

Ingeniería de Software II



El Caso del Robot Asesino

Programador de Silicon Valley Acusado por Homicidio no Premeditado

El Error del Programa Causó la Muerte del Operador del Robot

Especial para el SENTINEL-OBSERVER de Silicon Valley

Jane McMurdock, Fiscal de la Ciudad de Silicon Valley, anunció en la fecha la acusación de Randy Samuels con los cargos de asesinato no premeditado. Samuels trabajaba como programador en Silicon Techtronics, Inc., una de las empresas más nuevas de Silicon Valley en la arena de la alta tecnología. El cargo involucra la muerte de Bart Matthews, quien fuera muerto el pasado mes de mayo por un robot de la línea de armado.

Matthews, quien trabajaba como operador de robot en Cybernetics, Inc., en Silicon Heights, fue aplastado y murió a consecuencias de ello, cuando el robot que estaba operando produjo un mal funcionamiento y comenzó a oscilar su “brazo” violentamente. El brazo del robot alcanzó a Matthews, arrojándolo contra una pared. Matthews murió en forma casi instantánea a causa de los golpes recibidos, en un caso que conmocionó e indignó a muchos de Silicon Valley. De acuerdo con la dictamen de los cargos, Samuels fue quien escribió la pieza del programa de computadora en particular, que fue la responsable de la falla del robot. “Hay una evidencia incriminatoria”, anunció triunfante McMurdock en una conferencia de prensa mantenida en la Corte.

“Tenemos la fórmula manuscrita, suministrada por el físico del proyecto, que se suponía tenía que programar Samuels. Pero negligentemente malinterpretó la fórmula, y esto llevó a una horrible muerte. La sociedad debe protegerse de los programadores que cometen errores descuidadamente o de lo contrario nadie estará a salvo, y menos que nadie nuestras familias e hijos”, dijo.

El SENTINEL-OBSERVER ha podido obtener una copia de la fórmula manuscrita en cuestión. En realidad, existen tres fórmulas similares, garabateadas en un papel amarillo de un block borrador tamaño oficio. Cada una de las fórmulas describe el movimiento del brazo del robot en una dirección: este-oeste, norte-sur y arriba-abajo.

El SENTINEL-OBSERVER mostró las fórmulas a Bill Park, profesor de Física en la Universidad de Silicon Valley. Éste confirmó que estas ecuaciones podían ser usadas para describir el movimiento del brazo de un robot.

El SENTINEL-OBSERVER mostró entonces el código del programa a Bill Park, escrito por el acusado en lenguaje C de programación. Preguntamos al Profesor Park, quien está muy familiarizado con éste y muchos otros lenguajes de programación, si el código era o no correcto para las fórmulas dadas del brazo del robot.

La respuesta del Profesor Park fue inmediata: “¡No puede ser! Parece que interpretó los puntos y de las fórmulas como barras y, e hizo lo mismo con las x y las z . Se suponía que tenía que usar las derivadas, pero en su lugar tomó los promedios!. Si me preguntan, ¡está muy claro que es culpable!”

El SENTINEL-OBSERVER no pudo contactar a Samuels para entrevistarlo. “Se encuentra profundamente deprimido por todo esto”, nos dijo su novia por teléfono. “Pero, Randy cree que va a aliviarse en cuanto pueda decir su versión de la historia.”

Los que Desarrollaron al “Robot Asesino” Trabajaron bajo una Enorme Presión

Especial para el SENTINEL-OBSERVER de Silicon Valley

*Silicon Valley, EEUU
por Mabel Muckraker*

El SENTINEL-OBSERVER tomó conocimiento hoy que Randy Samuels y otros que trabajaron en el proyecto del “Robot Asesino” en Silicon Techtronics, estuvieron bajo tremendas tensiones para finalizar el software del robot para el 1º de enero de este año. Según una fuente bien informada, los altos niveles gerenciales advirtieron a los integrantes del staff del proyecto que “rodarían cabezas” si no se cumplía el objetivo del 1º de enero.

Randy Samuels, programador de Silicon Techtronics, fue acusado la semana pasada bajo los cargos de asesinato no premeditado en el ahora famoso “caso del robot asesino”. Samuels escribió el software defectuoso que causó que el robot industrial de Silicon Techtronics, Robbie CX30, aplastara y lesionara fatalmente a su operador, Bart Matthews. Matthews era un operador de robot en Cybernetics, Inc. Conforme a la Fiscal de Silicon Valley, Jane McMurdock, Samuels malinterpretó la fórmula matemática, “volviendo al inofensivo Robbie un asesino”.

Nuestra fuente informada, quien desea mantenerse en el anonimato y a la que llamaremos “Marta” por el resto de este artículo, tiene un íntimo conocimiento de todos los aspectos del proyecto Robbie CX30. Marta dijo al SENTINEL-OBSERVER que existía una enorme fricción entre el Jefe de División Robótica, Ray Johnson y el Gerente del Proyecto Robbie CX30, Sam Reynolds. “Se odiaban a muerte” manifestó Marta al SENTINEL-OBSERVER en una entrevista exclusiva.

“Hacia junio del año pasado el proyecto se encontraba atrasado seis meses y Johnson se puso furioso. Había rumores de que echarían a toda la División Robótica, que él lideraba, si Robbie [el robot CX30] no daba muestras de ser un éxito comercial. Él (Johnson) llamó a Sam (Reynolds) a su oficina y realmente lo destruyó. Quiero decir, uno podía oír los gritos desde el fondo de la oficina. Johnson le dijo a Sam que terminara el proyecto para el primero de enero o de lo contrario “rodarían cabezas”.

“Yo no estoy diciendo que Johnson le ordenara a Sam acortar camino”, agregó Marta. “Creo que la idea de cortar camino estaba implícita. El mensaje fue: “acortá camino si querés mantener tu puesto”.

De acuerdo con documentos provistos por Marta al SENTINEL-OBSERVER, el 12 de junio del año pasado fueron agregados al proyecto Robbie CX30 veinte nuevos programadores. Esto ocurrió algunos días después de la tormentosa reunión entre Johnson y Reynolds que Marta contó.

De acuerdo a Marta, los nuevos contratados eran un desastre. “Johnson, unilateralmente, hizo los arreglos de estas contrataciones, seguramente desviando recursos de otros aspectos del proyecto Robbie (CX30). Reynolds se oponía con vehemencia a esto. Johnson sólo conocía acerca de la fabricación de hardware. Esa era su especialidad. No pudo haber entendido las dificultades que nosotros estábamos teniendo con el software de la robótica. Usted no puede acelerar un proyecto de software agregando más gente. No es como en una línea de montaje”.

Según Marta y otras fuentes dentro del proyecto, la contratación de estos nuevos veinte programadores llevó a que se hiciera una reunión entre Johnson, Reynolds y todos los integrantes del proyecto de software del Robbie CX30. “Esta vez fue Sam (Reynolds) el que se puso furioso. Se quejó de que el proyecto no necesitaba más gente. Sostuvo que el problema principal era que Johnson y otros miembros a nivel directivo no entendían que el Robbie CX30 era fundamentalmente diferente de versiones anteriores del robot”. Estas fuentes dijeron al SENTINEL-OBSERVER que los nuevos empleados no estaban totalmente integrados al proyecto, aún seis meses después de su ingreso, cuando diez robots Robbie CX30, incluido al robot que mató a Bart Matthews, ya habían sido despachados. Según Marta, “Sam sólo quería mantener las cosas lo más simples posible. No quería que el nuevo personal complicara las cosas. “Se pasaron seis meses leyendo manuales. La mayoría de los empleados nuevos no sabían nada de robots y Sam no estaba como para perder su tiempo tratando de enseñarles”. Según Marta, la reunión del 12 de junio se hizo famosa en la corporación Silicon Techtronics porque fue en esa reunión donde Ray Johnson anunció su “Teoría Ivory Snow [“no existe el blanco perfecto, o bien, no hay blanco más blanco que el blanco nieve”] de diseño y desarrollo de software. De acuerdo a Marta, “Ray (Johnson) nos dio una gran presentación en multi-media, con diapositivas y todo. La esencia de esta “Teoría Ivory Snow” es simplemente que el blanco nieve es 99,44 porciento puro y que no hay razón por la que el software de robótica deba ser más puro que esto. Dijo repetidas veces que ‘El software perfecto era un oxímoron’ ”.

Marta y otros personajes anónimos que se acercaron con información, retrataron a Johnson como un gerente con una desesperada necesidad de ser ayudado por un éxito en el proyecto. Versiones anteriores de Robbie, CX10 y CX20, fueron experimentales en naturaleza y nadie esperaba que fueran éxitos comerciales. De hecho, la División Robótica de Silicon Techtronics estaba operando con sus finanzas en rojo desde su concepción seis años atrás. O triunfaba el CX30 o Silicon Techtronics quedaría fuera del negocio de robótica industrial. “Los robots Robbie anteriores tuvieron mucha prensa, especialmente acá en Silicon Valley”, dijo otra fuente que también quiere permanecer anónima. “Robbie CX30 iba a capitalizarse con la buena publicidad generada por los proyectos anteriores. Lo único es que Robbie CX30 era más revolucionario de lo que Johnson quería admitir. CX30 representaba un paso gigante hacia adelante en términos de sofisticación. Había muchísimas preguntas acerca de los parámetros industriales en los que debería trabajar el CX30. Mucho de lo que debía ejecutar era completamente nuevo, pero Johnson nunca lo pudo entender. Él sólo nos veía como unos perfeccionistas. Uno de sus dichos favoritos era ‘La perfección es la enemiga de lo bueno’ ”.

Los Compañeros Acusan: el Programador del “Robot Asesino” se Creía una Estrella

Especial para el SENTINEL-OBSERVER de Silicon Valley

*Silicon Valley, EEUU
por Mabel Muckraker*

Randy Samuels, el que fuera programador de Silicon Techtronics que fue acusado por escribir el software que causó el horrible incidente del “robot asesino” el pasado mes de mayo, era aparentemente una ‘prima donna’ que encontraba muy difícil aceptar críticas, aseguraron hoy varios compañeros de trabajo.

En una rueda de prensa con varios compañeros de trabajo de Samuels en el proyecto del “robot asesino”, el SENTINEL-OBSERVER pudo obtener importantes revelaciones acerca de la psiquis del hombre que puede haber sido criminalmente responsable de la muerte de Bart Matthews, operador de robot y padre de tres criaturas.

Con el permiso de los entrevistados, el SENTINEL-OBSERVER permitió a la profesora Sharon Skinner del Departamento de Psicología de Software en la Universidad de Silicon Valley, escuchar una grabación de la entrevista. La Profesora Skinner estudia la psicología de los programadores y otros factores psicológicos que tienen un impacto en el proceso de desarrollo del software.

“Estaría de acuerdo con la mujer que lo llamó ‘prima donna’”, explicó la Profesora Skinner. “Este es un término utilizado para referirse a un programador que simplemente no puede aceptar las críticas, o más precisamente, no puede aceptar su propia falibilidad”.

“Randy Samuels tiene lo que nosotros, psicólogos de programadores, llamamos una personalidad orientada hacia una tarea, lindando con una personalidad orientada hacia sí mismo. Le gusta poder completar cosas, pero su ego está muy densamente involucrado en su trabajo. En el mundo de la programación esto se lo considera un “no, no”, agregó el Profesor Skinner en su oficina tapizada de libros.

La Profesora Skinner continuó explicando algunos hechos adicionales sobre equipos de programación y personalidades del programador. “Básicamente, hemos encontrado que un buen equipo de programación requiere de una mezcla de tipos de personalidad, incluyendo a una persona que esté orientada hacia la interacción, que saca una enorme satisfacción del hecho de trabajar con otra gente, alguien que pueda ayudar a mantener la paz y a que las cosas se muevan en una dirección positiva. Muchos programadores están orientados hacia lo que es la tarea, y esto puede ser problemático si se tiene un equipo donde todos son de este modo.”

Los compañeros de trabajo de Samuels se mostraron muy reticentes a culpar a alguien por el desastre del robot, pero cuando se los presionó para que comentaran sobre la personalidad de Samuels y sus hábitos laborales, surgieron varios hechos importantes. Samuels trabajaba en un equipo formado por aproximadamente una docena de analistas, programadores y testers de software. (Esto no incluye a veinte programadores que fueron incorporados posteriormente y que nunca llegaron a estar activamente involucrados en el desarrollo del software de la robótica). Si bien cada individuo del equipo poseía una especialidad, casi todos estaban comprometidos en todo el proceso de software de principio a fin.

