

Ejercicio sobre desastres ocasionados por sistemas basados en computadoras (Tema 1: Introducción a la Ingeniería del Software)

1. Nombrar todos los sistemas que fallaron por un problema en:

1. La programación o prueba del sistema software.

- Mariner Burgs Out (1962) [1]
- World War III... Almost(1983) [4]
- AT&T Lines Go Dead (1990) [7]
- Patriot Fails Soldiers (1991) [8]
- Disastrous Study (1999) [13]
- British Passports to Nowhere (1999) [14]

2. El diseño del software o del hardware.

- Hartford Coliseum Collapse (1978) [2]
- Pentium Fails Long Division (1993) [9]
- Ariane Rocket Goes Boom (1996) [10]
- Mars Climate Crasher (1998) [12]
- EDS Drops Child Support (2004) [19]

3. El estudio del problemas (análisis y especificación de requisitos o de recursos necesitados)

- Y2K (1999) [15]
- Dot-Bomb Collapse (2000) [16]
- FBI's Trilogy Terminated (2005) [20]

4. Una mala documentación o uso.

- Cancer Treatment to Die For (2000) [18]

5. La seguridad.

- CIA Gives the Soviets Gas (1982) [3]
- Love Virus (2000) [17]

6. Un deficiente análisis del riesgo

- Medical Machine Kills (1985) [5]
- Wall Street Crash (1987) [6]
- Skynet Brings Judgement Day (1997) [11]

2. De toda esta lista de desastres:

1.¿Cuál es el intruso?

El intruso es el de Skynet y el día del Juicio Final [11], ya que es el único que por suerte no ha ocurrido en la vida real, sino en la ficción de la saga Terminator.

2.¿Cuál consideras que es el peor de todos? ¿por qué?

Personalmente creo que los peores desastres son aquellos que se llevan alguna vida por delante, como el tratamiento fallido de cáncer [18], el fallo de la máquina de radiación [5] o el sistema de detección de misiles fallido en Arabia Saudí [8]. Aunque las catástrofes con un alto coste monetario o de bienes materiales tienen gran importancia, no se las puede equiparar al valor de una vida.

3.¿Cuáles están relacionados con la industria armamentista?

- Mariner Burgs Out (1962) [1]
- World War III... Almost(1983) [4]
- Patriot Fails Soldiers (1991) [8]
- Skynet Brings Judgement Day (1997) [11]

4.¿Cuáles están relacionados con la industria aeroespacial?

- Ariane Rocket Goes Boom (1996) [10]
- Mars Climate Crasher (1998) [12]

5. ¿Cuáles están relacionados con la industria sanitaria?

- Cancer Treatment to Die For (2000) [18]
- Medical Machine Kills (1985) [5]

6. ¿Cuales están relacionados con las finanzas?

- Y2K (1999) [15]
- Wall Street Crash (1987) [6]
- AT&T Lines Go Dead (1990) [7]
- Dot-Bomb Collapse (2000) [16]
- EDS Drops Child Support (2004) [19]

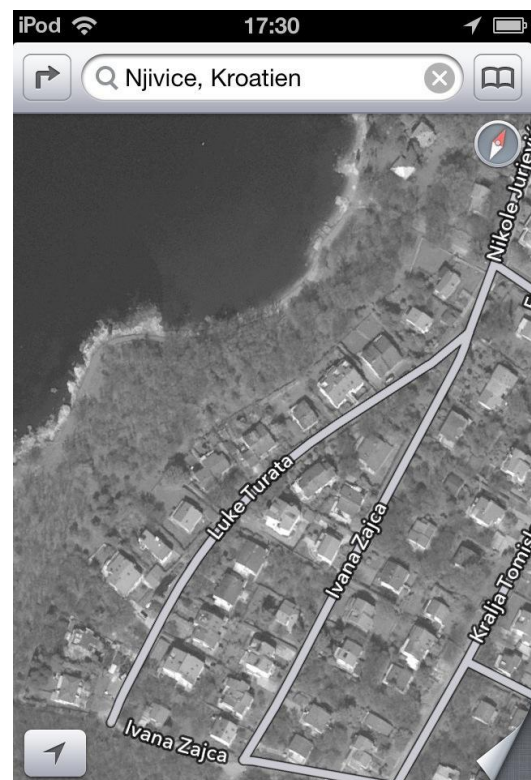
3. Investiga algún otro desastre ocasionado con un sistema software no incluido en esta lista y descríbelo.

