



CPE3243 วิศวกรรมซอฟต์แวร์ (Software Engineering)

Piyavit Laung-Aram
Major of Computer Engineering
Faculty of Engineering
Ramkhamhaeng University, Thailand

วิศวกรรมซอฟต์แวร์เบื้องต้น



วิศวกรรมซอฟต์แวร์คืออะไร?

วิศวกรรมซอฟต์แวร์ถูกกำหนดให้เป็นกระบวนการของการวิเคราะห์ความต้องการของผู้ใช้ จากนั้นจึงออกแบบ สร้าง และทดสอบแอปพลิเคชันซอฟต์แวร์ที่จะตอบสนองความต้องการเหล่านั้น



มาตรฐานจำกัดความต่างๆ ของวิศวกรรมซอฟต์แวร์กัน

- IEEE ในมาตรฐาน 610.12-1990 กำหนดวิศวกรรมซอฟต์แวร์ว่าเป็นการประยุกต์ใช้อย่างเป็นระบบ มีระเบียบวินัย ซึ่งเป็นแนวทางที่คำนวณได้สำหรับการพัฒนา การดำเนินการ และการบำรุงรักษาซอฟต์แวร์
- Fritz Bauer นิยามว่าเป็น 'การก่อตั้งและใช้หลักการทางวิศวกรรมมาตรฐาน ช่วยให้คุณได้รับซอฟต์แวร์ที่น่าเชื่อถือและทำงานได้อย่างมีประสิทธิภาพในเครื่องจริงอย่างประหยัด
- Boehm กำหนดวิศวกรรมซอฟต์แวร์ซึ่งเกี่ยวข้องกับ 'การประยุกต์ใช้ความรู้ทางวิทยาศาสตร์ในทางปฏิบัติกับการออกแบบที่สร้างสรรค์และการสร้างโปรแกรมคอมพิวเตอร์ นอกจากนี้ยังรวมถึงเอกสารที่เกี่ยวข้องซึ่งจำเป็นสำหรับการพัฒนาปฏิบัติการ และบำรุงรักษา"



สิ่งที่จะได้เรียนรู้ในวิชาวิศวกรรมซอฟต์แวร์

- วิศวกรรมซอฟต์แวร์คืออะไร?
- ทำไมต้องเป็นวิศวกรรมซอฟต์แวร์? วิฤตซอฟต์แวร์และแนวทางแก้ไข
- เหตุใดวิศวกรรมซอฟต์แวร์จึงเป็นที่นิยม?
- ความสัมพันธ์ระหว่างวิศวกรรมซอฟต์แวร์กับสาขาวิชาอื่นๆ
- ความท้าทายของวิศวกรรมซอฟต์แวร์
- คุณสมบัติสำหรับผลิตภัณฑ์ซอฟต์แวร์
- ลักษณะของซอฟต์แวร์ที่ดี



วิกฤตซอฟต์แวร์คืออะไร?

- เป็นช่วงปลายทศวรรษที่ 1960 ที่โครงการซอฟต์แวร์หลายโครงการล้มเหลว
- มีซอฟต์แวร์จำนวนมากที่พัฒนาเกินงบประมาณ ได้ผลลัพธ์เป็นซอฟต์แวร์ที่ไม่น่าเชื่อถือ มีราคาแพงในการบำรุงรักษา
- ซอฟต์แวร์ขนาดใหญ่กว่านั้นยากและค่อนข้างแพงในการบำรุงรักษา
- ซอฟต์แวร์จำนวนมากไม่สามารถตอบสนองความต้องการที่เพิ่มขึ้นของลูกค้าได้
- ความซับซ้อนของโครงการซอฟต์แวร์เพิ่มขึ้นเมื่อใดก็ตามที่ความสามารถของฮาร์ดแวร์เพิ่มขึ้น
- ความต้องการซอฟต์แวร์ใหม่เพิ่มขึ้นเร็วกว่าเมื่อเทียบกับความสามารถในการสร้างซอฟต์แวร์ใหม่

ปัญหาทั้งหมดข้างต้นนำไปสู่ 'วิกฤตซอฟต์แวร์'



