

# 綠色軟體

# 參考手冊 草案

Green Software Reference  
Handbook Draft



# Contents 目錄

## 04 編序

- 04 緣起
- 05 發展綠色軟體的目的與效益
- 07 手冊適用參考對象

## 08 第一章 綠色軟體發展與背景

- 09 第一節、綠色軟體的國際定義
- 09 第二節、綠色軟體國際相關發展及政策
- 11 第三節、綠色軟體的基本構件

## 15 第二章 綠色軟體的核心原則

- 16 第一節、能源效率 (Energy Efficiency)
- 17 第二節、硬體效率 (Hardware Efficiency)
- 19 第三節、碳意識 (Carbon Awareness)

## 21 第三章 綠色軟體與 AI 應用

- 22 第一節、綠色軟體在網路永續的角色
- 22 第二節、更永續的機器學習

## 26 第四章 軟體碳排放評估方法

- 27 第一節、國際標準 ISO/IEC 21031:2024 Software Carbon Intensity (SCI)
- 30 第二節、軟體的碳排放測量
- 33 第三節、軟體碳排放檢測（體驗）工具介紹（試算範例內容）

## 41 第五章 綠色軟體應用情境

- 42 第一節、國際綠色軟體相關參考制度介紹
- 45 第二節、國內綠色軟體應用場景 - 自願性綠色軟體標章制度構想

## 49 附錄一、專家諮詢與編輯團隊

## 51 附錄二、參考文獻

## 表目錄

- 45 表 1、同類型軟體比較範例
- 45 表 2、不同類型軟體比較
- 49 表 3、專家及學者感謝名單
- 50 表 4、台灣經濟研究院編輯團隊
- 50 表 5、協助本案意見蒐集與訪談會議之企業名單

## 圖目錄

- 16 圖 1、綠色軟體的原則
- 17 圖 2、能耗與利用率關係圖
- 18 圖 3、延長硬體壽命示意圖
- 18 圖 4、提高伺服器利用率示意圖
- 19 圖 5、空間轉移
- 20 圖 6、時間轉移
- 20 圖 7、需求塑造
- 29 圖 8、軟體碳強度計算標準
- 34 圖 9、以 CPU 運算為例
- 35 圖 10、Windows 軟體碳排放強度計算機 (首頁)
- 35 圖 11、Windows 軟體碳排放強度計算機 (檢測設定)
- 36 圖 12、Windows 軟體碳排放強度計算機 (檢測畫面)
- 36 圖 13、Windows 軟體碳排放強度計算機 (檢測完成)
- 37 圖 14、Windows 軟體碳排放強度計算機 (檢測報告)
- 37 圖 15、Windows 軟體碳排放強度計算機 (設定畫面)
- 38 圖 16、Windows 軟體碳排放強度計算機 (計算結果畫面)
- 39 圖 17、iOS 軟體碳排放強度計算機 (首頁)
- 39 圖 18、iOS 軟體碳排放強度計算機 (檢測畫面)
- 39 圖 19、iOS 軟體碳排放強度計算機 (檢測報告)
- 40 圖 20、Android 軟體碳排放強度計算機 (首頁)
- 40 圖 21、Android 軟體碳排放強度計算機 (測試畫面)
- 40 圖 22、iOS 軟體碳排放強度計算機 (檢測報告)
- 42 圖 23、德國藍天使「資源與能源高效的軟體產品」標章
- 44 圖 24、評級範例流程圖

# 編序 Preface

## 緣起

隨著資料中心耗電量急遽攀升，人們開始反思資訊科技在碳排放中的角色。早期的「綠色 IT」(Green IT) 運動多著重在硬體設備的節能，例如伺服器虛擬化、散熱管理與能源效率優化等；而軟體，長期被視為「無形」的，不被直接歸類為能源消耗者，直到雲端計算與 AI 興起，才逐漸顯露其龐大的間接環境影響。

當代數位科技在協助各領域及產業應對氣候變遷與降低碳排方面，正扮演日益關鍵的角色。自 1970 年代以來，硬體的處理效能約每兩年翻倍，意味著同樣的運算任務能以更快速度完成、消耗更少能源。然而，這類效率的提升在軟體層面卻未能同步實現。相反地，隨著軟體功能與資料需求不斷擴張，原本來自硬體的效率紅利反而被軟體本身愈趨龐大的資源需求所抵銷，進一步導致「反彈效應」(Rebound Effect)：即使單位運算變得更有效率，但因使用頻率與功能需求提升，整體資源消耗反而上升。

這種現象在「軟體膨脹」(Software Bloat) 中尤為明顯，例如新版本作業系統推出時，常因硬體需求提升，迫使用戶汰換原本仍可正常使用的裝置。再如，在處理相同任務時，不同軟體之間的能源與資源使用效率可能相差甚遠，一款文字處理軟體的耗能可能是另一款的四倍之多。這凸顯了軟體設計對資通訊科技整體能源效率的關鍵影響，因為軟體的架構與運作方式直接決定所需硬體規格與實際能源消耗。

### 一、綠色數位行動宣言 (Declaration on Green Digital Action)

COP29 在《綠色數位行動宣言》(Declaration on Green Digital Action) 中提到，建立一致性的碳排放計算標準的重要性。透過更嚴謹且完整的技術數據進行評估，將可有效提升對資通訊技術解決方案在溫室氣體排放與能源消耗方面的影響準確度，進而制定出更具效益的氣候行動目標。

## 二、綠色軟體基金會 (Green Software Foundation，以下簡稱 GSF)

綠色軟體基金會 (GSF) 於 2021 年提出綠色軟體的概念，強調軟體在全生命週期中應兼顧資源使用效率與碳排放管理，並倡導將環境永續納入軟體開發、部署與運行的每一階段。綠色軟體基金會 GSF 認為，軟體雖非實體產品，卻在運算、儲存與資料傳輸過程中消耗大量能源，其間接碳足跡不容忽視。因此，透過可量測、可追蹤、可改善的技術方法，推動軟體產業轉向低碳轉型，不僅有助於降低全球資訊與通信技術 (ICT) 領域的整體碳排放，也成為實現淨零目標的重要一環。

## ■ 發展綠色軟體的目的與效益

在現代雲端與數位服務環境中，綠色軟體不只是「減碳」的技術選項，更是一種讓系統運作更聰明、更具成本效益的思維。綠色軟體是強調「以更少的資源，達成更高的效能」。從雲端原生架構的採用，到具自動修復能力的彈性設計，綠色軟體將展現如何在節能、省成本的同時，仍能維持穩定、安全與高效的系統運作。這樣的設計不僅符合永續原則，也讓軟體產品更具競爭力。以下是摘自美國 O'Reilly Media 公司於 2024 年出版的《Building Green Software》，說明發展綠色軟體能夠同時達成的多重效益：

### 一、具備成本效益的綠色軟體

具成本效益的工作負載是指能以最少支出滿足業務需求的運作模式。實現這一點的關鍵，在於選擇最適合實際需求的資源，避免過度配置。這不僅符合綠色軟體的核心原則（關注能源消耗），也符合降低成本的考量。舉例來說，假設我們使用 Google 的雲端平台，那同時也應採用 Google 的雲端原生解決方案 (cloud-native solutions)，取代自行重新建置、部署與維護系統。因為雲端原生解決方案 (cloud-native solutions) 已針對該平台進行優化，能幫助系統更有效控制營運支出。相對地，如果我們選擇自行部署，不僅提高人力成本、增加維護負擔，也無法充分發揮平台效能，甚至可能導致更高的碳排放。

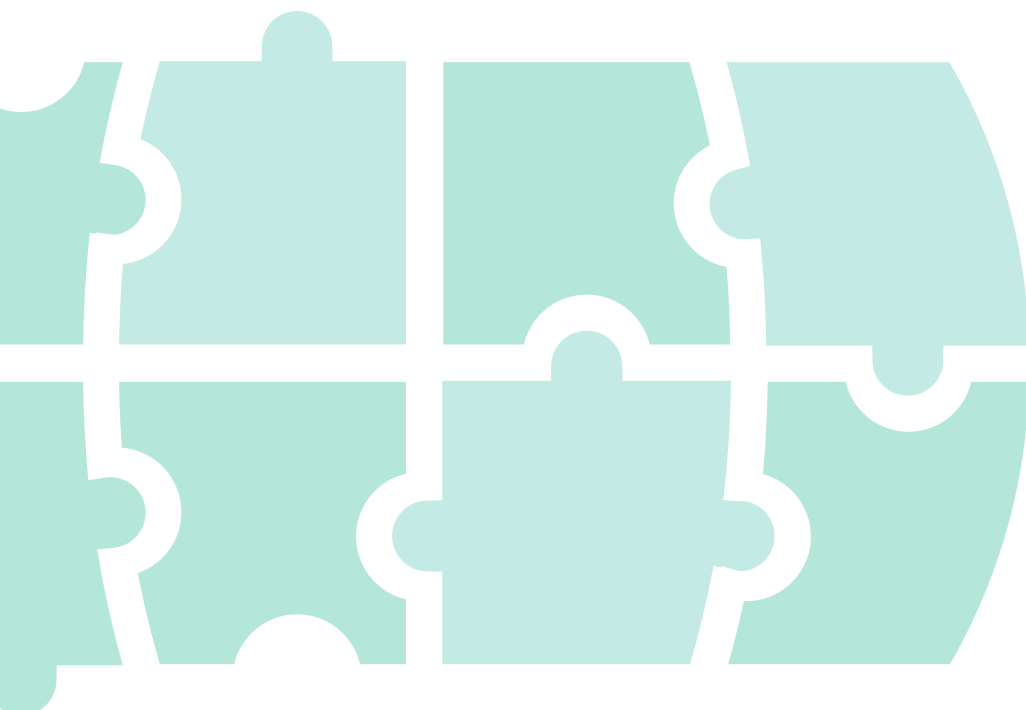
## 二、綠色軟體同時確保系統的可靠性與彈性

可靠性指的是系統在面對故障與錯誤時，仍能維持正常運作的能力，通常以百分比表示。例如，若系統設計要求在 3 毫秒內回應服務請求，而在 99% 的情況下皆能達成，則該系統的可靠性為 99%。彈性則是指系統在發生故障或中斷時，能夠快速恢復運作的能力，強調系統面對不可預期情況時的自我修復能力。

目前提升系統可靠性的常見方式是採用系統冗餘 (redundancy) 機制。當主要服務或應用異常終止時，立即啟用備援的伺服器、系統、硬體或網路資源，以達成故障轉移 (failover)。然而，維護這些備援副本需要使用到硬體與電力的資源，且備份系統在其整個生命週期中大多處於閒置狀態，因此也不可避免地造成資源浪費。

相較之下，綠色軟體較傾向採取的彈性設計，是一種更現代、動態且高效率的解決方案，能有效降低硬體資源的預留需求。彈性架構強調自動化與程式化的營運技術，結合 DevOps 與 SRE 工程常見的快速反應機制，例如：錯誤監控、自動擴展 (autoscaling) 與自動重啟等設計，目的是在問題發生時即時修復，而非依賴靜態的備援系統。在軟體開發與設計階段即導入綠色軟體的概念，能有效打造出具備彈性、永續性且更可靠的綠色系統。

以上內容透過實例，簡要說明發展綠色軟體的目的與其可能帶來的效益。後續章節將進一步闡述綠色軟體的核心概念與設計原則，協助讀者對綠色軟體的內涵與實踐方式有更全面的理解。



## ■ 手冊適用參考對象

本手冊撰寫的目的主要是為了提供直接或間接參與軟體設計、部署，以及使用軟體的人參考。根據 Google 推出的《Go Green Software》<sup>1</sup>，綠色軟體工程的流程通常涉及多種角色，包括產品經理、軟體架構師、開發人員、測試與品保工程師、資料架構師、運維與平台工程師，以及使用者體驗設計師等。這些角色各自於軟體生命週期中扮演不同但相互關聯的任務，若能協同合作，將能有效提升軟體在效能、可靠度與永續性間的整體平衡。