“Sam Reynolds tenía un background en procesamiento de datos. Dirigió unos cuantos proyectos de software de esa naturaleza”, dijo uno de los integrantes del equipo, refiriéndose al gerente del proyecto Robbie CX30. “Pero su rol en el proyecto era más que nada de líder. Asistía a todas las reuniones importantes y lo mantenía a Ray (Ray Johnson, el Jefe de División Robótica) sobre nuestras espaldas lo más posible.” Sam Reynolds, como ya fuera informado en el SENTINEL-OBSERVER de ayer, se encontraba bajo una severa presión para lograr producir un robot Robbie CX30 operativo para el 1 de enero de este año. Sam Reynolds no pudo ser ubicado para entrevistarlos ya sea sobre su rol en el incidente o sobre Samuels y sus hábitos en el trabajo.

“Éramos un equipo democrático, a excepción del liderazgo provisto por Sam (Reynolds)”, observó otro miembro del equipo. En el mundo del desarrollo de software, un equipo democrático es un equipo en donde todos los miembros de éste tienen un decir igual en el proceso de toma de decisiones. “Desafortunadamente, nosotros éramos un equipo de individualistas muy ambiciosos, muy talentosos - si debo referirme a mí mismo - y muy opinadores. Randy (Samuels) era justo el peor del grupo. Lo que quiero decir es que teníamos, por ejemplo, a dos chicos y a una chica con masters de CMU, y no eran tan arrogantes como Randy. CMU significa Universidad de Carnegie-Mellon, una líder nacional en enseñanza de ingeniería de software.

Un compañero comentó sobre un incidente que Samuels causó en una reunión de Quality Assurance. Esta reunión involucraba a Samuels y a tres revisores de un módulo de software que Samuels había diseñado e implementado. Tales reuniones son llamadas “revisiones de código”. Uno de los revisores mencionó que Samuels había usado un algoritmo sumamente ineficiente para lograr un determinado resultado y Samuels “se puso todo colorado”. Empezó a gritar una sarta de obscenidades y después se levantó y se fue. Nunca regresó.

“Le enviamos un memo con un algoritmo más rápido y a su tiempo usó este algoritmo en su módulo”, agregó el colega.

El módulo de software del incidente de la reunión de Quality Assurance fue el primero en ser identificado como una falla en el “asesino” del operador de robot. No obstante, este colega se apuró a señalar que la eficacia del algoritmo no era un tópico en el mal funcionamiento del robot. Era sólo que Randy hacía muy difícil para la gente el poderle comunicar las observaciones. Se tomaba todo muy a pecho. Se graduó con el puntaje más alto de su clase y luego se recibió con honores en ingeniería de software en Purdue. Definitivamente es muy inteligente.”

“Randy, en su pared, tiene este inmenso cartel hecho en Banner”, continuó este colega, “Decía, “DENME LA ESPECIFICACIÓN Y LES DARÉ EL PROGRAMA DE COMPUTACIÓN”. Ese es el tipo de arrogancia que tenía y también demuestra que tenía muy poca paciencia para desarrollar y verificar las especificaciones. Amaba el aspecto de solucionar el problema, la programación propiamente dicha”. No pareciera que Randy Samuels quedó atrapado en el espíritu de la “programación sin egolatría”, observó la Profesora Skinner cuando escuchó esta parte de la entrevista con los colegas de trabajo de Samuels. “La idea de una programación sin egocentrismo es que el producto de software pertenece al equipo y no a los programadores individuales. La idea es estar abierto a las críticas y estar menos atado al trabajo propio. Ciertamente que la tarea de revisión de código es coherente con esta filosofía en general.” Una colega habló acerca de otro aspecto de la personalidad de Samuels: su capacidad de ayuda. “Randy odiaba las reuniones, pero era muy bueno con las relaciones de uno a uno. Siempre estaba ansioso por ayudar. Recuerdo una vez que me encontraba encerrada en un camino sin salida y él, en vez de tan sólo señalarme la dirección correcta, se hizo cargo del problema y lo resolvió él mismo. Se pasó cerca de cinco días completos en mi problema”. “Por supuesto que mirado en retrospectiva, hubiera sido mejor para el

pobre Sr. Matthews y su familia que Randy se hubiese dedicado tan sólo a sus propias cosas”, agregó luego de una larga pausa.

El Proyecto del “Robot Asesino” Controvertido desde el Vamos

Bandos Enfrentados por el Modo en que Debía Proseguir el Proyecto

Especial para el SENTINEL-OBSERVER de SILICON VALLEY

***Silicon Valley, EEUU
por Mabel Muckraker***

Dos grupos, comprometidos con diferentes filosofías de desarrollo de software, casi se enfrentan violentamente durante las reuniones iniciales de planeamiento para el Robbie CX30, el robot de Silicon Techtronics que mató a un obrero de la línea de ensamble el pasado mes de mayo. Estaba en cuestionamiento si el proyecto Robbie CX30 debía proseguir de acuerdo con el “modelo de cascada” o el “modelo de prototipo”.

El modelo de cascada y el de prototipo son dos métodos comunes de organizar un proyecto de software. En el modelo de cascada, el proyecto de software pasa a través de etapas definidas de desarrollo. La primera etapa es el análisis de los requerimientos y especificaciones, durante la cual se intenta arribar a un acuerdo en cuanto a la funcionalidad detallada del sistema. A medida que el proyecto pasa de una etapa a la siguiente, existen limitadas oportunidades de dar marcha atrás y cambiar decisiones ya tomadas. Una desventaja de este enfoque es que los usuarios potenciales no tienen oportunidad de interactuar con el sistema recién hasta bien entrado en el ciclo de vida del mismo.

En el modelo de prototipo, se pone un gran énfasis en producir un modelo o prototipo operativo bien temprano durante el ciclo de vida del sistema. El prototipo es construido con el propósito de arribar a una especificación final de la funcionalidad del sistema propuesto. Los usuarios potenciales interactúan con el prototipo en forma temprana y con frecuencia hasta que son acordados los requerimientos. Este enfoque le da a los potenciales usuarios la oportunidad de interactuar con un sistema prototipo en forma temprana durante el ciclo de desarrollo, y mucho antes que el sistema final esté diseñado y codificado.

En un memorándum de fecha 11 de diciembre del anteaño pasado, Jan Anderson, un miembro del equipo original del proyecto CX30, atacó duramente la decisión tomada por el gerente de proyecto Sam Reynolds de emplear el modelo de cascada. El SENTINEL-OBSERVER obtuvo una copia del memo de Anderson, dirigido a Reynolds, y Anderson verificó la autenticidad del memorándum para este diario.

Reynolds despidió a Anderson el 24 de diciembre, justo dos semanas después que ella escribiera el memo.

El memo de Anderson hace referencia a una reunión anterior en la que ocurrió un fuerte intercambio de opiniones relacionadas con la filosofía del desarrollo del software.

En el memo, Anderson subrayó el siguiente párrafo.:

“No fueron mis intenciones impugnar su competencia durante nuestra reunión de ayer, pero debo protestar con mi mayor vehemencia contra la idea de que completemos el software de Robbie CX30 siguiendo el modelo de cascada que usted ya usó en otros proyectos. No necesito recordarle que aquellos eran proyectos de procesamiento de datos que involucraban el procesamiento de transacciones de negocios. El proyecto Robbie CX30 llevará un alto grado de interacción, tanto entre robot y componentes como entre robot y su operador. Dado que la interacción del operador con el robot es tan importante, la interfaz no puede estar diseñada como una idea de último momento.”

Randy Samuels, a quien se lo acusó de asesinato no premeditado por la muerte del operador Bart Matthews, padre de tres niños, había participado de la reunión del 11 de diciembre.

En una conversación con este diario, Anderson dijo que Samuels no tenía mucho para decir sobre la controversia cascada-prototipo, pero sí afirmó que daría “una mano” con tal que exoneraran a Samuels.

“El proyecto fue sentenciado a muerte mucho antes que Samuels malinterpretara esas fórmulas”, aclaró Anderson enfáticamente en la sala de su casa en los suburbios.

En conversación con este diario, Anderson hizo lo mejor de sí para explicar la controversia del método cascada vs. prototipo en términos simples. “El punto principal en realidad era si podíamos llegar a ponernos de acuerdo en los requerimientos del sistema sin dejar que los operadores del robot presintieran lo que teníamos en mente. Reynolds ha estado en el negocio de procesamiento de datos por tres décadas y es bueno en eso, pero nunca debería haber sido el gerente de este proyecto.”

Conforme a registros obtenidos por el SENTINEL-OBSERVER, Silicon Techtronics transfirió a Sam Reynolds de la División Procesamiento de Datos, que se encargaba de inventario y salarios, a la División Robótica, justo tres semanas antes de la reunión del 11 de diciembre a que alude Anderson en su memo.

Reynolds fue transferido a la División Robótica por el presidente de Silicon Techtronics, Michael Waterson. Reynolds reemplazaba a John Cramer, quien gerenciaba el anterior proyecto Robbie CX10 y CX20. Cramer fue puesto a cargo del proyecto CX30, pero murió inesperadamente en un accidente aéreo. Al colocar a Reynolds a cargo del proyecto CX30, nos dice nuestra fuente, que Waterson iba en contra del consejo de Ray Johnson, Jefe de la División Robótica. De acuerdo con estas fuentes, Johnson se oponía fuertemente a la alternativa de ponerlo a Reynolds como jefe del proyecto Robbie CX30. Estas fuentes dijeron al SENTINEL-OBSERVER que la elección de Waterson por Reynolds fue puramente una decisión de recorte de gastos. Era más barato transferir a Reynolds a la División Robótica que incorporar a un nuevo líder de proyecto fuera de la corporación.

La fuente anónima que el SENTINEL-OBSERVER llamará “Marta” describió la situación de este modo: “Waterson pensaba que sería más barato transferir a Reynolds a robótica antes que intentar encontrar afuera un nuevo gerente para el proyecto Robbie. Además, Waterson tendía a sospechar de la gente de afuera del grupo. Con frecuencia mandaba memos sobre cuánto tarda la gente en aprender “el modo de hacer las cosas de Silicon Techtronics”. Desde el punto de vista de Waterson, Reynolds era el gerente y fue transferido a su nuevo puesto en Robótica como un gerente y no como un experto técnico. Claramente, Reynolds se veía a sí mismo tanto gerente como experto técnico. Reynolds no tenía conciencia de sus propias limitaciones técnicas.”

Según Marta, Reynolds era muy renuente a gerenciar un proyecto que no usara el modelo de cascada que tan bien le había servido en el procesamiento de datos. Tildó al modelo prototipo como un “modelo de moda” en la reunión del 11 de diciembre, y después de una serie de intercambios verbales la cosa se puso muy personal.

“Anderson estaba especialmente expresiva”, recuerda Marta. “Tenía mucha experiencia con interfaces con usuarios y desde su perspectiva, la interfaz robot-operador era crítica para el éxito del CX30, dado que la intervención del operador sería frecuente y a veces crítica.” En su entrevista con el SENTINEL-OBSERVER, Jan Anderson comentó sobre este aspecto de la reunión del 11 de diciembre:

“Reynolds estaba en contra de “perder el tiempo” - para usar sus propias palabras - con cualquier tipo de análisis formal de las propiedades de los factores humanos y su interfaz con el usuario. Para él, las interfaces con el usuario real eran un tema periférico.”

“Para él [Reynolds], cualquier cosa nueva era “moda”, agrega Anderson. “Las interfaces de la computadora eran una moda, el diseño orientado a objetos era una moda, la especificación formal y las técnicas de verificación eran una moda, y por sobre todo, el modelo prototipo era una moda.”

Justo una semana después de la reunión del 11 de diciembre, el grupo del proyecto Robbie recibió un memo de Sam Reynolds concerniente al plan para el proyecto Robbie CX30.

“Era el modelo de cascada, como salido de un libro”, Anderson dijo a este reporter mientras revisaba una copia del memo con el plan del proyecto. “Análisis de requerimientos y especificación, luego diseño de arquitectura y diseño detallado, codificación, prueba, entrega y mantenimiento. En el modo de ver de Reynolds, no hacía falta tener ninguna interacción del usuario con el sistema hasta muy, pero muy avanzado el proyecto.” El SENTINEL-OBSERVER se ha enterado de que el primer operador que realmente usó el robot Robbie CX30 en una función industrial fue Bart Matthews, el hombre que fue muerto en la tragedia del robot asesino. Este primer uso de Robbie CX30 en un uso industrial fue cubierto por los medios, incluyendo este periódico. Como una gran ironía, El Informe Anual de Silicon Techtronics para los Accionistas, publicado el pasado mes de marzo, contiene en la brillante portada una foto de un sonriente Bart Matthews. A Matthews se lo muestra operando al mismísimo robot Robbie CX30 que lo aplastó hasta la muerte tan solo dos meses después de la toma fotográfica.

