

ระบบแจ้งเตือนการทำงานผิดพลาดของเครื่องทดสอบประสิทธิภาพฮาร์ดดิสก์

สุภัทรลักษณ์ โพธิ์หา

ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
Emails: 5606021630250@fitm.kmutnb.ac.th

บทคัดย่อ

ระบบแจ้งเตือนการทำงานผิดพลาดของเครื่องทดสอบประสิทธิภาพฮาร์ดดิสก์ เป็นการพัฒนาซอฟต์แวร์ที่ใช้กับเครื่องจักรของ บริษัท เอชจีเอสที (ประเทศไทย) จำกัด ซึ่งผลจากการพัฒนา ทำให้ได้ซอฟต์แวร์ต่างๆ ที่สามารถเฝ้าอำนวยความสะดวกสบายให้แก่ผู้ใช้งานเกี่ยวกับการจัดการกับข้อมูล และสามารถแจ้งเตือนการทำงานผิดพลาดของเครื่องจักรได้อย่างรวดเร็ว ไม่ซับซ้อน และเข้าใจง่ายมากยิ่งขึ้น ซึ่งในปัจจุบันนี้ทางบริษัทก็ได้เริ่มนำร่องติดตั้งระบบใหม่ควบคู่ไปกับระบบเก่าเพื่อทดสอบประสิทธิภาพการทำงานของระบบใหม่ เป็นระยะๆ ก่อนปรับเปลี่ยนมาเป็นระบบใหม่ทั้งหมด ซึ่งคาดว่าจะระบบใหม่นี้จะสามารถช่วยลดค่าใช้จ่ายเกี่ยวกับค่าบำรุงรักษาทางซอฟต์แวร์ของระบบเดิมได้ปีละประมาณ 7,251,769 บาท อีกทั้งสามารถช่วยขยายขีดจำกัดการใช้ข้อมูลจากเครื่องจักร เพื่อนำไปพัฒนาซอฟต์แวร์หรือนวัตกรรมใหม่ๆ ในอนาคตได้

ABSTRACT

The Warning Malfunction System for Hard disk Automated Test Platform. This is develop software for machine in HGST (Thailand) company. Which from the development of the system. The system can help to comfort for user about management with the data, and can warning malfunction of the machine quickly, not complicated and understand simple. Which in the currently, the company began to pilot a new system coupled with the old system. To test the performance periodically, modified into a whole system. And expect that the new system. It can help reduce the cost of software maintenance of the old system has approximately 7,251,769 per years. Also can help expand limits using data from the machine. For the development of software or innovation in the future.

คำสำคัญ—เครื่องทดสอบประสิทธิภาพของฮาร์ดดิสก์; ระบบตรวจสอบการทำงานเครื่องจักร; ล็อกไฟล์

1. บทนำ

ในปัจจุบันคงปฏิเสธไม่ได้ว่าระบบเทคโนโลยีมีความหลากหลายและพัฒนาก้าวหน้ามากขึ้นเรื่อยๆ เนื่องจากสามารถช่วยอำนวยความสะดวกสบายให้กับมนุษย์ ระบบเทคโนโลยีจึงได้เข้ามามีบทบาท ในการดำรงชีวิตของมนุษย์มากขึ้นด้วยเช่นกัน โดยเฉพาะในธุรกิจอุตสาหกรรมต่างๆ ที่นิยมนำระบบเทคโนโลยีเครื่องจักร มาประกอบใช้ในกระบวนการผลิตต่างๆ โดยประสานการทำงานร่วมกับมนุษย์ เพื่อให้ได้ผลิตภัณฑ์ที่มีประสิทธิภาพแม่นยำ มีคุณภาพ และรวดเร็ว เป็นไปตามระยะเวลาที่กำหนดไว้

การปรับเปลี่ยนมาใช้เทคโนโลยีเครื่องจักรนั้น สามารถทำงานได้ตลอด 24 ชั่วโมงโดยไม่ต้องหยุดพักเครื่อง แต่ต้องมีผู้ดูแลประจำเครื่อง ซึ่งในครั้งนี้ ทางผู้จัดทำเลือกศึกษาและพัฒนา ระบบของเครื่องจักรที่ใช้ในการทดสอบประสิทธิภาพฮาร์ดดิสก์ ซึ่งเครื่องจักรจะทำการทดสอบฮาร์ดดิสก์แบบอัตโนมัติ แต่ต้องมีผู้ดูแลประจำเครื่อง ในการป้อนฮาร์ดดิสก์เข้าสู่เครื่องจักร และคอยแก้ไขปัญหาหรือข้อผิดพลาดต่างๆ ซึ่งในการทดสอบแต่ละรอบ จะทดสอบฮาร์ดดิสก์ในปริมาณมาก และใช้เวลายาวนาน หากมีการทำงานผิดพลาดของเครื่องจักร และไม่ได้รับการแก้ไขปัญหาย่างทันท่วงที อาจจะส่งผลกระทบต่อกระบวนการผลิตล่าช้า ดังนั้นเครื่องจักรจึงต้องมีซอฟต์แวร์ในการจัดการกับข้อมูลต่างๆ และคอยแจ้งเตือนการทำงานผิดพลาดของเครื่องจักร เพื่อให้สามารถทำการตรวจสอบและแก้ไขปัญหาได้อย่างทันท่วงที แต่ด้วยซอฟต์แวร์ของระบบเดิมนั้น มีการออกแบบที่ค่อนข้างซับซ้อน เข้าใจยาก ไม่เฝ้าอำนวยความสะดวกสบายแก่ผู้ใช้งานเท่าที่ควร จึงเป็นปัญหาทำให้ผู้ใช้งานใช้งานซอฟต์แวร์ไปอย่างผิดๆ ง่ายๆ ไม่มีประสิทธิภาพ ส่งผลให้ข้อมูลที่ได้จากการใช้งานนั้น ผิดพลาดตามไปด้วย

