

# บทที่ 1

## พื้นฐาน Ionic Framework

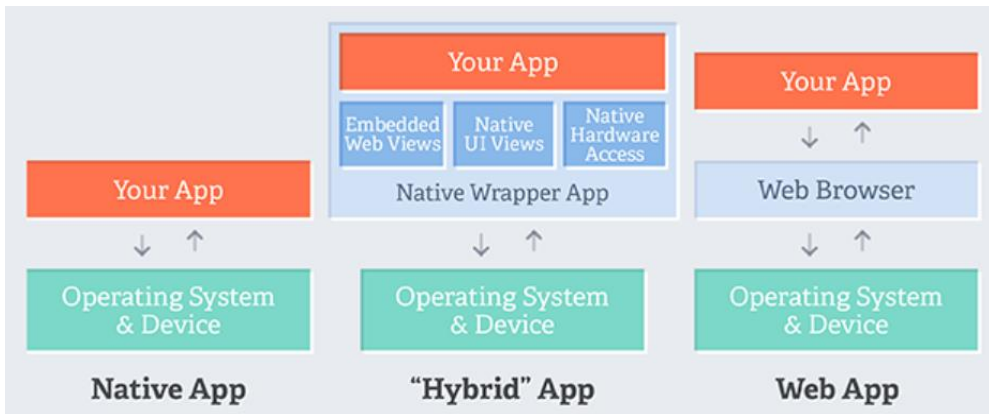
โมบายแอปพลิเคชัน (Mobile application) ในปัจจุบันได้รับความนิยมเป็นอย่างมาก เนื่องจากเป็นโปรแกรมขนาดเล็กที่ติดตั้งใช้งานบนอุปกรณ์เคลื่อนที่ ทำให้สามารถเข้าถึงข้อมูลได้ง่าย สะดวก รวดเร็ว เทคโนโลยีที่เกี่ยวข้องกับการพัฒนาโมบายแอปพลิเคชัน มีหลากหลาย ขึ้นอยู่กับรูปแบบของแอปพลิเคชันที่ต้องการรวมถึงรูปแบบการใช้งาน ในบทนี้จะกล่าวถึงประเภทของโมบายแอปพลิเคชัน รวมถึงไปถึงเครื่องมือสำหรับการพัฒนาโมบายแอปพลิเคชันที่สะดวก รวดเร็ว และเป็นที่นิยมในปัจจุบัน

## ประเภทของโมบายแอปพลิเคชัน

แอปพลิเคชันที่ใช้งานบนอุปกรณ์เคลื่อนที่ในปัจจุบันแบ่งออกเป็น 3 ประเภท ได้แก่

1. **เนทีฟแอปพลิเคชัน (Native application)** คือ แอปพลิเคชันที่ถูกพัฒนาโดยใช้ชุดคำสั่ง หรือเครื่องมือที่เตรียมไว้สำหรับพัฒนาบนระบบปฏิบัติการนั้น ๆ เช่น ระบบปฏิบัติการแอนดรอยด์ (Android) พัฒนาโดยใช้ภาษาจาวา และใช้เครื่องมือ Android Studio หรือ eclipse ในการพัฒนา หรือในระบบปฏิบัติการไอโอเอส (iOS) พัฒนาโดยใช้ภาษา Object-C และใช้เครื่องมือ XCode ในการพัฒนา เป็นต้น ซึ่งแอปพลิเคชันเหล่านี้หลังจากพัฒนาแล้วเสร็จจะไม่สามารถนำไปใช้กับระบบปฏิบัติการอื่น หรือไม่สามารถใช้งานข้ามระบบปฏิบัติการได้

2. **ไฮบริดจ์แอปพลิเคชัน (Hybrid application)** คือ แอปพลิเคชันที่ถูกพัฒนาขึ้นมาเพื่อให้สามารถใช้งานได้ในทุก ๆ ระบบปฏิบัติการของอุปกรณ์เคลื่อนที่ ซึ่งอาจใช้ชุดคำสั่งสำเร็จรูป หรือ เฟรมเวิร์ค (Framework) เข้ามาช่วยในการพัฒนา ซึ่งบางเฟรมเวิร์คอาจใช้ภาษาพื้นฐานที่เป็นที่นิยมในปัจจุบัน เช่น ภาษาเอชทีเอ็มแอล ภาษาซีเอสเอส ภาษาจาวาสคริปต์ เป็นต้น ทำให้ผู้พัฒนาไม่ต้องเรียนรู้ภาษาใหม่ และหลังจากพัฒนาเสร็จแล้วสามารถส่งออกแอปพลิเคชันให้สามารถใช้งานได้บนระบบปฏิบัติการที่หลากหลาย ทำให้พัฒนาแค่ครั้งเดียว ซึ่งในปัจจุบันมีหลายเฟรมเวิร์คให้เลือกใช้ เช่น JQuery Mobile, Xamarine, PhoneGap, Ionic Framework, Intel XDK, Framework7, Mobile Angular UI, Onsen UI, Kendo UI เป็นต้น



ภาพที่ 1.1 รูปแบบการทำงานของแอปพลิเคชันแต่ละชนิด

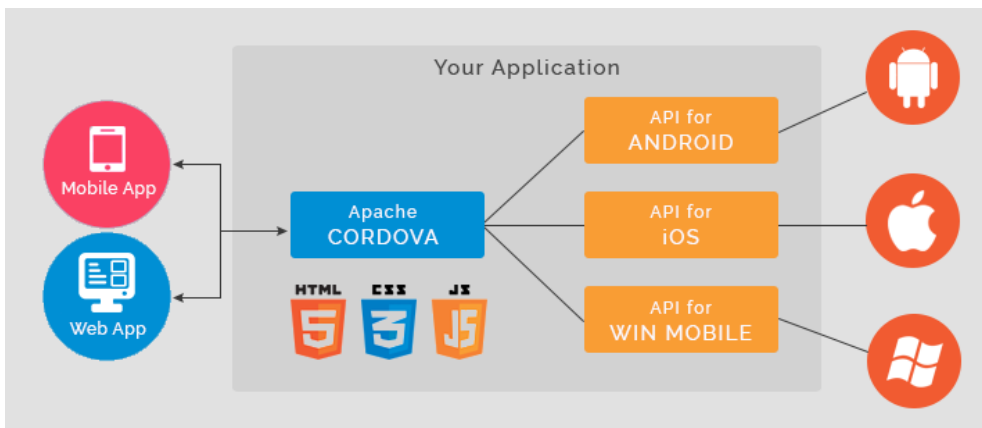
ที่มา : <https://developer.telerik.com/featured/what-is-a-webview/>

3. **เว็บแอปพลิเคชัน (Web application)** คือ แอปพลิเคชันที่ถูกพัฒนาขึ้นมาเพื่อให้แสดงผลระบบในรูปแบบของเว็บไซต์ และถูกกำหนดรูปแบบการแสดงผลข้อมูลให้เหมาะสมกับขนาดของอุปกรณ์เคลื่อนที่ ซึ่งสามารถเป็นเว็บเบราว์เซอร์บนอุปกรณ์แล้วเรียกใช้งานผ่านยูอาร์แอลได้ ทำให้ลดการประมวลผล และการใช้งานทรัพยากรในตัวเครื่องอุปกรณ์เคลื่อนที่ที่มีจำกัด

