

資訊專案文件與開源碼詮釋資料中文化
第三項
確定服務在行動設備上運作良好

執行單位：國家資通安全研究院
中華民國 112 年 9 月

Making sure your service works well on mobile

確定服務在行動設備上運作良好

Your service should work well on all mobile devices. Evaluate your options carefully to find the most suitable solution for your users.

您的服務應該在所有行動設備上都正常運行。仔細評估選擇，以找到最適合使用者的方案。

Finding the right approach for your service 為您的服務尋找適切的方案

Users expect government services to work on whatever device or browser they choose to use. 使用者期待政府服務能夠在他們採用的任何設備或瀏覽器上運作良好。

A responsive website is usually the best way to do this. Compared to native apps, responsive websites:

響應式網站通常是做到這一點的最佳方式，相較於原生應用程式，響應式設計網頁具有：

譯註：本文提到的原生應用程式(native apps)是指廣義的需下載、安裝於手機上的 app，與無須下載的 Web App 及 Progressive Web Apps(PWA)相對。而非指相對於 hybrid app 的狹義原生 app。

- do not require the user to download anything, making them easier for the user to find and use - and saving on the user's data allowance
不需要使用者先下載任何內容，這使得他們更容易取得和使用，並節省使用者網路用量。
- are usually easier and less expensive to develop, maintain and iterate because they do not need different versions for different operating systems
由於不需要針對不同的作業系統提供不同的版本，通常響應式網站的開發、維護和反覆運算更容易且成本更低。
- make it easier to measure how the service is used, compared to measuring how an app is used after it's been downloaded 相較於衡量 app 下載後的使用情況，響應式網站更容易衡量服務的使用情況。

Building a responsive website

建立響應式網站

Building a responsive website (or a responsive service that's part of GOV.UK) means using: 建立響應式網站(或屬於 GOV.UK 的響應式服務)代表：

- [responsive design](#) to make sure your website works on [different browsers and devices](#)

使用 [響應式設計](#) 確保您的網站可以在 [不同的瀏覽器和設備](#) 上運行

譯註：「不同的瀏覽器和設備」翻譯請參考中文化項目二：針對不同的瀏覽器和裝置進行設計

- [progressive enhancement](#), which will also make your service reliable 使用 [漸進增強策略](#)，使您的服務變得更可靠

譯註：「漸進增強」翻譯請參考中文化項目一：使用漸進增強策略打造有韌性的前端頁面

Building your service in this way means users get the same content and functionality, regardless of how they access your service. And it will help you to meet [government accessibility requirements](#).

以這種方式建立服務意味著無論使用者如何造訪服務，他們都可以獲得相同的內容和功能。它也將幫助您滿足[政府的無障礙要求](#)。

Building an app 開發 App

An app may be the right solution in some circumstances. For example:

在某些情況下，app 可能是正確的解決方案。例如：

- the service only works if it has a persistent presence on the user's device - for example, some apps designed to help the user live a healthier lifestyle
該服務僅在使用者的設備上持續存在時才有效 - 例如一些旨在幫助使用者以更健康方式生活的 app
- the service needs to interface with specific features of the user's device
該服務需要與使用者設備上的特定功能進行串接
- the user needs to collect and store data, but will not have consistent internet access
使用者需要收集和儲存資料，但網路並非總是保持連線狀態

If you think an app is the right option for your service, you'll need to show clear evidence as part of your [spend request](#) or internal pipeline process.

如果您認為 app 是服務的正确選擇，您需要在[支出請求](#)或內部流程中提供明確的證據。

You'll show that the need cannot be met by opening up your data via an Application Programming Interface (API) and letting the market meet any need for an app. For instance when Transport for London opened up its APIs, it resulted in the creation of Citymapper - a popular travel app.

您將需要證明，透過 API 開放資料並讓市場滿足對 app 的需求，是無法滿足這種需求的。例如，當倫敦交通局開放其 API 時，就催生了 Citymapper——一款流行的旅行 app。

Progressive Web Apps

漸進式網頁應用程式 (Progressive Web Apps, PWA)

Advances in browser technology have made it possible to do things on the mobile web that you could previously only do with native apps. Services that take advantage of these modern browser enhancements are often called mobile web apps or [Progressive Web Apps \(PWAs\)](#). You should use your [alpha discovery](#) and prototyping sessions to explore how they might help you.

瀏覽器技術的進步使得您可以在行動網路上執行以前只能使用原生應用程式(native apps)執行的操作。利用這些現代化瀏覽器增強功能的服務通常稱為行動 Web 應用程式或[漸進式網頁應用程式 \(Progressive Web Apps, PWA\)](#)。您應該利用 [Alpha 階段](#)和原型設計階段來探索它們如何為您提供幫助。

The user experience of PWAs on a handset is almost identical to a native app. But, unlike native apps, PWAs have a single codebase that works on the web like any normal website.

在手機上，PWA 的使用者體驗幾乎與原生應用程式相同。但不同於原生應用程式，PWA 擁有一個在網路上運行的單一程式庫，如同任何普通網站一樣。

A benefit of PWAs over native apps for the government is that there's no need to maintain lots of different operating system versions of the same app. This makes them relatively inexpensive to develop compared to native apps. It's also important to remember that you may still need to tailor your PWA to suit specific device features.

對於政府來說，PWA 相對於原生應用程式的好處是：無需替該程式維護多個不同作業系統版本。與原生應用程式相比，開發成本相對較低。但也請記得，您仍需要客製化您的 PWA 以滿足特定的設備功能。

If your user research is leading you to consider a native app, you should first look to see if PWAs might solve the problem.

如果您的使用者研究讓您考慮使用原生應用程式，您應該首先檢查 PWA 是否可以解決問題。

PWA functionality

PWA 的用途

With PWAs, users can access the following native APIs through their browser:

藉助 PWA，使用者可以透過瀏覽器存取以下的原生 API：

- offline mode 離線模式
- push notifications 推播通知
- Bluetooth 藍牙
- speech recognition 語音辨識
- file access 檔案存取
- geolocation 地理定位
- camera and sound recording
相機和錄音
- device motion 設備動作感測器

The site [What Web Can Do](#) is a good place to check for the current features available on the web platform.