จากปัญหาดังกล่าว ผู้จัดทำจึงได้พัฒนาระบบแจ้งเตือนการทำงานผิดพลาดของเครื่องทดสอบประสิทธิภาพฮาร์ดดิสก์ ระบบนี้ขึ้นมา เพื่อใช้ในการจัดการกับข้อมูลและทำการแจ้งเตือนปัญหาหรือผิดพลาดต่างๆ จากการทำงานของเครื่องจักรต่างๆ ได้โดยมีการออกแบบซอฟต์แวร์ในส่วนต่อประสานผู้ใช้ที่ไม่ซับซ้อน เข้าใจง่าย สามารถเฝ้าอำนวยความสะดวกสบายแก่ผู้ใช้งานได้มากยิ่งขึ้น ส่งผลให้กระบวนการทดสอบผลิตภัณฑ์ดำเนินไปได้อย่างรวดเร็ว มีคุณภาพ และมีประสิทธิภาพมากยิ่งขึ้น

2. ทฤษฎีที่เกี่ยวข้อง

การพัฒนาระบบแจ้งเตือนการทำงานผิดพลาดของเครื่องทดสอบประสิทธิภาพฮาร์ดดิสก์ในครั้งนี้ ทางบริษัท HGST (ประเทศไทย) จำกัด ได้เปิดโอกาสให้นักศึกษาโครงการสหกิจศึกษาเข้าร่วมเป็นส่วนหนึ่งในทีมพัฒนา ซึ่งส่วนในการพัฒนาของนักศึกษาจะอยู่ในส่วนของการพัฒนา Web Application ต่างๆ โดยมีทฤษฎีที่เกี่ยวข้อง ดังต่อไปนี้

2.1. Framework

Framework [1] หมายถึง โปรแกรมที่เขียนมาเป็นระบบพื้นฐาน มีโครงสร้างที่จะทำให้สามารถทำงานได้อย่างสะดวกมากยิ่งขึ้น ช่วยลดระยะเวลาในการพัฒนาโปรแกรมได้ ซึ่งถ้าต้องการทำงานอยู่ใต้ Framework จะต้องทำความเข้าใจกฎระเบียบ และวิธีการของ Framework นั้นๆ ถึงจะสามารถใช้งานระบบพื้นฐานนั้นได้ Framework เป็นส่วนสำคัญในการทำงานขนาดใหญ่ ที่มีโปรแกรมเมอร์จำนวนมาก เมื่อโปรแกรมเมอร์ทุกคนทำตามกฎระเบียบ และวิธีการของ Framework โดยเคร่งครัดแล้ว งานจะสำเร็จไปได้อย่างรวดเร็ว

2.2. Object-Oriented Programming (OOP)

OOP [2] หมายถึง การเขียนโปรแกรมเชิงวัตถุเป็นหนึ่งในรูปแบบการเขียนโปรแกรมคอมพิวเตอร์ ที่ให้ความสำคัญกับวัตถุ ซึ่งจะสามารถนำมาประกอบรวมกัน และนำมาทำงานร่วมกันได้ โดยการแลกเปลี่ยนข่าวสารเพื่อนำมาประมวลผล และส่งข่าวสารที่ได้ไปให้วัตถุอื่นๆ ที่เกี่ยวข้อง เพื่อให้สามารถทำงานต่อไปได้ ซึ่งแนวคิดการเขียนโปรแกรมแบบดั้งเดิม มักนิยมใช้การเขียนโปรแกรมแบบเชิงกระบวนการ ซึ่งให้ความสำคัญกับขั้นตอนกระบวนการที่ทำ โดยแบ่งโปรแกรมออกเป็น ส่วนๆ ตามลำดับขั้นตอนการทำงาน แต่แนวคิดการเขียนโปรแกรมเชิงวัตถุ นั้นให้ความสำคัญกับข้อมูล พฤติกรรมของวัตถุ และความสัมพันธ์กันระหว่างวัตถุกันมากกว่า

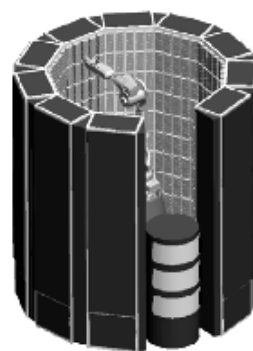
3. วิธีการดำเนินงาน

การพัฒนาระบบแจ้งเตือนการทำงานผิดพลาดของเครื่องทดสอบประสิทธิภาพฮาร์ดดิสก์ระบบนี้ ทางผู้จัดทำได้เลือกใช้ วิธีการพัฒนาแบบ Waterfall Model ซึ่งเป็นวิธีการพัฒนาซอฟต์แวร์ที่นิยมใช้ในการพัฒนาโครงการต่างๆ โดยมีขั้นตอน และวิธีการดำเนินงานดังต่อไปนี้

3.1. การศึกษาและวางแผนระบบ

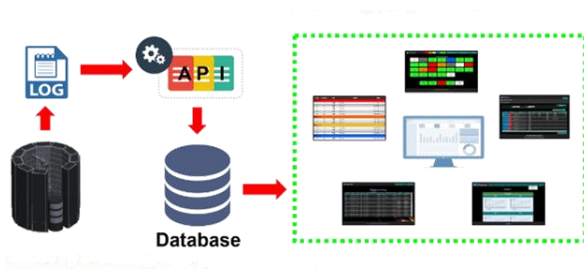
การศึกษาและวางแผนระบบ ทางทีมพัฒนาได้เลือกใช้เครื่องจักรที่มีชื่อว่า Neptune Tester [3] ซึ่งจะเป็นเครื่องจักรที่ใช้ในการทดสอบฮาร์ดดิสก์แบบอัตโนมัติ มนุษย์จะมีหน้าที่เพียงทำ

