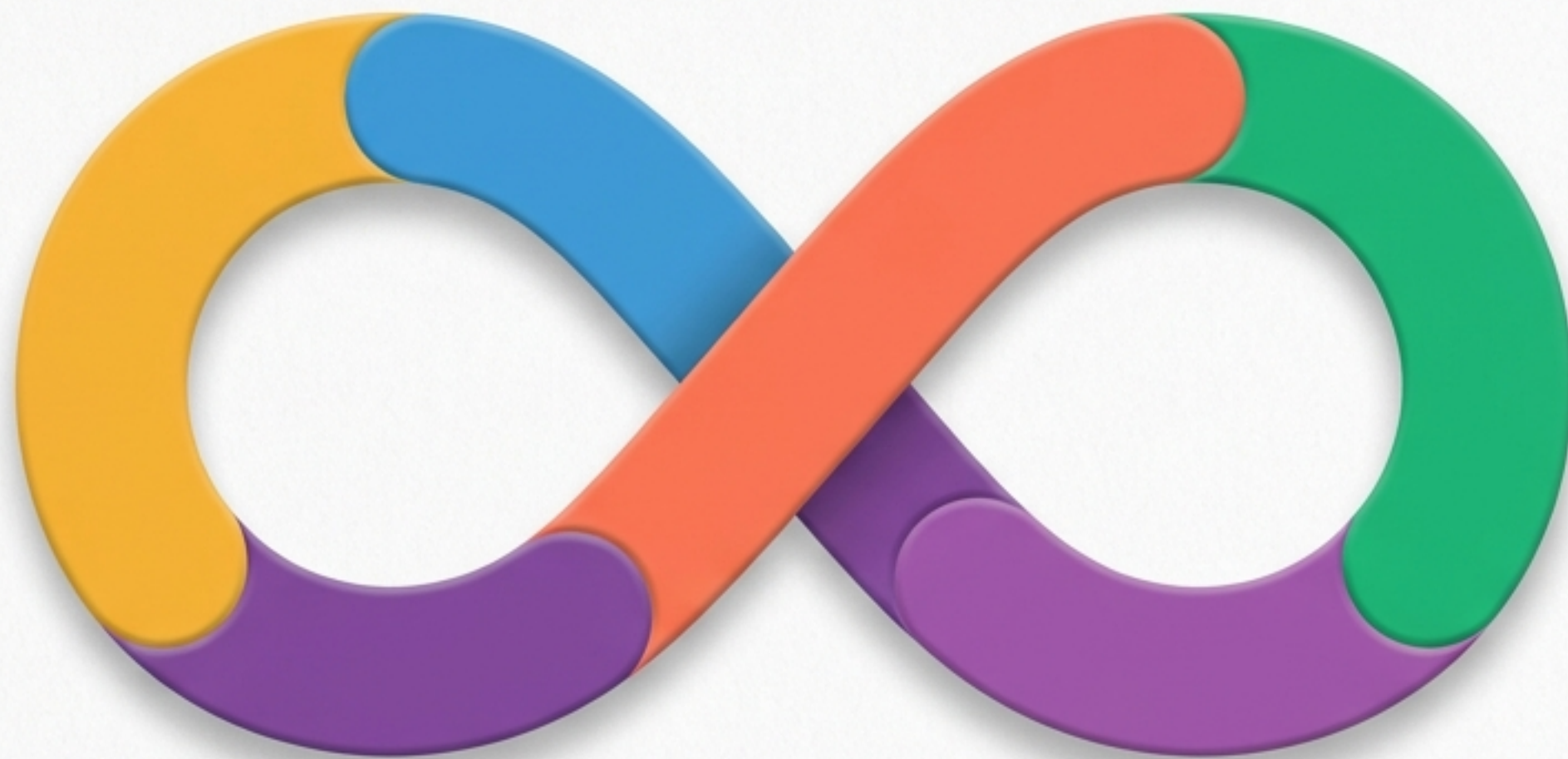


วงจรชีวิตของ DevOps

คู่มือฉบับสมบูรณ์เพื่อการส่งมอบซอฟต์แวร์ที่รวดเร็วและมีประสิทธิภาพ



DevOps คืออะไร?

