

USER STORY & 'INVEST'

การเขียน USER STORY ที่ดีด้วยหลัก INVEST

โดย

ปิโยรส ปิยจันทร์

<https://medium.com/@piyorot>

หนังสือเล่มนี้เป็นการรวบรวมบทความจากบล็อกส่วนตัวของผมที่เขียนไว้ใน <https://medium.com/@piyorot> ถ้าสนใจก็ติดตามผลงานกันได้ครับ เขียนทุกวันครับ



piyorot

ผมสนใจเรื่อง Project Management and Software Development process ครับ ก็เลย
เขียนบล็อกนี้ครับ <http://t.co/h2JWWI562d>

3 FOLLOWING



23 FOLLOWERS



เรื่องราวสมมติข้างล่างเป็นบทสนทนาระหว่าง

ศิริพงษ์ – Project Manager ผู้มีใจรักใน Agile Development และมีความเชี่ยวชาญพอสมควรในเรื่องของการเขียน User Story ... กับ

เอกภพ – Product Owner มือใหม่ที่ไม่เข้าใจว่า User Story คืออะไรและมีหลักการในการเขียนอย่างไร

ศิริพงษ์จึงเปิดฉากอธิบายเพื่อตอบข้อสงสัยของเอกภพ

เริ่มต้นด้วย User Story

Starting with User Story

ศิริพงษ์: “วันนี้เป็นวันแรกที่ทีมจะเริ่มใช้ Kanban อย่างเป็นทางการ คนในทีมพร้อมเต็มที่ ชักซ้อมความเข้าใจเรื่องกระบวนการทำงานกันพอสมควร และ Kanban Board ก็เตรียมไว้แล้ว แต่งานยังไม่มี ☹️” แล้วเดินไปหาเอกภพ Product Owner ของ CMS เพื่อคุยเรื่องนี้แต่เช้า

ศิริพงษ์: “เออครับ ทีมจะเริ่มทำงานวันนี้แล้วแต่ยังไม่มี Requirement ให้พี่เลย”

เอกภพ: “อ้อครับพี่ ทีมอยากได้ Requirement แบบไหนอะครับ?”

ศิริพงษ์: “อืม จากช่วงที่เราใช้ Scrum ทีมเราเคยหยิบ Requirement Document ของเอามาย่อเป็นงานเล็กๆด้วยเทคนิค User Story นะ ... เออเคยได้ยินมั๊ย? พี่คิดว่าทีมน่าจะโอเคกับ User Story”

เอกภพ: “อ้อครับ เคยได้ยินแต่ไม่รู้วลีๆแล้วมันคืออะไรครับ”

ศิริพงษ์: “โอเคครับ เดี่ยวพี่อธิบายให้ฟังคร่าวๆ คำถามแรกก่อนนะ ...” ศิริพงษ์เคยศึกษาเรื่องนี้จากเว็บไซต์ของปรมาจารย์ด้าน Agile Development ที่ชื่อ Mike Cohn แล้วจึงเล่าเรื่อง User Story ให้เอกภพฟังด้วยความมั่นใจ

“User Story คืออะไร?”

User Story คือคำอธิบายถึง Feature ที่สั้นและง่ายโดยผ่านมุมมองของคนที่ต้องการใช้งาน Feature นี้ซึ่งส่วนใหญ่ก็จะเป็น User หรือ Customer ของระบบ โดยมากแล้ว User Story จะถูกเขียนอยู่ในรูปแบบง่ายๆแบบนี้

As a <type of user>, I want to <some goal>, so that <some reason>.

ส่วนใหญ่ User Story จะถูกเขียนลงบนกระดานหรือการ์ดเล็กๆแล้วก็เอามาแปะบนผนังหรือบอร์ดอย่าง Kanban Board เพื่อใช้ในการทำ Planning หรือเวลาที่เรามา Requirement Discussion ครับ”

As a <type of user>,
I want to <some goal>,
so that <some reason>.

User Story Format

เอกภพ: “แล้วกระดาดใบเล็กๆมันจะพอหรือครับ ถ้าข้อมูลมันเยอะ”

ศิริพงษ์: “นี่แหละข้อดีของ User Story ครับ เพราะว่าคนคิดเค้าตั้งใจให้เนื้อที่มันไม่พอเพื่อกระตุ้นให้คนในทีมได้พูดคุยกัน สิ่งสำคัญในการทำงานคือการมีปฏิสัมพันธ์ระหว่างสมาชิกในทีมมากกว่าไปยึดติดอะไรที่เขียนในเอกสารนะครับ เหมือนเจตนารมณ์ข้อแรกของ Agile Development”

เอกภพ: “อ้อครับ ฟังดูก็พอเข้าใจแต่ยังไม่เห็นภาพชัดเจนอะครับ พี่พงษ์ช่วยยกตัวอย่างจริงๆของ User Story ให้ดูหน่อยได้มั้ยครับ?”

ศิริพงษ์: “ได้ครับ ...” ศิริพงษ์มีความสุขมากที่ได้โม้เรื่องของตัวเองรู้ ☺

เขียน User Story ให้เป็นอิสระต่อกัน

Writing an Independent User Story

ศิริพงษ์: “เอาเป็น Amazon.com แล้วกันนะ ถ้าพี่จะเขียน User Story ก็คงจะประมาณนี้” เค้ายกตัวอย่างมาซักรี่ห้าข้อ

- As a user, I want to search books, so that I can pick what I like.
- As a user, I want to read book preview and sample, so that I can check if this book fits my purpose.
- As a user, I want to read book reviews, so that I can decide to buy the good ones.
- As a user, I want to see my shopping cart, so that I know how much I have to pay.
- As a user, I want to check out, so that I can complete my purchase.

