

Git in Action

คู่มือปฏิบัติสำหรับนักพัฒนาและ
ทีมนักพัฒนาซอฟต์แวร์

อนุชิต ชโลธร

สำนักพิมพ์ก๊อปวาง



Git in Action

คู่มือปฏิบัติสำหรับนักพัฒนาและทีมนัก
พัฒนาซอฟต์แวร์

อนุชิต ชโลธร

คำนำสำนักพิมพ์

ในยุคที่ซอฟต์แวร์มีบทบาทสำคัญในแทบทุกธุรกิจ การสร้างผลิตภัณฑ์ดิจิทัลให้มีคุณภาพ ไม่ใช่เพียงเรื่องของการเขียนโปรแกรมเท่านั้น แต่ยังรวมถึงการจัดการกระบวนการพัฒนาอย่างมีระบบ เครื่องมือควบคุมเวอร์ชันจึงกลายเป็นองค์ประกอบที่นักพัฒนาไม่อาจละเลยได้ — และในจำนวนเครื่องมือที่มีอยู่ Git คือมาตรฐานที่โดดเด่นที่สุด

สำนักพิมพ์ของเราเล็งเห็นถึงความสำคัญของทักษะนี้ และเชื่อว่าการเข้าใจ Git อย่างถูกต้องตั้งแต่พื้นฐาน จะช่วยให้ทั้งนักศึกษา นักพัฒนา และผู้สนใจทั่วไปสามารถก้าวข้ามอุปสรรคในการทำงานจริง และสร้างสรรค์ซอฟต์แวร์ได้อย่างมีประสิทธิภาพมากขึ้น

หนังสือเล่มนี้ถูกเขียนขึ้นด้วยความตั้งใจที่จะทำให้ Git เป็นเรื่องเข้าใจง่าย ทุกคำอธิบายถูกกลั่นจากประสบการณ์จริงของผู้เขียนที่ทำงานกับ Git ในหลากหลายสภาพแวดล้อม ทั้งโปรเจกต์ส่วนตัว ทีมงานขนาดเล็ก ไปจนถึงระบบขนาดใหญ่ระดับองค์กร ผู้อ่านจะพบว่าทั้งเนื้อหา ตัวอย่าง และลำดับการนำเสนอเป็นไปอย่างเป็นระบบ ชัดเจน และเป็นมิตรต่อผู้เรียน

เราหวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะเป็นคู่มือที่ช่วยพัฒนาศักยภาพด้านเทคนิคของผู้อ่าน และเป็นบันไดสำคัญสู่การทำงานอย่างมืออาชีพในโลกของการพัฒนาซอฟต์แวร์

สำนักพิมพ์ขอขอบคุณผู้เขียนที่ทุ่มเทแรงกายแรงใจในการสร้างสรรค์
หนังสือเล่มนี้ และหวังว่าผู้อ่านจะได้รับทั้งความรู้และแรงบันดาลใจ
จากทุกหน้าในหนังสือเล่มนี้

คำนำนักเขียน

ผมเขียนหนังสือเล่มนี้เกิดขึ้นจากความเชื่ออย่างหนึ่ง — **“การเขียนโค้ดให้ดี ต้องเริ่มจากการจัดการความเปลี่ยนแปลงให้เป็นระบบ”**

ตลอดเส้นทางอาชีพในฐานะนักพัฒนา ผมพบว่าปัญหาที่ทำให้ทีมทำงานสะดุดไม่ใช่ความสามารถในการเขียนโค้ดเท่านั้น แต่คือการจัดการเวอร์ชัน การทำงานร่วมกัน และวิธีคิดที่ถูกต้องเกี่ยวกับประวัติของโปรเจกต์ สิ่งเหล่านี้กลายเป็นอุปสรรคใหญ่สำหรับหลายทีม แม้พวกเขาจะมีนักพัฒนาฝีมือดีมากมายก็ตาม

Git จึงเป็นทักษะสำคัญที่ผมหวังให้ทุกคน “เข้าใจ” ไม่ใช่แค่ “รู้คำสั่ง” เพราะเมื่อเข้าใจแนวคิดเบื้องหลัง Git จริง ๆ แล้ว การใช้งานจะกลายเป็นเรื่องง่าย ชัดเจน และทรงพลังอย่างยิ่ง