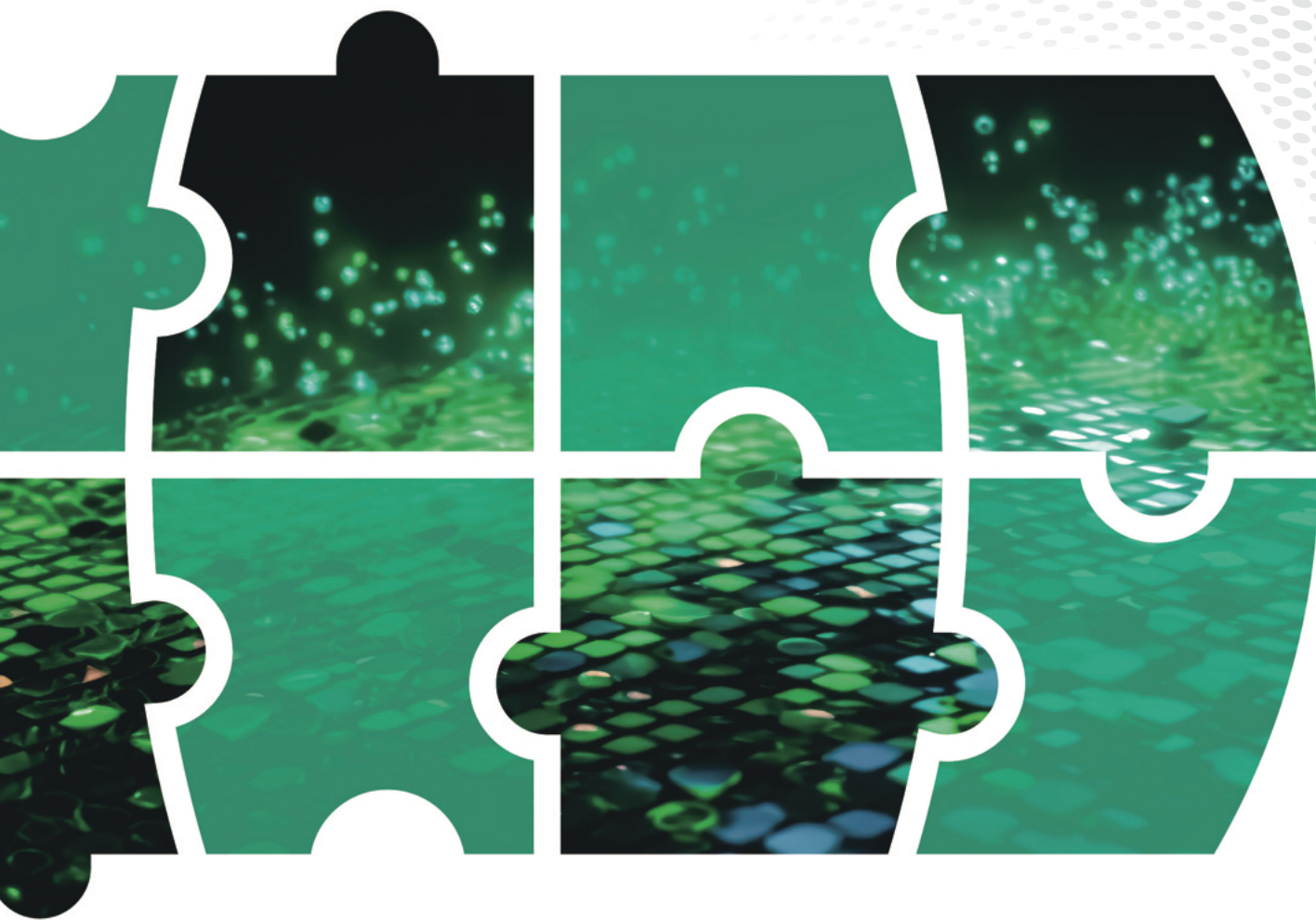
例如，產品經理應在產品的定義階段就先考慮到永續的需求；軟體架構師與開發人員則負責設計並實作高效率、低能耗的系統架構與程式碼；測試與品保工程師則需在驗證階段評估軟體執行時的能源效率；資料架構師與平台工程師可透過資料壓縮、雲端資源自動化擴展與關機機制，優化系統能源使用；而使用者體驗設計師則可藉由減少不必要的互動與運算，間接降低碳排放。唯有各角色在開發過程中共同導入綠色設計觀念，綠色軟體的概念方能真正融入並貫徹於整體工程流程之中。

本手冊作為推動綠色軟體導入與實踐之初步導引文件，旨在推廣綠色軟體理念並建立基礎認知，協助相關從業人員初步了解各角色於綠色軟體開發流程中的定位與職責。後續關於實務操作與技術應用之具體指引，未來將會持續補充與完善，以進行更深入的探討。

另須特別提醒，手冊的內容兼具概念性與技術性，其中部分章節（例如第三、第四及第五章）偏重實務操作與專業引導，主要供相關領域人員參考；一般讀者可先閱讀前述概念性章節，以建立基本認知。至於第五章有關綠色軟體應用情境的介紹，則歡迎所有對我國發展綠色軟體產品有興趣的讀者參閱。

---

<sup>1</sup> 資料來源：Google Cloud (2023). Go Green Software.



## 第一章 Chapter 1

# 綠色軟體發展與背景

在數位技術與永續發展日益交織的現在，綠色軟體逐漸成為全球關注的重要議題，「綠色軟體基金會」(Green Software Foundation，以下簡稱 GSF) 也隨之成立。GSF 由 Microsoft、GitHub、Accenture 與 ThoughtWorks 等全球領導企業於 2021 年共同創立，旨在打造一個可信賴的綠色軟體生態系，涵蓋專業人才、標準規範與技術工具等面向，並強調其在實務上的可實踐性。GSF 與 Linux 基金會亦合作推出綠色軟體從業人員的專業課程、認證考試與證照，進行相關技能培育與專業認證，全球已有二萬多人通過考試並取得證照<sup>2</sup>。

<sup>2</sup> 資料來源：Linux Foundation (2024). Green Software for Practitioners (LFC131).

## 第一節、綠色軟體的國際定義

綠色軟體基金會 (GSF) 指出，軟體不僅是氣候變遷問題的一部分，同時也能成為解決氣候問題的關鍵工具。綠色軟體的核心理念在於透過減少軟體本身所造成的碳排放，來降低其對氣候的負面影響。基此，綠色軟體主要工作有二：一是開發過程中的減碳，二是作為促進其他領域減碳的數位工具。綠色軟體的核心目標是減少碳排放，而非僅以碳中和作為最終目標，因此需要投入更多資源與努力。目前，降低軟體碳排放的主要策略聚焦於三個關鍵方向：使用更少的運算資源、消耗更少的能源，以及更智慧地利用能源。透過這些實際行動，綠色軟體將成為實現永續數位轉型的重要推手。

根據綠色軟體基金會 (GSF) 的定義<sup>3</sup>，綠色軟體是一門融合氣候科學、軟體設計、電力市場、硬體工程與資料中心設計等領域的新興跨學科領域。所謂綠色軟體，指的是具備高碳效率的軟體，亦即在開發與運行過程中能盡可能地減少碳排放。

## 第二節、綠色軟體國際相關發展及政策

全球多個政府與國際組織亦已紛紛投入綠色軟體相關政策與標準的制定與推動，展現出對永續數位轉型的高度重視。隨著氣候變遷議題日益受到關注，各國也開始從不同面向切入，積極發展因地制宜的綠色軟體策略。以下將介紹各國在此領域的具體作法與進展。

### 一、各國政府推動綠色軟體相關政策

#### ① 法國

法國於 2021 年發布「法國減少數位環境足跡 (Reducing the environmental footprint of digital technology in France)」法案，以法制方式規範數位科技對環境所造成的影響。其中，第 26 條明確規定，凡提供電視服務、隨選視聽媒體服務及影音分享平臺者，均須揭露其能源消耗與溫室氣體排放數據。此一規範不僅提升資訊服務業的環境資訊透明度，也促進軟體產品碳排放可計算化的需求，進而引導業者朝向低碳軟體研發與生態設計方向發展<sup>4</sup>。

<sup>3</sup> 資料來源：Green Software Foundation (2024). Green Software Introduction.

<sup>4</sup> 資料來源：French Government (2023). Environmental Legislation and Green IT.

## 二 新加坡

新加坡資訊媒體發展局 (IMDA) 也在 2023 年在「數位連結性藍圖 (Digital Connectivity Blueprint, DCB)」中強調將高效程式設計納入軟體開發的原則，希望藉由綠色軟體建構永續的軟體生態系，減緩日益增加的運算需求。作為全球推動綠色軟體的主要國際機構，該局於 2024 年主辦首屆的綠色軟體基金會全球峰會，進行關於綠色軟體如何推動綠色數位的討論，並發布「永續軟體開發指引 (Sustainable Software Development Guidelines)」，該指引結合氣候科學、軟體設計、潔淨能源與數位基礎設施等領域的議題，旨在打造具永續性與碳感知能力的應用程式，響應新加坡《2030 綠色計畫》的核心目標，應可借鏡規劃符應我國淨零目標的綠色軟體相關制度框架<sup>5</sup>。

## 三 西班牙

西班牙政府為促進綠色軟體的永續發展，於 2022 年通過「國家綠色演算法計畫 (National Green Algorithms Plan)」，發展「綠色設計」的人工智慧 (AI)，期望藉由演算法的設計與開發初期就考慮環境永續性，以實現數位創新與綠色轉型的雙重目標。相關推動策略包含研訂相關標準與工具來衡量演算法的能源消耗，使 AI 的開發人員意識到其技術選擇對環境的影響，建立一套自願性的實踐制度；設立「綠色科技品質標章 (Green Tech Quality Seal)」，用以表彰符合環境永續標準的 AI 模型與方案<sup>6</sup>。

## 二、國外資服產業在綠色技術與能源效率方面的發展現況

### 一 Gartner

美國標準普爾指數 500 的高德納諮詢公司 (Gartner) 業於 2024 年公布的五大策略性技術趨勢中，將「綠色軟體工程」列為重點之一，強調打造具備碳效率與碳意識的軟體工程學科，作為推動永續發展的重要方向<sup>7</sup>。

### 二 Microsoft

微軟為協助開發者在雲端平臺 Azure 上實現永續軟體設計，於其 Azure Well-Architected Framework 中整合了綠色軟體原則與操作指引，並與綠色軟體

<sup>5</sup> 資料來源：IMDA (2024). Sustainable Software Development Guidelines.

<sup>6</sup> 資料來源：Spanish Government (2025). Algoritmos Verdes.

<sup>7</sup> 資料來源：Gartner (2024). Top Five Strategic Technology Trends in Software Engineering for 2024.

基金會 (Green Software Foundation, GSF) 合作推動碳效率標準。微軟提出的永續設計方法論，強調從應用程式的業務需求出發，針對應用設計、平臺選擇、部署操作、儲存與網路架構等面向進行整體性評估，同時搭配碳排放量測工具 (如 Emissions Impact Dashboard)，以追蹤與改善軟體的碳足跡，與先前提到綠色軟體基金會 (GSF) 提出的六大原則一致<sup>8</sup>。

### 三 Google

Google 指出軟體產業在能源與資源消耗方面扮演著舉足輕重的角色，基於此，Google 推出了「環保永續軟體規範指南」，針對軟體工程流程中涉及的多種角色，包括產品經理、產品設計師、系統架構師、軟體開發人員與工程師、DevOps 工程師、資料架構師、資料工程師以及數據科學家等，提供具體可行的環保永續開發原則與實務建議。該指南旨在協助從業人員在軟體開發流程中落實能源效率、降低碳排放，作為推動環境永續的技術參考依據，進而引導軟體產業朝向更具永續性的發展方向邁進<sup>9</sup>。

### 四 NTT Data

日本最大資訊系統整合公司 NTT Data，作為全球 IT 產業排名前 10 的服務供應商，推出全新的綠色功能即服務平臺 (Green Function as a Service, FaaS)，提供軟體碳排強度的即時監控與計算，且支援 Information technology — Software Carbon Intensity (SCI) specification (ISO/IEC 21031:2024) 的計算方法，在業界實務上有很大的參考價值<sup>10</sup>。

## 第三節、綠色軟體的基本構件

### 一、碳排放

碳排放是指人類因為從事某件活動，直接或間接產生的溫室氣體，這些溫室氣體的總重量就被稱為碳排放量。所謂的溫室氣體，是指容易吸收太陽輻射的氣體，包括人類主要排放的溫室氣體種類包括二氧化碳 (CO<sub>2</sub>)、甲烷 (CH<sub>4</sub>)、氧化亞氮 (N<sub>2</sub>O)、氫

<sup>8</sup> 資料來源：Microsoft (2024). Sustainable Software Engineering Practices in Azure Kubernetes Service.

<sup>9</sup> 資料來源：Google Cloud (2023). Go Green Software.

<sup>10</sup> 資料來源：ISO/IEC (2024). Information technology — Software Carbon Intensity (SCI) specification (ISO/IEC 21031:2024).

氟碳化物 (HFC<sub>s</sub>)、全氟碳化物 (PFC<sub>s</sub>)、六氟化硫 (SF<sub>6</sub>)、三氟化氮 (NF<sub>3</sub>) 等<sup>11</sup>。在這些溫室氣體當中，二氧化碳是人類活動對氣候變遷最大的影響，因為二氧化碳最主要的排放源就是來自燃燒化石燃料產生電力的過程，例如煤炭、石油與天然氣等這些能源的使用<sup>12</sup>。雖然其他的溫室氣體，例如甲烷，可能對全球暖化的影響更顯著，然而二氧化碳是大氣中最常見、濃度最高的溫室氣體，同時也是與電力生產與科技產業關係最密切的溫室氣體種類。因為科技業大量依賴電力運作，如資料中心、伺服器與各類雲端服務，皆間接導致二氧化碳的大量排放。

## 二、電力

所有軟體從業人員都應該對其產品最終所消耗的能源負起責任，並致力於在開發與營運的每一個階段盡可能減少資源浪費，這是本手冊需要呼籲得重點之一。電力是一種能量形式；依據發電的來源不同，可分為初級電力及次級電力，我們這邊主要探討的是次級電力<sup>13</sup>。

### 一 高低碳能源

次級電力可由化石燃料或再生能源轉變而來，以燃煤電廠為例，煤炭在鍋爐中燃燒產生蒸汽，進而驅動渦輪機發電，這一過程將化學能轉換為電能，同時也會產生大量二氧化碳作為副產品。我們將這類透過燃燒化石燃料所產生的電力稱為高碳電力。在各種化石燃料中，煤炭的碳排放密度最高，高於石油與天然氣。相對地，像水力與風力這類低碳技術，在將動能轉換為電能的過程中，幾乎不會產生碳排放。

### 二 軟體在能源使用效率的角色

由於電力消耗間接導致碳排放，因此軟體的能源使用方式攸關其碳足跡表現，包含用電量、用電時間與地點等。假設採取智慧的用電策略，如提升低碳電力的使用，或是在高碳電力時段調整作業排程，將有助於電力能源去碳化的進程。此點將於第二章進行更詳細的討論。

筆記型電腦在閒置時仍會消耗電力，這稱為靜態功耗 (Static power draw)，意指設備在未執行任何操作時所產生的電力消耗。幾乎所有電子設備在閒置狀態

<sup>11</sup> 資料來源：臺灣碳權交易所股份有限公司 (2024)。【碳知識】碳名詞，你知道幾個？

<sup>12</sup> 資料來源：U.S. Environmental Protection Agency. Global Greenhouse Gas Overview (2025).