เอกภพ: “ครับ เห็นภาพมากขึ้นเยอะครับ แล้วถ้าผมจะเขียน User Story บ้าง ผมควรจะเขียนยังไงจะครับ?”

ศิริพงษ์: “มันก็มีคำแนะนำจาก Bill Wake ว่า User Story ที่ดีควรมีหน้าตาแบบนี้ครับ”

“INVEST” Rule for Good Stories

- **I**ndependent
- **N**egotiable
- **V**aluable (to users or customers)
- **E**stimatable
- **S**mall
- **T**estable

Bill Wake, author of Extreme Programming Explained and Refactoring Workbook

INVEST by Bill Wake

ศิริพงษ์: “เพื่ออธิบายที่ละข้อนะ เริ่มจาก Independent ก่อน

A Good User Story is Independent.

User Story ที่ดีต้องเป็นอิสระต่อกัน เพราะถ้า User Story หลายอันมีความสัมพันธ์โยงกันมันจะสร้างปัญหาต่อจัดลำดับความสำคัญ (Prioritization) ครับ เช่น ถ้ามี User Story อยู่สองอันซึ่งต้องทำอันแรกก่อนถึงจะทำอันที่สองได้ แล้วน้องบอกว่าอันที่สองสำคัญกว่ามันก็ไม่ได้เพราะสุดท้ายก็ต้องทำอันที่หนึ่งก่อนอยู่ดี”

เอกภพ: “เหมือน User Story ที่เป็น Create User กับ Login ปะครับ? ก่อนจะ Login ได้ก็ต้อง Create User ก่อน”

ศิริพงษ์: “อืมมมม ก็ไม่เชิงนะครับในกรณีนี้เพราะน้องในฐานะ Product Owner บอกว่าอยากได้ Login ก่อน Create User ทีมพี่ก็ทำให้ได้ครับแต่ Feature นี้อาจจะราคาแพงนิดนึง”

ศิริพงษ์: “เพราะในเชิง Technical แล้วมันทำได้หมดแหละ ใน User Story สำหรับการ Login อาจจะ

1. อ่าน Text File ที่เก็บข้อมูล Username/Password มา

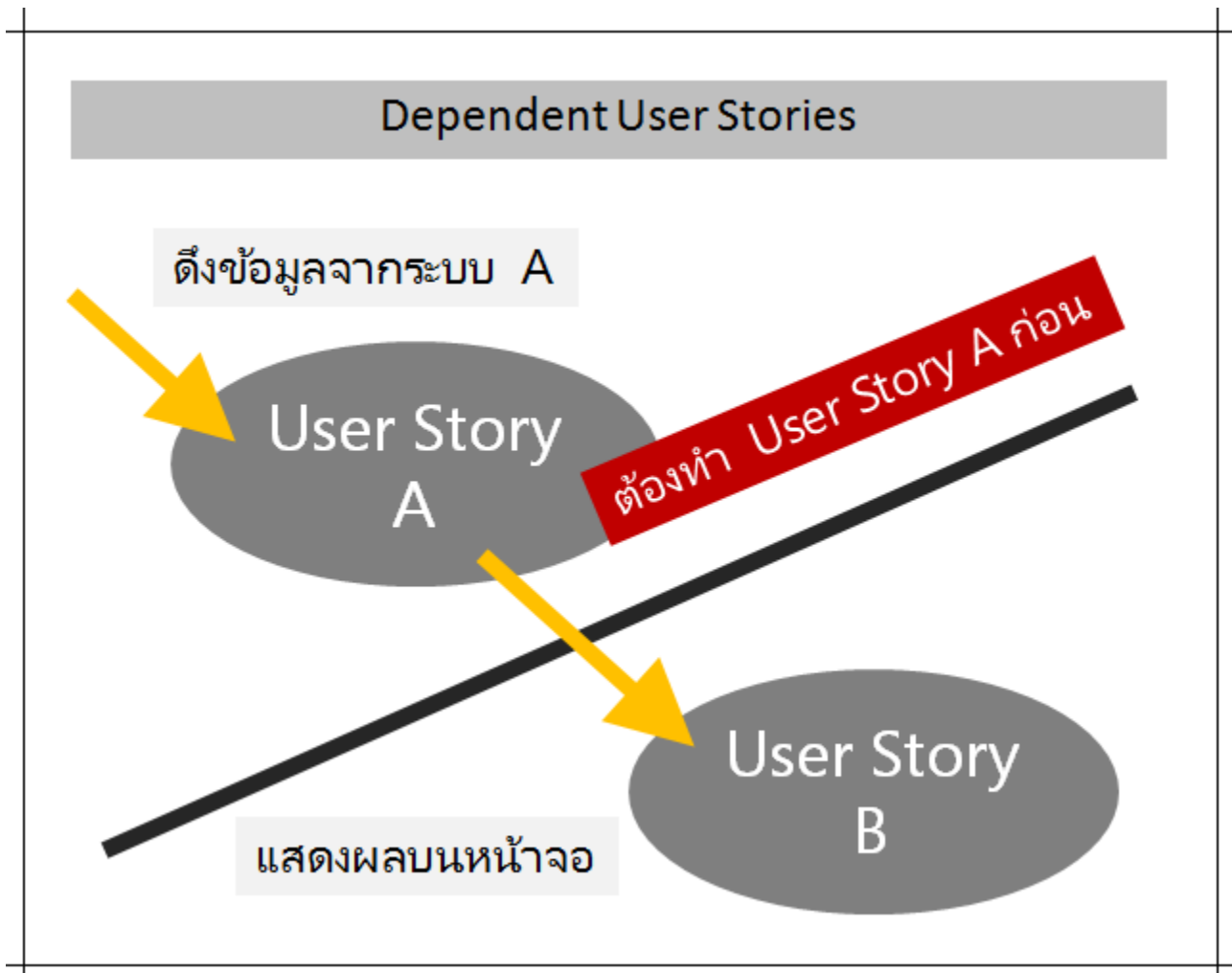
- เขียน DB Script ใส่ข้อมูลเข้าไปที่ Database ก่อน
- ไปดึงข้อมูลผู้ใช้งานจากเว็บ HR