### รูปแบบการทำงานของไฮบริดจ์แอปพลิเคชัน

ไฮบริดจ์แอปพลิเคชันถูกออกแบบมาเพื่อให้รองรับการใช้งานกับทุก ๆ ระบบปฏิบัติการของอุปกรณ์เคลื่อนที่ในการพัฒนาแค่ครั้งเดียว ทำให้ไม่เสียเวลาในการพัฒนา ซึ่งในปัจจุบันแอปพลิเคชันประเภทนี้ยังสามารถเรียกใช้งานทรัพยากรของเครื่องสมาร์ตโฟนได้ เช่น พื้นที่จัดเก็บข้อมูลบนเครื่อง รายชื่อผู้ติดต่อ จีพีเอส หรือ กล้องถ่ายภาพ เป็นต้น

รูปแบบการทำงานของไฮบริดจ์แอปพลิเคชันเมื่อเปิดใช้งานจะเรียกใช้คอมโพเนนท์ที่ชื่อว่า “WebView” ผ่านเอพีไอ (API) ซึ่งเป็นเว็บเบราว์เซอร์ที่แสดงผลในขอบเขตของแอปพลิเคชัน ซึ่งจะแสดงผลรหัสคำสั่งไฟล์เอกสารเอชทีเอ็มแอล ภาษาซีเอสเอส และจาวาสคริปต์ ซึ่งหมายถึงการพัฒนาเว็บเพจให้อยู่ภายในแอปพลิเคชัน เพื่อให้คอมโพเนนท์ดังกล่าวดึงไปแสดงผล ซึ่งผู้พัฒนาสามารถใช้งาน Adobe PhoneGap หรือ Apache Cordova เพื่อส่งออกไฟล์ให้อยู่ในรูปแบบของแอปพลิเคชันของแต่ละระบบปฏิบัติการได้ เช่น แอนดรอยด์จะใช้ไฟล์ที่มีส่วนขยายเป็น .apk เป็นต้น ดังภาพที่ 1.2



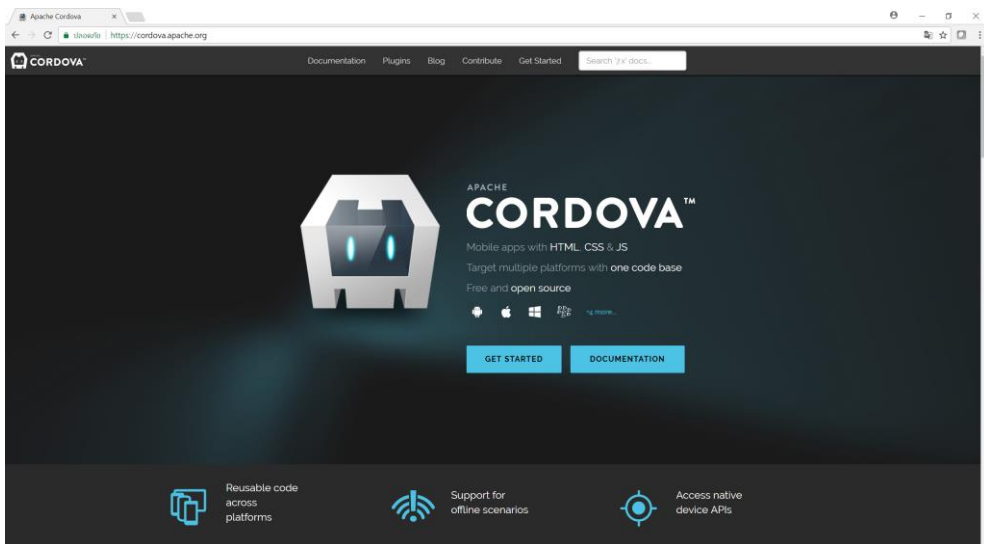
ภาพที่ 1.2 การทำงานของ Cordova

ที่มา : <https://www.samirkamble.com/installing-and-setting-up-apache-cordova-phonegap-environment/>

## รู้จักกับ Adobe PhoneGap และ Apache Cordova

PhoneGap และ Cordova มีรูปแบบการทำงานที่เหมือน ๆ กัน ซึ่งเดิม PhoneGap เป็นโครงการของบริษัท Nitobi ซึ่งเป็นซอฟต์แวร์แบบเปิดรหัส (Open source) และถูกซื้อไปโดยบริษัท Adobe เพื่อรวมเข้ากับโปรแกรม Adobe Dreamweaver ส่วน Cordova เป็นโครงการที่ถูกแยกออกมาและพัฒนาเป็นซอฟต์แวร์แบบเปิดรหัสตามเดิม ทำให้ทั้งสองเป็นเหมือนดังซอฟต์แวร์เดียวกัน ซึ่ง Cordova ถูกรวมเข้ากับซอฟต์แวร์หลายตัว เช่น Visual Studio Code หรือ Jet Brain เป็นต้น และยังถูกรวมไว้กับเฟรมเวิร์คสำหรับการพัฒนาโมบายแอปพลิเคชัน เช่น Ionic Framework เป็นต้น

Cordova จะทำงานคล้ายกับเว็บเบราว์เซอร์ซึ่งจะไปดึงส่วนของ WebView จากเอพีไอของแต่ละระบบปฏิบัติการมาใช้งานในรูปแบบของเนทีฟแอปพลิเคชัน หลังจากนั้นจะทำการเรียกไฟล์เอกสารเอชทีเอ็มแอลที่พัฒนาเขียนไว้มาแสดงผล ซึ่งรวมไปถึงไฟล์เอกสารซีเอสเอส และจาวาสคริปต์ และจะบีบอัดรวมไฟล์ทั้งหมดและส่งออกไฟล์ให้อยู่ในรูปแบบของแอปพลิเคชันตามแต่ละระบบปฏิบัติการ และฝังรหัสคำสั่งเอชทีเอ็มแอลไว้ข้างในแอปพลิเคชันนั้น ๆ



ภาพที่ 1.3 หน้าหลักเว็บไซต์ของ Cordova

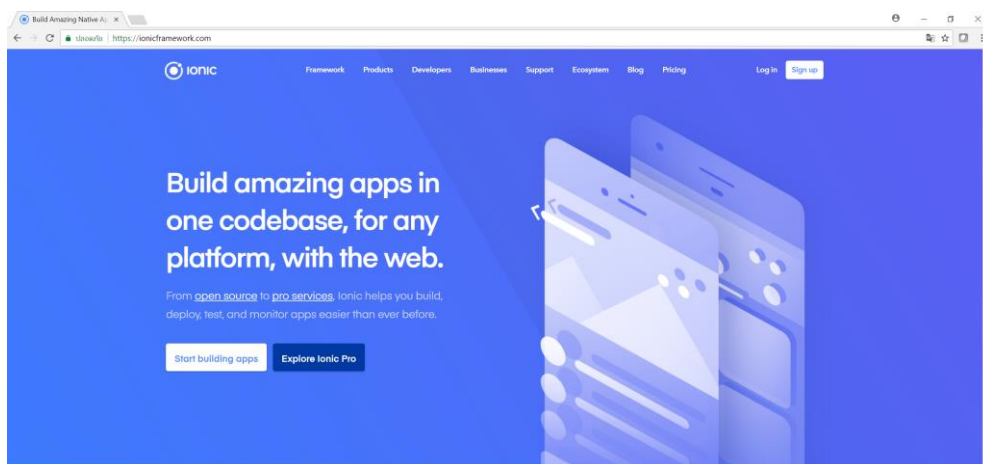
ด้วยคุณสมบัติของ Cordova รองรับระบบปฏิบัติการที่หลากหลาย เช่น Android, iOS, Amazon Fire OS, Bada, Blackberry, Firefox OS, Ubuntu, WebOS, Windows Phone เป็นต้น ศึกษาข้อมูลคุณสมบัติและรายละเอียดของ Cordova เพิ่มเติมได้จาก <https://cordova.apache.org/>