[What Web Can Do](#) 網站是查看目前網頁平臺可用功能的好地方。

Users can choose to install PWAs on any device that supports the technology. If a user chooses to install a PWA, they typically take up less space than the equivalent native app. 使用者可以選擇在支援該技術的任何設備上安裝 PWA。如果使用者選擇安裝 PWA，它們通常比同等的原生應用程式佔用更少的空間。

PWAs cost less than native apps, since they only need to be developed once for different platforms. They are also forward-compatible with future upgrades to operating systems. PWA 的成本低於原生應用程式，因為它們只需要在不同平臺開發一次。它們也與未來作業系統的升級具有向前相容性。

[Google Chrome](#), [Mozilla Firefox](#), [Microsoft Edge](#), [Apple Safari and iOS Safari](#), and [Samsung Internet](#) support PWA functionality, which equates to [90% of global browser use](#) as of April 2019.

[Google Chrome](#)、[Mozilla Firefox](#)、[Microsoft Edge](#)、[Apple Safari and iOS Safari](#) 和 [Samsung Internet](#) 支援 PWA 功能，截至 2019 年 4 月，全球 90% 的瀏覽器都支援。

PWAs for government services are still in their early stages, and you'll need to do user research and examine your data to check if they are the best option for your use case. As with any technology, PWAs do have their own drawbacks and constraints - for example, storage on a device can be a limiting factor. For this reason, it's important to consider how using a PWA will impact your service and users on a case by case basis.

針對政府服務而言，PWA 仍處於早期階段，您需要進行使用者研究並檢查分析資料，以確定它們是否為您使用場景的最佳選擇。與任何技術一樣，PWA 確實有其自身的缺點和限制。例如：設備上的儲存空間可能是一個限制因素。因此，根據具體情況考慮 PWA 的使用將如何影響您的服務和使用者非常重要。

Things to consider before developing a PWA

開發 PWA 之前需要考慮的事項

Before you develop a PWA, you need to be sure you know:

在開發 PWA 之前，您需要確保了解以下內容：

- who your users are
您的使用者是誰
- what platforms they use to access your app
他們使用哪些平臺來存取您的 app
- how your users use any existing websites or native apps for your service
使用者如何利用任何現有網站或原生應用程式來使用您的服務
- what kind of connectivity your users have
使用者的連線方式
- how much data your intended PWA will use
您預計 PWA 將使用多少資料
- if your analytics data shows adequate support for service workers and the [web app manifest](#)
您的分析資料對 Service Worker 和 [web app manifest](#) 是否有足夠的支援

Native apps 原生應用程式

Native apps are apps for specific operating systems that users download onto their devices. Sometimes an app may need access to wider functionality than any web app can currently provide. In this case, you may decide a native app is the best solution. For example, if the offline user data needs to be encrypted - this is currently easier and more secure to do in a native app.

原生應用程式是針對特定作業系統，由使用者下載到其設備上的應用程式。有時，應用程式可能需要存取比目前任何網路應用程式所能提供的更廣泛功能。在這種情況下，您可能會決定原生應用程式是最佳解決方案。例如，如果離線使用者資料需要加密——目前在原生應用程式中執行此操作更容易且更安全。

If you're developing a native app, where possible you'll need to make sure it works the same way on a variety of operating systems - such as Android and iOS.

如果您正在開發原生應用程式，需要盡可能確保在不同的作業系統(例如 Android 和 iOS)上運作方式相同。

If you have a small, closed user group who all have the same device, you'll only need to ensure it works for one operating system. You'll still need to understand the implications of being tied to one vendor, and support any updates to the operating system.

如果您有一個小型、封閉的使用者群體，他們都擁有相同的設備，則只需確保它適用於一種作業系統。但您仍然需要了解被特定供應商綁住造成的影響，並支援該作業系統的任何更新。

What to do if you're limited by device functionality

如果受到設備功能的限制該怎麼辦

If your native app requires functionality that's only available on a certain number of devices, you'll need to treat the app as an enhancement to your service.

如果您的原生應用程式需要的功能只在某些設備上可用，您只能將此 app 視為對您的服務的一項強化。

This means offering it to users who've got those devices as the easiest way for them to complete their task. However, you'll need to provide a way for other users with different devices to complete their task too - and take reasonable steps to make it as simple as possible for them to do so.

這表示讓擁有這些設備的使用者完成任務最便捷的方式，就是提供 app 給他們。然而，您仍需要為使用不同設備的其他使用者提供方法來完成他們的任務，並採取合理的步驟讓他們盡可能簡單地完成。

相關翻譯

響應式設計

<https://alistapart.com/article/responsive-web-design>

響應式網頁設計

設計師在印刷媒體中所知道的控制，以及他們在網路媒體中經常渴望的控制，其實只是印刷頁面限制的功能。我們應該懷抱著網路並不具相同的限制這個事實，並為這種靈活性去進行設計。但首先，我們必須「接受事物的變化」。

約翰·奧索普，《網頁設計之道》

英國建築師克裡斯多福·雷恩(Christopher Wren) 曾戲稱他選擇的領域「追求永恆」，這種說法有某種吸引力：不像網路經常感覺像是為了下周而努力，而建築學在很大程度上是一門永久性學科。

建築物的基礎先決定占地面積，進而確定了其骨架，進而塑造了外觀。建築過程中的每個階段都比上一階段更加固定、不可變。創意的決策實質上塑造了一個物理空間，定義了人們在其範圍內移動的方式，可能持續數十年甚至數個世紀。

然而，網路上的工作完全是另一回事。我們的工作特點是瞬息萬變，經常在一兩年內就被精鍊或替代。不一致的視窗寬度、螢幕解析度、使用者偏好以及使用者安裝的字體等等，只是我們發布作品時需要考慮的一些難以捉摸的因素，而多年來，我們已經非常熟練地做到了這一點。

但情況正在改變，變化的速度可能比我們想像的還要快。預計行動瀏覽將在三到五年內超過桌面瀏覽。三大主流遊戲機中，有兩個具有網頁瀏覽器(其中之一相當出色)。我們正在為滑鼠和鍵盤、T9 鍵盤、手持遊戲控制器和觸控介面進行設計。簡而言之，我們面臨著比以往更多的設備、輸入模式和瀏覽器。

近年來，我遇到了越來越多要求包含「iPhone 網站」作為其專案的公司。這是一個有趣的說法：從字面上看，它說明行動 WebKit 作為瀏覽器專案的特質，以及超越桌機的強大商業案例。但作為設計師，我認為我們經常對這種明確的要求感到舒適，因為它允許我們區分正在面臨的問題。我們可以將行動體驗隔離在與「非 iPhone 網站」不同的子網域、空間之外。但接下來呢？iPad 網站？N90 網站？我們真的可以繼續承諾為每個新的使用者代理提供其客製化的體驗嗎？在某些時候，這開始讓人感覺像是一場零和遊戲。但我們和我們的設計如何適應？