เยอะหลายแบบ น้องอยากได้แบบไหนล่ะ?” ศิริพงษ์อธิบายแนวคิดในเชิงลึก

ศิริพงษ์: “หรือถ้าเอาให้ง่ายก็ได้เนะ ถ้าน้องบอกว่าไป Integrate กับ Facebook/Twitter/Google+ อื่นๆ น้องก็ไม่จำเป็นต้องมี User Story สำหรับการ Create User ด้วยซ้ำ”

เอกภพ: “อ้อครับ พอเข้าใจครับ ... แล้วแบบไหนที่มันมี Dependency ต่อกันล่ะครับ?”

ศิริพงษ์: “ประมาณนี้เอะครับ” ศิริพงษ์วาดภาพในกระดานเพื่ออธิบาย



ศิริพงษ์: “User Story A บอกว่าไปดึงข้อมูลจากระบบ A แล้ว User Story B บอกว่าเอาข้อมูลนั้นมาแสดงผลบนหน้าจอ แบบนี้ถ้าไม่ทำ A น้องก็ไม่ได้ข้อมูลมาใช้ใน B แบบนี้ครับที่เรียกว่า Dependency”

เอกภพ: "อ้อครับ ... แล้วถ้าเจอแบบนี้ผมควรทำไงอะ?"

ศิริพงษ์: "Good question" ☺

รวม User Story เพื่อลด Dependency

Combining User Stories to Reduce Dependency

เอกภพกำลังสงสัยว่าถ้าเจอกลุ่มของ User Story ที่ไม่เป็นอิสระต่อกันควรจะทำอย่างไร

ศิริพงษ์: “มีสองวิธีเท่าที่พี่พอจะรู้นะ หนึ่ง ...

Combine Them Into a Larger but Independent Story

เอกก็จับสอง Story นี้มารวมกันเป็น User Story ใหม่ที่มีขอบเขตใหญ่กว่าเดิม เช่น

User Story A: As a user, I want to pull information from System A, so that ...

User Story B: As a user, I want to see information on my screen, so that ...

เมื่อจับรวมกันอาจจะเป็นแบบนี้ครับ

As a user, I want to see information from System A, so that I can review ...

การรวมกันนี้จะเป็นการบอกให้ Development Team ต้องทำทั้ง User Story A และ B ให้เสร็จในคราวเดียวกันครับ Dependency ก็หายไปแล้ว”

ศิริพงษ์: “จะถามต่อว่า ‘แล้วถ้ารวมกันแล้วขนาดของ User Story นี้ใหญ่มากหะครับ’ ใช่ปะ? ... จั๊นพี่แนะนำวิธีที่สอง ...

Combine Them Into a Larger but Independent Story Then Split

ขั้นตอนเหมือนข้อแรกครับแต่หลังจากรวม Story แล้ว เราจะ Split มันให้เล็กลงอีกครั้งด้วยวิธีการที่จะไม่ทำให้มันมี Dependency กัน ถ้า As a user, I want to see information from System A, so that I can review ... ยังใหญ่ไป ลองเขียนแบบนี้ดูครับ

System A	Splitting User Stories with Data Boundary
Firstname	As a user, I want to see my firstname, lastname, so that ...
Lastname	
Address	As a user, I want to see my address, so that ...
Photo	As a user, I want to see my profile photo, so that ...
Contacts	As a user, I want to see my contact list, so that ...
...	...

1. As a user, I want to see my firstname, lastname, so that I can ...
2. As a user, I want to see my address, so that I can ...
3. As a user, I want to see my profile photo, so that I can ...
4. As a user, I want to see my contact list, so that I can ...
5. ...

เราสามารถเลือกแสดงข้อมูลจาก System A บางส่วนก่อนได้ครับ แบบนี้เราจะได้กลุ่มของ User Story ที่เป็นอิสระต่อกันและขนาดไม่ใหญ่เกินไป แนวทางนี้เรียกว่าการแบ่ง Story ด้วย **Data Boundary**

เขียน User Story เพื่อสร้างการพูดคุย

Writing a User Story that Drives Conversation

ศิริพงษ์: “ไว้พี่สอนเทคนิคการแบ่ง User Story ต่อวันหลังนะ ตอนนี้มาดูต่อกับกฎข้อสองของการเขียน User Story ที่ดี นั่นคือ ...

A Good User Story is Negotiable.

ที่เค้าให้เขียน User Story ลงการ์ดที่มีขนาดเล็กเพราะเค้าไม่อยากให้ Development Team มองว่าการ์ดนี้เป็น Requirement Spec ที่สมบูรณ์แล้ว Mike Cohn แนะนำไว้แบบนี้ครับ

User Story or Story Card is a pointer to a requirement.

