กรมควบคุมโรค
DEPARTMENT OF DISEASE CONTROL

(ร่าง)

เอกสารขั้นตอนปฏิบัติในการพัฒนาซอฟต์แวร์อย่างมั่นคงปลอดภัยสารสนเทศ (Secure Software Development)

ประวัติการแก้ไขเอกสาร

ครั้งที่	วันที่	รายละเอียดการดำเนินการ
1	28 ก.พ. 2568	Draft Version 1

สารบัญ

1. วัตถุประสงค์

เพื่อให้การพัฒนาซอฟต์แวร์มีความมั่นคงปลอดภัย ตลอดวงจรชีวิตของการพัฒนาซอฟต์แวร์ให้เป็นไปอย่างปลอดภัย โดยยึดตามหลักการที่ได้ระบุไว้ในกรอบการพัฒนาซอฟต์แวร์ที่ปลอดภัยตาม NIST Secure Software Development Framework (SSDF) และความมั่นคงปลอดภัยสารสนเทศตามมาตรฐาน ISO/IEC 27001

2. ขอบเขต

เอกสารฉบับนี้ใช้กับการพัฒนาซอฟต์แวร์ทั้งหมดภายในกรมควบคุมโรค รวมทั้งการพัฒนาซอฟต์แวร์ของบุคคลที่สาม (Third Parties) และผู้ที่จ้างเหมาช่วง

3. บทบาทและความรับผิดชอบ

3.1. Chief Information Security Officer (CISO)

- กำหนดนโยบาย (Policy Development) ความปลอดภัยสารสนเทศที่ชัดเจน ครอบคลุมทุกแผนก เช่น นโยบายการจัดการรหัสผ่าน, นโยบายการเข้าถึงข้อมูล, นโยบายการป้องกันและตอบสนองต่อภัยคุกคามทางไซเบอร์ เป็นต้น
- กำหนดโครงสร้างหน้าที่และความรับผิดชอบ ที่ระบุอย่างชัดเจนว่าใครมีหน้าที่รับผิดชอบในส่วนใด เพื่อให้การปฏิบัติงานชัดเจน และไม่มีความสับสน
- จัดการสื่อสารและอบรมนโยบาย (Communication and Training) ไปยังพนักงานทุกระดับในองค์กร เพื่อให้ทุกคนเข้าใจบทบาทและหน้าที่ของตนเอง รวมถึงความสำคัญของการปฏิบัติตามนโยบายด้านความปลอดภัยสารสนเทศ
- ตรวจสอบและปรับปรุงนโยบาย (Review & Improvement) โดยต้องติดตามการนำไปปฏิบัติและประเมินผลอย่างสม่ำเสมอ เพื่อปรับปรุงนโยบายให้สอดคล้องกับภัยคุกคามใหม่ๆ รวมทั้งปรับปรุงนโยบายตามข้อเสนอแนะหรือจากผลการตรวจสอบภายในหรือภายนอกองค์กร

3.2. ฝ่ายจัดซื้อ (Procurement Team)

- รวมข้อกำหนดด้านความปลอดภัยในสัญญา (Security Clauses) ระบุเงื่อนไขหรือข้อกำหนดด้านความปลอดภัยที่ผู้ขายหรือคู่สัญญาต้องปฏิบัติ เช่น การปฏิบัติตามมาตรฐาน ISO/IEC 27001, การแจ้งเหตุละเมิดข้อมูล (Data Breach Notification)
- ประสานงานกับทีมความปลอดภัย (Security Coordination) ปรึกษาทีม Security หรือ Risk & Compliance เพื่อระบุข้อกำหนดเฉพาะทางที่เหมาะสมกับบริการหรือผลิตภัณฑ์ที่องค์กรจะจัดซื้อหรือใช้บริการ

- การประเมินและตรวจสอบผู้ขาย (Vendor Assessment) ดำเนินการหรือช่วยดำเนินการประเมินผู้ขายในด้านความปลอดภัย เพื่อให้มั่นใจว่าผู้ขายมีแนวปฏิบัติด้านความปลอดภัยที่ตรงตามมาตรฐานองค์กรก่อนจะทำสัญญา
- ติดตามและตรวจสอบการปฏิบัติตามสัญญา (Contract Compliance Monitoring) ดูแลให้ผู้ขายปฏิบัติตามข้อกำหนดที่ระบุไว้ในสัญญาตลอดระยะเวลาของการทำสัญญา

3.3. ทีมบริหารความเสี่ยงของผู้ให้บริการภายนอก (Third-Party Risk Management (TPRM) Team)

- ประเมินความปลอดภัยของผู้ขาย (Vendors), ผู้ให้บริการ (Service Providers) และพันธมิตรทางธุรกิจ (Third Parties) เพื่อให้แน่ใจว่าผู้ให้บริการภายนอกที่องค์กรทำงานด้วยมีแนวทางปฏิบัติด้านความปลอดภัยสารสนเทศที่สอดคล้องกับมาตรฐาน เช่น ISO 27001, ISO 27701, NIST, SOC 2, GDPR และ PDPA
- จัดทำนโยบายและเกณฑ์การประเมิน (Establish Vendor Risk Policies & Criteria)
- รวบรวมข้อมูลจาก Vendor (Collect Vendor Information) เกี่ยวกับมาตรการความปลอดภัยของ Vendor เช่น ISO 27001 Certification, SOC 2 Report, DPA (Data Processing Agreement)
- ประเมินความเสี่ยงของ Vendor (Conduct Security Risk Assessment) โดยใช้ Security Questionnaire (แบบสอบถามความปลอดภัย) เพื่อช่วยในการประเมินความเสี่ยงที่อาจเกิดขึ้นในการทำงาน
- จัดทำรายงานความเสี่ยงและข้อเสนอแนะ (Risk Report & Remediation Plan)
- ติดตามและตรวจสอบความเสี่ยงของ Vendor อย่างต่อเนื่อง (Ongoing Monitoring)

3.4. ทีมกฎหมาย (Legal Team)

- วิเคราะห์และกำหนดข้อกำหนดด้านกฎหมาย (Legal & Compliance Analysis)
- ตรวจสอบว่าข้อตกลง Vendor สอดคล้องกับ กฎหมายและมาตรฐานสากลที่เกี่ยวข้องกับข้อมูลและความปลอดภัย เช่น ISO/IEC 27001, ISO/IEC 27701, GDPR (General Data Protection Regulation), PDPA (Personal Data Protection Act)
- รวมข้อกำหนดด้านความปลอดภัยและกฎหมายในสัญญา (Drafting Security & Compliance Clauses) ที่ Vendor ต้องปฏิบัติตาม เช่น Data Protection Agreement (DPA), Confidentiality & Non-Disclosure Agreements (NDA), Incident Reporting Requirements, Data Retention & Deletion Policy เป็นต้น
- ตรวจสอบและเจรจากับ Vendor (Contract Review & Negotiation) ว่าสัญญาที่ Vendor เสนอมีความสอดคล้องกับข้อกำหนดขององค์กร และต่อรองเงื่อนไขหากพบว่าสัญญาไม่เป็นไปตามข้อกำหนดด้านความปลอดภัยและกฎหมาย

- บังคับใช้และติดตามผลการปฏิบัติตามสัญญา (Enforcement & Compliance Monitoring) ร่วมกับ Procurement และ TPRM Team เพื่อติดตามว่า Vendor ปฏิบัติตามข้อกำหนดในสัญญาหรือไม่

3.5. ทีมสร้างความตระหนักด้านความปลอดภัยสารสนเทศ (Security Awareness & Training Team)

- ออกแบบหลักสูตรการฝึกอบรมด้านความปลอดภัย (Develop Security Training Programs) เกี่ยวกับการทำ Secure Coding สำหรับนักพัฒนา โดยอิงตามแนวทางของ OWASP Top 10, NIST, และ ISO/IEC 27001
- จัดอบรมและเวิร์กชอปเกี่ยวกับ Secure Coding & SSDLC (Conduct Training & Workshops) และฝึกอบรมเกี่ยวกับแนวทาง Secure Software Development เช่น
  - การจัดการ Input Validation เพื่อป้องกัน SQL Injection
  - การป้องกัน Cross-Site Scripting (XSS)
  - แนวทางป้องกันการรั่วไหลของข้อมูล (Data Leakage Prevention)
  - สอนวิธีใช้ Static Code Analysis Tools (SAST) และ Dynamic Application Security Testing (DAST)
- ทำให้การเรียนรู้เป็นเรื่องสนุก (Gamification & Hands-on Exercises) โดยการใช้ Capture The Flag (CTF) Challenges ให้ผู้เข้าอบรมฝึกแก้ไขช่องโหว่จากโค้ดจริง หรือใช้เครื่องมืออย่าง OWASP WebGoat หรือ Juice Shop เพื่อให้ผู้เข้าอบรมเรียนรู้จากตัวอย่างจริง
- ประเมินผลการอบรมและติดตามผล (Training Evaluation & Continuous Improvement) โดยวิธีการทำ Pre-Test / Post-Test เพื่อวัดผลความเข้าใจของผู้เข้าอบรม

3.6. IT & Infra Team / Security Architecture Team

- กำหนด Security Baselines (Baseline Definition) ระบุและกำหนดการตั้งค่าด้านความปลอดภัยพื้นฐานของอุปกรณ์ทุกประเภท เช่น ระดับการเข้ารหัส, การตั้งค่าการใช้งาน Firewall, การอัปเดตและติดตั้งแพตช์รักษาความปลอดภัย, นโยบายการตั้งรหัสผ่านที่เข้มงวด
- นำ Security Baselines ไปปฏิบัติ (Implementation) โดยนำข้อกำหนดด้านความปลอดภัยไปติดตั้งและตั้งค่าในทุกระบบอย่างเป็นรูปธรรม เช่น การตั้งค่า Windows Server, Linux Server, Cloud Infrastructure, อุปกรณ์เครือข่าย และ Endpoint Devices ให้มีมาตรฐานเดียวกันทั่วทั้งองค์กร
- ตรวจสอบและดูแลรักษา (Monitoring & Maintenance) ให้แน่ใจว่าระบบและอุปกรณ์ยังคงมี Security Baselines ที่ถูกต้องอยู่เสมอ รวมถึงดำเนินการปรับปรุงแก้ไขเมื่อพบช่องโหว่หรือภัยคุกคามใหม่ๆ โดยอัปเดตให้ทันสมัยอย่างต่อเนื่อง

- จัดทำเอกสารและคู่มือ (Documentation) บันทึกขั้นตอน วิธีการ และมาตรฐาน Security Baselines ที่ชัดเจน เพื่อเป็นคู่มือให้ทีม IT ใช้อ้างอิงในการปฏิบัติงานและให้ผู้ตรวจสอบภายใน/ภายนอกตรวจสอบได้ง่าย
- ออกแบบ พัฒนา และกำหนดตัวชี้วัดของสถาปัตยกรรมด้านการรักษาความมั่นคงปลอดภัยของระบบ

3.7. หัวหน้าทีมพัฒนา / นักวิเคราะห์ระบบ (Development Leads / System Analyst)

- วิเคราะห์และออกแบบระบบให้สอดคล้องตามข้อกำหนด และความต้องการที่ได้รับ โดยให้คำนึงถึงความมั่นคงปลอดภัยร่วมด้วย
- นำ Security Guidelines มาใช้ในทีมพัฒนา เช่น OWASP Secure Coding Practices และ Security Baselines ที่กำหนดไว้จากทีม Security Architecture ไปใช้กับทีมของตนเอง
- สนับสนุนให้ทีมเขียนโค้ดที่ปลอดภัย (Secure Coding) ส่งเสริมให้สมาชิกในทีมเข้าใจและใช้หลักการเขียนโค้ดอย่างปลอดภัย เช่น ตรวจสอบ Input Validation, ป้องกัน SQL Injection, Cross-Site Scripting (XSS) และช่องโหว่อื่นๆ ที่อาจเกิดขึ้น
- ผสานการทดสอบความปลอดภัย (Security Testing) ในขั้นตอนพัฒนา วางแผนให้มีการตรวจสอบความปลอดภัยของโค้ดที่เขียน ทั้ง Static Analysis และ Dynamic Analysis เพื่อให้พบจุดอ่อนด้านความปลอดภัยได้ตั้งแต่ช่วงต้นๆ ของการพัฒนา
- เมื่อพบช่องโหว่ในระหว่างการพัฒนาหรือทดสอบ Development Lead ต้องจัดลำดับความสำคัญและให้ทีมทำการแก้ไขช่องโหว่นั้นอย่างชัดเจนและรวดเร็ว ก่อนจะปล่อยระบบขึ้นใช้งานจริง
- ประสานงานกับทีมความปลอดภัย (Security Teams) ร่วมมือและปรึกษากับ Security Teams หรือที่ปรึกษาด้านความปลอดภัย เพื่อให้มั่นใจว่าการดำเนินการพัฒนาเป็นไปตามมาตรฐานและแนวทางที่องค์กรกำหนด

3.8. ทีมรักษาความมั่นคงปลอดภัยสารสนเทศ (Security Team / Security Tester)

- ให้คำแนะนำ และกำกับดูแลด้านการรักษาความมั่นคงปลอดภัยสารสนเทศ
- ดำเนินการประเมินความมั่นคงปลอดภัยสารสนเทศเป็นระยะ ๆ ในรูปแบบ ดังนี้
    - Static Application Security Testing (SAST) เพื่อตรวจสอบช่องโหว่ของโค้ดแบบ Static ก่อน Deploy โดยผสานเข้าไปกับกระบวนการพัฒนา และระบบในการควบคุมเวอร์ชัน
    - Dynamic Application Security Testing (DAST) เพื่อจำลองการโจมตีจริงกับแอปพลิเคชันที่รันอยู่ โดยกำหนดให้ทดสอบความปลอดภัยของ API และ Web Apps
- นำเครื่องมือ Software Composition Analysis (SCA) มาใช้งาน เพื่อตรวจสอบ Dependencies และ Open-Source Libraries ที่อาจมีช่องโหว่

- ทำ Infrastructure Security Scanning เพื่อตรวจสอบความปลอดภัยของระบบโครงสร้างพื้นฐานด้านเทคโนโลยีสารสนเทศ

3.9. การรับรองคุณภาพ (QA / Test Engineer)

- ดำเนินการตรวจสอบข้อกำหนดด้านความมั่นคงปลอดภัยสารสนเทศระหว่างดำเนินการทดสอบ
- กำกับดูแลให้มีการดำเนินการควบคุมด้านความปลอดภัยสารสนเทศให้เป็นไปตามแนวปฏิบัติที่ได้กำหนดไว้

4. ความมั่นคงปลอดภัยพื้นฐานของการพัฒนาระบบให้มีความมั่นคงปลอดภัย (Secure development baseline)