動態基礎

讓我們考慮一個範例設計。我為一個虛擬的雜誌建立了一個簡單的頁面；它是一個基於在流動布局(fluid grid)上的簡單的兩欄布局，其中散布著不少動態圖像。作為非固定布局的長期支持者，我長期以來一直認為它們更具「未來性」，只因為它們與布局無關。在某種程度上，這是正確的：動態設計不對瀏覽器視窗寬度做任何假設，並且可以完美地適應具有縱向和橫向模式的設備。

had broken out. On landing at Bombay, I learned that my corps had advanced through the passes, and was already deep in the enemy's country.

victors & villains



SHERLOCK
HOLMES



DR JOHN HEMISH
WATSON



MYCROFT
HOLMES

大型的

圖像是佔空間。我們的布局雖然很動態，但不能很好地響應解析度或視窗大小的變化。

但是，無論是固定布局或是流動布局，沒有任何設計可以無縫地達到其最初的預期想像。範例設計隨著瀏覽器視窗調整大小時表現得非常好，但在較低解析度下，問題點很快就會出現。當在小於 800×600 的視窗中查看時，後面的插圖很快就會被裁剪，導致文字可能會以不合時宜的方式自動換行，並且底部的圖像變得過於緊湊而難以清晰辨認。受影響的不僅僅是解析度範圍：在寬螢幕顯示器上查看設計時，影像會迅速增大到難以處理的尺寸，從而擠佔了周圍的內容。

簡而言之，我們的動態設計在其設計的以桌面為中心的環境中運作得足夠好，但並未將最佳化效果延伸至桌面以外的地方

變得響應性

最近，一門名為「響應式架構」的新興學科開始研究如何讓實體空間對穿過的人群做出反應。透過嵌入式機器人技術和拉伸材料的結合，建築師正在嘗試設計藝術設備和牆壁結構，當人群接近它們時，它們會彎曲、收縮和膨脹。運動感測器可以與氣候控制系統配合使用，以在房間充滿人員時調節房間的溫度和環境照明。已經有公司生產出「智慧玻璃技術」，當房間的居住者達到一定的密度閾值時，該技術可以自動變得不透明，進而為他們提供額外的隱私層級。

Michael Fox 和 Miles Kemp 在他們的《互動建築》一書中將這種更具適應性的方法描述為「一個多循環系統，人們在這個系統中進行對談；持續和建設性的資訊交流。」我會強調這一點是因為我認為這是一個微妙但強大的區別：與其創造一成不變的空間來定義特定的體驗，不如建議居民和結構可以且應該相互影響。

這是我們前進的方向。我們可以將它們視為相同體驗的各個面向，而不是為數量不斷增加的網路設備量身訂製互不相關的設計。我們可以設計最佳的瀏覽體驗，但要將基於標準的技術嵌入到我們的設計中，使它們不僅更加靈活，而且更適合它們的媒體呈現。簡而言之，我們需要練習響應式網頁設計。但如何進行呢？

滿足媒體詢問

自 CSS 2.1 以來，我們的樣式表(style sheets)已經透過媒體類型實現到了某種程度的設備感知能力。如果您曾經編寫過列印樣式表，那麼您已經熟悉這個概念：

```
<link rel="stylesheet" type="text/css" href="core.css"
  media="screen" />
<link rel="stylesheet" type="text/css" href="print.css"
  media="print" />
```

為了讓我們設計的不僅僅是格式整齊的頁面列印輸出，CSS 規範為我們提供一系列[可接受的媒體類型](#)，每種媒體類型都針對特定類別的網路就緒設備。但大多數瀏覽器和設備從未真正接受該規範的精神，導致許多媒體類型的[實現不完美](#)，或完全被忽略。值得慶幸的是，W3C 建立了 [media query](#) 作為 CSS3 規範的一部分，改進媒體類型的承諾。media query 不僅使我們能夠針對某些設備類別，而且能夠實際檢查渲染我們工作的設備的物理特性。例如，隨著最近行動 WebKit 的興起，media query 成為一種流行的使用者端技術，用於向 iPhone、Android 手機及其同類產品提供客製化的樣式表。為此，我們可以將查詢合併到 link 樣式表的 media 屬性中：

```
<link rel="stylesheet" type="text/css"
  media="screen and (max-device-width: 480px)"
  href="shetland.css" />
```

該查詢包含兩個組成部分：

1. 媒體類型(screen)，以及
2. 括號內的實際查詢，包含要檢查的特定媒體特徵(max-device-width)，後面接著目標值(480px)。

簡單來說，我們正在詢問設備的水平解析度 (max-device-width) 是否等於或小於 480px。如果測試通過——換句話說，如果我們在 iPhone 等小螢幕設備上查看我們的成果——那麼該設備將載入 shetland.css。否則，將 link 被完全忽略。

設計師過去曾嘗試過解析度感知布局，主要依賴於 JS 驅動的解決方案，例如 [Cameron Adams 的傑出腳本](#)。但是 media query 規範提供了許多遠遠超出螢幕解析度的媒體功能，大大擴充我們可以透過查詢進行測試的範圍。此外，您可以透過 and 關鍵字將多個屬性值連結在單一查詢中測試：

```
<link rel="stylesheet" type="text/css"
  media="screen and (max-device-width: 480px) and (resolution: 163dpi)"
  href="shetland.css" />
```

此外，我們不限於在我們的 link 中合併 media query。我們還可以在 CSS 中加入媒體查詢，或將其作為 @media 規則的一部分：

```
@media screen and (max-device-width: 480px) {
  .column {
    float: none;
  }
}
```

或作為 @import 指令的一部分：

```
@import url("shetland.css") screen and (max-device-width: 480px);
```

但在每種情況下，效果都是相同的：如果設備通過了我們的 media query 的測試，相關的 CSS 就會應用於我們的標記。簡而言之，media query 就是[條件註解](#)。當布局超出其

最初的理想解析度時，我們可以透過手動來修正布局中的問題，而不是針對特定瀏覽器的特定版本。

適應、回應和克服

讓我們將注意力轉向頁面底部的圖像。在預設布局中，相關 CSS 目前如下所示：

```
.figure {  
  float: left;  
  margin: 0 3.317535545023696682% 1.5em 0; /* 21px / 633px */  
  width: 31.121642969984202211%; /* 197px / 633px */  
} li#f-mycroft,  
li#f-winter {  
  margin-right: 0;  
}
```

