法，
- **Platform Engineering**
(平臺工程)

The background features a series of concentric circles in light gray, some solid and some dashed, creating a ripple effect. A large blue speech bubble is centered on the page, containing white text.

企業為何需要
專門/專責平台工程團隊
服務內部的開發團隊？

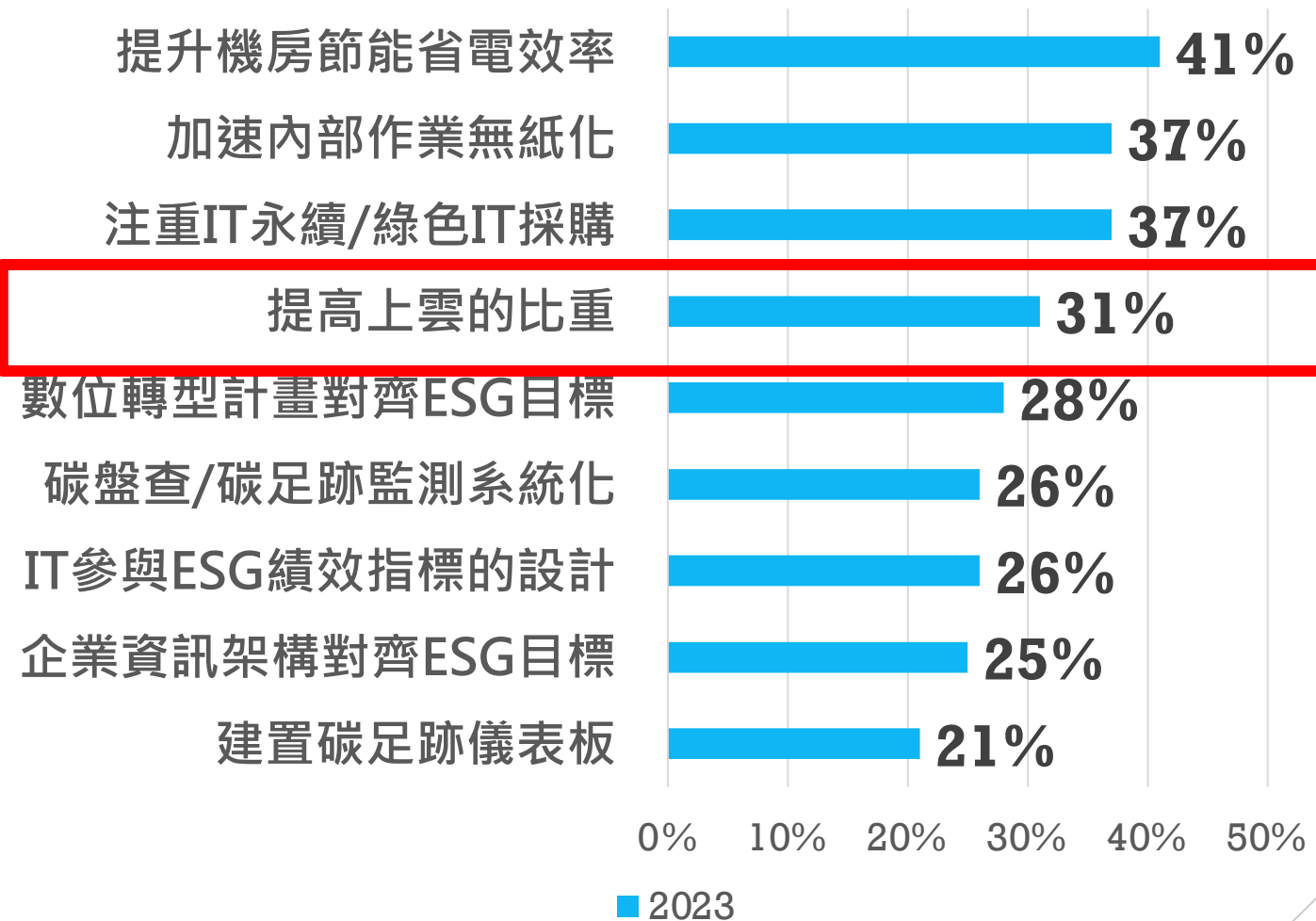
2023
108位CIO
年度優先目標
Top10

2023



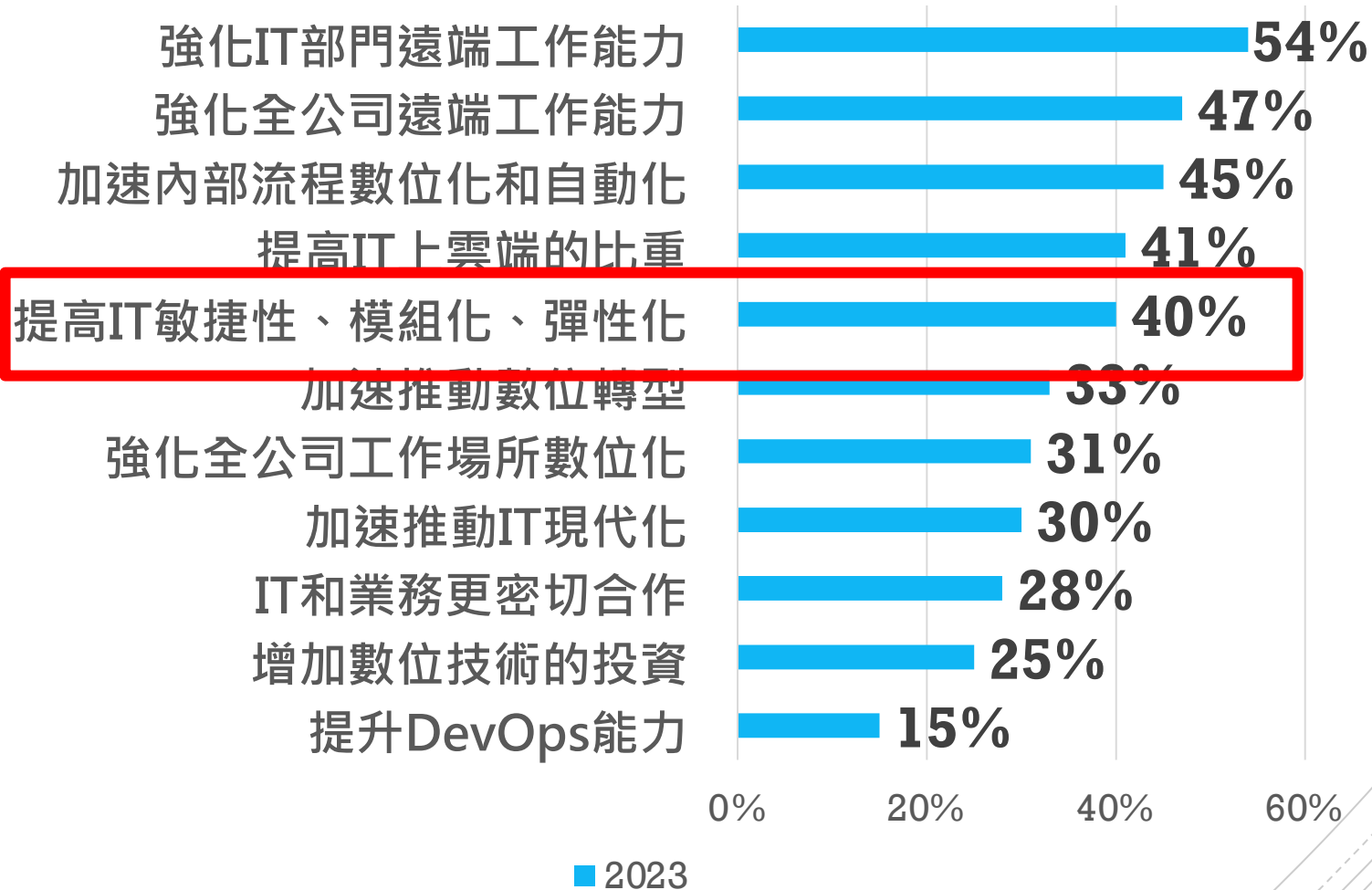
2023
108位CIO
10大IT因應ESG
做法

2023



2023
108位CIO
因應後疫情新
常態IT對策
Top12

2023



為何SRE需要更細分？

數位服務需求分兩大類

一類是與商業邏輯相關的需求，稱為功能性需求（Functional Requirement）

另一類與業務邏輯無關的需求，例如可觀察性、資安、流量、網路、擴充性、分散式優化等則稱為非功能性需求。

CTO Talk

數位轉型帶動企業微服務浪潮，SRE是

到了網際網路環境，一隻AP的程式碼，要加上非功能性能力和SRE團隊，才能打造出多數人可用。SRE可以說是AP現代化的第二階段，也是數位服務的重要構成。

檢視 編輯

文/ 王宏仁 | 2022-10-03 發表



F5執行副總裁兼技術長 Geng Lin (圖片來源 / F5)

曾負責打造摩根大通次世代雲端數位銀行服務的網路大廠技術長 Geng Lin

臺灣這幾年掀起一股IT現代化的浪潮，不少企業開始打造新一代資訊系統或數位服務，微服務浪潮在2022年超過1成大型企業要導入。但是，企業應用架構，大幅翻轉成微服務架構，不是一件容易的事。技術長Geng Lin更指出：「微服務對企業最難，也帶來新技能的挑戰。」