กระบวนการต่อเนื่องที่ผสานงาน พัฒนา (Dev) และ ปฏิบัติการ (Ops) เข้าด้วยกันตลอดการส่งมอบซอฟต์แวร์

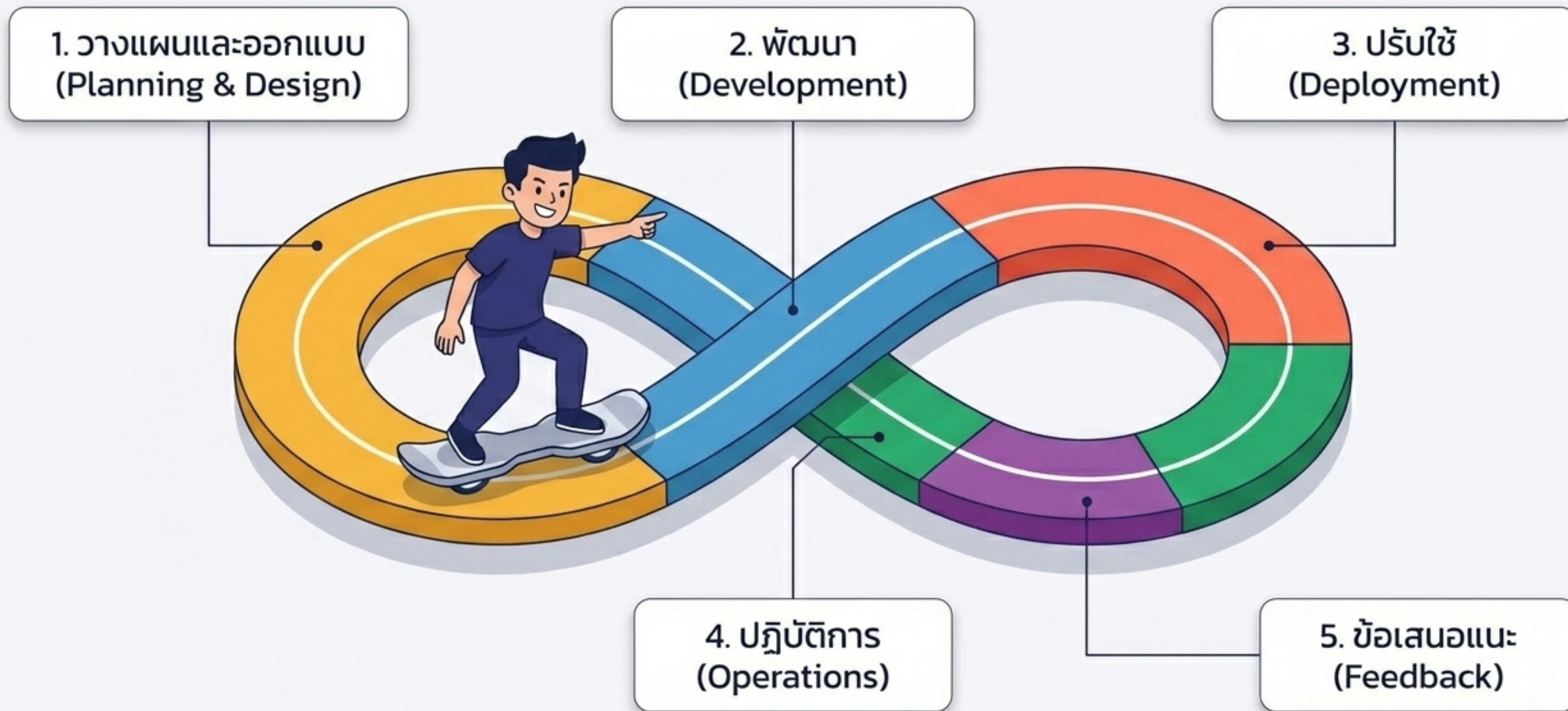
ทำงานร่วมกันเป็นทีมได้ดีขึ้น

ทำงานอัตโนมัติ



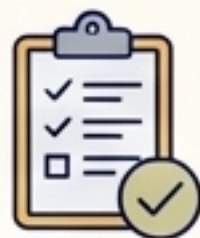
ปล่อยเวอร์ชันใหม่รวดเร็วและน่าเชื่อถือ

5 ขั้นตอนสำคัญของวงจรชีวิต DevOps

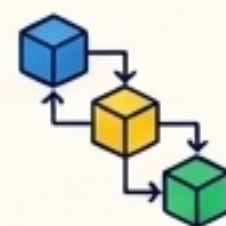


ขั้นตอนที่ 1: การวางแผนและการออกแบบ (Planning & Design)

พื้นฐานสำคัญเพื่อตอบโจทย์เป้าหมายทางธุรกิจและเทคโนโลยี



รวบรวมข้อกำหนด:
เป้าหมายธุรกิจ
เทคนิค และ
ความปลอดภัย



สถาปัตยกรรม
โซลูชัน:
แบ่งย่อยระบบ
(Microservices)
และวางแผน
Data Flow



ออกแบบ
โครงสร้าง
พื้นฐาน:
Cloud vs
On-premise,
ขยายขนาดได้
(Scalability)



