



สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID

DOP-08

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

DEVOPS / VERSION CONTROL

Git, GitHub and Azure Repos for Teams

ใช้ Git ทำงานเป็นทีมด้วย Pull Request และ Code Review บน GitHub และ Azure Repos

- 12 ชั่วโมง
- 2 วัน
- พื้นฐาน Command Line และแก้ไฟล์โค้ดเบื้องต้น
- ระดับ Beginner to Intermediate



ระยะเวลา: 2 วัน

12 ชั่วโมง (2 วัน วันละ 6 ชม.) · 09:00-16:00 น.



รูปแบบ: Classroom / Hybrid

Onsite หรือ Online ผ่าน Microsoft Teams · Public / In-house



รายละเอียดหลักสูตร

www.itgenius.co.th

ทีมพัฒนาซอฟต์แวร์จำนวนมากยังเจอปัญหาเดิม ๆ เช่น โค้ดทับกัน ไฟล์หาย ไม่รู้ว่าใครทำอะไร Merge แล้วระบบพัง หรือประวัติ Commit ทรนตรวจสอบย้อนหลังไม่ได้ เพราะสมาชิกใช้ Git ได้เพียงคำสั่งพื้นฐานโดยไม่เข้าใจสิ่งที่เกิดขึ้นเบื้องหลัง เมื่อเจอปัญหาจึงไม่กล้าใช้คำสั่งขั้นสูง ขณะที่หลายองค์กรใช้ทั้ง GitHub และ Azure DevOps ควบคู่กัน ทีมจึงต้องเข้าใจทั้งสองแพลตฟอร์มและรู้ว่าแต่ละแพลตฟอร์มควบคุมคุณภาพโค้ดอย่างไร

วันแรกปูพื้นฐาน Git ให้แน่นตั้งแต่กลไกภายใน การแตก Branch การ Merge และการแก้ Conflict ไปจนถึงเครื่องมือเชิงลึกอย่าง Interactive Rebase, Cherry-pick, Bisect และ Reflog เพื่อจัดระเบียบประวัติและกู้คืนงานได้อย่างมั่นใจ วันที่สองต่อยอดสู่การทำงานเป็นทีมบน GitHub และ Azure Repos ผ่าน Pull Request, Code Review, Rulesets และ Branch Policies โดยเทียบทั้งสองแพลตฟอร์มทีละขั้นตอน ผู้เรียนแบ่งกลุ่มทำ Workshop โปรเจกต์ TeamBoard Web เจอ Conflict จริงและรีวิวโค้ดของกันและกันจริง เพื่อกลับไปวางมาตรฐานการทำงานในทีมได้ทันที (อบรม 2 วัน วันละ 6 ชั่วโมง รวม 12 ชั่วโมง ระดับ Beginner to Intermediate)

เครื่องมือ

Git Git

GH GitHub

AZ Azure Repos

VS VS Code

PR Pull Request

ราคาค่าอบรม · PUBLIC / IN-HOUSE

6,900 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม โปรเจกต์ตัวอย่าง TeamBoard Web ชุด Cheat Sheet และ Certificate · ผู้เรียนใช้บัญชี GitHub และบัญชี Microsoft ของตนเอง (แบบ Free เพียงพอ) · รับผิดชอบ In-house ขอใบเสนอราคาได้

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th



01 วัตถุประสงค์ Objectives

- เข้าใจแนวคิดและกลไกภายในของ Git ทั้ง Working Directory, Staging Area, Repository และ Object Model
- ใช้คำสั่ง Git ที่จำเป็นในงานประจำวันได้ถูกต้อง ทั้งผ่าน Command Line และ VS Code
- สร้าง Branch, Merge, Rebase และแก้ไข Merge Conflict ได้อย่างมั่นใจ
- จัดระเบียบประวัติด้วย Interactive Rebase และ Cherry-pick ค้นหาต้นเหตุด้วย Bisect และกู้คืนงานด้วย Reflog ได้
- ทำงานกับ Remote Repository บน GitHub และ Azure Repos ได้
- ใช้ Pull Request และ Code Review ในการทำงานเป็นทีมได้ตามมาตรฐาน
- กำหนด Rulesets, Branch Protection และ Branch Policies เพื่อควบคุมคุณภาพโค้ดได้
- เลือก Branching Strategy และวางข้อตกลงการทำงานที่เหมาะสมกับทีมของตนเองได้