當代數位服務的
非功能需求
程式碼，遠遠
超過了功能性
需求

- 對企業來說，只有功能性需求的服務，才能創造差異化，這是商業創新的目的。
- 若沒有辦法將非功能性需求變成可重複使用的平臺能力，
- 企業開發團隊，可能高達70%時間開發與AP功能無關的事

70% 時間用於無法創造差異化價值的需求

非功能性需求 提高用戶體驗 必解課題

1. 現代的分散式應用或數位服務中，這些非功能性需求是提高用戶體驗必須解決的課題。
2. 平臺工程團隊負責這些與商業創新無關，但又是企業必要的非功能性需求。
3. 國外越來越多企業設立平臺工程團隊，或將現有IT維運團隊，轉型成平臺工程團隊來負責非功能性需求。
 - 平臺工程工程師和SRE工程師
 - 運用雲端原生技術，維運雲原生技術打造的服務、管理日益龐大的企業內部微服務，但兩者的本質和運作思維截然不同
 - 用新名詞，方便區分兩者各自的專業領域不同。

平臺工程思維 與傳統IT服務 思維大不同

- 「從平臺工程的思維，來打造企業內部應用，與用傳統IT服務的思維相比，兩者有很大的不同。」

■ By Google CTO辦公室技術總監
知名雲端戰略顧問Gregor Hohpe

平臺工程 VS. 傳統IT

平臺工程與傳統IT服務思維大不同

特性	平臺工程思維	傳統IT服務思維
規模化能力	成長性高	容易有瓶頸
邊際成本	低	中到高
採用阻力	低	高
互動形式	自助服務	工單派工
發展思維	顧客為中心	流程為中心
演進方式	持續進化	零星進化
延伸能力	開放或半開放	封閉
推廣策略	自主導入	強制導入
權責劃分	責任共享	責任分工

資料來源：Gregor Hohpe，iThome整理，2023年1月

自助式服務 VS. 工單派工

- **看長不看短**：同樣用雲原生技術打造內部後端服務，但平臺工程思維，更以長期、大規模發展來思考，因此，成長性高、容易擴充，更可以降低邊際成本
- **與內部顧客（開發團隊）的互動方式大不同**
 - 平臺工程：採取自助服務式設計，因此，用戶上手門檻低而能降低採用阻力。
 - 傳統IT：多以表單申請，或是工單派工形式（Ticket）來提出需求，內部員工常常抱怨等到天荒地老，IT抱怨工單做不完
- **自助式服務的好處**：減少例行性、重複性、無聊的開發工作，也能加快開發團隊交付應用的速度
- 有些平臺工程團隊的目標是，連非技術專業的一般員工也能用，減少更多「**工作小事的開發負擔**」

發展和推廣 策略大不同

- **開放性**：平臺工程思維，更容易建立一套通用的擴充機制，或是可以定出一套後端應用與外部介接的做法，讓後端服務可以整合到第三方應用，或嵌入外部提供的API，因此可以建立一個開放或半開放的應用體系。
- 例子：**國泰金控**可以建立一套使用外部開源程式碼的通用規範或可用開源軟體清單（程式庫），簡化和加速開發團隊使用外部開源資料的「審查流程」
- **導入策略**：平臺思維更容易採取激勵、自我學習的自主導入推廣方式，而不用像傳統IT服務，多半由上而下的要求來強制採用。

自助式 平台服務 如何更進化

- 自助式服務是內部服務平臺化後的重要呈現形式
- 以入口網站形式，結合開放式架構設計後，更可以整合NoCode開發工具、RPA工具、互動式BI或分析探索工具等
- 很適合延伸到非技術專業的一般員工
- 來分攤一部分的IT開發工作

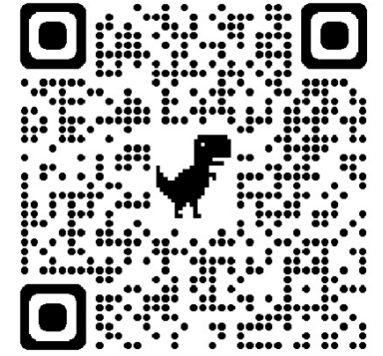
分攤開發責任 IT更專注業務

1. 現代軟體架構越來越**複雜**，才催生了平臺工程的崛起，甚至為了讓**非專家的一般員工**，降低運用內部後端服務的門檻，超前思考的企業會開始建置這類**維運型的平臺**。
2. 平臺工程做法可以將IT應用開發責任分散到一般員工，讓業務團隊也能分擔一些簡單但又高度客製化的小需求，讓開發團隊可以有更多人力專注於核心業務。
 - By國際分析機構Gartner分析副總裁Paul Delory

