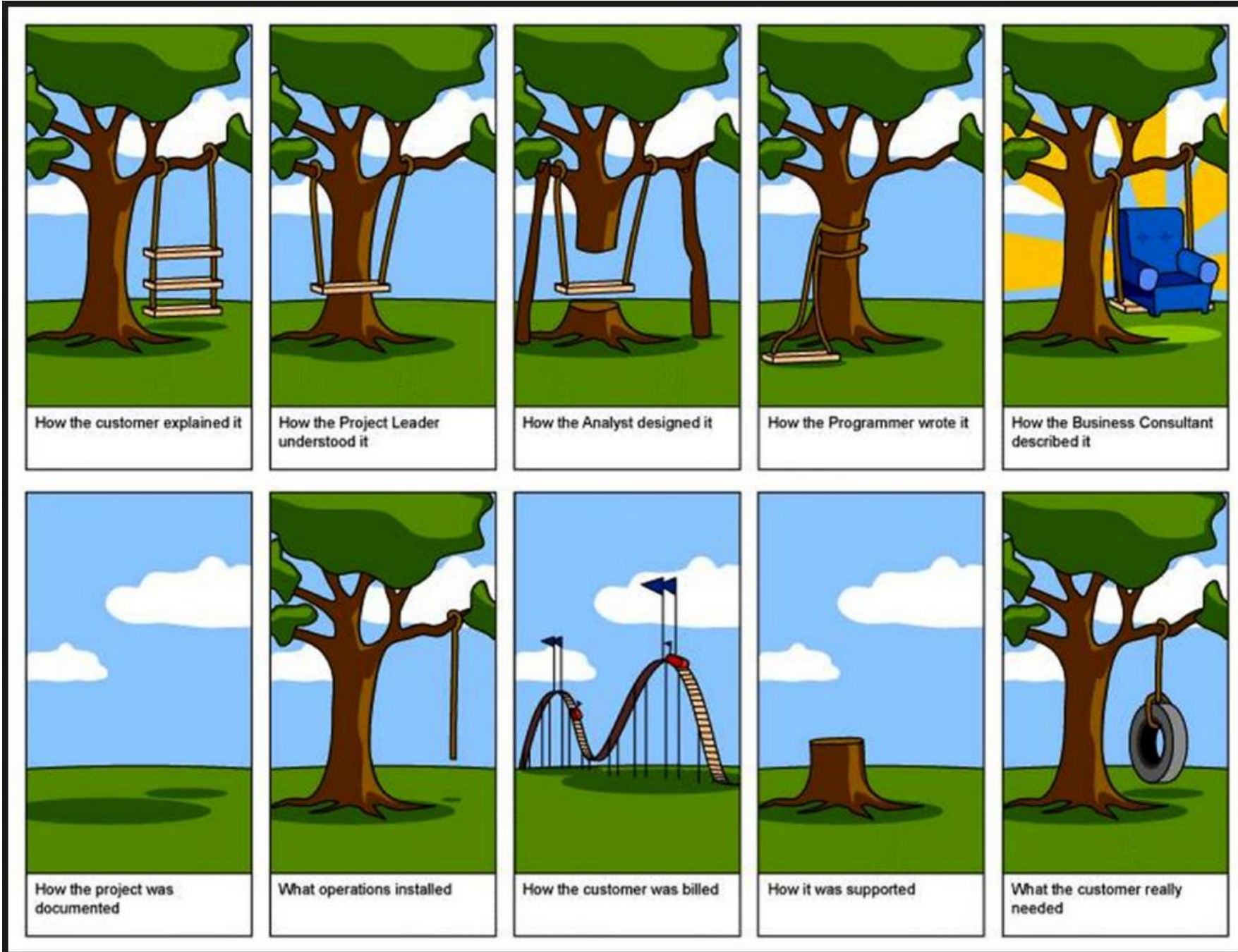


Software Project management

หัวข้อที่จะศึกษา

- 1 การบริหารโครงการซอฟต์แวร์
- 2 การบริหารความเสี่ยง
- 3 การบริหารบุคคล
- 4 การทำงานเป็นทีม

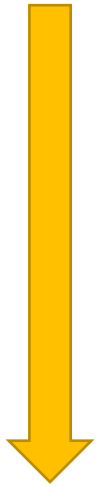


1 การบริหารโครงการซอฟต์แวร์

Software project management

- การบริหารโครงการซอฟต์แวร์ ให้ความสำคัญกับสิ่งใดบ้าง?
 - กิจกรรมที่ทำให้แน่ใจว่า สามารถส่งมอบซอฟต์แวร์ได้ตามกำหนด
 - ซอฟต์แวร์นั้นเป็นไปตามข้อกำหนดของลูกค้า
 - ความพึงพอใจ/ยอมรับซอฟต์แวร์ ทั้งจากผู้ใช้และผู้พัฒนาซอฟต์แวร์
- การบริหารโครงการซอฟต์แวร์ มีความสำคัญมาก เนื่องจาก
 - การพัฒนาซอฟต์แวร์ มีเรื่องค่าใช้จ่ายและกำหนดเวลาเข้ามาเกี่ยวข้อง
 - เป็นหน้าที่ของผู้พัฒนาซอฟต์แวร์ต้องดำเนินการ (ลูกค้าไม่รู้เรื่องด้วย)

ปัจจัยสู่ความสำเร็จในการบริหารโครงการซอฟต์แวร์



- การส่งมอบซอฟต์แวร์ ตามเวลาที่กำหนด
- การควบคุมค่าใช้จ่ายให้อยู่ในงบประมาณที่กำหนด
- การส่งมอบซอฟต์แวร์ที่ตรงตามความต้องการของผู้ใช้
- การรักษาสัมพันธภาพและบรรยากาศที่ดี ในทีมงานพัฒนาซอฟต์แวร์

ซอฟต์แวร์ที่ไม่ประสบความสำเร็จ

- ส่งมอบช้า
- ค่าใช้จ่ายเกินวงเงิน
- ไม่ตอบสนองความต้องการ
- ใช้งานไม่ได้
- มี Bug เยอะ, crash อยู่เรื่อย ๆ
- ลูกค้าไม่ยอมรับ (reject)