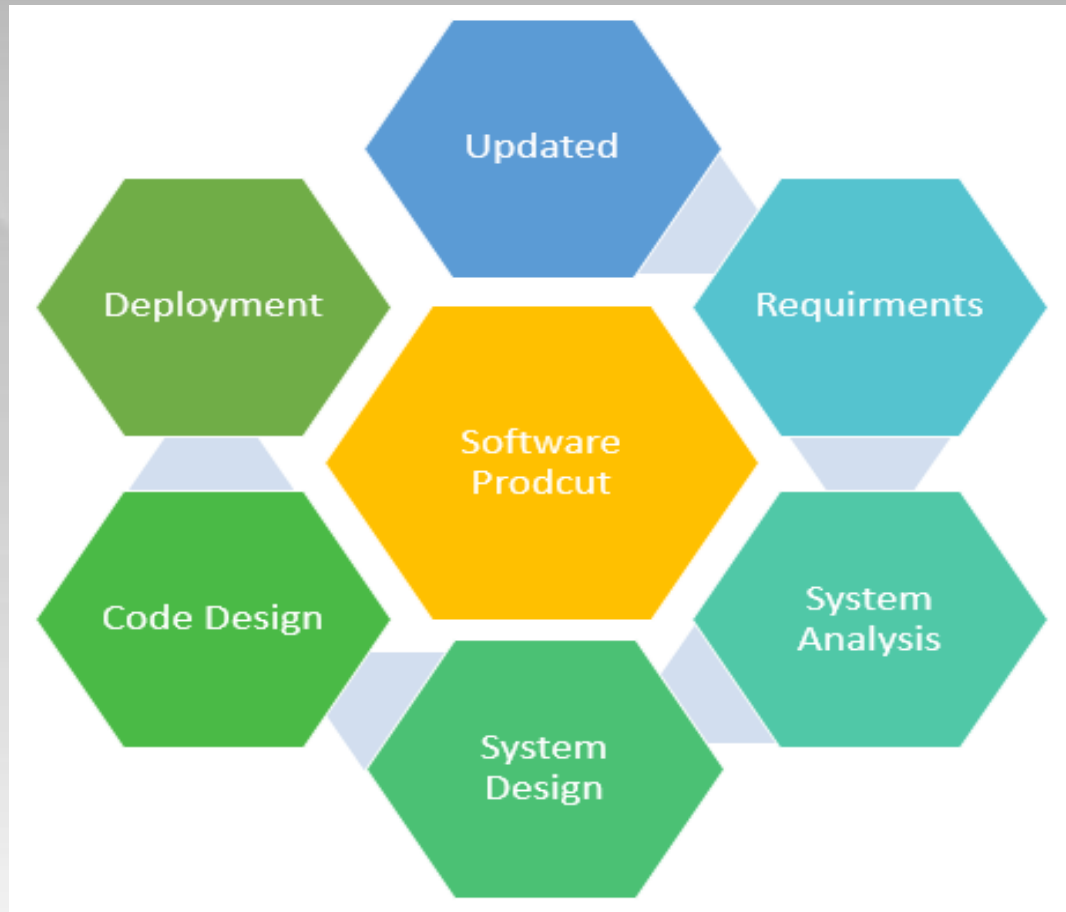
การแก้ไข้ปัญหา

วิธีแก้้ปัญหาคือเปลี่ยนความพยายามในการเขียนโปรแกรมที่ไม่มีหลักการควบคุมดูแลมาใช้หลักการทางวิศวกรรมซอฟต์แวร์ในควบคุมดูแลการพัฒนาซอฟต์แวร์ โมเดลทางวิศวกรรมเหล่านี้ช่วยให้บริษัทต่างๆ ปรับปรุงการดำเนินงานและส่งมอบซอฟต์แวร์ที่ตรงตามความต้องการของลูกค้า

- ปลายทศวรรษ 1970 มีการใช้หลักการทางวิศวกรรมซอฟต์แวร์อย่างแพร่หลาย
- ในช่วงปี 1980 ได้เห็นระบบอัตโนมัติของกระบวนการวิศวกรรมซอฟต์แวร์และการเติบโตของ (CASE) Computer-Aided Software Engineering
- ทศวรรษ 1990 มีการเน้นย้ำมากขึ้นในด้าน 'การจัดการ' ของมาตรฐานโครงการด้านคุณภาพและกระบวนการ เช่นเดียวกับ ISO 9001



ทำไมวิศวกรรมซอฟต์แวร์ถึงได้รับความนิยม?





ทำไมวิศวกรรมซอฟต์แวร์ถึงได้รับความนิยม? (ต่อ)

- ซอฟต์แวร์ขนาดใหญ่ – ในชีวิตจริงของเรา การสร้างกำแพงค่อนข้างสะดวกกว่าบ้านหรืออาคาร ในทำนองเดียวกัน เนื่องจากขนาดของซอฟต์แวร์มีขนาดใหญ่ขึ้น วิศวกรรมซอฟต์แวร์จึงเข้าช่วยให้คุณสามารถสร้างซอฟต์แวร์ได้ง่ายขึ้น มีความเป็นระบบและมีมาตรฐาน
- Scalability- หากกระบวนการพัฒนาซอฟต์แวร์ขึ้นอยู่กับแนวคิดและหลักการทางวิทยาศาสตร์และวิศวกรรม การสร้างซอฟต์แวร์ขึ้นมาใหม่ให้สามารถรองรับการปรับขนาดตามความต้องการที่มีอยู่จะง่ายกว่า
- ความสามารถในการปรับตัว: เมื่อใดก็ตามที่กระบวนการของซอฟต์แวร์ขึ้นอยู่กับแนวทางและหลักการทางวิทยาศาสตร์และวิศวกรรม การสร้างซอฟต์แวร์ใหม่บนรากฐานวิศวกรรมซอฟต์แวร์จะทำได้ง่าย



ทำไมวิศวกรรมซอฟต์แวร์ถึงได้รับความนิยม? (ต่อ)

- ต้นทุนการผลิต- ในด้านฮาร์ดแวร์ได้แสดงการผลิตจำนวนมากทำให้ต้นทุนของคอมพิวเตอร์และฮาร์ดแวร์อิเล็กทรอนิกส์ลดลง ในด้านซอฟต์แวร์ก็เช่นเดียวกัน การนำแนวคิดและหลักการทางวิทยาศาสตร์และวิศวกรรมมาใช้ในการผลิตซอฟต์แวร์จะทำให้ต้นทุนถูกลง
- Dynamic Nature- การเติบโตและการปรับการของซอฟต์แวร์อยู่เสมอ จะขึ้นอยู่กับสภาพแวดล้อมของผู้ใช้งานที่เปลี่ยนแปลงไป ต้องอาศัยแนวคิดและหลักการทางวิทยาศาสตร์และวิศวกรรมมาใช้ในการจัดการกับสิ่งเหล่านี้
- การจัดการคุณภาพ: การนำแนวคิดและหลักการทางวิทยาศาสตร์และวิศวกรรมมาใช้วิธีการพัฒนาซอฟต์แวร์ที่ดีขึ้นเพื่อจัดหาผลิตภัณฑ์ซอฟต์แวร์ที่มีคุณภาพ