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนาซอฟต์แวร์ทุกสายงานที่ต้องการใช้ Git ทำงานร่วมกับทีมอย่างถูกต้อง
- นักพัฒนาที่ใช้ Git ได้เฉพาะคำสั่งพื้นฐานและต้องการเข้าใจการทำงานเชิงลึกขึ้น
- Tech Lead และหัวหน้าทีมที่ต้องการวางมาตรฐาน Pull Request และ Code Review
- ทีมที่กำลังใช้หรือย้ายระหว่าง GitHub และ Azure DevOps
- QA, DevOps และผู้ดูแลระบบที่ต้องทำงานกับ Repository ของทีมพัฒนา

03 จุดเด่นของหลักสูตร Highlights

- อัปเดตตาม Git 2.56 รวมถึง git add --resolved สำหรับงานแก้ Conflict
- เข้าใจ Git จากข้างใน Object Model, Rebase, Bisect และ Reflog
- ครอบคลุม 2 แพลตฟอร์ม GitHub และ Azure Repos เทียบกันทีละขั้นตอน
- จำลองทีมจริง แบ่งกลุ่ม เจอ Conflict จริง รีวิวโค้ดกันจริง
- พร้อมวางมาตรฐานทีม Branching Strategy และ PR Checklist

04 ความรู้พื้นฐาน Prerequisites

- ใช้งานคอมพิวเตอร์และจัดการไฟล์/โฟลเดอร์ได้คล่อง
- ใช้งาน Command Line หรือ Terminal ขั้นพื้นฐานได้
- มีพื้นฐานการเขียนโปรแกรมหรือแก้ไขไฟล์โค้ดเบื้องต้น ภาษาใดก็ได้
- มีบัญชี GitHub (แบบ Free) และบัญชี Microsoft สำหรับสร้าง Azure DevOps Organization แบบ Free
- ไม่จำเป็นต้องเคยใช้ Git, GitHub หรือ Azure DevOps มาก่อน ผู้ที่ใช้อยู่แล้วจะได้เพิ่มความเข้าใจเชิงลึกและแนวปฏิบัติระดับทีม

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook (Windows 10/11, macOS หรือ Linux) พร้อมสิทธิ์ติดตั้งโปรแกรม
Version Control	Git 2.56 ขึ้นไป พร้อม Git Credential Manager หรือ SSH Key
Editor และ GUI	Visual Studio Code, Source Control และส่วนขยาย Git Graph
Remote Repository	GitHub (บัญชี Free) และ Azure Repos ใน Azure DevOps Services
โปรเจกต์ตัวอย่าง	TeamBoard Web เว็บขนาดเล็ก HTML/CSS/JavaScript ใช้ข้อมูลจำลอง

หมายเหตุ: ผู้เรียนเตรียมบัญชี GitHub และบัญชี Microsoft ของตนเอง แล้วสร้าง Azure DevOps Organization แบบ Free (สมาชิกกลุ่มละ 3-4 คนอยู่ในสิทธิ์ผู้ใช้ฟรีได้) สถาบันไม่ได้จัดเตรียมบัญชีให้ Workshop บน GitHub ใช้ Repository แบบ Public เพราะ Branch Protection และ Rulesets บน Private Repository ต้องใช้แพ็คเกจแบบเสียเงิน ส่วน Copilot code review เป็นการสาธิตภาพรวม ผู้เรียนไม่จำเป็นต้องมีแพ็คเกจ Copilot หลักสูตรนี้ต่างจาก Git, GitHub and Bitbucket for Beginners ซึ่งเป็นหลักสูตรพื้นฐาน 2 วัน โดยเจาะลึก Git ทั้งหมดภายใน Interactive Rebase, Bisect และ Reflog แล้วเน้นการทำงานเป็นทีมผ่าน Pull Request, Code Review และกฎควบคุม Branch บน GitHub และ Azure Repos

06 ภาพรวมการเรียนรู้ 2 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	ปูพื้นฐาน Git และกลไกภายในอย่าง Object Model, Branch, Merge และการแก้ Conflict แล้วต่อยอดสู่เครื่องมือเชิงลึกอย่าง Interactive Rebase, Cherry-pick, Bisect, Blame, Reflog และ Git Hooks พร้อม Workshop TeamBoard Web บนเครื่องของตนเอง
Day 2 Section 6-10	Remote Repository และ Branching Strategy, Pull Request และ Code Review บน GitHub, Rulesets, CODEOWNERS และ GitHub Projects, Branch Policies และ Auto-complete บน Azure Repos, เปรียบเทียบสองแพลตฟอร์ม และ Master Workshop ทำงานเป็นทีมจนออก Release แรก