平臺工程Case

- **Clarity AI**：利用**AI** 提供**ESG**分析，每週追蹤**200**萬個數據點，進行**ESG**評分，提供企業**ESG**分析所需
- **Clarity AI**平臺工程經理**Eduardo Ferro Aldama**經驗
- 平臺工程導入目的
 1. 降低認知負載/錯誤/阻力
 2. 改善工程團隊生產力：減少相互等待、提高部署頻率、縮短上線準備時間（**commit to production**）
 3. 發布系統和產品擁有權整合
- 來看一個降低負載的例子：平臺工程**Chatbot**
-

- Before / It required your full attention for more than 1 hour
- Now / Fire and forget, and it takes half time



Marina Lodeiro 12:47 PM
Ok, so here we go

12:47 Release time 🤪
👍 1 🙌 2 😊

```
/clarity release app backend stg admin_token 54e8dcf8
```

platform APP 12:48 PM
<ReleaseBackendCommand 2a638301-e1d8-4f61-b8de-ec7e68c02d29> scheduled to run as a kubernetes job, waiting for pod to start...

Marina Lodeiro 12:48 PM
/clarity release app frontend stg admin_token 54e8dcf8

platform APP 12:48 PM
<ReleaseFrontendCommand 8c674dd5-d352-459d-8417-b1d0574968d1> scheduled to run as a kubernetes job, waiting for pod to start...

platform APP 12:49 PM

- ✓ Releasing Backend to stg - `2a638301-e1d8-4f61-b8de-ec7e68c02d29`
- ✓ Communication
- ✓ Update postgres schema
- ✓ Update mongo schema
- ✓ Update i18n view pipeline
- ✓ Deploy view job
- ✓ Create hotfix branch hotfix branch
- ✓ Run e2e tests view pipeline (edited)

platform APP 12:49 PM
set the channel topic: [DEV→FH - STG→FG 🚧 ongoing 🚧 - PRO→FF] 📢

Low cognitive load

Fast & Clear Feedback

<https://www.youtube.com/watch?v=Bh4lvGUnVZc&t=1s>

Zalando 歐洲服飾圈的 Amazon

- 歐洲最大的快時尚平臺，23國銷售，從賣鞋子起家，到後來，賣各種快時尚的服裝，體育用品等，幾個數字讓大家了解一下這家公司的規模
- 一年營收約80億歐元，相當於台幣2400億元。
- 2022年Q1顧客人數有4800萬人，5800萬張訂單，都是透過網路平台完成
- 一年約130-150億台幣，投資在物流和科技平台上。



Welcome to Sunrise!

Your daily starting point

Take Action

Pipeline Activity

- Merge pull request #422 from engineering-community, [redacted] [VIEW](#)
succeeded [progress bar]
- New endpoint for Application Info (#187) [redacted] [VIEW](#)
succeeded [progress bar]
- Merge pull request #41 from developer-productivity, [redacted] [VIEW](#)
succeeded [progress bar]

[See all pipelines →](#)

Open Pull Requests

- [WIP] [redacted] [VIEW](#) [@](#)
review requested [progress bar]
- Helper script to [redacted] [VIEW](#) [@](#)

Health & Performance Indicators

Software Delivery (last week)

	STUPS-BUILD	STUPS	PLATFORM
Metrics		This Team	Target
Deployment Frequency		2 deploy days	5 deploy days
Lead Time		Coming Soon!	

[Learn More →](#)

What's new

Updates about the Platform

Check out the recording of the Welcome Demo for an overview of the current features and vision for Sunrise.

Sunrise日出平台 (雲原生技術自行開發)

■ 資料來源: Zalando

Databases

ADOPT

1. AWS DynamoDB
2. AWS S3
3. Elasticsearch
4. etcd
5. Exasol
6. PostgreSQL
7. Redis

ASSESS

12. Amazon MemoryDB
13. RocksDB

HOLD

14. Aerospike
15. Apache Cassandra
16. Consul
17. CouchBase
18. Hazelcast
19. HBase
20. Memcached
21. MongoDB
22. MySQL
23. Oracle DB
24. Solr
25. ZooKeeper