เลือกเครื่องมือ:
ตัดสินใจเลือก
Tech Stack
สำหรับ DevOps

เจาะลึกสถาปัตยกรรมและ โครงสร้างพื้นฐาน



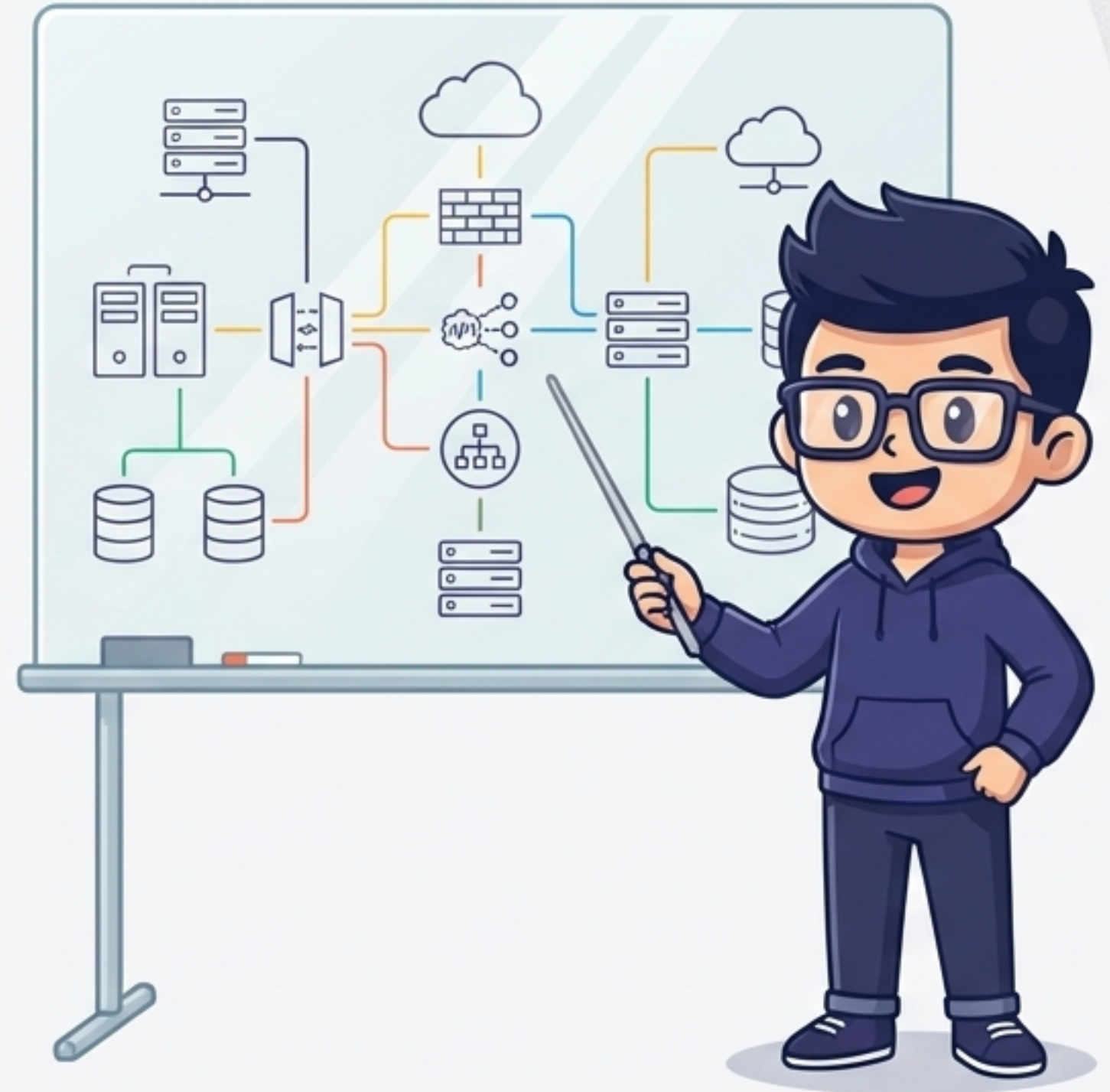
สถาปัตยกรรม

- **Microservices:** แบ่งแอปพลิเคชันเป็นส่วนย่อย
- **Integration:** เชื่อมต่อ API บุคคลที่สามและวางแผนการเคลื่อนย้ายข้อมูล



โครงสร้างพื้นฐาน

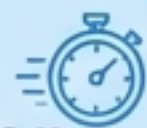
- **Environment:** เลือกระหว่างระบบคลาวด์ (Cloud) หรือระบบภายในองค์กร (On-premise)
- **Scalability & Network:** วางแผนเครือข่ายกระจายโหลด และรองรับปริมาณงานที่เพิ่มขึ้น



ขั้นตอนที่ 2: การพัฒนา (Development)

การเขียน จัดการ และทดสอบโค้ด
ในรูปแบบที่เป็นมิตรกับทีม
เพื่อส่งมอบงานอย่างรวดเร็ว