<sup>13</sup> 資料來源：經濟部能源署能源統計專區，能源統計手冊 113 年版。

下仍會持續耗電，且其耗電量會因裝置設定與硬體組成而有所差異。因此，大多數終端設備（如筆電與手機）通常具備節能模式的設計。

然而，節能模式雖有助於降低能耗，卻也存在使用上的取捨，例如裝置喚醒時間較長。這顯示在軟體設計上，往往需要在效能與能源效率之間取得平衡。一般而言，設備的運作效率越高，對環境的整體效益也越正向；但由於多數硬體的功耗與使用率並不成正比，因此我們應更加重視應用程式在能源效率方面的設計與優化。

### 三、硬體的隱含碳排

碳足跡無所不在，即使是關機狀態的新手機也都有隱含的碳排放。隱含碳 (Embodied carbon) 指的是設備在生產、製造及其使用壽命結束後的處理過程中所產生的碳排放。不同終端用戶設備的隱含碳差異頗大，某些設備在製造過程中的碳排放量遠高於使用過程中的碳排放量。

Apple 在 2022 年 9 月的 iPhone14s 的生命週期碳足跡報告中指出<sup>14</sup>，iPhone 手機有 82% 的碳排放發生在生產、運輸及報廢的處理過程，僅有 18% 是手機使用產生。換言之，對於終端用戶設備的軟體開發人員來說，應該盡量確保應用成是在每次升級時保持向後硬體的相容性。

由於隱含排放與軟體實際的能源使用較無直接關聯，往往也容易被忽略。以下簡要的介紹幾種主要的硬體類別，協助讀者初步理解各類硬體在碳足跡中所扮演的角色與影響。

#### 一 Central Processing Unit (以下簡稱 CPU)

CPU 是每一台計算設備中最關鍵的元件，可以稱為電腦的「大腦」。簡單來說，CPU 負責執行邏輯與數學運算，幾乎處理所有的計算作業與操作。而作為硬體設備，CPU 的運作需要電力，這不僅涉及日常的用電成本，也隱含生產與汰換過程中的環境成本。

對於資料中心的 CPU，功耗是碳排放的主要來源，現在的 CPU 設計會根據工作負載而調整電力的使用，以達到更高的效率。此外，大多數的 CPU 也具備電

---

<sup>14</sup> 資料來源：Apple Inc. (2022, September). iPhone 14 Product Environmental Report. Retrieved from

源管理機制，當處於閒置狀態時會進入低功耗模式，以減少能源浪費。現代 CPU 通常配備多核心架構，使開發者能夠執行多線程應用程式，進一步提升系統資源的利用率。這是一種有效但相對複雜的優化方式。

## 二 記憶體 (Memory)

記憶體 (Memory) 是電腦用來暫存或永久保存資料的裝置，讓 CPU 能快速存取所需資訊。記憶體主要分為隨機存取記憶體 (RAM) 與唯讀記憶體 (ROM) 兩種類型，RAM<sup>15</sup> 就像裝置的短期記憶體一樣，應用程式和檔案會從速度較慢的磁碟載入到 RAM 中，以便系統能更快速地存取它們；ROM<sup>16</sup> 是電腦中用來儲存無法被更改的程式或資料的記憶體。由於其硬體電路設計為固定格式，資料僅能讀取，無法寫入或修改，且在電源中斷後內容依然得以保留。整體而言，記憶體的容量與速度直接影響系統效能與反應速度。

## 三 磁碟儲存裝置 (Storage)

硬碟儲存裝置 (Storage) 負責保存作業系統、應用程式與用戶資料，主要有傳統硬碟 (HDD)、固態硬碟 (SSD)、網路附加儲存 (NAS) 三種類型。磁碟的作用與記憶體類似，都是資料儲存設備，但磁碟偏向永久保存。

總體而言，CPU、記憶體與磁碟這三大硬體元件在綠色軟體的碳足跡評估中的關注重點略有不同，特別是在軟體相容性與硬體壽命方面。例如，軟體對記憶體的需求變化相對緩慢，雖然 RAM 偶爾需要更新，但相較於 CPU 更容易解決，且 CPU 通常比記憶體更容易需要被汰換，從而導致更多的隱含碳排放。而磁碟在面對軟體升級時較不容易過時，因此其碳足跡的變動性相對較小。目前多數設備廠商在環保策略上多採使用環保材料或提升能源效率的措施，然而我們應該對軟體如何使用硬體有更多的關注。

---

<sup>15</sup> 資料來源：Microsoft. (n.d.). 關於電腦記憶體. Microsoft 支援.

<sup>16</sup> 資料來源：教育雲線上字典. (n.d.). 唯讀記憶體. 教育百科.



## 第二章 Chapter 2

### 綠色軟體的核心原則<sup>17</sup>

本章內容多數摘錄自綠色軟體基金會 (Green Software Foundation) 官方網站所發布之公開資料，並經編輯調整以利閱讀與在地理解。相關概念與內容之著作權屬原基金會所有，特此說明並致謝。

綠色軟體的核心包含能源效率 (Energy Efficiency)、硬體效率 (Hardware Efficiency)、碳意識 (Carbon Awareness) 三項原則。

<sup>17</sup> 資料來源：Green Software Foundation (2024). Green Software Introduction.



圖 1、綠色軟體的原則

資料來源：Green Software Foundation (2024). Green Software Introduction.

## 第一節、能源效率 (Energy Efficiency)

第一章我們提到，電力是從另一種能量類型轉換而來的二次能源，因此我們可以將用電量視為能源耗用衡量標準。提升軟體在執行時的能源使用效率，盡可能降低電力消耗是最有效的方法之一，但我們不該僅限於此。

所有軟體，從手機上運行的應用程式到資料中心運行的機器學習模型的訓練，都會消耗電力。因此綠色軟體的從業人員需對其產品的能源消耗負責，並致力於將其設計得盡可能降低能源消耗，確保在流程的每個環節都盡可能減少能源浪費。

能源消耗整個歷程的最後階段是產品的終端用戶，而非產品本身。這意味著我們的目標不僅是編寫最節能的程式碼或「最環保」的軟體，還要考慮最終用戶的感受，以及如何確保他們不會造成不必要的排放。包括將多項工作任務採批次方式進行處理，以提升能源比例效益，或調整使用者操作軟體的方式。

### 一、能量比例理論

能量比例是 Google 的工程師於 2007 年首次提出的理論，用以衡量電腦實際消耗電力與其執行有效工作速率的關係（即利用率）。利用率是指電腦資源的使用度，通常以百分比表示。當計算器處於滿載狀態時，利用率較高；反之，若處於閒置狀態，則利用率偏低。

然而，功耗與利用率並非成正比關係。例如，當利用率為 0% 時，電腦功耗為 100 瓦；當利用率提升至 50%，功耗為 180 瓦；當達到 100% 利用率時，功耗為 200 瓦。

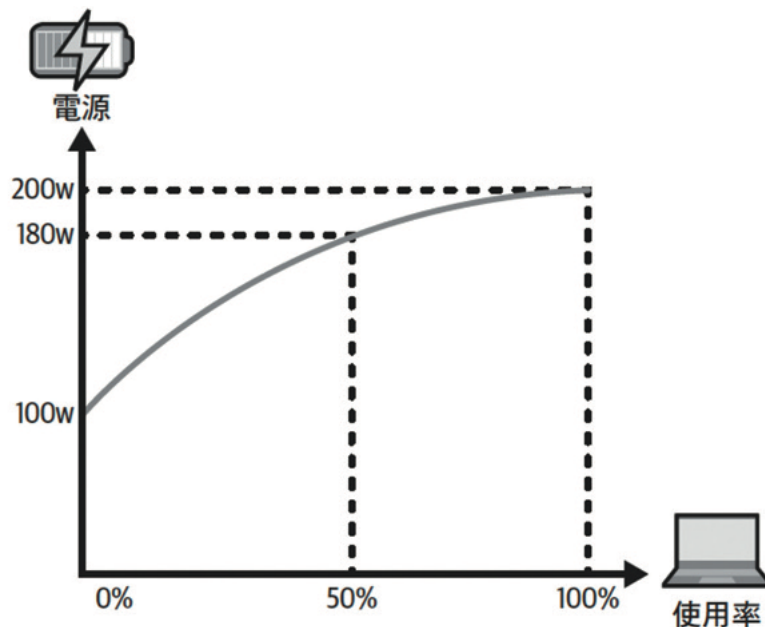


圖 2、能耗與利用率關係圖

資料來源：O'Reilly Media, Inc. Build Green Software. (2024). <https://reurl.cc/9nv720>

因此，我們應盡可能將工作集中於單一或較少的伺服器上運行，並讓伺服器以最高的利用率運行，從而最大限度地提高能源效率。

## ■ 第二節、硬體效率 (Hardware Efficiency)

在軟體開發過程中，所使用的硬體設備是一項不可忽視的重要因素，特別是其所帶來的隱性成本—隱含碳（此點在第一章已有進行說明）。提高硬體效率的主要方法有兩種，一是針對終端用戶設備，透過延長硬體使用壽命以減少頻繁汰換所帶來的隱含碳排放；二是針對雲端運算環境，則可透過提升硬體資源的使用率，以達到更高的能源與資源使用效率。

### 一、延長使用壽命

計算隱含碳排放的一種常見方法，是將設備在生產過程中產生的碳排放，平均攤銷至其預期使用壽命期間。例如，若製造一台伺服器會產生 4,000 公斤的 CO<sub>2</sub>eq，而

其預期使用年限為四年，則可推估其每年隱含碳排放為 1,000 公斤 CO<sub>2</sub>eq。若能將使用壽命再延長一年，那麼每年隱含碳排放就會下降到每年 800 公斤 CO<sub>2</sub>eq。

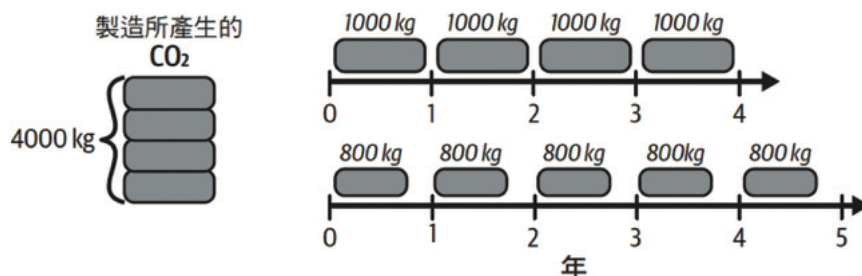


圖 3、延長硬體壽命示意圖

資料來源：O'Reilly Media, Inc. Build Green Software. (2024). <https://reurl.cc/9nv720>

## 二、提高設備利用率

在雲端運算領域中，提升硬體效率通常是提高伺服器利用率。考量到隱含碳排放的成本，以一台利用率達 100% 的伺服器支撐運算需求，遠優於同樣工作量下使用五台利用率僅 20% 的伺服器。換句話說，充分發揮伺服器的運算容量，不僅能維持相同的營運效能，還能有效降低整體硬體投入所帶來的隱性碳足跡。

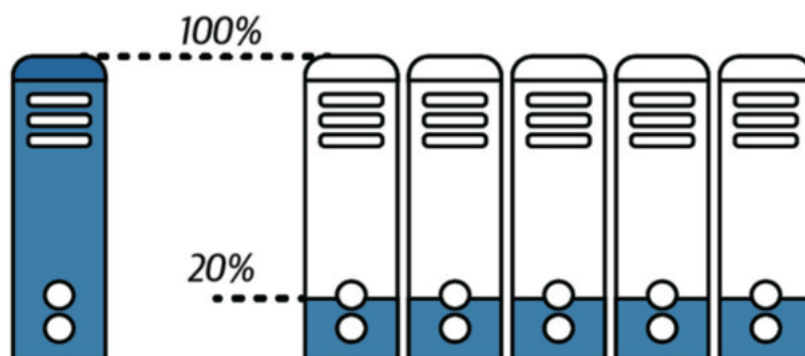


圖 4、提高伺服器利用率示意圖

資料來源：O'Reilly Media, Inc. Build Green Software. (2024). <https://reurl.cc/9nv720>

此外，伺服器使用率偏低的常見原因之一，是為了因應高峰期間的運算需求，確保尖峰時段仍能維持效能，企業讓多台伺服器採低利用率運行，預留冗餘資源。然而，這種方式會導致大量伺服器閒置，進而造成能源浪費與硬體隱含碳的提升。

若將相同情境轉移至公有雲環境，則能有效改善此問題。由於公有雲由多個組織共享資源，任何時點的剩餘容量都能被即時調度使用，大幅提高整體資源使用效率，並減少伺服器閒置率。

整體而言，將運算工作遷移至公有雲，可提升伺服器的使用彈性，並最佳化空間配置，因此相較於使用本地伺服器架構，更具節能與減碳潛力。

### 第三節、碳意識 (Carbon Awareness)

碳意識是指根據電網電力的碳強度來調整軟體的運作時間，使碳排放最小化。當更多的能源來自低碳能源時，碳強度較低；當更多的能源來自高碳能源時，碳強度較高。舉例來說，透過在電網碳強度較低的時段使用電力，可以促進所消耗的能源主要來自風能、太陽能等再生能源，並減少對高碳排放發電廠的依賴。具體而言，碳意識主要透過需求轉移 (Demand Shifting) 與需求塑造 (Demand Shaping) 來維持較高的碳意識。

需求轉移 (Demand Shifting) 分為空間轉移 (Spatial Shifting) 和時間轉移 (Temporal Shifting)。空間轉移 (Spatial Shifting) 是將軟體的運算移到碳強度較低的地區，如靠近再生能源生產的地區；時間轉移 (Temporal Shifting) 是將軟體運算的期程排在碳強度較低的時段，如風力充足或陽光強的時間。



圖 5、空間轉移

資料來源：Green Software Foundation (2024). Green Software Introduction.