## รู้จักกับ Ionic Framework

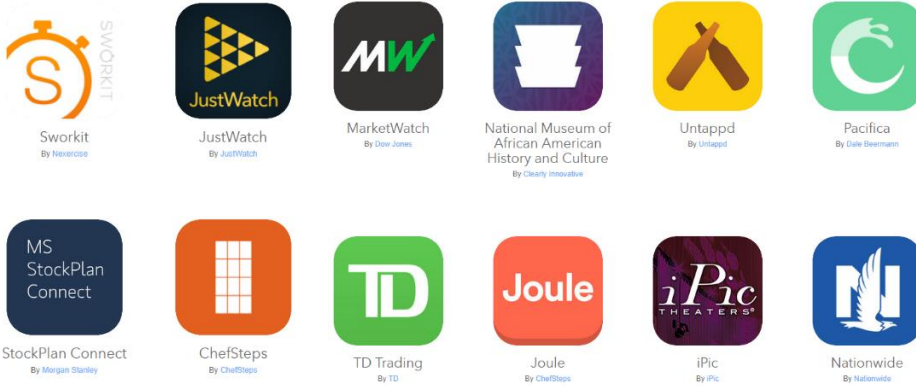
Ionic Framework เป็นเครื่องมือสำหรับการสร้างแอปพลิเคชันสำหรับอุปกรณ์เคลื่อนที่ หรือที่เรียกกันว่า โมบายแอปพลิเคชัน (Mobile application) ซึ่งสามารถพัฒนาได้โดยใช้ภาษาพื้นฐาน เช่น ภาษาเอชทีเอ็มแอล ภาษาซีเอสเอส และภาษาจาวาสคริปต์ในการพัฒนา และยังสามารถส่งออก รหัสคำสั่งที่พัฒนาเสร็จแล้วให้อยู่ในรูปแบบของแอปพลิเคชันที่รองรับระบบปฏิบัติการที่หลากหลาย ทั้งระบบปฏิบัติการแอนดรอยด์ ไอโอเอส หรือ วินโดวส์โมบาย

Ionic Framework มีการบริหารจัดการบริการต่าง ๆ โดยใช้รหัสคำสั่งซีแอลไอ (Command Line Interface: CLI) ในปัจจุบันมีการพัฒนาถึงเวอร์ชันที่ 3 ซึ่งมีการปรับปรุงให้เขียนรหัสคำสั่งที่สั้นลง มีความกระชับมากยิ่งขึ้น ทำงานได้ดีขึ้น มีส่วนประสานงานกับผู้ใช้ที่สวยงาม (Graphic User Interface: GUI) และยังคงเลือกใช้งานเทคโนโลยีสมัยใหม่ที่ได้รับการยอมรับและนิยมในปัจจุบัน เช่น Angular และ Sass เป็นต้น เว็บไซต์หลักของ Ionic Framework ดังภาพที่ 1.4



ภาพที่ 1.4 เว็บไซต์หลักของ Ionic Framework

ตัวอย่างแอปพลิเคชันที่พัฒนาโดยใช้ Ionic Framework ดังภาพที่ 1.5 ซึ่งได้รับความนิยมจากบริษัทชั้นนำที่หลากหลาย เช่น แอปพลิเคชันด้านสุขภาพของอเมริกาอย่าง “UNIQLO” เป็นต้น ซึ่งสามารถติดตามข้อมูลได้จากยูอาร์แอล <http://showcase.ionicframework.com/apps/top>



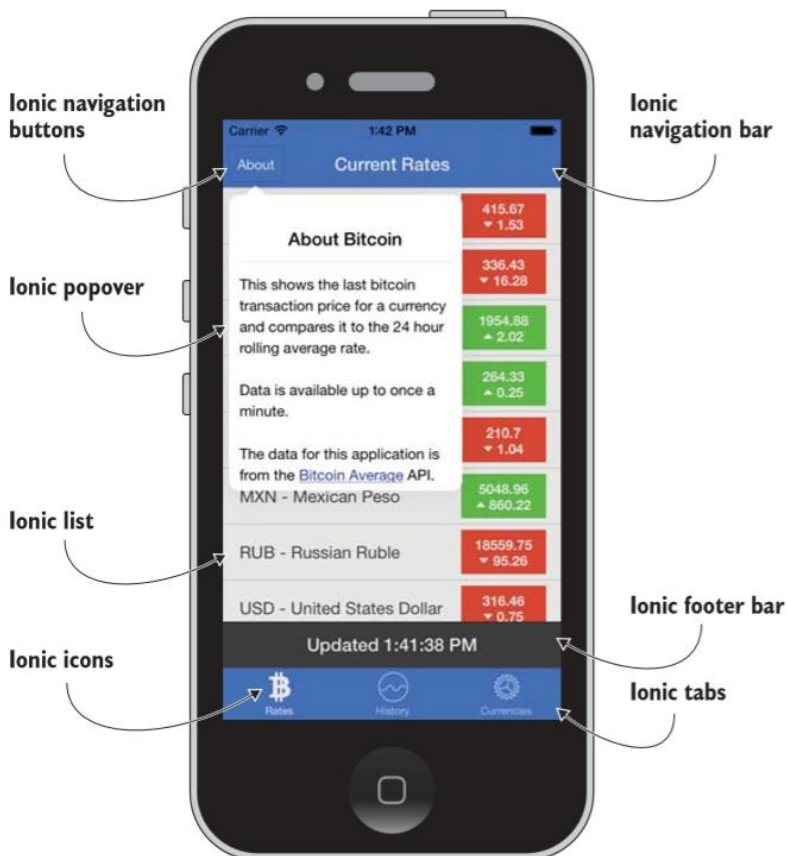
ภาพที่ 1.5 ตัวอย่างแอปพลิเคชันที่พัฒนาโดย Ionic Framework