Silicon Techtronics Prometió Entregar un Robot Seguro

Cuestionada la Calidad de Entrenamiento del Operador

Especial para el SENTINEL-OBSERVER de SILICON VALLEY

***Silicon Valley, EEUU
por Mabel Muckraker***

En una conferencia de prensa de esta tarde, un grupo de programadores que se autodenominan “Comité de Justicia para Randy Samuels”, distribuyó documentos que muestran que Silicon Techtronics se obligó a sí misma a hacer entrega de robots que “no causarían ningún daño corporal a los operadores humanos”. Randy Samuels es el programador que ha sido acusado de asesinato en el infame caso del “robot asesino”.

“No podemos entender como el Fiscal pudo acusar a Randy con esos cargos cuando, de hecho, la compañía Silicon Techtronics estaba legalmente obligada a producir y entregar robots seguros a Cybernetics”, dijo el vocero del comité, Ruth Witherspoon. “Creemos que en todo esto hay un encubrimiento y que hay algún tipo de confabulación entre la gerencia de SiliTech [Silicon Techtronics] y la oficina del Fiscal. Michael Waterson era uno de los más grandes contribuyentes de la campaña de reelección de la Sra. McMurdock del año pasado”. Michael Waterson es Presidente Ejecutivo de Silicon Techtronics. Jane McMurdock es la Fiscal de la ciudad de Silicon Valley. El SENTINEL-OBSERVER confirmó que Waterson hizo varios grandes aportes a la campaña de reelección de McMurdock el otoño pasado.

“A Randy le están haciendo pagar los platos rotos por una empresa que tiene estándares de control de calidad malos y no lo vamos a permitir! Gritó Witherspoon en una emotiva declaración a los periodistas. “Creemos que la política ha entrado en todo esto”.

Los documentos que fueron distribuidos por el comité por la “Justicia para Randy Samuels” eran porciones de lo que se llama un “documento de requerimientos”. Según Ruth Witherspoon y otros miembros del comité, este documento prueba que Samuels no fue legalmente responsable de la muerte de Bart Matthews, el desafortunado operador de robot que fue muerto por un robot de Silicon Techtronics en Cybernetics, Inc. en Silicon Heights el pasado mes de abril.

El documento de requerimientos es un contrato entre Silicon Techtronics y Cybernetics, Inc. Especifica con total detalle la funcionalidad del robot Robbie CX30 que Silicon Techtronics prometió entregar a Cybernetics. Según Witherspoon, el robot Robbie CX30 fue diseñado para ser un robot “inteligente” que pudiera ser capaz de operarse en una variedad de funciones industriales. Cada cliente de la corporación necesitó de documentos de requerimientos separados ya que Robbie CX30 no era un “robot de llave en mano” sino un robot que necesitaba ser programado de forma diferente para cada aplicación.

No obstante, todos los documentos de requerimientos que fueron acordados bajo los auspicios del proyecto Robbie CX30, incluyendo al acuerdo entre Silicon Techtronics y Cybernetics, contienen los siguientes fundamentos de importancia:

“El robot será de operación segura y aun bajo circunstancias excepcionales (ver Sección 5.2), el robot no causará daño corporal alguno al operador humano.”

“En el caso de condiciones excepcionales que potencialmente contengan el riesgo de daño corporal (ver Sección 5.2.4 y todas sus subsecciones), el operador humano podrá ingresar una secuencia de códigos de comando, según se describe en las secciones relevantes de la especificación funcional (ver Sección 3.5.2), que detendrá el movimiento del robot mucho antes que pueda ocurrir un real daño corporal.” Las “condiciones excepcionales” incluyen eventos inusuales tales como datos extraños desde los sensores del robot, movimiento errático o violento del robot o error del operador. Fue justamente esa condición excepcional la que llevó a la muerte a Bart Matthews.

Estos párrafos fueron extractados de las porciones del documento de requerimientos que trata sobre los “requerimientos no funcionales”. Los requerimientos no funcionales listan en detalle las restricciones bajo las cuales operaría el robot. Por ejemplo, el requerimiento de que el robot sería incapaz de dañar a su operador humano es una restricción y Silicon Techtronics, según Ruth Whitterspoon, estaba legalmente obligada a satisfacer este punto. La parte de los requerimientos funcionales del documento de requerimientos cubre (nuevamente en sumo detalle) el comportamiento del robot y su interacción con el entorno y su operador humano. En particular, los requerimientos funcionales especificaban el comportamiento del robot bajo cada una de las condiciones excepcionales esperadas. En su declaración a los periodistas en la conferencia de prensa, Whitterspoon explicó que Bart Matthews fue muerto cuando se produjo la condición excepcional 5.2.4.26. Ésta involucra un movimiento del brazo del robot extremadamente violento e impredecible. Esta condición requiere de la intervención del operador, a saber el ingreso de los códigos de comando mencionados en el documento, pero aparentemente Bart Matthews se confundió y no pudo ingresar con éxito estos códigos.

“Si bien el programa de Randy Samuels estaba mal - él en verdad malinterpretó las fórmulas de la dinámica del robot, como se informó a los medios. La condición excepcional 5.2.4.26 estaba diseñada para protegerse de justamente este tipo de contingencia”, dijo Whitterspoon a los periodistas. “Los valores del movimiento del robot generados por el programa de Randy identificaron correctamente a esta condición excepcional y el operador del robot recibió el debido aviso de que algo andaba mal”.

Whitterspoon dijo que poseía una declaración firmada de otro operador de robot de Cybernetics en donde afirmaba que las sesiones de entrenamiento ofrecidas por Silicon Techtronics nunca mencionaron a ésta ni a tantas otras condiciones excepcionales. Según Whitterspoon, el operador del robot ha jurado que ni a él ni a ningún otro operador de robot les fue dicho jamás que el brazo del robot podía oscilar violentamente.

Whitterspoon citó esta declaración en la conferencia de prensa. “Ni yo ni Bart recibimos entrenamiento para manejar este tipo de condición excepcional. Dudo mucho que Bart Matthews tuviese idea de qué se suponía debía hacer cuando la pantalla de la computadora comenzó a mostrar el mensaje de error”.

Las condiciones excepcionales que requieren de la intervención del operador causan un mensaje de error que se genera en la consola del operador. La Policía de Silicon Valley confirmó que cuando Bart Matthews fue muerto, el manual de referencia en su consola estaba abierto en la página del índice que contenía los códigos de ingreso para los “errores”.

Whitterspoon luego citó secciones del documento de requerimientos que obligan a Silicon Techtronics (el vendedor) a entrenar adecuadamente a los operadores de robots:

“El vendedor suministrará cuarenta (40) horas de entrenamiento para los operadores. Este entrenamiento cubrirá todos los aspectos de la operación del robot, incluyendo una cobertura exhaustiva de los procedimientos de seguridad que deben seguirse en caso de condiciones excepcionales que contengan potencialmente el riesgo de daño corporal. El vendedor proveerá y administrará instrumentos de prueba apropiados que serán usados para certificar el suficiente entendimiento del operador de las operaciones de la consola del robot y de los procedimientos de seguridad. Sólo los empleados del cliente que hayan pasado estas pruebas estarán habilitadas para operar el robot Robbie CX30 en una verdadera función industrial.

“El manual de referencia deberá suministrar instrucciones claras para la intervención del operador en todas las situaciones excepcionales, especialmente e inclusive aquellas que contengan potencialmente el riesgo de daño corporal.”

Según Whitterspoon, las declaraciones juradas de varios operadores de robots de Cybernetics Inc., aseguran que sólo se destinó un día laborable (aproximadamente 8 horas) al entrenamiento de los operadores. Es más, casi no se dedicó tiempo alguno al tratamiento de condiciones excepcionalmente peligrosas.

“La prueba escrita desarrollada por Silicon Techtronics para habilitar a un operador de robot era considerada por los empleados de Cybernetics como un chiste”, aseguró Whitterspoon.

“Obviamente Silicon Techtronics no le dedicó mucho al entrenamiento y procedimientos de prueba obligatorios según el documento de requerimientos según la evidencia en nuestro poder”.

*Reimpreso con el permiso de ROBOTIC WORLD
el diario de ROBOTICS AND ROBOTICS APPLICATIONS*

La Interfaz del “Robot Asesino”

*Dr. Horace Gritty, Departamento de Ciencias de la Computación y materias relacionadas
Universidad de Silicon Vallley
Silicon Valley, EEUU*

Resumen: El robot industrial Robbie CX30 se suponía que debía establecer un nuevo modelo de inteligencia de robots industriales. Desgraciadamente, uno de los primeros robots Robbie CX30 mató a un obrero de la línea de montaje, y ésto llevó a la acusación de uno de los que desarrollaron el software del robot, Randy Samuels. Este paper promueve la teoría de que quien debería estar en juicio en este caso sería el diseñador de la interfaz robot-operador. El robot Robbie CX30 viola casi todas las reglas del diseño de interfaz. Este paper se centra en cómo la interfaz del Robbie CX30 violó cada una de las “Ocho Reglas de Oro” de Shneiderman.

1. Introducción

El 17 de mayo de 1992, un robot industrial Robbie CX30 de Silicon Techtronics mató a su operador, Bart Matthews, en Cybernetics Inc., en Silicon Heights, un suburbio de Silicon Valley. La investigación de los hechos del accidente guiaron a las autoridades a la conclusión de que un módulo de software, escrito y desarrollado por Randy Samuels, un programador de Silicon Techtronics, fue el responsable del comportamiento errático y violento que a su vez llevó a la muerte por decapitación de Bart Matthews [NOTA AL PIE. Los medios de prensa manejaron la información haciendo creer que Bart Matthews había sido aplastado por el robot, pero la evidencia fotográfica dada a este autor muestra otra cosa. Tal vez, las autoridades trataban de proteger la sensibilidad pública].

Como experto en el área de interfaces con el usuario (1, 2, 3), se me pidió prestar ayuda a la policía en la reconstrucción del accidente. Para poder hacer esto, se le pidió a Silicon Techtronics que me suministrara un simulador de Robbie CX30 que incluyera por completo la consola del operador del robot. Esto me permitió investigar el comportamiento del robot sin tener que en realidad arriesgarme seriamente. Debido a mi profundo entendimiento de interfaces con el usuario y factores humanos pude reconstruir el accidente con extrema precisión. Sobre la base de esta reconstrucción, llegué a la conclusión de que fue el diseño de la interfaz y no el por cierto imperfecto diseño del software lo que debería haber sido visto como el criminal en este caso.

A pesar de mi descubrimiento, la Fiscal Jane McMurdock insistió en proseguir el caso contra Randy Samuels. Pienso que cualquier Computador Científico competente, dado una oportunidad de interactuar con el simulador del Robbie CX30, también habría concluido que el diseñador de la interfaz y no el programador debería haber sido acusado por negligencia, si no por homicidio no premeditado.

2. “Las ocho reglas de oro” de Shneiderman

Mi evaluación de la interfaz con el usuario del Robbie CX30 está basada en las “ocho reglas de oro” de Shneiderman (4). También utilicé otras técnicas para evaluar la interfaz, pero éstas serán publicadas en papers separados. En esta sección ofrezco una breve revisión de las ocho reglas de oro de Shneiderman, un tema que resultará más familiar para expertos en interfaces de computación como yo y no a hackers de robótica que leyeron este oscuro periódico.

Las ocho reglas de oro son:

1. Buscar siempre la coherencia. Tal como se puede ver más abajo, es importante que una interfaz con el usuario sea coherente a muchos niveles. Por ejemplo, los diseños de pantalla deben ser coherentes de una pantalla a la siguiente. En un entorno en que se usa una interfaz gráfica con el usuario (GUI), esto también implicará concordancia de un utilitario al siguiente.
2. Permitirle a los usuarios frecuentes el uso de shortcuts. Los usuarios frecuentes (o, ‘power users’) pueden desalentarse ante tediosos procedimientos. Permitirle a estos usuarios un procedimiento menos tedioso para completar una tarea dada.
3. Dar realimentación de información. Los usuarios necesitan ver las consecuencias de sus acciones. Si un usuario ingresa un comando pero la computadora no muestra ya sea que lo está procesando o lo ha procesado, esto puede dejar al usuario confundido o desorientado.
4. Diseñar diálogos que tengan un fin. Interactuar con una computadora es algo así como dialogar o conversar. Cada tarea debe tener un inicio, un desarrollo y un fin. Es importante que el usuario sepa cuándo una tarea está finalizada. El usuario necesita tener la sensación de que la tarea ha alcanzado su término.
5. Permitir manejos simples de los errores. Los errores del usuario deben estar diseñados dentro del sistema. Otro modo de decirlo es que no debe haber ninguna acción por parte del usuario que sea considerada como un error que está más allá de la capacidad del sistema para manejarlo. Si el usuario comete un error, debe recibir información útil, clara y concisa sobre la naturaleza de tal error. Debe resultar fácil para el usuario deshacer este error.
6. Permitir deshacer las acciones con facilidad. Más genéricamente, los usuarios deben poder deshacer lo que han hecho, sea ésto o no de naturaleza errónea.
7. Respalidar que el centro del control esté internamente. La satisfacción del usuario es alta cuando el usuario tiene la sensación de que es él o ella quien tiene el control y la satisfacción del usuario es baja cuando el usuario siente que la computadora tiene el control. Diseñar interfaces para reforzar la sensación de que es en el usuario donde yace el control en el ámbito de la interacción humano-computadora.
8. Reducir la carga de la memoria inmediata. La memoria inmediata del ser humano es notablemente limitada. Los psicólogos regularmente citan a la ley de Miller que dice que la memoria inmediata está limitada a siete piezas discretas de información. Hacer todo lo posible para liberar la carga en la memoria del usuario. Por ejemplo, en lugar de

pedirle al usuario que tipee el nombre de un archivo para abrirlo, presentar al usuario una lista de archivos disponibles en ese momento.