Sprints:
แบ่งงานเป็นส่วนย่อย
เพื่อความรวดเร็ว



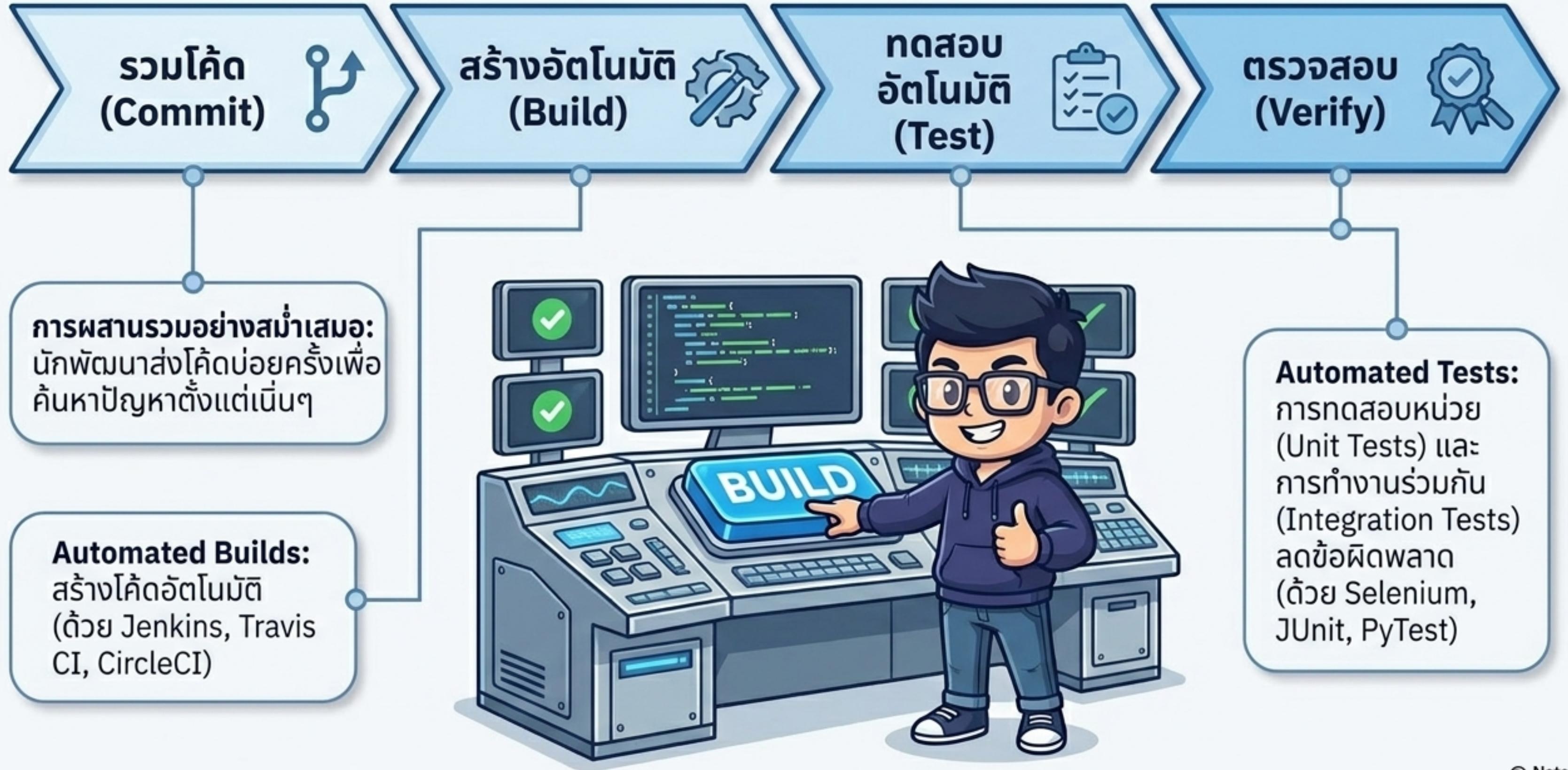
Collaboration:
ตรวจสอบโค้ด
(Code Review) และ
Pair Programming



Version Control:
ติดตามการเปลี่ยนแปลง
และจัดการสาขา
(Branching) ด้วย
Git หรือ SVN
ด้วย Git หรือ SVN