หนังสือเล่มนี้เกิดขึ้นจากประสบการณ์ที่ผมสะสมจากการทำงานจริง การสอน การให้คำปรึกษา และการแก้ปัญหภายในหลายโปรเจกต์ ผมตั้งใจนำเสนอ Git ในแบบที่อ่านง่าย สนุก และเป็นมิตรต่อผู้อ่านทุกระดับ โดยไม่ลดทอนความลึกซึ่งที่จำเป็นในงานระดับมืออาชีพ

ผมหวังว่าหนังสือเล่มนี้จะช่วยให้คุณใช้ Git ได้อย่างมั่นใจขึ้น เข้าใจ workflow ที่มีประสิทธิภาพ และสามารถนำไปประยุกต์ในงานของคุณ

ได้ทันที ไม่ว่าคุณจะเพิ่งเริ่มต้น หรือเป็นนักพัฒนาที่ต้องการก้าวสู่
ระดับที่สูงขึ้น

ท้ายที่สุดนี้ ขอขอบคุณผู้อ่านทุกท่านที่เลือกหยิบหนังสือเล่มนี้ และขอ
ให้ Git กลายเป็นเครื่องมือที่ทำให้การพัฒนาซอฟต์แวร์ของคุณราบ
รื่นขึ้นกว่าเดิม

สารบัญ

บทนำ	1
เข้าใจ Git จากรากฐานสู่การใช้งานจริง	1
Git คืออะไร	1
ปัญหาคลาสสิกที่ Git ถูกสร้างมาเพื่อแก้	2
ความแตกต่างระหว่าง Git และระบบอื่น	3
การเตรียมเครื่องมือที่จำเป็น	3
หนังสือเล่มนี้เหมาะกับใคร	4
สิ่งที่คุณจะได้จากหนังสือเล่มนี้	4
พร้อมเริ่มก้าวแรกไปกับ Git หรือยัง?	5
บทที่ 1: พื้นฐานของ Git	6
Repository คืออะไร	6
พื้นที่ทำงาน 3 ส่วนของ Git (Working Directory, Staging Area และ Repository)	6
การติดตั้ง Git	7
การตั้งค่าเริ่มต้น (git config)	9
การสร้าง Repository ใหม่ (git init)	9
การ Clone โปรเจกต์ (git clone)	10
สรุปท้ายบท	11

บทที่ 2: การทำงานกับไฟล์	13
ตรวจสอบสถานะไฟล์ (git status)	13
เพิ่มไฟล์เข้าสู่ Staging (git add)	14
บันทึกการเปลี่ยนแปลง (git commit)	15
ดูประวัติ Commit (git log)	16
การลบและเปลี่ยนชื่อไฟล์ด้วย Git	17
สรุปท้ายบท	18
บทที่ 3: การทำงานกับ Remote Repository	20
Remote คืออะไร และทำไมต้องใช้	20
การจัดการ Remote (git remote)	21
ดึงข้อมูลจาก Remote (git fetch, git pull)	22
git fetch	22
git pull	23
ส่งงานขึ้น Remote (git push)	24
สรุปท้ายบท	26
บทที่ 4: Branching & Merging	28
แนวคิดของ Branch	28
การสร้างและสลับ Branch	29
การรวม Branch (git merge)	31
วิธีแก้ Conflict ตอน Merge	32
Workflow การทำงานกับ Branch	34

Git Flow	34
Trunk-Based Development (TBD)	35
สรุปท้ายบท	36
บทที่ 5: Rebase และการจัดการประวัติ	37
แนวคิดของ Rebase คือ การเปลี่ยนฐานของ Branch	37
Rebase เทียบกับ Merge เลือกใช้อะไรดี?	38
Git Merge	38
Git Rebase	39
Workflow การ Rebase Feature Branch กับ Main	40
Interactive Rebase (git rebase -i)	41
การแก้ไขประวัติด้วย git reset	42
สรุปท้ายบท	43
บทที่ 6: การแก้ปัญหาและกู้คืนข้อมูล	45
การยกเลิกการแก้ไขใน Working Directory	45
git restore	45
git checkout --	46
การนำไฟล์ออกจาก Staging Area (unstage)	46
git restore --staged	46
การย้อนกลับ Commit ที่ Push ไปแล้ว (git revert)	47
การใช้ git reset เพื่อกู้คืน	48
สื่อดยอดเครื่องมือกู้ภัย: git reflog	48