TRIAL

8. Amazon Feature Store
9. Amazon Redshift
10. Druid
11. HDFS

Infrastructure

ADOPT

45. Amazon SageMaker
46. AWS CloudFormation
47. AWS CloudFront
48. Docker
49. Kubernetes
50. OpenTracing

ASSESS

55. Slurm
56. WebAssembly

HOLD

57. STUPS
58. ZMON

Zalando Tech Radar

■ <https://opensource.zalando.com/tech-radar/>

Data Management

ADOPT

26. Airflow
27. AWS EMR
28. AWS Kinesis
29. AWS SNS
30. AWS SQS
31. Databricks
32. Kafka
33. Nakadi
34. Spark

ASSESS

39. AWS Athena
40. AWS Glue
41. dbt

HOLD

42. AWS Data Pipeline
43. Hadoop
44. YARN

TRIAL

35. Flink
36. Google BigQuery
37. Presto
38. RabbitMQ

Languages

ADOPT

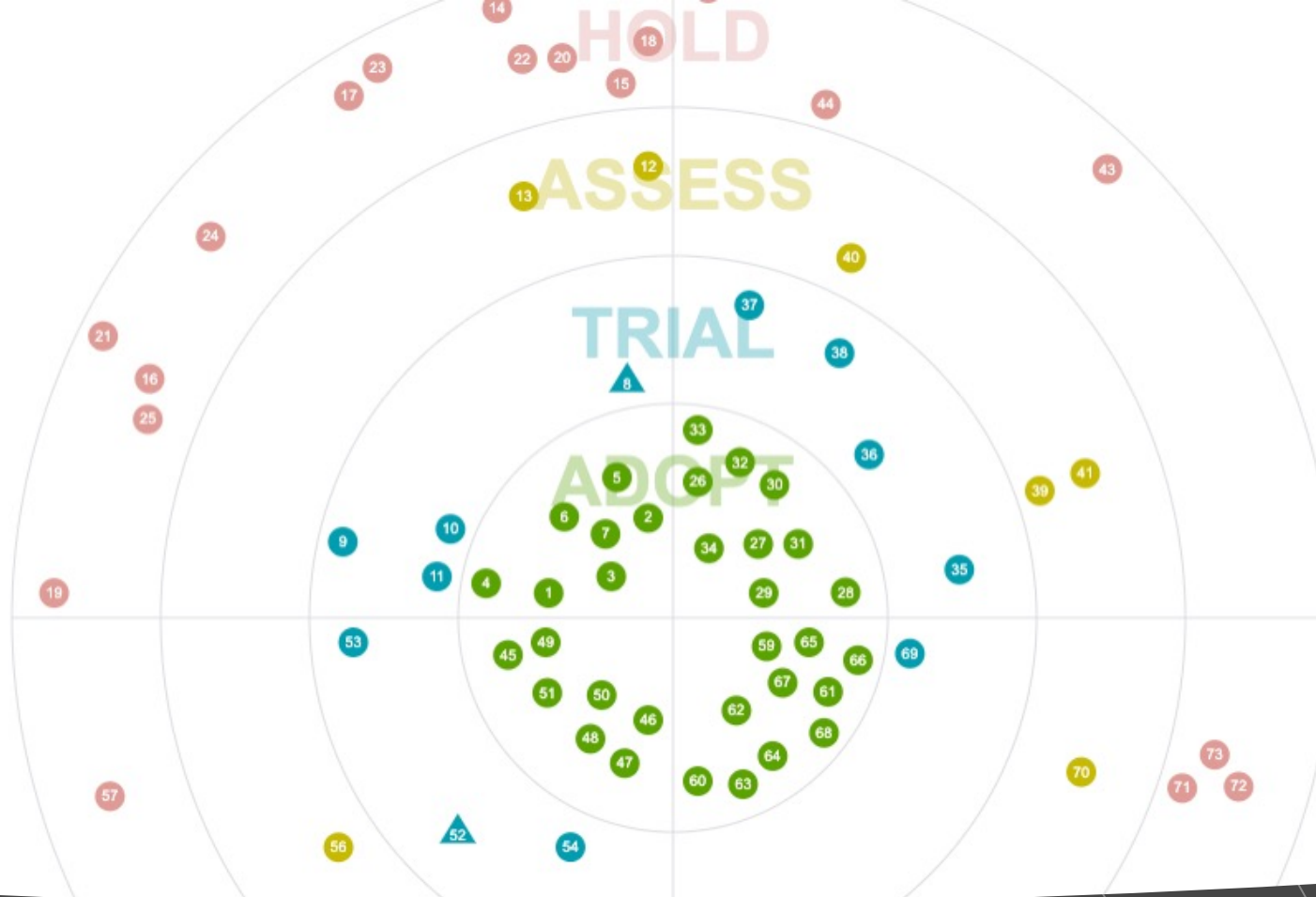
59. Go
60. GraphQL
61. Java
62. JavaScript
63. Kotlin
64. OpenAPI (Swagger)
65. Python