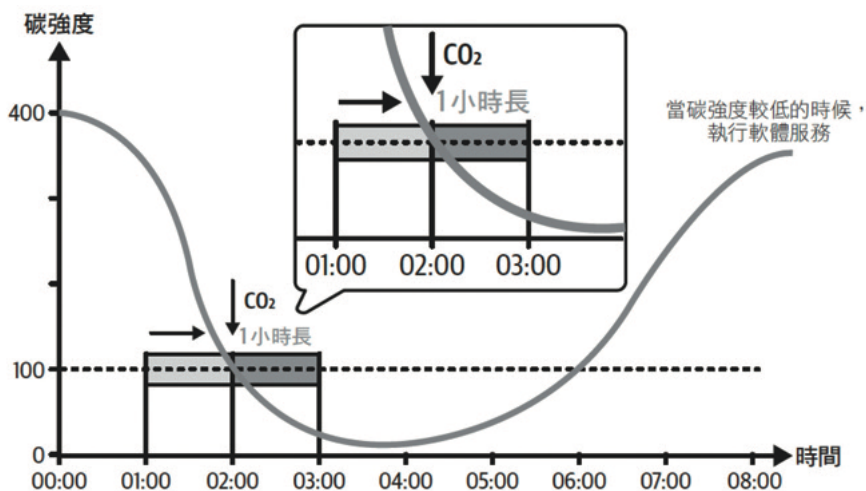


圖 6、時間轉移

資料來源：Green Software Foundation (2024). Green Software Introduction.

需求塑造 (Demand Shaping) 是藉由配合電力的供給狀況，調整軟體的運算行為。例如碳強度低的時候，增加運算的作業（如提高運算品質、執行更多功能等）；碳強度高的時候，減少運算作業、降低性能消耗（如圖 2）。

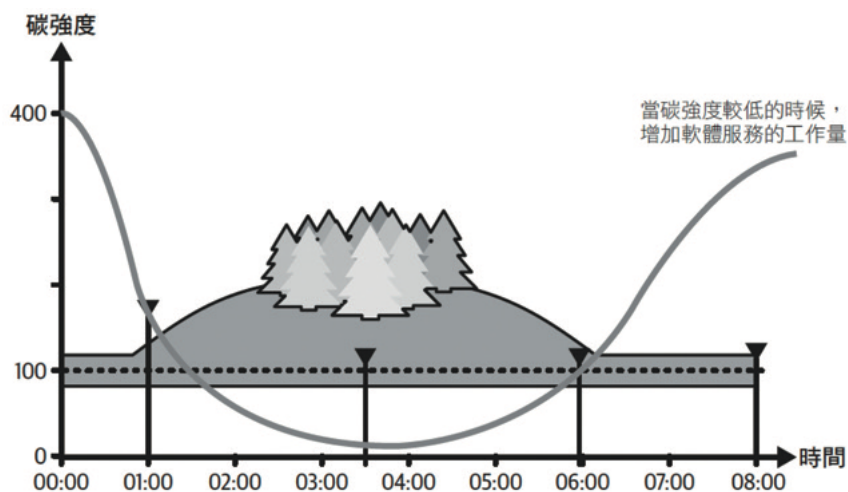


圖 7、需求塑造

資料來源：Green Software Foundation (2024). Green Software Introduction.



## 第三章 Chapter 3

# 綠色軟體與 AI 應用

本章內容主要摘錄自美國 O'Reilly Media 公司於 2024 年出版的《Building Green Software》<sup>18</sup>，並經編輯調整以利閱讀與在地理解。相關概念與內容之著作權屬原基金會所有，特此說明並致謝。

<sup>18</sup> 資料來源：O'Reilly Media, Inc. (2024). Build green software.

## ■ 第一節、綠色軟體在網路永續的角色

人工智慧 (Artificial Intelligence, 以下簡稱 AI) 與機器學習 (Machine Learning, 以下簡稱 ML) 的快速發展，正深刻影響並重塑各行各業。然而，其龐大的運算需求也推動資料中心持續擴張規模，帶來能源消耗與碳排放的挑戰，使「綠色 AI」成為迫切課題。

AI 模型的規模已遠遠超越摩爾定律<sup>19</sup> 的增長速度，特別是深度學習的領域。最早 Emma Strubell, Ananya Ganesh, and Andrew McCallum 等人 2019 年的《Energy and Policy Considerations for Deep Learning in NLP》<sup>20</sup> 研究即指出，訓練一個大型模型的碳排放量可能相當於一輛汽車整個生命週期的排放。近年來，模型參數動輒突破數千億，導致訓練與推論的能源與碳成本急遽上升。Bolón-Canedo 與 Alonso-Betanzos(2024) 綜整近幾年的相關研究，大型深度學習模型在訓練與推論階段皆需消耗大量電力與硬體資源，若未搭配再生能源與高效率設計，將對實現淨零排放目標造成壓力，並進一步說明「綠色 AI」技術的重要性。<sup>21</sup>

因此，綠色軟體強調減碳與效率的理念，涵蓋從設計、開發到部署與維運的完整過程。透過在每個階段導入綠色思維，能有效降低數位服務的環境衝擊，並在 AI 與網路持續發展的同時，確保數位轉型與永續發展得以並行。

## ■ 第二節、更永續的機器學習

### 一、資料收集

機器學習生命週期中的資料收集階段，是減少碳排策略中經常容易被忽略的重要階段。事實上資料的品質對於後續模型的準確度有關鍵的影響，隨著模型需求增加，資料集規模不斷擴大，帶來的碳排放與能源消耗也隨之增加，因此「綠色的資料收集」的重要性也日益提升。相關的綠色資料收集方法包括：

---

<sup>19</sup> 資料來源：林怡廷 . (2020 年, 3 月), 張忠謀：摩爾定律已到極限，半導體未來在系統設計。數位時代。

<sup>20</sup> 資料來源：Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. arXiv.

<sup>21</sup> 資料來源：Bolón-Canedo, V., & Alonso-Betanzos, A. (2024). A review of green artificial intelligence: Towards a more sustainable AI. Neurocomputing.

- ① **善用既有的資料集**：利用既有資源可避免重複建立資料管線，減少額外的碳排放。
- ② **審慎評估資料需求**：只蒐集必要數據，避免「越多越好」的迷思，以降低能耗。
- ③ **注意倫理議題**：大型資料集往往涉及知情同意與隱私爭議，例如許多人無法撤回自己在資料集中的紀錄。例如 RLHF（人類回饋強化學習）的標註工作曾引發勞動剝削與資料同意權的爭議。
- ④ **需求轉移**：若資料蒐集並非即時需求，可選擇在低碳電力供應時段或地點進行，以降低整體碳強度。

## 二、機器學習模型的設計與訓練

模型的設計與訓練是機器學習生命週期中最具特色的環節，這一階段的能耗龐大，但同時也有很多的研究成果與方法可供採用，以降低碳排放並提升永續性。

### ① 模型縮減技術

大型模型的訓練需要龐大資源，因此「縮小模型」是減碳的核心策略。相關方法例如模型稀疏化 (pruning)：透過移除不必要的參數或連結以減少計算量；知識蒸餾 (distillation)：將大型模型的知識轉移至小型模型，兼顧效能與效率；量化感知訓練 (Quantization Aware Training)：以低精度數值（如 8-bit 整數）取代高精度浮點數，降低儲存與運算需求<sup>22</sup>。

### ② 運算位置策略

訓練任務通常並非即時需求，因此可以透過調整時間與地點來降低碳強度，例如：需求轉移 (demand shifting)：將訓練安排在低碳電力供應的時段或地點；邊緣運算 (edge computing)：在資料來源處進行處理與儲存，減少傳輸能耗。聯邦學習 (Federated Learning, FL)：由使用者端裝置分散訓練，再整合成果。研究顯示，雖然 FL 收斂速度較慢，但在小型資料集或低複雜度模型中，整體碳效益可能優於集中式訓練。

### ③ 避免重複訓練並多利用遷移學習

避免從零開始訓練是另一個有效的綠色策略。例如直接使用公開模型。如果找不到適合的模型重複使用，還有另一個方法是遷移學習 (transfer learning)：

---

<sup>22</sup> 資料來源：Meta AI.(n.d).LLM-QAT: Data-Free Quantization Aware Training for Large Language Models.

將既有的大型模型微調以適應新任務，例如現在已經有一個模型已經學會辨認「貓的照片」。這個模型已經知道怎麼辨認毛髮、耳朵、眼睛等特徵。如果你的新任務是要辨認「猴子」，那麼你不需要再從頭教它所有影像辨識的能力，只要在原有的基礎上做一些「微調」(fine-tuning)，它就能快速適應新任務。這不僅能降低碳排放，也能減少成本，是兼具環境與經濟效益的選擇。

#### 四 專用硬體的應用

AI 的運算效率與硬體息息相關，近年來，為 AI 運算專門設計的晶片也快速的在發展，像是圖形處理器 (Graphics Processing Unit ,GPU)：原本用於影像處理，但非常適合平行運算，是目前最常見的 AI 訓練硬體；張量處理器 (Tensor Processing Unit, TPU)：Google 開發的專用晶片，以高容量記憶體矩陣架構設計，專門用於深度學習的模型訓練。

#### 五 綠色程式碼

以下介紹 Google 《Go Green Software》<sup>23</sup> 中提及的多項程式撰寫方法，開發者可透過這些作法提升程式設計的能源效率與永續性。

在模型訓練階段，開發者可透過程式層級的優化，降低不必要的運算量與能源消耗。其中，限制訓練輪次並採用早停 (Early Stopping) 策略，是常見且有效的方法。當模型在驗證集上的表現已趨於穩定時，即可自動終止訓練，避免在收斂後持續運算造成資源浪費；亦可配合動態學習率調整與梯度裁剪等技術，進一步縮短訓練時間並維持模型效能。

此外，精簡訓練資料與程式碼結構亦是重要手段。開發者應避免重複讀取與過度處理資料，僅載入訓練所需的特徵與樣本，並透過資料分割 (sharding) 與批次載入 (batch loading) 方式提升資料處理效率。在程式設計上，刪除冗餘模組與未使用函式可降低記憶體占用與 I/O 操作，減少整體系統負荷。透過這些優化措施，模型在維持精度的同時，能顯著減少 GPU 或 TPU 的能源消耗，達到節能與減碳的雙重目標。

另外，有關資料處理的程式設計方式，也會對機器學習系統的永續性產生影響。在模型開發過程中，透過適當的資料處理與架構設計，可顯著降低模型的

---

<sup>23</sup>資料來源：Google Cloud (2023). Go Green Software.

運算負荷與碳排放量。例如，資料科學家可在特徵工程階段採用降維技術，以減少模型輸入維度並縮小訓練資料集的規模；在模型訓練階段，則可利用實驗追蹤機制記錄參數設定與結果，避免重複訓練造成的能源浪費；完成訓練後，亦可透過模型量化 (Post-Training Quantization) 或使用能效最佳的加速器進行部署，使模型在推論階段以較低的功耗運作，同時維持可接受的準確度。這些程式層級的優化方法不僅有助於減少運算資源的使用，也能提升機器學習系統的整體永續性。

同樣重要的是，在模型部署與維護的階段，也應關注碳排放議題。雖然目前相關研究仍有限，但由於訓練僅需進行一次，而推論 (inference) 在實際應用中會被執行無數次，長期下來往往是更主要的能耗來源。除了前述模型壓縮、量化等技術以降低推論階段的資源需求外，也可引入機器學習操作 (Machine Learning Operations, 以下簡稱 MLOps)<sup>24</sup> 的概念。MLOps 是指從開發到部署與監控整個機器學習生命週期的管理過程，強調自動化監測、持續更新與資源配置最佳化。換句話說，它提供了一套框架，確保模型在生產環境中不會因維護不當而造成額外的能源浪費，進而讓 AI 應用在全生命週期中更加高效與永續。

AI 與機器學習的發展為軟體帶來前所未有的機遇與挑戰。要實現綠色數位轉型，必須從資料收集、模型設計與訓練、部署與維護等環節全面考量，並善用先進的硬體與綠色的部署維運工具。透過實踐這些措施，才能在維持技術進步的同時，兼顧能源效率與碳排減量。

---

<sup>24</sup>資料來源：Google Cloud. (2025). What is MLOps?. Google Cloud.