มันเป็นแค่เครื่องมือชี้ทางให้ Development Team เดินไปคุยกับลูกค้า (User หรือ Customer) เพื่อให้เข้าใจสิ่งที่เค้าอยากได้มากขึ้น มันเป็นการสร้างการทำงานที่เน้น Individuals and interaction > processes and tools”

ศิริพงษ์: “ลองดูตัวอย่างนี้ครับ”

ศิริพงษ์: “ถ้า Development ทีมเห็นข้อมูลแค่นี้คงหงุดหงิดเพราะว่าเริ่มทำงานไม่ได้และการพูดคุยจะเกิดขึ้นครับ เช่น

Development Team: ‘มีคำถามครับ เาแค่ชื่อกับนามสกุลหรือครับ คำนำหน้าพวกนาย นาง นางสาว ไม่เอาหรือครับ?’

ลูกค้า: ‘เอาด้วยครับ ลืมคิดไปเลย’

Development Team: ‘ชื่อเอาภาษาไทยหรืออังกฤษ หรือทั้งสองครับ?’

ลูกค้า: ‘เอาแค่ภาษาไทยก่อนครับ’

...

As a user, I want to see firstname and lastname, so that I can review it.

Simply User Story That Encourages a Conversation

บทสนทนาก็จะดำเนินต่อไปจนทั้ง Development ทีมและลูกค้ามีความเข้าใจตรงกันใน User Story นี้ จากนั้นข้อมูลเพิ่มเติมที่ได้มาถูกอัปเดตลงการ์ด และข้อมูลเหล่านี้จะใช้ในการ Test ได้ด้วย”

As a user, I want to see firstname and lastname, so that I can review it.

Note: เอาคำนำหน้าชื่อด้วยเฉพาะ นาย นาง นางสาว

Note: เอาแค่ชื่อภาษาไทย ไม่เอาอังกฤษ

Note: ...

User Story With Additional Information

ศิริพงษ์: "ในทางกลับกัน ถ้าการ์ดนั้นมีข้อมูลอยู่เต็มไปหมดแบบนี้ Development ทีมก็จะไม่รู้สึกรว่าต้องคุยกับลูกค้าแล้วพวกเขาก็จะเริ่มทำงานไปเลย และปัญหาจะเกิดขึ้นตามลำดับแบบนี้ครับ

1. ลูกค้าเขียนการ์ดที่มีข้อมูลเต็มแผ่นแล้วแปะไว้บนบอร์ด
2. ผ่านไปสองเดือน Development Team หยิบการ์ดนี้มาทำโดยไม่พูดคุยเพิ่มเติมในเนื้อหาเกี่ยวกับลูกค้า
3. Development Team เองงานมาให้ลูกค้าดู
4. ลูกค้าก็บอกว่า 'อ้อ ตอนนี้ผมอยากได้ทั้งชื่อไทยและอังกฤษแล้วอะครับ การ์ดนี้ผมเขียนไว้ตั้งสองเดือนแล้ว ข้อมูลในนั้นมันไม่ค่อยจะอัปเดตเท่าไรแล้วอะครับ'



As a user. I want to see firstname and lastname. so that I can review it.

Note: เอาคำนำหน้าชื่อด้วยเฉพาะ นาย นาง นางสาว พวกยศ ตำแหน่งเช่น พล.อ มรว. ยังไม่เอา เอาแค่ชื่อภาษาไทยไม่เอาอังกฤษ เวลาแสดงผลต้องแยก Field กันสำหรับคำนำหน้าชื่อ ชื่อและนามสกุลแต่ต้องอยู่ในบรรทัดเดียวกันแต่ถ้าชื่อหรือนามสกุลยาวเกินว่า 20 ตัวอักษรให้แยกบรรทัดกัน ถ้าชื่อหรือนามสกุลหายไปจากฐานข้อมูลให้แสดงผล Field ที่หายไปเป็นสีแดงเพื่อเป็นการแจ้งเตือน