ความท้าทายในการบริหารโครงการซอฟต์แวร์ (1)

- ซอฟต์แวร์ ไม่มีตัวตน (intangible)
 - ซอฟต์แวร์เป็นสิ่งที่ไม่สามารถมองเห็นหรือจับต้องได้
 - ผู้บริหารโครงการไม่สามารถมองเห็นความคืบหน้าในการพัฒนา
 - ถึงจะมีเอกสารกองโตวางตรงอยู่หน้า
- ซอฟต์แวร์ส่วนใหญ่ มักจะมีเพียงหนึ่งเดียว (unique)
 - ถึงจะมีซอฟต์แวร์ที่คล้าย ๆ กันกับที่มีอยู่เดิม ก็ต้องมีส่วนที่แตกต่างและมีความยากบางอย่างอยู่ในนั้น
 - ถึงแม้ผู้บริหารโครงการจะมีประสบการณ์มาก ก็ยังต้องเผชิญปัญหาที่แก้ไขยากในโครงการใหม่เสมอ

ความท้าทายในการบริหารโครงการซอฟต์แวร์ (2)

- กระบวนการพัฒน์แวร์ มีความแตกต่างและหลากหลาย ตามชนิดและองค์กรพัฒนาซอฟต์แวร์
 - เราไม่สามารถพยากรณ์ได้ว่าจะใช้กระบวนการใดกับซอฟต์แวร์ชนิดใด (รวมทั้งใช้กับทีมงานที่มีอยู่) เพื่อให้เกิดความเสี่ยงน้อยที่สุด
 - ในบางครั้ง การนำประสบการณ์จากโครงการเดิมมาใช้ อาจสร้างจุดอ่อนใหม่ ๆ ขึ้นมาก็เป็นไปได้

ปัจจัยที่เกี่ยวข้องกับการบริหารโครงการซอฟต์แวร์

- ขนาดของบริษัทพัฒนาซอฟต์แวร์
- กลุ่มลูกค้า หรือ ลูกค้า
- ขนาดซอฟต์แวร์
- ชนิดของซอฟต์แวร์
- วัฒนธรรมองค์กร (ผู้พัฒนาซอฟต์แวร์)
- ผู้บริหารโครงการที่ประสบความสำเร็จจากองค์กรหนึ่ง อาจประสบความสำเร็จล้มเหลวในการบริหารโครงการในอีกองค์กรหนึ่ง

กิจกรรมสากลที่ใช้ในการบริหารโครงการซอฟต์แวร์

- Project planning
 - ประมาณการงบประมาณ, วางแผนงาน, จัดกำลังคน
- Risk management
 - ประเมินความเสี่ยง, สอดส่องดูแล, รับมือและแก้ไขปัญหา
- People management
 - เลือกคนและจัดทีม, หาแนวทางที่ก่อให้เกิดประสิทธิภาพสูงสุด

เครื่องมือที่ใช้ในการบริหารโครงการซอฟต์แวร์

- Reporting (การเขียนรายงาน)
 - ผู้บริหารโครงการ มีหน้าที่เขียนรายงานความก้าวหน้า
 - เพื่อเสนอต่อ ลูกค้า และ บริหารของบริษัท
- Proposal writing
 - ทุกโครงการ ต้องเริ่มจากการเขียน proposal (เอกสารข้อเสนอ) เพื่อใช้เป็นภาพร่างของงาน
 - Proposal ไม่ใช่เอกสารที่ได้จากการออกแบบซอฟต์แวร์

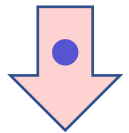
2 การบริหารความเสี่ยง

การบริหารความเสี่ยง

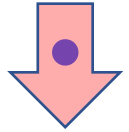
- เริ่มจากการค้นหาความเสี่ยง และวางแผนการรับมือ เพื่อให้ส่งผลกระทบต่อโครงการน้อยที่สุด
- การสร้างซอฟต์แวร์ มีความเสี่ยงเสมอ อาจมีสาเหตุมาจาก
 - การเปลี่ยนแปลง requirement, การกำหนด requirement ที่ไม่รัดกุม, การประเมินความสามารถของทีมงานที่ไม่ตรงตามความจริง ฯลฯ
- หน้าที่ของผู้บริหารโครงการคือ
 - ศึกษาและทำความเข้าใจผลกระทบของความเสี่ยงต่อโครงการ
 - วางแผนการหรือกำหนดขั้นตอน เพื่อหลีกเลี่ยงความเสี่ยงให้มากที่สุด

การจำแนกความเสี่ยง

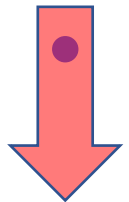
- เราสามารถจำแนกความเสี่ยงได้เป็น 2 มิติ
 - ชนิดของความเสี่ยง (ทางเทคนิค, ทางการบริหาร, ...) “มองที่เหตุ”
 - ผลกระทบจากความเสี่ยง “มองที่ผล”



ความเสี่ยงอาจกระทบต่อตารางเวลาหรือทรัพยากร



ความเสี่ยงอาจกระทบต่อคุณภาพหรือประสิทธิภาพของซอฟต์แวร์



ความเสี่ยงจากปัจจัยภายนอก เช่นจากผลดำเนินการทางธุรกิจ ก็อาจจะส่งผลกระทบต่อทั้งโครงการ

ตัวอย่างความเสี่ยงชนิดต่างๆ

ความเสี่ยง	ผลกระทบ	รายละเอียด
พนักงานเปลี่ยนงาน	โครงการ	พนักงานที่มีประสบการณ์อาจจะลาออกก่อนโครงการจะสำเร็จ
การเปลี่ยนแปลงการจัดการ	โครงการ	การเปลี่ยนแปลงการจัดการในระดับต่าง ๆ อาจจะส่งผลต่อโครงการที่ร้ายแรงแตกต่างกัน
ไม่สามารถจัดหา hardware	โครงการ	อาจจะไม่สามารถจัดหา hardware ที่เป็นส่วนสำคัญต่อการทำงานของระบบได้ทันเวลา
มีการเปลี่ยน requirement	โครงการและผลผลิต	อาจจะมีการเปลี่ยนแปลงของ requirement มากกว่าคิดกันไว้