การนำฮาร์ดดิสก์บรรจุลงกล่องในตำแหน่งต่างๆ ให้ถูกต้อง จากนั้นจะนำกล่องเข้าสู่เครื่องจักรผ่านช่องนำเข้า ทางด้านหน้าของเครื่องจักร เมื่อกล่องที่บรรจุฮาร์ดดิสก์เข้าสู่เครื่องจักรแล้ว หุ่นยนต์ภายในเครื่องจักรจะทำการนำฮาร์ดดิสก์ไปยังช่องทดสอบฮาร์ดดิสก์แต่ละตัว จากนั้นทำการตรวจสอบบาร์โค้ดและทำการทดสอบ ซึ่งจะมีการควบคุมสิ่งเร้าต่างๆ หากมีข้อผิดพลาดเกิดขึ้น เครื่องจักรจะทำการแจ้งเตือนข้อผิดพลาดต่างๆ ผ่านทางหน้าจอที่ถูกติดไว้หน้าเครื่องจักร เมื่อทำการทดสอบเสร็จสิ้นแล้ว เครื่องจักรจะนำฮาร์ดดิสก์บรรจุลงในกล่อง ณ ตำแหน่งเดิมกับตอนนำเข้า แล้วจะส่งกล่องออกมานอกเครื่องจักร จึงจะถือว่าทำการทดสอบเสร็จสิ้น โดยจะเห็นว่า เป็นการลดระยะเวลาลดขั้นตอนการทำงานของมนุษย์ และสามารถช่วยเพิ่มปริมาณงานที่ทำให้มากขึ้นได้อีกด้วย



รูปที่ 1. แบบจำลองเครื่องจักรที่มีชื่อว่า “Neptune Tester”

ซึ่งข้อมูลที่ถูกนำมาใช้ในการพัฒนาซอฟต์แวร์ต่างๆ นั้น จะมาจากระบบฐานข้อมูล ซึ่งเป็นข้อมูลที่ถูกส่งมาจากรองการของเครื่องจักรแต่ละเครื่อง เมื่อพบข้อผิดพลาดต่างๆ เกิดขึ้น เครื่องจักรจะทำการรวบรวมข้อมูลและข้อผิดพลาดต่างๆ ให้อยู่ในรูปแบบโครงสร้างของ XML โดยจะมีข้อมูลการจราจรทางคอมพิวเตอร์ และรายละเอียดของการทำงานผิดพลาดต่างๆ จัดเก็บเป็นแท็กเรียบร้อย มีลักษณะโครงสร้างเดียวกันกับการใช้แท็กของ XML เช่น แท็กชื่อเครื่อง แท็กรหัสปัญหา และแท็กสถานะการทำงาน เป็นต้น จากนั้นเครื่องจักรจะทำการนำข้อมูลที่ได้อ่านมาสร้างเป็น Log File เพื่อส่งให้กับระบบที่ต้องการนำข้อมูลไปพัฒนาต่อ โดยการพัฒนาระบบครั้งนี้ ข้อมูลใน Log File จะไม่สามารถใช้งานได้เลยทันที ต้องผ่าน API สำหรับอ่านข้อมูลใน Log File เสียก่อน เพื่อกู้คืนข้อมูลให้อยู่ในรูปแบบของสารสนเทศที่สมบูรณ์ จึงจะสามารถใช้ประโยชน์จากข้อมูลเหล่านั้นได้อย่างมีประสิทธิภาพ ซึ่งหลังจากได้ Log File แล้ว จะมีโครงสร้างในการนำข้อมูลมาพัฒนาเป็นซอฟต์แวร์ต่างๆ ดังต่อไปนี้



รูปที่ 2. โครงสร้างเบื้องต้นในการพัฒนาระบบใหม่

จากภาพจะเห็นได้ว่า เมื่อได้ข้อมูล Log File จากเครื่องจักรแล้ว ข้อมูลจะถูกแปลงด้วย API ต่างๆ แล้วส่งไปจัดเก็บในระบบฐานข้อมูลเพื่อใช้ในการพัฒนาซอฟต์แวร์ต่างๆ ซึ่งส่วนที่นักศึกษารับผิดชอบนั้น จะอยู่ในบริเวณกรอบเส้นประสีเขียวเป็นการพัฒนา Web Application โดยมีความต้องการของผู้ใช้งานทั้งหมด แบ่งออกเป็น 4 ส่วน ดังต่อไปนี้

ส่วนที่ 1 Web Application “Neptune Status” ผู้ใช้งานต้องการโปรแกรม ที่มีหน้าที่สำหรับการแสดงสถานะการทำงานของเครื่องจักรแต่ละเครื่อง และทำการแจ้งเตือน เมื่อพบข้อผิดพลาดต่างๆ จากการการทำงานของเครื่องจักรเกิดขึ้น

ส่วนที่ 2 Web Application “Neptune Downtime” ผู้ใช้ต้องการโปรแกรม ที่มีหน้าที่สำหรับการจัดการกับข้อมูล และทำการตอบสนองต่อข้อผิดพลาดต่างๆ ที่เกิดขึ้น เมื่อมีการทำงานผิดพลาดของเครื่องจักร ระบบจะต้องทำการแจ้งข้อผิดพลาดผ่านทาง Web Application นี้ เพื่อให้ผู้ดูแลหรือผู้ควบคุมเครื่องจักร รับทราบถึงปัญหา และทำการแก้ไขปัญหาต่างๆ นั้นได้โดยเร็ว ซึ่งจะต้องสามารถรับรู้ได้ด้วยว่ามีผู้ดูแลหรือผู้ควบคุมเครื่องจักรคนใด เข้าไปแก้ไขหรือซ่อมแซมแล้วหรือยัง เพื่อลดปัญหาการแก้ไขข้อผิดพลาดซ้ำซ้อนของทีมงาน

ส่วนที่ 3 Web Application “Neptune Report” ผู้ใช้งานต้องการโปรแกรมที่สามารถทำหน้าที่สำหรับรายงานสถานะการทำงานของเครื่องจักรแต่ละเครื่องได้ โดยประมวลผลการทำงานออกมาให้อยู่ในรูปแบบกราฟ เพื่อให้มองเห็นภาพรวมการทำงานของเครื่องจักรได้ง่ายยิ่งขึ้น และผู้ใช้งานต้องสามารถดาวน์โหลดข้อมูลต่างๆ เก็บไว้ได้ด้วย โดยเป็นไฟล์ CSV เพื่อให้สามารถนำไปใช้ในการวิเคราะห์ และวางแผนกระบวนการผลิตต่อไปในอนาคต

และส่วนสุดท้าย ส่วนที่ 4 Web Application “Neptune NlogConfiguration” ผู้ใช้งานต้องการโปรแกรมที่มีหน้าที่สำหรับการจัดการข้อมูลการติดตั้งต่างๆ เพื่อให้สามารถบันทึก แก้ไข และลบข้อมูลที่เกี่ยวข้องกับการตั้งค่าของระบบได้

จากการศึกษาข้อมูลเบื้องต้นเกี่ยวกับระบบและการศึกษาความต้องการของผู้ใช้งานระบบ ผู้จัดทำนั้นได้วางแผนการดำเนินงานทั้งหมด ตั้งแต่วันที่ 1 มิถุนายน 2559

ถึงวันที่ 30 พฤศจิกายน 2559 เป็นระยะเวลา 6 เดือนดังต่อไปนี้

เดือนมิถุนายน เป็นช่วงของการศึกษาระบบเดิม ศึกษาเกี่ยวกับฮาร์ดแวร์ และซอฟต์แวร์ต่างๆ ของเครื่องจักร

เดือนกรกฎาคม เป็นช่วงของการพัฒนา Web Application “Neptune Status” เพื่อให้ได้ซอฟต์แวร์ที่ตรงตามความต้องการของผู้ใช้งาน

เดือนสิงหาคม เป็นช่วงของการพัฒนา Web Application “Neptune Downtime” เพื่อให้ได้ซอฟต์แวร์ที่ตรงตามความต้องการของผู้ใช้งาน

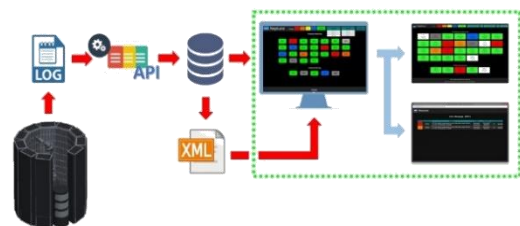
เดือนกันยายน เป็นช่วงของการพัฒนา Web Application “Neptune Report” เพื่อให้ได้ซอฟต์แวร์ที่ตรงตามความต้องการของผู้ใช้งาน

เดือนตุลาคม เป็นช่วงของการพัฒนา Web Application “Neptune NlogConfiguration” เพื่อให้ได้ซอฟต์แวร์ที่ตรงตามความต้องการของผู้ใช้งาน

เดือนพฤศจิกายน เป็นช่วงของการทดสอบและติดตั้งระบบ เพื่อทดลองใช้งานจริง ปรับปรุง แก้ไข และเพิ่มคุณสมบัติเพิ่มเติม

3.2. การวิเคราะห์ระบบ

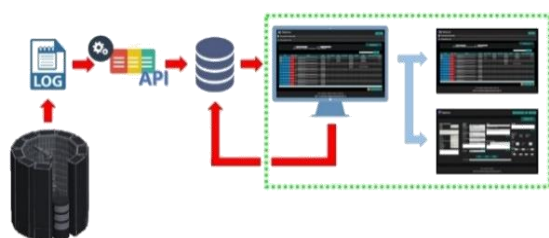
ทางผู้จัดทำได้ทำการวิเคราะห์โครงสร้างเบื้องต้นของระบบ ที่ใช้ในการพัฒนางานแต่ละส่วน โดยสรุปออกมาในรูปโครงสร้างดังต่อไปนี้



รูปที่ 3. โครงสร้างการพัฒนาโปรแกรม “Neptune Status”

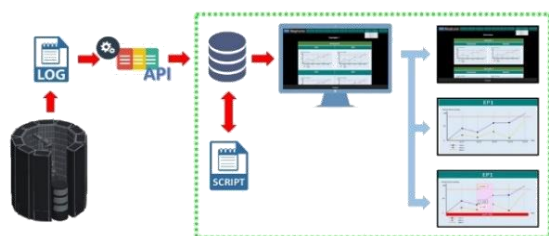
จากรูปที่ 3 โครงสร้างการพัฒนาโปรแกรม “Neptune Status” เมื่อเครื่องจักร มีการทำงานผิดพลาดเกิดขึ้น เครื่องจักรจะทำการสร้างข้อมูลความผิดพลาดออกมาในรูปแบบ Log File หลังจากที่มีข้อมูลถูกสร้างขึ้นแล้ว ระบบจะมี API ในการดึงข้อมูล Log File เพื่อนำมาผ่านกระบวนการแปลงข้อมูล ให้อยู่ในรูปแบบข้อมูลที่สามารถทำความเข้าใจได้ง่ายยิ่งขึ้น จากนั้นข้อมูลจะถูกเพิ่มลงระบบฐานข้อมูล เพื่อใช้ในการพัฒนาโปรแกรม “Neptune Status” ซึ่งการพัฒนาโปรแกรมนี้นี้ จะมีการใช้ข้อมูลจากสองส่วนด้วยกัน คือ ส่วนแรก จะติดต่อกับ

ระบบฐานข้อมูลโดยตรง เพื่อใช้ในส่วนอธิบายรายละเอียดของปัญหาต่างๆ และส่วนที่สอง คือ ส่วนที่จะกลั่นกรองเฉพาะข้อมูลที่สำคัญๆ ในการแจ้งเตือน โดยข้อมูลที่ถูกลั่นกรองออกมานั้น จะอยู่ในรูปแบบโครงสร้างไฟล์ XML เพื่อลดภาระการใช้งานกับระบบฐานข้อมูลโดยตรง เนื่องจากต้องมีการใช้งานข้อมูลในส่วนการแจ้งเตือน ทุกๆ 1 วินาที ซึ่งไฟล์ XML ทางคณะผู้จัดทำ จะกำหนดให้มีการปรับปรุงข้อมูล ทุกๆ ครั้ง ที่มีการทำงานผิดพลาดของเครื่องจักร เพื่อลดภาระการทำงานระหว่างไฟล์ XML กับ ระบบฐานข้อมูลให้น้อยลงยิ่งขึ้น



รูปที่ 4. โครงสร้างการพัฒนาโปรแกรม “Neptune Downtime”

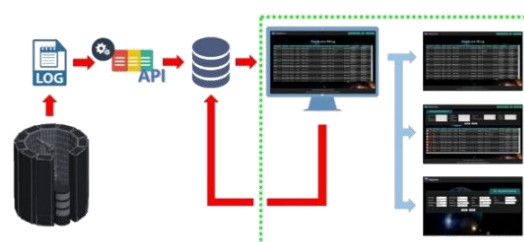
จากรูปที่ 4 โครงสร้างการพัฒนาโปรแกรม “Neptune Downtime” เมื่อเครื่องจักร มีการทำงานผิดพลาดเกิดขึ้น เครื่องจักรจะทำการสร้างข้อมูลความผิดพลาดออกมาในรูปแบบ Log File หลังจากข้อมูลที่ถูกรวบรวมขึ้นแล้ว ระบบจะมี API ในการดึงข้อมูล Log File เพื่อนำมาผ่านกระบวนการแปลงข้อมูล ให้อยู่ในรูปแบบข้อมูลที่สามารถทำความเข้าใจได้ง่ายยิ่งขึ้น จากนั้น ข้อมูลจะถูกเพิ่มลงระบบฐานข้อมูล เพื่อใช้ในการพัฒนาโปรแกรม “Neptune Downtime” ซึ่งจะมีการติดต่อกับระบบฐานข้อมูล เพื่อแสดงรายละเอียดการทำงานต่างๆ ของเครื่องจักร และหากมีการแก้ไขหรือลบข้อมูลจากโปรแกรม การกระทำต่างๆ เหล่านั้น จะถูกส่งไปปรับปรุงที่ระบบฐานข้อมูล เพื่อให้ได้ข้อมูลตามที่ผู้ใช้งานต้องการ



รูปที่ 5. โครงสร้างการพัฒนาโปรแกรม “Neptune Report”

จากรูปที่ 5 โครงสร้างการพัฒนาโปรแกรม “Neptune Report” เมื่อเครื่องจักร มีการทำงานผิดพลาดต่างๆ เกิดขึ้น

เครื่องจักรจะทำการสร้างข้อมูลความผิดพลาดออกมาในรูปแบบ Log File หลังจากข้อมูลที่ถูกรวบรวมขึ้นแล้ว ระบบจะมี API ในการดึงข้อมูล Log File เพื่อนำมาผ่านกระบวนการแปลงข้อมูล ให้อยู่ในรูปแบบข้อมูลที่สามารถทำความเข้าใจได้ง่ายยิ่งขึ้น จากนั้น ข้อมูลจะถูกเพิ่มลงระบบฐานข้อมูล เพื่อใช้ในการพัฒนาโปรแกรม “Neptune Report” ซึ่งจะมีการติดต่อกับระบบฐานข้อมูล เพื่อแสดงรายละเอียดการทำงานต่างๆ ของเครื่องจักรในรูปแบบกราฟ เพื่อให้เห็นประสิทธิภาพการทำงานได้ง่ายยิ่งขึ้น และทางระบบจะมีการสั่งให้โปรแกรม Script ทำงานตามเวลาที่กำหนด เพื่อทำการรวบรวมข้อมูลประจำวัน ประจำเดือน และประจำปี เป็นต้น



รูปที่ 6. โครงสร้างการพัฒนา “Neptune NLogConfiguration”

จากรูปที่ 6 โครงสร้างการพัฒนาโปรแกรม “Neptune NLogConfiguration” เมื่อเครื่องจักร มีการทำงานผิดพลาดเกิดขึ้น เครื่องจักรจะทำการสร้างข้อมูลความผิดพลาดออกมาในรูปแบบ Log File หลังจากข้อมูลที่ถูกรวบรวมขึ้นแล้ว ระบบจะมี API ในการดึงข้อมูล Log File เพื่อนำมาผ่านกระบวนการแปลงข้อมูล ให้อยู่ในรูปแบบข้อมูลที่สามารถทำความเข้าใจได้ง่ายยิ่งขึ้น จากนั้น ข้อมูลจะถูกเพิ่มลงระบบฐานข้อมูล เพื่อใช้ในการพัฒนาโปรแกรม “Neptune NLogConfiguration” ซึ่งจะมีการติดต่อกับระบบฐานข้อมูล เพื่อแสดงรายละเอียดการทำงานในส่วนต่างๆ ของเครื่องจักร และหากมีการแก้ไขหรือลบข้อมูลจากโปรแกรม การกระทำเหล่านั้น จะถูกส่งไปปรับปรุงที่ระบบฐานข้อมูล เพื่อให้ได้ข้อมูลตามที่ผู้ใช้งานต้องการ

3.3. การออกแบบระบบ

การพัฒนากระบวนการแจ้งเตือนการทำงานผิดพลาดของเครื่องทดสอบประสิทธิภาพฮาร์ดดิสก์ ทางผู้จัดทำได้มีการออกแบบโปรแกรมต่างๆ เป็น 3 ส่วนใหญ่ๆ ได้แก่

ส่วนที่หนึ่ง คือ การออกแบบผังงาน (Flowchart) เพื่อให้สามารถทราบขั้นตอนการพัฒนาโปรแกรมเบื้องต้น ซึ่งเป็นการวางแผนการพัฒนาโปรแกรมให้สามารถพัฒนาได้ง่าย และรวดเร็วมากยิ่งขึ้น

ส่วนที่สอง คือ การออกแบบส่วนต่อประสานผู้ใช้ เพื่อให้ผู้ใช้งาน ได้ทราบถึงส่วนในการรับข้อมูล และส่วนในการแสดงผลข้อมูล ว่าตรงตามความต้องการผู้ใช้งานหรือไม่ เพื่อให้การพัฒนาโปรแกรม สามารถตอบสนองความต้องการของผู้ใช้งานได้มากที่สุด

ส่วนที่สาม คือ ส่วนของการออกแบบตารางจัดเก็บข้อมูลต่างๆ เพื่อให้มีการจัดเก็บข้อมูลอย่างเป็นระเบียบ สามารถใช้งานได้ง่าย และมีประสิทธิภาพ

3.4. การพัฒนาและติดตั้งระบบ

Web Application “Neptune Status” ผู้จัดทำได้พัฒนาด้วย ภาษา PHP, HTML, CSS, XML และ JavaScript มีการใช้งาน ฟังก์ชัน AJAX เพื่อให้สามารถรองรับการเปลี่ยนแปลงของข้อมูล แบบ Real Time โดยไม่ต้องทำการ Reload หน้า Web Application ใหม่ ซึ่งจะทำให้ Web Application พร้อมรองรับทุกการเปลี่ยนแปลงของข้อมูลได้ตลอดเวลา

Web Application “Neptune Downtime” ทางผู้จัดทำได้พัฒนาด้วย ภาษา PHP, HTML, CSS และ JavaScript มีการเลือกใช้ฟังก์ชันการทำงานของ Data Table ซึ่งจะทำให้การพัฒนาส่วนแสดงผลข้อมูลในรูปแบบตาราง ใช้งานได้ง่ายมากยิ่งขึ้น

Web Application “Neptune Report” ทางผู้จัดทำได้พัฒนาด้วย ภาษา PHP, HTML, CSS และ JavaScript โดยมีการเรียกใช้ API ในการสร้างกราฟของ Amcharts ซึ่งเป็น API ที่ใช้งานง่าย และมีรูปแบบกราฟหลากหลายให้เลือกใช้งานอีกด้วย

Web Application “Neptune NlogConfiguration” ทางผู้จัดทำ ได้พัฒนาด้วย ภาษา PHP, HTML, CSS และ JavaScript มีการใช้งานฟังก์ชัน AJAX เพื่อให้ข้อมูล สามารถรองรับการเปลี่ยนแปลงแบบ Real Time โดยไม่ต้องทำการ Reload หน้า Web Application ใหม่ทั้งหมด ซึ่งจะทำให้ Web Application พร้อมรับทุกการเปลี่ยนแปลงของข้อมูล เช่นเดียวกับการพัฒนา Web Application “Neptune Status” ที่ได้อธิบายมาก่อนหน้านี้

การติดตั้งระบบ ผู้จัดทำได้พัฒนาให้สามารถใช้งานได้ทั้งระบบปฏิบัติการ Windows และระบบปฏิบัติการ Linux แต่ในครั้งนี้ ผู้จัดทำได้ทำการติดตั้งโปรแกรมบนระบบปฏิบัติการ CentOS7 ใช้โปรแกรม Apache เวอร์ชัน 2.4.6, PHP เวอร์ชัน 5.4.16 และ MySQL เวอร์ชัน 5.5.47-Maria DB จากนั้นจึงนำ Source Code ทั้งหมด ไปวางในลงโฟลเดอร์ที่ใช้สำหรับ Run โปรแกรมภาษา PHP จากนั้นทำการสร้างระบบฐานข้อมูล และแก้ไขการเชื่อมต่อต่างๆ ที่ถูกต้องเพียงเท่านี้ ก็จะสามารถติดตั้งระบบให้สามารถใช้งานได้โดยสมบูรณ์

3.5. การบำรุงรักษาระบบ

การบำรุงรักษาในส่วน Web Application “Neptune Status” หลังจากที่ได้ทำการทดลอง ใช้งานโปรแกรมมาระยะหนึ่งพบว่า เมื่อผู้ใช้งานเปิดโปรแกรมทั้งไว้สักครู่หนึ่ง การกระพริบสลับสีระหว่างสีขาวกับสีแดง และการส่งเสียงแจ้งเตือนของโปรแกรม จะเร็วบ้างช้าบ้าง ไม่มีความสม่ำเสมอ ทางผู้จัดทำ จึงได้ทำการแก้ไขปัญหของโปรแกรม จนสามารถทำให้การกระพริบสลับสีระหว่างสีขาวกับสีแดงและการส่งเสียงแจ้งเตือนข้อผิดพลาดต่างๆ มีความสม่ำเสมอมากยิ่งขึ้น

การบำรุงรักษาในส่วน Web Application “Neptune Downtime” หลังจากที่ได้ทำการทดลอง ใช้งานโปรแกรมมาระยะหนึ่งพบว่า ในส่วนของ Downtime List ช่องการค้นหาข้อมูล ไม่สามารถใช้งานได้ ในกรณีที่ทำการค้นหาแล้วไม่พบข้อมูล โปรแกรมจะไม่สามารถเรียกใช้ข้อมูล ให้กลับมาทำงานปกติได้ ดังนั้นผู้จัดทำจึงเปลี่ยนมาใช้ฟังก์ชันการทำงานของ Data Table แทน เนื่องจากมีฟังก์ชันการทำงานที่หลากหลาย และสามารถตอบสนองความต้องการของผู้ใช้งานได้มากยิ่งขึ้น

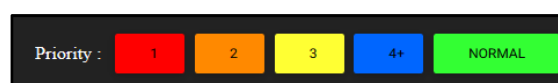
การบำรุงรักษาในส่วน Web Application “Neptune Report” หลังจากที่ได้ทำการทดลอง ใช้งานโปรแกรมมาระยะหนึ่งพบว่า ผู้ใช้งานต้องการให้การแสดงผลของกราฟแกน Y มีช่วงข้อมูลที่ถี่มากขึ้น ทางผู้จัดทำจึงได้ทำการปรับปรุงแก้ไขในส่วนนี้ เพื่อให้ได้กราฟในรูปแบบที่ผู้ใช้งานต้องการมากที่สุด

การบำรุงรักษาในส่วน ของ Web Application “Neptune NlogConfiguration” หลังจากที่ได้ทำการทดลอง ใช้งานโปรแกรมมาระยะหนึ่ง ไม่พบปัญหาจากการใช้งาน โปรแกรม ผู้จัดทำจึงไม่ได้บำรุงรักษาอะไรเพิ่มเติมมากนัก มีเพียงแค่การแก้ไขข้อความต่างๆ เพื่อให้ผู้ใช้งานเข้าใจได้ง่ายมากยิ่งขึ้น

4. ผลการดำเนินงาน

จากการดำเนินงานทั้งหมดทุกๆ โปรแกรมสามารถทำงานได้ตามความต้องการของผู้ใช้งาน มีฟังก์ชันการทำงานต่างๆ และส่วนต่อประสานผู้ใช้ต่างๆ ที่ง่ายต่อความเข้าใจ ใช้งานจัดวางที่เข้าใจง่าย มีระเบียบ แต่อย่างไรก็ตาม โปรแกรมต่างๆ ถูกพัฒนามาเพื่อใช้งานเฉพาะด้าน ดังนั้นผู้ใช้งานจำเป็นต้องศึกษาวิธีใช้งานก่อนเสมอ เพื่อให้โปรแกรมสามารถทำงานได้อย่างถูกต้องและสมบูรณ์ที่สุด ซึ่งแต่ละโปรแกรมมีส่วนต่อประสานผู้ใช้ ดังต่อไปนี้

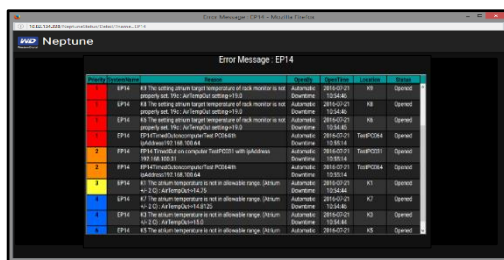
4.1. Web Application “Neptune Status”



รูปที่ 7. แถบสีที่ใช้กำหนดลำดับความสำคัญของปัญหาต่างๆ



รูปที่ 8. โปรแกรมเริ่มต้น Web Application “Neptune Status”



รูปที่ 9. รายละเอียดการทำงานผิดพลาดของเครื่องจักรแต่ละตัว

4.3. Web Application “Neptune Report”

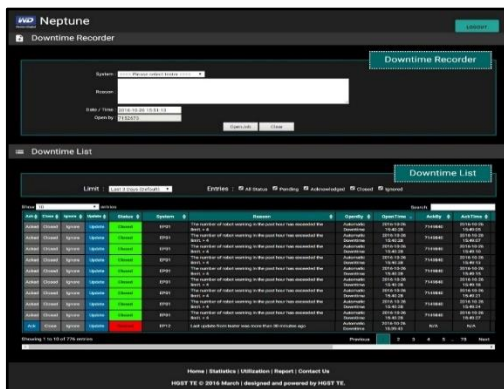


รูปที่ 12. กราฟรายงานข้อมูลภาพรวมการทำงานของเครื่องจักร



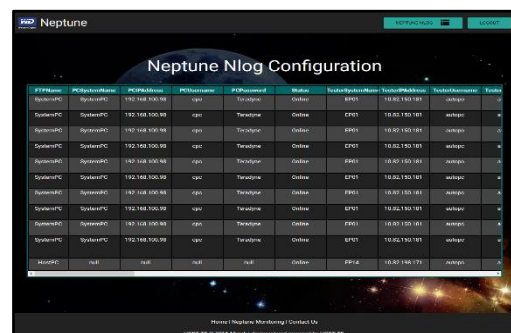
รูปที่ 13. ส่วนสำหรับดาวน์โหลดข้อมูลในรูปแบบไฟล์ CSV

4.2. Web Application “Neptune Downtime”

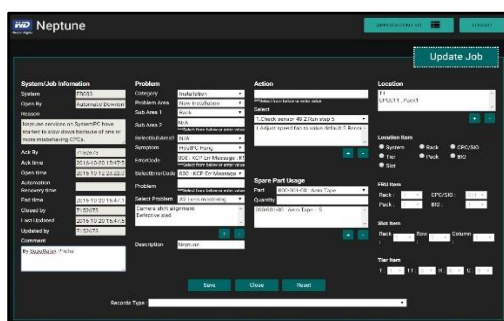


รูปที่ 10. โปรแกรมเริ่มต้น Web Application “Neptune Downtime”

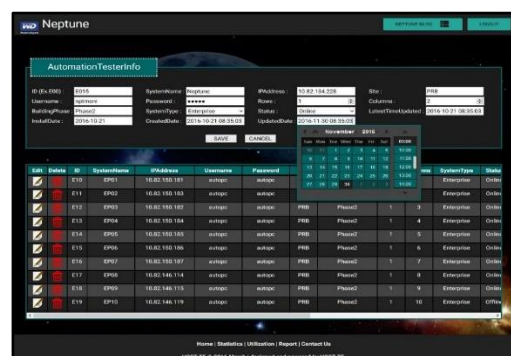
4.4. Web Application “Neptune NlogConfiguration”



รูปที่ 14. โปรแกรมเริ่มต้น “Neptune NlogConfiguration”



รูปที่ 11. ส่วนสำหรับการแก้ไขข้อมูลปัญหาต่างๆ



รูปที่ 15. ตัวอย่างการเพิ่มข้อมูลในส่วน “AutomationTesterInfo”