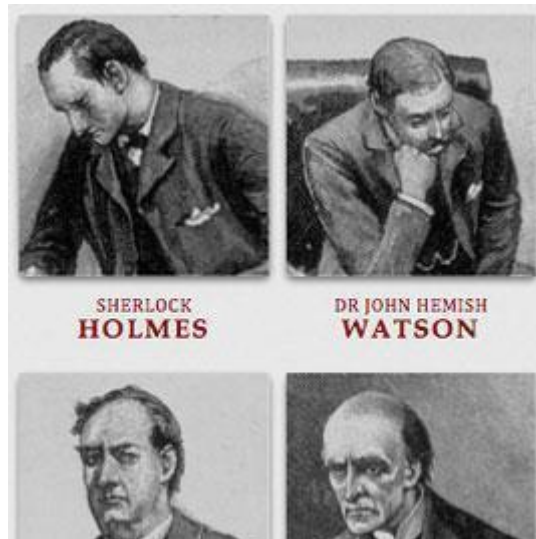
我已經省略許多排版屬性，將重點著重於布局：每個 `.figure` 元素的大小大約是所在欄 (column) 的三分之一，每列 (row) 末尾的兩張圖片 (li#f-mycroft, li#f-winter) 的右側邊距為零。這種方法效果相當好，直到視窗明顯比我們原來的設計更小或更寬。透過 media query，我們可以應用特定解析度的定點修復，調整我們的設計以更好地回應顯示效果的變化。

首先，一旦視窗低於某個解析度閾值 (例如：600 px)，我們就會對頁面進行線性化處理。因此，我們在樣式表的底部建立一個新 `@media` 區塊，如下所示：

```
@media screen and (max-width: 600px) {  
  .mast,  
  .intro,  
  .main,  
  .footer {  
    float: none;  
    width: auto;  
  }  
}
```

如果您在現今的桌面瀏覽器中查看 [我們的更新頁面](#)，並將視窗大小縮小到低於 600px，media query 將停用設計主要元素上的浮動，而在文件流程中將每個區塊堆疊在一起。因此，我們的小型化設計效果很好，但圖片仍然沒有聰明地縮小。如果我們引入另一個 media query，我們可以相應地改變它們的布局：

```
@media screen and (max-width: 400px) {  
  .figure,  
  li#f-mycroft {  
    margin-right: 3.317535545023696682%; /* 21px / 633px */  
    width: 48.341232227488151658%; /* 306px / 633px */  
  } li#f-watson,  
  li#f-moriarty {  
    margin-right: 0;  
  }  
}
```



我們的圖片可以響應地改變其布局，以更好地適應較小的顯示器。

不要介意難看的百分比；我們只是重新計算流動布局的寬度以適應新的線性化布局。簡而言之，當視窗的寬度低於 400px 時，我們將從三欄布局轉變為兩欄布局，使圖像更加突出。

實際上，我們也可以對寬螢幕顯示器採取相同的方法。對於更大的解析度，我們可以對影像進行六橫排處理，將它們全部放在同一列中：

```
@media screen and (min-width: 1300px) {
  .figure,
  li#f-mycroft {
    margin-right: 3.317535545023696682%; /* 21px / 633px */
    width: 13.902053712480252764%; /* 88px / 633px */
  }
}
```

現在，我們的影像在解析度範圍的兩端都能很好地呈現，最佳化了它們的布局以適應視窗寬度和設備解析度的變化。



透過在新的 media query 中指定更寬的範圍 min-width，我們可以將圖片轉換為單列布局。

但這只是開始。根據我們嵌入在 CSS 中的 media queries，我們可以改變的不僅僅是一些圖像的位置：我們可以根據不同的解析度範圍引入新的替代布局，也許使導覽在寬屏視角中更加突出，或者在較小的顯示器上將其重新放置在 logo 上方。



透過響應式設計，我們不僅可以在較小的設備上線性化內容，還可以在各種顯示器上最佳化的呈現。

但響應式設計並不僅限於布局變更。Media queries 使我們能夠在頁面重塑時進行一些極其精確地微調：我們可以為較小的螢幕增加目標區域連結，以在觸控設備上更好地遵守費茨定律；選擇性地顯示或隱藏可以增強頁面導覽的元素；我們甚至可以練習響應式排版來逐漸改變文字的大小和行距，進而最佳化提升文字顯示器的閱讀體驗。

一些技術說明

值得注意的是，Media queries 在現今瀏覽器中享有令人難以置信的強大支援。桌面瀏覽器(例如：Safari 3+、Chrome、Firefox 3.5+ 和 Opera 7+)都可以原生解析 media query，最新的行動瀏覽器(例如：Opera Mobile 和行動 WebKit)也是如此。當然，這些桌面瀏覽器的舊版本不支援 media query。儘管 Microsoft 承諾在 [IE9 中支援 media query](#)，但 Internet Explorer 目前不提供原生實作。

但是，如果您有興趣實現舊版瀏覽器對 media query 的支持，那麼還有一個 JavaScript 的替代方案可以考慮：

- [2007 年的 jQuery 外掛程式](#) 提供了一定程度的 media query 支援，當附加到不同的 link 元素時僅實現 min-width 和 media-wide 屬性。
- 最近，[css3-mediaqueries.js 也被](#)發布，該函式庫承諾透過 @media 區塊包含時「使 IE 5+、Firefox 1+ 和 Safari 2 透明地解析、測試和應用 CSS3 Media Queryies」。儘管很大程度上是 1.0 版本，但我個人發現它非常強大，我計畫持續關注它的發展。

但如果您對使用 JavaScript 不感興趣，那也是完全可以理解的。[然而，這強化了在靈活的網格](#)之上建構布局的理由，確定您的設計在瀏覽器不支援 media query 和設備中具有一定程度的靈活性。

前進的道路

流暢的網格、靈活的圖像和 media query 是響應式網頁設計的三個技術要素，但它也需要不同的思維方式。我們可以使用 media query 來逐步增強我們在不同瀏覽環境中的成果，而不是將我們的內容隔離為不同的、特定於設備的體驗。這並不是說不存在針對特定設備的單獨網站的商業案例。例如，如果您的行動網站的使用者目標比桌面網站的使用者目標範圍更有限，因此為每個網站提供不同的內容可能是最好的方法。

但這種設計思維不要成為我們的預設思考方式。現在，我們比以任何時候都更需要設計出能夠讓人以不同體驗梯度來瀏覽的作品。響應式網頁設計為我們提供了一條前進的道路，最終使我們能夠「依事物的潮起潮落進行設計」而設計。