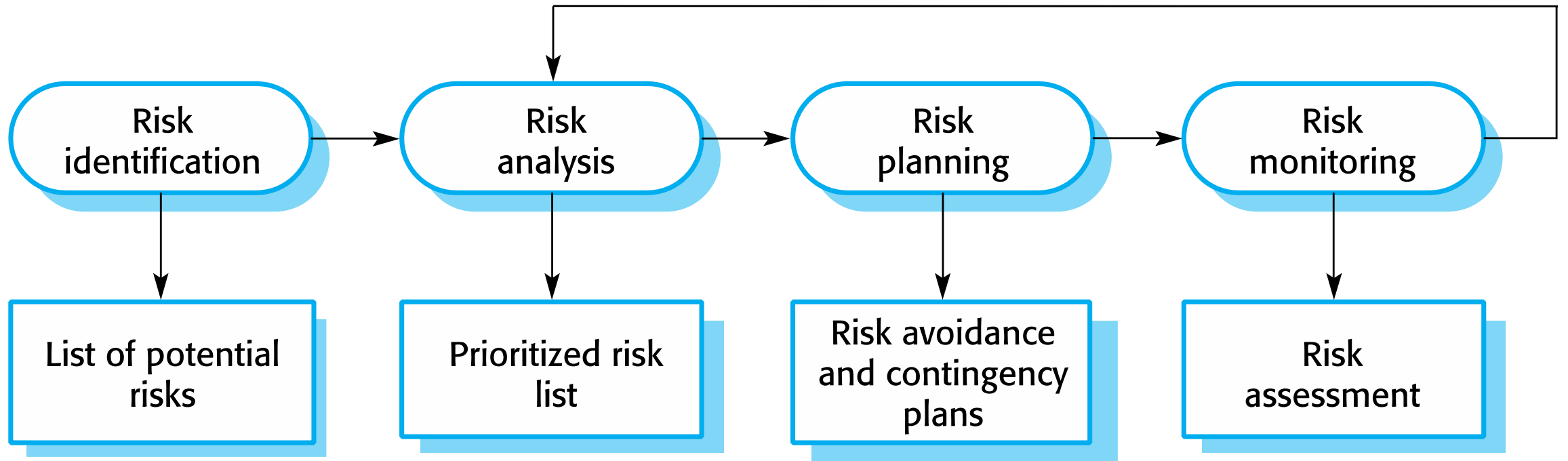
ตัวอย่างความเสี่ยงชนิดต่างๆ

ความเสี่ยง	ผลกระทบ	รายละเอียด
ออกข้อกำหนดล่าช้า	โครงการและผลผลิต	ข้อกำหนดที่สำคัญเกี่ยวกับการประสานงานของส่วนต่างๆ ในโครงการ ออกมาล่าช้า
ไม่สามารถประเมินขนาดที่แท้จริงของโครงการ	โครงการและผลผลิต	ผู้บริหารโครงการที่ประสบการณ์น้อย อาจจะไม่สามารถประเมินขนาดที่แท้จริงของโครงการ
เครื่องมือไม่มีประสิทธิภาพ	ผลผลิต	เครื่องมือต่างๆ ที่ใช้ในการพัฒนาระบบ ด้อยประสิทธิภาพ
เทคโนโลยีเปลี่ยน	ธุรกิจ	เทคโนโลยีที่ใช้ในโครงการถูกแทนที่ด้วยเทคโนโลยีที่ใหม่กว่า
ผลิตภัณฑ์จากคู่แข่ง	ธุรกิจ	คู่แข่งออกผลิตภัณฑ์คล้ายกันออกมาวางตลาดก่อนงานของเราเสร็จ

กระบวนการบริหารความเสี่ยง

1. การระบุความเสี่ยง (Risk identification)
 - จำแนกออกมาเป็นความเสี่ยงระดับโครงการ (project) ระดับผลิตภัณฑ์ (product) หรือระดับธุรกิจ (business)
2. การวิเคราะห์ความเสี่ยง (Risk analysis)
 - ประเมินความเป็นไปได้และผลกระทบของความเสี่ยงดังกล่าว
3. การวางแผนความเสี่ยง (Risk planning)
 - จัดทำแผนงานเพื่อหลีกเลี่ยงหรือลดผลกระทบของความเสี่ยง
4. การติดตามความเสี่ยง (Risk monitoring)
 - ติดตามความเสี่ยงตลอดโครงการ

กระบวนการบริหารความเสี่ยง



1.การระบุความเสี่ยง (Risk identification)

- อาจจะเป็นกิจกรรมระดับทีม หรือหัวหน้าทีมทำโดยลำพังก็ได้
- อาจใช้รายการตรวจสอบความเสี่ยงทั่วไป เพื่อระบุความเสี่ยงในโครงการ เช่น
 - ความเสี่ยงจากการเปลี่ยนแปลงเทคโนโลยี (Technology risks)
 - ความเสี่ยงจากองค์กร (Organizational risks)
 - ความเสี่ยงจากผู้คน (People risks)
 - ความเสี่ยงจากความต้องการที่เปลี่ยนแปลงไป (Requirements risks)
 - ความเสี่ยงจากการประมาณที่ผิดพลาด (Estimation risks)

ตัวอย่างของประเภทความเสี่ยงชนิดต่าง ๆ

ประเภทความเสี่ยง	ความเสี่ยงที่เป็นไปได้
การเปลี่ยนแปลงเทคโนโลยี	<ul style="list-style-type: none">• ฐานข้อมูลที่ใช้ในระบบไม่สามารถประมวลผลการทำธุรกรรมเป็นจำนวนมากในเวลาต่อวินาทีตามที่คาดไว้• ส่วนประกอบซอฟต์แวร์ที่นำมา reuse มีข้อบกพร่อง ไม่สามารถใช้ได้ตามที่วางแผนไว้
จากผู้คน	<ul style="list-style-type: none">• พนักงานที่มีทักษะที่จำเป็นหาได้ยากหรือไม่มีเลย• พนักงานป่วย โดยเฉพาะคนที่สำคัญต่องาน หรือไม่สามารถใช้งานได้ในช่วงเวลาที่สำคัญเร่งด่วน• ไม่สามารถฝึกอบรมพนักงานเพื่อเพิ่มทักษะได้
จากองค์กร	<ul style="list-style-type: none">• มีการปรับโครงสร้างองค์กรใหม่ ทำให้ผู้บริหารรายอื่นมารับผิดชอบโครงการ• ปัญหาทางการเงินขององค์กร บังคับให้มีการลดงบประมาณโครงการ