## 第四章 Chapter 4

# 軟體碳排放評估方法

關於綠色軟體的評估，綠色軟體基金會（Green Software Foundation，以下簡稱 GSF）於 2021 年提出了一套軟體碳強度的 (Software Carbon Intensity, SCI) 的計算方法，後於 2024 年 3 月由國際標準化組織 (ISO) 採納為國際標準「ISO/IEC 21031：2024 軟體碳強度 (SCI) 規範 (Information technology — Software Carbon Intensity (SCI) specification)，是目前國際間主要依循的可靠、一致的衡量標準。本節主要摘錄自 BS ISO/IEC21031:2024 的文件，經編輯調整以利閱讀與在地理解。相關概念與內容之著作權屬國際標準化組織 (ISO) 所有，特此說明並致謝。

## ■ 第一節、國際標準 ISO/IEC 21031:2024 Software Carbon Intensity (SCI)

ISO/IEC 21031 定義了一種評估軟體碳排放速率的方法，其目的除了提升應用程式的永續意識與透明度，也在於協助使用者與開發者在選擇工具、方法與服務時，做出的決策更符合永續的精神。

### 一、軟體碳強度 (Software Carbon Intensity, 以下簡稱 SCI)

SCI 並非用來計算軟體碳排放的總量，而是一種分數型指標。分數越低，代表軟體對環境的影響越小，並隱含「持續降低分數」的永續目標。該標準著重說明使用者和開發人員瞭解如何改進軟體以減少或避免產生排放，而非依賴碳抵換 (Carbon Offset) 的手段，因此更適合作為組織朝向氣候目標邁進的策略工具。

換言之，欲降低 SCI 的分數，僅能透過實際消除碳排來源來實現，例如使用較少的硬體資源、降低能源消耗、使用低碳或再生能源等，若採用碳中和 (Carbon Neutrality)、碳抵換 (Carbon Offset) 等減碳的方法，SCI 的分數是不會反映出來的。

### 二、SCI 的適用場景

依據 ISO/IEC 21031 的內容，SCI 可適用於各種軟體，包括：大型分散式雲端系統 (distributed cloud system)、單機版開源函式庫 (small monolithic open source library)、本地應用程式 (on-premise application)、無伺服器架構 (Serverless Functions) 等。其運行環境也包括個人電腦、私有資料中心，或是超大規模的公有雲平台等。

### 三、SCI 分數的比較

根據文件說明，若欲進行軟體碳強度 (SCI) 之比較，應先建立明確的基準參照數據作為依據，並確保不同軟體或版本間的量測方式、假設前提、模型設定與功能單位一致，以維持數據的可比性與公正性。

### 四、SCI 分數的特性

#### ① 減碳策略需要聚焦在綠色軟體的核心原則

SCI 分數的計算架構是根據綠色軟體的原則進行設計，因此欲改善（降低）

SCI 分數的表現，必須針對軟體的碳意識 (Carbon Awareness)、能源效率 (Energy Efficiency) 以及硬體效率 (Hardware Efficiency) 等面向進行優化。

### ㉔ 關注系統整體性的影響層面

SCI 的核心精神在於以「系統層級」的方式推動減碳，強調軟體應整合運算資源、資料儲存與網路傳輸等各環節的整體效益，而非僅針對單一模組或功能進行局部最佳化。若僅著重於某一部分的效能提升，可能造成其他環節的能源消耗上升，反而抵銷整體減碳成效。因此，在評估軟體碳強度時，需明確界定軟體運行的範圍與基礎設施邊界，並依據規範排除不在評估範圍內的項目，以確保結果的完整性與一致性。

要實現這樣的系統觀，需明確界定軟體及其相關基礎設施的邊界，同時考慮本規範中所列的排除項目。為了實踐這種系統性思維 (system-level thinking)，必須清楚界定軟體本身及其相關基礎設施的邊界（詳見後面 SCI 分數計算與報告的流程）確保評估的全面性與一致性。

### ㉕ 較低的操作及導入門檻

為了提供更廣泛的應用，SCI 分數設計強調簡單且便利的操作方式，即使是缺乏相關經驗或訓練的使用者，也能依循 SCI 規範中的指引進行操作。SCI 的設計原則強調降低門檻與成本，因此其分數的計算不會需要付費的資料、服務或工具，進而確保所有使用者都能無障礙的參與。

## 五、SCI 分數計算與報告的流程

### ㉑ 步驟一：劃定邊界 (Bound)：

確定軟體邊界，軟體邊界可以是：運算資源、記憶體、訓練 ML 模型、操作 ML 模型，如果邊界包括雲端資料中心運營，應考慮雲端資料中心的 PUE 效率。

### ㉒ 步驟二：選擇規模 (Scale)：

由於 SCI 是一個比率 (表示每個功能單位所產生的碳排放量)，應選擇最能描述應用程式規模的功能單位，例如：時間、機器、資料量等。

### ㉓ 步驟三：定義量測方法 (Define)：

對於軟體邊界內列出的每個軟體組件，確定量化方法；可以基於實際測量、遠端測量 (遙測) 或基於模型的計算數據。

#### ④ 步驟四：進行量化 (Quantify)：

計算每個軟體組件的比率。整個應用程式的 SCI 值是系統中每個軟體元件的 SCI 值的總和。

#### ⑤ 步驟五：進行揭露 (Report)：

揭露 SCI 分數、軟體邊界和計算方法。

### 六、SCI 方法論與計算方法

#### ① 名詞定義

1. E(Energy) 表示軟體在執行過程中實際的耗電量。
2. I(Location-based marginal carbon intensity) 表示依地點計算的邊際碳強度。
3. M(Embodied emissions) 表示軟體所運行硬體設備本身的隱含碳排放。
4. O(Operational emissions) 表示軟體運作時基於能源使用所產生碳排放。
5. R(Functional unit) 是功能單位。

#### ② 計算公式

$$SCI = ((E * I) + M) \text{ per } R$$

圖 8、軟體碳強度計算標準

資料來源：thoughtworks 網站。 <https://reurl.cc/6Kk3gM>。

#### 1. $SCI = C \text{ per } R$

C 指的是排碳總量，每個功能單元 R，可以是：使用者、機器、時間（分）、資料量等。

#### 2. $SCI = (O + M) \text{ per } R$

O 指的是軟體應用程式在操作過程中所產生的碳排放量，M 則是指硬體裝置建立和處置過程中的碳排放量（意即隱含碳排，詳見第一章第三節）。

#### 3. $O = (E * I)$

為了計算軟體在運作階段所產生的碳排放，需將軟體執行所在硬體的耗電量

乘以該地區精準的邊際碳排放係數 (granular marginal emissions rate)。功能單元工作的耗電量  $E$ ，單位為 kWh，電力的碳排放係數  $I$ ，單位為  $\text{gCO}_2\text{eq/kWh}$ 。

$$4. M = TE * TS * RS$$

$TE$ = 總體隱含排放， $TS$ = 時間份額（指硬體總使用壽命中為軟體保留的使用份額）， $RS$  = 資源份額（指硬體總可用資源中為軟體保留的使用份額）。

$$5. M = TE * (TiR/EL) * (RR/ToR)$$

$TiR$ = 保留時間（硬體為軟體保留使用的時間長度）  $EL$ = 預期壽命（設備預計安裝使用的時間）。 $RR$ = 保留資源（為軟體保留使用的資源數量）  $ToR$ = 資源總量（可用的資源總數）。

$$6. SCI = ((E * I) + TE * (TiR/EL) * (RR/ToR)) \text{ per R}$$

最終公式整合軟體運行階段與隱含碳部分排放，能完整反映軟體在運作與硬體使用中對碳強度的實際貢獻，為後續進行軟體低碳化設計與改善的重要基礎。

## 第二節、軟體的碳排放測量

在進行軟體碳排放量測之前，首先必須明確理解「電力排碳強度」與「硬體設備的隱含碳排」這兩項關鍵概念，並進一步界定軟體碳排放的計算邊界、功能單元與量測方法。在了解這些基礎設定與參數，才能進行具有一致性與可比較性的碳排放測量與評估。

### 一、電力排碳強度 (Carbon Intensity of Electricity)

電力排碳強度是指在發電過程中，為生產 1 度電所排放的溫室氣體量，一般以「公斤二氧化碳當量／度電」( $\text{kg CO}_2\text{e/kWh}$ ) 表示。而電力排碳係數是用以換算電力使用量對應的碳排放量的轉換比例常數，通常由政府或國際機構公布，用來計算排放量，例如我國 113 年度電力排碳係數為 0.474 公斤  $\text{CO}_2\text{e}/$ 度<sup>25</sup>，其計算方法為（發電業及自用發電設備設置者躉售公用售電業電量之電力排碳量－線損承擔之電力排碳量）除以公用售電業總銷售電量。另外亦可參考國際上的 Electricity Maps 電力地圖網站<sup>26</sup>，該網站提供臺灣地區每小時的電力排碳係數。

<sup>25</sup> 資料來源：經濟部能源局。(2025).113 年度電力排碳係數。

<sup>26</sup> 資料來源：Electricity Maps, Interactive App – Real-time electricity data

## 二、硬體設備的隱含碳排 (Embodied Emissions)

硬體設備的隱含碳排是指硬體在製造、運輸與報廢過程中產生的碳排放，在軟體碳排放量測中，硬體的隱含碳排分攤方式依據硬體使用壽命與軟體實際佔用時間而定，以確保硬體的隱含碳排能被合理的分擔與計算至軟體碳排放中。硬體隱含碳排的取得方式可以透過硬體設備生產製造商所提供的數據，或是藉由查詢硬體隱含碳排資料庫取得，例如：Boavista 網站。另外也可由 Boavista 網站中查詢不同硬體設備的平均使用年限，來作為預期壽命的參考。

## 三、軟體邊界 (Software boundary)

計算 SCI 分數的第一步是決定軟體系統的界限是什麼，也就是在計算 SCI 分數時要包含或排除哪些軟體元件。SCI 的計算應包括對軟體執行有重大貢獻的基礎設施和系統，包含以下幾這些項目：

compute resources 運算資源

storage 儲存設備

networking equipment 網路設備

memory 記憶體

monitoring 系統監控

idle machines 閒置機器

logging 日誌紀錄

scanning 掃描

build and deploy pipelines 建置與部署流程

testing 測試

training ML models 機器學習模型訓練

operations 系統運維

backup 備份

resources to support redundancy 系統備援的資源

resources to support failover 故障切換的資源

End user devices 終端使用者裝置

IoT devices 物聯網裝置

Edge devices 邊緣運算裝置

#### 四、軟體功能單元 (Functional unit)

計算 SCI 分數的第二個步驟，是決定用哪一個功能單元來作為軟體碳排放的計算內容。建議的軟體功能單元包含以下這些項目：

API call/request API 呼叫／請求

Benchmark 基準

User 使用者

Machine 機器

Minute/time unit 分鐘／時間單位

Device 裝置

Physical site 實體站點

Data volume 資料量

Batch/Scheduled Job 批次／排程作業

Transaction 交易

Database read/write 資料庫讀寫

#### 五、如何測量軟體的排放量 (Measurement)

測量有兩種方法：直接測量和間接計算

**直接測量**是使用某種形式的計算器來測量。例如：使用牆壁插座中的硬體裝置來測量軟體的能耗，或者在硬體上使用直接測量能源消耗的計算工具。

**間接計算**涉及間接量測，通常是使用某種形式的模型。例如：基於 CPU 利用率估計能耗的模型，這被視為間接計算而不是直接測量。

如果能直接測量應該以直接測量能耗為主，而不是採用間接計算。

## 六、開始計算軟體排放量 SCI

依照前述步驟所描述的方法與第一節的軟體 SCI 計算公式，可開始量化軟體的 SCI 分數，也可以為同一個應用程式計算多個不同的 SCI 分數，並可以了解應用程式在不同情境下的碳排放表現。例如：以 YouTube 串流應用程式為例，計算該軟體的碳排放可以選擇「每分鐘或每五分鐘的碳排放量」作為指標，也可以計算「每位使用者每日的碳排放量」，或是以「每單位營收的碳排放量 (carbon per revenue)」為指標，提供各種不統碳排放的分析維度。以下為使用 Asus ZenBook 在 Windows 作業系統中，使用 Chrome 瀏覽器觀看 YouTube 影片 1 小時的碳排放量計算範例：

$$E = 0.000214354 \text{ kWh}$$

$$I = 0.499 \text{ kg CO}_2/\text{kWh}$$

$$TE = 11.56 \text{ kg CO}_2$$

$$TiR = 1 \text{ 小時}$$

$$EL = 4 \text{ 年}$$

$$RR = 1.43$$

$$ToR = 100$$

$$\text{依照公式：} SCI = ((E * I) + TE * (TiR/EL) * (RR/ToR))$$

可算出觀看 YouTube 影片 1 小時的碳排放量 SCI 為

$$0.0005730763 \text{ kg CO}_2$$

## 第三節、軟體碳排放檢測（體驗）工具介紹（試算範例內容）

在本節中將介紹軟體碳排放檢測原理及三款運行在不同平台的軟體碳強度計算機，分別是 Windows 平台、iOS 平台與 Android 平台，這三款計算機均是以軟體運行 1 小時作為軟體功能單元來計算軟體的碳強度，在不同的平台上可以選擇 CPU 或是整機作為軟體邊界來進行軟體碳排放計算（可以到 GreenWeb 網站或是 GitHub 網站下載）。

特別說明：下面將介紹範例計算機，係為配合軟體碳排放檢測之推廣與手冊說明案例所開發，旨在協助使用者透過實際操作了解軟體碳排放的量測方式與原理，並作為入門參考與示範性工具，協助業者與開發者更具體地掌握軟體碳排放檢測流程。未來正式版工具仍將持續優化與擴充功能，以提升操作精度與應用範圍，並支援更多實務場域與軟體型態之檢測需求。