สำหรับการบริหารจัดการด้านความมั่นคงปลอดภัยสารสนเทศของระบบงาน หรือแอพพลิเคชันปัจจุบันของกรมควบคุมโรค ได้เลือกใช้ NIST Secure Software Development Framework (SSDF) เป็นกรอบแนวคิดหลัก เนื่องจากเป็นแนวทางที่พัฒนาโดย National Institute of Standards and Technology (NIST) ซึ่งได้รับการยอมรับในระดับสากลในด้านการบริหารจัดการความมั่นคงปลอดภัยของซอฟต์แวร์ในทุกขั้นตอนของการพัฒนา การจัดหา และการบำรุงรักษา

หลักการสำคัญของ NIST Secure Software Development Framework (SSDF)

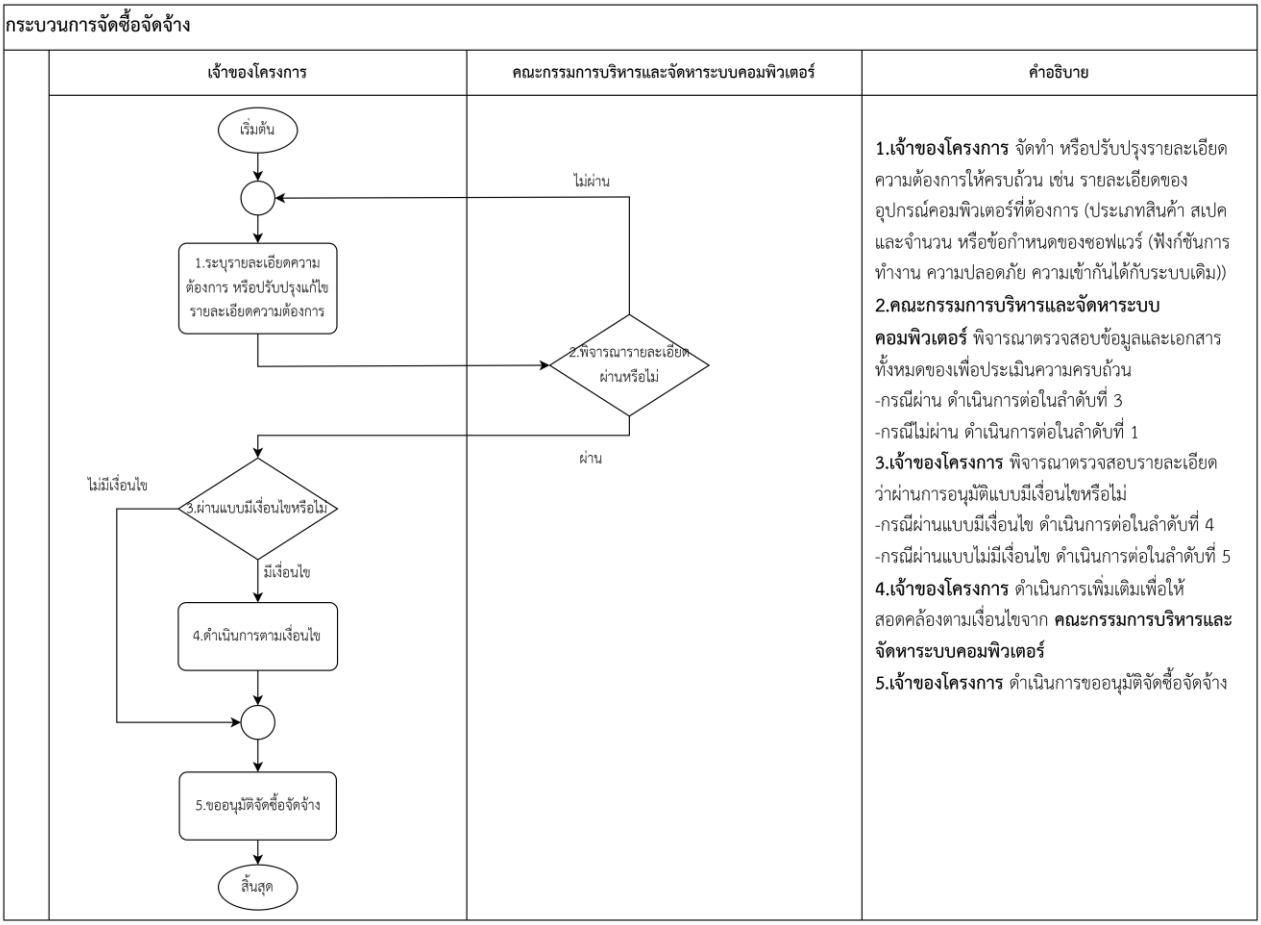
NIST Secure Software Development Framework (SSDF) ประกอบด้วยหลักการสำคัญ 4 ด้าน ซึ่งครอบคลุมทุกมิติของความมั่นคงปลอดภัยในกระบวนการพัฒนาซอฟต์แวร์ ดังนี้

1. **Prepare the Organization (การเตรียมองค์กร)** พื้นฐานความมั่นคงปลอดภัยในกระบวนการพัฒนาซอฟต์แวร์ เช่น การจัดทำนโยบาย การกำหนดบทบาทหน้าที่ และการสร้างสภาพแวดล้อมที่ปลอดภัยในการพัฒนาซอฟต์แวร์
2. **Protect the Software (การปกป้องซอฟต์แวร์)** ใช้แนวทางปฏิบัติ เช่น การควบคุมการเข้าถึง การจัดการองค์ประกอบของซอฟต์แวร์ และการปฏิบัติตาม Secure Coding Standards
3. **Produce Well-Secured Software (การผลิตซอฟต์แวร์ที่มีความมั่นคงปลอดภัย)** การพัฒนาซอฟต์แวร์ และการทดสอบความมั่นคงปลอดภัย เช่น การทดสอบโค้ด (SAST/DAST) และการตรวจสอบความสมบูรณ์ของซอฟต์แวร์
4. **Respond to Vulnerabilities (การตอบสนองต่อช่องโหว่)** จัดทำกระบวนการจัดการช่องโหว่ เช่น การติดตามและแก้ไขช่องโหว่ที่พบของซอฟต์แวร์

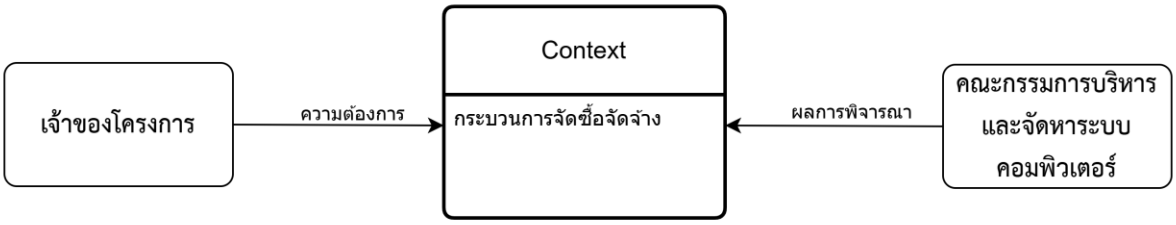
ภาพที่ 1 NIST SSDF ประกอบหลักการสำคัญ 4 ด้าน

5. ขั้นตอนปฏิบัติ

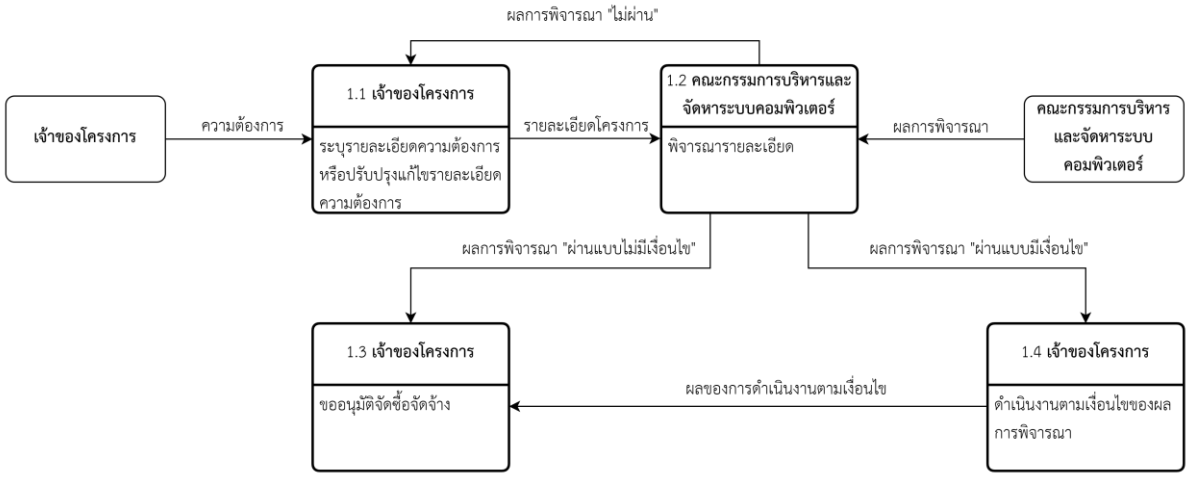
5.1 กระบวนการจัดซื้อจัดจ้าง



5.1.1 บริบทกระบวนการจัดซื้อจัดจ้าง

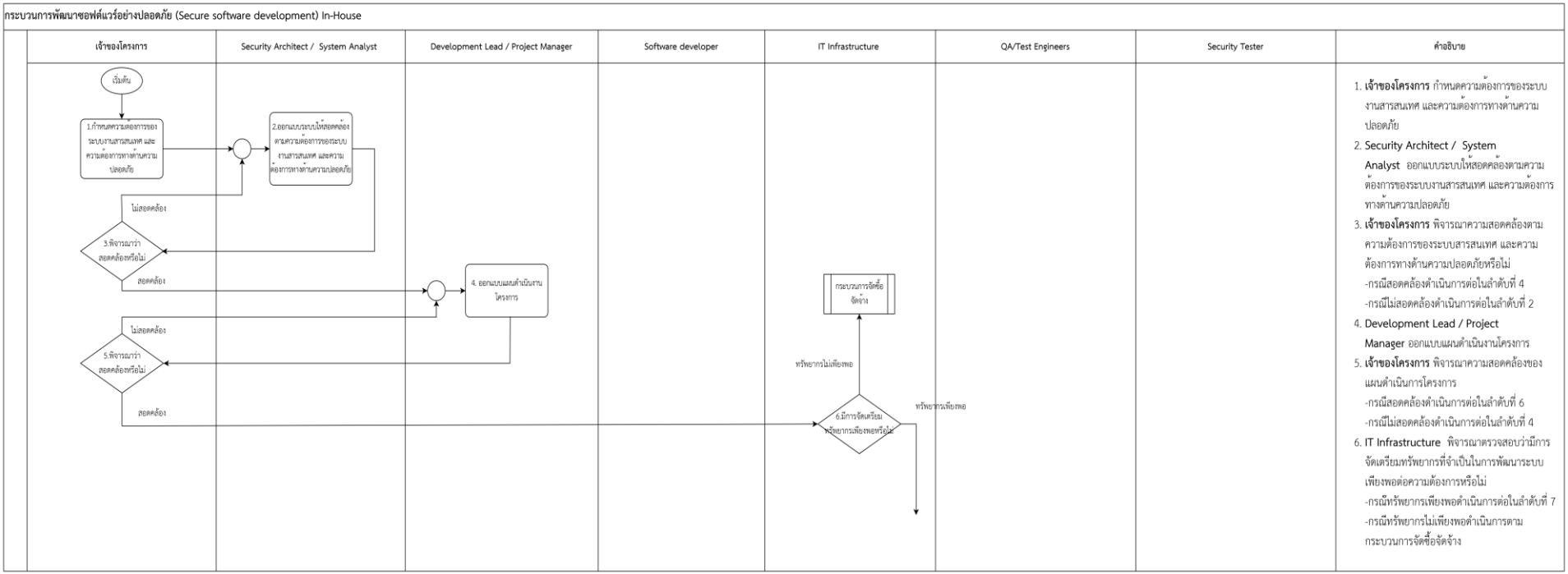


5.1.2 ขั้นตอนของกระบวนการจัดซื้อจัดจ้าง

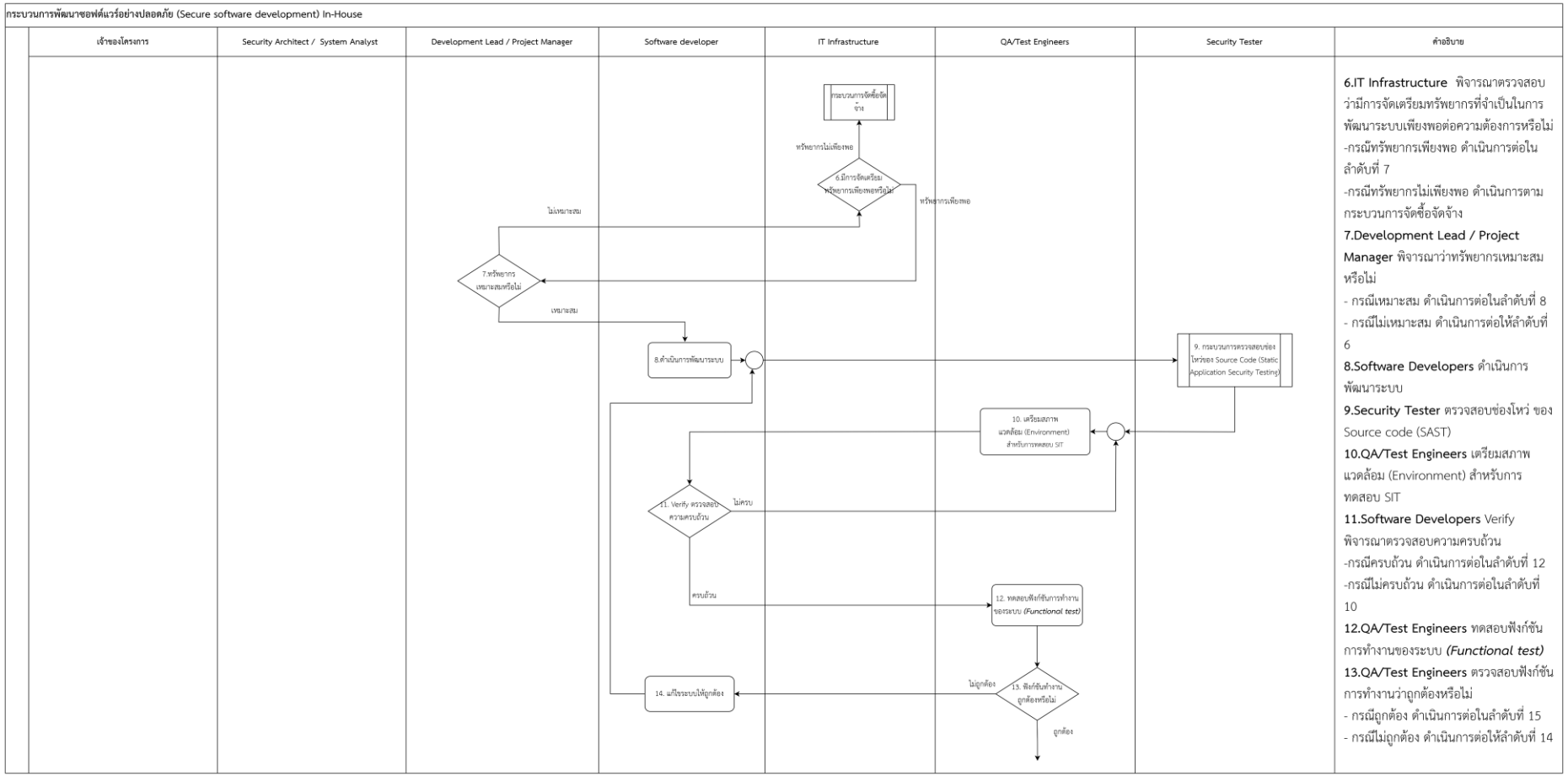


5.2 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development)