ตัวอย่างของประเภทความเสี่ยงชนิดต่าง ๆ

ประเภทความเสี่ยง	ความเสี่ยงที่เป็นไปได้
จากเครื่องมือ	<ul style="list-style-type: none">• code ที่สร้างโดยเครื่องมืออัตโนมัติสำหรับสร้างรหัสซอฟต์แวร์ ใช้ไม่ได้ผล• เครื่องมือต่าง ๆ ในการพัฒนาซอฟต์แวร์ไม่สามารถทำงานร่วมกันได้แบบครบวงจร
จากความต้องการที่เปลี่ยนแปลงไป	<ul style="list-style-type: none">• มีการเปลี่ยนแปลงข้อกำหนด ที่ทำให้ต้องออกแบบส่วนหลักของระบบใหม่• ลูกค้าไม่เข้าใจถึงผลกระทบจากการเปลี่ยนแปลงข้อกำหนด
การประมาณที่ผิดพลาด	<ul style="list-style-type: none">• ประเมินเวลาในการพัฒนาซอฟต์แวร์ต่ำเกินไป• ประเมินอัตราการซ่อมแซมข้อบกพร่องต่ำเกินไป• ประเมินขนาดของซอฟต์แวร์ต่ำเกินไป

2.การวิเคราะห์ความเสี่ยง (Risk analysis)

- ทำการประเมินความเป็นไปได้ และความรุนแรงของความเสี่ยงแต่ละชนิด
- ความเป็นไปได้
 - น้อยมาก (very low), น้อย (low), ปานกลาง (moderate), สูง (high) และ สูงมาก (very high)
- ความรุนแรง
 - ไม่สำคัญ (insignificant), พอประมาณ (tolerable), จริงจัง (serious) และ ร้ายแรง (catastrophic)

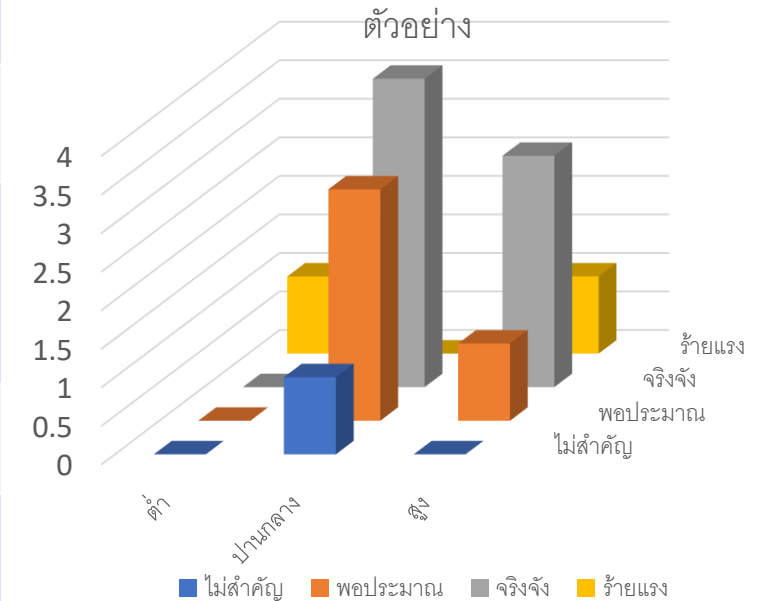
ตัวอย่าง

ความเสี่ยง	ความเป็นไปได้	ความรุนแรง
1 องค์กรประสบปัญหาทางการเงิน จึงบังคับให้ลดงบประมาณโครงการ	ต่ำ	ร้ายแรง
2 หาพนักงานที่มีทักษะที่จำเป็นได้ยากหรือเป็นไปไม่ได้	สูง	ร้ายแรง
3 พนักงานคนสำคัญ ป่วยในช่วงที่สำคัญของการพัฒนาโครงการ	ปานกลาง	จริงจัง
4 ความผิดพลาดในส่วน reuse ซึ่งต้องได้รับการซ่อมแซมก่อนนำกลับมาใช้ใหม่	ปานกลาง	จริงจัง
5 มีการเปลี่ยนแปลงข้อกำหนด ที่ทำให้ต้องออกแบบใหม่	ปานกลาง	จริงจัง
6 องค์กรมีการปรับโครงสร้างใหม่ เพื่อให้ผู้บริหารคนอื่นมารับผิดชอบโครงการ	สูง	จริงจัง
7 ฐานข้อมูลที่ใช้ในระบบไม่สามารถประมวลผลการทำธุรกรรมเป็นจำนวนมาก ในเวลาต่อวินาทีตามที่คาดไว้	ปานกลาง	จริงจัง
8 ระยะเวลาในการพัฒนาซอฟต์แวร์ต่ำเกินจริง	สูง	จริงจัง

ตัวอย่าง

ความเสี่ยง	ความเป็นไปได้	ความรุนแรง
9 เครื่องมือต่าง ๆ ในการพัฒนาซอฟต์แวร์ไม่สามารถทำงานร่วมกันได้แบบครบวงจร	สูง	จริงจัง
10 ลูกค้าไม่เข้าใจถึงผลกระทบจากการเปลี่ยนแปลงข้อกำหนด	ปานกลาง	พอประมาณ
11 ไม่สามารถฝึกอบรมพนักงานเพื่อเพิ่มทักษะได้	ปานกลาง	พอประมาณ
12 ประมาณอัตราการซ่อมแซมข้อบกพร่องต่ำเกินไป	ปานกลาง	พอประมาณ
13 ประมาณขนาดของซอฟต์แวร์ต่ำเกินไป	สูง	พอประมาณ
14 code ที่สร้างโดยเครื่องมืออัตโนมัติสำหรับสร้างรหัสซอฟต์แวร์ ใช้ไม่ได้ผล	ปานกลาง	ไม่สำคัญ