User Story That Contains Too Many Details

ดังนั้นถ้าอยากเขียน User Story ที่ Negotiate ได้ในการ์ดต้องมี

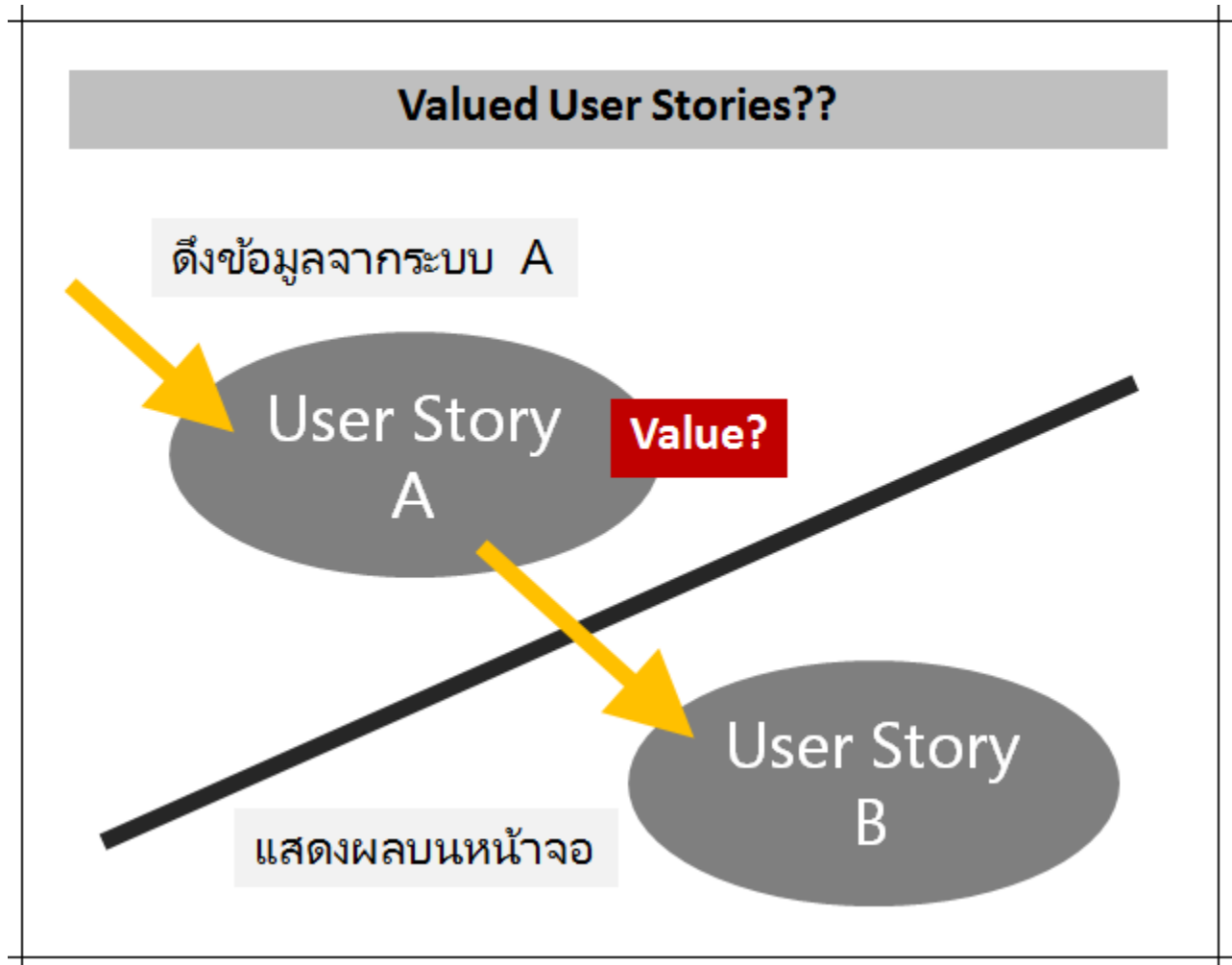
1. ข้อมูลที่บอกว่าลูกค้าอยากได้อะไร เขียนซัก 1-2 ประโยคก็พอ ถ้าให้ดีก็เป็นตามรูปแบบที่เคยบอกไป
2. เขียน Note ข้อมูลเพิ่มเติมที่ได้จากการพูดคุย รวมถึงคำถามที่ต้องการคำตอบด้วยครับ

ศิริพงษ์: “เป็นไง ยังไหม้ย? พี่จะได้ต่อเรื่อง Valuable”

เขียน User Story ให้มีคุณค่ากับผู้ใช้

Writing a User Story That Has Value to Users

ศิริพงษ์: “คนส่วนใหญ่มักจะคิดว่า User Story ที่ตัวเองเขียนนั้นมีคุณค่ากับผู้ใช้แต่จริงๆ แล้วมันอาจจะไม่ใช่เสมอไป พยายามตัวอย่างเดิมนะพี่ถามเอกว่าถ้าทีมเราทำ User Story A เสร็จมันมีคุณค่ากับผู้ใช้มั๊ย?”



User Stories That Provide No Value to The Users.

เอกภพ: “เออ ... ก็มีมั้งครับ ได้ข้อมูลมาเก็บไว้”

ศิริพงษ์: “ข้อมูลเก็บไว้ที่ไหน? ใน Database ปะ? จริงอยู่มันดูว่ามีคุณค่าอย่างน้อยสำหรับ Developer และมันก็โอเคนะถ้าผู้ใช้เค้ายินดีจะเขียน SQL เพื่อดึงข้อมูลที่เขาอยากเห็นขึ้นมาดูเอง น้องคิดว่าเค้าจะยินดีทำแบบนั้นปะ?” เค้ารำยาว

เอกภพ: "... คงไม่ครับ"

ศิริพงษ์: "ครับ ดังนั้น ..."

A Good User Story is Valuable to Users, or Purchasers.

ถ้าจะให้ดีที่สุดคือเราให้ผู้ใช้เป็นคนเขียน User Story เองรับรองได้ว่าเค้าจะเขียนแบบที่มีคุณค่าให้ตัวเองแน่นอน แต่ถ้าเราในฐานะ Product Owner หรือ Development Team ต้องเขียนเอง เราก็มีหลักการง่ายๆเวลาเขียน User Story ให้มีคุณค่าคือ

คิดถึงเวลาเรากินเค้ก



Think of Writing Valuable User Stories The Way You Eat a Cake.

ถ้าจะกินให้อร่อยก็ต้องใช้ช้อนตักจากบนลงล่างให้ผ่านทุกชั้นของเค้กจริงมั๊ย? เหมือนกันกับเปรียบเทียบชั้นเค้กกับ Application Layer ของเราดูผู้ใช้จะได้ประโยชน์สูงสุดครับเพราะแบบนี้พี่เลยเขียน User Story เหล่านั้นใหม่ให้เป็นแบบที่บอกไปครับ”

System A	User Stories That Provide Value to Users
Firstname	1 As a user, I want to see my firstname, lastname, so that ...
Lastname	
Address	2 As a user, I want to see my address, so that ...
Photo	3 As a user, I want to see my profile photo, so that ...
Contacts	4 As a user, I want to see my contact list, so that ...

User Stories That Provide Value to The Users.

ศิริพงษ์: “หลักการถัดไปคือ

หลีกเลี่ยงคำศัพท์เทคนิคต่างๆ

คำศัพท์เช่น Multi-Thread, Partition, Page/Table Lock, Connection Pool, Common Classes, Load Balancer, อะไรประมาณนี้ เพราะหนึ่งผู้ใช้หรือ Product Owner ไม่เข้าใจคำศัพท์พวกนั้นทำให้ยากต่อการจัดลำดับความสำคัญของ User Story สองคือมันเป็นสัญญาณอันตรายที่ User Story นี้จะมีคุณค่าเฉพาะกับ Developer หรือ Tester เท่านั้นครับ ตัวอย่าง

All external requests must be through Load Balancer.

เจอแบบนี้ผู้ใช้คงไม่เข้าใจว่ามันคืออะไรครับ เราลองเปลี่ยนมาเขียนโดยใช้มุมมองของเค้าดู อาจจะเป็นประมาณนี้

As a user, I want to 10 reports concurrently in background, so that I can continue doing other jobs.

การที่จะตอบโจทย์ User Story ข้างบน ทีมเราอาจจะตัดสินใจใช้หรือไม่ใช้ Load Balancer ก็ได้จริงมั้ย?
มันเป็น Technical Decision ที่ลูกค้าหรือผู้ใช้อาจจะไม่เกิดครับ”

ศิริพงษ์: “ถัดไป ...

เข้าใจความแตกต่างของ Users, และ Purchasers

ความแตกต่างของทั้งสาม Role นี้จะมีผลอย่างมากต่อการตัดสินใจว่า User Story นี้มีคุณค่าหรือไม่ครับ
ถ้าเราขาย E-Commerce Platform

- Users คือคนที่ใช้ซอฟต์แวร์ของเรา
- Purchasers คือคนที่ซื้อซอฟต์แวร์จากเราอาจจะเป็นผู้บริหารระดับสูงหรือคนที่ดูแลระบบ (Administrator)

พียกตัวอย่าง User Story นี้ครับ

All configurations must be stored and read from the single file.

ในมุมมองของ Users แล้วงานนี้ไม่มีคุณค่ากับเค้าเลยแต่ถ้าเป็น Admin ละ User Story นี้จะมีประโยชน์กับชีวิตเค้ามาก เอกต้องเข้าใจความแตกต่างตรงนี้ก่อนด้วยครับ”

ศิริพงษ์: “เยอะมัยๆ ฮ่าๆ ต่อไป Estimatable”

เขียน User Story ที่ประเมินขนาดได้

Writing a User Story That Can Be Estimated

เอกภพ: “พี่ครับ แต่ Kanban ไม่ได้บังคับว่าเราต้องประเมินขนาดของงานนี้ครับ?” เอกภพถามด้วยความสงสัย

ศิริพงษ์: “ใช่ครับ แต่การที่จะทำให้งานบน Kanban Board ไหลได้อย่างสะดวก การแบ่งงานใหญ่ให้เป็นงานที่เล็กลงก็จะช่วยเราได้เยอะและข้อมูลที่จะบอกเราได้ว่างานไหนใหญ่หรือเล็กก็มาจากการประเมินขนาดของงานครับ”

ศิริพงษ์: “คุณสมบัติข้อสี่ของ User Story ที่ดีคือ ...

A Good User Story Is Estimatable.

โดยปกติแล้วมีอยู่สามสาเหตุที่ทำให้การประเมินขนาดของ User Story เป็นเรื่องยาก

Development Team ไม่มีความรู้เฉพาะทาง (Domain Knowledge)

ง่าย ๆ คือ Development Team ไม่เข้าใจเรื่องราวที่อยู่ในการ์ด เช่น

As a user, I want to see an average velocity on a burndown chart for the current sprint, so that I can use this data for my analysis.

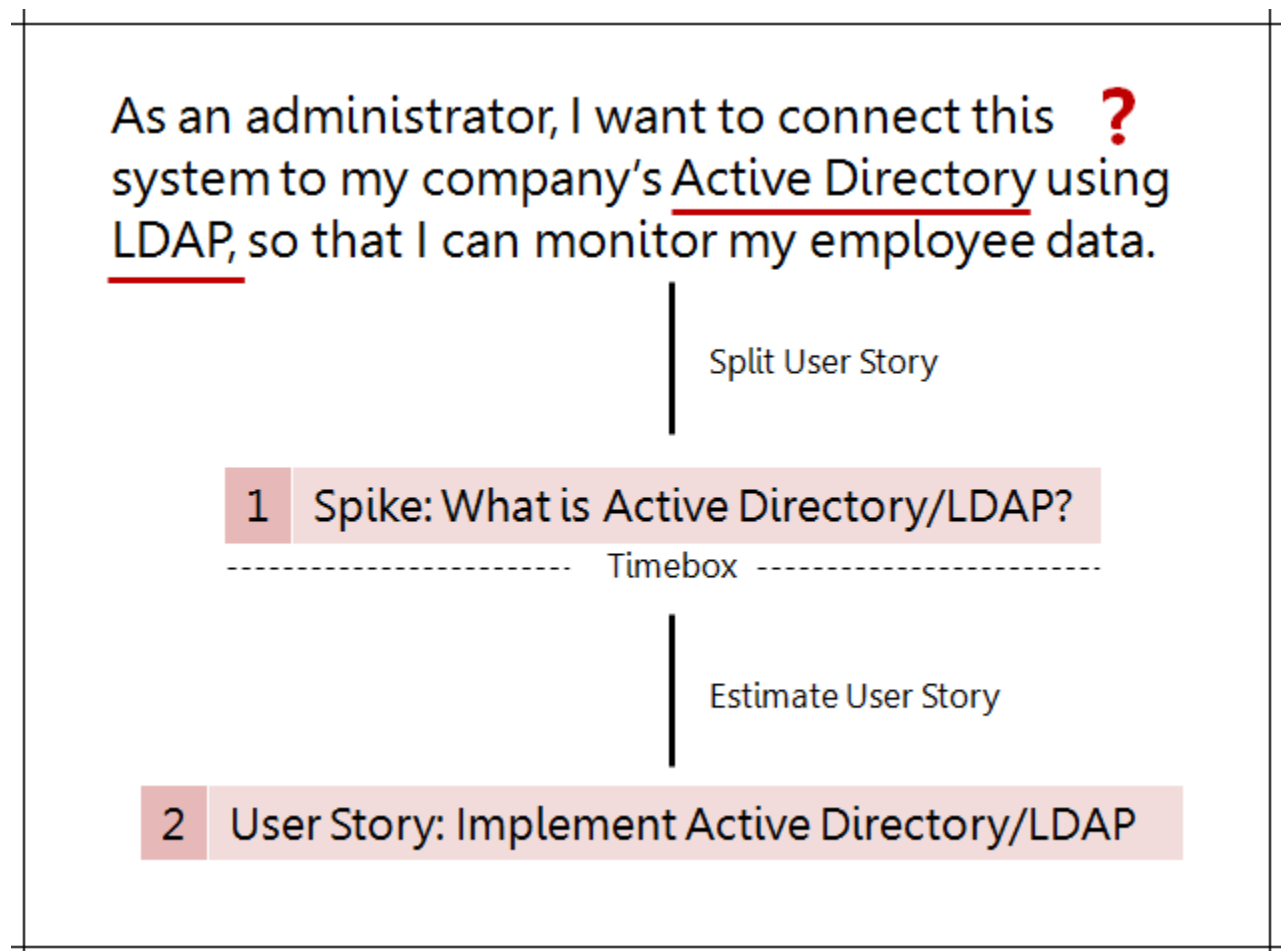
กรณีนี้ Development Team ไม่คุ้นเคยกับ Agile Development และคำว่า “velocity”, “burndown chart”, และ “sprint” ทำให้ไม่สามารถประเมินขนาดของงานได้ วิธีการแก้ปัญหาที่ดีคือการเดินไปคุยกับคนเขียน User Story นี้ อาจจะเป็น Product Owner หรือ User ให้เค้าอธิบายให้ฟังในรายละเอียดว่ามันคืออะไร เราก็มีหน้าที่อัปเดตข้อมูลที่ได้ลงไปในการ์ดด้วยครับ

Development Team ไม่มีความรู้ทางเทคนิค (Technical Knowledge)

กรณีนี้คือ Development Team เจอกับ User Story ที่มี Technology หรือ Technique ที่ไม่คุ้นเคยมาก่อนทำให้ไม่สามารถประเมินขนาดของงานได้ เช่น

As an administrator, I want to connect this system to my company's Active Directory using LDAP, so that I can monitor my employee data.

ถ้า Development Team ไม่เคยทำงานกับ Active Directory หรือ LDAP เลยการประเมินขนาดงานก็จะยากมากครับ วิธีการแก้ปัญหาคือให้ Developer ซักคนหรือสองคนไปทำ "Spike" หรือ Research ดูว่า Active Directory และ LDAP นั้นมันคืออะไรจะเอามาใช้งานได้อย่างไร กำหนดระยะเวลา Research ให้ชัดเจน (Timebox) เอาแค่ให้ได้ข้อมูลที่จะประเมินขนาดของงานเราได้พอครับ ถ้าเราใช้วิธีนี้เราจะแบ่ง User Story ที่ประเมินขนาดงานไม่ได้ออกเป็นสองส่วน



Spike

1. Spike เพื่อหาข้อมูลมา Estimate และ
2. User Story ที่ Implement จริง

User Story ใหญ่เกินไป

อันนี้ก็ตรงไปตรงมาครับ User Story นั้นใหญ่เกินไป เช่น สำหรับเว็บไซต์หางาน

- As a user, I want to find a job, so that ...
- As a user, I want to manage my resumes, so that ...

วิธีการจัดการปัญหานี้คือเราต้องย่อย User Story เหล่านี้ให้เล็กลง อาจจะใช้เทคนิค Data Boundary หรืออื่นๆ ไว้พี่จะเล่าให้ฟังต่อวันหลังครับ”

เขียน User Story ที่มีขนาดเหมาะสม

Writing a User Story That is Appropriately Sized

ศิริพงษ์: “หนึ่งในสาเหตุที่ทำให้การประเมินขนาดของ User Story เป็นเรื่องยากคือ User Story นั้นมีขนาดใหญ่เกินไปครับ ดังนั้น ...

A Good User Story is Small.

โดยทั่วไป User Story ใหญ่ (หรือที่เรียกว่า Epic) ที่เราเจอบ่อยๆมันจะมีอยู่สองลักษณะ

The Compound Story

User Story ประเภทนี้มันใหญ่เพราะมันประกอบด้วย User Story เล็กๆหลายอันรวมกันครับ เช่น ตอนแรกที่ Product Owner เขียนมาคือ

As a user, I want to manage my resumes, so that ...

ตอนที่เค้าเขียนก็คงรู้สึกว่ามันโอเคแล้ว แต่พอได้คุยกับ Development Team เค้าก็ได้ข้อมูลเพิ่มเติมว่าจริงแล้วมันคือ

- As a user, I want to create my resumes, so that ...
- As a user, I want to edit my resumes, so that ...
- As a user, I want to delete my resumes, so that ...
- As a user, I want to deactivate my resumes, so that ...
- As a user, I want to share my resumes, so that ...
- ...

เมื่อย่อย User Story ใหญ่ (Epic) ให้เป็น User Story ที่เล็กลงแล้ว บางอันก็จะมีขนาดที่เหมาะสมต่อการทำงาน แต่บางอันก็อาจจะย่อยให้เล็กลงไปอีกได้อีก เช่น As a user, I want to create my resumes, so that ... ประกอบด้วย

- I want to create my resume, which includes professional profile, working experience, salary history, skills, interests, photo, address, phone, mobile, email.

- I want to save my resume as draft.
- I want to publish my resume.
- I want to have multiple resumes.
- ...

การย่อย User Story ให้เลือกแบบนี้มีประโยชน์ครับ แต่ต้องมั่นใจว่ามันไม่เล็เกินไป เช่น I want to edit my firstname so that, หรือ I want to edit my available date เล่นแก้กันทีละ Field แบบนี้ก็ไม่วายนะ เล็กเกินไป



Compound



Complex

Compound vs. Complex

The Complex Story

แบบที่สองเป็น User Story ที่ซับซ้อนซึ่งไม่ง่ายต่อการแยกย่อยด้วยการใช้เทคนิค Data Boundary หรือ CRUD เหมือนกับ Compound Story ครับ ถ้าเราเจอ User Story ที่มีความซับซ้อนในตัวเองเราสามารถใช่วิธีการแบ่งมันด้วย Spike / Implement แบบตัวอย่างที่แล้วได้ครับ”

เขียน User Story ให้ทดสอบได้

Writing A Testable User Story

ศิริพงษ์: “มาถึงคุณสมบัติของ User Story ที่ดีข้อสุดท้ายแล้วครับ อดทนหน่อย เค้าให้กำลังใจเอกภพซึ่งกำลังนั่งมีนอยู่

A Good User Story is Testable.

User Story ที่ดีต้องทดสอบได้ ไม่งั้น Developer จะรู้ได้ยังไงที่เขียนโค้ดมานานถูกต้องสมบูรณ์มั้ยจริงปะ เอกอาจจะสงสัยว่ามีด้วยหรือที่ทดสอบไม่ได้ พี่ตอบให้เลยว่ามีครับ ส่วนใหญ่มันจะเป็น User Story ที่เป็นพวก Non-Functional Requirement ทั้งหลาย เช่น

- As a user, I want the software that is easy to use, so that ...
- As a user, I never want to wait too long to get a revenue report generated, so that ...

ถ้าเจอกรณีแรก เราอาจจะปรับการเขียน User Story ให้ทดสอบได้แบบนี้ครับ

- As a first-time user, I want to complete common tasks without training or reading user manual. คำว่า easy to use ถูกเปลี่ยนเป็นอะไรที่จับต้องได้มากขึ้น คนที่ใช้ระบบครั้งแรกสามารถทำงานได้โดยไม่ต้องมีการสอนหรืออ่านคู่มือ แบบนี้ทดสอบได้ครับถึงแม้จะเป็นการทดสอบโดยใช้การสังเกต (Observation) แทน Automated Testing หรือ
- As a first-time user, I want to be able to complete this task within 10 seconds. คราวนี้คำว่า easy to use ถูกเปลี่ยนเป็นตัวเลข 10 วินาทีที่วัดได้อย่างชัดเจน

สำหรับกรณีที่สอง ลองใช้วิธีการนี้ครับ

- As a user, I want to see a revenue report popped up on a screen within three seconds. คำว่า wait too long ถูกเปลี่ยนเป็นตัวเลขสามวินาทีที่สามารถทดสอบได้และที่ดีมากคือทำ Automated Test ได้ด้วย หรือ
- As a user, I want to see a revenue report popped up on a screen within three seconds in 85% of all cases. คล้ายๆเดิมแต่เราเพิ่มเงื่อนไขว่าทุก 100 ครั้งที่ใช้งาน Revenue report จะต้องถูกสร้างให้เสร็จภายในสามวินาทีอย่างน้อย 85 ครั้ง การกำหนดเงื่อนไขแบบนี้จะช่วยให้ User Story มีความใกล้เคียงความจริงมากขึ้นครับ การสร้าง Report มันก็ขึ้นกับปัจจัยหลายอย่าง เช่น จำนวนข้อมูล, ปริมาณงานที่ระบบกำลังประมวลผลอยู่, ความสามารถของเครื่อง

เซิร์ฟเวอร์, และอื่นๆ ดังนั้นการที่จะบอกว่า 'ไม่เกิดขึ้นเลย (never)' นั้นเกือบจะเป็นไปไม่ได้เลย
ครับ"

ศิริพงษ์: "ยากมัย INVEST ... มีคำถามปะครับ?"

เอกภพ: "INVEST พอเข้าใจแต่ต้องลองทำจริงจะครับ ที่เล่ามาทั้งหมดผมสนใจวิธีการแบ่ง User Story มาก
ที่สุดครับ พี่แนะนำมีเทคนิค Data Boundary แล้ว CRUD คืออะไร?"

ศิริพงษ์: "ได้ครับ เดี่ยวพี่อธิบายเทคนิคต่างๆที่ใช้ในการแบ่ง User Story ให้ฟัง"