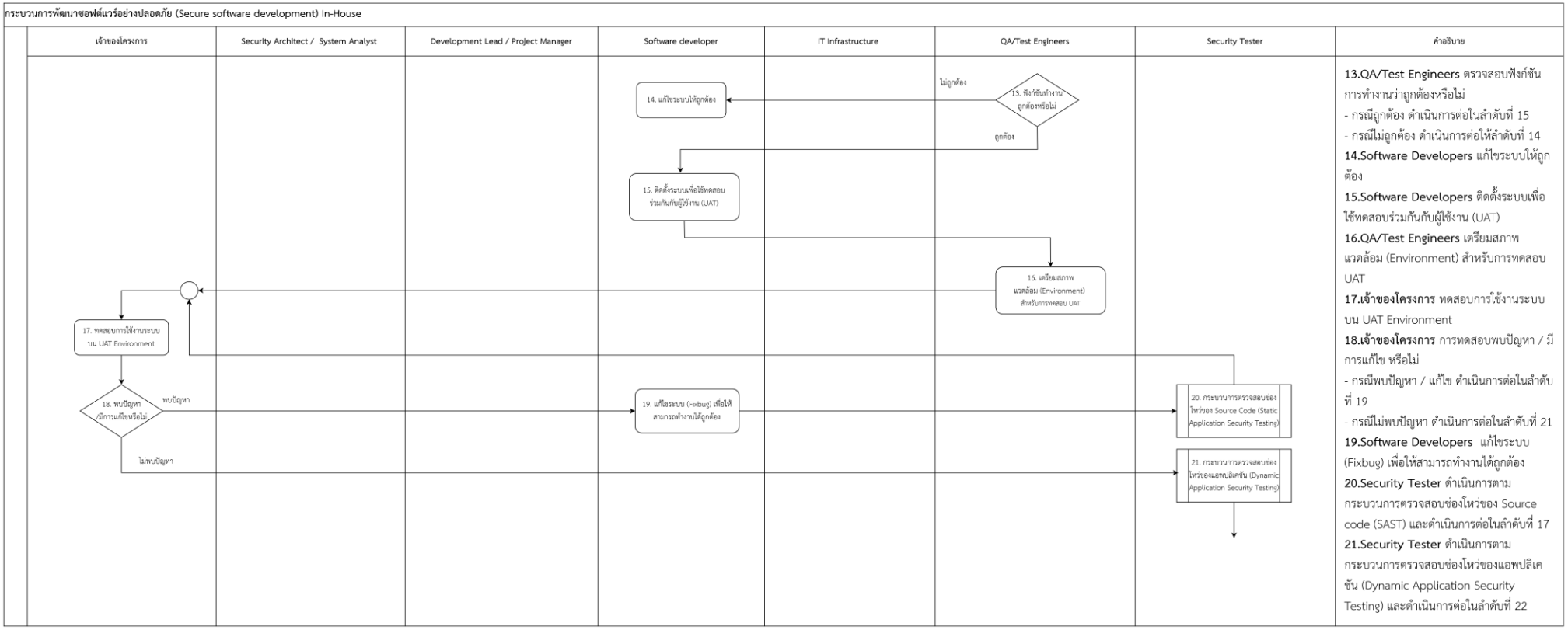
5.2.1 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development) In-House (ส่วนที่ 1)



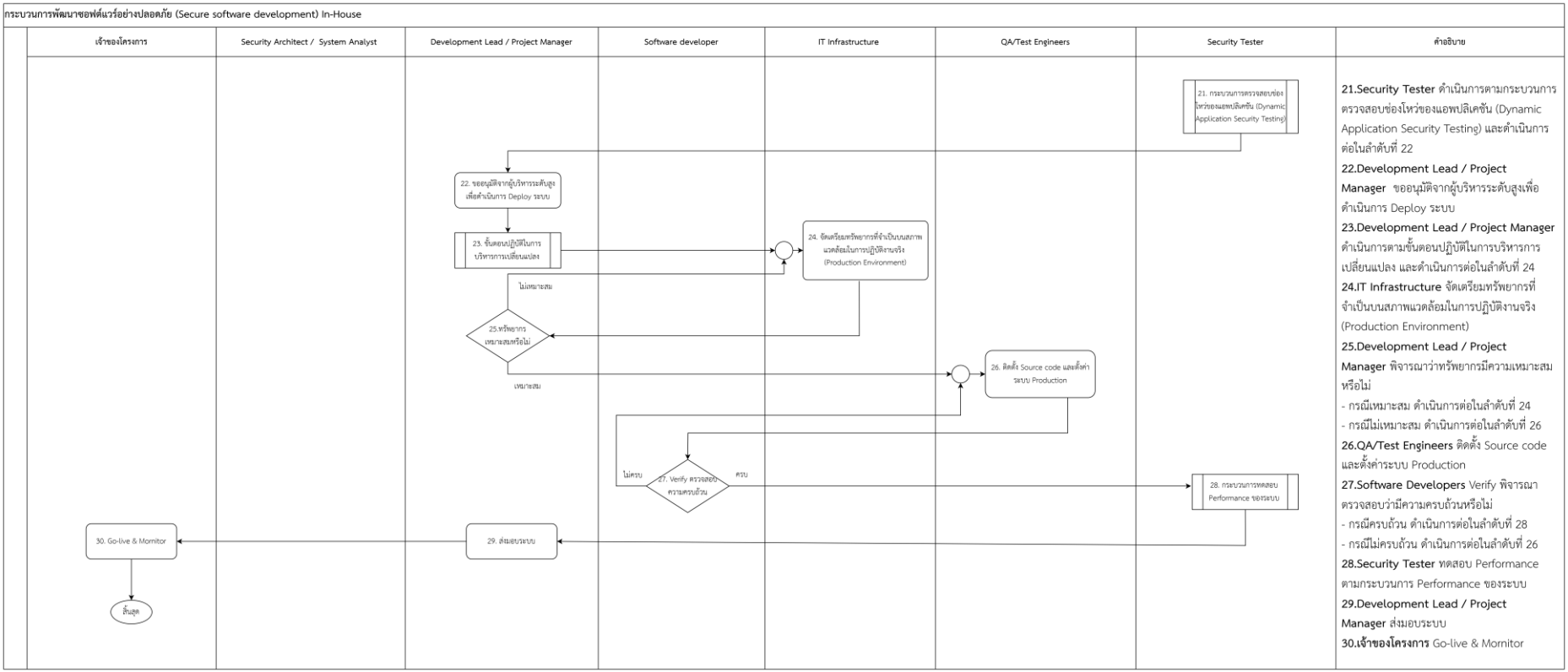
5.2.2 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development) In-House (ส่วนที่ 2)



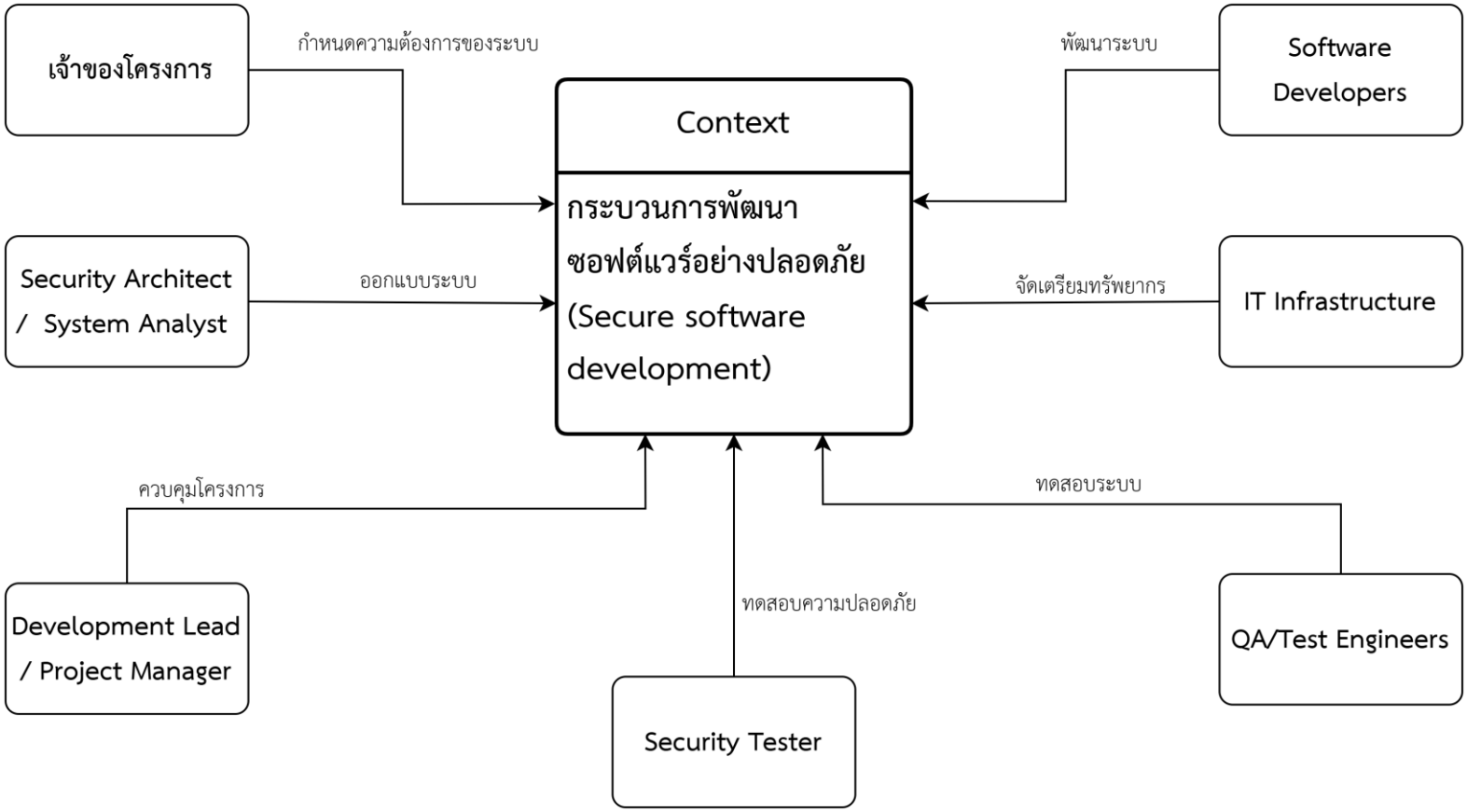
5.2.3 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development) In-House (ส่วนที่ 3)



5.2.4 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development) In-House (ส่วนที่ 4)

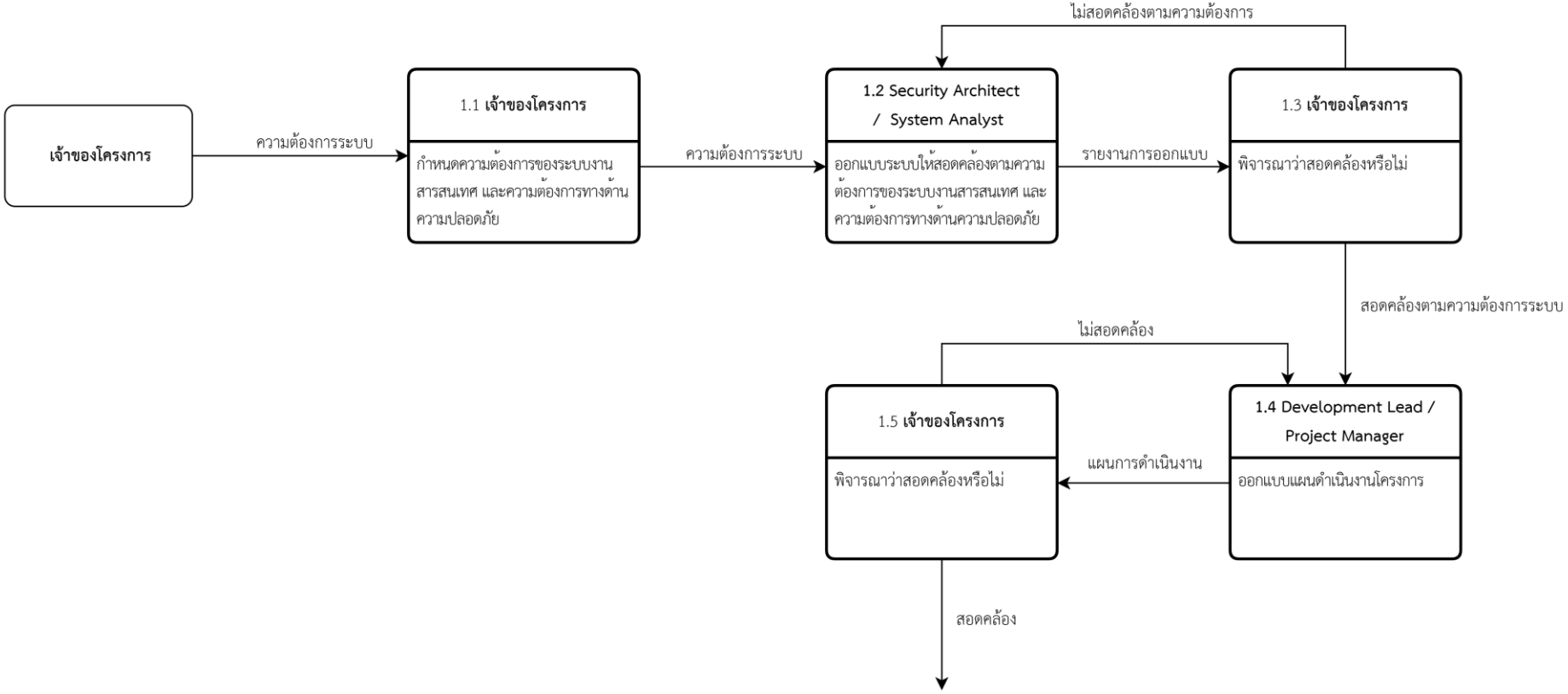


5.2.1 บริบทกระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (In-House)