En 2012 Apple lanzó al mercado la versión 6 de su sistema IOS. Uno de los cambios que se introdujeron en esta versión fue la sustitución de “Google Maps” por “Apple Maps” para intentar imponer su propio software al de Google. Sin embargo, el trabajo de mapeo y diseño fue, cuanto menos, un auténtico desastre. Había ciudades enteras que no estaban registradas, localizaciones erróneas, imágenes de satélite tapadas por las nubes y un largo etcétera. Se llegó a dar un caso en el que un usuario pidió a la aplicación que lo llevara al aeropuerto y Apple Maps lo llevó a las pistas de despegue en lugar de a la puerta del mismo. *¿Bug o feature?*

Las consecuencias de la mala programación y testeo de la aplicación fue la pérdida de la credibilidad de la empresa. Sin embargo, no parece que haya afectado mucho al beneficio de la empresa, debido al gran renombre que tiene sobre la comunidad.



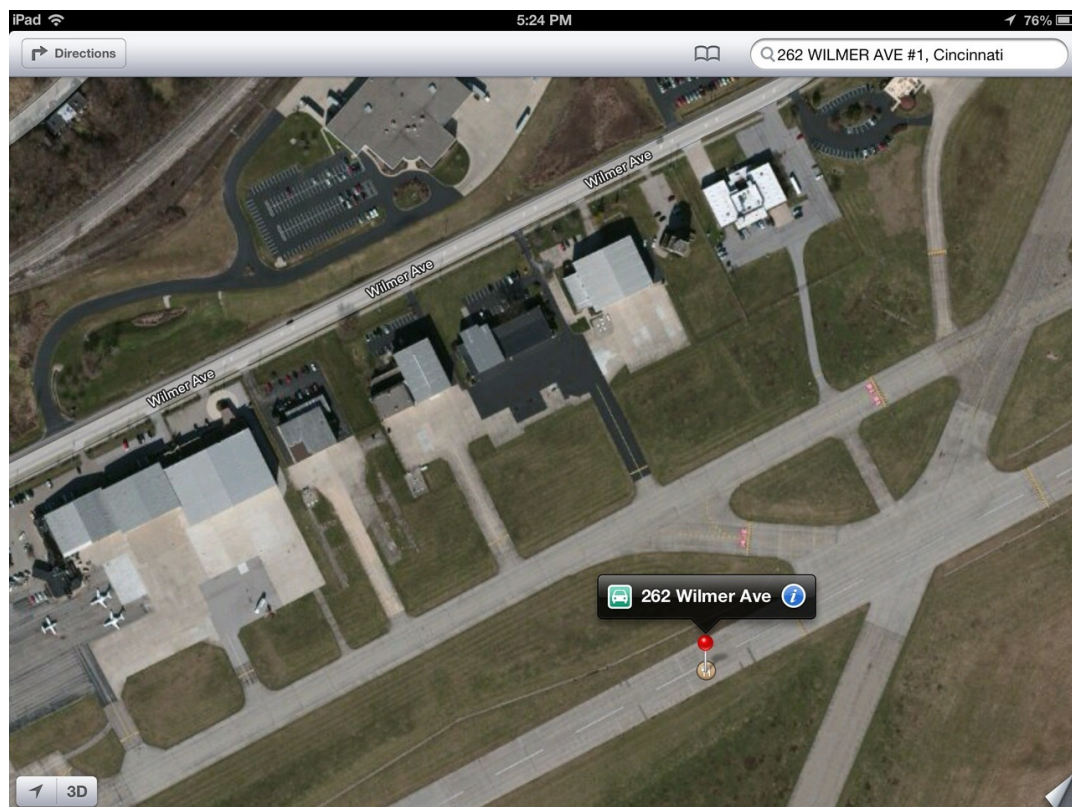
Una estructura, cuanto menos, curiosa



Alguien olvidó recargar la tinta de color



Escale el London Eye y habrá llegado a su destino



Literalmente, el aeropuerto

Carlos Sánchez Páez
2ºA (A2)
Fundamentos de Ingeniería del Software
Curso 2017-2018



Extraño pero cierto



Otro caso destacable es el de las baterías del Samsung Galaxy Note 7. Debido a que Apple sacó al mercado su nuevo iPhone antes de lo previsto, Samsung decidió llevar al mercado su Samsung Galaxy Note 7 sin antes hacer los oportunos testeos para poder hacerle competencia.