### ข้อดีของ Ionic Framework

ข้อดีของการพัฒนาแอปพลิเคชันโดยใช้ Ionic Framework มีดังนี้

1. เป็นซอฟต์แวร์แบบเปิดรหัส (Open Source) ซึ่ง Ionic เปิดให้ใช้งานได้ฟรี ซึ่งเป็นเฟรมเวิร์คที่เป็นที่นิยมและมีการพัฒนาคุณสมบัติของเฟรมเวิร์คอย่างต่อเนื่อง
2. มีชุมชน (Community) ที่เข้มแข็งสำหรับตอบคำถามปัญหาที่เกี่ยวข้องกับการใช้งานเฟรมเวิร์คและแก้ไขปัญหาให้โดยทีมนักพัฒนาจากทั่วโลก
3. แอปพลิเคชันเดียวสามารถรองรับได้ทุก ๆ ระบบปฏิบัติการ (One app many platforms) นักพัฒนาสามารถส่งออกแอปพลิเคชันที่พัฒนาให้รองรับการทำงานในทุก ๆ ระบบปฏิบัติการซึ่งสามารถอัปเดตขึ้นบนสโตร์ของระบบปฏิบัติการนั้นได้อีกด้วย ทำให้ลดระยะเวลาในการพัฒนา รวมไปถึงลดค่าใช้จ่ายที่เกิดขึ้นในระหว่างการพัฒนา
4. รองรับการสร้างคอมโพเนนต์ (Component creation) นักพัฒนาสามารถกำหนดรูปแบบการแสดงผล หรือออกแบบหน้าจอส่วนประสานงานกับผู้ใช้ได้อย่างอิสระ เช่น การออกแบบไอคอน ภาพหน้าจอต้อนรับ กำหนดความกว้าง หรือ ความสูง เป็นต้น

5. รองรับการแสดงผลในหลากหลายเบราว์เซอร์ (Comprehensive browser support) ซึ่งเฟรมเวิร์กมีการเรียกใช้งาน “WebView” เพื่อนำไฟล์เอกสารเอชทีเอ็มแอลมาแสดงผลในขอบเขตของแอปพลิเคชัน ซึ่งเปรียบเสมือนเบราว์เซอร์สำหรับเรียกดูเว็บไซต์ ซึ่งในแต่ละระบบปฏิบัติการจะมีรูปแบบการแสดงผลที่แตกต่างกัน แต่ด้วยคุณสมบัติของ Cordova ที่ได้รับการปรับปรุงคุณสมบัติให้มีการเรียกใช้งานปลั๊กอิน Crosswalk เพื่อให้รองรับการแสดงผลเอชทีเอ็มแอลเวอร์ชัน 5 ซีเอสเอสเวอร์ชันใหม่ รวมไปถึงความสามารถอื่น ๆ ของเว็บ และยังเพิ่มความเร็วในการแปลความหมายของคำสั่งเอชทีเอ็มแอล ทำให้รองรับการแสดงผลบนแอปพลิเคชันที่สวยงาม รวดเร็ว ไม่ผิดเพี้ยน ทำให้รองรับการแสดงผลในทุก ๆ เบราว์เซอร์



ภาพที่ 1.6 ส่วนประสานงานกับผู้ใช้ของ Ionic Framework

ที่มา : (Jeremy Wilken, 2016)

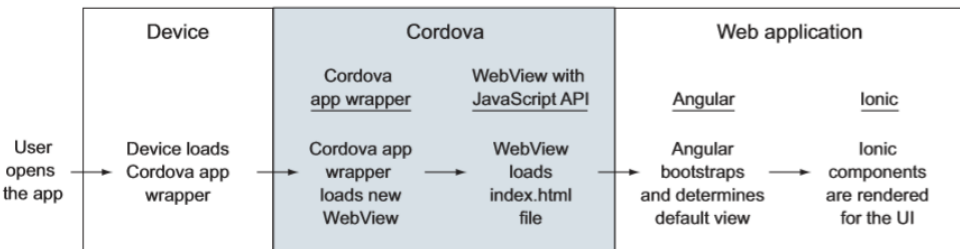
6. มีส่วนประสานงานกับผู้ใช้ที่สวยงาม (UI Design) มีส่วนประสานงานให้เลือกใช้งานที่ครบถ้วน หลากหลาย และใช้งานได้ง่าย ซึ่งเรียกใช้งานโดยใช้คำสั่งเอชทีเอ็มแอล ตกแต่งโดยใช้ภาษาซีเอสเอส และกำหนดรูปแบบการทำงานโดยใช้จาวาสคริปต์ ซึ่งเป็นที่คุ้นเคยในหมู่นักพัฒนาเว็บไซต์อยู่แล้ว เช่น การเรียกใช้งานปุ่ม ฟอรั่ม สไลด์เมนู แท็บ หรือ กล่องข้อความ เป็นต้น และยังปรับเปลี่ยนรูปแบบการแสดงผลให้ในแต่ละระบบปฏิบัติการแบบอัตโนมัติ ตัวอย่างดังภาพที่ 1.6

7. ใช้เทคโนโลยีสำหรับพัฒนาที่ได้รับการยอมรับ (Web Technology) ซึ่งนอกจากใช้ภาษาพื้นฐานในการพัฒนาแล้ว ยังมีกรเลือกใช้เฟรมเวิร์คสำหรับการพัฒนาที่เป็นที่นิยมอย่างกว้างขวาง เช่น Angular และ Sass เป็นต้น ทำให้รองรับการใช้งานแอปพลิเคชันที่หลากหลายรูปแบบ

8. เครื่องมือใช้งานง่าย (Simple to use) เช่น การลากวาง (Drag and drop) รองรับการแจ้งเตือน (Push notification) มีไลบรารีสำหรับเรียกใช้ภาพไอคอนที่สวยงามและหลากหลาย และเครื่องมือสำหรับการจัดการบริการต่าง ๆ ของเฟรมเวิร์คผ่านคำสั่งแบบซีแอลไอ (CLI)

### การทำงานของ Ionic Framework

Ionic Framework ได้ออกแบบเครื่องมือเพื่อให้สามารถสร้างแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ได้อย่างรวดเร็ว ง่าย และสวยงาม ซึ่ง Ionic Framework ได้เพิ่ม Angular ซึ่งเป็นเฟรมเวิร์คสำหรับการพัฒนาแอปพลิเคชันบนเว็บ (Web application framework) และการส่งออกแอปพลิเคชันจะใช้ Cordova เพื่อจัดองค์ประกอบให้อยู่ในรูปแบบของเนทีฟแอปพลิเคชัน รูปแบบการทำงานของ Ionic Framework ดังภาพที่ 1.7 ซึ่งประกอบไปด้วย



ภาพที่ 1.7 การทำงานของ Ionic Framework

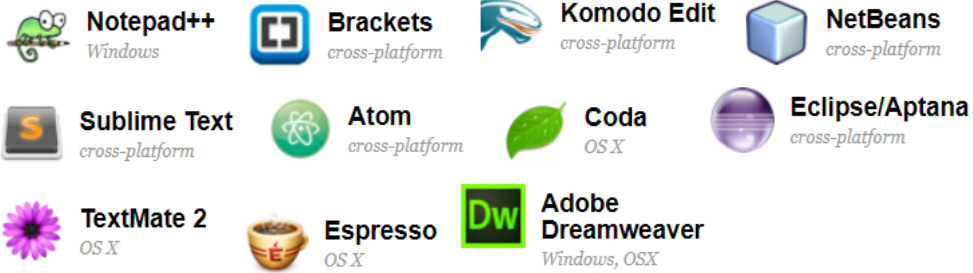
ที่มา : (Jeremy Wilken, 2016)