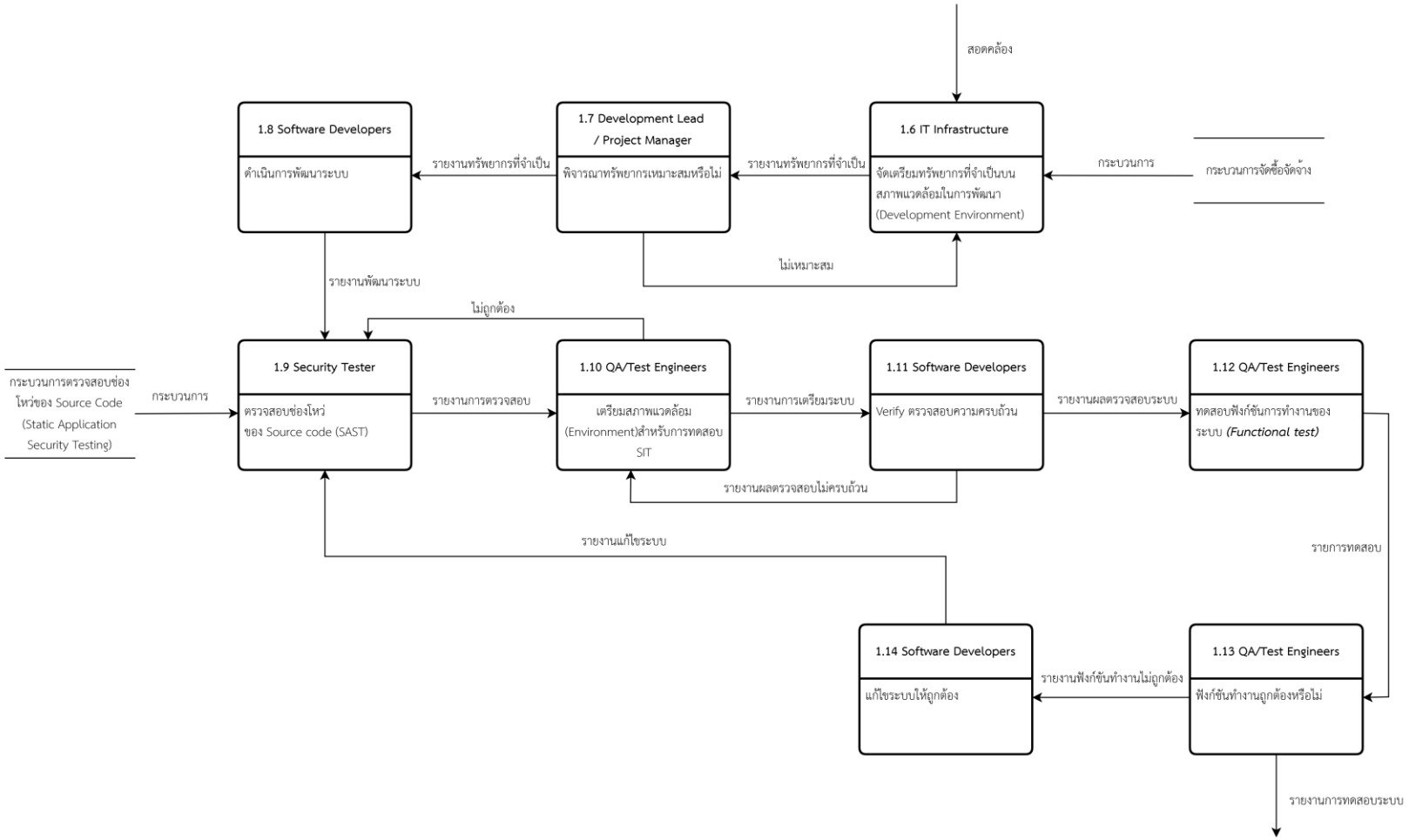


5.2.2 ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (In-House)

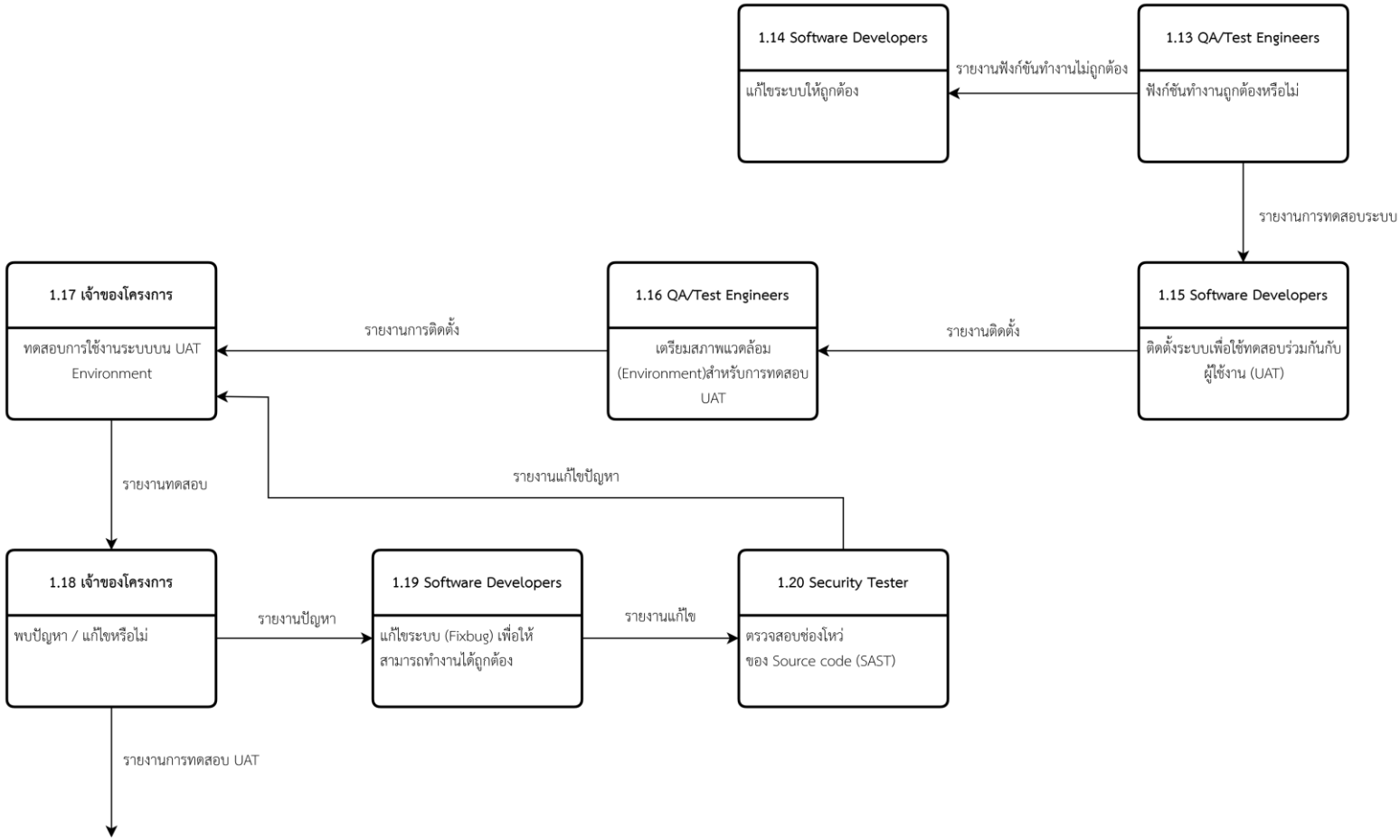
กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย ส่วนที่ 1 (In-House)



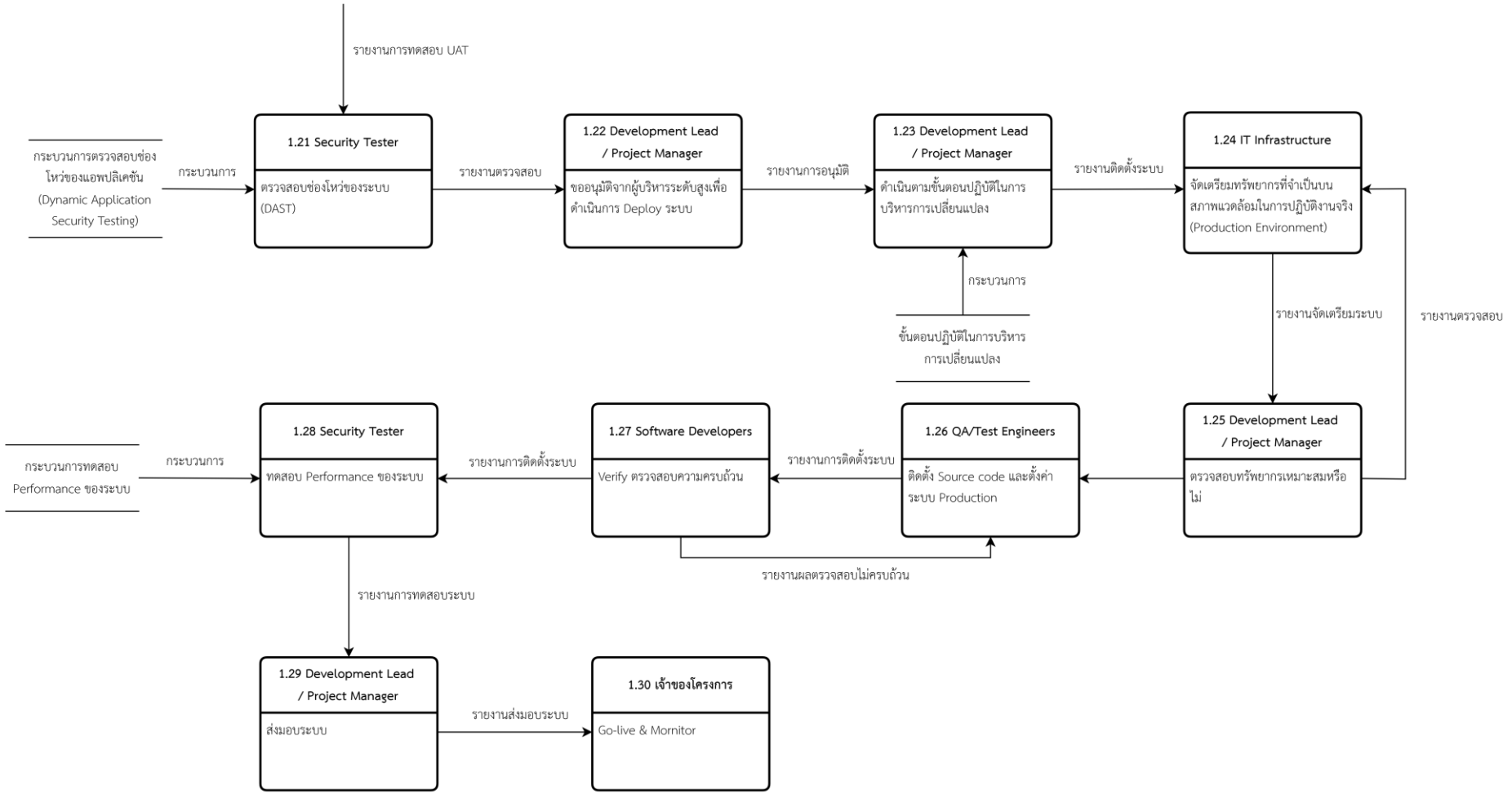
กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย ส่วนที่ 2 (In-House)



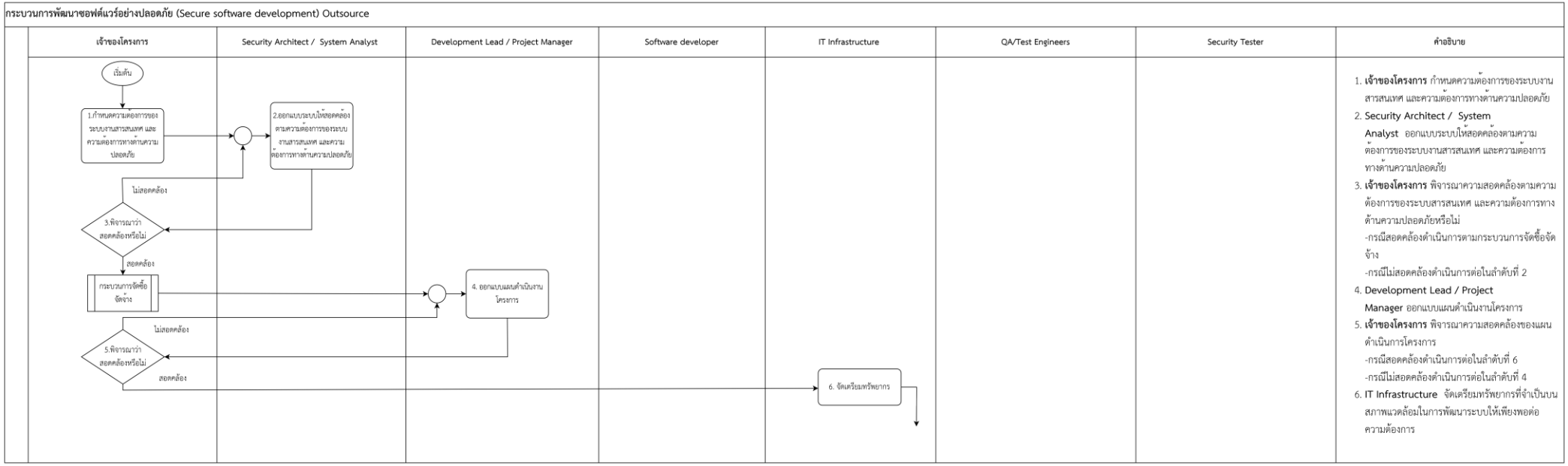
กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย ส่วนที่ 3 (In-House)