	ไม่สำคัญ (insignificant),	พอประมาณ (tolerable),	จริงจัง (serious)	ร้ายแรง (catastrophic)
สูงมาก (very high)				
สูง (high)		13	6, 8, 9	2
ปานกลาง (moderate)	14	10, 11, 12	3, 4, 5, 7	
ต่ำ (low)				1
น้อยมาก (very low)				



3.การวางแผนความเสี่ยง (Risk planning)

- พิจารณาแต่ละความเสี่ยงและพัฒนากลยุทธ์เพื่อจัดการความเสี่ยงดังกล่าว
- กลยุทธ์การหลีกเลี่ยงความเสี่ยง (Avoidance strategies)
 - ความเป็นไปได้ที่จะทำให้เกิดความเสี่ยงลดลง
- กลยุทธ์ลดผลกระทบจากความเสียหาย (Minimization strategies)
 - ผลกระทบของความเสี่ยงต่อโครงการหรือผลิตภัณฑ์จะลดลง
- แผนฉุกเฉิน (Contingency plans)
 - หากความเสี่ยงเกิดขึ้น แผนฉุกเฉินคือแผนการจัดการกับความเสียหายนั้น

จะเกิดอะไรขึ้น

- จะเกิดอะไรขึ้นถ้าวิศวกรหลายคนป่วยในเวลาเดียวกัน
- จะเกิดอะไรขึ้นถ้าภาวะเศรษฐกิจตกต่ำทำให้งบประมาณลดลง 20% สำหรับโครงการ
- จะเกิดอะไรขึ้นถ้าประสิทธิภาพของซอฟต์แวร์โอเพนซอร์สไม่เพียงพอและผู้เชี่ยวชาญเฉพาะด้านซอฟต์แวร์โอเพนซอร์สคิดจะลาออก
- จะเกิดอะไรขึ้นถ้าบริษัทที่ให้บริการและดูแลรักษาส่วนประกอบซอฟต์แวร์เล็กกิจการ
- เกิดอะไรขึ้นถ้าลูกค้าไม่สามารถออกข้อกำหนดตามระยะเวลาที่คาดการณ์ไว้

กลยุทธ์สำหรับบริหารความเสี่ยง

ความเสี่ยง	กลยุทธ์
ปัญหาทางการเงินขององค์กร	<ul style="list-style-type: none">จัดเตรียมเอกสารสรุปสำหรับผู้บริหารระดับสูง เพื่อแสดงว่าโครงการมีส่วนสำคัญอย่างยิ่งต่อเป้าหมายของธุรกิจนำเสนอเหตุผลที่แสดงให้เห็นว่าการลดงบประมาณโครงการจะไม่คุ้มค่า
ปัญหาการสรรหาบุคลากร	<ul style="list-style-type: none">แจ้งเตือนลูกค้าถึงปัญหาที่อาจเกิดขึ้นและความเป็นไปได้ที่จะเกิดความล่าช้าตรวจสอบส่วนประกอบสำเร็จรูปที่ต้องนำมาใช้ในกรณีที่ไม่ต้องการจ้างคน
ความเจ็บป่วยของพนักงาน	<ul style="list-style-type: none">จัดระเบียบทีมใหม่เพื่อให้มีการซ้อนทับกันมากขึ้นในการทำงานมอบหมายงานให้พนักงานแต่ละคนเข้าใจในงานของกันและกัน
ส่วนประกอบที่บกพร่อง	<ul style="list-style-type: none">เปลี่ยนส่วนประกอบที่อาจเป็นข้อบกพร่อง ด้วยส่วนประกอบที่ซื้อมาจากแหล่งที่น่าเชื่อถือ

กลยุทธ์สำหรับบริหารความเสี่ยง

ความเสี่ยง	กลยุทธ์
มีการเปลี่ยนแปลงความต้องการ	<ul style="list-style-type: none">• ตรวจสอบย้อนกลับเพื่อประเมินผลกระทบที่เกิดจากการเปลี่ยนแปลงความต้องการ• ออกแบบระบบให้พร้อมรับการเปลี่ยนแปลงให้มากที่สุด
การปรับโครงสร้างองค์กร	<ul style="list-style-type: none">• จัดทำเอกสารสรุปสำหรับผู้บริหารระดับสูง เพื่อแสดงให้เห็นว่าโครงการมีส่วนสำคัญต่อเป้าหมายของธุรกิจอย่างไร
ประสิทธิภาพของฐานข้อมูล	<ul style="list-style-type: none">• ตรวจสอบความเป็นไปได้ในการซื้อฐานข้อมูลที่มีประสิทธิภาพสูงขึ้น
การประเมินเวลาในการพัฒนาต่ำกว่าความเป็นจริง	<ul style="list-style-type: none">• ตรวจสอบส่วนประกอบที่อาจจะเข้ามาเสริม• จัดหาเครื่องมือในการสร้าง code สำหรับโปรแกรมโดยอัตโนมัติ

4.การติดตามความเสี่ยง (Risk monitoring)

- ประเมินความเสี่ยงแต่ละอย่างที่ระบุไว้เป็นประจำ
 - เพื่อตัดสินใจว่าจะมีความเป็นไปได้น้อยหรือมากน้อยเพียงใด
- ประเมินว่าผลกระทบของความเสี่ยงมีการเปลี่ยนแปลงหรือไม่
- ความเสี่ยงหลักแต่ละข้อควรได้รับการกล่าวถึงในที่ประชุมติดตามความคืบหน้า

ตัวชี้วัดความเสี่ยง

ประเภทความเสี่ยง	ตัวชี้วัดที่เป็นไปได้
การประมาณการที่ผิดพลาด	<ul style="list-style-type: none">• ไม่ปฏิบัติตามกำหนดการที่ตกลงกันไว้• ความไม่ชัดเจนในการรายงานข้อบกพร่อง
องค์กร	<ul style="list-style-type: none">• การนิินหาของพนักงานในองค์กร• การขาดการดำเนินการโดยผู้บริหารระดับสูง
ผู้คน	<ul style="list-style-type: none">• ขวัญกำลังใจของบุคลากรที่แย่• ความสัมพันธ์ระหว่างสมาชิกในทีมที่แย่• การหมุนเวียนพนักงานสูง
ความต้องการที่เปลี่ยนแปลง	<ul style="list-style-type: none">• คำขอเปลี่ยนความต้องการจำนวนมาก• ข้อร้องเรียนของลูกค้า.