1. อุปกรณ์ (Device) คือ อุปกรณ์สำหรับเรียกใช้งานแอปพลิเคชัน ซึ่งมาจากการติดตั้งลงบนระบบปฏิบัติการของอุปกรณ์นั้น ๆ ซึ่งแอปพลิเคชันจะถูกดาวน์โหลดและติดตั้งจากสโตร์ของแต่ละระบบปฏิบัติการ ซึ่งจะกำหนดให้เอพีไอ (API) สำหรับแอปพลิเคชันสามารถเข้าถึงและใช้งานทรัพยากรภายในอุปกรณ์ เช่น ตำแหน่งพิกัดจีพีเอส กล้องถ่ายภาพ พื้นที่สำหรับจัดเก็บข้อมูล เป็นต้น
2. Cordova app wrapper เป็นส่วนที่โหลดเนทีฟแอปพลิเคชันขึ้นมาเพื่อเรียกใช้งาน WebView ซึ่ง Cordova จะไปเรียกไฟล์เอกสารเอชทีเอ็มแอลขึ้นมาเพื่อประมวลผลคำสั่งและแสดงผลในส่วน of WebView นั้น ซึ่งรวมไปถึงไฟล์เอกสารซีเอสเอส และจาวาสคริปต์ด้วย ซึ่งการทำงานแบบนี้เรียกว่า ไฮบริดแอปพลิเคชัน ที่สามารถทำงานแบบข้ามระบบปฏิบัติการได้
3. Cordova JavaScript API เป็นเสมือนสะพานที่เชื่อมต่อระหว่างแอปพลิเคชันและอุปกรณ์ที่ใช้แสดงผล ซึ่ง JavaScript API จะทำงานอยู่ส่วนด้านหลังระหว่างการประมวลผล ซึ่งเป็นเครื่องมือที่ทำให้เว็บแอปพลิเคชันที่อยู่ภายในสามารถทำงานร่วมกันกับเนทีฟแอปพลิเคชันของแต่ละระบบปฏิบัติการ
4. Angular เป็นเฟรมเวิร์คสำหรับพัฒนาเว็บแอปพลิเคชันที่เป็นที่นิยมอย่างแพร่หลาย ซึ่งจะช่วยให้เว็บแอปพลิเคชันที่แสดงผลอยู่ในส่วนของ WebView สามารถทำงานได้อย่างต่อเนื่อง ซึ่งจะใช้ในการบริหารจัดการข้อมูลภายในเพื่อนำมาแสดงผลที่หน้าจอแอปพลิเคชัน
5. Ionic เป็นส่วนที่สร้างหน้าจอสำหรับประสานงานกับผู้ใช้ (Graphic User Interface: GUI) ซึ่งจะถูกพัฒนาร่วมกันกับ Angular เพื่อใช้ในการออกแบบหน้าจอ ซึ่งจะรวมถึง การสร้างแท็บ ปุ่ม ระบบนำทางของแอปพลิเคชัน เมนูต่าง ๆ ซึ่ง Ionic จะรวบรวมเครื่องมือเพื่อให้ผู้พัฒนาได้เลือกใช้งานได้อย่างสะดวก ง่าย และสวยงาม เพื่อบริหารจัดการข้อมูลภายในแอปพลิเคชันได้อย่างมีประสิทธิภาพ

## โปรแกรม Text Editor สำหรับพัฒนาแอปพลิเคชัน

การเขียนคำสั่งหรือแก้ไขรหัสคำสั่งสำหรับการพัฒนาแอปพลิเคชันจะต้องใช้โปรแกรมสำหรับแก้ไขจำพวก Text Editor ซึ่งแล้วแต่ความถนัดของผู้พัฒนาเอง ในปัจจุบันมีโปรแกรมให้เลือกใช้งานหลากหลาย ผู้พัฒนาสามารถเลือกดาวน์โหลดและติดตั้งได้จากเว็บไซต์โดยทั่วไป ตัวอย่างโปรแกรมประเภท Text Editor ดังภาพที่ 1.8



ภาพที่ 1.8 ตัวอย่างโปรแกรมประเภท Text Editor

ในหนังสือเล่มนี้ผู้เขียนเลือกใช้โปรแกรม Visual Studio Code สำหรับเป็นโปรแกรมในการแก้ไขรหัสคำสั่ง ซึ่งข้อดีของโปรแกรมนี้นี้คือเป็นโปรแกรมที่ให้ใช้งานได้ฟรี มีเครื่องมือสำหรับนักพัฒนาที่หลากหลาย และรองรับการทำงานร่วมกันกับแถบคำสั่งประเภทคอมมานด์ไลน์ ซึ่งสามารถเข้าไปศึกษารายละเอียดคุณสมบัติของโปรแกรมและดาวน์โหลดได้จาก <https://code.visualstudio.com/>

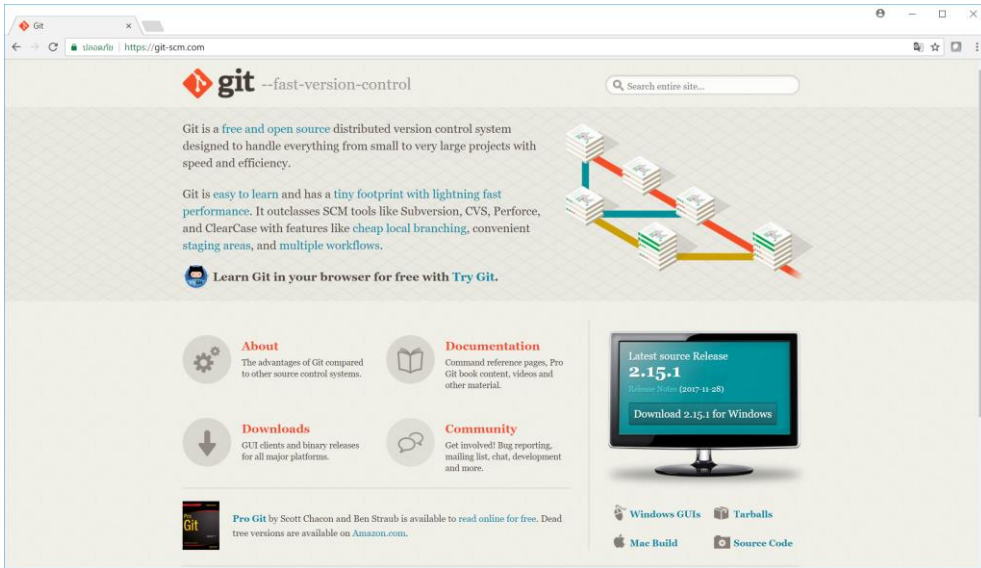
## การติดตั้ง Git

Git เป็นตัวสำรองรหัสคำสั่ง (Backup source code) ของผู้พัฒนา หรือเรียกอีกอย่างหนึ่งว่า “Version Control” ซึ่งเป็นตัวจัดการเกี่ยวกับเวอร์ชันของโปรเจกต์งานของนักพัฒนา ซึ่งทำให้สามารถย้อนกลับไปดูได้ว่าพัฒนาเมื่อใด แก้ไขเมื่อใด หรือไฟล์นั้นถูกเขียนโดยใครบ้าง เป็นต้น ซึ่งเหมาะสำหรับการทำงานร่วมกันหลายคน