วิศวกรรมซอฟต์แวร์เปรียบเทียบกับการ ก่อสร้างอาคาร



กระบวนการในการก่อสร้างอาคาร

1. คุยกับตัวแทนเพื่อบอกความต้องการ
2. การวิเคราะห์ความต้องการร่วมกับตัวแทน
3. การออกแบบคร่าว ๆ และประเมินต้นทุน
4. ตกลงเงื่อนไขสัญญาว่าจ้าง
5. การเก็บความต้องการของลูกค้าโดยละเอียด
6. การวิเคราะห์ความต้องการของลูกค้าโดยละเอียด
7. ออกแบบ
8. ก่อสร้าง
9. ควบคุมการก่อสร้าง
10. ทดสอบอาคาร
11. ส่งมอบงาน
12. การบำรุงรักษา



การเริ่มโครงการ

แบบบ้าน SEACON
กลุ่มบ้าน ราคา 4 ล้านบาท ขึ้นไป

ค้นหา
แบบบ้าน SEACON



4-5 ล้านบาท



5-6 ล้านบาท



6-8 ล้านบาท



8-10 ล้านบาท



10 ล้านบาทขึ้นไป



บ้านสามชั้น



คุยกับตัวแทนเพื่อบอกความต้องการ





คุยกับตัวแทนเพื่อบอกความต้องการ (ต่อ)

- งบประมาณ
- ความต้องการเบื้องต้น
- แบบบ้าน



แหล่งที่มา <https://www.seacon.com/>,
<https://home.kapook.com/>,
<https://bestswgarden.com/>



การวิเคราะห์ความต้องการร่วมกับตัวแทน





การออกแบบคร่าว ๆ และประเมินต้นทุน





ตกลงเซ็นสัญญาว่าจ้าง





การเก็บความต้องการของลูกค้าโดยละเอียด



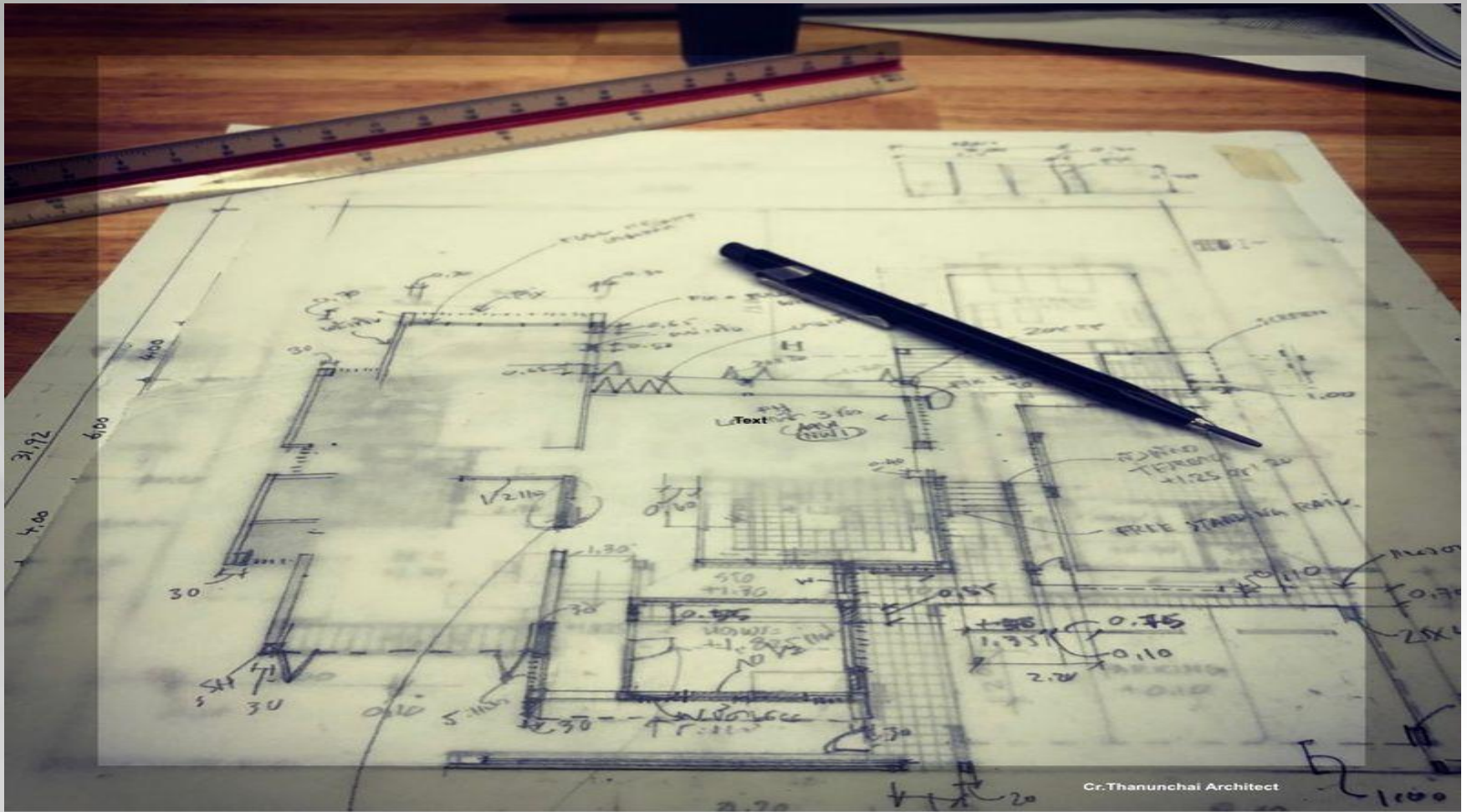


การวิเคราะห์ความต้องการของลูกค้าโดยละเอียด





ออกแบบ (ร่วมกับลูกค้า)