3. *Revisión de la consola del robot*

La interfaz del operador de Robbie CX30 violó todas y cada una de las reglas de Shneiderman. Muchas de estas violaciones fueron directamente responsables del accidente que terminó con la muerte de un operador de robot.

La consola del robot era una IBM PS/2 modelo 55SX con un procesador 80386 y un monitor EGA color con resolución 640 x 480. La consola tenía un teclado, pero no mouse. La consola estaba empotrada en una estación de trabajo que tenía, además, estantes para manuales y un área para tomar notas y para leer manuales. No obstante, el área para escribir/leer estaba a bastante distancia de la pantalla de la computadora, o sea que era bastante incómodo y cansador para el operador manejar cualquier tarea que requiriera de mirar algo en el manual y luego actuar rápidamente con respecto al teclado de la consola. La silla del operador estaba deficientemente diseñada y estaba demasiado alta con respecto a la posición de la consola y al área de escribir/leer. Esto resentía mucho la espalda del operador y también causaba excesivo cansancio de la vista.

No puedo comprender cómo un sistema sofisticado como este no pudo incluir un aparato de mejor diseño para los ingresos de los datos. Uno sólo podría concluir que Silicon Techtronics no tenía mucha experiencia en tecnología de interfaces con el usuario. El documento de requerimientos (5) especificaba un sistema manejado por menús, lo cual era una elección razonable. Sin embargo, en un utilitario en donde lo esencial era la rapidez, especialmente cuando la seguridad del operador estaba en juego, el uso de un teclado para todas las tareas de selección de opción de menús fue una elección de extremada pobreza, que requería de mucho uso del teclado para lograr el mismo efecto que podía haberse conseguido casi instantáneamente mediante un mouse. (Ver el paper escrito por Foley et al. (6) En realidad, se me ocurrieron todas estas ideas antes que Foley las publicara, pero él me ganó).

El operador del robot podía interactuar con el robot y de este modo producir un impacto sobre su comportamiento al hacer las opciones en un sistema de menús. El menú principal consistía de 20 ítemes, demasiados en mi opinión, y cada ítem del menú principal tenía un submenú tipo desplegable asociado a éste. Algunos de los submenús tenían tanto como veinte ítemes - nuevamente, demasiados. Es más, parecía haber muy poca rima o razón en cuanto a por qué los ítemes de los menús estaban listados en el orden en que lo estaban. Hubiese sido mucho mejor una organización alfabética o funcional.

Algunos de los ítemes en los menús desplegables tenían hasta cuatro menús pop up relacionados a éstos. Éstos aparecerían en secuencias a medida que se hacía la selección correspondiente en los submenús. Ocasionalmente, una elección de un submenú abriría un cuadro de diálogo en la pantalla. Un cuadro de diálogo requiere de cierto tipo de interacción entre el operador y el sistema, por ejemplo para resolver ciertos temas, como ser, ingresar cuál es el diámetro de un dispositivo dado a ser bajado en el baño de ácido.

El sistema de menús presenta una estricta jerarquía de elección de menús. El operador podría volver hacia atrás en esta jerarquía apretando la tecla de escape. Esta tecla escape también podría terminar los diálogos.

El uso de color en la interfaz fue muy poco profesional. Había demasiados colores en un espacio demasiado chico. Los contrastes eran muy fuertes y el resultado, para este revisor, resultó en una

severa fatiga ocular en tan sólo quince minutos de uso. Hubo uso excesivo de efectos musicales tontos y flashes cuando se ingresaban opciones o códigos erróneos.

Uno debería preguntarse por qué Silicon Techtronics no intentó un enfoque más sofisticado para el diseño de la interfaz. Luego de un cuidadoso estudio del dominio de los utilitarios del Robbie CX30, he llegado a la conclusión de que una interfaz de manipulación directa, que mostrara literalmente al robot en la consola del operador, habría sido lo ideal. El entorno tan visual dentro del cual operaba el robot se prestaba naturalmente al diseño de metáforas de pantalla apropiadas para ese entorno, metáforas que el operador podría entender con facilidad. Esto permitiría que el operador manipulara el robot mediante el manejo, en la consola del robot, de la representación gráfica del robot en su entorno. He solicitado a uno de mis estudiantes en el doctorado, Susan Farnsworth, que investigara un poco más esta posibilidad.

4. *En qué modo la interfaz del robot Robbie CX30 violó las ocho reglas de oro*

La interfaz con el usuario de Robbie CX30 violó todas y cada una de las reglas de oro en diferentes modos. En este paper sólo trataré unas pocas instancias de violaciones de reglas, dejando la discusión más en detalle para futuros artículos y mi próximo libro [NOTA AL PIE:

CODEPENDENCIA: Cómo los Usuarios de Computadoras permiten deficientes Interfaces con el Usuario, Angst Press, Nueva York. Este libro presenta una teoría radicalmente nueva con respecto a la relación entre la persona y su máquina. Esencialmente, algunas personas necesitan una interfaz de mala calidad a los fines de evitar ciertos problemas psicológicos no resueltos en sus vidas.]. Lo que haré es destacar esas violaciones que fueron relevantes en este accidente en particular.

4.1 Buscar siempre la coherencia

En la interfaz del usuario de Robbie CX30 hubieron muchas incoherencias. Los mensajes de error podían aparecer en casi cualquier color y podían estar acompañados por casi cualquier tipo de efecto musical. Además, los mensajes de error podían aparecer en casi cualquier lugar de la pantalla.

Cuando Bart Matthews vio el mensaje de error de la condición excepcional que ocurrió luego, la cual requería la intervención del operador, es probable que fuera esa la primera vez que veía ese mensaje en especial. Además, el mensaje de error apareció en un cuadro verde, sin ningún efecto de sonido. Este es el único mensaje de error de todo el sistema que aparece en verde y sin ningún tipo de acompañamiento de orquesta.

4.2 Permitir que los usuarios frecuentes utilicen shortcuts

Este principio no aparece de ningún modo en todo el diseño de la interfaz. Por ejemplo, hubiera sido una buena idea permitir que los usuarios frecuentes pudieran ingresar la primera letra de la opción de un submenú o menú en lugar de requerírseles el uso de las teclas del cursor y luego la tecla “enter” para elegir esa opción determinada. El mecanismo de selección de menús de este sistema debe haber provocado al operador bastante fatiga mental.

Es más, debería haberse permitido algún tipo de sistema de tipeo anticipado que permitiera al usuario frecuente ingresar una secuencia de opciones de menú sin tener que esperar a que apareciera realmente el menú en pantalla.

4.3 Ofrecer realimentación¹ de información

En muchos casos el usuario no tiene idea de si el comando que acaba de ingresar se está procesando o no. Este problema se exagera además por las inconsistencias en el diseño de la interfaz con el usuario. En algunos casos al operador se le da una realimentación detallada de lo que el robot está ejecutando. En otros, el sistema permanece misteriosamente silencioso. En general, al usuario se lo lleva a que espere algún tipo de realimentación y por consiguiente se queda confundido cuando ésta no se le da. En la pantalla, no hay una representación visual del robot y su entorno, y la visión que tiene el operador del robot a veces está obstruida.

4.4 Diseñar diálogos que tengan fin.

Hay muchos casos en los que una secuencia dada de teclado representa una idea holística, una tarea completa, pero al operador se lo deja sin el tipo de realimentación que le confirme que la tarea ha sido en efecto completada. Por ejemplo, hay un diálogo bastante complicado que se necesita cuando se quiere sacar un elemento de un baño de ácido. Sin embargo, luego de completar este diálogo, el usuario es llevado a otro diálogo nuevo, y no relacionado con éste, sin que se le informe que el diálogo anterior ha finalizado.

4.5 Ofrecer manejo simple de los errores

El sistema pareciera estar diseñado para que el usuario se lamentara por cualquier ingreso erróneo. No sólo el sistema permite numerosas oportunidades para el error, sino que cuando un error en realidad ocurre, es probable que no se repita por algún tiempo. Ello se debe a que la interfaz con el usuario hace que recuperarse de un error sea una odisea tediosa, frustrante y a veces enfurecedora. Algunos de los mensajes de error eran directamente ofensivos y condescendientes.

4.6 Permitir deshacer las acciones con facilidad

Como se menciona en el párrafo anterior, la interfaz con el usuario hace muy difícil la tarea de deshacer entradas erróneas. En general, el sistema de menús permite deshacer fácilmente las acciones, pero esta filosofía no alcanza al diseño de los cuadros de diálogo y al manejo de condiciones excepcionales. Desde un punto de vista práctico (opuesto al teórico), la mayoría de las acciones son irreversibles cuando el sistema está en un estado de condición excepcional, y esto ayudó a llegar a la tragedia del robot asesino.

¹"Feedback"

4.7 Promover que uno sea el centro del control.

Muchas de las deficiencias tratadas en los párrafos precedentes disminuyeron la sensación de “tener el control”. Por ejemplo, no recibir información, no poder concluir las interacciones, no permitir deshacer con facilidad las acciones en el momento en que surgen las excepciones, todas estas cosas actúan para disminuir la sensación de que el usuario posee el control sobre el robot. Hubieron muchas características de esta interfaz que hicieron que el operador sintiera que hay un enorme bache entre la consola del operador y el robot en sí, mientras que un buen diseño de interfaz hubiera hecho transparente la interfaz con el usuario y le hubiera dado al operador del robot la sensación de estar en contacto directo con el mismo. En un caso, le ordené al robot mover un elemento desde baño de ácido hasta la cámara de secado y pasaron 20 segundos antes de que el robot pareció responder. De este modo, no tuve la sensación de estar controlando al robot. Tanto la respuesta demorada del robot como la falta de realimentación en la pantalla de la computadora, me hicieron sentir que el robot era un agente autónomo, la verdad, un sentimiento como mínimo perturbador.

4.8 Reducir la carga de la memoria de corto plazo

Un sistema que se maneja por medio de menús es en general bueno en términos de la carga de memoria que crea en los usuarios. No obstante, hay una gran variación entre implementaciones particulares de sistemas de menú en lo que hace a carga de memoria. La interfaz con el usuario de Robbie CX30 tenía menús muy grandes sin una obvia organización interna. Esto crea una gran carga al operador en términos de memoria y también en términos de tiempos de búsqueda, el tiempo que le lleva al operador ubicar una opción determinada de un menú.

Muchas pantallas de diálogo requerían que el usuario ingresara con el teclado números de partes, nombres de archivos, y otra información. El sistema podría haberse diseñado fácilmente de forma de mostrarle al usuario estos números de parte, etc., sin necesitar que el usuario recordara estas cosas de su propia memoria. Esto incrementaba la carga sobre la memoria del usuario.

Para finalizar, y esto es realmente imperdonable, increíble como puede parecer, ¡no había ninguna instalación de ayuda en línea o sensible al contexto! Si bien he ido a los cursos de entrenamiento ofrecidos por Silicon Techtronics, muchas veces me encontré navegando por los manuales de referencia para poder encontrar la respuesta a aún las más básicas preguntas, tales como: “¿Qué significa esta opción de menú? ¿Qué pasa si selecciono esta opción?”

5. Una reconstrucción de “la tragedia del robot asesino”

Las fotos policiales de la escena del accidente no son nada agradables de ver. La consola del operador estaba salpicada con bastante cantidad de sangre. No obstante, la calidad de las fotos es excepcional y utilizando técnicas de ampliación pude descubrir los siguientes factores de importancia sobre el momento en que se produjo el accidente:

1. La luz NUM LOCK estaba encendida

El teclado IBM contiene un tablero numérico que puede operar de dos modos. Cuando la luz NUM LOCK está encendida, esa parte se comporta como una calculadora. Del otro modo, las teclas pueden usarse para mover el cursor en la pantalla.

2. Había sangre esparcida en el tablero numérico

Las huellas ensangrentadas indican que Bart Matthews estaba usando el tablero numérico en el momento en que fue golpeado y muerto.

3. Se encontraba titilando en verde un mensaje de error

Esto nos dice la situación de error vigente en el momento en que ocurrió la tragedia. El mensaje de error decía “ROBOT DYNAMICS INTEGRITY ERROR-45”²

4. Había un manual de referencia que estaba apoyado abierto sobre el área de lectura/escritura de la estación de trabajo

Uno de los cuatro volúmenes del manual de referencia estaba abierto en la página del índice que contenía el ítem “ERRORES/MENSAJES”.

5. En la pantalla también había un mensaje que mostraba instrucciones al operador

El mensaje aparecía en amarillo en la parte inferior de la pantalla. En el mensaje se leía . “POR FAVOR INGRESE INMEDIATAMENTE LA SECUENCIA DE COMANDOS PARA CANCELAR EL ERROR DINÁMICO DEL ROBOT!!!”

En base a las evidencias físicas más otras evidencias contenidas en los registros del sistema, y basándose en la naturaleza del error ocurrido (error de integridad de dinámica del robot - 45, el error que estuvo causado por el programa de Randy Samuels), he llegado a la conclusión de que ocurrió la siguiente secuencia de eventos en la fatal mañana de la tragedia del robot asesino:

10:22:30 “ERROR DE INTEGRIDAD DE DINÁMICA DEL ROBOT - 45” aparece en la pantalla. Bart Matthews no lo ve porque no hay efecto de audio o señal sonora tal como ocurre con todas las otras situaciones de error. Además, el mensaje de error aparece en verde, lo que en todos los otros contextos significa que hay un proceso completándose con normalidad.

10:24:00 El robot comienza a moverse lo suficientemente violento como para que Bart Matthews lo note.

10:24:05 Bart Matthews se da cuenta del mensaje de error, no sabe lo que significa. No sabe qué hacer. Intenta con el submenú “cancelación de emergencia”, un submenú de uso genérico para apagar el robot. Este involucra SEIS opciones de menú por separado, pero el Sr. Matthews no se da cuenta de que la luz NUM LOCK está encendida. Por ende, las opciones del menú no están ingresando dado que las teclas del cursor operan como teclas de calculadora.

10:24:45 El robot gira del baño de ácido y comienza a arrastrar la consola del operador, con sus brazos dentados batiéndose con gran amplitud. Nadie esperaba que un operador tuviera que huir de un robot descontrolado, así que Bart Matthews queda atrapado en su área de trabajo por el robot que avanza. Más o menos para este momento es que Bart Matthews saca el manual de referencia y empieza a buscar el error ERROR DE INTEGRIDAD DE DINÁMICA DEL ROBOT - 45 en el índice. Ubica con éxito una referencia a mensajes de error en el índice.

² Error de integridad de dinámica de robot-45

10:25:00 El robot ingresa al área del operador. Bart Matthews abandona la búsqueda del procedimiento del operador ante un error de integridad de dinámica del robot. En su lugar, intenta una vez más ingresar la secuencia de “cancelación de emergencia” desde el teclado numérico, momento en que es golpeado.

6. Resumen y conclusiones

Si bien el módulo de software escrito por Randy Samuels causó en verdad que el robot Robbie CX30 oscilara fuera de control y atacara a su operador humano, un buen diseño de la interfaz hubiera permitido al operador terminar con el comportamiento errático del robot. En base al análisis de la interfaz del usuario del robot llevado a cabo utilizando las ocho reglas de oro de Shneiderman, el experto en diseño de interfaces ha arribado a la conclusión de que el diseñador de la interfaz y no el programador fue la parte más culpable en este desafortunado evento.

7. Referencias

1. Gitty, Horace (1990). The Only User Interface Book You'll Ever Need. Vanity Press, Oshkosh, WI, 212 pag. [El único libro sobre Interfaz del usuario que usted necesitará]
2. Gitty, Horace (1992). What We Can learn from the Killer Robot [Lo que podemos aprender del robot asesino], charla dada en el Simposio Internacional de la Universidad de Silicon Valley sobre Seguridad de Robots e Interfaces con el Usuario, Marzo de 1991. También por aparecer en las Notas de Alumnos de la Universidad de Silicon Valley.
3. Gitty, Horace (se espera para 1993). CODEPENDENCY: How Computer Users Enable Poor User Interfaces, Angst Press, New York [Cómo los usuarios de computadoras permiten interfaces deficientes]
4. Shneiderman, Ben (1987), Designing the User Interface, Addison-Wesley, Reading MA, 448 pag. [Diseño de Interfaces]
5. DOCUMENTO DE REQUERIMIENTOS DEL ROBOT INDUSTRIAL INTELIGENTE Robbie CX 30: versión de Cybernetics Inc., Documento Técnico N° 91-0023XA, Silicon Techtronics Corporation, Silicon Valley, USA 1245 pag.
6. Foley, J. P., Wallace, V. L., y Chan, P. (1984): The Human Factors of Computer Graphics Interaction Techniques [Los factores humanos de las técnicas de interacción de gráficas de computación] IEEE COMPUTER GRAPHICS AND APPLICATIONS, 4(11) pag. 13-48.

Ingeniero de Software Cuestiona la Autenticidad de las Pruebas de Software del “Robot Asesino”

La Indagación de un Profesor de la Universidad de Silicon Valley Provoca Serios Cuestionamientos Legales y Éticos

Especial para el SENTINEL-OBSERVER DE SILICON VALLEY

***Silicon Valley, EEUU
por Mabel Muckraker***

El caso del “robot asesino” dio un giro significativo ayer cuando un profesor de la Universidad de Silicon Valley presentó un informe en donde cuestiona la autenticidad de las pruebas que fueron hechas por Silicon Techtronics al software del “robot asesino”. El Profesor Wesley Silber, Profesor de Ingeniería de Software, dijo en una conferencia de prensa realizada en la universidad que los resultados de las pruebas reflejados en los documentos internos de Silicon Techtronics no concordaban con los resultados de las pruebas obtenidos cuando él y sus colegas ensayaron el software real del robot.

Silicon Valley aún está reaccionando por el anuncio del Profesor Silber, que podría jugar un papel importante en el juicio a Randy Samuels, el programador de Silicon Techtronics que fue acusado por homicidio no premeditado en el ahora infame incidente del “robot asesino”. Presionada por su reacción por el informe del Profesor Silber, la Fiscal Jane McMurdock reiteró su confianza en que el jurado encontrará culpable a Randy Samuels. Sin embargo, la Fiscal Jane McMurdock impresionó a los periodistas cuando agregó “pero, esto en verdad promueve la posibilidad de nuevas acusaciones”.

Ruth Whitterspoon, la vocero del “Comité de justicia para Randy Samuels”, también estuvo exultante cuando habló a este periódico. “McMurdock no puede tener ambas cosas”. O el programador es el responsable por esta tragedia o se deberá hacer responsable a la gerencia por ello. Creemos que el informe de Silber exonera a nuestro amigo y colega Randy Samuels.”

El gerente Ejecutivo de Silicon Techtronics Michael Waterson hizo la siguiente tibia declaración sobre el informe de Silber:

“Tan pronto se anunció la acusación de Randy Samuels personalmente le pedí a un estimado ingeniero de software, el Dr. Wesley Silber, que llevara a cabo una indagación objetiva sobre los procedimientos de aseguramiento de la calidad en Silicon Techtronics. Como gerente ejecutivo de este proyecto, siempre he insistido en que la calidad es lo primero, a pesar de lo que hayan podido leer en los periódicos.

“Le pedí al profesor Silber que condujera una investigación objetiva de todos los aspectos de aseguramiento de la calidad de Silicon Techtronics. Prometí al Profesor Silber que tendría acceso a toda la información relevante a esta infortunada situación. Le dije en una reunión frente a frente en mi oficina que debía proseguir la investigación hasta su final sin importar a donde terminara, sin importar las implicancias.

“Basándome en la información que yo recibía de mis gerentes, nunca se me ocurrió que hubiera un problema de que los procedimientos de aseguramiento de la calidad fueran ya sea débiles o deliberadamente alterados. Quiero asegurarle al público que la o las personas responsables de esta falta en el aseguramiento de la calidad del software dentro de la División Robótica de Silicon Techtronics serán exhortados a encontrar trabajo en otro lado.”

Roberta Matthews, viuda de Bart Matthews, el operador del robot que fue muerto en el incidente, habló telefónicamente desde su casa con el Sentinel-Observer. “Aún quiero ver al Sr. Samuels condenado por lo que le hizo a mi marido. No entiendo de dónde viene toda la conmoción. El hombre que asesinó a mi esposo debería haber probado su propio software!”

El SENTINEL-OBSERVER entrevistó al Profesor Silber justo antes de su conferencia de prensa. En las paredes de su oficina estaban colgados numerosos premios recibidos a raíz de su trabajo en el campo de ingeniería de software y aseguramiento de la calidad del software. Comenzamos la entrevista pidiendo al Profesor Silber que explicara por qué a veces el software no es confiable. Contestó a nuestra pregunta citando la enorme complejidad del software.

“Los grandes programas de computadora son indiscutiblemente los artefactos más complejos creados por la mente humana”, explicó el Profesor Silber sentado frente a un monitor de grandes dimensiones. “En algún momento un programa de computación está en uno de los tantos estados posibles, y hay una imposibilidad práctica de asegurar que el programa se comportará como corresponde en cada uno de esos estados. No tenemos el tiempo suficiente para hacer tal tipo de prueba exhaustiva. De modo que usamos estrategias de prueba o heurísticas que muy probablemente encontrarán los errores o bugs, si es que existe alguno.”

El Profesor Silber ha publicado numerosos papers sobre ingeniería de software. Estuvo en la primera plana cuando el año pasado publicó su lista de “Aerolíneas a Evitar Si su Vida Dependiera de Ello”. Esa lista enumeraba las aerolíneas de cabotaje que él consideraba irresponsables por su compra de aviones que están controlados casi por completo por software de computación.

Poco tiempo después de los cargos contra Randy Samuels en el caso del “robot asesino”, el gerente Ejecutivo de Silicon Techtronics, Michael Waterson, pidió al Profesor Silber que condujera una revisión objetiva de los procedimientos de aseguramiento de la calidad de Silicon Techtronics. La intención de Waterson era contrarrestar la mala publicidad de su empresa luego de las acusaciones de Samuels.

“El Aseguramiento de la Calidad” se refiere a aquellos métodos que usa un especialista de desarrollo de software para asegurar que el software es confiable, correcto y robusto. Estos métodos se aplican a todo lo largo del ciclo de vida de desarrollo del producto de software. En cada etapa se aplican los métodos de aseguramiento de calidad adecuados. Por ejemplo, cuando un programador escribe código, una medida de aseguramiento de calidad es probar el código confrontándolo en verdad con los datos de prueba. Otro método sería correr programas especiales, llamados analizadores estáticos, confrontándolo con el nuevo código. Un analizador estático es un programa que busca patrones sospechosos en los programas, patrones que podrían indicar un error o bug.

Estas dos formas de aseguramiento de calidad son denominadas pruebas dinámicas y pruebas estáticas, respectivamente.

El software consiste de componentes discretos o unidades que eventualmente se combinan para crear un sistema más grande. Las unidades mismas deben ser probadas, y este proceso de prueba individual de las unidades es llamado prueba unitaria. Cuando las unidades se combinan, se deben probar los subsistemas integrados y este proceso se llama prueba de integración.

El Profesor Silber comentó al SENTINEL-OBSERVER sobre su trabajo en Silicon Techtronics: “Mike [Waterson] me dijo de ir allí [a la compañía] y conducir una revisión de sus procedimientos de prueba de software y de hacer públicos mis hallazgos. Mike parecía confiado, tal vez debido a lo que le habían dicho sus gerentes, en el sentido de que no encontrarían nada malo en los procedimientos de aseguramiento de calidad de Silicon Techtronics.”

Luego de arribar a Silicon Techtronics, el Profesor Silber centró su atención en los procedimientos para ensayo dinámico de software en la compañía.

Ayudado por un grupo de graduados, el Profesor Silber descubrió una discrepancia entre el comportamiento real de la sección del código del programa (escrito por Randy Samuels) que causó que el robot Robbie CX30 matara a su operador, y el comportamiento según se lo registró en la documentación de pruebas de Silicon Techtronics. Este descubrimiento en realidad fue hecho por Sandra Henderson, una estudiante graduada en ingeniería de software que está completando su doctorado con el Profesor Silber. Entrevistamos a la Sra. Henderson en uno de sus laboratorios de computación para egresados en la Universidad de Silicon Valley. “Encontramos un problema en la prueba de la unidad”, explicó la Sra. Henderson. “Acá están los resultados de la prueba, que nos dio el Sr. Waterson en Silicon Techtronics, que se supone están hechos para código C [lenguaje de programación] que Randy Samuels escribió y que causó el incidente del robot asesino. Como puede ver, todo está claramente documentado y organizado. Hay dos juegos de prueba: uno basado en una prueba de caja blanca y otro en una prueba de caja negra. Basándonos en nuestros propios estándares para probar software, estos juegos de prueba están bien diseñados, completos y rigurosos. “La prueba de caja negra implica ver la unidad de software (o sus componentes) como una caja negra que tiene comportamientos predecibles de input y output. Si en el juego de prueba el componente demuestra los comportamientos esperados para todos los inputs, entonces pasa la prueba. Los juegos de prueba están diseñados para cubrir todos los comportamientos “interesantes” que una unidad podría mostrar pero sin tener conocimiento alguno sobre la estructura o naturaleza del código en realidad. La prueba de caja blanca implica cubrir todos los pasos posibles a través de la unidad. Así, la prueba de caja blanca se hace con vasto conocimiento de la estructura de la unidad. En la prueba de caja blanca, el juego de prueba debe causar que cada sentencia del programa se ejecute por lo menos una vez de modo que ninguna quede sin ser ejecutada.

Sandra Henderson prosiguió explicando el significado de la prueba de software. “Ni la prueba de caja blanca ni de caja negra “prueban” que un programa esté correcto. No obstante, los probadores de software, tales como los que se emplean en Silicon Techtronics, pueden volverse bastante expertos en el diseño casos de prueba para descubrir nuevos bugs en el software. La actitud apropiada es que una prueba es exitosa cuando se encuentra un bug.” Básicamente, el probador le da un juego de especificaciones y hace lo mejor de sí para demostrar que el código que está probando no satisface sus especificaciones”, explicó la Sra. Henderson.

La Sra. Henderson luego mostró a este reporter los resultados de las pruebas que ella en verdad obtuvo cuando corrió el código crítico del “robot asesino” usando los juegos de prueba de la compañía, tanto para ensayo de caja blanca como de caja negra. En muchos casos, los resultados registrados en los documentos de prueba de la compañía no fueron los mismos que los generados por el verdadero código del robot asesino.

Durante su entrevista de ayer con el SENTINEL-OBSERVER, el Profesor Silber discutió la discrepancia. “Verá, el software que en verdad fue entregado junto con el robot Robbie CX30 no fue el mismo que el que supuestamente fue probado, ¡por lo menos de acuerdo con estos documentos!. Hemos podido determinar que el verdadero “código asesino”, tal como lo llamamos, fue escrito después de que se condujeron supuestamente las pruebas del software. Esto sugiere varias posibilidades. Primero, el proceso de prueba de software, por lo menos para esta parte crítica del software, fue falseado deliberadamente. Todos sabemos que hubo una enorme presión para tener listo a este robot en una fecha determinada. Otra posibilidad es que hubo una cierta dificultad en la versión de la gerencia en Silicon Techtronics, en cuanto a que el código correcto fue verdaderamente escrito, y probado con éxito, pero en el producto entregado se insertó el código equivocado.”

Solicitamos al Profesor Silber que explicara que quería decir con “versión de la gerencia”. “En un proyecto dado, un componente dado de software puede tener varias versiones, versión 1, versión 2, etc.

Esto refleja la evolución de ese componente a medida que avanza el proyecto. Se necesita tener algún tipo de mecanismo para tener control de las versiones de los componentes de software en un proyecto tan complejo como este. Tal vez el probador de software probó una versión correcta del código de dinámica del robot, pero en realidad se entregó una versión equivocada del mismo. No obstante, esto trae a colación una pregunta en cuanto a qué pasó con el código correcto.”

El Profesor Silber se reclinó en su sillón. “Realmente esto es una gran tragedia. Si el código asesino hubiese sido pasado por el proceso de prueba de un modo honesto, el robot nunca hubiese asesinado a Bart Matthews. Entonces, la pregunta es, ¿qué pasaba en Silicon Techtronics que no permitió una prueba honesta del código crítico?”

El SENTINEL-OBSERVER preguntó al Profesor Silber si estaba de acuerdo con el concepto de que la interfaz del usuario fue la primordial culpable en este caso. “No creo en el argumento que esgrime mi colega, el Profesor Gritty, de que toda la culpabilidad en este caso pertenece al diseñador o diseñadores de la interfaz. Conuerdo con ciertas cosas que dice, pero no con todo. Debo preguntarme a mí mismo si Silicon Techtronics estaba poniendo mucho énfasis en la interfaz del usuario como la última línea de defensa contra el desastre. Esto es, ellos sabían que había un problema, pero pensaron que la interfaz del usuario podría permitirle al operador manejarlo.”

El SENTINEL-OBSERVER preguntó entonces al Profesor Silber sobre los cargos que se le hacían en cuanto que nunca debería haber aceptado la designación de Waterson para conducir una investigación objetiva del accidente. Las críticas señalan que la Universidad de Silicon Valley, y en particular el Profesor Silber, tenían muchos intereses comunes con Silicon Techtronics, y de ese modo no podía ser elegido para conducir una investigación objetiva.

“Pienso que mi informe habla por sí mismo,” replicó el Profesor Silber, visiblemente molesto por nuestra pregunta. “Ya les he dicho a ustedes los periodistas una y otra vez que no se trató de una investigación gubernamental sino de una interna de la corporación, de modo que creo que Silicon Techtronics tenía el derecho de elegir a quien se le ocurriera. Creo que yo les resultaba una persona con integridad.”

Ayer tarde, Sam Reynolds, el gerente de Proyecto del CX30 contrató a una abogada, Valerie Thomas. La Sra. Thomas hizo estas declaraciones en favor de su cliente:

“Mi cliente está escandalizado de que alguien de Silicon Techtronics haya podido engañar al Profesor Silber en lo concerniente a las pruebas de software del robot Robbie CX30. El Sr. Reynolds asegura que el software fue probado y que él y otros sabían muy bien el hecho de que había algo que no funcionaba en el software de dinámica del robot. Sin embargo, el Sr. Ray Johnson, el superior inmediato de mi cliente en Silicon Valley, decidió que el robot fuera entregado a Cybernetics, Inc., basándose en la teoría del Sr. Johnson: “Nada es tan blanco como la nieve”.³

Conforme a esa teoría, el software estaba casi libre de bugs y por ende podía ser liberado. Según el Sr. Johnson, el riesgo de falla era muy pequeño y el costo por demorar más la entrega del robot era muy alto. “Según mi cliente, el Sr. Johnson creyó que las condiciones del medio ambiente que podría llegar a disparar un comportamiento errático y violento del robot eran extremadamente improbables de ocurrir. Aún más, el Sr. Johnson creyó que el operador del robot no podría estar en peligro debido a que la interfaz del usuario fue diseñada de modo de permitir al operador detener el robot fijo en sus guías en el caso de un movimiento del robot que comprometiera la vida del operador.”

El Sr. Johnson, Jefe de la División Robótica de Silicon Techtronics, no pudo ser ubicado para obtener sus comentarios. Randy Samuels será juzgado el mes entrante en la Corte de Silicon Valley. Cuando se lo contactó por teléfono, Samuels derivó todas las preguntas a su abogado, Alex Allendale.

Allendale tenía esto para decir con respecto a los descubrimientos del Profesor Silber:

“Mi cliente remitió el software en cuestión del modo usual junto con la documentación usual y con la esperanza de que su código fuera probado exhaustivamente. Desconocía hasta el momento de que saliera a la luz el informe del Profesor Silber, que el código involucrado en esta terrible tragedia no había sido probado adecuadamente o que los resultados de prueba pudieran haber sido falsificados.

“El Sr. Samuels nuevamente quiere expresar su gran pesar por este accidente. Él, más que nadie, quiere que se haga justicia en este caso. El Sr. Samuels nuevamente extiende sus más sentidas condolencias a la Sra. Matthews y a sus hijos.”

³Teoría “Ivory Snow”

Empleado de Silicon Techtronics Admite Falsificación de las Pruebas del Software

Mensajes Tomados del Correo Electrónico Revelan Nuevos Detalles en el Caso del “Robot Asesino”

Una Asociación de Computadores Científicos Lanza una Investigación sobre Violaciones al Código de Ética

Especial para el SENTINEL-OBSERVER DE Silicon Valley

***Silicon Valley, EEUU
por Mabel Muckraker***

Cindy Yardley, una probadora de software de Silicon Techtronics, admitió hoy que ella fue la persona que creó las pruebas de software fraudulentas del “robot asesino”. Las pruebas fraudulentas fueron reveladas a principios de esta semana por el profesor Wesley Silber de la Universidad de Silicon Valley, con lo que se ha dado en llamar “El Informe Silber”.

Se cuestionan los procedimientos de aseguramiento de la calidad que fueron realizados en el código del programa escrito por Randy Samuels, el programa acusado por asesinato no premeditado en el incidente del robot asesino. El Informe Silber afirma que los resultados de las pruebas reflejados en documentos internos de Silicon Techtronics son inconsistentes con respecto a los resultados de las pruebas obtenidos cuando fue probado el verdadero código del robot asesino.

Ayer al mediodía, en un acontecer inesperado, anunció su renuncia al cargo de Jefe de Seguridad de Silicon Techtronics, el Sr. Max Worthington, en una conferencia de prensa que fue transmitida en vivo por CNN y otros informativos.

Worthington sacudió a los periodistas cuando comenzó su conferencia de prensa con el anuncio “Yo soy Marta”.

Worthington describió de este modo sus responsabilidades en Silicon Techtronics: “Básicamente, mi trabajo era proteger a Silicon Techtronics de todos los enemigos, locales y extranjeros. Por extranjeros quiero significar adversarios de otras corporaciones. Mi papel era más que nada de dirección. Aquellos que trabajaban bajo mi supervisión tenían muchas responsabilidades, incluyendo la de proteger la planta en sí, estar alertas por espionaje industrial e incluso sabotaje. También yo era responsable de vigilar a los empleados que pudiesen estar abusando de drogas o que de algún modo estuviesen siendo desleales con Silicon Techtronics.”

Luego Worthington apuntó a una pila de volúmenes que había en una mesa a su izquierda. “Estos volúmenes representan tan solo algunos de los relevamientos electrónicos de empleados que yo hice a lo largo de los años para mi superior, el Sr. Waterson. Estas son impresiones de mensajes por E-mail que los empleados de Silicon Techtronics se enviaron entre sí y a personas de otros sitios. Puedo decir con gran certeza que nunca jamás se le dijo a ningún empleado que se hacía este tipo de requisa electrónica. No obstante, creo que la evidencia muestra que algunos empleados sospechaban que esto podía estar pasando.”

Varios periodistas preguntaron a los gritos quién en Silicon Techtronics estaba al tanto de esta requisa.

Worthington respondió. “Nadie sabía de esto a excepción del Sr. Waterson y yo, y uno de mis asistentes que era el responsable de en verdad conducir el monitoreo. Mi asistente producía un informe especial, resumiendo toda la actividad por E-mail de la semana, y ese informe era para que lo viera Waterson y yo solamente. Si se lo solicitaba, mi asistente podía dar un recuento más detallado de las comunicaciones electrónicas”.

Worthington explicó que estaba poniendo a disposición de la prensa las transcripciones del correo electrónico porque quería que saliera a luz toda la verdad sobre Silicon Techtronics y el incidente del robot asesino.

Los mensajes de E-mail entre empleados de Silicon Techtronics en verdad revelaron nuevas facetas del caso. Un mensaje de Cindy Yardley al Jefe de División Robótica, Ray Johnson, indica que ella falsificó a su pedido los resultados de las pruebas. Aquí está el texto del mensaje:

a: Ray Johnson
de: Cindy Yardley
re: Software de Samuels

Terminé de crear los resultados de las pruebas de software para ese software problemático, según tu idea de usar una simulación en vez del software propiamente dicho. Adjunto encontrarás el documento de prueba modificado, mostrando la simulación exitosa.
¿Le deberíamos decir a Randy sobre esto? Cindy

La respuesta de Johnson al mensaje de Yardley sugiere que él sospechaba que el correo electrónico podía no ser seguro:

En respuesta a: Cindy Yardley
de: Ray Johnson
re: Software de Samuels

Sabía que podría contar contigo. Estoy seguro de que tu dedicación a Silicon Techtronics te será pagada con creces. Por favor, en el futuro usá un medio de comunicación más seguro cuando discutimos este tema. Te aseguro que el modo en que manejamos esto fue completamente transparente, pero yo tengo mis enemigos acá mismo en la propia SiliTech.

Ray.

Estas comunicaciones fueron intercambiadas justo unos días antes que se enviara el robot Robbie CX30 a Cybernetics Inc. Este hecho es importante porque las pruebas de software falsificadas no fueron parte de un encubrimiento en el incidente del robot asesino. Estos hechos parecen indicar que el propósito de falsificar las pruebas de software era asegurarse de que el robot Robbie CX30 fuera entregado a Cybernetics, Inc. en la fecha que fue negociada entre Silicon Techtronics y Cybernetics.