การแก้ Conflict แบบละเอียด	50
สรุปท้ายบท	51
บทที่ 7: Git Stash เก็บงานด่วน	52
git stash คืออะไร?	52
การสร้าง Stash ใหม่	53
การเรียกคืนงาน	54
git stash pop	54
git stash apply	54
การจัดการ Stash หลายรายการ	55
Workflow ตัวอย่าง	56
สรุปท้ายบท	57
บทที่ 8: การทำงานร่วมกับทีม	58
Workflow สำหรับทีม	58
Pull Request / Merge Request หัวใจของการทำงานร่วมกัน	59
การตรวจสอบคุณภาพโค้ด (Code Review)	61
การแก้ Conflict แบบ Collaborative	62
วิธี Push แบบปลอดภัย --force-with-lease	63
สรุปท้ายบท	64
บทที่ 9: Git ขั้นสูง	65
Git Hooks: สคริปต์อัตโนมัติสำหรับ Workflow ของคุณ	65

Git Submodules: จัดการโปรเจกต์ซ้อนโปรเจกต์	66
การใช้ .gitignore อย่างมีประสิทธิภาพ	67
การแท็กเวอร์ชัน (Tagging)	69
การทำ Release	70
สรุปท้ายบท	71
บทที่ 10: การใช้ Git กับเครื่องมืออื่น	72
แพลตฟอร์ม Git Hosting (GitHub, GitLab, Bitbucket)	72
GitHub	72
GitLab	73
Bitbucket	74
เลือกอย่างไรให้เหมาะกับทีม	75
การใช้ CI/CD กับ Git	75
Continuous Integration (CI)	76
Continuous Deployment / Continuous Delivery (CD)	77
Git เป็นหัวใจของการ Trigger	77
เลือกใช้ Git GUI เมื่อ Command Line ไม่ใช่คำตอบเดียว	78
สรุปท้ายบท	79
บทที่ 11: เคสตัวอย่างการใช้งานจริง	81
เคสที่ 1: การพัฒนาฟีเจอร์ใหม่ในทีม	81
เคสที่ 2: การแก้บั๊กด่วนบน Production (Hotfix)	83
Best Practices สำหรับ Developer	85

สรุปท้ายบท	86
บทที่ 12: สูตรแก้ปัญหาที่พบบ่อย	88
สูตรที่ 1: ตามล่าหา Commit ต้นตอของบั๊ก ด้วย git bisect	88
เป้าหมาย	88
เครื่องมือที่ใช้	88
ขั้นตอนใช้งาน git bisect แบบ Step-by-Step	89
สูตรที่ 2: ลบไฟล์ข้อมูลลับ (Secret) ที่เผลอ Commit เข้าไปในประวัติแล้ว	91
เป้าหมาย	92
เครื่องมือที่ใช้	92
คำเตือนสำคัญ ก่อนทำ	92
ขั้นตอนลบไฟล์ลับออกจากประวัติ Git	93
สรุป	95
สูตรที่ 3: กู้คืน Branch ที่เผลอลบไป (git branch -D)	95
เป้าหมาย	96
เครื่องมือที่ใช้	96
วิธีการกู้คืน Branch ที่ถูกลบ	96
สรุป	97
สูตรที่ 4: แก้ปัญหาบรรทัดใหม่ (Line Ending) ต่างกันระหว่าง Windows และ macOS/Linux	98
เป้าหมาย	98

เครื่องมือที่ใช้	99
วิธีแก้ปัญหา Line Ending ข้ามระบบปฏิบัติการ	99
สรุป	100
สรุปท้ายบท	100
บทที่ 13: เบื้องหลังการทำงานของ Git	102
โฟลเดอร์ .git: หัวใจของ Repository	102
Git Objects: สิ่งก่อสร้างพื้นฐาน	103
Blob Object	103
Tree Object	104
Commit Object	104
Tag Object	104
ความสัมพันธ์ของ Git Objects และที่มาของประวัติ	105
References (Refs) และ Branch Pointers	107
Git Internals Commands	107
สรุปท้ายบท	108
ภาคผนวก	109
คำสั่ง Git ที่ใช้บ่อย	109
การตั้งค่าและเริ่มต้น	109
การทำงานพื้นฐาน	109
การทำงานกับ Remote	110
Branching & Merging	110