ตัวชี้วัดความเสี่ยง

ประเภทความเสี่ยง	ตัวชี้วัดที่เป็นไปได้
เทคโนโลยี	<ul style="list-style-type: none">• การส่งมอบฮาร์ดแวร์หรือซอฟต์แวร์สนับสนุนล่าช้า• รายงานปัญหาเทคโนโลยี (ที่ใช้ในโครงการ) ที่เกิดขึ้นบ่อยครั้ง
เครื่องมือ	<ul style="list-style-type: none">• ความลังเลใจของสมาชิกในทีมที่จะใช้เครื่องมือใหม่ๆ• รายงานปัญหาเกี่ยวกับเครื่องมือ CASE (Computer-Aided Software Engineering) การบริหารความเสี่ยง (Risk management)• ความต้องการสำหรับเวิร์คสเตชันที่ประสิทธิภาพสูง

3 การบริหารบุคคล

การบริหารบุคคล (Managing people)

- คนเป็นสินทรัพย์ที่สำคัญที่สุดขององค์กร
- งานหลักของผู้บริหารจะทุ่ม่งเน้นคน
 - การขาดความเข้าใจเกี่ยวกับคน จะทำให้การบริหารไม่ประสบความสำเร็จ
- การจัดการทรัพยากรมนุษย์ที่ไม่มีประสิทธิภาพ เป็นส่วนสำคัญในการทำให้โครงการล้มเหลว

ปัจจัยการบริหารคน

- ความมั่นคงทางอารมณ์
 - สมาชิกในทีมทุกคนควรได้รับการปฏิบัติอย่างเท่าเทียมกัน โดยไม่มีคนโปรด ไม่มีการเลือกปฏิบัติ
- การเคารพความแตกต่าง
 - สมาชิกในทีมต่างมีทักษะที่แตกต่างกัน และทุกคนต้องยอมรับความแตกต่างเหล่านี้
- ความสามัคคี
 - มีส่วนร่วมกับสมาชิกในทีมทั้งหมดและตรวจสอบให้แน่ใจว่ามุมมองของทุกคนได้รับการพิจารณาอย่างสำคัญเท่าเทียมกัน
- ความสุจริต
 - ต้องมีความซื่อสัตย์กับทุกสิ่งที่เกิดขึ้น ทั้งสิ่งที่ดีและไม่ดีในโครงการ

การสร้างแรงจูงใจคน

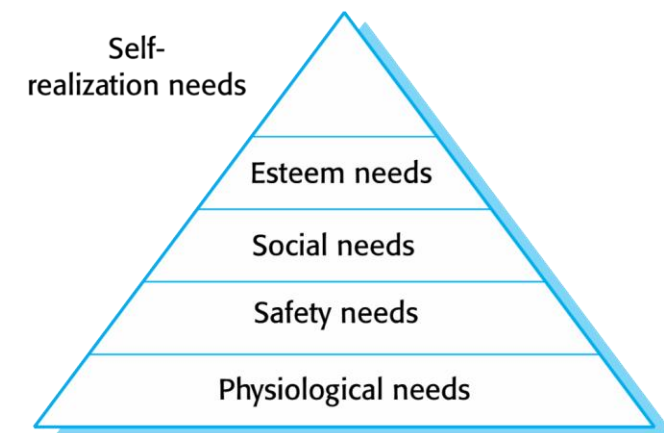
- บทบาทที่สำคัญของผู้จัดการคือ การกระตุ้นทุกคนที่ทำงานในโครงการ
- แรงจูงใจคือ การจัดงานและสภาพแวดล้อมในการทำงาน เพื่อกระตุ้นให้คนทำงานได้อย่างมีประสิทธิภาพ
 - ถ้าคนไม่ได้รับแรงกระตุ้นพวกเขาจะไม่สนใจงานที่พวกเขากำลังทำอยู่
 - พวกเขาจะทำงานช้า
 - มีแนวโน้มที่จะทำผิดพลาดและจะไม่นำไปสู่เป้าหมายที่ดีขึ้นของทีมหรือองค์กร
- การสร้างแรงจูงใจเป็นเรื่องซับซ้อน แต่สามารถแบ่งเป็นประเภทต่าง ๆ เช่น
 - ความต้องการขั้นพื้นฐาน (เช่น อาหาร การนอนหลับ ฯลฯ);
 - ความต้องการส่วนบุคคล (เช่น เคารพนับถือตนเอง);
 - ความต้องการทางสังคม (เช่น ได้รับการยอมรับเป็นส่วนหนึ่งของกลุ่ม)

ลำดับความต้องการของมนุษย์



คน - ต้องการความพึงพอใจ

- ในการพัฒนาซอฟต์แวร์ ความต้องการทางสรีรวิทยาและความปลอดภัยขั้นพื้นฐาน ไม่ได้ เป็นปัญหา
- ความต้องการทางสังคม
 - มีสิ่งอำนวยความสะดวกส่วนกลาง
 - อนุญาตการสื่อสารแบบไม่เป็นทางการ เช่น ผ่านทางเครือข่ายทางสังคม
- ความภาคภูมิใจในตนเอง
 - ต้องการให้ผู้อื่นรับรู้ความสำเร็จของตน (ต้องการคนเห็นความสำคัญ)
 - ต้องการรางวัลที่เหมาะสม
- ความต้องการในตนเอง
 - ต้องการการฝึกอบรม - คนมักต้องการเรียนรู้เพิ่มเติม
 - ความรับผิดชอบ - อยากได้งานที่มีความรับผิดชอบสูงขึ้น