Las transcripciones del correo electrónico revelan que hubieron repetidos mensajes de Ray Johnson a diferentes personas en el sentido de que la División Robótica iba a ser cerrada definitivamente si el proyecto Robbie CX30 no era completado en término. En uno de los mensajes, diserta con su líder de proyecto, Sam Reynolds, acerca de la “Teoría Ivory Snow”.

a: Sam Reynolds
de: Ray Johnson
re: ¡no seas un perfeccionista!

Sam:

Tú y yo hemos tenido nuestras diferencias, pero debo decirte que personalmente me caes bien. Por favor entiende que todo lo que hago es con el propósito de SALVAGUARDAR TU TRABAJO Y EL TRABAJO DE TODOS EN ESTA DIVISIÓN. Yo te veo a tí y a toda la gente que trabajan para mí en la División Robótica como mi familia.

Waterson fue claro: quiere tener el proyecto del robot completado en término. Y punto. Entonces, no tenemos otro recurso más que el de "Ivory Snow". Sabes lo que quiero decir con eso. No tiene que ser perfecto. La interfaz del usuario es nuestro respaldo si esta versión del software para el robot tiene algunas fallas. El operador del robot va a estar seguro porque podrá cancelar cualquier movimiento del robot en cualquier momento. Concuerdo contigo en cuanto a que los requerimientos no funcionales son en algunas partes demasiado vagos. Lo ideal sería que si estos no fueran tiempos de apuros, cuantificáramos el tiempo que le llevaría al operador detener el robot en un caso de accidente. Sin embargo no podemos renegociar esto ahora. Como tampoco tenemos tiempo para diseñar requerimientos no funcionales nuevos y más precisos. No puedo enfatizar suficientemente de que estos son tiempos de apurarse. A Waterson no le cuesta nada deshacerse de toda la División Robótica. Sus amigos del Wall Street sólo le van a decir ¡Felicitaciones!. Verás, para Waterson nosotros tan sólo somos del montón.

Ray.

En este mensaje Ray Johnson parecería estar menos preocupado por la seguridad de comunicarse por correo electrónico.

El SENTINEL-OBSERVER entrevistó ayer por la tarde a Cindy Yardley en su propia casa. No se pudieron contactar ni a Ray Johnson ni a Sam Reynolds.

La Srta. Yardley estaba notoriamente ofuscada porque sus mensajes por E-mail fueran dados a conocer a la prensa. "De alguna forma me siento aliviada. Sentí una enorme culpa cuando ese hombre fue muerto por un robot que yo ayudé a producir. Una tremenda culpa."

El SENTINEL-OBSERVER preguntó a la Srta. Yardley si es que ella había hecho una elección ética al acceder a falsear los resultados de las pruebas de software. Respondió con gran emoción. "Nada, pero nada a lo largo de mi experiencia y background me preparó para algo como lo que me pasó. Estudié ciencias de la computación en una universidad de gran nivel y allí me enseñaron sobre pruebas de software, ¡pero jamás me dijeron que alguien con poder sobre mí me pediría generar una prueba falseada!"

"Cuando Johnson me pidió que lo hiciera, me llamó a su oficina, como para mostrarme las trampas del poder; verá, algún día me gustaría estar en un puesto gerencial. Me senté en su oficina y vino directamente y me dijo "Quiero que falsifiques los resultados de las pruebas del software de Samuels. No quiero que Reynolds se entere de nada de esto".

Yardley contuvo las lágrimas. “Me aseguró de que probablemente nadie vería jamás los resultados de las pruebas dado que el robot era perfectamente seguro. Era tan sólo una cuestión interna, un tema de prolijidad, en caso de que alguien de Cybernetics o de un puesto alto dentro de la corporación le diera curiosidad de ver los resultados de las pruebas. Le pregunté si estaba seguro de que el robot era seguro y todo eso y me dijo “¡Es seguro! La interfaz del usuario es nuestra línea de defensa. En alrededor de seis meses podemos enviar una segunda versión del software del robot y para ese entonces este problema de Samuels estará resuelto.”

Yardley se reclinó en su asiento como si lo que dijera a continuación necesitara de un énfasis especial. “Entonces me dijo que si yo no falsificaba las pruebas, todos los de la División Robótica perderían sus trabajos. Sobre esa base decidí falsificar las pruebas, trataba de proteger mi trabajo y el de mis compañeros.”

La Srta. Yardley está al presente cursando un grado de Maestría en Administración de Empresas en la Universidad de Silicon Valley.

Luego el SENTINEL-OBSERVER pregunto a la Srta. Yardley si aún sentía que había tomado una decisión ética, en vista de la muerte de Bart Matthews. “Creo que fui manipulada por Ray Johnson. Él me dijo que el robot era seguro”.

Otra revelación, contenida en las transcripciones del correo electrónico dadas a conocer, fue el hecho de que Randy Samuels hurtó parte del software que usó en el proyecto del robot asesino. Este hecho se reveló en un mensaje que Samuels envió a Yardley cuando ella probó por primera vez su software y dio resultados erróneos:

En respuesta a: Cindy Yardley
de: Randy Samuels
re: maldito si lo sé

Por mi vida, no puedo entender qué es lo que anda mal en esta función `balancear_brazo()`. Verifiqué la fórmula de la dinámica del robot una y otra vez y pareciera estar implementada correctamente. Como sabes, la función `balancear_brazo()` invoca a 14 funciones diferentes. A cinco de ellas las tomé tal cual del paquete estadístico PACKSTAT 1-2-3. ¡Por favor no se lo digas a nadie! No son éstas las que causarían el problema, ¿o sí?

Randy.

Los expertos le dijeron al SENTINEL-OBSERVER que tomar software de paquetes comerciales de software como el PACKSTAT 1-2-3 es una violación a la ley. El software tal como el inmensamente popular PACKSTAT 1-2-3 está protegido por el mismo copyright que protege al material impreso.

Mike Waterson, Presidente Ejecutivo de Silicon Techtronics, emitió una enojosa declaración porque Max Worthington había dado a conocer las transcripciones del correo electrónico “confidencial”. Las declaraciones de Waterson decían, en parte, que “Yo le pedí a nuestros abogados que intervinieran en este tema. Consideramos que esas transcripciones son propiedad exclusiva de Silicon Techtronics. Nuestra intención es efectuar cargos ya sea civiles o criminales contra el Sr. Worthington.”

Como reacción a lo ocurrido ayer en el caso del robot asesino, la ACM o Association for Computer Machinery anunció su intención de investigar si algún miembro de la ACM de Silicon

Techtronics ha violado el Código de Ética de la ACM. La ACM es una asociación internacional de computadores científicos con 85.000 miembros.

La Dra. Turina Babbage, presidente de la ACM, hizo una declaración en la Conferencia de Ciencias de la Computación de ACM que se lleva a cabo cada invierno y que esta temporada se hará en Duluth, Minnesota.

Un extracto de las declaraciones de la Dra. Babbage sigue a continuación:

“Todos los miembros de la ACM están ligados por el Código de Ética y Conducta Profesional de la ACM [NOTA AL PIE: Un borrador de este código fue dado a conocer en Comunicaciones de la ACM, Mayo 1992. Por favor nótese que las declaraciones hechas por la ficticia Dra. Babbage contienen citas del verdadero código de ACM] Este código establece, en parte, que los miembros de ACM tienen el imperativo moral de contribuir con el bienestar de la sociedad y los hombres, evitar daños a terceros, ser honestos y confiables, dar crédito adecuado a la propiedad intelectual, acceder a los recursos de comunicación y de computación sólo cuando así lo estén autorizados, respetar la privacidad de terceros y honrar la confidencialidad.

Más allá de eso, existen responsabilidades profesionales tales como la obligación de cumplir los contratos, acuerdos y responsabilidades asignadas, y de dar evaluaciones profundas y completas de los sistemas de computación y de sus impactos, poniendo especial énfasis en los riesgos potenciales.

Varias de las personas involucradas en el caso del robot asesino son miembros de la ACM y hay causas para creer que han incurrido en violación del código de ética de nuestra asociación. Por lo tanto, estoy solicitando al directorio de la ACM designar una Fuerza de Tareas para investigar a los miembros de la ACM que puedan haber violado groseramente el código.

No tomamos este paso a la ligera. Esta sanción ha sido aplicada sólo rara vez, pero el incidente del robot asesino no sólo ha costado una vida humana, sino que ha causado mucho daño a la reputación de la profesión de computación.”

La Revista Dominical del SENTINEL-OBSERVER - Una Conversación con el Dr. Harry Yoder

por
Robert Franklin

Harry Yoder es una figura muy bien conocida en el campo universitario de Silicon Valley. El profesor de Tecnología y Ética de la Computación de Samuel Southerland ha escrito numerosos artículos y textos sobre ética y el impacto social de las computadoras. Sus clases son muy famosas, y muchos de sus cursos están completos mucho antes de que finalice el período de inscripción. El Dr. Yoder ha recibido su Doctorado en ingeniería eléctrica del Instituto de Tecnología de Georgia en 1958. En 1976 recibió un grado en “Maestría en Divinidad” del Harvard Divinity School. En 1983 recibió un Master en Ciencias de la Computación de la Universidad de Washington. Ingresó en la facultad de la Universidad de Silicon Valley en 1988.

Entrevisté al Dr. Yoder en su oficina del campus. Mi intención era obtener su reacción con respecto al caso del robot asesino y “leer su pensamiento” sobre los temas éticos que involucra el caso.

SENTINEL-OBSERVER : Ir de la ingeniería eléctrica al estudio de religión parece un gran salto.

Yoder: Yo era un ingeniero electricista por profesión, pero todos los seres humanos tienen una vida interior, ¿no lo cree así?

SENTINEL-OBSERVER : Sí

Yoder: De qué se trata su vida interior?

SENTINEL-OBSERVER : Trata de hacer lo correcto. También se trata de lograr la excelencia en lo que hago. ¿Es eso lo que lo llevó a la Escuela de Divinidad de Harvard? ¿Usted quería clarificar su vida interior?

Yoder: Sucedían muchas cosas en la Escuela de la Divinidad, y muchas de ellas eran muy poderosas. Sin embargo, más que nada yo quería comprender la diferencia entre lo que estaba bien y lo que estaba mal.

SENTINEL-OBSERVER : ¿Y qué hay de Dios?

Yoder: Sí, estudié mi propia religión cristiana y a la mayoría de las religiones del mundo, y todas ellas tenían cosas interesantes que decir acerca de Dios. No obstante, cuando yo discuto sobre ética en un foro tal como este, que es secular, o cuando discuto ética en mis cursos de ética en la computación, no coloco a esa discusión en un contexto religioso. Pienso que la fe religiosa puede ayudarle a una persona a abrazar la ética, pero por otra parte, todos sabemos que ciertas personalidades notorias que se han autopronunciado religiosas han sido altamente no éticas. De este modo, cuando yo discuto sobre ética de la computación, el punto de partida no es la religión, sino más bien un acuerdo común entre mis estudiantes y yo de que queremos ser gente ética, que el luchar por la excelencia ética es una tarea humana que vale la pena. Por lo menos, lo que no queremos es herir a otros, no queremos mentir, robar, hacer trampas, asesinar, etc.

SENTINEL-OBSERVER : ¿Quién es el responsable de la muerte de Bart Matthews?

Yoder: Por favor discúlpeme si lo remito nuevamente a la escuela de la Divinidad de Harvard, pero creo que uno de mis profesores de allí tiene la respuesta correcta a su pregunta. Este profesor era un hombre mayor, tal vez de setenta años, de la Europa Oriental, un rabino. Este rabino dijo que de acuerdo al Talmud, una tradición antigua de la ley judía, si se derrama sangre inocente en un pueblo, entonces los líderes de ese pueblo deben ir a los límites del mismo y realizar un acto de penitencia. Esto es además de la justicia que se le aplicará a la persona o personas que cometieron el asesinato.

SENTINEL-OBSERVER : Ese es un concepto interesante.

Yoder: ¡Y uno de verdad! Un pueblo, una ciudad, una corporación son sistemas en que la parte está ligada al todo y el todo a la parte.

SENTINEL-OBSERVER : Usted quiere decir que los líderes de Silicon Techtronics, tales como Mike Waterson y Ray Johnson, deberían haber asumido la responsabilidad por este incidente desde el vamos. Además, tal vez otros individuos, como ser Randy Samuels y Cindy Yardley, comparten una carga especial de responsabilidad.

