Development of an Initial Prototype Application for a hardware simulator
June, 2010
CAPSTONE
iSTAR Simulator for CCURE9000
Development of an Initial Prototype Application for a hardware simulator
June, 2010
Who we are

• Tyco International's Software House technologies include the innovative C•CURE® 9000 security and event management system and the flagship C•CURE 800/8000 access control solution. Combined with a suite of powerful door controllers and the industry's first true multi-technology reader, Software House technologies are among the most powerful in the industry. Add an unsurpassed integration platform that allows customers to integrate seamlessly with critical business applications, including American Dynamics Intellex® digital video management systems, and it's easy to see why Software House solutions are the hands-down choice for security-critical applications. In fact, nearly half of the companies listed on the Fortune 500 use Software House branded solutions to help solve their most complex business operation challenges
Proposal

Problem Statement

- A need to have the ability to test the scalability of our access control system using simulation from small (less than 16 doors) to a large (more than 1000 doors) due to the amount of hardware, space, personnel, and time needed to do a good quality control before releasing the product for usage in the industry.
Proposed Solution:

- Create a software simulation tool that can emulate a real hardware controller(s) (iSTAR family) by sending real messages to the host as programmed. The simulator should also have the capability of receiving messages from the host and reacting to those message accordingly.

- The Simulator shall have a user friendly GUI that is intuitive and easy to use. This GUI should give the user the following high level capabilities:
  - Program the type of messages to generate.
  - The frequency of the messages (throttle).
  - Ability to acknowledge/reject messages and react to them.
  - Ability to gather metrics about response times and quality of service.

- The solution shall be very secure preventing this tool from running in a production site without explicit intent.
Skills

• Software House standard tools & Skills:
  – C# language as the standard language for development for all of the access control system. This will enable developers to look and learn from existing code samples and potential reuse of some existing proven code fragments.
  – Use Bug tracking (Lotus Notes and Microsoft Team System)
  – Use requirement tool – Borland Caliber
  – Use 6Sigma rally point process for Software Development life cycle management.
What you will learn

- Industry best practices for Software Development to include:
  - Requirements definition
  - Design specifications
  - Modular code design and development methodology
  - Low level driver solutions
  - User friendly Graphical User Interfaces (GUI) design
  - International translation implementation
  - Debugging, Optimizing and Scalability of applications
  - Coding standards and best practices
  - Team Work and development collaboration
  - Unit Testing and Quality Assurance approaches
  - Automation
Objective

• This project will get the team of developers to understand the messaging system between the hardware (iSTAR Family) and the host system.
• Get a better understand on the communication system using TCP/IP, and the .NET libraries including the robotic engine.
• Get more knowledge about how an access control system works.
• Build a user friendly GUI for user to configure and use the simulator.
• Use the 6Sigma process to take project from concept to finish.
• Work with a global Software Team that produces the leading access control products in the Industry.
• Understand full life cycle development challenges in real world deployments
Project Deliverables

• Refine the project requirements and develop the design specifications
  – Electrical Engineering NEU
    • Graphical User Interface
  – Industrial Design Engineering NEU / SJTU
• Develop prototype and test key concepts NEU / SJTU
  – e.g. Unit Test Cases, Performance Metrics, Throttling
• Perform demonstrations of the prototype to the Software House team (engineering, QA, Product management, and sales) and gather feedback and input
• Story board for the project
• Lessons learned and feedback on the whole process
  – What worked well
  – What needs improvement
  – Documentation and next steps for the product
# Project Timeline & progression

<table>
<thead>
<tr>
<th>Task</th>
<th>Time Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction of NEU Electrical Engineering team members</td>
<td>July 2010</td>
</tr>
<tr>
<td>Defining of GUI requirements by NEU Electrical Engineering team members</td>
<td>July – August 2010</td>
</tr>
<tr>
<td>Introduction of the NEU Industrial Engineering team members</td>
<td>September 2010</td>
</tr>
<tr>
<td>Defining of the simulator drivers requirements</td>
<td>September – December 2010</td>
</tr>
<tr>
<td>Introduction of the SJTU Industrial Engineering team members</td>
<td>January 2011</td>
</tr>
<tr>
<td>Implementation of prototype by NEU Electrical Engineering, NEU Industrial Engineering and SJTU Industrial Engineering</td>
<td>February 2011 – April 2011</td>
</tr>
<tr>
<td>Demonstration of prototype</td>
<td>May 2011</td>
</tr>
<tr>
<td>Final project report</td>
<td>April 2011</td>
</tr>
</tbody>
</table>