ประเภทบุคลิกภาพ

- โดยทั่วไป เราสามารถใช้ลำดับความต้องการพื้นฐาน เพื่อสร้างแรงจูงใจ แต่ในทางปฏิบัติ คนมีบุคลิกภาพที่แตกต่างกัน
- การสร้างแรงจูงใจควรคำนึงถึงบุคลิกภาพที่แตกต่างกัน:
 - คนที่มุ่งเน้นงาน (Task-oriented people)
 - มีแรงบันดาลใจจากความสำเร็จของงานที่พวกเขาทำ
 - คนที่มุ่งเน้นปฏิสัมพันธ์ (Interaction-oriented people)
 - มีแรงจูงใจจากการแสดงออกและการกระทำของเพื่อนร่วมงาน
 - คนที่มุ่งมั่นในตนเอง (Self-oriented people)
 - มีแรงบันดาลใจจากความสำเร็จส่วนบุคคลและการเป็นที่รู้จัก

วิธีการสร้างแรงจูงใจในแต่ละประเภทบุคลิกภาพ

- คนที่มุ่งเน้นงาน (Task-oriented people)
 - แรงจูงใจในการทำงาน คือการมีผลงานเป็นของตัวเอง
- คนที่มุ่งเน้นปฏิสัมพันธ์ (Interaction-oriented people)
 - แรงจูงใจหลักคือการแสดงออกและการกระทำของเพื่อนร่วมงาน
 - คนพวกนี้ไปทำงานเพราะชอบไปที่ทำงาน (ไปเจอเพื่อน ๆ)
- คนที่มุ่งมั่นในตนเอง (Self-oriented people)
 - ความสำเร็จในการทำงาน หมายถึงการบรรลุความต้องการส่วนบุคคล เช่น ได้ รวยขึ้น ได้เล่นกีฬาที่ชอบ ได้เดินทาง ฯลฯ

การสร้างสมดุลของแรงจูงใจในการทำงาน

- แรงจูงใจส่วนบุคคลประกอบด้วยองค์ประกอบทุก ๆ ชั้นที่กล่าวมา
- ความสมดุลในแต่ละชั้นสามารถเปลี่ยนแปลงได้ ขึ้นอยู่กับสถานการณ์ส่วนบุคคลและเหตุการณ์ภายนอก
- แรงบันดาลใจ ไม่ได้เกิดจากปัจจัยด้านบุคคลเพียงอย่างเดียว แต่ยังเกิดจากการได้เข้ากลุ่มและวัฒนธรรมในสังคมอีกด้วย
 - หัวหน้างานเพียงคนเดียว ไม่อาจสร้างแรงจูงใจได้เท่ากับทุกคนในองค์กร
- คนไปทำงานเพราะพวกเขามีแรงจูงใจจากคนที่พวกเขาร่วมงานด้วย

4 การทำงานเป็นทีม

การทำงานเป็นทีม

- วิศวกรรมซอฟต์แวร์ส่วนใหญ่เป็นกิจกรรมที่ทำเป็นทีม
 - ข้อกำหนดการพัฒนาโครงการซอฟต์แวร์ส่วนใหญ่จะไม่สามารถดำเนินการได้โดยลำพัง
- ทีมที่ดีจะเหนียวแน่นและมีจิตวิญญาณของทีม
 - คนในทีมมีแรงบันดาลใจจากความสำเร็จของทีม เช่นเดียวกับเป้าหมายส่วนบุคคล
- ปฏิสัมพันธ์ของทีมเป็นปัจจัยสำคัญในความสำเร็จของการปฏิบัติงานของทีม
- ความยืดหยุ่นในการจัดทีม มีจำกัด
 - ผู้จัดการต้องทำอย่างดีที่สุดเท่าที่จะทำได้กับทุกคนที่อยู่ในทีม

ความเหนียวแน่นของทีม

- ในทีมที่เหนียวแน่น สมาชิกจะเห็นว่า “ทีม” มีความสำคัญมากกว่า “บุคคล” เสมอ (ต่อให้คนนั้นคือ หัวหน้า ก็ตาม)
- ข้อได้เปรียบของทีมที่เหนียวแน่น ได้แก่
 - ทีมสามารถพัฒนามาตรฐานคุณภาพของทีมได้
 - สมาชิกในทีมเรียนรู้จากกันและกัน และทำความรู้จักงานของกันและกัน
 - การยับยั้งสิ่งแย่ๆ ที่เกิดจากความไม่รู้จะทำได้ดีขึ้น
 - แบ่งปันความรู้ - สามารถรักษาความต่อเนื่องได้หากสมาชิกในทีมออกจากโครงการ
 - มีการเปลี่ยนแปลงและปรับปรุงอย่างต่อเนื่อง
 - สมาชิกในกลุ่มทำงานร่วมกันเพื่อให้ได้ผลลัพธ์ที่มีคุณภาพ
 - แก้ไขปัญหาโดยไม่คำนึงถึงบุคคลที่สร้างสรรค์หรือออกแบบโปรแกรม

ประสิทธิภาพของทีม

- ทีมที่ประกอบด้วยสมาชิกที่มีแรงจูงใจเดียวกัน ก็อาจเป็นปัญหาได้
 - คนที่มุ่งเนื้องาน - ทุกคนต้องการที่จะทำสิ่งที่ตัวเองอยากได้ผลงาน
 - คนที่มุ่งมั่นในตนเอง - ทุกคนต้องการที่จะเป็นเจ้านาย
 - คนที่มุ่งเน้นปฏิสัมพันธ์ - การเซทมากเกินไป ไม่ทำงานเพียงพอ
- ทีมที่มีประสิทธิภาพสูงสุด จะเกิดขึ้นเมื่อมีความสมดุลของคนทุกประเภท
- ทีมที่สมบูรณ์แบบอาจเป็นเรื่องยากที่จะทำได้
 - เนื่องจากวิศวกรซอฟต์แวร์มักจะ บ้างงาน (task-oriented)
- คนที่มุ่งเน้นปฏิสัมพันธ์ จะมีความสำคัญต่อองค์กรมาก
 - สามารถสัมผัสและขจัดความตึงเครียดที่เกิดขึ้นได้

การจัดทีม

- วิธีการจัดทีมมีผลต่อ :
 - การตัดสินใจของกลุ่ม
 - วิธีการแลกเปลี่ยนข้อมูลภายในทีม
 - การมีปฏิสัมพันธ์ระหว่างทีมและผู้มีส่วนได้เสียของโครงการ
- คำถามที่สำคัญ ได้แก่ :
 - ผู้จัดการโครงการควรเป็นผู้นำด้านเทคนิคของทีมนี้หรือไม่?
 - ใครจะมีส่วนร่วมในการตัดสินใจทางเทคนิคที่สำคัญและจะทำอย่างไร?
 - จะมีการสื่อสารระหว่างผู้มีส่วนได้เสียภายนอกและผู้บริหารอาวุโสของบริษัทอย่างไร?
 - ทีมสามารถรวมกลุ่มคนที่ไม่ได้เก่งอย่างเดียวกันได้อย่างไร?
 - สามารถแชร์ความรู้กันได้อย่างไรในทีม?