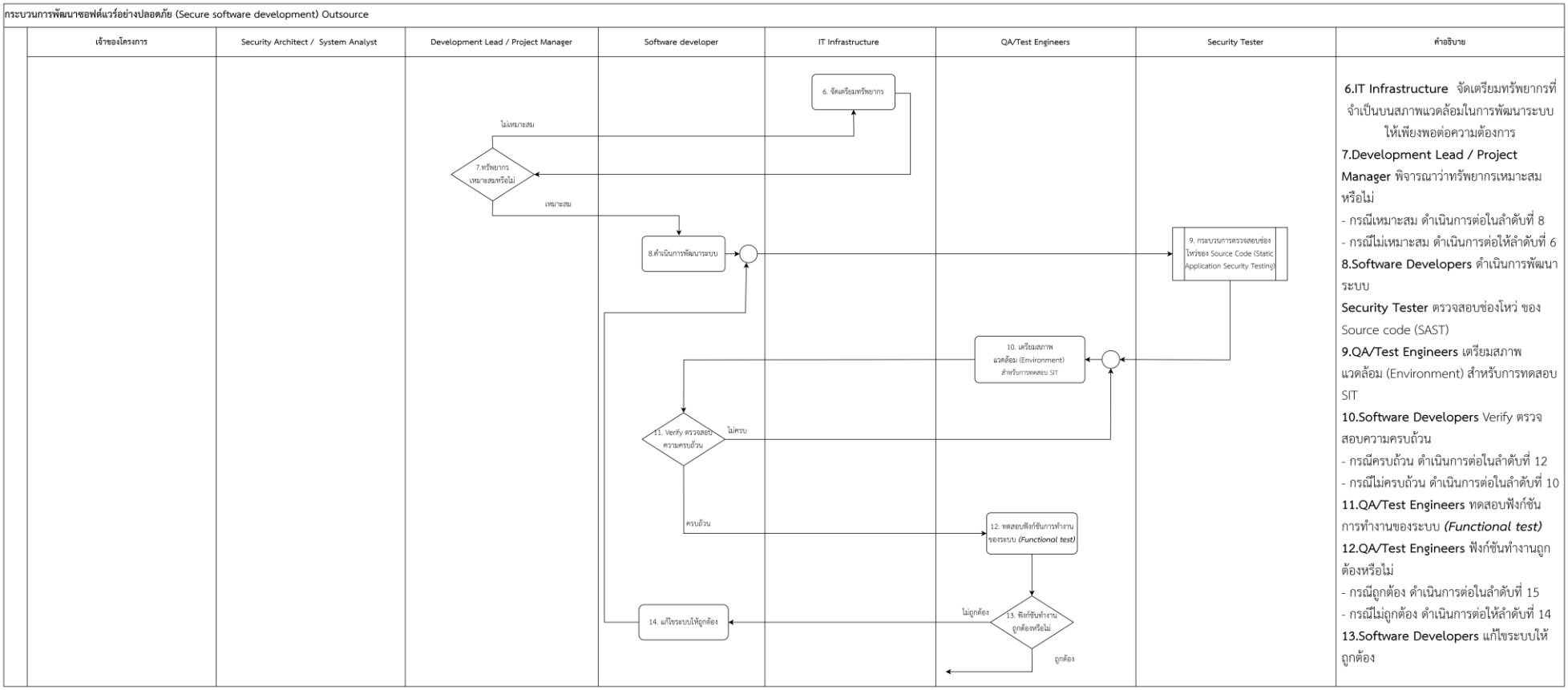
กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย ส่วนที่ 4 (In-House)



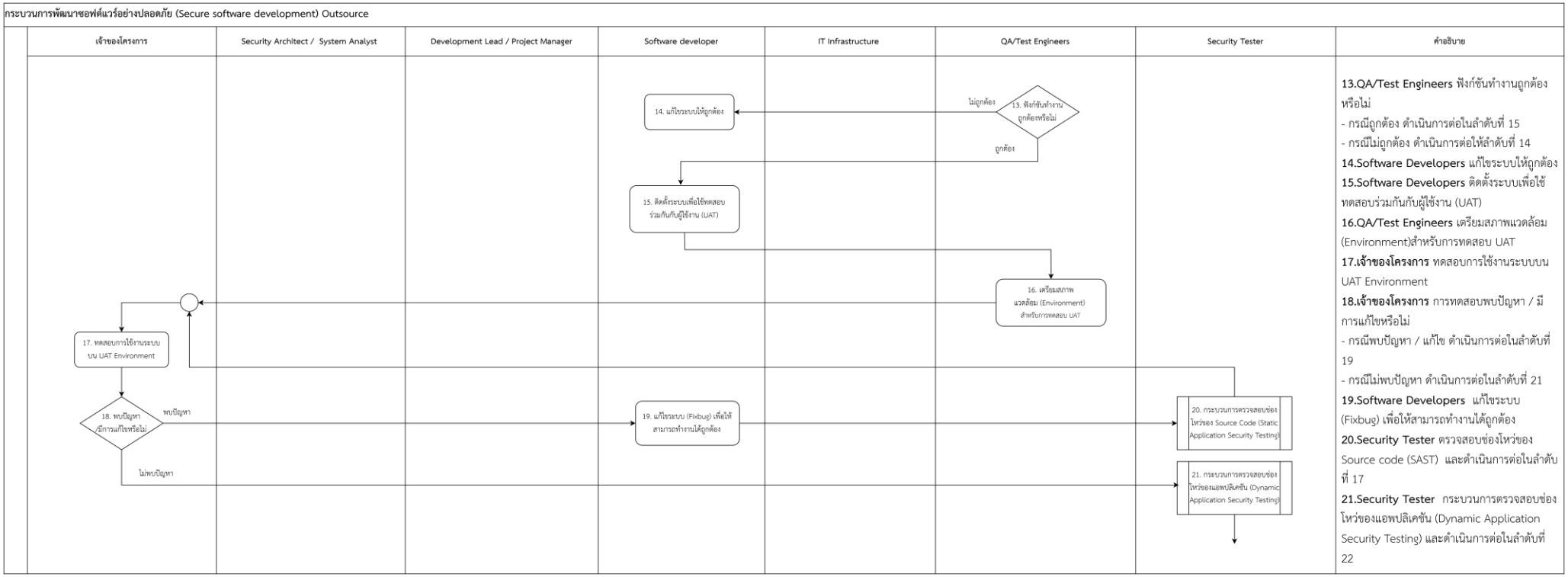
5.2.3 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development) Outsource (ส่วนที่ 1)



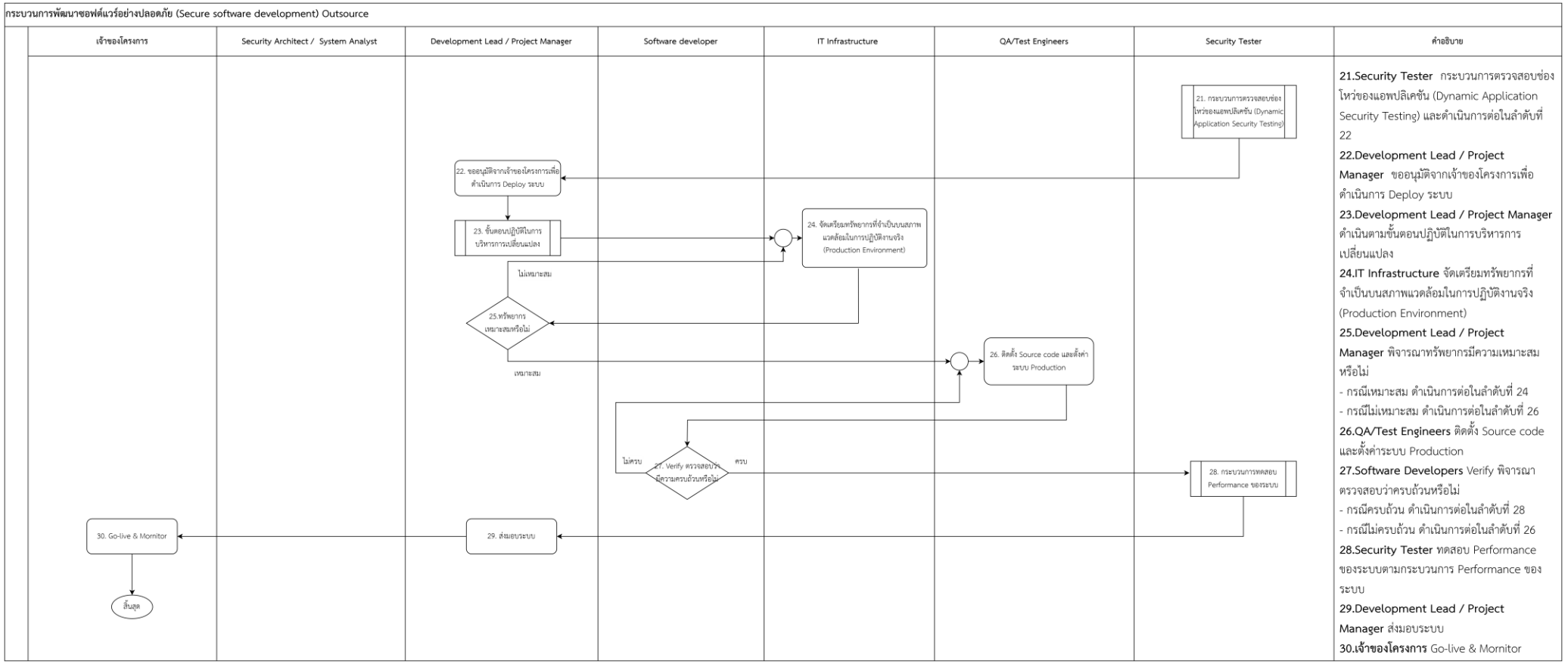
5.2.4 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development) Outsource (ส่วนที่ 2)



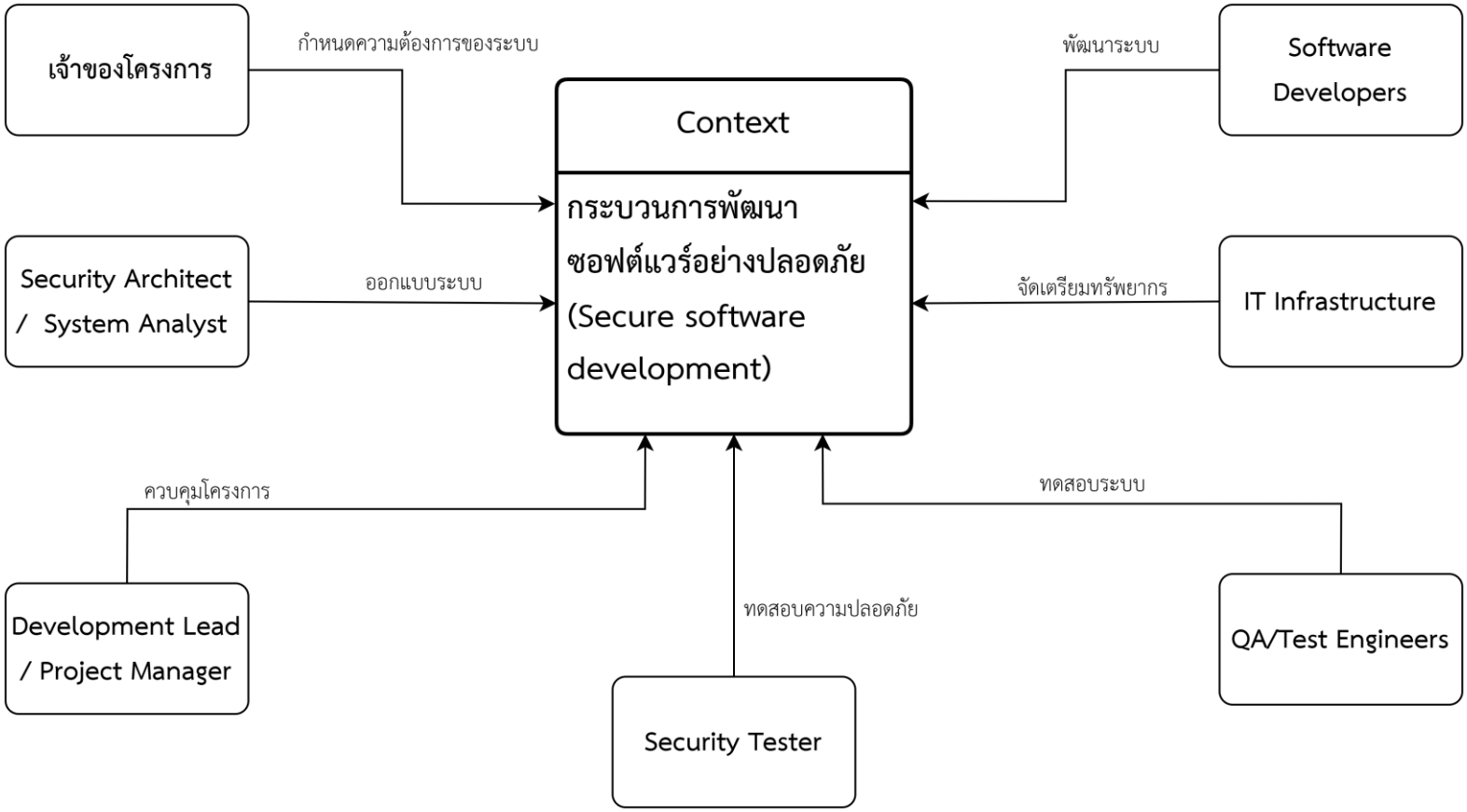
5.2.5 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development) Outsource (ส่วนที่ 3)



5.2.6 กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Secure Software Development) Outsource (ส่วน 4)

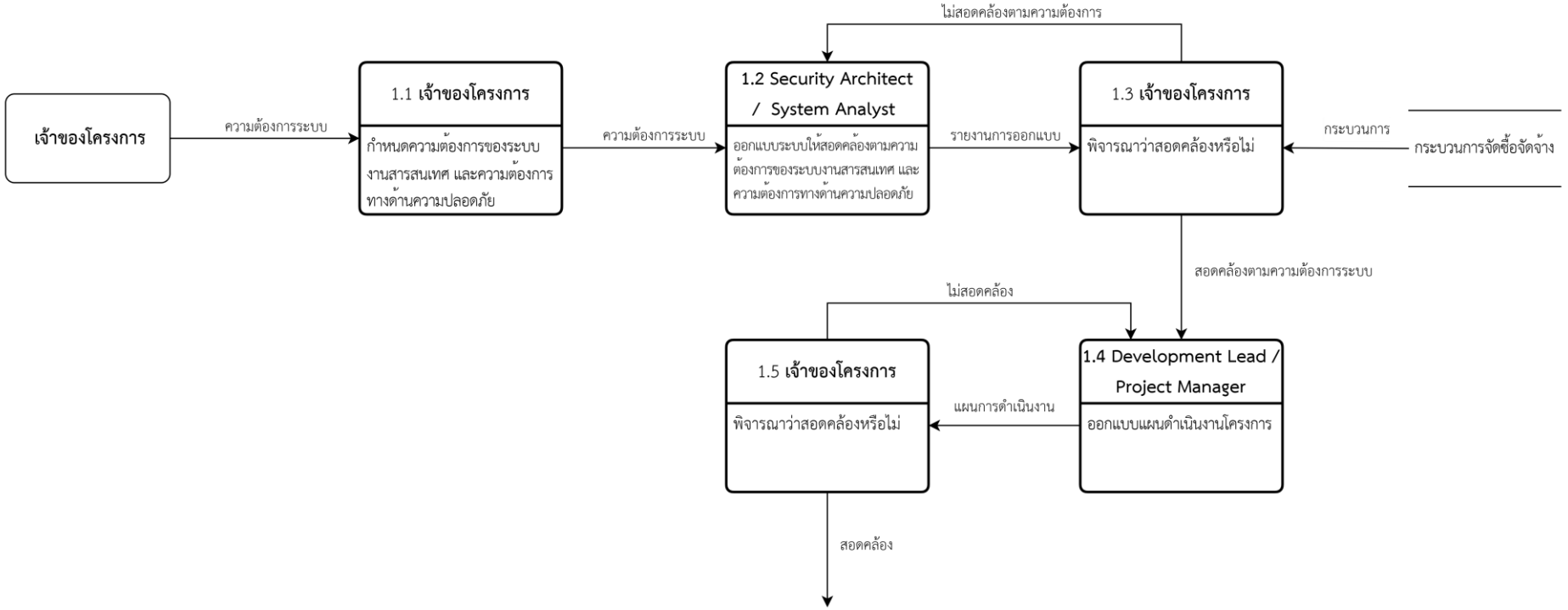


5.2.7 บริบทกระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Outsource)