หลังจากติดตั้ง Git ผู้พัฒนาสามารถเรียกใช้คำสั่ง Bower ซึ่งเป็นตัวจัดการแพ็คเกจ (Package Manager) และใช้ Git สำหรับดาวน์โหลดไลบรารีต่าง ๆ รวมไปถึงการดาวน์โหลดเทมเพลตที่นำมาใช้ในการพัฒนาแอปพลิเคชันผ่านคำสั่งคอมมานด์ไลน์ได้อีกด้วย ขั้นตอนการติดตั้งดังนี้

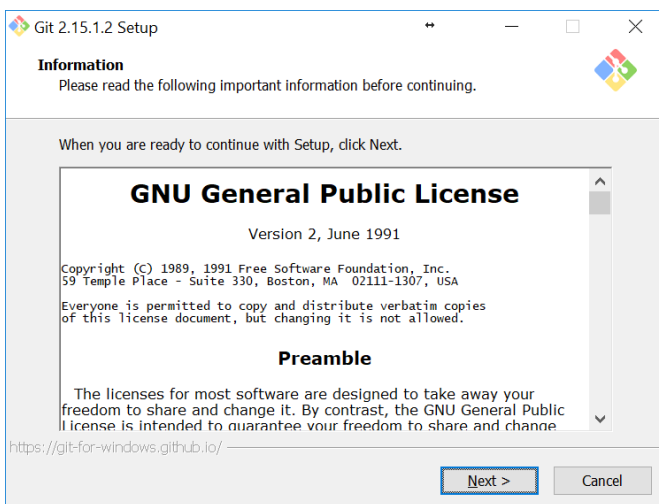
1. เข้าไปที่เว็บไซต์ <https://git-scm.com/> แล้วเลือกดาวน์โหลดเวอร์ชันที่ต้องการเพื่อให้รองรับการแสดงผลบนระบบปฏิบัติการของผู้พัฒนาเอง ซึ่งในขั้นตอนนี้เลือก Download for Windows ดังภาพที่ 1.9





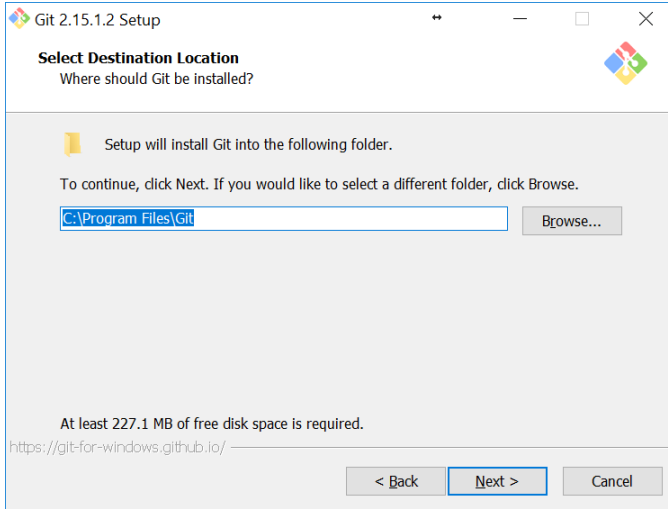
ภาพที่ 1.9 ดาวน์โหลดตัวติดตั้ง Git

2. หลังจากดาวน์โหลดแล้วจะได้ตัวติดตั้งซึ่งอยู่ด้านล่างของเว็บเบราว์เซอร์ หรือ ดูที่ โฟลเดอร์ Downloads ของคอมพิวเตอร์ แล้วทำการคลิกไฟล์ที่ดาวน์โหลดมาเพื่อทำการติดตั้ง Git ตัวอย่างดังภาพที่ 1.10 ซึ่งจะปรากฏข้อตกลงการใช้งาน จากนั้นคลิกปุ่ม Next เพื่อไปยังขั้นตอนต่อไป



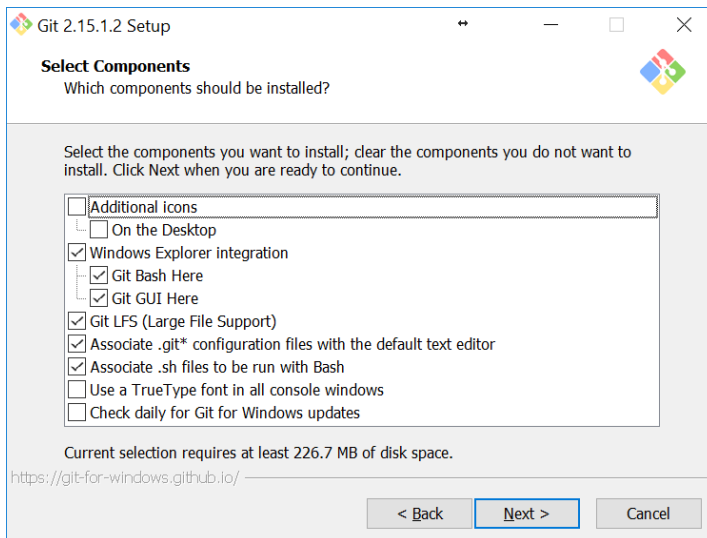
ภาพที่ 1.10 ข้อตกลงการใช้งาน Git

3. เลือกตำแหน่งที่ใช้สำหรับจัดเก็บไฟล์ติดตั้ง จากนั้นคลิกปุ่ม Next เพื่อไปยังขั้นตอนถัดไป ดังภาพที่ 1.11



ภาพที่ 1.11 เลือกตำแหน่งการติดตั้ง

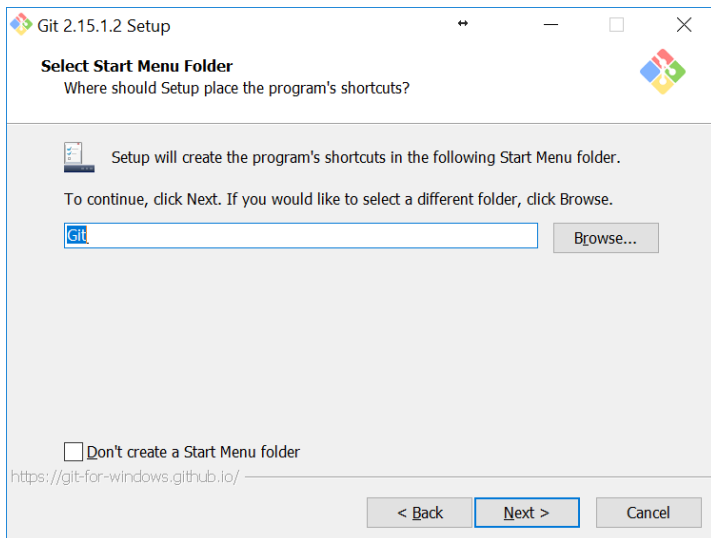
4. เลือกคอมโพเนนท์ที่ต้องการติดตั้งใช้งาน ซึ่งหากไม่ต้องการตัวเลือกอื่น ๆ แล้วให้ใช้ค่าปกติที่โปรแกรมเลือกไว้ให้เป็นค่าเริ่มต้น จากนั้นคลิกปุ่ม Next ดังภาพที่ 1.12