การจัดทีม

- ทีมวิศวกรรมซอฟต์แวร์ขนาดเล็กมักตั้งขึ้นอย่างไม่เป็นทางการ
 - ไม่มีโครงสร้างที่เข้มงวด
- โครงการขนาดใหญ่อาจมีโครงสร้างลำดับชั้น
 - แล้วตั้งทีมย่อย ๆ เพื่อทำหน้าที่รับผิดชอบโครงการย่อยต่าง ๆ
- การพัฒนาแบบ agile มักจะเป็นทีมที่ไม่เป็นทางการ
 - โครงสร้างที่เป็นทางการมักจะเป็นอุปสรรคต่อการแลกเปลี่ยนข้อมูล

ทีมที่ไม่เป็นทางการ

- ทุกคนในทีมมีอำนาจในการตัดสินใจที่มีผลต่อระบบ
- หัวหน้าทีมทำหน้าที่เป็นผู้ติดต่อภายนอกทีม
- งานทุกอย่างจะทำในนามของทีม โดยมีการจัดสรรงานตามความสามารถและประสบการณ์ของสมาชิกแต่ละคน
- วิธีนี้ประสบความสำเร็จสำหรับทีมที่สมาชิกทุกคนมีประสบการณ์และความสามารถสูง

การสื่อสารในทีม

- การสื่อสารที่ดีเป็นสิ่งสำคัญสำหรับการทำงานเป็นทีมที่มีประสิทธิภาพ
- ต้องมีการแลกเปลี่ยนข้อมูล
 - สถานะของงาน
 - การตัดสินใจในการออกแบบ
 - การเปลี่ยนแปลงในการตัดสินใจก่อนหน้านี้
- การสื่อสารที่ดีจะช่วยเสริมสร้างความสามัคคีของทีม เป็นการส่งเสริมความเข้าใจกันและกัน

ปัจจัยของการสื่อสารในทีม

- ขนาดของกลุ่ม
 - เมื่อกลุ่มมีขนาดใหญ่ขึ้น จะทำให้สื่อสารกับสมาชิกคนอื่น ๆ ได้ยากขึ้น
- โครงสร้างกลุ่ม
 - ทีมที่มีโครงสร้างอย่างไม่เป็นทางการจะมีการสื่อสารที่ดีกว่ากลุ่มที่มีโครงสร้างตามลำดับชั้น
- องค์ประกอบของทีม
 - การสื่อสารของทีมจะดีขึ้น เมื่อมีคนที่มีบุคลิกภาพแตกต่างกันร่วมทีม
 - ทีมควรมีการผสมผสานมากกว่าที่จะมีแต่คนเพศเดียวกัน
- สภาพแวดล้อมในการทำงานทางกายภาพ
 - องค์กรที่มีสถานที่ทำงานและสภาพแวดล้อมที่ดี สามารถช่วยสนับสนุนการสื่อสารได้

สรุป

สรุปประเด็นสำคัญ

- การจัดการโครงการที่ดีมีความสำคัญ
 - ช่วยให้โครงการวิศวกรรมซอฟต์แวร์มีการพัฒนาตามกำหนดเวลาและภายในงบประมาณ
- การจัดการซอฟต์แวร์แตกต่างจากการจัดการด้านวิศวกรรมแบบอื่น ๆ
 - ซอฟต์แวร์ไม่มีตัวตน
 - โครงการอาจเป็นเรื่องแปลกใหม่หรือมีนวัตกรรมใหม่
 - อาจจะไม่มีความรู้หรือประสบการณ์เดิมที่จะสามารถนำมาประยุกต์กับโครงการใหม่ ๆ ได้
 - กระบวนการซอฟต์แวร์เป็นเรื่องใหม่ ยังไม่พัฒนาเต็มขั้นเหมือนกระบวนการวิศวกรรมแบบอื่น ๆ
- การจัดการความเสี่ยงเกี่ยวข้องกับ
 - การระบุและประเมินความเสี่ยงของโครงการ
 - การวางแผนในการรับมือกับความเสี่ยง
 - ผลที่เกิดขึ้นกับโครงการหากความเสี่ยงนั้นเกิดขึ้น
 - การวางแผนที่จะหลีกเลี่ยงหรือจัดการกับความเสี่ยงที่อาจเกิดขึ้น

สรุปประเด็นสำคัญ

- การจัดการคนเกี่ยวข้องกับ
 - การเลือกคนที่เหมาะสมในการทำงานในโครงการ
 - การจัดทีมและสภาพแวดล้อมการทำงาน
- คนมีแรงบันดาลใจจาก
 - การปฏิสัมพันธ์กับคนอื่น
 - การรับรู้ของผู้บริหารและคนรอบข้าง
 - การให้โอกาสในการพัฒนาตนเอง

สรุปประเด็นสำคัญ

- ทีมพัฒนาซอฟต์แวร์ควรมีขนาดค่อนข้างเล็กและเหนียวแน่น
 - ปัจจัยสำคัญที่มีอิทธิพลต่อประสิทธิภาพของทีมคือ คนในทีม
 - ทีมต้องพัฒนาวิธีการจัดการและการสื่อสารระหว่างสมาชิกในทีม
- การสื่อสารภายในทีมได้รับอิทธิพลจากปัจจัยต่าง ๆ
 - สถานะของสมาชิกในทีม
 - ขนาดของทีม
 - องค์ประกอบทางเพศของบุคคลในทีม
 - บุคลิกและช่องทางการสื่อสารที่มีอยู่

คำถาม???