กุญแจสู่ความสำเร็จของการเขียนโปรแกรม

บันทึกสรุปความคิดเห็นและข้อแนะนำที่ได้จากการสัมนาที่ Software Park เมื่อวันที่ 16 มีนาคม 2554

ข้อสังเกต:

1) ข้อมูลสรุปดังกล่าวเหมาะสำหรับคนที่ได้เข้าร่วมสัมนาเท่านั้น เนื่องจากไม่มีการอธิบายเพิ่มเติมในแต่ละส่วนแต่อย่างใด
2) ข้อสรุปดังกล่าวเป็นการรวมความเห็นต่างๆเข้าด้วยกันเท่านั้น เทคนิคหรือวิธีการใดควรถูกนำไปใช้นั้นขึ้นอยู่กับสถานะการณ์และสิ่งแวดล้อมขององค์กรนั้นๆ

1. การทำตาม design ที่เกิดขึ้น ใช้ diagram อะไรในการออกแบบที่ใกล้ code ขนาดนี้?
    a. *ใช้ฟังก์ชั่น list เท่านั้น (อาจจะอยู่ในฟอร์มของ CRC)*
    b. *ใช้ free-form diagram*
    c. *ใช้ UI sketch*
2. การเขียนแบบ object-oriented จริงๆ
    a. *ใช้ O-O ประมาณ 10% และเหตุผลหลักคือเพื่อการ reuse*
    b. *ทำโดยใช้ Top-Down approach บวกกับการ Mapping ระหว่าง O-O กับ database ซึ่งเป็น relational database*
3. การเขียนโปรแกรมโดยคำนึงถึง non-functional requirements ด้วย มั่นใจอย่างไรว่า non-functional requirements ถูกพิจารณาจริงๆ?
    a. *ไม่สามารถทราบได้ว่าพิจารณาหรือไม่*
    b. *ส่วนใหญ่จะหวังพึ่งประสบการณ์ของผู้พัฒนาว่าโดย sense แล้วน่าจะพิจารณา*
4. การเขียนโปรแกรมที่นำกลับมาใช้ใหม่ได้ (และ bug น้อยลงจากการปรับปรุง code ให้ดีขึ้นเรื่อยๆ)
    a. *บางองค์กรมีทีมทำโดยเฉพาะ*
    b. *ตัดสินใจว่าจะทำเป็น reusable component/module หรือไม่ จากการที่มีการเรียกใช้บ่อยแค่ไหน*
    c. *บางองค์กรมีการจัดทำ utility class และมีการบริหารจัดการ class นี้โดยเฉพาะ*
5. การเขียน comment ใน code
    a. *บางองค์กรไม่สนับสนุนให้เขียน comment เลยเพราะเกรงว่าจะไม่มีการ update comment เหล่านั้น แต่ใช้การเขียน code ที่ใช้ชื่อตัวแปร ชื่อ class หรือ ชื่อ method ที่เข้าใจง่ายแทน*
    b. *บางองค์กรก็มีการทำมาตรฐานในการเขียน comment ใช่ในองค์กร*
    c. *มีการใช้ program ที่ชื่อ checkStyle ในการตรวจสอบวิธีการเขียนโปรแกรมของโปแกรมเมอร์*

ถ้ามีการเปลี่ยนแปลงเกิดขึ้น!!

1. การทำ versioning control ทำอย่างไร?
    a. *ใช้ tool ในการทำ แต่ต้องระวังเรื่อง commit และ merge*
    b. *สิ่งที่น่าสนใจคือ มีทั้งความเห็นว่าควร commit <u>บ่อยๆ</u> และให้ commit แค่<u>วันละครั้ง</u> แต่ทั้ง 2 วิธีก็เหมาะกับ process และวิธีการทำงานที่ต่างกัน*
2. ถ้ามีการทำ Change management และการใช้ Change request form จริงๆ
    a. *ใช้ change management tool และ form โดยทั่วไป*
    b. *ใช้ Continuous Integration Tool สำหรับการที่รองรับการเปลี่ยนแปลงที่เกิดขึ้นตลอดเวลาไม่หยุด (ในกรณีนี้ treat การเปลี่ยนแปลงว่าเป็น norm)*
    c. *ใช้ Maven Project ในการทำ configuration ของโครงการ*