DAY
1

Git จากพื้นฐานสู่เครื่องมือเชิงลึก

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 Lab: Git Fundamentals และ Git Internals

- แนวคิด Version Control แบบกระจาย และโครงสร้าง Working Directory, Staging Area, Repository
- ตั้งค่า git config และคำสั่งหลัก init, status, add, commit, log, diff และ .gitignore
- หลักการเขียน Commit Message ที่ดี และรูปแบบ Conventional Commits
- Object Model ของ Git (Blob, Tree, Commit, Tag) รวมถึง Reference, Branch และ HEAD
- Lab: สร้าง Repository ของ TeamBoard Web แล้วเปิดดู Object ใน .git ด้วย git cat-file

2 Lab: Branching, Merging และ Conflict Resolution

- สร้างและสลับ Branch ด้วย git switch และ git branch
- Fast-forward Merge, Three-way Merge และ Rebase ควรเลือกใช้เมื่อใด
- แก้ Conflict ด้วย Command Line และ VS Code แล้ว Stage ด้วย git add --resolved (Git 2.56)
- ทำความสะอาด Branch ที่ Merge แล้ว และใช้ Tag กำหนดเวอร์ชัน
- Lab: จำลอง Conflict จากสอง Branch แล้วแก้ไขให้ถูกต้องทั้งสองวิธี

3 Lab: History Rewriting จัดระเบียบประวัติให้สะอาด

- แก้ไข Commit ล่าสุดด้วย git commit --amend
- Interactive Rebase (git rebase -i): squash, fixup, reword, reorder และ drop
- นำ Commit เฉพาะไปใช้ใน Branch อื่นด้วย git cherry-pick
- กฎทอง: เขียนประวัติใหม่เฉพาะ Branch ส่วนตัว และ Push ด้วย --force-with-lease อย่างปลอดภัย
- Lab: รวบรวม Commit ที่รกรกให้เป็นชุดที่อ่านง่ายก่อนส่งงาน

4 Lab: Debugging History, Recovery และ Git Hooks

- Log ขั้นสูงแบบกราฟและ Graph, git blame, git log -S และ git grep
- ค้นหา Commit ที่ทำให้เกิด Bug ด้วย git bisect ทดสอบ และแบบอัตโนมัติด้วย git bisect run
- git restore, git stash และความต่างระหว่าง revert กับ reset (soft, mixed, hard)
- กู้คืน Commit หรือ Branch ที่หายด้วย git reflog
- Git Hooks ฝั่ง Client เช่น pre-commit และ commit-msg สำหรับตรวจสอบก่อน Commit

5 Workshop Day 1: TeamBoard Web บนเครื่องของตนเอง

- Clone Repository ที่เตรียมไว้ซึ่งมีประวัติ Commit สกและ Bug ซ่อนอยู่
- พัฒนาฟีเจอร์ใหม่บน Branch แยก แล้วจำลองและแก้ Merge Conflict
- จัดระเบียบประวัติด้วย Interactive Rebase และหา Commit ที่ทำให้หน้าเว็บผิดด้วย git bisect
- Workshop: กู้คืน Branch ที่ถูกลบด้วย Reflog และติดตั้ง Hook ตรวจสอบรูปแบบ Commit Message

DAY
2

ทำงานเป็นทีมบน GitHub และ Azure Repos

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Remote Repository และ Branching Strategy ของทีม

- clone, remote, fetch, pull, push, Upstream Branch และการตั้งค่า pull แบบ Merge หรือ Rebase
- ยืนยันตัวตนด้วย SSH Key หรือ HTTPS ผ่าน Git Credential Manager, หลาย Remote และ Fork
- เปรียบเทียบ GitHub Flow, Trunk-based Development และ Git Flow ตามขนาดทีมและรอบ Release
- ข้อตกลงทีม: ชื่อ Branch, Commit Convention, ขนาด PR, ใครรีวิว และ Merge แบบใด
- Lab: เชื่อม Repository เดียวกันกับ GitHub และ Azure Repos แล้ว Push งานขึ้นทั้งสองที่