## 一、軟體碳排放檢測原理說明（以 CPU 運算為例）

計算軟體碳排放時，可以依照不同的需求設定所需要的軟體邊界與軟體功能單元，以下的範例，是以 CPU 作為軟體邊界，以軟體運行 1 小時作為軟體功能單元來進行示範。如下圖：

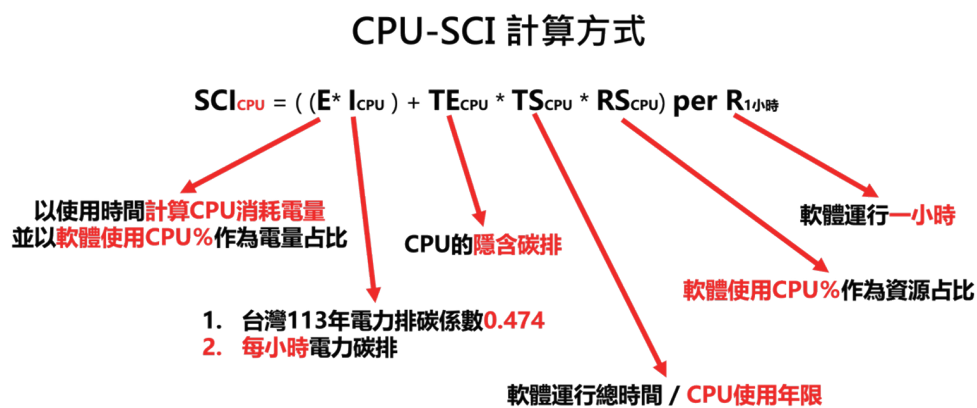


圖 9、以 CPU 運算為例

在圖 9 中，E 是以 CPU 運行 1 小時所需的電量乘以軟體運行中佔用 CPU 資源比例作為軟體運行的耗電量，之後再乘以不同的電力排碳係數（去年台灣的平均值 0.474 或每小時台灣地區的電力排碳係數）即可計算出軟體執行一小時所產生的碳排放量。

接下來再以 CPU 的隱含碳排 TE 乘以軟體的運行時間 1 小時除以 CPU 的使用年限（一般為 4 年，35,040 小時），再乘以軟體運行中佔用 CPU 資源比例後，即為運行軟體 1 小時所需要分攤的 CPU 隱含碳排。

將軟體執行一小時所產生的碳排放量加上運行軟體 1 小時所需要分攤的 CPU 隱含碳排後，除以軟體運行的時間 1 小時，即可算出軟體以 CPU 為軟體邊界與運行 1 小時作為軟體功能單元的軟體碳強度 SCI 值。

## 二、Microsoft Windows 軟體碳排放檢測工具說明



圖 10、Windows 軟體碳排放強度計算機（首頁）

圖 10 為 Windows 軟體碳排強度計算機的首頁，該計算機會自動抓取運行的電腦裝置型號與 CPU，並自動獲取電腦裝置與 CPU 的隱含碳排。點選下方的檢測報告可以檢視上次的檢測結果，點選操作說明可進一步獲得操作的說明，點選隱私權政策可以了解該計算機對於隱私權的保護，點選測試設定，即可開始設定要檢測哪個軟體的碳排放。



圖 11、Windows 軟體碳排放強度計算機（檢測設定）

在檢測設定中可以勾選是否包含背景程式，並可依照關鍵字搜尋軟體名稱，或是使用下拉選點選想要檢測的軟體，然後選擇要測試的範圍（整機或是 CPU）與測試時間長度（1~60 分鐘）後，點擊開始檢測即開始檢測。

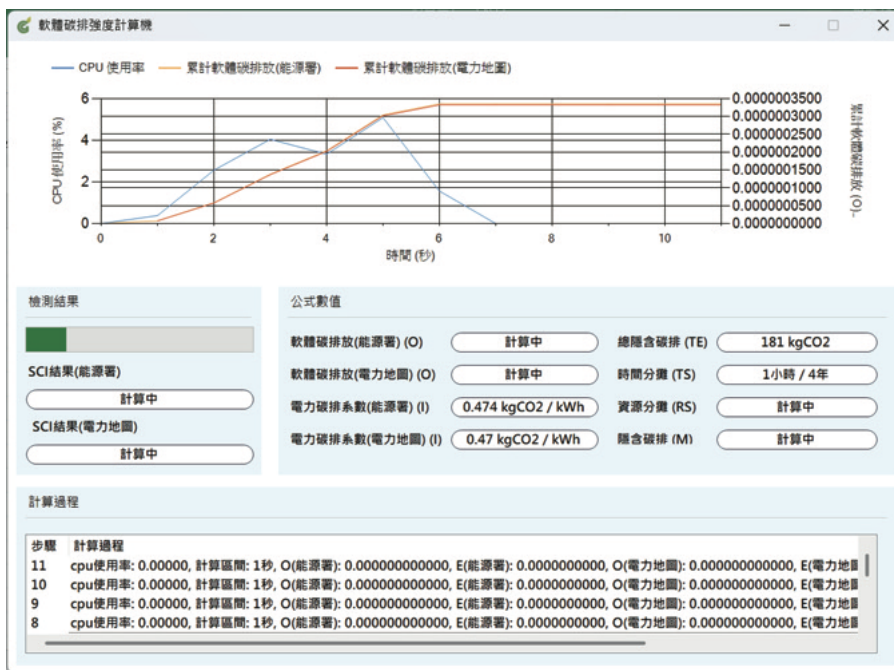


圖 12、Windows 軟體碳排強度計算機 (檢測畫面)

在圖 12 檢測畫面上方可以看到 CPU 資源佔比的變化與軟體碳排放的累計曲線，在中間的公式數值可以看到總體隱含碳排 TE 與時間份額 TS，以及台灣地區的即時電力排碳係數與過去一年的平均電力排碳係數，下方計算過程可以看到計算機的計算統計數值。

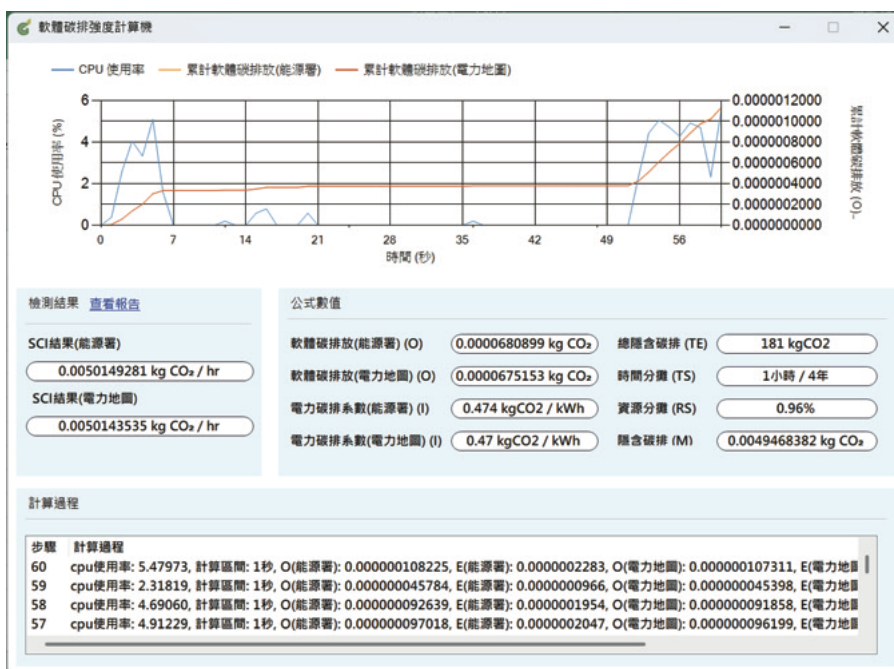


圖 13、Windows 軟體碳排強度計算機 (檢測完成)

在測試完成後，可以在畫面中看到二個 SCI 結果，分別是以台灣地區的即時電力排碳係數與過去一年的平均電力排碳係數來計算，另外可以在公式數值中看到資源分攤 RS 與隱含碳排 M 的計算結果，另外也可以點選查看報告，便可看到完整的檢測報告內容，如圖 14。



圖 14、Windows 軟體碳排放強度計算機 (檢測報告)

以下為使用 Windows 軟體碳排放強度計算機在 Asus ZenBook 筆記型電腦中，計算 Chrome 瀏覽器觀看 YouTube 影片 1 小時 CPU 所產生的碳排放量為例，首先選擇檢測程式：chrome，並選擇檢測範圍：CPU 與選擇檢測時長：5 分鐘後，點選：開始檢測。如下圖。



圖 15、Windows 軟體碳排放強度計算機 (設定畫面)

經過五分鐘的計算之後即可算出分別使用能源署的電力排碳係數與電力地圖網站的每小時電力排碳係數所算出來的 SCI 分數，與其他項目的計算，如：隱含碳排等。

如圖 16。

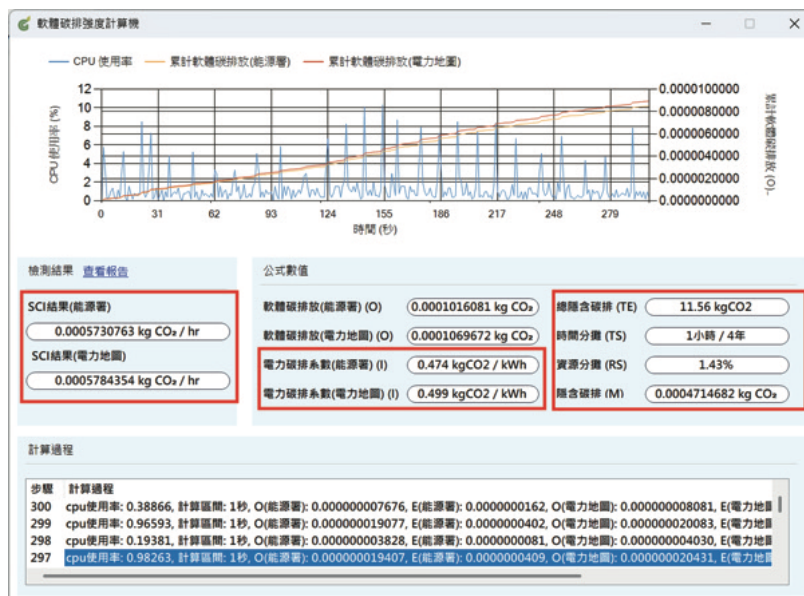
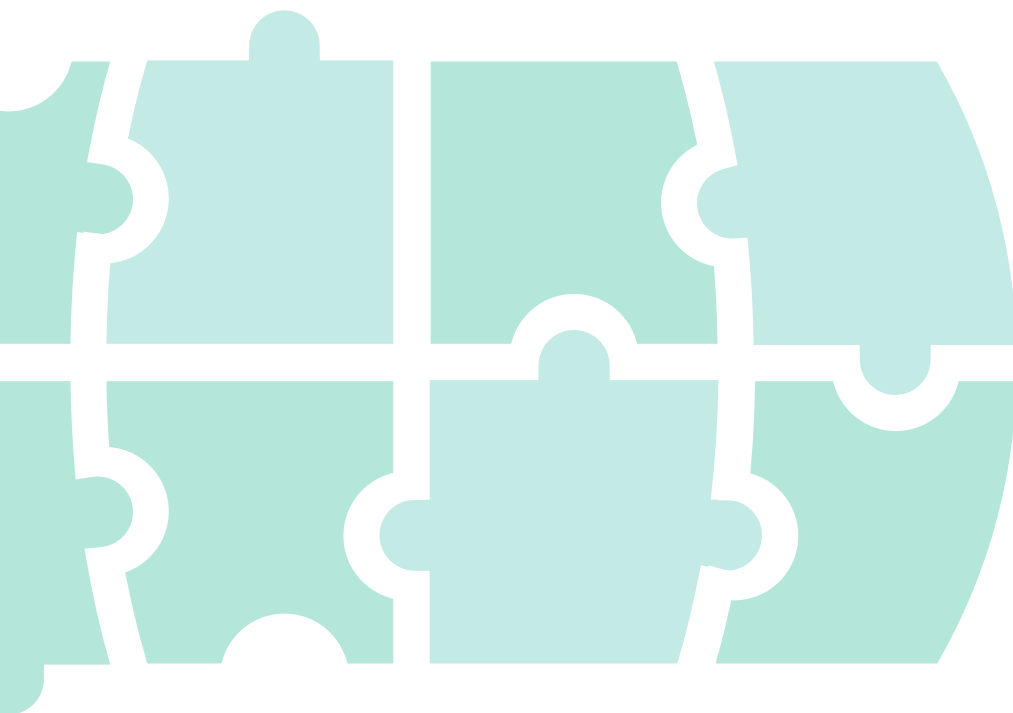


圖 16、Windows 軟體碳排強度計算機（計算結果畫面）



### 三、Apple iOS 軟體碳排放檢測工具說明

圖 17 為 iOS 軟體碳排放強度計算機的首頁，該計算機會自動抓取運行 iOS 裝置型號與系統版本，並自動獲取 iOS 裝置的隱含碳排。點選開始測試，即可開始設定要檢測哪個軟體的碳排放，點選操作說明可進一步獲得操作的說明，點選隱私權政策可以了解該計算機對於隱私權的保護。

在圖 18 的檢測畫面中可以輸入想要檢測的軟體 APP 名稱，並在 iOS 的「設定」應用程式中選取「電池」來找到該軟體 APP 的使用日期、前景使用時間、背景使用時間與耗電百分比及電池健康度後，於畫面中輸入後，點選開始計算後，下方會即刻顯示計算結果，並可以點選檢測告觀看完整的報告內容。如圖 19。