相關翻譯

政府的無障礙要求

無障礙和輔助數位化 讓您的服務具有無障礙性：簡介

<https://www.gov.uk/service-manual/helping-people-to-use-your-service/making-your-service-accessible-an-introduction#meeting-government-accessibility-requirements>

您的服務必須可供所有需要的人使用，含僅供公務員使用的服務。如果未使您的服務具有無障礙性，您可能會違法。

在設計或建立任何東西之前，您需要考慮使用者如何存取和使用您的服務。無障礙性與輔助數位支援是不同的概念，[輔助數位支援](#) 意味著幫助數位技能較低或網路存取受限的使用者。

滿足政府無障礙要求

為了滿足政府的無障礙要求，數位服務必須：

- 至少符合 [Web 內容無障礙指南 \(WCAG 2.1\)](#) 的 AA 級
- 研究最常用的 [輔助科技](#) - 含螢幕放大鏡、螢幕報讀器和語音辨識工具
- 將身心障礙人士納入 [使用者研究](#)
- 具備 [無障礙性聲明](#)，解釋服務的無障礙性——您需要在服務進入公開 beta 測試時發布此聲明

政府數位服務(GDS)正在研究如何評估 [WCAG 2.2 中的新成功標準](#)，並將於 2024 年 10 月開始監控額外的標準。在 2024 年 10 月之前，英國政府將遵循 [WCAG 2.1 AA 級](#) 持續監控網站和應用程式的無障礙性。

如果您的服務符合政府的無障礙性要求，那麼您還將滿足適用於公共部門網站和行動應用程式的無障礙性法規。

該法規的全名為《2018 年公共部門機構(網站和行動應用程式)(第 2 號)無障礙法規》(the Public Sector Bodies (Websites and Mobile Applications) (No. 2) Accessibility Regulations 2018)。

您可以透過閱讀 [最新指南](#) 來了解有關規定的更多資訊。

從一開始就考慮無障礙性

在英國，[五分之一的人患有身心障礙](#) - 這可能是視覺、聽覺、運動(影響精細運動)或認知(影響記憶和思考)方面的障礙。

無障礙的概念不僅適用於身心障礙人士——所有使用者在不同時間和不同情況下都會有不同的需求。某人使用服務的能力可能會受到以下因素的影響：

- 地點——可能位於吵鬧的咖啡館、陽光明媚的公園或 wifi 速度較慢的區域
- 健康——他們可能感到疲倦、中風復健或手臂骨折
- 設備——他們可能使用手機或較舊的瀏覽器
- 無障礙性是為了確定您的服務可以被盡可能多的人使用。從一開始就考慮這一點將有助於您：
- 確定沒有任何人被排除在外

- 儘早發現您的服務是否有部分尚未無障礙化——如果您儘早發現問題，解決問題的成本通常會更低

無障礙性的預算

將提供無障礙服務的成本納入您的預算中。例如，您的團隊可能需要培訓或時間來學習無障礙相關知識，並且您需要支付無障礙性稽核的費用。

如果您的團隊中有無障礙專家，您可以避免稽核成本(如果他們可以做無障礙稽核)。

無障礙性是整個團隊的責任

無障礙性不是一個人的責任。團隊中的每個人都有責任讓您的服務具備無障礙性。

重要的是，每個團隊成員都要了解如何避免意外地使東西變得難以使用(inaccessible)。這含開發人員、內容設計師和互動設計師。

使用者研究人員和測試人員可以幫助發現無障礙性問題，以便團隊的其他成員可以排除問題。

產品和交付經理也應該了解無障礙性，這樣他們就可以確保從一開始就考慮無障礙性並有效地將其建立到服務中。

在 [Department for Work and Pensions\(DWP\)針對特定工作角色的指南](#) 中可以了解更多關於每個專業領域如何幫助建立無障礙性服務。

與身心障礙使用者一起研究

在進行研究時，您需要含身心障礙人士和使用輔助科技的人。

您可能需要支付以下費用：

- 協助您尋找參與者的[招募機構](#)
- 幫助參與者參與研究的費用——這可能含助手、計程車或手語翻譯等幫助溝通的人員

了解更多有關[招募使用者研究參與者的](#)資訊。

在探索(discovery)階段採取的無障礙性措施

在探索階段，您應該了解有視覺、聽覺、運動和認知障礙的人可能如何使用您的服務，以及他們可能遇到的問題。

您甚至可以在進行任何使用者研究之前透過以下方式執行此操作：

- 當您探索要解決的問題時，[對無障礙性的意義有一個清晰的理解](#)。
- 嘗試了解[使用者可以擁有的能力範圍](#)。
- 閱讀[身心障礙使用者的個人資料](#)，以了解無障礙性如何影響個人使用者。

您還應該加入[無障礙性社群](#)。

在 alpha 階段的無障礙性措施

在 alpha 階段，您應該考慮您的設計是否符合 [WCAG 2.1 設計原則](#)——這將幫助您滿足所有使用者的需求。

您還應該開始考慮您的 [無障礙性稽核](#)。儘管您在測試版之前不會需要進行稽核，但可能需要一些時間來安排稽核。您的部門中可能有人具備執行此操作所需的技能和稽核經驗。如果沒有，您將需要聘請外部無障礙專家。

您可以要求您的稽核員在 alpha 測試期間花幾天時間，審查您的設計和 [原型](#) 是否有潛在的無障礙性問題。

您還可以透過 [與身心障礙使用者進行研究會談](#) 來改進您的設計。

在 beta 階段的無障礙性措施

在 beta 階段，您需要開始 [測試無障礙性](#)，並進行無障礙性稽核。

每次開發新功能時，結合執行手動和自動測試意味著您可以解決以後修復成本高昂的問題。

檢查您的團隊可以使用哪些測試工具和軟體，以確認是否需要購買。

[與身心障礙使用者一起進行研究會談](#) 有助您檢查正在建立的內容具備無障礙性。

當您開始公開 beta 測試時，您需要發布一份無障礙性聲明。

進行無障礙性稽核

在進入公開 beta 測試之前，您必須進行無障礙性稽核，證明您的服務可與輔助科技配合使用並符合 [WCAG 2.1 AA 等級](#)。它將提供您發布無障礙聲明所需的資訊。