ASSESS

70. R

HOLD

71. Clojure
72. Haskell



Search engineering docs

Engineering About Communication Developer journey Guidelines Practices

Practices

Practices

API First

Capacity Planning

Continuous Delivery

Continuous Integration

Dependency Discovery

Experimentation

Feature Flags

Libraries

Migrations

Observability

Playbooks

Postmortems

Contribution models

Inner Source and Contribution Models

Open Source

Reviews

API Reviews

Architecture Reviews

Continuous Delivery

Continuous Delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be **reliably released at any time** and, when releasing the software, without doing so manually. It aims at building, testing, and releasing software with greater speed and frequency. The approach helps **reduce the cost, time, and risk** of delivering changes by allowing for more **incremental updates to applications in production**.¹

Continuous Delivery vs Continuous Integration

While Continuous Delivery is about releasing software, **Continuous Integration** is the necessary practice to frequently integrate changes from different contributors. Both practices are often just abbreviated as "CI/CD".

Zalando's **Continuous Delivery Platform (CDP)** provides the capabilities to implement Continuous Delivery:

- code is authored, committed, and pushed to a short-lived branch on GitHub Enterprise
- a Pull Request (PR) is opened to merge the short-lived branch to the main branch
- CDP runs automated tests and deploys the PR changes to a test/preview environment
- the PR is merged after verifying that changes are correct
- CDP automatically deploys the updated main branch to production

The above workflow should ensure that the main branch always reflects the latest state of what is deployed in production. Therefore, any new merged change is an incremental automated update to the application in production. This reduces risks and is the essence of Continuous Delivery.

Continuous Delivery vs Continuous Deployment

With Continuous Delivery, you can release changes at any time, but the trigger ("deploy to production") can be manual. With **Continuous Deployment**, you automatically release changes to production. Code changes are pushed live once they pass validation and pass all tests. This means customers



Table of contents

Good Practices

If it hurts, do it more often

Limit work-in-progress (WIP)

Measure Software Delivery Performance

Community Forums

Further Reading

Filters: Organizations (5/532) Repositories Event type (All)

Latest pipeline run	Time	Status	Action
master-85 master → 02119c	7m ago 3m, 51s	BUILD DEPLOY-TEST-STAGING DEPLOY-STAGING DEPLOY-LIVE	
pr-132-3 → db98df	8m ago 3m, 40s	BUILD STAGING-DEP... STAGING-DEP... STAGING-DEP... STAGING-DEP... LIVE-DEPLOY... LIVE-DEPLOY...	
SIZING / master-129 master → 182e29	19m ago	BUILD VISUAL-TEST DEPLOY-PR S-100 DEPLOY-LIVE L-100	
CEP / CEP/GH#268 moved pr-270-1 → 9f39e5	19m ago 2m, 33s	BUILD DEPLOY-STAG... TRAFFIC-100... DEPLOY-PROD... TRAFFIC-10-P... TRAFFIC-50-P... TRAFFIC-100...	
SIZING / Adjust pr-113-2 → af86fd	19m ago	TEST BUILD-PR BUILD-PR... DEPLOY... TRAFFIC... DEPLOY... PROD-TR... PROD-TR... PROD-TR...	
SIZING /	22m ago		

持續交付平台

■ 資料來源: Zalando

[AUTOMATED] Update cronjob API version #85

Merged merged 1 commit into master from :cronjob_migration on Oct 20, 2021

Conversation 2 Commits 1 Checks 0 Files changed 2

commented on Sep 10, 2021

THIS IS AN AUTOMATED PR FROM THE CLOUD INFRA TEAM

We are preparing to upgrade to Kubernetes 1.21. In order to have a frictionless upgrade we need to migrate the CronJobs that are still using the deprecated version `v2alpha1`.

So far we didn't find any upgrade issues and it's as easy as changing the version in the deployment file in your Github repository. We created this [doc](#) to give you more details and links to the external documentation.

This commit changes any reference of the `v2alpha1` to `v1beta1`. We kindly ask you to review and merge this PR as soon as possible. After the update to kubernetes 1.21, planned for until the end of September, you won't be able to deploy changes to these `CronJobs` without merging this PR (or cherry-picking its changes).

[AUTOMATED] Fix manifests for Kubernetes v1.16 #1773

Closed wants to merge 1 commit into master from fix-apidversions-kubernetes-v1.16

Conversation 2 Commits 1 Checks 0 Files changed 2

commented on Feb 6, 2020

THIS IS AN AUTOMATED PR FROM THE CLOUD INFRASTRUCTURE TEAM

This updates the `apiVersion` to make sure deployments from this repository will continue to work when the Zalando Kubernetes clusters are being updated to Kubernetes v1.16.