7 Lab: Pull Request และ Code Review บน GitHub

- สร้าง Repository เชิญสมาชิก และจัดการสิทธิ์ระดับ Repository และ Organization
- Pull Request, Draft Pull Request และการรีวิว: Comment, Suggest Changes, Request Changes, Approve
- รูปแบบการ Merge (Merge Commit, Squash, Rebase) และผลกระทบต่อประวัติของ Repository
- หลักการรีวิวโค้ดที่ดี และภาพรวม Copilot code review (ต้องใช้แพ็คเกจ Copilot แบบเสียเงิน)

8 Lab: GitHub Governance และ GitHub Projects

- Rulesets และ Branch Protection: บังคับรีวิว ห้าม Force Push และกำหนด Status Checks
- CODEOWNERS, Pull Request Template และ Issue Template
- เชื่อม Pull Request กับ Issues และติดตามงานด้วย GitHub Projects
- Lab: ตั้ง Ruleset และ CODEOWNERS แล้วเปิด Draft Pull Request ให้เพื่อนรีวิวจนพร้อม Merge

9 Lab: Azure Repos และการเปรียบเทียบกับ GitHub

- โครงสร้าง Organization, Project, Repository, Required และ Optional Reviewers และ Auto-complete
- Branch Policies: จำนวนผู้รีวิวขั้นต่ำ การเชื่อม Work Items การแก้ Comment ให้คน และ Build Validation
- เชื่อมงานกับ Work Items ใน Azure Boards
- เปรียบ Rulesets กับ Branch Policies และ Issues กับ Work Items พร้อมแนวทางทำงานข้ามแพลตฟอร์ม
- Lab: ตั้ง Branch Policies แล้วเปิด Pull Request ที่เชื่อม Work Item และตั้ง Auto-complete

10 Master Workshop: ทำงานเป็นทีมจนออก Release แรก

- แบ่งกลุ่ม 3-4 คน นำ TeamBoard Web ขึ้น GitHub และ Azure Repos แล้วกำหนดข้อตกลงของทีม
- ตั้ง Rulesets, CODEOWNERS และ Branch Policies แล้วแบ่งงานพัฒนาคนละฟีเจอร์
- เปิด Pull Request ที่เชื่อม Issue หรือ Work Item รีวิวงานของเพื่อน และแก้ Conflict ที่เกิดขึ้นจริง
- สรุป Pull Request Checklist และมาตรฐานการทำงานร่วมกันที่น่ากลับไปใช้ในทีม
- Workshop: Merge ทุกฟีเจอร์เป็น Release แรกพร้อม Tag เวอร์ชันบนทั้งสองแพลตฟอร์ม

07 เรียนจบแล้วได้อะไร Outcomes

- เข้าใจ Git อย่างแท้จริง รู้ว่าแต่ละคำสั่งทำอะไรกับ Repository
- จัดระเบียบประวัติได้ Interactive Rebase และ Cherry-pick
- ค้นหาและกู้คืนได้ Bisect หาต้นเหตุ Reflog กู้งานที่หาย
- ทำงานได้ 2 แพลตฟอร์ม GitHub และ Azure Repos พร้อมรู้ความต่าง
- มาตรฐาน Code Review Pull Request และกฎควบคุม Branch
- วาง Branching Strategy เลือกแนวทางที่เหมาะสมกับทีมได้ทันที

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรม และโปรเจกต์ตัวอย่าง TeamBoard Web พร้อมเอกสาร Lab ของแต่ละ Workshop
- Cheat Sheet คำสั่ง Git ทั้งงานประจำวันและเครื่องมือเชิงลึก พร้อมขั้นตอนกู้คืนงานในสถานการณ์ผิดพลาดที่พบบ่อย
- ตัวอย่าง Branching Strategy, Commit Convention, Git Hook ตรวจ Commit Message, Pull Request Template และ Pull Request Checklist สำหรับทีม
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร ไกยม วิทยากรของสถาบันไอทีจีเนียสสอน Git แบบอธิบายกลไกเบื้องหลังให้เห็นภาพ แล้วพาลงมือทำกันที่ในสถานการณ์ที่ทีมพัฒนาเจอจริง ทั้งการแก้ Conflict การกู้คืนงาน และการรีวิวโค้ดผ่าน Pull Request ผู้เรียนจึงไม่ได้แค่จำคำสั่ง แต่เข้าใจเหตุผลของแต่ละขั้นตอน และนำมาตรฐานที่ได้กลับไปใช้กับทีมของตนเองได้ทันที

ราคาค่าอบรม 6,900 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

6,900 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE