### การนำแนวคิดของ PSP มาใช้ในการปรับปรุงแก้ไขอุตสาหกรรม

PSPได้รับการสอนในหลายมหาวิทยาลัยโดยมีผลงานที่น่าประทับใจ และเป็นที่สนใจของอุตสาหกรรม เป็นเครื่องช่วยในการฝึกอบรมวิศวกรรมซอฟต์แวร์ ในขณะที่มีการเผยแพร่รายงานเกี่ยวกับการสอนของPSPในการตั้งค่าห้องเรียน(ที่มหาวิทยาลัยและอุตสาหกรรม)มีการศึกษาระบบน้อยมากในการดำเนินการPSPในอุตสาหกรรม มีหลักฐานมากมายเกี่ยวกับประสิทธิภาพของงานเขียนโปรแกรมจริง ประสิทธิภาพจะวัดได้จากจำนวนวิศวกรที่ผ่านการฝึกอบรมที่ใช้PSPในการทำงานประจำวันและการปรับปรุงความสามารถในการผลิตและการกำจัดข้อบกพร่อง ในบทความนี้รายงานเกี่ยวกับการศึกษาการใช้แนวคิดPSPบางอย่างในองค์กรเชิงพาณิชย์ วิธีการสอบถามเชิงประจักษ์ที่เราใช้เป็นงานวิจัย ผลงานของเราระบุถึงปัญหาที่เกิดขึ้นในช่วงสี่กิจกรรมหลักของการดำเนินงานของPSP การวางแผน การอบรม การประเมินผลและการใช้ประโยชน์ เราอธิบายว่าปัญหาเหล่านี้ถูกกล่าวถึงอย่างไรและบทเรียนทั่วไปที่เรียนรู้จากการดำเนินงาน การถ่ายโอนข้อมูลโดยรวมของอัตราการฝึกอบรมของPSPเป็น46.5% สำหรับวิศวกรในการศึกษาของเรา ผู้ที่ใช้แนวคิดPSPที่สอนทั้งหมดในที่ทำงานช่วยปรับปรุงขีดความสามารถในการตรวจพบข้อบกพร่องของพวกเขา

# 1 บทนำ

## 1.1หลักสูตรPSPและประสิทธิภาพ

(ต่อจากนี้PSP)[14]เป็นชุดวิวัฒนาการของวิศวกรรมซอฟต์แวร์ส่วนบุคคลการเรียนรู้และการปฏิบัติ ในช่วงต้น PSPมีวัตถุประสงค์เพื่อเป็นหลักสูตรการเรียนในชั้นเรียนสำหรับนักศึกษาระดับบัณฑิตศึกษาหรือผู้สำเร็จการศึกษาระดับสูง หลักสูตรPSPได้รับการแนะนำในหลายมหาวิทยาลัย เช่น มหาวิทยาลัยคาร์เนกีเมลลอน[12] มหาวิทยาลัยการบิน[19] มหาวิทยาลัยแมคกิลล์[34][35] มหาวิทยาลัยแมสซาชูเซตส์และมหาวิทยาลัยแบรดลีย์[15] ยิ่งไปกว่านั้น ได้กล่าวไว้ว่าทั้งนักศึกษามหาวิทยาลัยและวิศวกรที่มีประสบการณ์ได้รับประโยชน์อย่างมากจากการทำงานบนPSP[13] หลักสูตรPSPยังถูกนำมาใช้ในการฝึกอบรมวิศวกรรมมืออาชีพในอุตสาหกรรม[12][13][23][33][26]

มาตรการที่มีอยู่ของประสิทธิภาพของการฝึกอบรมPSPจะขึ้นอยู่กับสิ่งที่นักเรียนทำในแบบฝึกหัดของหลักสูตร สามมาตรการทั่วไปของประสิทธิภาพของการฝึกอบรม PSP คือ: (a) สัดส่วนของนักเรียนที่จบหลักสูตรPSP (b) ขอบเขตของการปรับปรุงประสิทธิภาพของโปรแกรมเมอร์และ (c) ขอบเขตของการปรับปรุงในการตรวจหาข้อบกพร่องและทักษะการกำจัด

อัตราการสำเร็จการศึกษาที่ได้รับในห้องเรียนแตกต่างกันไป ตัวอย่างเช่น หลักสูตรPSPของมหาวิทยาลัยมีอัตราการสำเร็จการศึกษา53% (8ใน15ของนักเรียน)[31] อัตราการสำเร็จ4%(1ใน24ของนักเรียน)สำหรับหลักสูตรที่สอนในอุตสาหกรรม[31]

อัตราความสำเร็จจากหลักสูตรอุตสาหกรรม 45% (นักเรียน 9 คนที่เข้าร่วมการทำแบบฝึกหัดครั้งสุดท้ายเมื่อเทียบกับ 20 คนที่จบการศึกษาครั้งแรก)[26] และอัตราการจบหลักสูตร77%(10จาก13)[24] สำหรับการฝึกอบรมที่โมโตโรลา

นอกจากนี้ มีการระบุไว้ว่านักเรียนที่เรียนผ่านหลักสูตร PSP มีการปรับปรุงประสิทธิภาพโดยเฉลี่ยตั้งแต่ประมาณ 21% [17]ถึง 35% [12]และการปรับปรุงกว่าสิบครั้งในจำนวนของข้อบกพร่องการทดสอบ[12]และลดจาก 58% ในจำนวนเฉลี่ยของข้อบกพร่องที่พบ

## 1.2 ประสิทธิภาพของ psp ในอุตสาหกรรม

กับผลลัพธ์ที่น่าประทับใจของ psp ซึ่งเป็นสิ่งที่น่าสนใจในอุตสาหกรรม เช่นเดียวกับวิธีการฝึกอบรมวิศวกรซอฟต์แวร์ แต่อย่างไรก็ตาม จากมุมมองในอุตสาหกรรม ประสิทธิภาพของ psp ควรที่จะมีพื้นฐานของการประมวลผลตามสิ่งที่วิศวกรทำในงานเขียนโปรแกรมจริงของพวกเขา รวมถึงสภาพแวดล้อมในการเขียนด้วย ในทางตรงกันข้ามกับหลักสูตรการออกกำลังกาย และ การจัดการเกี่ยวกับห้องเรียน การปรับปรุงหลักสูตรการออกกำลังกายในห้องเรียน อาจจะไม่เป็นประโยชน์เท่ากับงานเขียนโปรแกรมจริงและสภาพแวดล้อม

ความแตกต่างระหว่างห้องเรียนและงานเขียนโปรแกรมจริง คือ การเน้นส่วนสำคัญ โดยผลกระทบที่อาจเกิดขึ้นจากการนำมาใช้ซ้ำ เพื่อประโยชน์ของ psp ในการจัดการหลักสูตร สำหรับตัวอย่าง มารยาทในการออกกำลังกายของ psp ได้รับการออกแบบมาเพื่อช่วยให้สามารถนำมาใช้ซ้ำได้จากการออกกำลังกายครั้งต่อไป แต่กลับพบว่าขอบเขตของการใช้ซ้ำ

ในส่วนของโค้ดจะเพิ่มสูงขึ้นถึง 75% สำหรับการออกกำลังกายบางอย่าง กับค่าเฉลี่ยประมาณการอยู่ที่ 40% ของการใช้ซ้ำ แต่การใช้ซ้ำในระดับสูงอาจช่วยอธิบายการปรับปรุงที่เห็นได้ชัดขึ้นโดยนักเรียน ในแง่ของการผลิตที่เพิ่มขึ้นและความหนาแน่นลดลงเมื่อเวลาผ่านไป

จนถึงปัจจุบันมีหลักฐานเพียงเล็กน้อย เพื่อใช้ประโยชน์ของ psp กับงานเขียนโปรแกรมจริง แต่กลับมีเพียงรายงานล่าสุดเกี่ยวกับจำนวนวิศวกรเท่านั้น ผู้ซึ่งยังคงใช้แนวคิดของ psp หลังจากจบหลักสูตร ถ้าวิศวกรไม่ได้ใช้ทักษะ psp ของพวกเค้า หลังจากจบหลักสูตรแล้วแรงจูงใจ ในการจัดการหลักสูตร psp ก็มีน้อยมาก

ในฐานะการสำรวจเบื้องต้นของการเตรียมตัวสำหรับ psp ในอุตสาหกรรมเราได้สัมภาษณ์ผู้ที่จบปริญญาตรีจากมหาวิทยาลัย mcgill ที่เคยเข้าร่วมหลักสูตร psp และมีการเขียนโปรแกรมในอุตสาหกรรม เราต้องการทราบว่า psp มีผลต่อกระบวนการส่วนบุคคลอย่างไร ไม่มีพวกเขาใช้เทคนิค psp ที่ได้รับการสอน ดังนั้นจึงจำเป็นที่จะต้องดำเนินการตรวจสอบระบบ psp แบบเรียลไทม์ในสภาพแวดล้อมการเขียนโปรแกรมแบบเรียลไทม์เพื่อทำความเข้าใจกับอุปสรรคในการใช้ psp กับงานเขียนโปรแกรมจริงและเป็นแนวทางในการดำเนินการในอุตสาหกรรมที่ประสบความสำเร็จ

## 1.3 ภาพรวมของกระดาษ

ในเอกสารนี้ เรานำเสนอการศึกษาการดำเนินการของแนวคิดของ PSP ในองค์กรทางการค้า วิธีการสอบถามข้อมูลเชิงประจักษ์ที่เราจ้างงานการดำเนินการวิจัยได้ ผลของเราระบุปัญหาที่พบในระหว่างกิจกรรมหลักสี่ของการดำเนินการของ PSP การวางแผน การฝึกอบรม การประเมิน และใช้ประโยชน์จาก เราอธิบายว่า ปัญหาเหล่านี้ได้รับการแก้ไข และเรียนทั่วไปจากการใช้งาน ผลลัพธ์เหล่านี้จะหวังว่าจะเป็นประโยชน์สำหรับองค์กรอื่น ๆ ที่เริ่มดำเนินการในการใช้ PSP เป็นพาหนะสำหรับการปรับปรุงกระบวนการของการเป็นวิศวกรซอฟต์แวร์ พวกเขายังมีผลงานการวิจัยวรรณคดีในแง่ของการนำเสนอแนวทางการใช้เทคโนโลยีกระบวนการผลิต และ ในการระบุปัญหาและอุปสรรคของการดำเนินงานและกลยุทธ์ในการเอาชนะพวกเขา

เราดำเนินการศึกษากับวิศวกร 28 สั้น ๆ ผลลัพธ์ของเราระบุว่า 57% (16/28) วิศวกรซอฟต์แวร์ที่ได้รับการสอนแนวคิดของ PSP (คือวัดและที่มีโครงสร้างรหัส) มีโอกาสที่จะใช้แนวคิดในงานเขียนโปรแกรมจริง ยัง 46.5% (13/28) วิศวกรยังใช้แนวคิดในงานเขียนโปรแกรมเจ็ดเดือนหลังจากนั้น วิศวกรที่ใช้รหัสรีวิวเห็นการปรับปรุงอย่างมากในความสามารถกำจัดข้อบกพร่อง

ส่วนถัดไปของกระดาษที่นำเสนอภาพรวมของ PSP เพื่อทำความรู้จักผู้อ่าน มีแนวคิดพื้นฐาน และยัง แนะนำบริบทและวัตถุประสงค์ของการศึกษาของเรา ส่วนที่ 3 เป็นคำอธิบายของวิธีการวิจัยของเรา ในหัวข้อที่ 4 เราอธิบายปัญหาที่เราพบในการใช้ PSP แนวคิด วิธีเหล่านี้ได้รับการแก้ไข และการประเมินการดำเนินงานและผลประโยชน์ของตน นี่คือตาม ด้วยบทเรียนสำคัญที่เราได้เรียนรู้ระหว่างกิจกรรมที่ดำเนินการในส่วนที่ 5 ในที่สุด ในส่วนที่ 6 เราสรุปกับบทสรุปของกระดาษ และเสนอเป็นกรอบสำหรับการประเมินการใช้งาน PSP

# 2 ที่มาของการวิจัย

## 2.1 กระบวนการพัฒนาซอฟต์แวร์ส่วนบุคคล (psp)

PSP มีวัตถุประสงค์เพื่อปรับปรุงกระบวนการพัฒนาซอฟต์แวร์ของวิศวกรซอฟต์แวร์เพื่อการพัฒนาของวิธีปฏิบัติงานซอฟต์แวร์ที่ดีขึ้นอย่างต่อเนื่อง แนวทางปฏิบัติเหล่านี้จะลดขนาดของแนวทางปฏิบัติ 12 ข้อจากแนวทางการปรับปรุงซอฟต์แวร์แบบ CMM PSP แบ่งออกเป็น 7 ขั้นตอนที่แตกต่างกัน

ในระยะแรกของ PSP ให้นักเรียนเรียนรู้วิธีการวัดผลงานของพวกเขา เช่นเดียวกับวิธีการใช้แบบฟอร์มของ PSP และสคริปต์ กระบวนการส่วนบุคคลจะถูกกำหนดไว้ล่วงหน้า ประกอบด้วยการออกแบบ เขียนโปรแกรม คอมไพล์ และการทดสอบ กระบวนการเหลานี้มีการวางแผนล่วงหน้าตั้งแต่ขั้นตอนการวางแผน และดำเนินถึงขั้นตอนสุดท้ายคือขั้นตอนวิเคราะห์ระบบ

ต่อมา จะเน้นการวัดขนาด และการประเมินทรัพยากร แบบประเมินขนาด และจะได้เรียนรู้เทคนิคการประมาณขนาด และการใช้งานขนาดและการวางแผนทรัพยากร โฟกัสของนักเรียนต่อไปคือ การจัดการข้อบกพร่อง พวกเขาสอน code และทบทวนการออกแบบ เพื่อที่จะพบข้อบกพร่องตั้งแต่ต้นรวมทั้งรายละเอียดการออกแบบ เทคนิคการวิเคราะห์ และการดำเนินวิเคราะห์แบบง่าย

สุดท้ายการทำงานแบบวงรอบจะช่วยให้นักเรียนมีระดับทักษะที่สามารถเรียนรู้โปรแกรมขนาดใหญ่ขึ้น

## 2.2 การศึกษาบริบทและวัตถุประสงค์

องค์กร CAE Electronics Ltd. ตั้งอยู่ใน Montréal เป็นผู้จัดจำหน่ายซอฟต์แวร์จำลองการบินชั้นนำ (ต่อจากนี้ไปจะเรียกว่า CAE) CAE มีวิศวกรซอฟต์แวร์ประมาณ 1,200 คน

ฝ่ายทรัพยากรบุคคลร่วมกับเว็บไซต์ SEPG ตัดสินใจเกี่ยวกับกลยุทธ์การปรับปรุงกระบวนการวัดตาม การดำเนินการตามแนวความคิดจาก PSP ก็ถือว่าเป็นส่วนหนึ่งของกลยุทธ์โดยรวมนี้

ในกรณีแรก CAE ตัดสินใจว่าการศึกษาควรจะดำเนินการต่อไปนี้ :

1. ปรับแต่ง PSP ให้สอดคล้องกับบริบทของ CAE,
2. สนับสนุนการเปลี่ยนแปลงบรรยากาศภายใน CAE เน้นให้ความสำคัญกับการเรียนรู้
3. การวัดเชิงประเมินขอบเขตวิธีการดำเนินงานของ PSP มีผลต่อการเปลี่ยนแปลงการทำงานของวิศวกรซอฟต์แวร์ที่มีส่วนร่วม
4. ประเมินประโยชน์ของ PSP ภายใน CAE (จากที่ได้บอกไว้ก่อนหน้านี้ในส่วนของประโยชน์ PSP นั้นมีพื้นฐานมาจากการฝึกแบบ PSP ในห้องเรียน)

แน่นอนการประเมินเทคโนโลยีใหม่ ๆ ผ่านการศึกษานำร่องคือแนวทางปฏิบัติที่แนะนำในระหว่างนวัตกรรมนั้นๆ กำลังเขาสู่องค์กร ในบทความนี้เรามีความกังวลเกี่ยวกับการศึกษานำร่องนี้

การวางแผนสำหรับการศึกษานำร่องเริ่มต้นในตอนท้ายของปีค.ศ. 1994 และโครงการนำร่องเสร็จสิ้นในเดือนกุมภาพันธ์ ปีค.ศ. 1996 เรามี 28 วิศวกรซอฟต์แวร์ที่มีส่วนร่วมในการศึกษาครั้งนี้  รายละเอียดของวิธีการวิจัยที่ใช้และการดำเนินงานตาม

# 3 วิธีการวิจัย

วิธีที่เราจะใช้เพื่อศึกษาความเป็นไปได้ของการนำไปใช้งานการวิจัยและการดำเนินการคือ PSP วิธีการวิจัยการดำเนินการต่อไปนี้,นักวิจัยที่มีส่วนร่วมโดยตรงในข้อมูลเบื้องต้น,การตรวจสอบ ,และการประเมินผลของการวางแผนการเปลี่ยนแปลงขององค์กร โดยปกติค่านี้จะทำให้เป็น effort ในการทำงานร่วมกันกับผู้ให้การสนับสนุนของการเปลี่ยนแปลงที่มีอยู่ภายในองค์กร การวิจัยเชิงปฏิบัติการมุ่งมั่นที่จะบรรลุวัตถุประสงค์ : การแก้ปัญหาการเปลี่ยนแปลงขององค์กรที่ใช้งานได้จริง และเพื่อเป็นการเพิ่มสต็อกความรู้ทางวิทยาศาสตร์ของเรา มีข้อกำหนดที่สมบูรณ์มากขึ้นได้ตามที่ระบุไว้ในการดำเนินการดังนี้ ( นอกจากนี้ยังมีลักษณะที่ไม่มีการดำเนินการเกี่ยวกับการวิจัยในด้านของปัจจัยต่างๆเช่นภายในและภายนอกจะมีอายุการใช้งานที่ให้ไว้โดย Jenkins ) :

การวิจัยการดำเนินการพร้อมกันการให้ความช่วยเหลือในการแก้ปัญหาการใช้งานจริงและความรู้ทางวิทยาศาสตร์ที่มีการขยายการใช้งานรวมทั้งช่วยเพิ่มความสามารถของ” actors”ที่กำลังดำเนินการร่วมกันในสถานการณ์ที่มีการส่งทันทีโดยใช้ข้อมูลความคิดเห็นในกระบวนการที่เป็นวงจรโดยมุ่งไปที่ความเข้าใจที่เพิ่มขึ้นของสถานการณ์ทางสังคมที่กำหนดให้เป็นหลักที่ใช้ได้สำหรับการทำความเข้าใจกระบวนการของการเปลี่ยนแปลงในระบบสังคมและดำเนินการภายในกรอบความรับผิดชอบทางจริยธรรมที่เป็นที่ยอมรับทั้งสองฝ่าย

ในขณะที่ไม่มีการตรวจสอบอย่างเป็นระบบในการใช้งานและการนำไปปฏิบัติในการดำเนินการวิจัยในวิศวกรรมซอฟต์แวร์ได้มีการจัดทำขึ้น ซึ่งเราสามารถที่จะเขียนในข้อสรุปเริ่มต้นบางส่วนในการนำไปปฏิบัติได้โดยการดูที่ฟิลด์ในเครือของระบบสารสนเทศเพื่อการจัดการ (MIS) ระบบสารสนเทศเพื่อการวิจัยเชิงประจักษ์ , แบบดั้งเดิมที่มีการใช้งานวิจัย 4 วิธี : 1.กรณีศึกษา, 2.ขอบเขตข้อมูล , 3.การทดสอบภาคสนาม และ 4.การทดลองในห้องปฏิบัติการ ได้การดำเนินการเมื่อไม่นานมานี้มีงานวิจัยที่ได้รับการว่าจ้างโดยนักวิจัยของระบบสารสนเทศเพื่อการจัดการและจะได้รับการพิจารณาว่าเป็นวิธีที่ถูกต้องเพื่อรับความรู้เกี่ยวกับระบบสารสนเทศเพื่อการจัดการที่เกี่ยวข้องปรากฎการณ์ ยิ่งไปกว่านั้นถ้าเราพิจารณาว่าการศึกษาของเราจะต้องอยู่ในการนำเทคโนโลยีสารสนเทศมาใช้ในองค์กรแล้วการดำเนินการวิจัยนี้ได้รับการพิจารณาให้เป็นวิธีการวิจัยที่มีผลบังคับใช้

# 4 ผลลัพธ์จากการวิจัย

คล้ายกับ[23] เราได้แยกการทำงานของ PSP เป็น 4 กิจกรรม: การวางแผน การฝึกอบรม การประเมิน และการใช้ประโยชน์ กิจกรรมเหล่านี้ไม่จำเป็นต้องทำตามลำดับ  วิธีการเหล่านี้ให้แนวทางที่เป็นประโยชน์สำหรับการจัดกลุ่มประเด็นปัญหาที่เกิดขึ้นในระหว่างการศึกษา สำหรับแต่ละกิจกรรมเหล่านี้เราสามารถนำเสนอรายละเอียดของการดำเนินกิจกรรมกับปัญหาและวิธีแก้ไขปัญหาเหล่านี้ได้

## 4.1 การวางแผนการดำเนินงาน

วัตถุประสงค์ของขั้นตอนการวางแผนคือการกำหนดวิธีการโดยรวมสำหรับการดำเนินงานเบื้องต้น เช่น เกณฑ์การคัดเลือดผู้มีส่วนร่วมกับงาน และจัดเตรียมเอกสาร ( เช่นแบบฟอร์มเก็บรวบรวมข้อมูล และสไลด์การบรรยาย )

## 4.1.1 แนวทางทั่วไปสำหรับนำ PSP ไปใช้

ในการฝึกอบรมวิศวกรบน PSP สองตัวเลือกหลักที่ได้รับการพิจารณา : การเรียนการสอนในชั้นเรียนและการใช้งานการเขียนโปรแกรมจากหนังสือ PSP และการผสมผสานกันของการเรียนการสอนในชั้นเรียนและการใช้ในอาชีพการเขียนโปรแกรม

ในอดีตเราได้ติดตามการสอนในแบบฉบับของ Humphrey's อย่างใกล้ชิดในห้องเรียนที่มีการสอนPSP และ ให้ผู้เข้าร่วมทำแบบฝึกหัดในชั้นเรียนเพื่อการฝึกฝนเทคนิคใหม่ที่พวกเขาเรียนรู้ ช่วงหลังยังใช้การบรรยายในห้องเรียนแต่ถึงอย่างไรก็ตามผู้เข้าร่วมการนำเทคนิคต่างๆที่พวกเขาเรียนรู้ในการเขียนโปรแกรมที่แท้จริงของตนในปัจจุบัน วิธีการที่เก่าแก่ก้มีข้อเสียดังที่แสดงอยู่ในรายการด้านล่าง :

•จากประสบการณ์ที่ผ่านมาของเราการเรียนการสอนที่เป็นรุ่นก่อนหน้าของ PSP ที่ McGill University,การนำเสนอแบบเต็ม PSP ก็ประมาณ 13 สัปดาห์และมีการบรรยาย 3 ชั่วโมงต่อสัปดาห์ นอกจากนี้ยังให้นักเรียนใช้เวลาประมาณ 4.5 ชั่วโมงของ ” effort”ต่องานที่ มอบหมาย ( ทั้งหมดของนักศึกษามีโปรแกรมเมอร์ที่มีประสบการณ์ที่มีค่ามัธยฐานของประสบการณ์ 6 ปีแต่ถึงอย่างไรก็ตามเพียง 50% ที่มีประสบการณ์ในอุตสาหกรรม ) จำนวนเวลานี้ไปอยู่ที่ประมาณ 70 ชั่วโมงความมุ่งมั่นไม่รวมเวลาสำหรับการแสดงความคิดเห็นและสำหรับการเขียนรายงาน ฮัมฟรีย์บันทึกย่อแบบเต็มรูปแบบที่แน่นอนระหว่าง 150 ถึง 200 ชั่วโมงต่อ” effort”ของวิศวกร ” effort”ดังกล่าวจะถือว่าเป็นการลงทุนที่มีขนาดใหญ่เกินไปตั้งแต่ช่วงเวลานี้จะต้องได้รับการนำออกมาจากโปรเจคกำหนดการสำหรับผู้ใช้หลายคนที่ทำงานอยู่ในโครงการเดียวกันวิธีนี้จะทำให้ผู้จัดการจากการสนับสนุนการฝึกอบรมการดำเนินการนี้เป็นสิ่งสำคัญเนื่องจากการสนับสนุนของหัวหน้างานเป็นสิ่งสำคัญที่ดีเทอร์มิแนนต์ของการถ่ายโอนข้อมูลในเชิงบวกของการฝึกอบรมทักษะในการใช้งาน

• 4 Professional นักพัฒนาซอฟต์แวร์ที่ก่อนหน้านี้มีการนำหลักสูตร PSP ที่มหาวิทยาลัย McGill ถูกสัมภาษณ์เพื่อกำหนดวิธีการใช้ PSP มีอิทธิพลต่อกระบวนการส่วนบุคคลของตนได้ไม่มีใครเลยสักคนที่เคยใช้แนวคิดของ PSP ในงานการตั้งโปรแกรมที่แท้จริงของตน ระบุเหตุผลที่ทำให้ SPconcepts ไม่มีการใช้ในการฝึกปฏิบัติรวม :

•การเขียนโปรแกรมขององค์กรงานแทนที่จะมุ่งเน้นไปที่การทำแบบฝึกหัดการตั้งโปรแกรมในห้องเรียน PSP โดยปกติแล้วมีการดำเนินการในทีมได้ เห็นได้ชัดว่ามันไม่ใช่วิธีที่งานส่วนตัวของ PSP สามารถนำมาใช้กับทีมงาน

•ต้องใช้ PSP ในคอลเลกชันของข้อมูลเป็นจำนวนมากได้ โดยไม่มีเครื่องมืออัตโนมัติในการจัดการและการแปลความหมายข้อมูลนี้มันอาจจะกลายเป็นเรื่องยุ่งยากที่จะนำไปใช้กับ PSP ได้ ใช้เครื่องมือที่เหมาะสมไม่พร้อมใช้งานสำหรับบุคคลเหล่านี้ในองค์กรของตนได้

•มีการขาดการสนับสนุนด้านการจัดการได้ ค่าใช้จ่ายในการดำเนินการที่คอลเลกชันของข้อมูลและการตีความในระยะสั้นจะต้องมาจากเวลาของพวกเขาเองเว้นแต่การลงโทษทางเศรษฐกิจของการบริหารจัดการได้ ห้องพักได้รับการนำไปสู่การทำงานล่วงเวลาอยู่แล้วดังนั้นจึงไม่ได้มีเรื่องเวลาของพวกเขาเองสำหรับ PSP อีกต่อไป การสนับสนุนด้านการจัดการมีความจำเป็นที่จะต้องรักษาทักษะที่ได้เรียนรู้ในระหว่างการฝึกอบรม

•ไม่มีฮีโร่ในองค์กรที่ทำงานใน ผู้ที่เป็นโปรแกรมเมอร์ที่มีวินัยในการดำเนินการและผู้ที่มีความบกพร่องน้อยกว่าจะได้รับความนับถือมากกว่าการแก้ปัญหาผู้บันทึกโปรเจคในวิกฤตที่เกิดจากแนวปฏิบัติในการเขียนโปรแกรมที่ไม่เพียงพอ ซึ่งหมายความว่าการให้รางวัลในปัจจุบันโครงสร้างขององค์กรที่มี incongruent กับหลักการของ PSP ซึ่งทำให้เป็นเรื่องที่ยากมากที่จะฝึกฝนเทคนิค PSP ในงานที่ได้รับมอบหมาย ดังนั้นจึงไม่มีแรงบันดาลใจสำหรับการลงโทษทางวินัยในขั้นตอนการเขียนโปรแกรม

•แต่ก็เป็นการยากที่จะใช้ในการนำแนวทางปฏิบัติในการดำเนินการทางวินัยของ PSP อยู่คนเดียวเมื่อส่วนที่เหลือของทีมงานของคุณจะไม่ได้ทำเช่นนั้น การขาดการสนับสนุนจากเพื่อนร่วมงานที่เชื่อว่าเป็นการยับยั้งการฝึกฝน PSP ในการทำงานจึงเป็นสิ่งที่เห็นได้ชัดว่าไม่มีปัญหาในการถ่ายโอนที่ใช้งานได้จริงสิ่งที่ได้เรียนรู้ในห้องเรียนเพื่อมหาวิทยาลัยการตั้งโปรแกรมการใช้งานจริงและสภาพการใช้งาน

•มีความเชื่อว่าโดยทั่วไปจะมีเพียงร้อยละ 20 ของงานที่สำคัญมีการเรียนรู้ทักษะจากการฝึกอบรมอย่างเป็นทางการและการศึกษาที่เหลือจะได้เรียนรู้ไปใช้ในการทำงาน . ดังนั้นจึงมีกรณีของการให้บริการที่ผู้เข้าร่วมประชุมพร้อมด้วยการฝึกอบรมในงานในฐานะที่เป็นส่วนหนึ่งของ PSP รดำเนินงานตามแผนได้

•ที่เราต้องการเพื่อประเมินประโยชน์ของแนวความคิดในแบบเรียลไทม์ PSP ที่มีต่อสภาพแวดล้อมการทำงานของ CAE และไม่ได้อยู่ในการตั้งค่าเทียม ดังนั้นเราจึงต้องทำการประเมินประโยชน์ของแนวคิดในการเขียนโปรแกรมจริง PSP งาน

•เราต้องการที่จะได้รับข้อมูลสำหรับการปรับแต่งการทำงานของ PSP รูปแบบของ CAE ได้ ดังนั้นจึงเป็นสิ่งจำเป็นเพื่อการศึกษาการทำงานในแบบเรียลไทม์ PSPwith วิศวกรงานที่ไม่ได้อยู่บนแบบฝึกหัดในชั้นเรียน

ทางเลือกสำหรับการผลิตจึงถูกที่มีการผสมผสานของการบรรยายในห้องเรียนตามด้วยการตั้งโปรแกรมการทำงานในงานที่ต้องทำ

## 4.1.2 การเลือกผู้เข้าร่วม

ตามหลักการผู้เข้าร่วมในการศึกษานำร่องจะถูกสุ่มเลือกวิศวกรของ CAE เพื่อให้แน่ใจว่าตัวอย่างที่เข้าร่วมเป็นตัวแทนที่เราต้องการพูดคุย แต่ถึงแม้ว่าจะไม่สามารถเลือกตัวแทนที่เราต้องการพูดคุยได้ เราก็พยายามที่จะรับสมัครผู้เข้าร่วมจากแผนกต่างๆ ภายใน CAE เพื่อให้แน่ใจว่ามีตัวแทนที่เหมาะสม โดยรวมแล้วเรามีผู้เข้าร่วมทั้งหมด 7 แผนก

เกณฑ์สำคัญอื่นๆ ที่ได้รับการพิจารณาในระหว่างการคัดเลือกผู้เข้าร่วม คือเราต้องการเลือกผู้เข้าร่วมจากกลุ่มเดียวกันเพื่อเข้าร่วมการศึกษา เพื่อให้แน่ใจว่าเพื่อนร่วมงานของผู้เข้าร่วมแต่ละคนจะได้เรียนรู้และใช้ PSP และด้วยเหตุนี้การสนับสนุนซึ่งกันและกัน

## 4.1.3 การออกแบบการศึกษา

หนึ่งในวัตถุประสงค์ของการศึกษานำร่องคือการประเมินประโยชน์ของการปฏิบัติตามเทคนิคของ PSP ดังนั้น ภายในการดำเนินงานที่เป็น PSP จึงเป็นการออกแบบการวิจัยเพื่อประเมินแนวคิดของ PSP

สำหรับการประเมินผลเราใช้การออกแบบการทดลองกึ่งทดลองแบบทั่วไปที่แสดงในรูปที่ 1 สัญลักษณ์ที่เราใช้เพื่ออธิบายการออกแบบมีดังต่อไปนี้: "O" หมายถึงการสังเกต / การวัดของตัวแปรตาม และ " X " แทนการแทรกแซง นอกจากนี้แกนนอนหมายถึงเวลาที่ดำเนินการ **The same design can be extended to take into account more than two interventions.**  การออกแบบนี้จะถูกนำมาใช้เพราะเราคาดหวังว่าช่วงเวลาระหว่าง และ จะเป็น สิ่งจำเป็นสำหรับผู้เข้าร่วมและกำหนดรูปแบบการเก็บข้อมูล

ในช่วงเริ่มต้นของการประเมินผล (เวลา ) ผู้เข้าร่วมได้รับการแทรกแซง

เป็นโมดูลของ PSP ที่ครอบคลุมการวัดพื้นฐานของ PSP จากนั้นผู้เข้าอบรมได้พัฒนาโปรแกรมจริงและเก็บข้อมูลที่เกี่ยวข้องของ PSP ในเวลาเดียวกัน

ภายหลังจากนั้นห้าเดือนผู้เข้าอบรมได้รับการแทรกแซงครั้งที่สอง. ขั้นตอนนี้คือการทบทวนการเขียนโค้ด เหตุผลที่เราเลือกบททบทวนการขียนโค้ดคือการที่เราต้องการให้การแทรกแซงครั้งที่สองเป็นเรื่องที่กระตุ้นเพื่อให้ผู้เข้าร่วมและผู้บริหารเลือกใช้ PSP Humphrey ระบุว่าการทบทวนการเขียนโค้ดจะนำมาซึ่งผลประโยชน์ในเชิงบวกอย่างมากต่อกระบวนการส่วนบุคคล [14].

ประเด็นหนึ่งที่ต้องพิจารณาคือแรงจูงใจของผู้เข้าร่วมทั้งหมดเป็นอาสาสมัคร คือ ไม่จำเป็นต้องเข้าร่วมหลักสูตร PSP หรือใช้ PSP เป็นส่วนหนึ่งของงาน

## 4.1.4 คำนิยามของการเขียนโปรแกรม

(programming task = งานการเขียนโปรแกรม ) ??

สำหรับแนวทางการทำงานเราต้องกำหนดงานการเขียนโปรแกรมให้เกี่ยวข้องกับ PSP พวกเราพบว่าคำนิยามสากล (universal definition) นั้นไม่สามารถเป็นไปได้กับทุกคนในงาน ดังนั้นพวกเราจึงสัมภาษณ์ผู้มีส่วนร่วมแต่ละคนถึงการศึกษาพื้นฐานและถามพวกเขาเกี่ยวกับแนวทางของพวกเขาในปัจจุบัน การสัมภาษณ์นี้ขึ้นอยู่กับแนวทางส่วนตัวของพวกเขาในปัจจุบัน เราจะระบุไว้อย่างชัดเจนสำหรับผู้เข้าสัมภาษณ์แต่ละคนว่าทำงานการเขียนโปรแกรมแบบใด (we defined for each interviewee what is a programming task) ยกตัวอย่างเช่น วิศวกรบางคนควรจะพัฒนาโมดูลที่ใช้เวลาน้อยกว่าสองสัปดาห์ (ซึ่งอาจจะทำให้ต้องแบ่งโมดูลขนาดใหญ่ให้เล็กลง) สำหรับวิศกรคนอื่นๆอาจมองว่ามันคือเปลี่ยนความต้องการ

ปัญหาหนึ่งที่เกิดขึ้นคือ “การปิดกั้น” ("blocking") ปัญหานี้เกิดขึ้นเมื่อความคืบหน้าของโปรแกรมไม่สามารถดำเนินต่อได้เนื่องจากขาดทรัพยากรที่จำเป็น บางครั้งส่งผลให้เกิดเปลี่ยนโปรแกรมกลางคันหรือบางครั้งโปรแกรมอาจจะไม่เสร็จสิ้นก่อนจบการวิจัยด้วยเหตุผลดังกล่าว

blocking เป็นปัจจัยสำคัญที่มีผลให้วิศวกรที่แตกต่างกันมีจำนวนโปรแกรมที่เสร็จสิ้นต่างกันในระหว่างการศึกษา จำนวนของโปรแกรมที่เสร็จสมบูรณ์ต่อวิศวกรอยู่ในระดับต่ำเป็น 1 และระดับสูงเป็น 10ในช่วงระยะเวลาของการศึกษา

แผนภูมิข้างต้นให้สัญลักษณ์ของกระบวนการของผู้เข้าร่วมก่อนที่จะมีการแนะนำโค้ด ซึ่งจะเป็นพื้นฐานสำหรับการเปรียบเทียบผลกระทบของการเปลี่ยนแปลงกระบวนการในอนาคต ผู้เข้าร่วมแสดงความเปลี่ยนแปลงความพยายามมากที่สุดในขั้นตอนการรวบรวมและทดสอบของเธอเมื่อเทียบกับโค้ด(เป็นเปอร์เซ็นต์ของความพยายามทั้งหมด) เธอยังคงใช้เวลาอย่างไม่มีเวลาการวางแผน เป็นที่น่าสนใจที่จะต้องทราบว่าสำหรับบางโปรแกรม มีความพยายามในการเขียนโค้ดเป็นศูนย์ เนื่องจากบางครั้งผู้เข้าร่วมทดสอบโค้ดโดยวิศวกรคนอื่นๆเท่านั้น นอกจากนี้ เป็นที่น่าสนใจว่าบางโปรแกรมผู้เข้าร่วมไม่ได้ทดสอบด้วยตัวเอง นี้จะอธิบายถึงความเปลี่ยนแปลงที่มีขนาดใหญ่ ในเปอร์เซ็นต์ความพยายามในการทดสอบ เห็นได้ชัดว่าเธอฉีดทุกข้อบกพร่องในระหว่างการออกแบบและเขียนโค้ด และลบทั้งหมดออกในระหว่างการรวบรวมและการทดสอบ ส่วนใหญ่ประเภทข้อผิดพลาดของเธอคือไวยากรณ์ อย่างไรก็ตาม มันเป็นข้อผิดพลาดด้านสภาพแวดล้อมที่ใช้เวลามากที่สุดในการแก้ไข มันจะดีขึ้นโดยรวมถ้าเธอฉีดข้อผิดพลาดประเภทสภาพแวดล้อมน้อยลง

รูปที่2 : สัญลักษณ์กระบวนการสำหรับหนึ่งในผู้เข้าร่วม

ภาพที่ 3 ประเภทของกิจกรรมที่ใช้ในการบันทึกเวลาโดยละเอียด รหัสที่มีเครื่องหมายดอกจัน (\*) ระบุถึงสิ่งที่เรากำหนดให้เป็นงานที่ไม่ใช่การเขียนโปรแกรม.

**Participant Design Code Compile Unit Test**

**1** 0.65 1.11 0.62 1.55

**2** 1.06 0.61 0.43 1.44

**3** 1.07 0.58 0.25 0.88

A Analye (วิเคราะห์): ทำการวิเคราะห์ปัญหาของโปรแกรม

D Design (ออกแบบ): รวมถึงการทำเอกสารให้เป็นปัจจุบัน

DR Design Review

CO Code

CR Code Review

CM Compile

TE Unit Test

IN Integrate (การรวบรวม)

\*B Break: การถูกขัดจังหวะจากตัวคุณเองหรือจากคนอื่น ๆ

\*CL Clean Up (ทำความสะอาด): โต๊ะทำงาน กระดาษ การจัดการไฟล์และแฟ้มข้อมูลในคอมพิวเตอร์

\*CF Configutation Management (การจัดการต่าง ๆ ): builds การจัดการไลบรารี่, file check-in/check-out

\*M Maintenance (การบำรุงรักษา): ระบบต่าง และคอมพิวเตอร์

\*ME Meeting (การประชุม)

\*MM Correspondence (การติดต่อสื่อสาร): E-mail บันทึกช่วยจำ และอื่นๆ

\*S Support (การสนับสนุน): การที่คุณให้คำแนะนำหรือช่วยเหลือคนอื่นเกี่ยวกับบางสิ่งที่คุณรู้หรือคุณสามารถทำได้

\*TR Training (การทบทวน): การสัมนา การอบรม หรือเรียนรู้ทักษะใหม่

\*TS Timesheet (ตารางเวลา): ทุกคนทำ 1 ครั้ง/สัปดาห์

\*X Interruption (การถูกขัดจังหวะ)

ภาพที่ 4 เวลาเฉลี่ยเวลาที่ไม่ถูกขัดจังหวะสำหรับโปรแกรมที่ 4 (Average uninterrupted hours for four programming tasks)

## 4.1.5 การปรับเปลี่ยนแบบฟอร์มการรวบรวมข้อมูล

การออกแบบรูปแบบการรวบรวมข้อมูลอาจเป็นข้อพิจารณาที่สำคัญที่สุดประการหนึ่งในการดำเนินการของ PSP ในขณะที่ PSP มีรูปแบบต่างๆ ปัญหาก็เกิดขึ้นเมื่อพยายามใช้สิ่งเหล่านี้ในสภาพแวดล้อมของการเขียนโปรแกรมจริง ต่อไปนี้เป็นข้อควรพิจารณาบางอย่างที่เราต้องแก้ไข:

* มีบางส่วนที่ไม่ตรงกันระหว่างโมเดลกระบวนการส่วนบุคคลตามหลัก PSP และกระบวนการของผู้เข้าร่วม ตัวอย่างเช่น ผู้เข้าร่วมบางคนทำงานบำรุงรักษาเป็นส่วนใหญ่ดังนั้นจึงต้องคำนึงถึงข้อบกพร่องที่แฝงอยู่เป็นหลัก ผู้เข้าอบรมอาจมีข้อกำหนดและขั้นตอนการวิเคราะห์ในขณะที่คนอื่นๆ สร้างต้นแบบอย่างละเอียดและเป็นขั้นตอนการสร้างต้นแบบ อีกตัวอย่างหนึ่งคือผู้เข้าร่วมในภาพที่ 2 ซึ่งในบางกรณีได้ทดสอบโปรแกรมเท่านั้น กระบวนการเหล่านี้ไม่ตรงกับรูปแบบที่สันนิษฐานไว้ใน PSP ดังนั้น เราต้องปรับแต่ง (หรือช่วยในการปรับแต่ง) รูปแบบการเก็บรวบรวมข้อมูลสำหรับผู้เข้าร่วมแต่ละคน
* หลังจากการออกแบบฟอร์มใหม่ มันเป็นสิ่งสำคัญที่จะลองพวกเขาออกกับผู้เข้าร่วมในระยะแรกเวลาเพื่อให้แน่ใจว่า จะพอใจกับรูปแบบใหม่เช่น ในกรณีหนึ่ง ผู้ใช้ในรูปแบบอย่างมีความสุขจนเขาวิ่งออกมาจากสำเนา หลังจากที่เขากลายเป็นความผิดหวังเนื่องจากแบบฟอร์มออกแบบให้มีทั้งสองด้านของแผ่นหนึ่ง และเขาไม่มีเครื่องถ่ายเอกสารสองด้านสองในย่านนี้ เราขึ้นมาออกแบบให้เหมาะสมกับด้านเดียว เป็นอีกตัวอย่าง เอกสารสับสนการดูเหมือนอันตรายระยะ "จริง" ในการวางแผนโครงการ มีคำคล้ายในฝรั่งเศส"Actuelle" หมายถึงปัจจุบัน เขาเข้าใจว่า เขามีการวางแผนโครงการที่แยกต่างหากสำหรับเก็บแต่ละวัน มากกว่าเพียงหนึ่งสำหรับทั้งหมดโปรแกรม โดย trialing ในรูปแบบ และติดตามคงไม่คาดคิดว่าความยากลำบากเช่นนี้จะระบุ และแก้ไข
* ผู้เข้าร่วมบางส่วนพบ bothersome ให้เต็มบันทึกแบบฟอร์มสำหรับข้อผิดพลาดไวยากรณ์ทุกข้อบกพร่อง(เช่น พลาดอัฒภาคหรือตัวแปรที่สะกดผิดชื่อ) เราพัฒนาแล้วรายการตรวจสอบข้อผิดพลาดไวยากรณ์ทั่วไปที่ผู้เข้าร่วมสามารถเพียงขีดเมื่อเกิดข้อผิดพลาดไวยากรณ์นี้พยายามบันทึกไวยากรณ์ที่ลดลงอย่างมากข้อผิดพลาด ตัวอย่างของฟอร์มที่ใช้ในการบันทึกการกำจัดข้อผิดพลาดไวยากรณ์ที่ฉีดในระหว่างการเขียนโค้ดถูกกำหนดในภาคผนวก สำหรับเรื่องนี้แบบฟอร์ม วิศวกรวางในเซลล์ระยะซึ่งข้อบกพร่องถูกเอาออก จะถือว่าว่า แต่ละข้อผิดพลาดไวยากรณ์ใช้เวลาหนึ่งนาทีเพื่อแก้ไข ถ้าใช้เวลานาน แล้ววิศวกรไม่สามารถเติมค่าข้อบกพร่องปกติการบันทึกแบบฟอร์ม

## 4.2 การฝึกฝน

กิจกรรมการฝึกอบรมเริ่มตั้งแต่แรก PSP บรรยายจนกว่าจะสิ้นสุดของการศึกษา จำนวนของปัญหาจะต้องได้รับการแก้ไขในการสั่งซื้อเพื่อให้แน่ใจว่าผู้เข้าร่วมดำเนินการโดยใช้แนวคิด PSP ในงานเขียนโปรแกรมของพวกเขา

-มีความจำเป็นสำหรับความคิดเห็นอย่างต่อเนื่องในการสั่งซื้อเพื่อเสริมสร้างสิ่งที่ผู้เข้าร่วมได้เรียนรู้ เราทำอย่างนี้โดยมีการประชุมเป็นประจำกับผู้เข้าร่วมในแต่ละบุคคลที่จะช่วยให้ตีความข้อมูลที่พวกเขาได้เก็บรวบรวมเพื่อติดตามความคืบหน้าของพวกเขาและที่จะตอบคำถาม ตัวอย่างของข้อมูลที่ถูกตีความในหนึ่งในการประชุมเหล่านี้จะได้รับในรูปที่ 2 LN นี้ตัวอย่างเช่นเราต้องการที่จะสร้างพื้นฐานลายเซ็นกระบวนการหนึ่งของผู้เข้าร่วม ได้เรียนรู้ในห้องเรียน ถึงแม้ว่าเราจะไม่ได้

รับการสนับสนุนของผู้บริหารในช่วงเริ่มต้นของการศึกษาที่เราไม่ได้จริงๆเกี่ยวข้องกับพวกเขาใน

การดำเนินการ วันต่อวันของมัน หนึ่งในผู้เข้าร่วมเห็นว่ามันจะดีถ้ากลุ่มของเธอเป็นผู้นำได้มาด้วยและถามว่า "PSP ไปอย่างไร ?" ในบางครั้งบางคราว

-บางผู้เข้าร่วมรู้สึกที่พวกเขาใช้เวลาน้อยในทำงานเขียนโปรแกรมและที่จริงinterrupts ใช้เวลามาก

ดังนั้น เราขอให้พวกเขารักษาเวลาโดยละเอียดโดยบันทึกที่ความละเอียด 15 นาทีเป็นการ

กำหนดเวลาของพวกเขาอยู่ใช้เวลาของผู้รักษาเวลา บันทึก เราพบว่าประมาณ 25% ของของพวกเขาใช้เวลาที่ใช้ในงาน ไม่ใช่เขียนโปรแกรม การรายการของงานประเภทที่ใช้ถูกกำหนดใน

รูปที่ 3 หลังจากนำเสนอพวกเขาเหล่านี้หมายเลข ผู้เข้าร่วมได้มากขึ้นเชื่อว่า ในความเป็นจริงใช้เวลาส่วนใหญ่บนงานเขียนโปรแกรม อย่างไรก็ตาม เราพบจำนวนติดกันเวลาที่ผู้ใช้โดยไม่ถูกขัดจังหวะเป็น

## 4.3.1 การประเมินการถ่ายโอนการเรียนรู้ในการฝึกอบรบ

โดยทั่วไปเชื่อว่ามีเพียง 10 % ของการนำผลของการฝึกอบรบมาใช้ทำงานจริงในอุตสาหกรรม US สิ่งนี้ทำให้การถ่ายโอนการเรียนรู้ในการฝึกอบรบ เป็นตัวชี้วัดสำคัญของผลลัพธ์ในการฝึก PSP

เราวัดการถ่ายโอนการเรียนรู้ในการฝึกอบรบโดยเปอเซนต์ของผู้ที่ยังเก็บข้อมูลเกี่ยวกับกระบวนการส่วนบุคลของตนโดยมีผู้เข้าเรียน 46.5% ค่านี้ถูกคำนวนหลังจาก 7 เดือน ที่มีการฝึกครั้งแรกในการใช้แบบ PSP ที่มาจากทฤษฎีของ CAE ที่เคยทำสำเร็จมาก่อน

ผลการประเมินเมื่อเทียบกับผู้ที่ฝึกอบรบแล้วคำนวณร่วมกับการเปลี่ยนแปลงในวิศวะกรรมที่ทำเป็นประจำ หลังจากอบรบเสร็จแล้ว เขาได้รับการประมาณ 45% สำหรับการทบทวนโค้ด 65% สำหรับการฝึกการจัดการ 5 เดือนต่อมาสรุปได้ว่าอัตราของเราซึ่งมากจากการทำแบบฝึกหัดโดยใช้โปรแกรมการเขียนโปรแกรมจริง หลังจาก7เดือนเป็นสิ่งที่ดี

จากการประเมินได้ตรวจสอบจากรายการมี ผู้เข้าร่วม 28 คน ที่ลงทะเบียนร่วม ภายหลังจากเริ่มมีผู้เข้าร่วม 5 คน เข้าโปรแกรมที่ไม่ได้อัพเดท 3 คน สร้างแฟ้มใหม่, 1 คน ลาคลอด และ 3 บริษัท ที่เหลือไม่ได้เข้าร่วมในโปรแกรม PSP

แนวคิดที่ได้รับการสอน ดังนั้น 57% เบื้องหลังการประเมินไม่มีการแตกต่างทางนัยสำคัญทางสถิติระหว่างผู้ใช้ทั้ง2กลุ่มหลัง 7 เดือน ผู้เข้าร่วมที่เหลืออยู่ 3 คนที่ออกไป มีการฝึกโอนย้าย 46.5% หนึ่งปัญหาส่งผลกระทบและมีปัญหาค่อนข้างมากในการฝึกขั้นพื้นฐานได้แก้ปัญหาแล้ว ใช้โปรแกรม PSP ได้จริงภายหลัง 7 เดือนคิดเป็น 81%

หลังจากเสร็จมีผู้รับรองการใช้โปรแกรมได้ดี 16 คน ที่เป็นไปตามสมมติฐานที่ตั้งไว้ของโปรแกรมประยุกต์ PSP จุดที่จะสัมพันธ์กันมาจากการคำนวณ เป็นปัจจัยการวัดความเกี่ยวข้องกันระหว่างตัวแปร dichotomous และตัวแปรที่ต่อเนื่อง ตัวแปร dichotomus ให้ผู้เข้าร่วมใช้ในรูปแบบการสอนหลัง 7 เดือน ตัวแปรต่อเนื่องได้มาจากจำนวนผู้ใช้ในแผนกเดียวกันหลังจากที่สอน 7 เดือน ถ้ามีความสัมพันธ์เป็นบวกและมีความสัมพันธ์ที่ผู้ใช้เหมือนกันในแผนกเดียวกันและใช้ PSP ในรูปแบบเดียวกัน จุดสำคัญในการดูความสัมพันธ์ 0.3 และไม่เป็นไปตามที่คาดหวังไว้ ถึงแม้ว่าจะสามารถโต้เถียงได้จากจำนวนผู้ใช้แบบเดียวกันก็ตาม ดังนั้นพวกเราใช้ตัวแปร dichotomized ที่ต่อเนื่องในผู้เข้าร่วม และในกลุ่มที่ใช้PSP อยู่แล้วเราใช้การทดสอบ Fisher exact test เป็นตัวกำหนด จากที่เคยใช้แบบทดสอบ One tailed test แล้สไม่มีความแตกต่างทางนัยสำคัญทางสถิติ ในระดับ alpha 0.1 จึงไม่สามรถรับรองได้ในกลุ่มที่ใช้รูปแบบเหมือนกันในการฝึก

## 4.3.2 การประเมินประโยชน์ของแนวคิด PSP

เราได้รับการประเมินแนวโน้มในทำซอฟต์แวร์ (ตัวชี้วัดเป็น LOC/hr ) และได้รับผลลัพธ์ที่แตกต่างกันของแต่ละคน การทดสอบ

Daniels ถูกนำมาใช้เพื่อหาแน้วโน้ม monotonic ไม่มีแนวโน้มของการมีนัยสำคัญทางสถิติ อย่างไรก็ตามค่าบวก ( เฉลี่ย 0.18 ) บ่งบอกถึง การใช้เวลาเกินในการพัฒนาซอฟต์แวร์ และ ค่าลบ (เฉลี่ย -0.41) บ่งบอกถึงการใช้เวลาน้อยของการพัฒนาซอฟต์แวร์ แต่อย่างไรก็ตามแนวโน้มข้อบกพร่องนี้แสดงให้เห็นว่ามีความสอดคล้องกับผลลัพธ์ก่อนหน้าในที่พวกเขาไม่พบการเปลี่ยนแปลง

ในระดับของการทำแบบฝึกหัด PSP

หนึ่งในตัวชี้วัดของประโยชน์ของการใช้แนวคิด PSP แนวความคิดต่างๆที่ได้ถูกนำมาใช้ในอดีตคือข้อบกพร่อง

การใช้ตัวชี้วัดนี้ สมุมติว่าข้อบกพร่องหมดไปแล้ว คุณภาพของCodeกำลังได้รับการปรับปรุง เรารู้สึกว่านั้นเป็นตัวชี้วัดที่ไม่เหมาะสม เนื่องจากมันยากที่จะเปลี่ยนแปลงค่าความหนาแน่นของข้อบกพร่อง ขณะที่เรามีเพียงข้อมูลจากการทดสอบย่อยเท่านั้น

ถ้าความหนาแน่นของข้อบกพร่องน้อยลงนั้นหมายความว่าการตรวจพบข้อบกพร่องน้อยลงหรือมมีข้อบกพร่องน้อย ตรงกันข้ามถ้าข้อบกพร่องมีมากขึ้น นั้นหมายความว่า การตรวจพบข้อบกพร่องนั้นถูกปรับปรุง

การเพิ่มขึ้นของความหนาแน่นของข้อบกพร่องเห็นได้ชัดในกรณีที่ผู้เข้าอบรบได้ทำ Code review หลังจากทำ Code review แล้วความหนาแน่นของข้อบกพร่องของผู้เข้าอบรมเพิ่มขึ้นอย่างรวดเร็ว ความหนาแน่นของข้อบกพร่องโดยเฉลี่ยนอยู่ที่ประมาณ 88 defects/KLOC ถึง 265 defects/KLOC หลังจากทำ Code review เราไม่มีสิ่งที่พิสูจน์ของความซับซ้อนปัญหาใหญ่ๆ ความซับซ้อนของโลจิค หรือ ความต่างของขนาดโปรแกรมระหว่างก่อนทำการทบทวนโค้ดและหลังทบทวนโค้ด การใช้การตีความและสมมติฐานจากการศึกษาก่อนหน้านี้ จะบ่งบอกว่าคุณภาพของโค้ดลดลง ยังไงก็ตาม เราคาดว่าประสิทธิภาพของการตรวจพบข้อบกพร่องของวิศวกร มีการปรับปรุงหลังจากทำ Code review ผลที่ได้นี้คล้ายคลึงกับความหนาแน่นของข้อบกพร่องที่เพิ่มขึ้นเล็กน้อยเมื่อทำอย่างสม่ำเสมอสำหรับหลักสูตร PSP ที่เห็นได้ชัดเจน

อีกหนึ่งตัวชี้วัดที่ขึ้นอยู่กับการทำงานใน PSP คือ Yield เป็นเปอร์เซนต์ของข้อบกพร่องของแต่ละเฟส ค่าเฉลี่ยของ Yield สำหรับการเขียนโปรแกรมโดยไม่ทำ Code review คือ 1.65% อย่างไรก็ตาม Yield ประกอบด้วยหลายโปรแกรมที่ถูกพัฒนา กับ Yield ของ 0% ถ้าเราพัฒนาโปรแกรมกับ Yield ดีกว่า 0% และค่าเฉลี่ย Yield โดยไม่ทำ Code review นั้นประมาณ 12% ค่าเฉลี่ย Yield เมื่อทำการ Code review คือ 27.7% ดังนั้น Yield มากกว่าการทำ Code review 2 ครั้ง ไม่น่าแปลกใจใช้วิธี Yield คำนวณกระนั้นก้ดี

ผู้เขียนใช้เปอร์เซนต์ของเวลสที่ใช้ในการทดสอบเพื่อประเมินการปรับปรุงจาก PSP เหตุผลก็คือมันแสดงให้เห็นถึงการปรับปรุงทักษะการกำจัดข้อบกพร่องในช่วงต้น จากข้อมูล การลดลงของค่าเฉลี่ยในเปอเซนของเวลาที่ใช้ในการทำการทดสอบ จากประมาณ 37% ก่อน Code review เป็น 17% สิ่งนี้ทำให้เห็นการปรับปรุงในการกำจัดข้อบกพร่องในช่วงต้น

เรายังดำเนินการทดสอบ post-hoc เพื่อตรวจสอบปัญหาเมื่อวิศวกรทำการทบทวนโค้ด Humphrey ตั้งข้อสังเกตว่าวิศวกรถามถึงความจำเป็นของการทำทบทวนโค้ดก่อนคอมไพล์ เราสามารถนำเสนอหลักฐานที่รับรองโดย Humphrey เกี่ยวกับความสัมพันธ์ระหว่างข้อบกพร่องตอนคอมไพล์ และ ข้อบกพร่องตอนทดสอบ เราพบคความสัมพันธ์ของ 0.69 ซึงมีนัยสำคัญที่ 0.001 สำหรับการทดสอบขณะนี้ไม่ได้หมายความถึงความสัมพันธ์เชิงสาเหตุดูเหมือนว่าข้อบกพร้องที่พบในการคอมไพล์จะเจอมากกว่าในเฟสของการtest นี่เป็นการเสริมสร้างเหตุผลสำหรับการ ทบทวนโค้ด ก่อนคอมไพล์

## 4.4 ประโยชน์

ระบุว่าหนึ่งในวัตถุประสงค์ของการศึกษา PSP คือการสร้างบรรยากาศสำหรับการกระจายแนวคิดการวัดส่วนที่เหลือขององค์กรการศึกษา PSP ความสำคัญและผลจะต้องได้รับการเลื่อนตำแหน่งในเว็บไซต์

วิธีหนึ่งที่เราทำนี้คือผ่านทางจดหมายข่าว บริษัท นอกจากนี้ยังช่วยให้ผู้เข้าร่วมในการศึกษามองเห็นได้ ที่จะรักษาความกระตือรือร้น

# 5 บทเรียนที่ได้เรียนรู้

บางส่วนของบทเรียนสำคัญที่เราได้เรียนรู้ในการศึกษาครั้งนี้มีรายละเอียดดังนี้

* ความจำเป็นในการปรับแต่งของกระบวนการส่วนบุคคลได้รับการยอมรับใน [15] ยังเป็นสิ่งสำคัญในการปรับแต่ง PSP รูปแบบการเก็บรวบรวมข้อมูลเพื่อกระบวนการส่วนบุคคลของผู้เข้าร่วมแต่ละคนและสภาพแวดล้อมการทำงาน จุดนี้ยังถูกตั้งข้อสังเกตใน [31] กระบวนการที่มีโครงสร้างสำหรับการประมวลผลข้อมูลได้รับการนำเสนอใน [25] ซึ่งสามารถใช้กระตุ้นกระบวนการส่วนบุคคล
* รูปแบบที่ทุกคนจะต้องทำกับผู้เข้าร่วมในสภาพแวดล้อมการทำงานของพวกเขาจริง แม้ว่ารูปแบบที่ได้รับการออกแบบเพื่อให้พอดีกับกระบวนการส่วนตัวของพวกเขา การใช้งานจริงในงานเขียนโปรแกรมจริงอาจแสดงให้เห็นข้อบกพร่องในการออกแบบของแบบฟอร์ม
* มันเป็นสิ่งสำคัญที่จะมีเครื่องมืออัตโนมัติที่สนับสนุนการเก็บข้อมูลผู้เข้าร่วม และการวิเคราะห์ข้อมูล นอกจากนี้ยังพบเป็นปัญหาธรรมชาติเร่งรัดของ PSP สำหรับวิศวกรมืออาชีพ [31] นอกจากนี้ มันจะดีกว่าถ้าแบบฟอร์มเก็บข้อมูลที่มีอยู่ในรูปแบบที่สามารถแก้ไขได้สำหรับผู้เข้าร่วมเพื่อให้พวกเขาสามารถกำหนดแบบฟอร์มเอง เพื่อให้พวกเขาสามารถปรับแต่งรูปแบบของตัวเองที่พวกเขาได้รับความเข้าใจที่ดีขึ้น(เช่นการลบ หรือเพิ่มกิจกรรม)
* มันจะดีกว่ามากที่จะให้หัวหน้างาน หรือผู้บริหารของผู้เข้าร่วมอย่างเป็นทางการของ PSP เพื่อให้พวกเขาเข้าใจและดูผลประโยชน์ของตน นี้จะช่วยให้ได้รับความมุ่งมั่นที่แข็งแกร่งจากการจัดการสำหรับ PSP นี้จะคล้ายกับวิธีการแบบบนลงล่าง เพื่อแนะนำ PSP ในองค์กรแนะนำใน [ 17 ] และปฏิบัติในองค์กรหนึ่ง [ 26 ]
* บรรยายควรครอบคลุมทั้งหมดของวงจรทั่วไปที่มีผลบังคับใช้ในองค์กร ไม่เพียงแต่หนึ่งในต้นฉบับ PSP ซึ่งทำให้การเรียนการสอนเกี่ยวข้องกับผู้เข้าร่วมงานการเขียนโปรแกรมจริง
* ข้อเสนอแนะเพื่อการประชุมผู้เข้าร่วมประชุมมีความสำคัญในการสั่งซื้อเพื่อ reenforce แนวคิดที่ว่าพวกเขาได้เรียนรู้ นอกจากนี้พวกเขาต้องดูข้อมูลที่พวกเขาถูกนำมาใช้ในการเก็บรวบรวมมิฉะนั้นพวกเขาอาจจะเสียดอกเบี้ย

เราได้ระบุไว้ข้างต้นจำนวนของบทเรียนที่เราเชื่อว่ามีความสำคัญในการดำเนินงานที่ PSP นอกจากนี้ยังมีที่เป็นไปได้ยืนยันบทเรียนเหล่านี้มีผลการวิจัยจากรายงานล่าสุดอื่น ๆ เกี่ยวกับการเรียนการสอน PSP

# 6 สรุปผล

**ระดับ 1 – การประเมินคุณภาพ**

ระดับของการประเมินนี้ เป็นหนึ่งในการวัดของ PSP เพื่อวัดคุณภาพของผู้เข้าร่วมการฝึกอบรม . ในเรื่องการวัดความพึงพอใจของลูกค้า โดยปกติจะใช้แบบสอบถามในการประเมินตอนส่วนท้ายของหลักสูตรการฝึกอบรม

**ระดับ 2 การประเมินผลการเรียนรู้**

ในระดับนี้เราจะวัดจากการพัฒนาและความรู้ที่เพิ่มขึ้นหลังจากที่พวกเขาได้ใช้กระบวนการของ PSP ( เช่น พวกเขามีปฏิกิริยาทางบวก หมายความว่าพวกเขาได้เรียนรู้ในระดับหนึ่ง ) การประเมินประเภทนี้มักจะทำในระหว่างการฝึกอบรมเพราะสามารถประเมินได้อย่างชัดเจน โดยการประเมินในที่นี้เราจะเปรียบเทียบสมรรถนะระหว่างก่อนและหลังอบรม

**ระดับ 3 การประเมินพฤติกรรม**

ในระดับนี้จะประเมินขอบเขตพฤติกรรมที่เกิดขึ้นหลังจากได้เรียนรู้กระบวนการของ PSP โดยสันนิษฐานว่าพวกเขาต้องเรียนรู้มิฉะนั้นพวกเขาจะไม่มีโอกาสที่จะเปลี่ยนพฤติกรรม การประเมินพฤติกรรมสามารถทำได้โดยการคำนวณสัดส่วนของผู้เข้าร่วมที่สามารถใช้แนวคิดและวิธีการ PSP มาใช้ในงานการเขียนโปรแกรมจริงของพวกเขา การประเมินนี้จะกระทำในระหว่างการเขียนโปรแกรมจริง

**ระดับที่ 4 – การประเมินผล**

ในระดับที่สี่หรือสุดท้าย จะประเมินประโยชน์ที่ได้รับเมื่อพวกเขาได้เขียนโปรแกรมจริง ( สมมติว่าพวกเขามีการเปลี่ยนแปลงพฤติกรรม) ในกรณีของการใช้แนวคิด PSP ผลลัพธ์ที่สามารถประเมินได้มีอย่างน้อยสองแบบ ตัวอย่างเช่น บุคคลหนึ่งสามารถประเมินว่าทักษะในการแก้ไขข้อบกพร่องในแต่ละกิจกรรมมีการปรับปรุงเมื่อเทียบกับก่อนที่จะใช้แนวคิด PSP ดังนั้นสำหรับโครงการนี้สามารถสรุปได้ว่า ข้อบกพร่องของงานลดน้อยลงหลังจากที่ได้ใช้ PSP เป็นแนวคิดในการทำงาน

รูปที่ 5 : ระดับการประเมินของ PSP ที่ใช้งานในการประเมินคุณภาพงาน

ในบทความนี้เราได้นำเสนอรายละเอียดของกระบวนการซอฟต์แวร์ในเรื่องแนวคิดการทำงานในสภาพแวดล้อมการเขียนโปรแกรมจริง ( ในการศึกษานี้เราวัดจากโครงสร้างและการทวนสอบ(reviews) ) จากการประเมินในองค์กร เราพบว่าเจ็ดเดือนหลังจากที่เริ่มต้นโครงการ 46.5 % ของผู้เข้าร่วมยังคงใช้แนวคิดของ PSP ในการเขียนโปรแกรมงานจริง นอกจากนี้ผู้ที่ใช้การทวนสอบ (reviews)ยังแสดงให้เห็นถึงการปรับปรุงอย่างมากในการแก้ไขข้อบกพร่องของพวกเขา

วัตถุประสงค์ของการศึกษา( PSP ) เพื่อเลือกแนวคิดที่เหมาะกับองค์กร ( B ) มีความตระหนักที่เพิ่มขึ้นของการวัดภายในองค์กร ( C ) เราประเมินว่าผู้เข้าร่วมหลายคนยังคงใช้ PSP เป็นแนวคิดโดยใช้วิธีการของเราและ ( d ) เราสามารถประเมินประโยชน์ของการใช้แนวคิด PSP ได้

นอกจากนี้เรายังได้ระบุจำนวนของปัจจัยที่ควรพิจารณาเพื่อให้แน่ใจว่าการดำเนินงานจะประสบความสำเร็จ บางทีสิ่งที่สำคัญที่สุดคือการกําหนดระดับปัญหาให้กับองค์กรและการปฏิบัติในส่วนของตัววิศวกรให้เครื่องมืออัตโนมัติสำหรับใช้งานโดยวิศวกร และได้รับความมุ่งมั่นในการจัดการและการสนับสนุนสำหรับการใช้งาน

ในขณะที่องค์กรหนึ่งจะให้แนวทางเบื้องต้นสำหรับคนอื่นๆในการดำเนินการ แต่ในส่วนของกระบวนการซอฟต์แวร์นั้นจะเน้นไปตามส่วนบุคคล หลังจากนั้นจะคอยติดตามผลเพื่อดูว่าในอนาคตจากการแนะแนวทางเบื้องต้นนั้นจะได้รับผลที่คล้ายกันเหมือนกระบวนการซอฟต์แวร์ส่วนบุคคลไหม

การส่งเสริมการใช้ PSP ในการทำงานนั้น เพื่อความสะดวกในการเปรียบเทียบของพวกเขา เราได้ดัดแปลง Kirkpatrick(คืออะไรก็ไม่รู้) กรอบการประเมินการฝึกอบรมจากระดับการประเมินง่าย ( ระดับ 1 ) ให้ได้ข้อมูลมากที่สุดและมีความซับซ้อน ( ระดับ 4 ) ซึ่งได้สรุปไว้ในรูปที่ 5