圖 17、iOS 軟體碳排放強度計算機（首頁）



圖 18、iOS 軟體碳排放強度計算機（檢測畫面）

#### 軟體碳排放強度(SCI) 計算結果



圖 19、iOS 軟體碳排放強度計算機（檢測報告）

## 四、Google Android 軟體碳排排放檢測工具說明

圖 20 為 Android 軟體碳排強度計算機的首頁，該計算機會自動抓取運行 Android 裝置型號與系統版本，並自動獲取 Android 裝置的隱含碳排。點選開始測試，即可開始設定要檢測哪個軟體的碳排放，點選操作說明可進一步獲得操作的說明，點選隱私權政策可以了解該計算機對於隱私權的保護。

在圖 21 的檢測畫面中可以下拉選擇想要檢測的軟體 APP 名稱，並在 Android 的「設定」應用程式中選取「電池」來找到該軟體 APP 的前景使用時間，背景使用時間與耗電百分比及電池健康度（非必填）後，於畫面中輸入後，點選開始計算後，下方會即刻顯示計算結果，並可以點選檢測告觀看完整的報告內容。如圖 22。



左 - 圖 20、Android 軟體碳排強度計算機（首頁）

中 - 圖 21、Android 軟體碳排強度計算機（測試畫面）

右 - 圖 22、iOS 軟體碳排強度計算機（檢測報告）



## 第五章 Chapter 5

### 綠色軟體應用情境

本章第一節將帶領讀者認識目前國際間與軟體碳效率相關的兩個主要制度：德國的「藍天使環保標章」及綠色軟體基金會 (Green Software Foundation) 推動的「軟體碳效率評級制度」。第二節則介紹我國正在研議中的「自願性綠色軟體標章制度」，內容涵蓋政策依據、制度設計與評估原則，並說明標章制度在公信力、技術可行性與產業誘因等面向的設計思考與推動方向。

## 第一節、國際綠色軟體相關參考制度介紹

為建立符合我國產業發展與政策需求的綠色軟體標章制度，了解國際間相關制度的發展經驗與推動架構具有重要參考價值。本節介紹目前具有代表性的兩項制度：德國藍天使 (Blue Angel) 的「資源和能源效率軟體產品基本授獎標準 (Resource and Energy-Efficient Software Products)」以及綠色軟體基金會 (GSF) 推動的「軟體碳效率評級制度 (Software Carbon Efficiency Rating, SCER)」。這兩項制度分別以環境標章與碳效率分級的方式推動軟體永續化，提供我國自願性綠色軟體標章制度設計與推動規劃的重要參考。

### 一、資源和能源效率軟體產品基本授獎標準 (Resource and Energy-Efficient Software Products, 以下簡稱 DE-UZ 215)<sup>27</sup>

德國的「藍天使環保標章 (Blue Angel)」是國際上發展悠久的其中一個環境標章制度 (1978 年由德國政府創立)，起初主要是針對硬體、耗材、辦公設備、清潔等產品，之後在 2020 年提出 DE-UZ 21，針對資源與能源高效的軟體產品授予藍天使標章（標章樣式如圖 23），是最早將軟體納入環境標章體系的國際制度之一。DE-UZ 215 將軟體產品納入評選範疇，針對軟體在能源使用與資源效率上的表現訂定明確標準，鼓勵開發者設計更具環境效益的軟體。被授予標章的軟體產品必須在硬體資源使用上特別高效，且在使用過程中也能達到能源消耗較低的產品。



圖 23、德國藍天使「資源與能源高效的軟體產品」標章

資料來源：Resource and Energy-Efficient Software Products DE-UZ 215(2020)，BLUE ANGEL The Environmental Label。

<sup>27</sup> 資料來源：German Environment Agency (2020). Resource and Energy-Efficient Software Products, DE-UZ 215. <https://produktinfo.blauer-engel.de/uploads/criteriafile/en/171/DE-UZ%20215-202406-en-Criteria-V4.pdf>

DE-UZ 215 標準中指出了軟體測試適用的情境標準、測試程序以及效能評估指標等內容，並明確要求軟體產品的碳效率評估必須分別測量軟體在不同應用階段（如啟動、閒置、執行任務等）下的能源與資源效率表現。

除了對能源消耗進行量化評估外，DE-UZ 215 也特別強調軟體對硬體使用壽命的影響，要求不得強迫用戶頻繁升級硬體，並應促進使用者自主控制功能資源的開關與運作方式，其他還包含可更新性、無廣告干擾、資料保護與可用性等非功能性的永續設計原則等等。以下為其規範的重點摘要：

- ① **適用範圍**：DE-UZ 215 標準目前僅開放具有使用者介面的應用程式申請標章，例如桌面應用、系統工具、行動應用與具前後端運算能力的 Web App 等。
- ② **申請與管理**：標章由德國環境標章制度的執行與管理機構 RAL gGmbH 核發；申請者須與 RAL 簽署使用合約，並於合約期滿時提交「資源效率報告」，說明產品為提升能源與資源效率所採取的措施。此外，標章僅適用於經核准的軟體版本，若產品進行重大升級，應重新申請驗證。
- ③ **評估原則**：軟體應以最小的資源使用與能源消耗提供功能，並避免迫使使用者提早汰換硬體（例如更新導致需升級設備）。
- ④ **使用者自主性**：不得限制使用者操作或創造不必要的依賴。
- ⑤ **測試方法**：標準測試須採三階段能耗測試，包括基準負載、閒置負載與實際使用負載三種情境，並要求進行多次重複的量測以確保統計可靠性。

## 二、軟體碳效率評級規範 (Software Carbon Efficiency Rating Specification, 以下簡稱 SCER)<sup>28</sup>

綠色軟體基金會 (GSF) 提出的 SCER 是「規範的規範」，用來說明若要制定一套評估軟體碳強度／碳效率的評級規範，該規範應包含哪些要素。它提供軟體開發人員、IT 專業人員、政策制定者與企業決策者等在起草評級規範時可依循的共通框架。

制訂評級規範制度是讓功能類似的軟體在相同基準下比較碳排放，並以 ISO/IEC 21031 所定義的 SCI 計算分數為主要衡量標準，針對各類型軟體設計標準化測試流程，

---

<sup>28</sup> 資料來源：Green Software Foundation(2023). Software Carbon Efficiency Rating (SCER)Specification. <https://reurl.cc/QYIR0Z>

以計算 SCI 分數並提供一致性的評量依據。規範的內容應詳細規定標準測試流程、評級的規範與揭露方式、軟體的分類等。

### 一 標準的測試流程

1. 明確定義測試過程中的工作負載量，包括軟體在基準測試中可能承受各類操作情境，以及對應的工作負載指標，如同時使用者數量、資料規模或混合運算比例等。
2. 設定標準化的測試環境，降低外部變數對結果的影響；相關條件須明確描述測試所使用的硬體規格（如處理器、記憶體、儲存與網路等）及軟體配置（如作業系統、中介軟體與元件版本）。
3. 詳細說明且可重複進行的測試方法，包含清楚的環境設定步驟、測試執行流程（例如啟動軟體與工作負載、監控效能、記錄結果）以及資料蒐集與分析方式，確保結果的完整性與一致性。

### 二 評級規範

SCER 強調軟體碳效率評級規範是根據同一類別中軟體產品的 SCI 分數表現，以界定其「軟體碳效率等級」。因此，規範中應明確定義各評級等級的標準，說明 SCI 分數如何轉換為具體的評級結果，並建立統一的評級揭露格式。

在實際操作上，評級流程首先需蒐集同一類別中多個軟體應用程式的 SCI 分數，並依分數高低進行排序；其次，計算各軟體在整體分布中的百分位位置；最後，依據既定的評級對照表，將軟體歸入對應的等級範圍。評級流程參考如圖 24。



圖 24、評級範例流程圖

資料來源：本專案繪製。

### 三 評級規範

軟體碳效率評級的揭露可分為兩個層面：其一為相對評級，即在同一軟體類別中比較各產品的 SCI 分數表現，顯示其同類型軟體中的碳效率排序；其二為絕對分數

揭露，以軟體每功能單位所產生的 SCI 分數（例如每使用單位產生之  $\text{gCO}_2\text{e}/\text{USF}$ ）提供標準化的碳效率量化依據。

在相對評級中，例如下表所示，資料庫軟體甲的 SCI 分數為 305，低於資料庫乙的 455，因此在評級中獲得 A 級，優於乙的 B 級。此結果表示軟體甲在運行過程中的碳排放量較低，整體碳效率表現較佳。

表 1、同類型軟體比較範例

	資料庫甲	資料庫乙
評級	A	B
SCI 分數 (單位： $\text{gCO}_2\text{e}/\text{USF}$ )	305	455

值得注意的是，對於不同類別的軟體，如資料庫與桌面應用程式，在下表中顯示兩者皆為 B 級，但 SCI 分數差異明顯：資料庫甲為 600，而桌面程式乙僅 66。這反映出不同軟體類別的運作模式與能源需求差異，即使評級相同，絕對 SCI 分數仍能提供更細緻的碳效率資訊，供使用者進行橫向理解與參考。

表 2、不同類型軟體比較

	資料庫甲	資料庫乙
評級	A	B
SCI 分數 (單位： $\text{gCO}_2\text{e}/\text{USF}$ )	305	455

總地來說，相對評級與絕對評級的結合，讓使用者能夠理解不同技術在運行時相對的能源密集程度。

## 第二節、國內綠色軟體應用場景 - 自願性綠色軟體標章制度構想

本節旨在說明如何將「綠色軟體」理念於我國具體化與落地實踐，並探討可應用的推動場景。以建立自願性綠色軟體標章制度為例，透過制度化的引導機制，協助企業在軟體開發階段導入綠色設計思維，提升產品的能源效率與運算效能，同時以低碳

設計理念優化軟體生命週期。此制度不僅有助於推動產業邁向永續轉型，也能呼應國際市場對綠色與永續產品的重視，使我國軟體產業與全球供應鏈的永續要求接軌。藉由建立具公信力的標章體系，企業可在參與國際供應鏈、永續採購及 ESG 評估時，展現具體的環境績效，進而彰顯臺灣軟體產業於國際舞台的永續實力與競爭力。

以下將從「自願性綠色軟體標章」之定位、目的、運作方向、申請方式、測量方法、使用方式及管理查核等面向提出構想。

### 一 標章構想的定位與目的

標章制度是為了鼓勵產業自主參與，希望業者在軟體設計、開發與服務階段融入節能減碳與永續思維。透過輔導業者進行軟體碳排放分數計算與自我宣告或承諾，協助企業逐步建立環境績效的自我檢視與揭露能力，展現軟體產品在環保面向上的努力與成果。因此，本制度希望以自願性質的方式推動，以「鼓勵而非篩選」為原則，確保制度對企業而言是「助力而非負擔」。此設計旨在避免不同企業間因規模或資源差異而產生不必要的差異化，並透過友善的導入環境，促使業者自發投入綠色設計與持續改善，逐步形塑產業整體的永續發展文化。

### 二 標章構想的運作方向

未來若推動相關機制，可考慮由專業單位協助建立一個統籌辦公室作為單一窗口，負責受理申請與技術檢視等相關作業，並協助整合文件審查、輔導及查核流程。此構想旨在確保運作過程具備透明、公正與專業性，同時讓推動工作能在主管機關的指導下穩健進行。

為使後續推動方向更趨完善，可考慮建立一個專家審議會，邀集專家學者及相關團體代表參與，提供專業意見與建議。審議會可以定期或視情況臨時召開，審議內容包括如軟體碳強度測試方法訂定、抽測及測試結果確認、違規或爭議案件處理，以及申請、審查、核發及追蹤管理案件等等，確保構想的發展兼具專業判斷與社會公信力。

## 三、申請要求與方式

在初步構想中，申請流程可能包含軟體產品的基本資料，例如產品名稱、版本、主要功能、申請單位及適用範圍等。同時，參考國際制度，還需要準備軟體碳強度 (Software Carbon Intensity, SCI) 分數與綠色軟體自我宣告文件，作為後續評估的參考依據。

對於 SCI 分數的計算，可由申請者提供產品或產品開發階段予主管機關或其委辦單位進行軟體碳強度 (SCI) 分數的計算，由廠商自行決定揭露與否，主管機關暫不公開（僅供利害關係人查詢）。此外，考量軟體碳排放的量測方式仍在發展中，且相關標準與驗證體系尚待建立，初期可能開放採取較為彈性的計算作法。申請者可依現有參考方法進行初步估算，並提供相關資料供後續確認。這樣的設計思維可兼顧參與的便利性與數據的可信度，逐步累積經驗，為未來機制的完善奠定基礎。

至於綠色軟體自我宣告文件，作為整體構想中的核心要素之一。希望申請者在文件中說明產品於設計、開發、部署及使用階段的永續作法與承諾，例如節能策略、雲端資源管理或使用者端效能優化等。透過這樣的方式，鼓勵業者展現綠色設計理念，並促進社會大眾對軟體永續價值的理解與信任。宣告內容例如：

- ㊟ 能依照該軟體功能選擇適當之功能單元計算出軟體邊界所產生的軟體碳排放強度；
- ㊟ 承諾以永續的碳意識及提高能源效率與硬體效率的原則設計軟體 ( 需提出佐證資料 ) ；
- ㊟ 軟體開發設計相關人員取得綠色軟體相關課程認證；
- ㊟ 鼓勵軟體廠商以永續節能的方式運行軟體；
- ㊟ 宣告軟體的最低支援年限；
- ㊟ 宣告軟體最低支援的硬體規格與年限；
- ㊟ 宣告軟體最低支援的作業系統清單、版本；