References

Docs: [howtos/update-api-versions/](#)

Announcement: <https://groups.google.com/>

Upstream Issue:

開發高度自動化
GitOps

■ 資料來源:Zalando

- Lead Time $\hat{=}$ **Commit to Prod**
- Release Frequency $\hat{=}$ **Deploys/week/dev**
- Time to Restore Service $\hat{=}$ **MTRS from incidents**
- Change Fail Rate $\hat{=}$ **Incidents/Deploys**

DORA指標衡量 DevOps績效

■ 資料來源:Zalando



請參考@3 用表格整理出平臺功能的重要特色和屬性



重要特色	特色簡介
Platform as a product	平台以產品形式存在，需要基於用戶需求進行設計和演進，並優先考慮支持大多數用例的通用能力
User experience	平台應通過一致的界面提供其能力並關注用戶體驗。平台應該在各種界面上滿足用戶的需求，例如 GUI、API、命令行工具、IDE 和門戶等
Documentation and onboarding	平台應該附帶適當的文檔，以滿足用戶對其提供的能力的需求。平台還應提供工具來加速新項目的啟動，以幫助用戶快速、簡單地使用所需的平台服務
Self-service	平台應該是自助服務的。用戶必須能夠自主自動地請求和接收能力
Reduced cognitive load for users	平台的一個重要目標是減少產品團隊的認知負荷。平台應該封裝實現細節，隱藏由其架構引起的任何複雜性。
Optional and composable	平台應該是可選擇且可組合的。它應該允許產品團隊僅使用其提供的部分能力，並在必要時在其平台外提供和管理自己的能力
Secure by default	平台應該是默認安全的，並提供符合組織定義的規則和標準的合規性和驗證能力



CNCF Platforms White Paper 平臺工程特性

1. 平臺產品化(Platform as a product)
2. 重視使用體驗/開發者體驗(User experience)
3. 豐富文件和新手指南(Documentation and onboarding)
4. 自助服務(Self-service)
5. 盡可能降低認知負載(Reduced cognitive load for users)
6. 提供選擇和組合能力(Optional and composable)
7. Secure by default思維

平臺團隊 基本任務

1. 要有平臺產品經理
2. 研究平臺使用者（開發者）的需求，規劃相關功能藍圖
3. 負責平臺價值（效益）的行銷、推廣、倡導行為和教育
4. 管理和開發使用者介面、可觀測性能力和相關服務，包括平臺入口網站、**API**、文件和範本、**CLI**工具

提醒！
SRE方法論不見
得能 完全套用

- SRE是一套從顧客導向思維來發展的網站維運方法論
- 外部顧客與內部顧客（員工）的角色大不相同
 - 例如找不到顧客，但看得到員工
- 規模和需求相差很大
- 不少SRE方法論，不見得全盤適用到內部員工，必須轉換思考角度才行

IDP 內部開發平台 指南

Internal Developer Platform

Search

Learn

What is an Internal Developer Platform (IDP)?

Why use an Internal Developer Platform (IDP)?

The 5 Core Components of an Internal Developer Platform (IDP) ▶

When do you need an Internal Developer Platform (IDP)?

How do Internal Developer Platforms (IDPs) relate to other concepts?

Ecosystem

Platform tooling to build your IDP ▶

PaaS and DevOps Platforms ▶

Community

Join

Internal Developer Platform

Everything the WWW has around Internal Developer Platforms in one curated space. It helps you understand the why, how, what and who.

A modern way to run engineering teams

While self-built IDPs have been around in elite teams for around 5 years, they're now going mainstream in 2021, embraced by huge enterprises like Spotify, Airbnb, and Zalando.



Kelsey Hightower 
@kelseyhightower · [Follow](#)



I'm convinced the majority of people managing infrastructure just want a PaaS. The only requirement: it has to be built by them.

7:08 AM · Apr 12, 2017 

 1.2K  Reply  Copy link

[Read 55 replies](#)

[-> What is an Internal Developer Platform \(IDP\)](#)

A blue speech bubble graphic containing the text 'IDP 五大核心'. The bubble has a tail pointing downwards. The background features faint, concentric circular lines.

IDP 五大核心

Core Component

Application Configuration Management

Short Description

Manage application configuration in a dynamic, scalable and reliable way.

Infrastructure Orchestration

Orchestrate your infrastructure in a dynamic and intelligent way depending on the context.

Environment Management

Enable developers to create new and fully provisioned environments whenever needed.