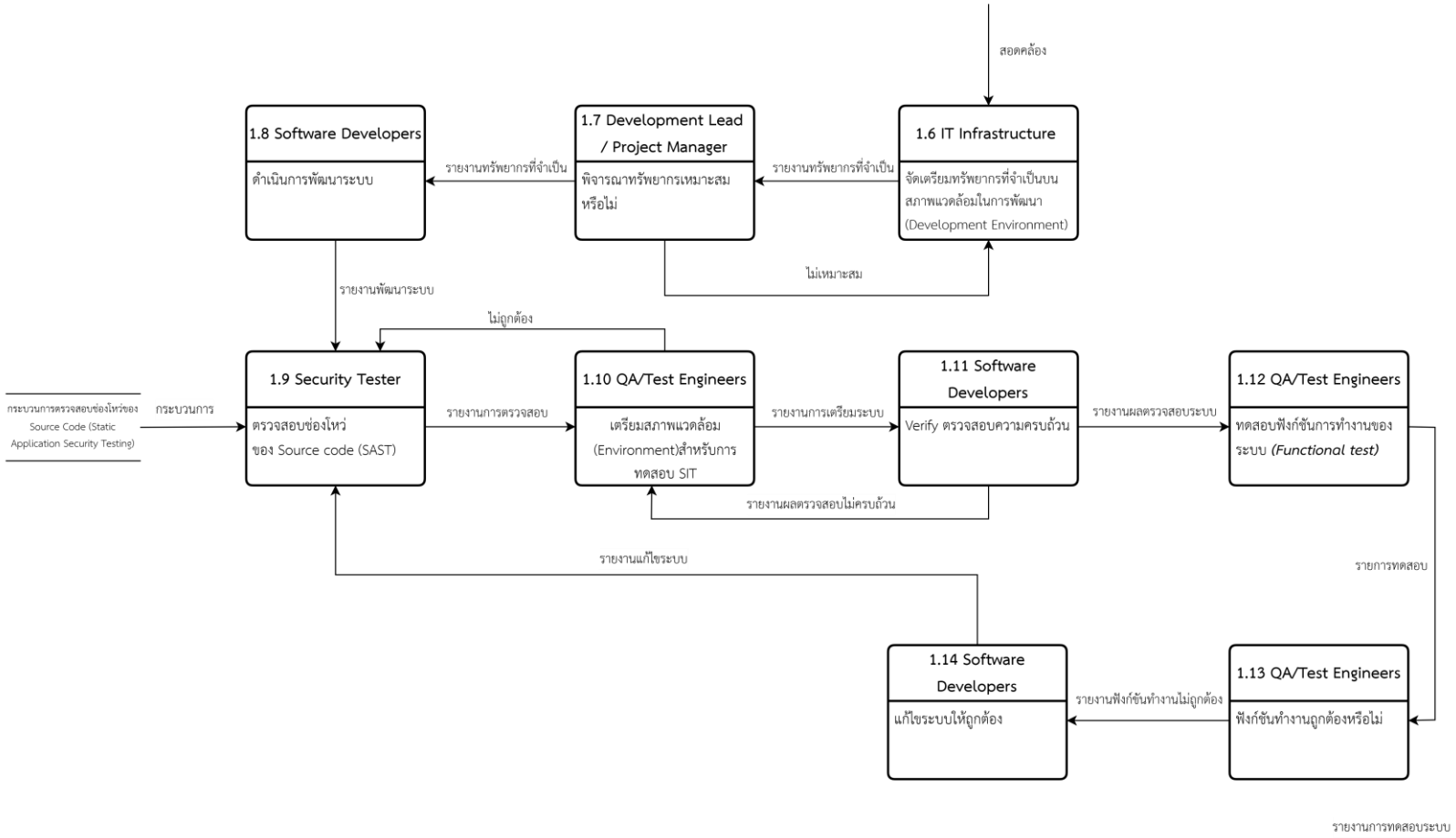


5.2.8 ขั้นตอนของกระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย (Outsource)

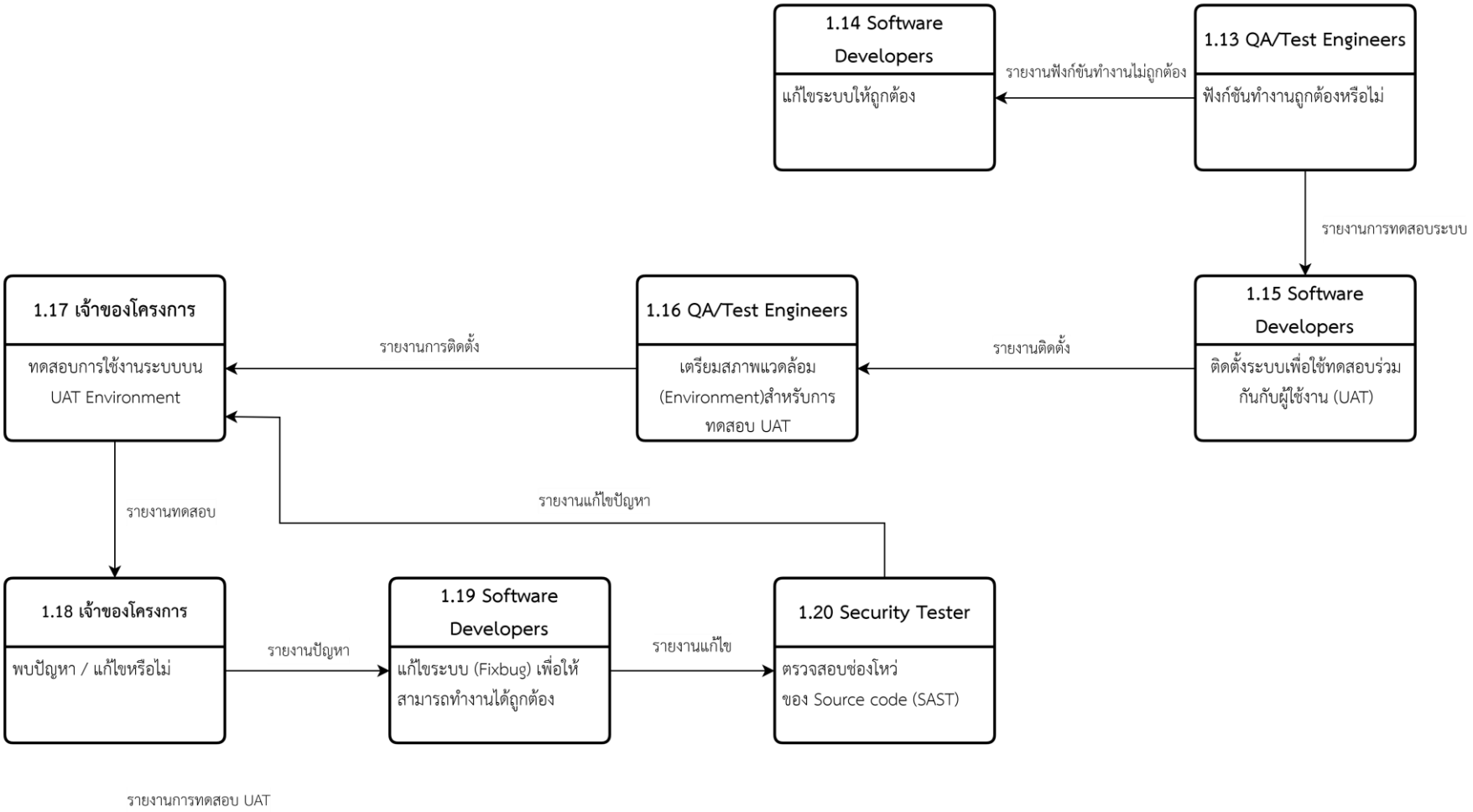
กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย ส่วนที่ 1 (Outsource)



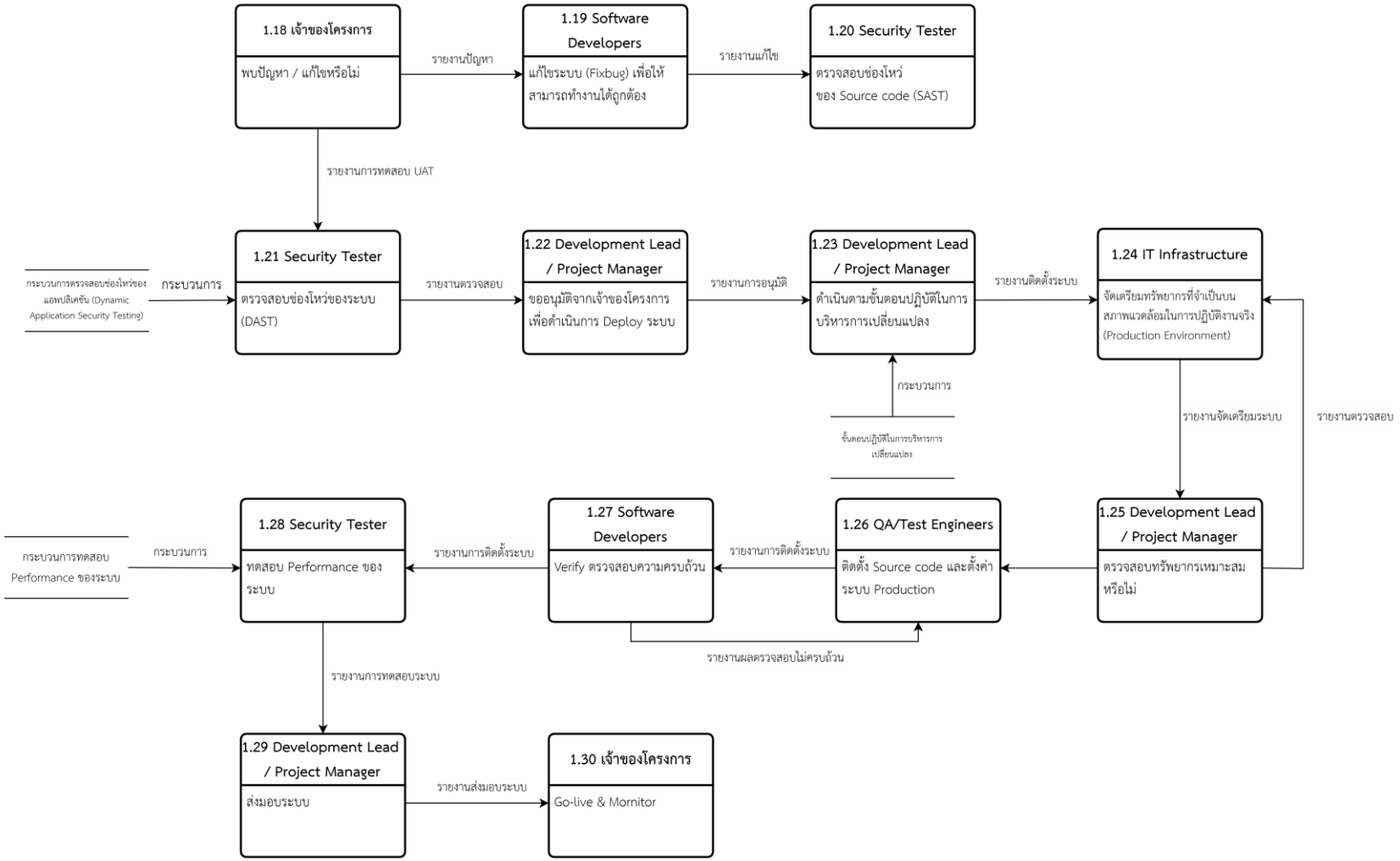
กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย ส่วนที่ 2 (Outsource)



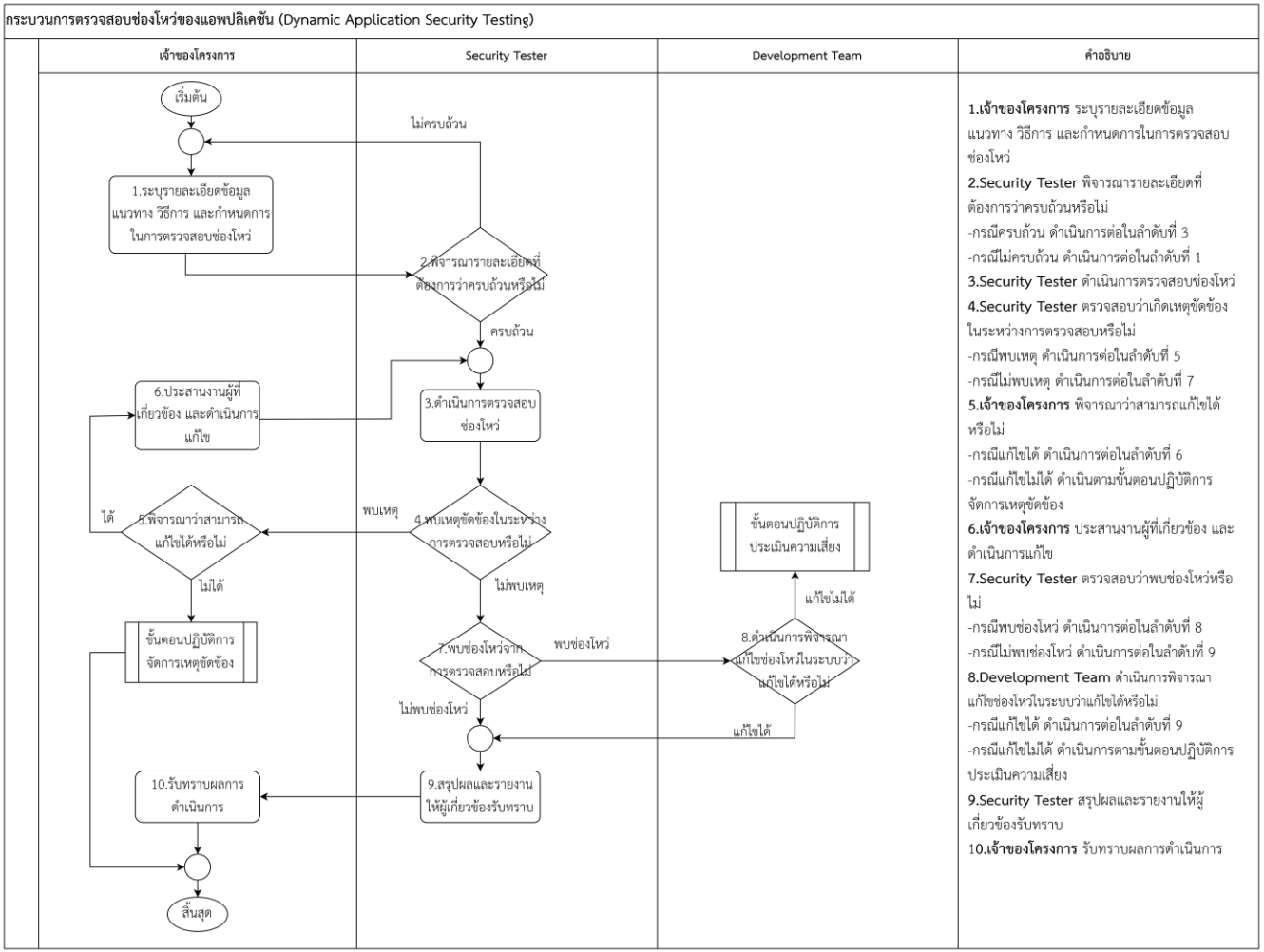
กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย ส่วนที่ 3 (Outsource)