藉由上述承諾，逐步建立可信且具操作性的申請流程，進而鼓勵軟體使用者優先採用具有綠色軟體標章的軟體。

#### 四、評估與測量方法

為確保標章的評估結果具國際一致性與可比性，測量方法傾向依循由微軟、埃森哲等國際企業共同創立之綠色軟體基金會 (Green Software Foundation, GSF) 所提出的「軟體碳強度計算方法 (ISO/IEC 21031)」及綠色軟體核心原則（如手冊第三章與第四章所述），進行評估方法與指標設計。這樣的作法也是為了促進制度與國際標準接軌，並提升國內產業於國際市場的採用度與認可度。

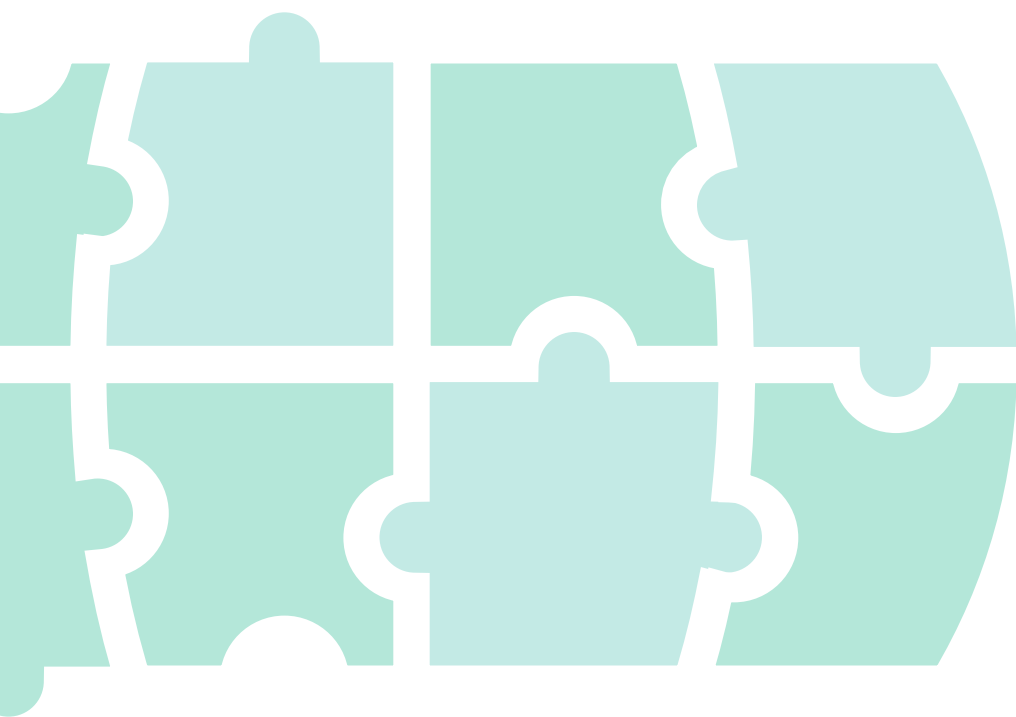
## 五、標章的使用與資訊揭露

未來推動方向中，希望取得標章的軟體產品，能以清楚、透明的方式說明自身的基本資訊，例如產品類別、主要功能，以及與能源使用或碳排放表現相關的指標。這樣的揭露方式能幫助使用者了解軟體在永續設計上的努力與成果。

至於標章的呈現方式，未來考慮開放多元的標示途徑，例如放在軟體介面、官方網站或產品說明文件中，讓使用者能輕鬆辨識。若有其他展示或標示的需求，也建議經過確認與核准後再使用，以確保標章呈現的一致性與可信度，避免出現誤導消費或使用者的情形。

## 六、標章的管理與查核

為確保後續推動能維持穩定與持續改善，團隊考慮思考設計一個定期檢視與更新的機制，例如每年回顧軟體的能源效率與碳排放表現，並鼓勵業者持續優化。若未來有導入查核或追蹤機制，也可採取抽樣或輔導方式，以協助業者逐步提升相關綠色設計能力，同時維持整體構想的公正與可信度。



## 附錄一、專家諮詢與編輯團隊

本手冊由財團法人台灣經濟研究院受數位發展部委託執行「綠色數位服務規劃及推廣計畫案」所編製，為確保內容兼具專業性與可行性，本案於編製過程中共召開六次專家諮詢會議，並廣泛徵詢產、官、學、研代表之意見。相關建議均已納入研析，據以修訂與完善手冊內容，期能作為後續政策研議與產業實務推動之重要參考。本手冊為現階段研究成果，內容將配合制度推動情形與技術發展持續滾動修訂。

特此感謝在手冊編製過程中提供指導與寶貴意見的專家學者與相關代表（名單如下）。

表 3、專家及學者感謝名單

領域	姓名	所屬機構 / 單位	職稱
學界	胡誌麟	國立中央大學通訊工程學系	教授
	郭財吉	國立台灣科技大學工業管理系	教授
	郭建宏	國立臺北科技大學 環境工程與管理研究所	博士後研究員
	黃世禎	國立臺灣科技大學資訊管理系	教授
	黃俊龍	國立陽明交通大學資訊工程學系	教授
	楊景元	國立臺北教育大學資訊科學系	助理教授
	劉遠楨	國立臺北教育大學資訊科學系	特聘教授兼任副校長
	陳律言	國立聯合大學能源工程學系	副教授
	陳彥宏	臺北市立大學資訊科學系	教授
產業	李永和	智慧資安科技股份有限公司	處長
	邱華洲	中華電信研究院	資深研究員
	張純嘉	公共電視國際影音平臺 資訊安全組	組長
	葉傑倫	思邁智能股份有限公司	業務副總
	劉信男	國眾電腦股份有限公司	經理
	劉尚昇	財團法人台灣商品檢測驗證中心	課長
	陳大為	財團法人環境與發展基金會	技術總監
	蔡景鷹	台灣微軟股份有限公司	副總
	戴榮	趨勢科技股份有限公司	技術總監

表 4、台灣經濟研究院編輯團隊

姓名	所屬機構 / 單位	職稱
陳彥豪	台灣經濟研究院研究一所	所長
徐幸瑜	台灣經濟研究院研究一所	副主任 / 副研究員
陳脩德	台灣經濟研究院研究一所	計畫主持人
陳秉松	台灣經濟研究院研究一所	助理研究員

此外，為充實自願性綠色軟體標章制度之研議內容，本院亦於計畫執行期間向多家軟體與資通相關企業請益與訪談，以掌握業界實務需求，並作為制度設計與內容補強之重要參考。

特別感謝下列企業單位於本案執行期間所提供的協助與意見（名單如下）。

表 5、協助本案意見蒐集與訪談會議之企業名單

所屬機構 / 單位
中華民國資訊軟體服務商業同業公會
華碩電腦股份有限公司
神通資訊科技股份有限公司
精誠資訊股份有限公司
趨勢科技股份有限公司
網擎資訊軟體股份有限公司

## 附錄二、參考文獻

- [1] Green Software Foundation. (2021). What Is Green Software? <https://greensoftware.foundation/articles/what-is-green-software>
- [2] Electricity Maps, Interactive App – Real-time electricity data [https://app.electricitymaps.com/map/live/fifteen\\_minutes?utm\\_source=chatgpt.com](https://app.electricitymaps.com/map/live/fifteen_minutes?utm_source=chatgpt.com)
- [3] French Government. (2023). Environmental Legislation and Green IT.
- [4] Green Software Foundation. (2023). Software Carbon Efficiency Rating (SCER) Specification. [https://github.com/Green-Software-Foundation/scer/blob/dev/Software\\_Carbon\\_Efficiency\\_Rating\(SCER\)/Software\\_Carbon\\_Efficiency\\_Rating\\_Specification.md](https://github.com/Green-Software-Foundation/scer/blob/dev/Software_Carbon_Efficiency_Rating(SCER)/Software_Carbon_Efficiency_Rating_Specification.md)
- [5] Google Cloud. (2023). Go Green Software. <https://cloud.google.com/resources/go-green-software/guide>
- [6] Green Software Foundation. (2024). Green Software Introduction. <https://learn.greensoftware.foundation/introduction/>
- [7] Green Software Foundation. (2024). Software Carbon Intensity (SCI) Specification Achieves ISO Standard Status, Advancing Green Software Development. <https://greensoftware.foundation/articles/sci-specification-achieves-iso-standard-status>
- [8] Gartner. (2024). Top Five Strategic Technology Trends in Software Engineering for 2024. <https://www.gartner.com/en/newsroom/press-releases/2024-05-16-gartner-identifies-the-top-five-strategic-technology-trends-in-software-engineering-for-2024>
- [9] ISO/IEC. (2024). Information technology — Software Carbon Intensity (SCI) specification (ISO/IEC 21031:2024). <https://www.iso.org/standard/86612.html>
- [10] Linux Foundation. (2024). Green Software for Practitioners (LFC131). <https://training.linuxfoundation.org/training/green-software-for-practitioners-lfc131/>
- [11] Infocomm Media Development Authority. (2024). Singapore’s Digital

Connectivity Blueprint. <https://www.imda.gov.sg/-/media/imda/files/programme/digital-connectivity-blueprint/digital-connectivity-blueprint-report.pdf>

- [12] IMDA. (2024). Sustainable Software Development Guidelines. <https://reurl.cc/1KEpoV>
- [13] Infocomm Media Development Authority. (2024). Singapore's Green Data Centre Roadmap. <https://www.imda.gov.sg/-/media/imda/files/how-we-can-help/green-dc-roadmap/green-dc-roadmap.pdf>
- [14] Microsoft. (2024). Sustainable Software Engineering Practices in Azure Kubernetes Service. <https://learn.microsoft.com/zh-tw/azure/aks/concepts-sustainable-software-engineering>
- [15] NTT Data. (2024). Contributing to Sustainability with Green Software. <https://www.nttdata.com/global/en/insights/focus/2024/contributing-to-sustainability-with-green-software>
- [16] Spanish Government. (2025). Algoritmos Verdes. <https://portal.mineco.gob.es/es-es/comunicacion/Paginas/algoritmos-verdes.aspx>
- [17] ThoughtWorks. (2024). Software Sustainability: Green Software Practices. <https://reurl.cc/6Kk3gM>
- [18] German Environment Agency. (2020). Resource and Energy-Efficient Software Products, DE-UZ 215.
- [19] O'Reilly Media, Inc. (2024). Build Green Software. <https://reurl.cc/9nv72O>
- [20] Green Software Foundation. (2021). What Is Green Software? <https://greensoftware.foundation/articles/what-is-green-software>
- [21] Green Software Foundation. (2024). Software Carbon Intensity (SCI) Specification Achieves ISO Standard Status, Advancing Green Software Development. <https://greensoftware.foundation/articles/sci-specification-achieves-iso-standard-status>
- [22] Meta AI.(2023).LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. <https://arxiv.org/abs/2305.17888>

- [23] O'Reilly Media, Inc. (2024). Build green software. <https://reurl.cc/9nv72O>
- [24] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. arXiv. <https://arxiv.org/abs/1906.02243>
- [25] Bolón-Canedo, V., & Alonso-Betanzos, A. (2024). A review of green artificial intelligence: Towards a more sustainable AI. Neurocomputing.
- [26] Google Cloud. (2025). What is MLOps?. Google Cloud. <https://cloud.google.com/discover/what-is-mlops>
- [27] 環境部 .(2025)。 碳足跡資訊網。 <https://cfp-calculate.tw/cfpc/Carbon/WebPage/InstitutionDesc.aspx>
- [28] 環境部 .( 2025)。 第一類環保標章簡介。綠色生活資訊網。 <https://greenlifestyle.moenv.gov.tw/greenLabel/GreenMarkIntroFirst>
- [29] Google.( 2025)。 打造環保永續軟體。 <https://cloud.google.com/resources/software-guide?hl=zh-TW>
- [30] 經濟部能源署能源統計專區，能源統計手冊 113 年版。 <https://www.esist.org.tw/publication/handbook?tab=%E4%BE%9B%E9%9C%80%E6%8F%90%E8%A6%81&subtab=>
- [31] Microsoft. (2025). 關於電腦記憶體 . Microsoft 支援 . <https://support.microsoft.com/zh-tw/windows/%E9%97%9C%E6%96%BC%E9%9B%BB%E8%85%A6%E8%A8%98%E6%86%B6%E9%AB%94-79a75280-86ff-457c-bfbd-8439e7f7f88b>
- [32] 教育雲線上字典 . (2025). 唯讀記憶體 . 教育百科 . <https://pedia.cloud.edu.tw/Entry/Detail/?title=%E5%94%AF%E8%AE%80%E8%A8%98%E6%86%B6%E9%AB%94&search=%E5%84%B2>
- [33] 林怡廷 . (2020 年，3 月) ，張忠謀：摩爾定律已到極限，半導體未來在系統設計。數位時代。 <https://www.bnext.com.tw/article/56678/semiconductor-moris-chang-moore%27s-law-limited->
- [34] 經濟部能源署 .(2025).113 年度電力排碳係數。 [https://www.moeaea.gov.tw/ecw/populace/content/ContentDesc.aspx?menu\\_id=26678](https://www.moeaea.gov.tw/ecw/populace/content/ContentDesc.aspx?menu_id=26678)



# 綠色軟體參考手冊 草案

Green Software Reference Handbook Draft



moda

數位發展部  
Ministry of Digital Affairs