ภาพที่ 1.12 เลือกคอมโพเนนท์ที่ต้องการติดตั้งใช้งาน

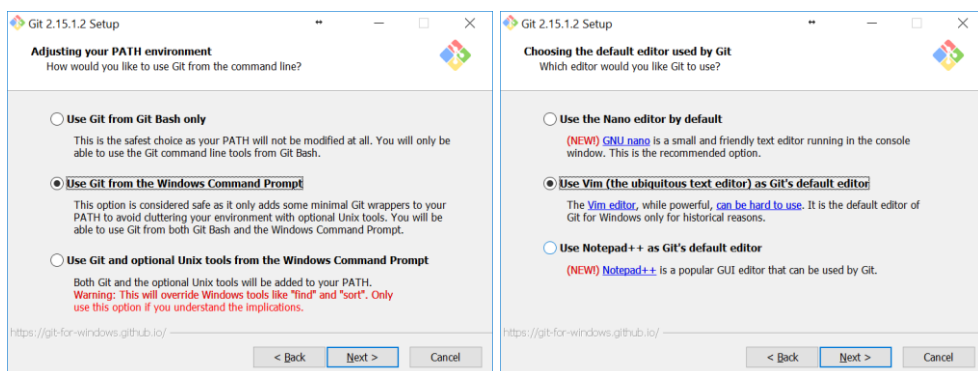


5. กำหนดชื่อโฟลเดอร์ที่ใช้จัดเก็บไอคอนที่เมนูเริ่มต้นของระบบปฏิบัติการวินโดวส์ จากนั้นคลิกปุ่ม Next เพื่อไปยังขั้นตอนถัดไป ดังภาพที่ 1.13



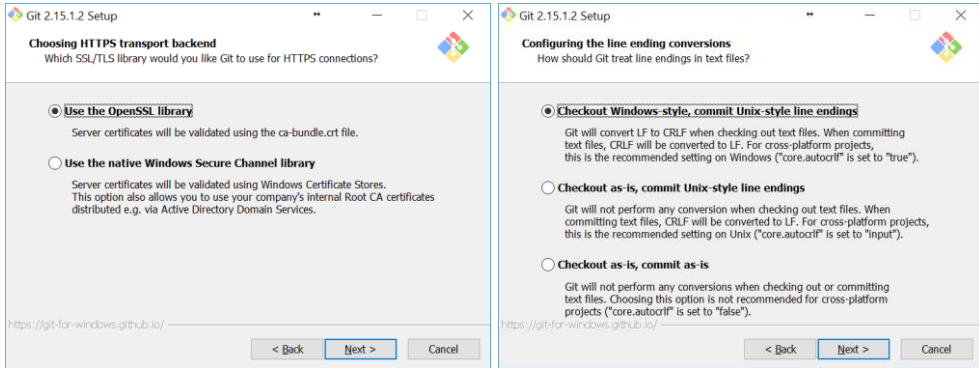
ภาพที่ 1.13 กำหนดชื่อโฟลเดอร์สำหรับจัดเก็บไอคอน

6. เลือกโปรแกรมสำหรับแก้ไขที่ใช้สำหรับ Git ซึ่งเป็นค่าเริ่มต้นจากนั้นคลิกปุ่ม Next และที่หน้าต่าง Adjust your PATH environment ให้คลิกปุ่ม Next ดังภาพที่ 1.14



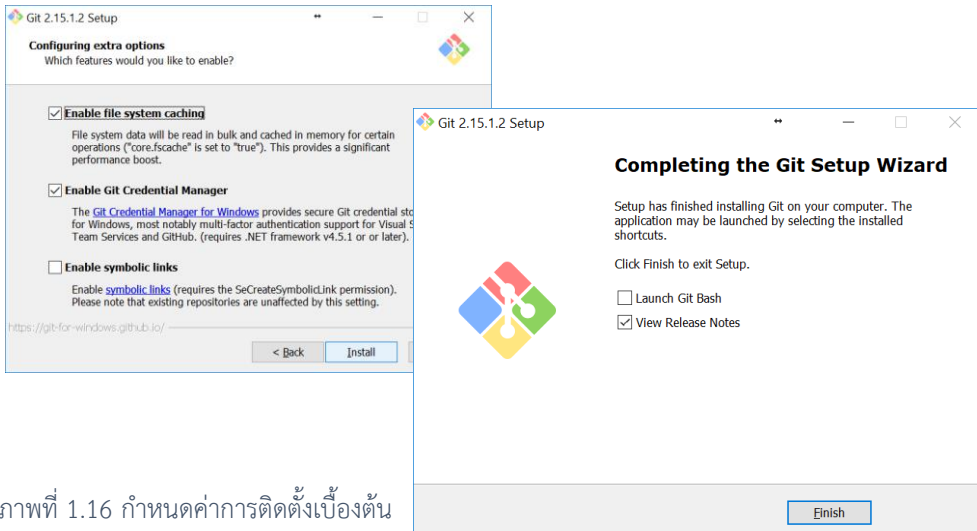
ภาพที่ 1.14 กำหนดคุณสมบัติเบื้องต้น

7. ที่หน้าต่าง Choosing HTTPS transport backend สำหรับการรับส่งข้อมูล ให้ใช้ค่าเริ่มต้น จากนั้นคลิกปุ่ม Next จะปรากฏหน้าต่าง Configuring the line ending conversions ให้คลิกที่ปุ่ม Next ดังภาพที่ 1.15



ภาพที่ 1.15 กำหนดคุณสมบัติการรับส่งข้อมูลเบื้องต้น

8. จากนั้นจะปรากฏหน้าต่าง Configuring extra options ให้ทำการคลิกที่ปุ่ม Install เพื่อทำการติดตั้ง Git ดังภาพที่ 1.16



ภาพที่ 1.16 กำหนดค่าการติดตั้งเบื้องต้น

9. ซึ่งเมื่อติดตั้งเสร็จแล้วจะปรากฏหน้าต่าง Completing the Git Setup Wizard ขึ้นมา ให้คลิกที่ปุ่ม Finish เพื่อสิ้นสุดการติดตั้ง