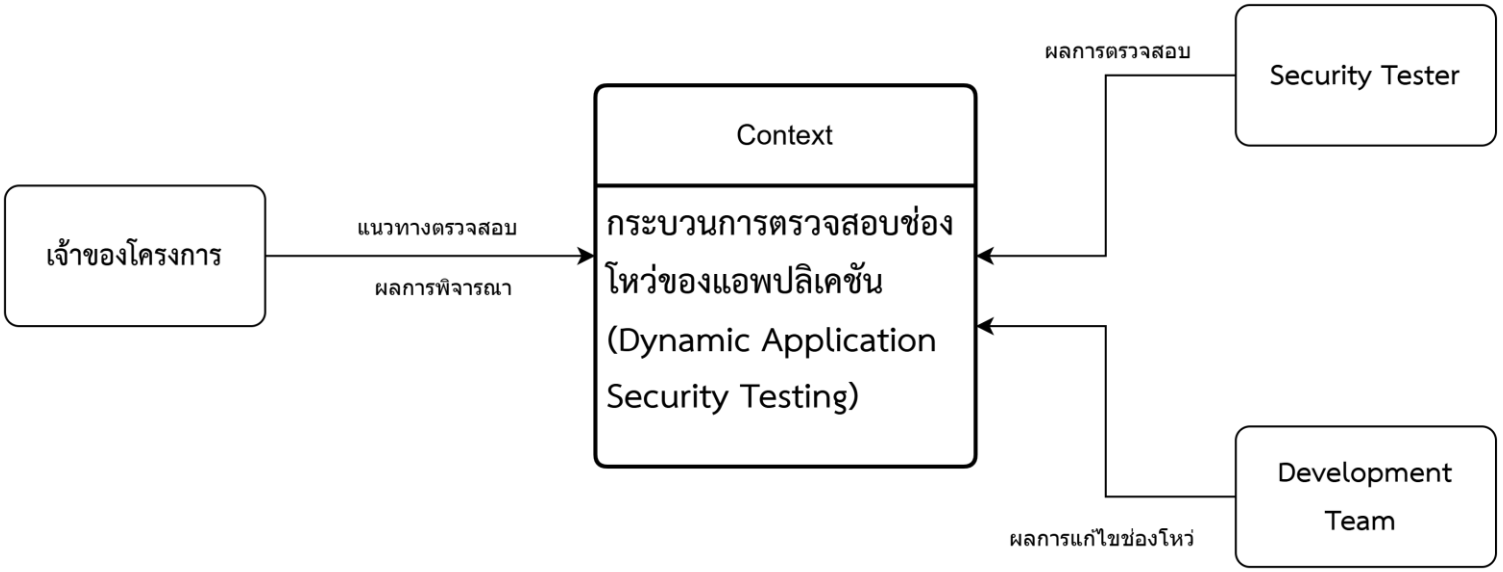
กระบวนการพัฒนาซอฟต์แวร์อย่างปลอดภัย ส่วนที่ 4 (Outsource)



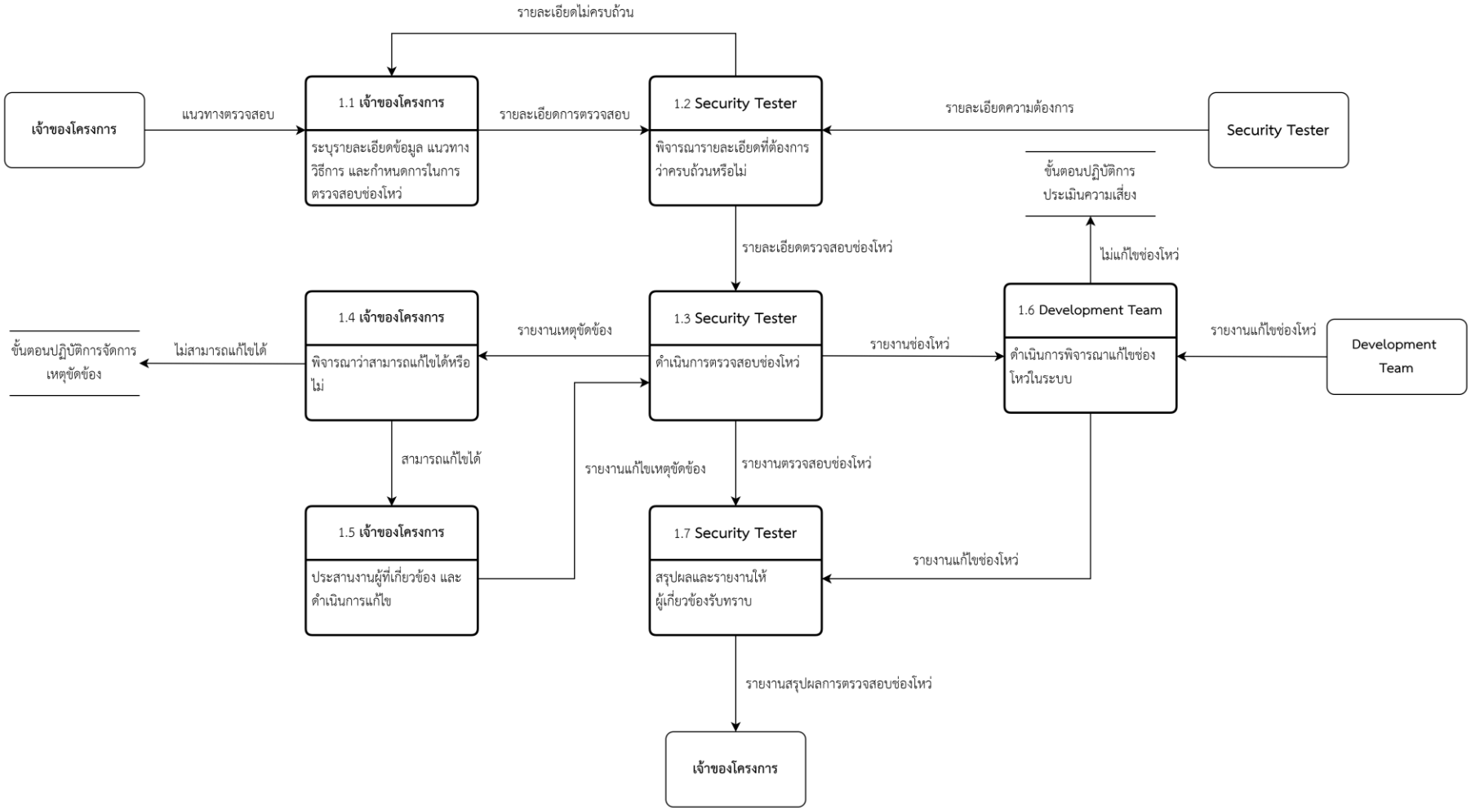
5.3 กระบวนการทดสอบเจาะระบบ (Dynamic Application Security Testing)



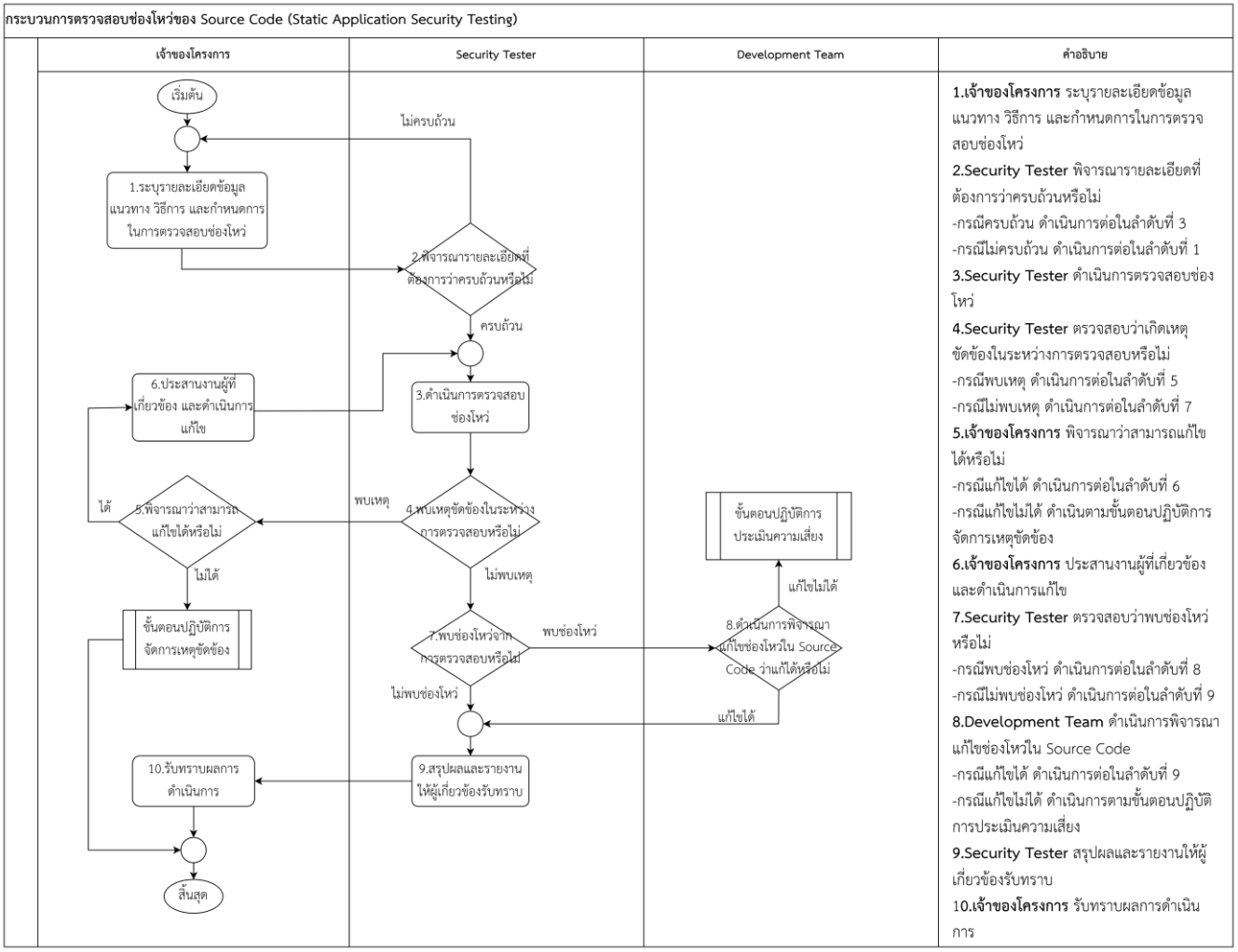
5.3.1 บริบทกระบวนการทดสอบเจาะระบบ (Dynamic Application Security Testing)



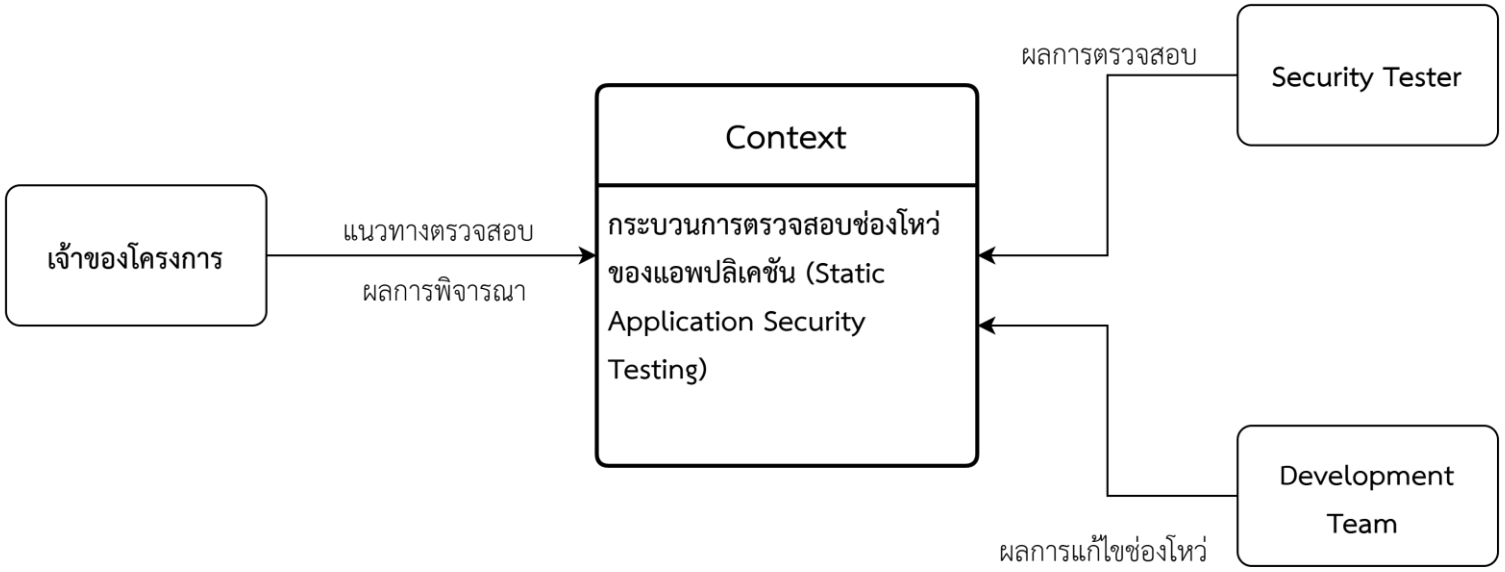
5.3.2 ขั้นตอนของกระบวนการทดสอบเจาะระบบ (Dynamic Application Security Testing)