ออกแบบ (ต่อ)



ชุดที่ 6 แปลนพื้นที่บ้าน



ชุดที่ 6 แปลนพื้นที่บ้าน





ก่อสร้าง

Faculty of Engineering | Ramkhamhaeng
University



แหล่งที่มา <http://a-visiongroup.com/>



ก่อสร้าง (ต่อ)

- เตรียมพื้นที่
- งานวางผังอาคาร
- งานเสาเข็ม
- งานฐานรากโครงสร้างชั้นล่าง
- งานโครงสร้างชั้นสอง โครงหลังคา และโครงสร้างงานระบบสุขาภิบาล
- งานมุงหลังคา และโครงสร้างบันได
- งานก่อผนัง ติดตั้งวงกบไม้ประตู-หน้าต่าง และงานระบบไฟฟ้า-ประปา
- งานฉาบผนัง และงานติดตั้งฝ้าเพดาน ใน



ก่อสร้าง (ต่อ)

- งานวัสดุตกแต่งพื้นผิว ติดตั้งอุปกรณ์ ติดตั้งประตู-หน้าต่างและงาน Built-In
- วัสดุตกแต่งผนังและพื้น
- ระบบแสงสว่างและติดตั้งดวงโคม
- ติดตั้งบานประตู หน้าต่างไม้ ชุดประตู-หน้าต่างไวนิล/อะลูมิเนียม
- งาน Built-in
- ติดตั้งวัสดุอุปกรณ์ สุขภัณฑ์ในห้องน้ำ และอุปกรณ์เครื่องครัว
- สวนและทางเดินรอบบ้าน
- ทำความสะอาดและตรวจความเรียบร้อยในขั้นตอนการเก็บงาน



ควบคุมการก่อสร้าง





ทดสอบอาคาร





ส่งมอบงาน





การบำรุงรักษา





ความสัมพันธ์ระหว่างวิศวกรรมซอฟต์แวร์กับสาขาวิชาอื่น

วิศวกรรมซอฟต์แวร์เกี่ยวข้องกับสาขาวิชาอื่นอย่างไร

- วิทยาการคอมพิวเตอร์: วิศวกรรมซอฟต์แวร์เกี่ยวข้องกับพื้นฐานทางวิทยาศาสตร์ทางคณิตศาสตร์และฟิสิกส์
- วิทยาการจัดการ: วิศวกรรมซอฟต์แวร์เป็นงานที่ต้องใช้แรงงานมาก ซึ่งต้องการทั้งการควบคุมด้านเทคนิคและการจัดการ ดังนั้นจึงใช้กันอย่างแพร่หลายในด้านวิทยาการจัดการ



ความสัมพันธ์ระหว่างวิศวกรรมซอฟต์แวร์กับสาขาวิชาอื่น

- เศรษฐศาสตร์: ในภาคส่วนนี้ วิศวกรรมซอฟต์แวร์จะช่วยคุณในการประเมินทรัพยากรและควบคุมต้นทุน ต้องมีการพัฒนาระบบคอมพิวเตอร์ และข้อมูลควรได้รับการบำรุงรักษาอย่างสม่ำเสมอภายในงบประมาณที่กำหนด
- วิศวกรรมระบบ: ซอฟต์แวร์ส่วนใหญ่เป็นส่วนประกอบของระบบที่ใหญ่กว่ามาก ตัวอย่างเช่น ซอฟต์แวร์ในระบบตรวจสอบอุตสาหกรรม หรือซอฟต์แวร์การบินบนเครื่องบิน ควรใช้วิธีการทางวิศวกรรมซอฟต์แวร์ในการศึกษาระบบประเภทนี้



ความท้าทายของวิศวกรรมซอฟต์แวร์

นี่คือความท้าทายที่สำคัญบางประการที่วิศวกรซอฟต์แวร์ต้องเผชิญ:

- ในพื้นที่ที่มีความสำคัญต่อความปลอดภัย เช่น อวกาศ การบิน โรงไฟฟ้า นิวเคลียร์ ฯลฯ ต้นทุนของความล้มเหลวของซอฟต์แวร์อาจมีจำนวนมาก เนื่องจากมีความเสี่ยงต่อชีวิต
- ความต้องการของตลาดที่เพิ่มขึ้นสำหรับเวลาตอบสนองที่รวดเร็ว
- การจัดการกับความซับซ้อนที่เพิ่มขึ้นของความต้องการซอฟต์แวร์สำหรับแอปพลิเคชันใหม่
- ความหลากหลายของระบบซอฟต์แวร์ควรมีการสื่อสารระหว่างกัน



คุณสมบัติสำหรับผลิตภัณฑ์ซอฟต์แวร์

คุณลักษณะของผลิตภัณฑ์ซอฟต์แวร์ใด ๆ รวมถึงคุณลักษณะที่แสดงโดยผลิตภัณฑ์เมื่อติดตั้งและใช้งาน ไม่ใช่บริการที่จัดหาให้โดยผลิตภัณฑ์ แต่มีความเกี่ยวข้องกับพฤติกรรมแบบไดนามิกของผลิตภัณฑ์และการใช้ผลิตภัณฑ์ตัวอย่างของแอตทริบิวต์เหล่านี้คือ