安排稽核可能需要一些時間，因此請在 alpha 階段開始規劃。

了解 [如何進行無障礙性稽核](#)，含如何尋找供應商。

發布無障礙聲明

獲得稽核結果後，您應該準備好草稿並發布您的無障礙聲明。

無障礙性聲明必須解釋：

- 您的服務有多方便(有多無障礙)
- 服務是否存在無障礙性問題
- 您計畫如何解決這些問題

您可以使用 [無障礙聲明](#) 範例來幫助您撰寫。您的無障礙性聲明應該讓使用者易於找到。這意味著它應該在每個頁面上都可獲得。

對於行動應用程式，無障礙性聲明應可從以下任一方式取得：

- 您的網站
- 下載行動應用程式時提供的資訊
- 在行動應用程式內

在服務上線階段的無障礙性措施

一旦服務上線，繼續測試和研究服務非常重要。

檢查您新增的功能是否符合無障礙性要求，並繼續與身心障礙使用者進行研究。

如果您對服務進行重大變動，您可以進行另一次稽核，以確保您仍符合無障礙性要求。

您的無障礙性聲明應至少每年審查一次，如果您所做的變動影響了網站的無障礙性，則應更新無障礙聲明。

讓您的服務的非數位部分變得無障礙

您應該確保服務中的非數位部分也是無障礙性的。例如，您應該確定為失聰或有語言障礙的使用者提供聯絡資訊(透過簡訊、電子郵件或親自與英國手語翻譯或唇語閱讀人員聯繫)。

如果您的服務會需要寄出信件，請確定您也提供了無障礙的替代方案(例如大字體、點字或音訊 CD)。

相關翻譯

支出請求

支出控制：檢查您的服務是否需要才花錢

<https://www.gov.uk/service-manual/agile-delivery/spend-controls-check-if-you-need-approval-to-spend-money-on-a-service>

如果您想花錢購買數位或技術服務，您可能必須獲得中央數位和資料辦公室(CDDO)的核准。

這個過程被稱為「支出控制」。

本指南涉及舊的支出控制流程。如果您的部門正在使用新的[支出控制管道流程](#)，請遵[循流程](#)。請諮詢您的數位或技術主管以找出答案。

為什麼有些服務必須獲得支出核准

CDDO 要求服務獲得支出核准，以確保它們的建置符合[技術實務守則](#)。

您應該在開始建立服務之前閱讀守則，並在做出技術決策時使用。

支出控制流程的存在是為了確保：

- 您已經考慮了使用該服務的人的需求
- 您獲得最佳利益
- 您僅在符合政府數位和技術策略的計畫和專案上花費
- 私人企業和公眾可以了解政府如何做出在數位專案或技術上花錢的決策
- 在花費資金之前，您可以發現並解決您的服務中的任何問題
- 您使用[敏捷方法開發服務](#)

CDDO 如何決定哪些服務需要核准

CDDO 根據以下 3 個因素決定您的服務是否需要核准：

- 您所在的部門是否為中央政府部門——[下載公共部門分類指南](#)以了解詳情
- CDDO 將您的服務歸類為數位服務還是技術服務
- 您希望花費多少的金額

何時您的服務被視為數位服務

如果您的服務是線上並且將由非政府人員使用，CDDO 會將您的服務視為數位服務。

非政府人員如：

- 公民
- 企業
- 其他非政府組織

以下是數位服務的範例：

- 交換資料服務(使用者向政府提供個人詳細資料的服務，例如申請護照)
- 資通服務(使用者無需提供個人詳細資訊的服務，例如稅務或福利計算器(a tax or benefit calculator))
- 網路應用程式
- 行動應用程式
- 網站
- 企業間網路(Extranet)

如果您在以下方面建立內容(而不是服務)，CDDO 不會將您的服務視為數位服務：

- 現有站點
- (企業)內部網路(Intranet)
- [廣告、行銷和通訊控制](#)下列出的領域

您的數位服務何時需要獲得核准

您必須獲得核准來使用金費以開發新的數位服務，並且該服務符合以下一個或多個條件：

- 符合大眾利益
- 使用 [身分保證](#)
- 需要網域註冊
- 使用外部使用(external-facing)的數位交換資料
- 具有網站或行動應用程式

對於任何其他數位服務，您必須獲得核准才能花費超過 100,000 英鎊。

您的服務何時會被視為技術服務

CDDO 會將您的服務視為技術服務，如果符合以下所有條件：

- 僅供內部使用(internal facing)
- 由政府為政府提供

您何時需要技術核准

您必須獲得核准才能在技術上花費超過 100 萬英鎊。

如果您在政府共用服務(Government Shared Services)提供的任何技術上花費超過 100 萬英鎊，您還必須透過 SSfG-Secretariat@cabinetoffice.gov.uk 與他們聯繫以討論支出。這含以下技術：

企業資源規劃(ERP)系統

人力資源系統

財務或會計系統

採購系統

「技術」是指您計畫建立的服務的所有部分，含以下類型的支援：

託管聯絡人(contacts in hosting)

資料中心

語音和影片

安全性

協力廠商支援

技術支出計入您的總支出

如果 CDDO 將您的專案歸類為技術專案，則您在以下方面花費的任何資金都將計入您的總支出的一部分，用於在檢查是否需要支出核准：

- 合約修改、延期或續約
- 授權
- 透過現有框架進行支出，例如雲端或商業框架
- 可行性或概念驗證研究

- 試驗性專案
- 專案(或專案的任何部分)和計畫——[找出專案和計畫之間的區別](#)
- 任何更改、增強、維護或更新
- 通用基礎設施解決方案，含語音和資料通訊
- [PSN](#)，無論是固定的還是移動的
- 您已分解為較小專案的計畫，這些專案單獨低於閾值，但總體高於閾值的專案也屬其中

輔助數位支出計入您的總支出

您應該考慮將以下類型的[輔助數位支援](#)作為支出的一部分：

- 面對面的支援
- 電話支援，例如呼叫中心
- 協力廠商仲介支援，例如服務整合商、顧問公司或合作夥伴
- 允許使用者存取數位服務的其他管道，例如信件

通常不會獲得核准的專案

CDDO 通常不會核准您在以下活動上花費資金：

- 簽訂價值超過 1 億英鎊的 IT 合約
- 在您與政府同一部門簽訂了為您提供服務的合約後，與一家公司簽訂了為您提供系統整合的合約——了解為什麼[政府放棄了塔式模式](#)
- 自動續約
- 提供持續超過 2 年的新託管合約
- 建立原生或混合式應用程式

相關翻譯

漸進式網頁應用程式 (Progressive Web Apps, PWA)

漸進式網路應用程式(PWA)為英國政府帶來了新的行動應用機會

<https://technology.blog.gov.uk/2018/03/26/progressive-web-apps-bring-us-new-mobile-opportunities/>