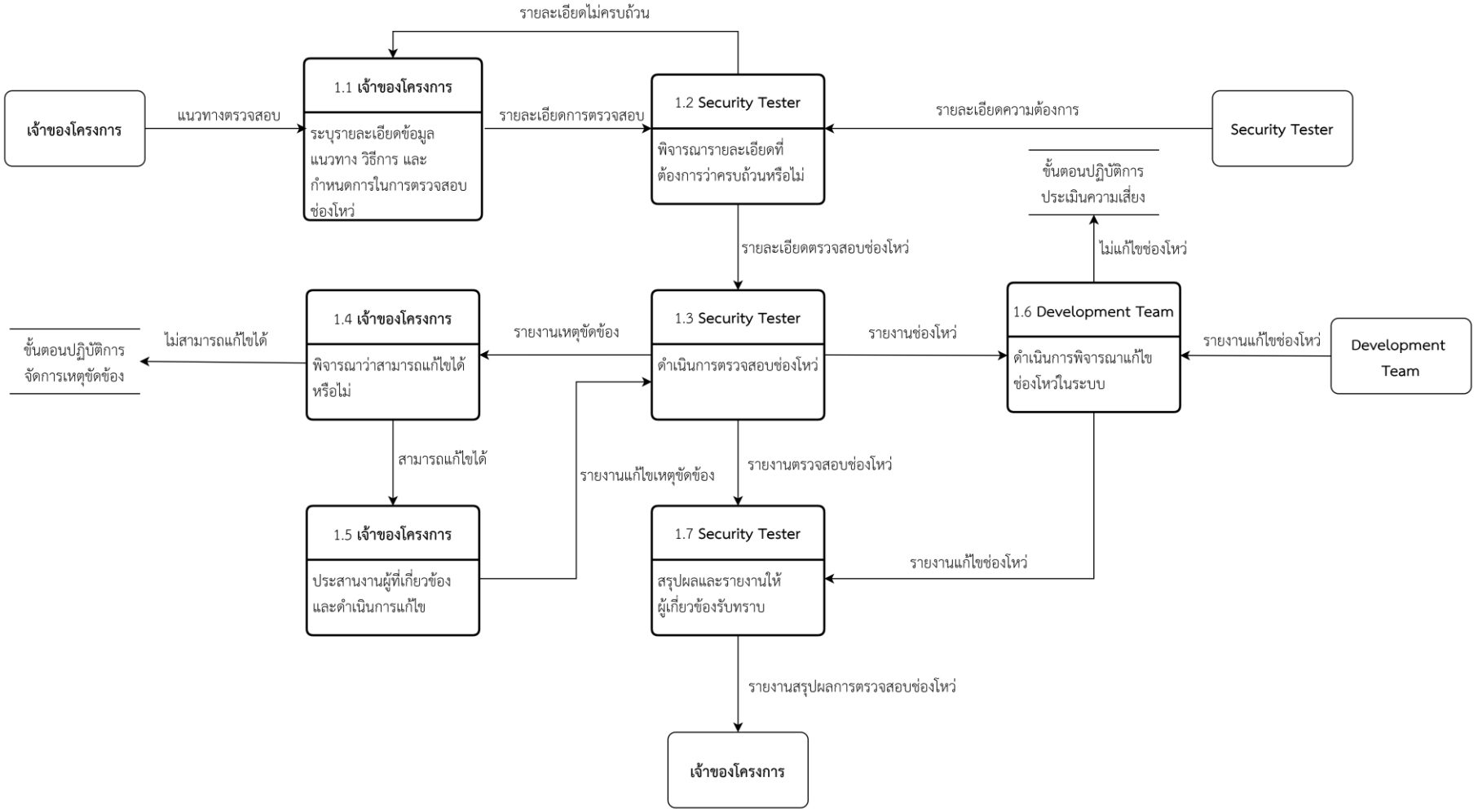
5.4 กระบวนการตรวจสอบช่องโหว่ของ Source Code (Static Application Security Testing)



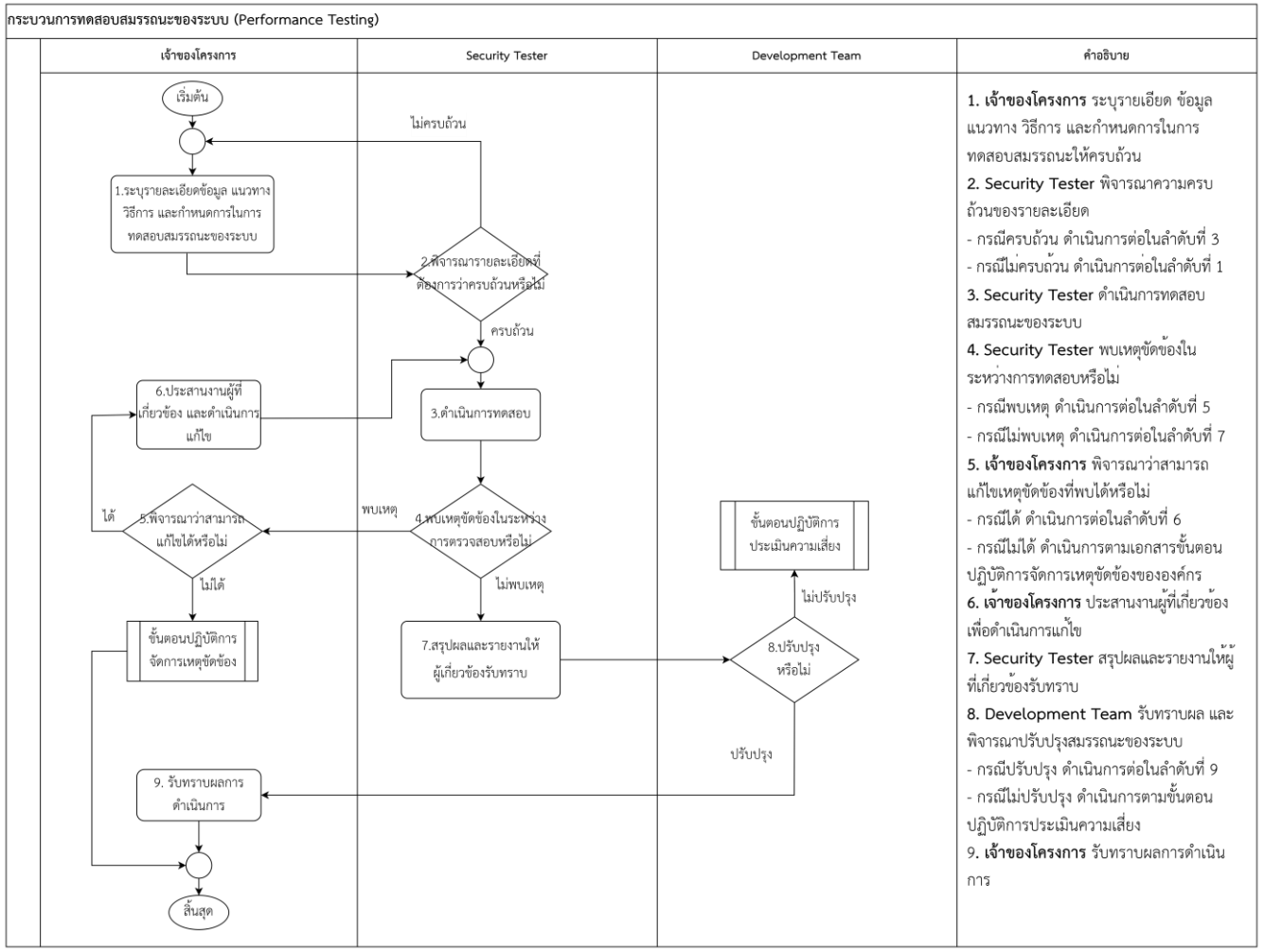
5.4.1 บริบทกระบวนการตรวจสอบช่องโหว่ของ Source Code (Static Application Security Testing)



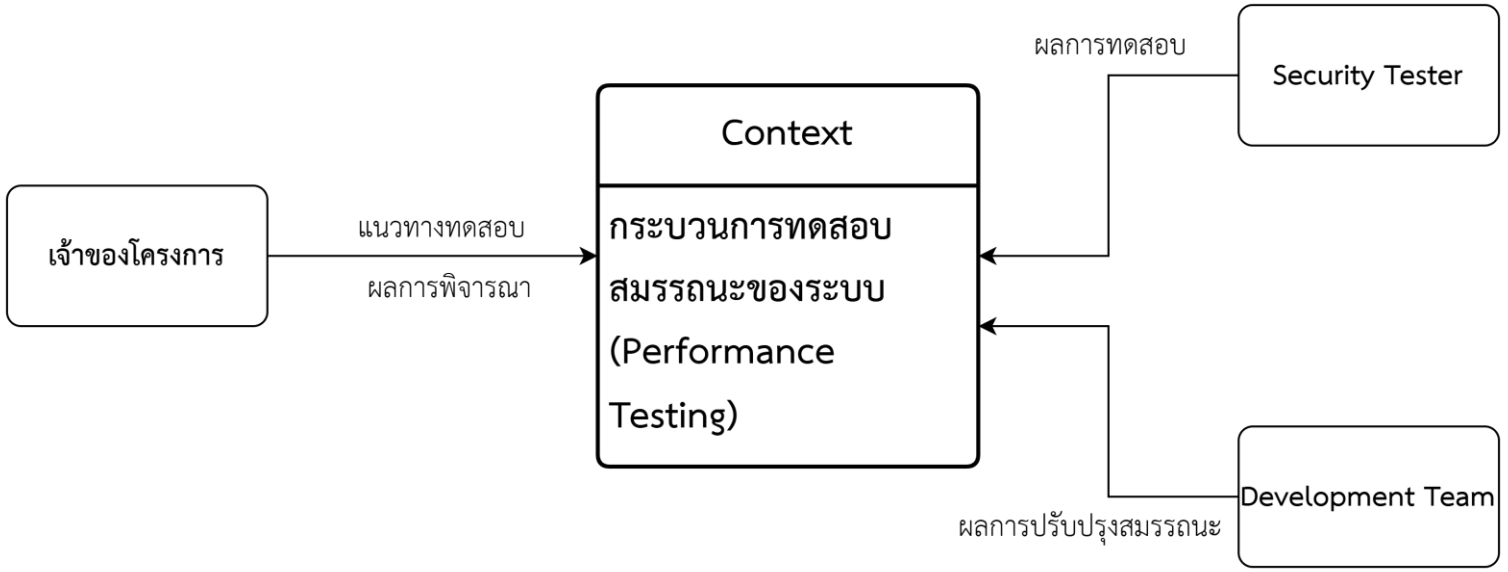
5.4.2 ขั้นตอนของกระบวนการตรวจสอบช่องโหว่ของ Source Code (Static Application Security Testing)



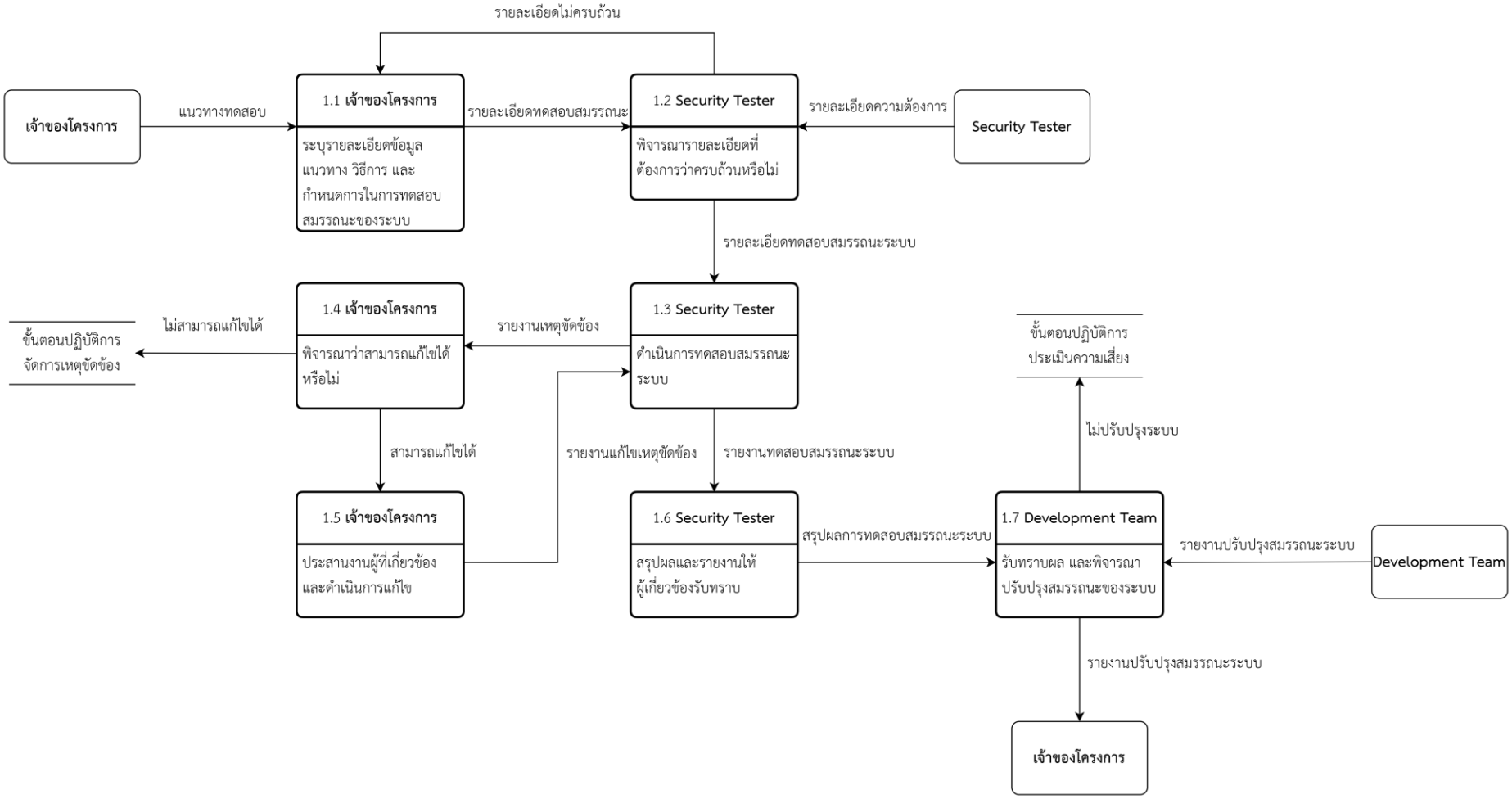
5.5 กระบวนการทดสอบสมรรถนะของระบบ (Performance Testing)



5.5.1 บริบทกระบวนการทดสอบสมรรถนะของระบบ (Performance Testing)



5.5.2 ขั้นตอนของกระบวนการทดสอบสมรรถนะของระบบ (Performance Testing)



6. การทบทวน/แก้ไข (Review)

ทบทวนเพื่อการปรับปรุงปีละ 1 ครั้ง หรือมีการเปลี่ยนแปลงที่เกี่ยวกับด้านสารสนเทศในเรื่องการพัฒนาระบบสารสนเทศ

7. เอกสารอ้างอิง

- NIST Secure Software Development Framework (SSDF)
- ISO/IEC 27001:2022
- ISO/IEC 27701:2019
- Secure development policy