ประสิทธิภาพ ความน่าเชื่อถือ ความทนทาน การบำรุงรักษา ฯลฯ

อย่างไรก็ตาม ความสำคัญเชิงสัมพัทธ์ของคุณลักษณะเหล่านี้แตกต่างกันไปในแต่ละระบบซอฟต์แวร์



คุณสมบัติสำหรับผลิตภัณฑ์ซอฟต์แวร์ (ต่อ)

ลักษณะผลิตภัณฑ์

อธิบาย

การบำรุงรักษา

ซอฟต์แวร์ควรมีการพัฒนาเพื่อตอบสนองความต้องการ
ที่เปลี่ยนแปลงไปของลูกค้า

ความน่าเชื่อถือ

ขึ้นกับคุณลักษณะต่างๆ ซอฟต์แวร์ที่เชื่อถือได้ไม่ควร
ก่อให้เกิดความเสียหายทางกายภาพหรือทางเศรษฐกิจ
ในเวลา que ระบบล้มเหลว

ประสิทธิภาพ

ซอฟต์แวร์แอปพลิเคชันควรใช้ทรัพยากรระบบไม่มาก
เกินไป เช่น หน่วยความจำและโปรเซสเซอร์

การใช้งาน

แอปพลิเคชันซอฟต์แวร์ควรมี UI และเอกสารประกอบเฉพาะ

การเพิ่มประสิทธิภาพแอตทริบิวต์ข้างต้นเป็นสิ่งที่ท้าทายมาก ตัวอย่างเช่น การนำเสนอ UI ที่ดีขึ้น
สามารถลดประสิทธิภาพของระบบได้



ลักษณะของซอฟต์แวร์ที่ดี

ซอฟต์แวร์ทุกตัวต้องมีคุณสมบัติดังต่อไปนี้:

- การทำงานที่ดี
- การเปลี่ยนผ่าน
- การบำรุงรักษาที่ดี



การทำงานที่ดี

คุณลักษณะนี้ทำให้เราทราบว่าซอฟต์แวร์ทำงานได้ดีเพียงใดในการดำเนินการซึ่งสามารถวัดได้จาก:

- งบประมาณ
- ประสิทธิภาพ
- การใช้งาน
- ความน่าเชื่อถือ
- ความถูกต้อง
- ฟังก์ชันการทำงาน
- ความปลอดภัย



การเปลี่ยนผ่าน

นี่เป็นส่วนสำคัญเมื่อซอฟต์แวร์ถูกย้ายจากแพลตฟอร์มหนึ่งไปยังอีกแพลตฟอร์มหนึ่ง

- การทำงานร่วมกัน
- การนำกลับมาใช้ใหม่ได้
- การพกพา
- การปรับตัว



การปรับตัว การเปลี่ยนผ่าน และการเคลื่อนย้ายที่ดี

แง่มุมนี้พูดถึงว่าซอฟต์แวร์มีความสามารถในการปรับตัวในสภาพแวดล้อมที่เปลี่ยนแปลงอย่างรวดเร็วได้ดีเพียงใด

- ความยืดหยุ่น
- การบำรุงรักษา
- ความเป็นโมดูล
- ความสามารถในการปรับขนาด



สรุป

- วิศวกรรมซอฟต์แวร์เป็นกระบวนการของการวิเคราะห์ความต้องการของผู้ใช้ จากนั้นจึงออกแบบ สร้าง และทดสอบแอปพลิเคชันซอฟต์แวร์ที่จะตอบสนองความต้องการนั้น
- เหตุผลสำคัญสำหรับการใช้วิศวกรรมซอฟต์แวร์ ได้แก่
 - 1) ซอฟต์แวร์ขนาดใหญ่
 - 2) ความสามารถในการปรับขนาด
 - 3) การปรับตัว
 - 4) ต้นทุน และ
 - 5) ลักษณะไดนามิก



สรุป (ต่อ)

- ในช่วงปลายทศวรรษ 1960 ซอฟต์แวร์จำนวนมากใช้แบบประมาณเกินงบประมาณ ดังนั้นจึงมีซอฟต์แวร์ที่ไม่น่าเชื่อถือซึ่งมีราคาแพงในการบำรุงรักษา
- ปลายทศวรรษ 1970 เห็นการใช้หลักการทางวิศวกรรมซอฟต์แวร์อย่างแพร่หลาย• แนวคิดวิศวกรรมซอฟต์แวร์
 - 1) วิทยาการคอมพิวเตอร์
 - 2) วิทยาการจัดการ
 - 3) วิศวกรรมระบบ และ
 - 4) เศรษฐศาสตร์



สรุป (ต่อ)

- ความต้องการของตลาดที่เพิ่มขึ้นสำหรับเวลาตอบสนองที่รวดเร็วเป็นความท้าทายที่ใหญ่ที่สุดของสาขาวิศวกรรมซอฟต์แวร์
 - 1) การบำรุงรักษา
 - 2) ความน่าเชื่อถือ
 - 3) ประสิทธิภาพ และ
 - 4) การใช้งานเป็นคุณลักษณะที่สำคัญที่สุดของผลิตภัณฑ์ซอฟต์แวร์ • ตาม

ลักษณะที่สำคัญที่สุดของซอฟต์แวร์ที่ดีคือ 1) การใช้งาน 2) การเปลี่ยนผ่าน 3) การบำรุงรักษา



สรุป (ต่อ)

- สามลักษณะที่สำคัญที่สุดของซอฟต์แวร์ที่ดีคือ
 - 1) การใช้งาน
 - 2) การเปลี่ยนผ่าน
 - 3) การบำรุงรักษา

10 ขั้นตอนในการเป็นวิศวกร/นักพัฒนาซอฟต์แวร์



วิศวกรรมซอฟต์แวร์คืออะไร?

- วิศวกรรมซอฟต์แวร์เป็นกระบวนการของการวิเคราะห์ความต้องการของผู้ใช้ จากนั้นจึงออกแบบ สร้าง และทดสอบแอปพลิเคชันซอฟต์แวร์ที่จะตอบสนองความต้องการเหล่านั้น



วิศวกรรมซอฟต์แวร์คืออะไร?

- วิศวกรรมซอฟต์แวร์เป็นกระบวนการของการวิเคราะห์ความต้องการของผู้ใช้ จากนั้นจึงออกแบบ สร้าง และทดสอบแอปพลิเคชันซอฟต์แวร์ที่จะตอบสนองความต้องการเหล่านั้น
- วิศวกรรมซอฟต์แวร์เป็นอาชีพที่เปลี่ยนแปลงตลอดเวลา ดังนั้นจึงควรนำเทคโนโลยีใหม่ๆ มาปรับใช้ เนื่องจากมีลักษณะที่เปลี่ยนแปลงไป จึงมีจุดเริ่มต้นที่แตกต่างกันมากมายในวิชาชีพไอทีอย่างไรก็ตาม ไม่มีทางเดียวที่จะบรรลุเป้าหมายนี้ได้ ต่อไปนี้เป็นขั้นตอนในการเป็นวิศวกรซอฟต์แวร์มืออาชีพ



ขั้นตอนที่ #1: ชัดเจนเกี่ยวกับเป้าหมายสุดท้ายของคุณ

การเริ่มต้นอาชีพใหม่ไม่ใช่เรื่องง่าย อย่างไรก็ตาม เมื่อคุณมีจุดหมายสุดท้ายในใจ จะช่วยให้คุณในการค้นหาและจัดการกับอุปสรรคในถนนของคุณ คุณควรมีเป้าหมายที่ชัดเจนเช่น

- ฉันต้องการสร้างอาชีพของฉันในฐานะวิศวกรซอฟต์แวร์
- ฉันต้องการทำงานเป็นวิศวกรซอฟต์แวร์ที่บริษัท ไอทีที่จัดตั้งขึ้น
- ฉันต้องการทำงานเป็นทีมและได้รับค่าตอบแทนที่ดี













ขั้นตอนที่ #2: เลือกภาษาที่จะเรียนรู้

วิศวกรรมซอฟต์แวร์ไม่เคยเน้นที่การเขียนโค้ดเพียงอย่างเดียว อย่างไรก็ตาม คุณควรรู้อย่างน้อยสองภาษา และเข้าใจอย่างลึกซึ้งถึงวิธีการทำงานของภาษาเหล่านั้น ไม่มีข้อตกลงร่วมกันว่าภาษาใดมีประโยชน์มากที่สุดด้านล่างนี้เป็นรายการภาษาโปรแกรมยอดนิยมบางภาษา

- Python
- JavaScript
- C#
- C/C++
- Ruby
- Java และ ฯลฯ












TIOBE รายงานอันดับภาษาโปรแกรม สิงหาคม 2024

Aug 2024	Aug 2023	Change	Programming Language		Ratings	Change
1	1			Python	18.04%	+4.71%
2	3	▲		C++	10.04%	-0.59%
3	2	▼		C	9.17%	-2.24%
4	4			Java	9.16%	-1.16%
5	5			C#	6.39%	-0.65%
6	6			JavaScript	3.91%	+0.62%
7	8	▲		SQL	2.21%	+0.68%
8	7	▼		Visual Basic	2.18%	-0.45%
9	12	▲		Go	2.03%	+0.87%
10	14	▲		Fortran	1.79%	+0.75%



TIOBE รายงานอันดับภาษาโปรแกรม สิงหาคม 2024

11	13	▲		MATLAB	1.72%	+0.67%
12	23	▲		Delphi/Object Pascal	1.63%	+0.83%
13	10	▼		PHP	1.46%	+0.19%
14	19	▲		Rust	1.28%	+0.39%
15	17	▲		Ruby	1.28%	+0.37%
16	18	▲		Swift	1.28%	+0.37%
17	9	▼		Assembly language	1.21%	-0.13%
18	27	▲		Kotlin	1.13%	+0.44%
19	16	▼		R	1.11%	+0.19%
20	11	▼		Scratch	1.09%	-0.13%

TIOBE Programming Community Index

Source: www.tiobe.com

แหล่งที่มา <https://www.tiobe.com/tiobe-index/>



ขั้นตอนที่ #3: รับปริญญาวิศวกรรมคอมพิวเตอร์ วิทยาการคอมพิวเตอร์ หรือสาขาที่เกี่ยวข้อง

ก่อนอื่นคุณควรได้รับปริญญาหรือสำเร็จการศึกษาวิศวกรรมคอมพิวเตอร์ วิทยาการคอมพิวเตอร์หรือสาขาที่เกี่ยวข้อง คุณควรมีอย่างน้อย 1 ปริญญาตรีเพื่อเริ่มต้นอาชีพของคุณในฐานะวิศวกรซอฟต์แวร์วิชาเอกวิทยาการคอมพิวเตอร์จะช่วยให้คุณมีพื้นฐานที่เป็นประโยชน์มากที่สุดสำหรับการออกแบบซอฟต์แวร์ ผู้สัมภาษณ์จะถามคำถามเกี่ยวกับโครงสร้างข้อมูลและอัลกอริทึม ดังนั้น ความรู้เชิงทฤษฎีจากปริญญาวิทยาการคอมพิวเตอร์แบบดั้งเดิมจึงเตรียมคุณให้พร้อมสำหรับสิ่งนี้ได้ดีที่สุดอย่างไรก็ตาม คุณจะใช้เวลาส่วนใหญ่นอกซอฟต์แวร์เขียนโค้ดในชั้นเรียน และเรียนรู้วิธีเขียนซอฟต์แวร์จริง (ความเห็นในปัจจุบันจะต่างจากนี้....)