Yoder: Sí, responsabilidad, no culpabilidad. La culpabilidad es un concepto legal y la culpabilidad o la inocencia de las partes involucradas, sean ya en lo criminal o lo civil, será decidida en la corte. Estimo que una persona es responsable por la muerte de Bart Matthews si su acción ha ayudado a causar el incidente - es una cuestión de causalidad, independiente de los juicios éticos y legales. Las cuestiones de responsabilidad podrían serle de interés a los ingenieros de software y gerentes, quienes tal vez querrían analizar qué es lo que anduvo mal, de modo de evitar que similares problemas ocurrieran en el futuro.

Mucho de lo que salió de los medios con respecto a este caso indica que Silicon Techtronics era una organización enferma. Esa enfermedad creó el accidente. ¿Quién creó la enfermedad? La gerencia creó a esa enfermedad, pero también los empleados que no tomaron las decisiones éticas correctas contribuyeron con la misma.

Tanto Randy Samuels como Cindy Yardley eran recién egresados. Se graduaron en ciencias de la computación y su primera experiencia en el mundo laboral fue en Silicon Techtronics. Uno debería preguntarse si recibieron alguna enseñanza sobre ética. Relacionado a esto está la cuestión de si alguno de ellos tenía con anterioridad experiencia en trabajos en grupo. En el momento en que se los asignó al desarrollo del robot asesino, ¿ellos vieron la necesidad de ser personas éticas? ¿Vieron que el éxito como profesionales requiere de un comportamiento ético? Hay mucho más para ser un científico en computación o un ingeniero de software que tan sólo la habilidad y el conocimiento de la técnica.

SENTINEL-OBSERVER : Sé con seguridad que ninguno de los dos tomó cursos sobre ética o ética de la computación.

Yoder: Lo sospechaba. Veamos a Randy Samuels. Basándome en lo que leí en su periódico y en otros lados, era básicamente de los del tipo “hacker”. Amaba la computación y la programación. Comenzó a programar en los primeros años de la secundaria y continuó a lo largo de toda la carrera universitaria. El punto importante es que Samuels aún era un “hacker” cuando entró en Silicon Techtronics y ellos le permitieron que él siguiera siendo así.

Estoy usando el término “hacker” de un modo un tanto peyorativo y tal vez esto no sea justo. El punto que estoy tratando de remarcar es que Samuels nunca maduró más allá de su angosto enfoque como hacker. En Silicon Techtronics, Samuels aún mantuvo esta actitud en lo que hacía a sus funciones de programador, la misma que tenía cuando estaba en la secundaria. Su percepción de la vida y de sus responsabilidades no creció. Él no maduró. No hay evidencia de que tratara de desarrollarse y convertirse en una persona ética.

SENTINEL-OBSERVER : Una dificultad, en lo que hace a enseñar ética, es que en general los estudiantes no les gusta que se les diga “esto está bien y aquello está mal”.

Yoder: Los alumnos necesitan entender que el tocar temas de ética es parte de ser computadores científicos o ingenieros de software profesionales.

Una cosa que me ha fascinado acerca de la situación de Silicon Techtronics es que a veces es difícil ver los límites entre lo legal, lo técnico y lo ético. Los temas técnicos involucran temas de gerencia y de computación. He llegado a la conclusión de que este desvanecimiento de los límites resulta del hecho de que la industria de software aún se encuentra en pañales. Los temas éticos surgen abruptamente en parte porque hay una ausencia de lineamientos técnicos y legales.

En particular, no existen prácticas normalizadas para desarrollar y probar software. Hay estándares, pero no lo son realmente. Una broma muy común entre los computadores científicos es que lo bueno de los estándares es que hay muchos para elegir.

Ante la ausencia de prácticas normalizadas aceptadas universalmente para ingeniería de software, surgen muchos juicios de valor, probablemente más que para cualquier otra forma de producción. Por ejemplo, en el caso del robot asesino, hubo una controversia con respecto al uso del modelo de cascada versus el de prototipo. Debido a que no había un proceso de desarrollo de software estandarizado, esto se transformó en una controversia, y los temas éticos surgen por el modo en que se resuelve la controversia. Usted recordará que el modelo de cascada fue elegido no por sus méritos sino porque el gerente de proyecto tenía experiencia en éste.

SENTINEL-OBSERVER : ¿Usted cree que Cindy Yardley actuó éticamente?

Yoder: Al principio su argumento parece poderoso: ella, efectivamente, mintió, para así salvar los puestos de trabajo de sus compañeros, y por supuesto, el de ella. Pero, ¿siempre es correcto mentir, para crear una falsedad, en un marco profesional?

Un libro que he usado en mis cursos de ética de computación es el “Ethical Decision Making and Information Technology” (Toma de Decisión Ética y Tecnología de la Información) de Kallman y Grillo [NOTA AL PIE: Este texto es un texto real y está publicado por McGraw-Hill]. En este libro se dan algunos de los principios y teorías que están detrás de la toma ética de decisiones. Yo uso este y otros libros para ayudar a que los alumnos desarrollen sus apreciaciones sobre la naturaleza de dilemas éticos y toma ética de decisiones.

Kallman y Grillo presentan un método para la toma de decisión ética y parte de su método consiste en el uso de cinco pruebas: la prueba de la mamá: ¿le diría Ud. a su mamá lo que hizo?; la prueba de la TV: ¿le diría Ud. a una audiencia nacional de TV lo que hizo?; la prueba del olfato: ¿lo que Ud. hizo tiene mal olor?; la prueba de ponerse en los zapatos del otro: ¿le gustaría que el otro le haga lo que Ud. hizo?; y la prueba del mercado: ¿sería su acción una buena estrategia de venta?

Lo que hizo Yardley reprobó todas estas pruebas, pienso que casi todos concuerdan conmigo. Por ejemplo, ¿pueden imaginar a Silicon Techtronics usando una campaña publicitaria que diga algo como?:

“En Silicon Techtronics el software que usted recibirá de nosotros está libre de bugs, porque aún cuando haya uno, distorsionaremos los resultados de las pruebas para esconderlo, usted nunca se enterará.. **“La ignorancia es la felicidad”**

Esto demuestra que el altruismo aparente no es un indicador suficiente de un comportamiento ético. Uno podría preguntarse qué otros motivos no declarados tenía la Srta. Yardley. ¿Podría ser que la ambición personal la llevara a aceptar la explicación que le dio Ray Johnson y su afirmación de que el robot era seguro?

SENTINEL-OBSERVER : ¿Existe alguna fuente de guía ética para gente que se ve confrontada con un dilema ético?

Yoder: Algunas empresas dan guía ética, en la forma de políticas de la corporación, y existe un documento así en Silicon Techtronics, o por lo menos así me dijeron. Yo no lo vi. Un empleado también podría remitirse a los lineamientos éticos que proveen sociedades profesionales, tales como la ACM. Además, el o la empleada podría leer sobre el tema para obtener una mejor percepción de la toma ética de decisiones. Por supuesto que uno siempre debe consultar con su propia conciencia y con sus convicciones más profundas.

SENTINEL-OBSERVER : ¿Usted cree que Randy Samuels actuó éticamente?

Yoder: Robar software en el modo que lo hizo es tanto ilegal como no ético.

Pienso que el punto más importante con Randy Samuels nunca fue discutido en los medios de prensa. Honestamente dudo que Samuels tuviera el conocimiento necesario para su puesto. Este tipo de conocimiento se lo llama conocimiento de la especialidad. Samuels tenía conocimiento de computación y programación, pero no tenía un sólido conocimiento de física, en especial de la mecánica clásica. Su falta de conocimiento en el dominio de la aplicación fue una causa directa del horrible accidente. Si alguien con conocimientos de matemáticas, estadísticas y física hubiera programado al robot en lugar de Samuels, probablemente hoy Bart Matthews estuviera vivo. No tengo dudas de ello. Samuels malentendió la fórmula física porque no entendió su significado e importancia en la aplicación en el robot. Puede ser que la gerencia sea en parte responsable por la situación. Puede que Samuels les haya dicho acerca de sus limitaciones y la gerencia habrá dicho. “Y bueno, qué importa”

Samuels tenía dificultades en trabajar en equipo, hacer revisiones en conjunto, y programar sin egoísmo. ¿Es posible que estuviera intentando esconder su falta de experiencia en el área?

SENTINEL-OBSERVER : ¿Cree que Ray Johnson actuó éticamente?

Yoder: ¡Este tema del “Ivory Snow”! El problema con la teoría del “Ivory Snow” es que es tan solo eso, una teoría. Si fuera más que una teoría, o sea una metodología real para mantener la probabilidad de la falla dentro de límites estadísticamente determinados, como lo que se llama “ingeniería de software en sala limpia” [cleanroom software engineering], entonces habría menos culpabilidad en ese punto.

Basándome en la información que dispongo, la teoría de Ivory Snow fue tan sólo una racionalización para sacarse de encima a software fallado y entregarlo en término al cliente. Esta teoría sólo es válida, ética y profesionalmente, si al cliente se le informa de los bugs de los que se tiene conocimiento, o de impurezas, utilizando la jerga. En el caso de Silicon Techtronics, la teoría Ivory Snow funcionó así: ¡sabemos que no es puro, pero el cliente cree que sí lo es!

Desde luego, presionar a Cindy Yardley como lo hizo Ray Johnson tampoco es ético. ¿El creía en lo que le dijo a la Srta. Yardley, es decir, que el robot era seguro, o fue eso una mentira del momento? Si el creía que el robot era seguro, entonces ¿por qué cubrirse con pruebas falsas? Si la interfaz con el usuario era tan importante como la última línea de defensa, entonces ¿por qué evitar pruebas más rigurosas de la interfaz?

SENTINEL-OBSERVER : ¿Qué piensa de Mike Waterson en todo esto?

Yoder: Si Johnson es el padre de la teoría Ivory Snow, Waterson el es abuelo. Su exigencia de que el robot estuviera completado para una fecha determinada o de lo contrario “rodarían cabezas”, puede haber causado que Johnson formulara la teoría Ivory Snow. Verá, es evidente que Johnson pensaba que era imposible entregar a Cybernetics Inc. el robot CX 30 para una fecha determinada, a menos que el software fuera con bugs.

En muchos sentidos pienso que Waterson actuó sin ética e irresponsablemente. Pone a Sam Reynolds a cargo del proyecto del robot, cuando aún él, Reynolds, carecía de experiencia con robots e interfaces con el usuario modernas, Reynolds rechazó la idea de desarrollar un prototipo, lo que podría haber permitido el desarrollo de una mejor interfaz.

Waterson creó una atmósfera opresiva entre sus empleados, que en sí mismo es falta de ética. No sólo amenazó con despedir a todos los de la División Robótica si el robot no se terminaba a tiempo, sino que hurgó en comunicaciones por correo electrónico privadas de toda la corporación, un derecho controvertido que algunas empresas alegan tener.

Mi creencia personal es que este tipo de investigación es falta de ética. La naturaleza del E-mail es algo así como un híbrido de correspondencia común y conversación telefónica. Monitorear o espiar la correspondencia ajena está considerado no ético, tal como lo es interferir un teléfono. Por cierto, estas actividades también son ilegales bajo la mayoría de las circunstancias. O sea, creo que monitorear a los empleados del modo que lo hizo Waterson es un abuso de poder.

SENTINEL-OBSERVER : ¿Usted cree que en esto el fiscal tiene un caso?

Yoder: ¿Contra Randy Samuels?

SENTINEL-OBSERVER : Sí.

Yoder: Lo dudo, a menos que ella tenga información que hasta ahora no se ha hecho pública. El asesinato no premeditado, a mi entender, implica un tipo de acto irresponsable y negligente, que causa la muerte de un tercero. ¿Se aplica esta descripción a Samuels? Pienso que la mejor apuesta de la fiscal es hacer hincapié en su falta de conocimiento en el área de aplicación, si puede mostrarse que Samuels se involucró deliberadamente en un fraude.

La semana pasada leí que el 79% de la gente está a favor de la absolución. La gente es proclive a acusar a la compañía y a sus gerentes. La otra noche, uno de los noticieros dijo, “Samuels no es un asesino, es un producto de lo que lo rodea”.

SENTINEL-OBSERVER : Podría nuevamente decir su posición sobre el tema de la responsabilidad final en el caso del robot asesino?

Yoder: En mi mente, el tema de la responsabilidad del individuo versus la responsabilidad de la corporación, es un tema muy importante. La corporación creó un entorno en el que podían ocurrir este tipo de accidentes. Aún así, los individuos, dentro de ese sistema, actuaron sin ética e irresponsablemente, y fueron los que de hecho causaron el accidente. Una compañía puede crear un entorno que saca a flote lo peor de sus empleados, pero cada empleado también puede contribuir a empeorar ese ambiente corporativo. Este es un lazo cerrado que se alimenta a sí mismo, un sistema en el sentido clásico. Entonces, hay cierta responsabilidad de la corporación y cierta responsabilidad de los individuos en el caso del robot asesino.

SENTINEL-OBSERVER : Muchas gracias Profesor Yoder.