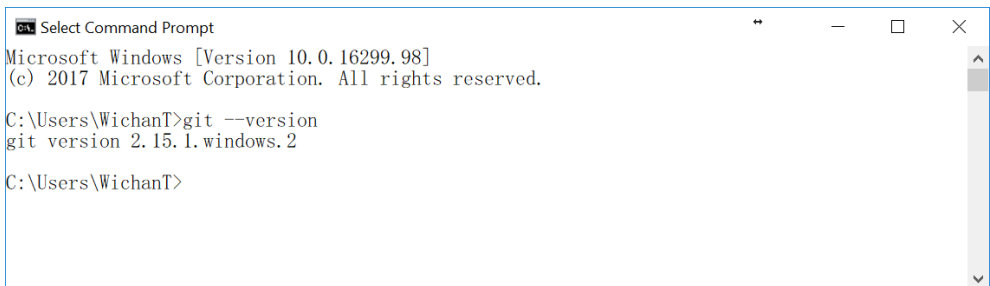


## การตรวจสอบเวอร์ชันของ Git

หลังจากติดตั้ง Git สำหรับการบริหารจัดการเวอร์ชันของแอปพลิเคชันที่พัฒนาแล้ว หากต้องการทราบถึงเวอร์ชันของ Git ที่ติดตั้ง ให้ผู้พัฒนาพิมพ์คำสั่งผ่านคอมมานด์ไลน์ โดยเปิดโปรแกรม Command Prompt ขึ้นมา จากนั้นพิมพ์คำสั่ง

```
git --version
```

ซึ่งระบบจะแจ้งเวอร์ชันปัจจุบันของ git ดังภาพที่ 1.17



```

Select Command Prompt
Microsoft Windows [Version 10.0.16299.98]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\WichanT>git --version
git version 2.15.1.windows.2

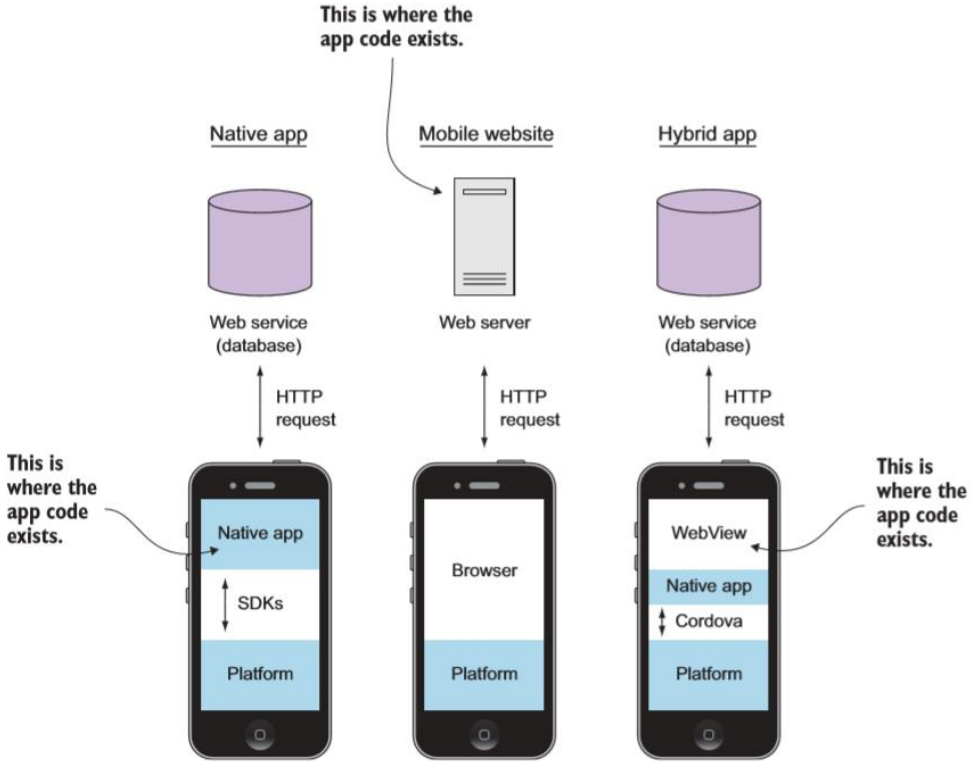
C:\Users\WichanT>
  
```

ภาพที่ 1.17 การตรวจสอบเวอร์ชันปัจจุบันของ Git

## สรุป

ในบทนี้ได้กล่าวถึงพื้นฐานเกี่ยวกับแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ ซึ่งเป็นที่นิยมอย่างแพร่หลายในปัจจุบัน ซึ่งแอปพลิเคชันแบ่งออกเป็น 3 ชนิด คือ 1) เนทีฟแอปพลิเคชัน ซึ่งเป็นการพัฒนาแอปพลิเคชันสำหรับพัฒนาให้รองรับเฉพาะระบบปฏิบัติการนั้น ๆ 2) ไฮบริดจ์แอปพลิเคชัน ที่เป็นการนำเอาพื้นฐานภาษาเซซีเอ็มแอล ภาษาซีเอสเอส และจาวาสคริปต์ พัฒนาเป็นแอปพลิเคชัน และแสดงผลผ่าน Webview เพื่อให้สามารถส่งออกแอปพลิเคชันให้รองรับการแสดงผลในทุก ๆ ระบบปฏิบัติการ และ 3) เว็บแอปพลิเคชัน ซึ่งนำเสนอข้อมูลผ่านเว็บไซต์ โดยเชื่อมต่อผ่านทางเว็บเบราว์เซอร์ของอุปกรณ์นั้น ๆ ซึ่งแต่ละชนิดก็มีข้อดีข้อเสียที่แตกต่างกันออกไป ผู้พัฒนาควรเลือกรูปแบบการพัฒนาที่เหมาะสมกับงานที่ต้องการ

ในส่วนของเครื่องมือที่ใช้ในการบีบอัดไฟล์โปรเจกต์ที่พัฒนาแล้วนั้นที่นิยมในปัจจุบันสามารถเลือกใช้ PhoneGap หรือ Cordova ได้ ซึ่งทั้งสองตัวนี้ถือว่ามีรูปแบบการทำงานที่คล้ายคลึงกัน ซึ่ง PhoneGap ถูกซื้อไปโดย Adobe ส่วน Cordova ถูกแยกออกมาพัฒนาให้อยู่ในรูปแบบของซอฟต์แวร์แบบเปิดรหัส ทำให้สามารถใช้งานได้ฟรี ไม่มีค่าใช้จ่าย และพัฒนาต่อเนื่องได้อย่างรวดเร็ว รูปแบบการทำงานของแอปพลิเคชันแต่ละชนิด ดังภาพที่ 1.18



ภาพที่ 1.18 รูปแบบการทำงานของแอปพลิเคชันแต่ละชนิด  
ที่มา : (Jeremy Wilken, 2016)

นอกจากนั้นแล้วยังกล่าวถึงพื้นฐานของ Ionic Framework ซึ่งผู้อ่านคงพอเข้าใจแนวทางการทำงานของเฟรมเวิร์คสำหรับการพัฒนาแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ ซึ่ง Ionic Framework ได้จัดเตรียมเครื่องมือสำหรับออกแบบและพัฒนาแอปพลิเคชันไว้หลากหลาย เพื่อให้ผู้พัฒนาเรียกใช้งาน เช่น ปุ่ม ระบบนำทาง ลิสต์รายการ ไอคอน แท็บเมนู และอื่น ๆ

ในบทถัดไปจะกล่าวถึงการเตรียมเครื่องมือสำหรับการพัฒนาแอปพลิเคชัน ซึ่งเป็นเครื่องมือหลักในการเขียนคำสั่งและประมวลผลคำสั่งเพื่อให้แสดงผลได้ตามความต้องการของผู้พัฒนา