ขั้นตอนที่ #4: เสริมการศึกษาและองค์ความรู้ของตนเอง

หนังสือที่เราใช้ศึกษามักจะล้าสมัย หนังสืออ้างอิงได้รับการแก้ไขช้ากว่าการอัปเดตซอฟต์แวร์มาก สถาบันการศึกษาทุกแห่งมีแนวคิดเชิงทฤษฎีและวิธีคิดที่สำคัญต่อความสำเร็จของคุณแม้ว่าคุณจะได้รับเงินเท่าไรก็สามารถใช้ความรู้เกี่ยวกับซอฟต์แวร์ในโลกแห่งความเป็นจริงได้ นี่คือการที่การศึกษาของคุณช่วยคุณได้ต่อไปนี้เป็นวิธีเสริมการเรียนรู้ของคุณ

- StackOverflow เป็นเว็บไซต์คำถามและคำตอบที่น่าเชื่อถือที่สุดสำหรับนักพัฒนา คุณสามารถค้นหาตามเทคโนโลยี พื้นที่ปัญหา หรือภาษาที่คุณต้องการปรับปรุง
- เว็บไซต์เช่น CodinGame และ CodeWars นำเสนอโซลูชันการเขียนโปรแกรมหลายพันรายการซึ่งช่วยให้คุณทดสอบทักษะของคุณได้



ขั้นตอนที่ #5: อ่าน โค้ดที่เขียน โดยนักพัฒนาที่มี ประสบการณ์

อีกแง่มุมที่เป็นประโยชน์ในการเป็นวิศวกรซอฟต์แวร์คือการเรียนรู้วิธีการอ่านโค้ดของผู้อื่น คุณสามารถทำสิ่งนี้ได้อย่างง่ายดายโดยไปที่ไซต์ต่างๆ เช่น ที่เก็บ GitHub และดูเอกสารประกอบอย่างไรก็ตาม บางครั้งเอกสารก็ผิดพลาดเช่นกัน แต่ซอร์สโค้ดส่วนใหญ่มีความถูกต้องในไซต์เหล่านี้ ดังนั้น โดยการเรียนรู้ที่จะอ่านโค้ดนั้น คุณจะสามารถเข้าใจว่าโปรแกรมเฉพาะทำงานอย่างไร



ขั้นตอนที่ #6: ค้นหาชุมชนวิศวกรซอฟต์แวร์

คุณต้องค้นหาชุมชนเพื่อเรียนรู้ว่าแนวทางใดที่ทำให้คุณประสบความสำเร็จในฐานะวิศวกรซอฟต์แวร์ ดังนั้น เมื่อคุณอยู่ท่ามกลางชุมชนของคนที่มีความคิดเหมือนกัน คุณจะมึระบบสนับสนุนอัตโนมัติ ชุมชนวิศวกรรมซอฟต์แวร์ในโลกแห่งความเป็นจริงจะให้คำแนะนำเกี่ยวกับจุดที่จะเน้นการเรียนรู้ของคุณ ไซต์อย่าง Meetup เป็นตัวเลือกที่ดีในการค้นหาผู้เชี่ยวชาญคนอื่นๆ



ขั้นตอนที่ #7: สร้างโครงการ

ไม่มีอะไรเหมือนประสบการณ์จริง คุณต้องเริ่มสร้างโครงการในภาษาที่คุณคุ้นเคย ด้วยการทำเช่นนี้ คุณไม่เพียงแต่ปรับปรุงการเรียนรู้ของคุณ แต่ยังพบนายจ้างที่มีศักยภาพอีกด้วย เมื่อคุณเรียนรู้วิธีเขียนโค้ด สร้างโครงการ ฯลฯ คุณควรค่อยๆ ช่วยสร้างเครือข่ายมืออาชีพของคุณ



ขั้นตอนที่ #8: ฝึกฝนทักษะก่อนสัมภาษณ์

ต่อไป คุณต้องให้ความสำคัญกับเรซูเม่ของคุณ เนื่องจากมันมีบทบาทสำคัญในการจ้างงานของคุณ คุณจะต้องเชี่ยวชาญสามสิ่ง ได้แก่ ประวัติย่อจดหมายสมัครงาน และสถานะออนไลน์ของคุณ การว่าจ้างผู้จัดการฝ่ายทรัพยากรบุคคลจะยังพิจารณาคุณพวกเขาเพื่อรับภาพรวมของทักษะและประสบการณ์ของคุณ นอกจากนี้ จำเป็นต้องมีจดหมายปะหน้าเมื่อสมัครงานออนไลน์ เป็นโอกาสของคุณที่จะได้รับความสนใจจากใครบางคนและสร้างความประทับใจ



ขั้นตอนที่ #9: จัดการสัมภาษณ์งานของคุณ

ในที่สุด ก็ถึงเวลาเผชิญหน้ากับการสัมภาษณ์ บริษัทไอทีส่วนใหญ่เริ่มต้นด้วยการสัมภาษณ์แบบคัดกรอง ซึ่งมักจะทำทางอินเทอร์เน็ตที่บ้าน หลังจากนั้นคุณต้องเผชิญการสัมภาษณ์ทางโทรศัพท์ หากคุณเคลียร์ทั้งสองการสัมภาษณ์ได้สำเร็จ คุณจะได้สัมภาษณ์ในสถานที่จริง



ขั้นตอนที่ #10: ยอมรับข้อเสนองาน

หลังจากที่คุณเคลียร์รอบการสัมภาษณ์และแสดงทักษะใหม่ของคุณสำเร็จแล้ว คุณจะมีโอกาสได้รับการเสนองาน เลือกสิ่งที่ดีที่สุดที่จะช่วยให้คุณพัฒนาอาชีพการงานของคุณ



วิศวกรซอฟต์แวร์ทำอะไร?