5. สรุปผล

จากการพัฒนาทุกส่วนของระบบ โปรแกรมต่างๆ สามารถช่วยอำนวยความสะดวกสบายให้แก่ผู้ใช้งานได้มากขึ้น เช่น การแจ้งเตือนข้อผิดพลาด การจัดการกับข้อมูล และอื่นๆ มีการออกแบบส่วนต่อประสานผู้ใช้ ให้อยู่ในรูปแบบที่เข้าใจง่าย และสะดวกมากยิ่งขึ้น แต่เนื่องด้วยระบบ ถูกพัฒนาเพื่อใช้กับเครื่องจักรที่เฉพาะเจาะจงหากผู้ใช้งานไม่ทราบถึงส่วนต่างๆ ของเครื่องจักร อาจทำให้ผู้ใช้งานเข้าใจได้ยาก และส่งผลให้ใช้งานผิดรูปแบบไม่ตรงตามความเป็นจริง ดังนั้น ผู้ใช้งานระบบ จึงจำเป็นต้องศึกษาวิธีการใช้งานของระบบให้เข้าใจเสียก่อน เพื่อให้การใช้งานเป็นไปอย่างถูกต้องและสมบูรณ์ ซึ่งจะช่วยให้เกิดประสิทธิภาพในการใช้งานเพิ่มมากขึ้นได้อีกด้วย

6. ผลที่ได้รับหรือประโยชน์จากการพัฒนา

จากการพัฒนาระบบทั้งหมด ทำให้ได้ซอฟต์แวร์หรือโปรแกรมที่สามารถใช้งาน และเข้าใจได้ง่ายมากยิ่งขึ้น สามารถเอื้ออำนวยความสะดวกสบายให้กับผู้ใช้งานได้อย่างแท้จริง เนื่องจากเป็นการพัฒนาภายใต้คำแนะนำของผู้ใช้งานโดยตรง จึงได้ซอฟต์แวร์ที่ตรงตามความต้องการของผู้ใช้งานมากที่สุด อีกทั้งผลจากการพัฒนาระบบ นอกจากจะได้ระบบที่สามารถทำการจัดการกับข้อมูล และแจ้งเตือนปัญหาเมื่อพบข้อผิดพลาดจากการทำงานของเครื่องจักรแล้ว ยังสามารถขยายขีดจำกัด การใช้ข้อมูลจากเครื่องจักร เพื่อนำมาเป็นแนวทางในการพัฒนาซอฟต์แวร์หรือนวัตกรรมใหม่ๆ ต่อไปในอนาคตได้อีกด้วย

ระบบแจ้งเตือนการทำงานผิดพลาดของเครื่องทดสอบประสิทธิภาพฮาร์ดดิสก์ มีแผนการปรับเปลี่ยนมาใช้ระบบใหม่นี้ ในช่วงไตรมาสสุดท้ายของปี 2559 โดยบริษัทจะเริ่มนำร่องติดตั้งระบบใหม่ควบคู่ไปกับการทำงานของระบบเดิมก่อน เพื่อทดสอบประสิทธิภาพการทำงานของระบบใหม่ให้แน่ชัดว่า สามารถทำงานได้อย่างถูกต้องและสมบูรณ์ที่สุด ซึ่งทางบริษัท เอชจีเอสที (ประเทศไทย) จำกัด คาดว่า ระบบใหม่นี้ จะสามารถช่วยลดค่าใช้จ่ายเกี่ยวกับค่าบำรุงรักษาทางซอฟต์แวร์ของระบบเดิมได้ประมาณ 7,251,769 บาทต่อปี ซึ่งตัวเลขดังกล่าวนี้ จะมาจากการคำนวณของสมการง่าย ๆ ดังต่อไปนี้

$$A * B = C \quad (1)$$

A แทน ค่าบำรุงรักษาทางซอฟต์แวร์แต่ละตัว

B แทน จำนวนเครื่องจักรทั้งหมด

C แทน ค่าบำรุงรักษาทางซอฟต์แวร์ทั้งหมด

จะได้ $250,061 * 29 = 7,251,769$ บาทต่อปี

7. กิตติกรรมประกาศ

การพัฒนาระบบแจ้งเตือนการทำงานผิดพลาดของเครื่องทดสอบประสิทธิภาพฮาร์ดดิสก์ สำเร็จลงไปได้ด้วยดี ทางผู้จัดทำต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์สุปิตี กุลจันทร์ ซึ่งเป็นอาจารย์ที่ปรึกษา ที่ได้กรุณาให้คำปรึกษา ทั้งในเรื่องการทำงาน การใช้ชีวิต การปรับตัว และคอยแนะนำแนวทางในการพัฒนาระบบนี้ จนสำเร็จลงไปได้ด้วยดี

ในส่วนของหน่วยงานกรณีศึกษา ทางผู้จัดทำ ขอขอบคุณ คุณภัทรพล เวียงนาค และสมาชิกแผนก Test Engineering#5 ของบริษัท เอชจีเอสที (ประเทศไทย) จำกัด ที่ให้ความอนุเคราะห์ในการพัฒนาระบบ ให้ความรู้ ความเข้าใจ ทั้งในส่วนการทำงาน และในส่วนการใช้ชีวิตต่างๆ

สุดท้ายนี้ทางผู้จัดทำ ต้องขอขอบคุณบุคลากรและคณาจารย์ทุกท่าน รวมไปถึง พี่ๆ เพื่อนๆ ในสาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีและการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่คอยให้คำแนะนำ และคอยให้ความช่วยเหลือเสมอมา จนทำให้การพัฒนาระบบสำเร็จลงไปได้ด้วยดี

เอกสารอ้างอิง

- [1] ความแตกต่างของ Library, Framework, API คืออะไร. [ออนไลน์]. 31 มีนาคม 2555. [สืบค้นวันที่ 14 กุมภาพันธ์ 2560]. จาก www.thaiflashdev.com/home/index.php?topic=190
- [2] แนวคิดการเขียนโปรแกรม OOP. [ออนไลน์]. 13 มีนาคม 2556. [สืบค้นวันที่ 14 กุมภาพันธ์ 2560]. จาก srisunthorn.wordpress.com/2013/03/13/1056
- [3] Neptune Automated Test Platform. [ออนไลน์]. [สืบค้นวันที่ 14 กุมภาพันธ์ 2560]. จาก www.teradyne.com