Deployment Management

Implement a delivery pipeline for Continuous Delivery or even Continuous Deployment (CD).

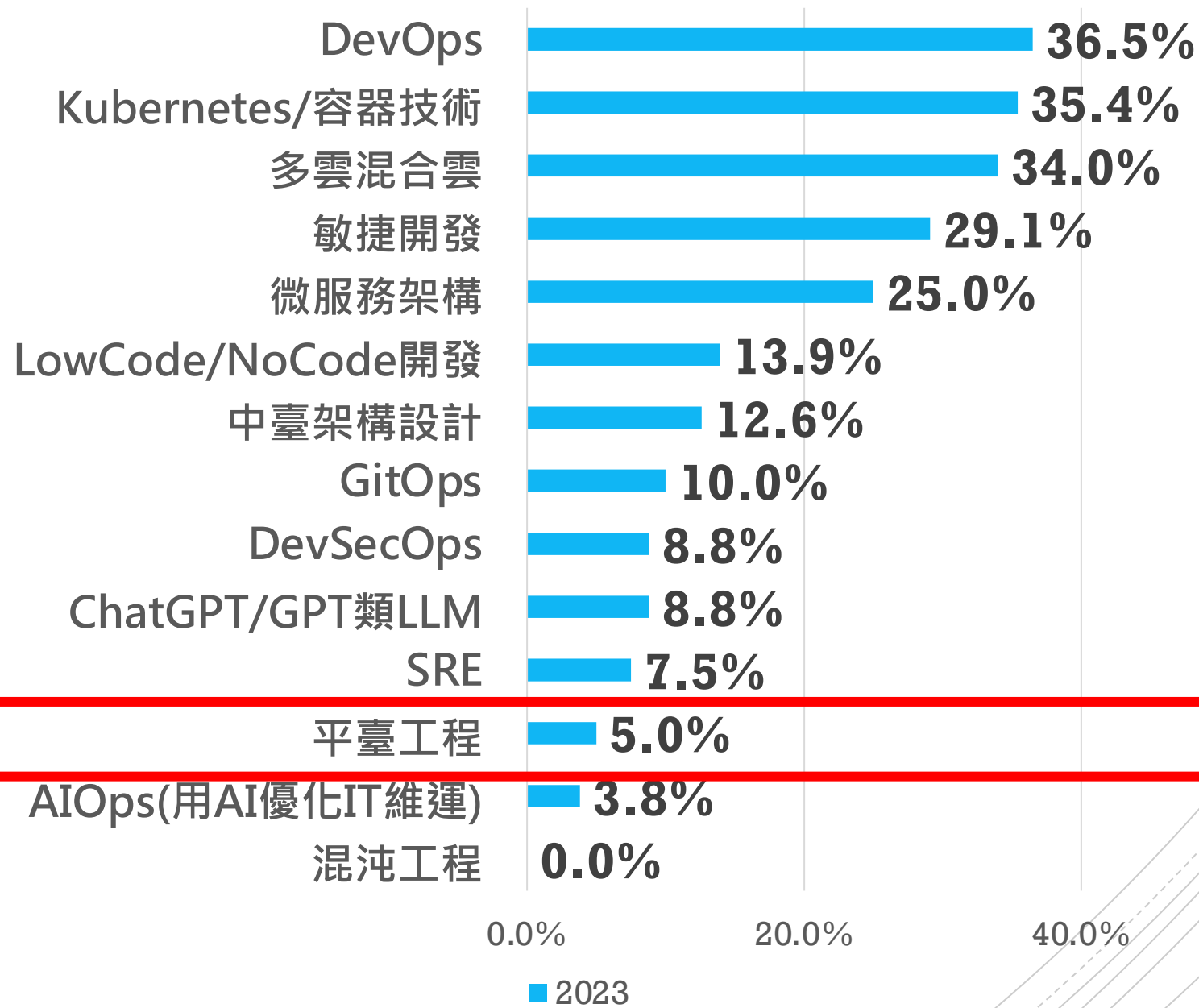
Role-Based Access Control

Manage who can do what in a scalable way.

平臺工程浪潮 已經吹進台灣

- 臺灣這兩年積極招募SRE人才的企業，大多不是直接提供2C端服務的產業
- 這些企業需要SRE人才的雲原生技術能力，來服務內部DevOps團隊
- 平臺工程浪潮早已吹進臺灣IT現代化
- 雲原生技術採用潮，會更帶動企業內部IT，開發團隊擁抱DevOps
- 維運團隊也要開始擁抱現代化，平臺工程正是邁向維運現代化的關鍵思維。

2023
108位CIO
平台工程/SRE
採用趨勢



謝謝大家！