นี่คือบทบาทสำคัญของวิศวกรซอฟต์แวร์

- ควรเป็น โปรแกรมเมอร์ที่ดีและควรรอบรู้ใน โครงสร้างข้อมูลและ อัลกอริทึม
- ควรคล่องแคล่วในภาษาโปรแกรมมากกว่าหนึ่งภาษา
- ควรคุ้นเคยกับแนวทางการออกแบบต่างๆ
- รับผิดชอบในการกำหนดความเป็นไปได้ของการดำเนินงานซอฟต์แวร์ ผ่านการกำหนดปัญหา การประเมิน การวิเคราะห์ และการพัฒนาโซลูชัน
- จัดทำเอกสารและสถิติวิธีแก้ปัญหาโดยการพัฒนาฟังก์ชัน ไคอะแกรม เอกสารประกอบ ไคอะแกรม เลย์เอาต์ ข้อคิดเห็นเกี่ยวกับโค้ด ฯลฯ



วิศวกรซอฟต์แวร์ทำอะไร? (ต่อ)

- พัฒนาโซลูชันโดยปฏิบัติตามมาตรฐานการพัฒนาซอฟต์แวร์
- ปรับปรุงการดำเนินงานโดยทำงานวิเคราะห์ระบบให้เสร็จสิ้น และให้คำแนะนำสำหรับขั้นตอนและนโยบายที่ไม่มีประสิทธิภาพ
- การรับและอนุญาตให้ใช้ซอฟต์แวร์หลังจากได้รับข้อมูลที่เกี่ยวข้องจากผู้ขาย
- การทดสอบและอนุมัติผลิตภัณฑ์
- การให้ข้อมูลที่เกี่ยวข้องผ่านการรวบรวม วิเคราะห์ และสรุปประเด็นการบริการและการพัฒนา



อาชีพต่างๆ ในสาขาวิศวกรรมซอฟต์แวร์

วิศวกรรมซอฟต์แวร์มีโอกาสมากมายสำหรับมืออาชีพ แม้ว่าสถานที่
อุตสาหกรรมและที่ตั้งทางภูมิศาสตร์จะมีบทบาทสำคัญในการสร้างรายได้
และนายจ้างยังพิจารณาระดับการศึกษาและประสบการณ์เพื่อกำหนด
แพ็คเกจเงินเดือนนี้คือตัวเลือกอาชีพที่สำคัญที่คุณมีหลังจากเลือกหลักสูตร
นี้



ตัวเลือกอาชีพที่สำคัญที่คุณมี

ตำแหน่งงาน	คำอธิบาย	วุฒิการศึกษาที่ต้องการ	เงินเดือนประจำปี
วิศวกรซอฟต์แวร์	วิศวกรซอฟต์แวร์ทำงานร่วมกับโปรแกรมเมอร์ นักออกแบบ และผู้ทดสอบเพื่อวางแผนฟังก์ชันอื่นๆ เพื่อให้ซอฟต์แวร์ทำงานได้อย่างมีประสิทธิภาพ	ปริญญาตรี	\$83,515
วิศวกรซอฟต์แวร์หลัก	วิศวกรซอฟต์แวร์หลักดูแลรายละเอียดทางเทคนิคทั้งหมดของโครงการ งานหลักของพวกเขา รวมถึงการปรับขนาดอย่างมีประสิทธิภาพและการเพิ่มประสิทธิภาพสูงสุด	ปริญญาตรี มีประสบการณ์ มีการรับรอง	\$132,964
Lead Software Development Engineer	สร้างซอฟต์แวร์ที่กำหนดเองเพื่อตอบสนองความต้องการขององค์กร การออกแบบซอฟต์แวร์ทั้งหมด บริหารจัดการทีมวิศวกรซอฟต์แวร์	ปริญญาโท	\$106,650
Lead Software Development Engineer			



จรรยาบรรณและแนวปฏิบัติที่ดีที่สุดสำหรับวิศวกรซอฟต์แวร์

แนวทางปฏิบัติที่ดีที่สุดสำหรับวิศวกรซอฟต์แวร์มีดังนี้

- วิศวกรซอฟต์แวร์ควรดำเนินการในลักษณะที่เป็นประโยชน์ต่อลูกค้าและนายจ้าง
- ตรวจสอบให้แน่ใจว่าผลิตภัณฑ์ซอฟต์แวร์และการดัดแปลงที่เกี่ยวข้องซึ่งควรเป็นไปตามมาตรฐานระดับมืออาชีพสูงสุด
- ต้องรักษาความซื่อสัตย์และความเป็นอิสระในแนวทางวิชาชีพของตน
- ส่งเสริมแนวทางจริยธรรมสำหรับการพัฒนาและบำรุงรักษาซอฟต์แวร์
- วิศวกรซอฟต์แวร์จะต้องสนับสนุนเพื่อนร่วมงานของตน



Q & A