Continuous Integration (CI) และการทดสอบ

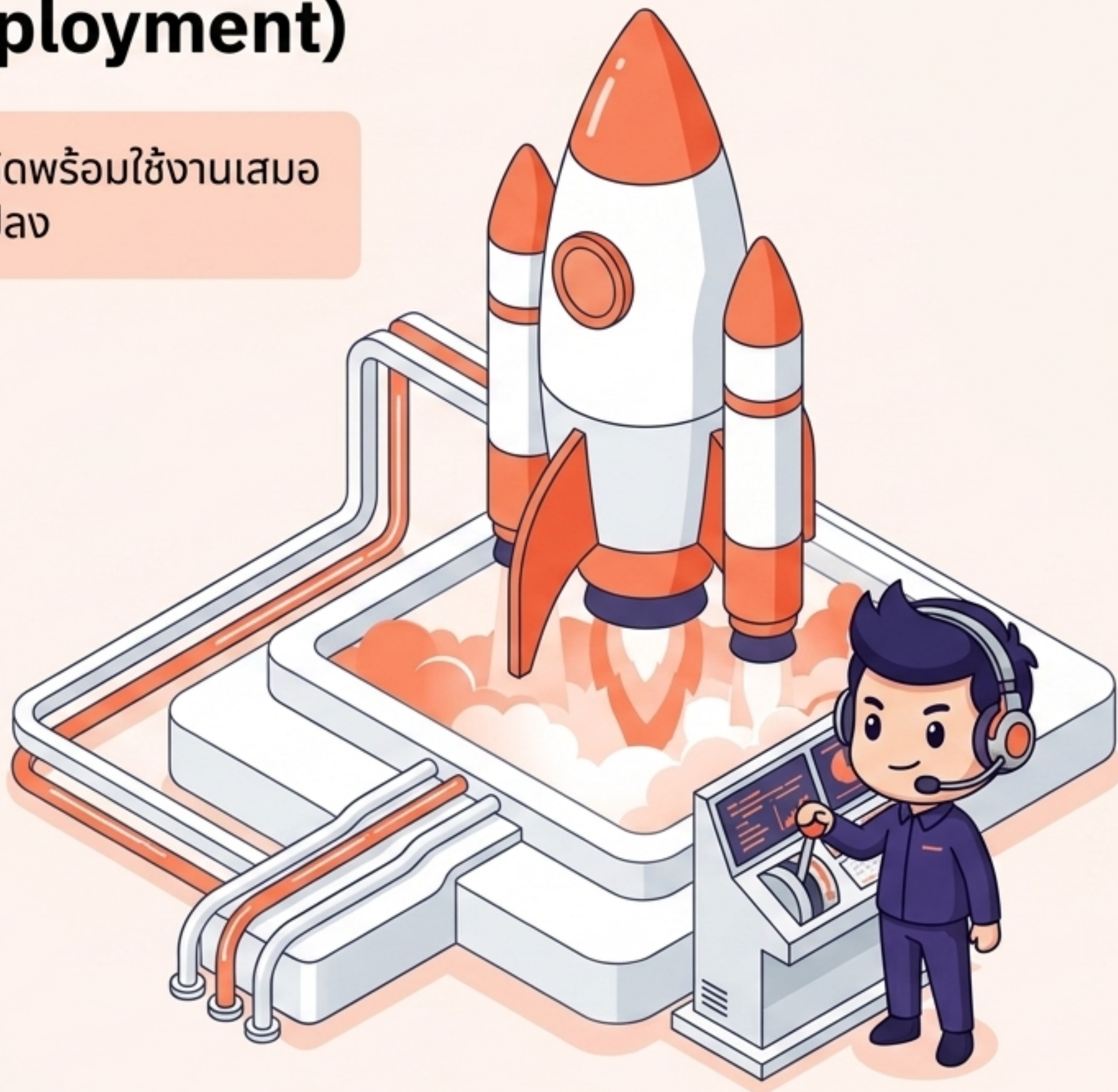


ขั้นตอนที่ 3: การใช้งานจริง (Deployment)

การส่งมอบอย่างต่อเนื่อง (Continuous Delivery) - โค้ดพร้อมใช้งานเสมอ
ทดสอบอัตโนมัติและพร้อมปรับใช้ทันทีหลังการเปลี่ยนแปลง

นำไปใช้งานได้ทุกเมื่อ:
ปล่อยการเปลี่ยนแปลงให้ผู้ใช้ได้บ่อยขึ้น

Deployment Pipelines:
กระบวนการอัตโนมัติที่จัดการตั้งแต่สร้าง
ทดสอบ จนถึงปรับใช้ ปราศจากข้อผิดพลาด



โครงสร้างพื้นฐานในรูปแบบโค้ด (IaC)

ระบบเดิม (Manual Setup)



ตั้งค่าด้วยมือ: ใช้เวลานาน
เสี่ยงต่อข้อผิดพลาดในการตั้งค่า

IaC (Infrastructure as Code)