當公共部門設計服務時，通常需要考慮行動應用程式。近年來都要先為網站建立服務，然後在某些有限的情況下，再建立獨立的原生行動應用程式。

漸進式網路應用程式 (Progressive web app, PWA) 的崛起意味著英國政府不再需要將這些不同的平臺開發工作視為孤島(Silo)。

手機上 PWA 的使用者體驗幾乎與原生應用程式相同，但與原生應用程式不同的是，PWA 具有像是一般網站一樣執行的唯一的原始碼。與原生應用程式相比，PWA 對政府的一個顯著好處是其開發成本相對較低，因為英國政府不必維護相同應用程式的許多不同版本。

目前，Android 在 PWA 市場處於領先地位，但 iOS 將在今年(2018)夏天晚些時候迎頭趕上。Gartner 表示 PWA 將澈底顛覆行動應用程式領域，而且它成為網路互動的新標準只是時間問題。

因此，如果英國政府著眼於更大的前景(而不僅僅是 PWA 和原生應用程式之間的權衡)，新技術將為數位服務帶來新的標準，這將帶來行動化的好處。PWA 為政府帶來了可能的轉變。現在是為行動服務使用者帶來一致體驗的時候了。

為什麼英國政府應該擁抱 PWA

PWA 的成長可以透過以下事實來解釋：桌面和行動瀏覽器已經採用了 service worker(服務程式)，讓網站的行為就像應用程式一樣。當應用程式伺服器未連接到網路時，service worker 充當應用程式伺服器的代理，快取使用者所需的幾乎所有內容。service worker 允許的推播通知意味著在網路連線恢復時，所有新資料都會傳遞到應用程式。在行動設備上離線工作的能力對於一些政府的國際援助工作尤其重要，但也適用於英國的服務，因為英國的使用者可能並不總是擁有最好的網路覆蓋範圍，例如在農村地區。

除了低開發維護成本(因為 PWA 適用於所有設備)和離線工作的能力之外，該技術還為英國政府帶來了更多好處。這些在網上都有詳細介紹，因此不會在這篇文章中涵蓋每一點。總而言之，雖然 PWA 也可以快速為使用者安裝(您不必造訪應用程式商店並嘗試找到合適的應用程式)，它們利用使用者設備和瀏覽器上所有新功能，自動使它們對於政府開發人員來說，維護起來更容易、更便宜。作為政府，英國政府不斷尋找讓服務更無障礙的方法。PWA 為英國政府帶來了一個新的機會。

發布 PWA 指南

在這個新技術範例中，英國政府需要像平常一樣嘗試透過網路進行交付，但當對行動技術有明確需求時，應該考慮使用 PWA，且只把原生應用程式當成最後手段。英國政府最近更新了[服務手冊中有關行動技術的指南](#)，以適應這種方法。

正如約翰·曼佐尼([John Manzoni](#))最近所說，重要的是繼續尋找新的方法來更有效地開發數位服務，採取創新的技術和數據，以便英國政府能夠更快地前進。透過為政府提供有關 PWA 的正確指南，英國政府可以處於提供行動數位服務的最前沿，以便服務團隊清楚使用情境和優勢。

最近，我們一直在與主要行動設備供應商進行交流，試圖找出哪些情況仍然需要使用原生應用程式，我們計畫對剛剛發布的《服務手冊》指南進行修訂，以明確這一點。。

一些平臺似乎可以幫助設計和建立 PWA，我們知道其中一個是開源的。我們還希望鼓勵政府中的其他人，如果他們認為仍然有明顯需要使用原生應用程式而不是 PWA，請與我們聯絡。

[相關翻譯](#)

Alpha 階段探索

Alpha 階段如何進行

<https://www.gov.uk/service-manual/agile-delivery/how-the-alpha-phase-works>

Alpha 是您嘗試不同解決方案來解決從探索(discovery)階段中發現到的問題。

在 alpha 階段，請花費時間建立原型並測試不同的想法。不要害怕挑戰目前的做法：alpha 階段是探索新方法的機會。

您不必對整個更廣泛的使用者旅程(wider user journey)進行原型設計。

譯註：更廣泛的旅程(wider journey)指的是整個使用者體驗或服務所涵蓋的更大範圍。您甚至可能不想對您正在處理的所有事務或元素進行原型設計：通常只需專注於您認為最具挑戰性的領域是有合理的。就可以完成測試風險最大的假設所需最低限度的工作。

Alpha 服務不應供公眾使用。

使用您想嘗試的任何線上解決方案，建立足夠複雜的東西來讓您測試不同的想法，而非正式環境品質的程式碼。預期在 alpha 結束時，您可能會拋棄任何程式碼以及測試過的想法。

譯註：正式環境品質的程式碼意即不複雜的程式碼，正式環境的程式碼通常會要求越簡單越乾淨越不容易出錯。

到 alpha 階段結束時，您應該能夠決定您測試過的哪些想法值得推進到 beta 版。

Alpha 階段通常持續 6 到 8 週。這意味著您應該在 alpha 階段開始後的兩週內[預約進行 alpha 評估](#)。

在 alpha 階段，考慮一下[您的團隊需要哪些人](#)是很有幫助的。

在您的探索(discovery)階段結束時，如果您認為不值得進行 alpha 測試，這是可以接受的。

您可以[下載一張海報，解釋 alpha 階段的用途](#)。

專注於測試您最高風險的假設

Alpha 階段的一個關鍵部分是[識別並測試風險最高的假設](#)。這些風險是什麼取決於您正在建立的服務。

以您想要解決的問題可能是減少孤獨感和孤立感為例，如果您認為線上資通服務可能有助於解決該問題，那麼您可能會優先進行研究，以了解您試圖幫助的人們目前如何接收資訊；如果他們根本不使用網際網路，他們就不太可能使用您的服務。

或者，能夠與現有技術整合可能是優先事項。[致力於投票登記](#)的團隊意識到他們需要建立一個與 400 多個地方議會的登記系統整合的 API。如果不可能，這項服務就無法發揮作用——這就是他們在 alpha 階段關注的重點。

同樣地，如果您可能的解決方案之一依賴於尋找解決法律限制的方法，您可能需要花費部分 alpha 階段時間來確定其可行性。

在 alpha 階段達到標準

在 alpha 階段，有一些標準您會特別需要密切關注。

為使用者解決整體問題

[服務標準](#) 的第 2 點表示，您需要努力為使用者解決整個問題。

在 alpha 版本中，這可能意味著顯示：

- 您如何知道您的旅程範圍是正確的
- 您已經了解了您的服務所涉及的更廣泛的使用者旅程
- 您對使整個旅程盡可能正常運作所需發生的事情有一個概念(特別是，您能夠談論屬於同一旅程的其他服務，以及更改這些服務所涉及的機會和挑戰)
- 您正在公開工作(Working in the open)，並已開始與負責旅程其他部分的團隊建立關係(如果有必要的話)
譯註：公開進行工作(Working in the open)強調透明度和開放性，通常含分享和公開許多相關訊息，例如項目進展、決策過程、程式碼和文件。
- 您了解影響您的服務的任何法律、合約或技術限制
- 您計畫將使用者向政府提交相同資訊的次數降至最低

正確確定您的旅程的範圍

正確確定交換資料範圍可能是使其易於使用的最重要部分。使用 alpha 階段來探索[從使用者的角度來探索具有意義的範圍](#)。

與使用者更廣泛的旅程結合

並非所有交換資料都是更廣泛使用者旅程的一部分。但如果您的交換資料是這樣，您應該在探索階段中探索更廣泛的旅程。

您可以將粗略的旅程地圖(或類似的工具)帶入您的 alpha 評估，顯示[您的服務在使用者旅程中的位置](#)、涉及的不同組織以及人們存取不同管道。

如果您風險最高的假設之一是，可以對其他服務進行更改以便為使用者提供簡單、直觀的旅程，請在 alpha 階段，透過與負責這些其他服務的人員交談來測試這一假設。在 alpha 版本結束時，確保您清楚可以更改哪些內容以及進行這些更改的難度或成本。

如果您面臨跨政府或更廣泛合作的任何阻礙，您需要展示您正如何解決這些問題，例如透過與潛在合作夥伴建立關係。

無論哪種方式，您都應該與與您在同一領域工作的其他人交談並建立關係。一個好的起點是檢查[您工作的區域是否有服務社群](#)。

您還應該在 alpha 測試過程中與 [GOV.UK 內容團隊取得聯繫](#)。他們將幫助您確定您的交換資料應如何與 GOV.UK 結合。

處理限制條件

在 alpha 階段來探索法律、合約或 [舊有技術](#)(legacy technology) 中不可撼動的限制條件，這些條件將影響您計畫建立的服務。在 alpha 階段結束時，您應該能夠解釋：

- 如何在這些限制條件下建立滿足使用者需求的服務
- 如果這些限制條件是可以長期消除的(例如，透過改變技術平臺或以不同的方式與供應商簽訂合約)，組織的計畫如何達成消除這些限制

公開工作(Working in the open)

在 alpha 階段，您應該繼續公開談論您正在建立的原型以及您從中學到的東西。例如，透過部落格文章和公開展示和講述。作為其中的一部分，請考慮如何與作業交付同事分享您所學到的知識。

盡可能重複使用使用者資訊

通常的情況是，在與相關服務互動時，使用者已經向政府提供了您所需要的相同資訊。

使用 alpha 階段來調查是否可以重複使用該數據，以便使用者在從一個任務轉移到另一個任務時不必多次提供相同的資訊。除非有公共政策、信任或法律原因不共用數據，否則您將需要展示您正在朝向共用數據方向努力。

提供跨不同管道的整合體驗

服務標準的第 3 點表示，您的服務需要在使用者可能用來存取它的所有管道上正常運作。

這涉及了解您的服務的線上和離線部分如何連接在一起，以及使用者因此遇到的任何痛點。

在 alpha 階段，您可以透過以下方式努力實現這一目標：

- 在 alpha 階段實驗和 [使用者研究](#) 中包含離線元素，如字母，特別是在涉及風險假設的情況下(例如，您將能夠更改字母的內容)
- 考慮在協力廠商或非政府組織內開始的使用者旅程，例如轉介
- 邀請營運交付同事參與 alpha 階段——他們將對最高風險的假設有一個非常有用的視角

確保每個人都可以使用您的服務

對於線上原型，您無法在正式環境的程式碼中測試所有的無障礙技術。但在 alpha 階段，您應該能夠展示您：

- 了解 [WCAG 無障礙性原則](#)- 這將幫助您識別和測試您的服務可能面臨的任何特定無障礙性挑戰
- 將身心障礙人士納入您的使用者研究中
- 您不必在 alpha 階段進行 [無障礙性稽核](#)，但值得開始考慮它，因為他們可能需要時間來安排。如果您在 alpha 期間指定稽核員，他們可以審查您的 alpha 設計或原型是否存在潛在的無障礙性問題，從而為您提供充足的時間來解決問題。
- 不管怎樣，在 alpha 版本結束時，您應該制定一個計畫來解決 beta 版的無障礙性問題。
- 您還應該能夠展示您已經考慮了更廣泛意義上的包容性。這意味著：

- 您已經考慮過特定人群在存取該服務時是否可能會遇到痛點(例如，如果您要求提供某人的永久位址，而您的使用者含無家可歸者，您就需要證明您已經制定了計畫來防止他們被排除在外)
- 含在使用者研究中對數位化信心較低的人
- 您已經開始考慮如何為需要線上幫助的人設計輔助數位支援模型

決定是否進入 beta 階段

當您擁有足夠強大的原型來幫助您決定是否進入 beta [階段](#)時，alpha 階段就完成了。要進入 beta 階段，您需要確定：

- 您可以創造出滿足使用者需求並且具有成本效益的東西
- 您將擁有交付所需內容所需的預算和人員——這含研究預算

您應該能夠使用您在探索階段結束時確定的成功指標來解釋您是如何做出此決定的。如果您的 alpha 階段結束了，但您不確定自己可以做這些事情，您可以完全停止或決定重複探索階段或 alpha 階段。

當您確信想要進入 beta 階段時，您的交付經理和服務所有者應該開始製定測試版業務提案。[這應該含您的測試團隊中需要哪些人的資訊](#)。

Alpha 需要考慮的其他事項

在 alpha 階段，您還需要考慮其他一些事情。您需要表明您已經開始考慮：

- 您希望在 beta 階段選擇的程式設計工具類型，以及為什麼您認為它有這個價值
- 您將如何識別對服務的威脅、如何處理它們以及如何及時了解威脅
- 您將如何開源程式碼——以及如何透過離線管道分享您的方法和流程
- 您是否要使用通用平臺
- 如果您的服務的線上部分出現技術問題，您的使用者將受到怎樣的影響

您還應該繼續完善用於[衡量服務成功程度的指標](#)。

如果您建立了任何新的設計模式 - 或學到了跟現有設計模式有關的任何有用資訊——請透過 [GOV.UK 設計系統](#) 分享您的心得