- **IaC:** จัดการโครงสร้างพื้นฐานด้วยสคริปต์โค้ดแทนการ
ทำด้วยมือ (Terraform, Ansible)
- **Configuration Management:** ติดตามการเปลี่ยนแปลง
และรักษาความสม่ำเสมอของระบบ (Chef, Puppet)

ขั้นตอนที่ 4: การดำเนินงาน (Operations)

ตรวจสอบระบบ จัดการเหตุการณ์ และทำให้แอปพลิเคชันทำงานได้อย่างราบรื่น

การตรวจสอบและบันทึกข้อมูล (Monitoring & Logging): ติดตามตัวชี้วัด ค้นพบปัญหาได้ก่อนกระทบผู้ใช้

การวางแผนกำลังการผลิต (Capacity Planning): คาดการณ์ทรัพยากร (เซิร์ฟเวอร์/พื้นที่) ให้เพียงพอต่อความต้องการเพื่อหลีกเลี่ยงระบบล่ม



การจัดการเหตุการณ์และการเพิ่มประสิทธิภาพ



รับมือปัญหาอย่างรวดเร็ว (Incident Management)

มีกระบวนการตอบสนองที่ชัดเจน
-> ลดเวลาหยุดทำงาน (Downtime)
-> เรียนรู้เพื่อป้องกันไม่ให้เกิดซ้ำ

การเพิ่มประสิทธิภาพ (Performance Optimization)

ค้นหาจุดที่เป็นปัญหา (Bottlenecks)
และปรับแต่งระบบเพื่อเวลาตอบสนอง
ที่รวดเร็วที่สุด

ขั้นตอนที่ 5: ข้อเสนอแนะและการปรับปรุง (Feedback & Improvement)

เรียนรู้จากกระบวนการทำงานและผู้ใช้
เพื่อพัฒนาผลิตภัณฑ์ให้ดียิ่งขึ้นอย่างต่อเนื่อง

วงจรป้อนกลับต่อเนื่อง: รับฟังความเห็นเชิงลึก
จากผู้ใช้หลังการอัปเดตทุกครั้ง

การประชุมทบทวน (Retrospectives): ทีมแชร์
สิ่งที่ทำได้ดีและต้องปรับปรุงหลังจบแต่ละสปรินต์



การพัฒนาแบบวนซ้ำและการตัดสินใจด้วยข้อมูล

1

ตัดสินใจด้วยข้อมูล: ใช้ข้อเท็จจริง และตัวชี้วัด (KPI) เพื่อเป็นแนวทาง ลดการคาดเดาและได้ผลลัพธ์ที่ดีกว่า



2

การพัฒนาแบบวนซ้ำ (Iterative): ปล่อยอัปเดตบ่อยขึ้นทีละเล็กทีละน้อย ตอบสนองต่อการเปลี่ยนแปลงความต้องการได้อย่างทันถ่วงที



สรุปเครื่องมือสำคัญในวงจรชีวิต DevOps



วางแผน/พัฒนา (Dev)	ทดสอบ/CI (Integrate)	ปรับใช้/CD (Deploy)	ดำเนินงาน/ข้อเสนอแนะ (Ops)
Git, SVN (Version Control)	Jenkins, CircleCI, Travis CI Selenium, JUnit, PyTest	Terraform, Ansible (IaC) Chef, Puppet (Config Management)	[Monitoring Tools] [Log Management]

วัฒนธรรมแห่ง DevOps (The Culture of DevOps)



**DevOps ไม่ใช่แค่ชุดเครื่องมือ แต่คือ วัฒนธรรมการทำงาน
ที่เน้นความร่วมมือ การทำงานอัตโนมัติ และการตัดสินใจด้วยข้อมูล**

เมื่อวงจรชีวิตของ DevOps สมบูรณ์แบบ เราจะสามารถส่งมอบซอฟต์แวร์ที่รวดเร็ว
ปลอดภัย และสร้างประสบการณ์ที่ดีที่สุดให้กับผู้ใช้งานอย่างยั่งยืน