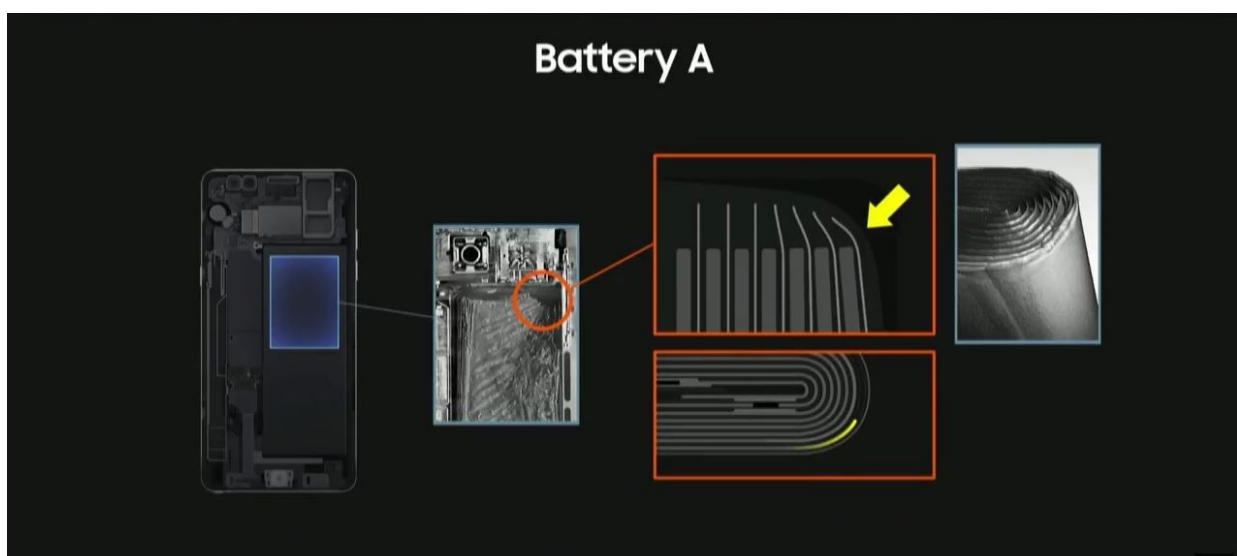
Resulta que en la fase de diseño hubo un gran problema con la batería al intentar hacer el dispositivo muy delgado pero a su vez admitir carga rápida. El caso es que se provocaba un contacto y la batería explotaba, siendo el teléfono peligroso para su uso.

Samsung tuvo que pedir disculpas públicamente e iniciar una campaña de recogida de los modelos en cuestión para sustituirlos por otro modelo. Como curiosidad, el kit de devolución incluía incluso guantes ignífugos. Incluso se llegó a aprobar una directiva en los aeropuertos por la que no se permitía que estos modelos de teléfono móvil embarcaran en los aviones, por seguridad.

Este gran fiasco por parte de Samsung le costó alrededor de 5.000.000\$ además de la pérdida de credibilidad de la empresa.



Sin duda, un buen kit de inicio



4. Expón, de forma breve, cuál es tu opinión respecto a todo lo aprendido ¿Cuál sería las posibles soluciones para evitar, en la medida de lo posible, los fallos en un sistema software?

Estos desastres ocasionados mayormente por un fallo de programación pueden ocasionar una gran tragedia. Como hemos visto, hay casos en los que incluso pueden acabar con vidas. Y todo por un fallo que podría haber sido detectado si no se hubiera tenido tanta prisa por llevar el producto al mercado para poder comenzar a obtener ganancias (por ejemplo, el caso de las baterías del Samsung Galaxy Note 7).

Desde mi punto de vista, lo mejor para evitar que un producto defectuoso salga al mercado, con las consecuencia desastrosas que puede conllevar, es una fase amplia de testeo y revisión. Esta fase debe ser desarrollada por personal que no haya tenido contacto con el desarrollo del código, ya que así se cuestiona la validez de cada una de las líneas del código y puede detectar fallos que se volvieron “invisibles” a ojos de los desarrolladores. También sería óptimo simular los distintos usos que va a tener el sistema para ver si responde correctamente.

Otra solución que se me ocurre es que existiera un organismo regulador que verificara la seguridad, eficiencia y eficacia del software para compañías más serias (empresa armamentística, sanidad, etc.). Es decir, que un software no se pudiera utilizar sin el veredicto positivo de este organismo.

Fuentes

- <https://www.fayerwayer.com/2013/01/apple-anuncia-que-gano-us-13-100-millones-el-ultimo-trimestre-del-2012/>
- <http://theamazingios6maps.tumblr.com/>
- http://www.nytimes.com/2012/09/27/technology/personaltech/apples-new-maps-app-is-upgraded-but-full-of-snags-review.html?smid=tw-nytimesbusiness&_r=0
- <https://www.intertech.com/Blog/15-worst-computer-software-blunders/>
- <https://www.cnet.com/news/apple-maps-in-ios-6-what-you-need-to-know-faq/>
- <https://www.cultofmac.com/442033/is-apple-maps-still-the-laughing-stock-of-maps-apps-friday-night-fights/>
- <https://www.theguardian.com/technology/blog/2012/sep/24/apple-maps-problems-list>
- https://www.huffingtonpost.com/2012/09/20/apple-map-fails-ios-6-maps_n_1901599.html
- <https://blogs.cornell.edu/info2040/2012/11/07/apple-maps-a-failure-explained-by-network-effects/>
- <http://www.elmundo.es/economia/2016/10/14/5800d29ce2704e37228b45e0.html>
- <http://www.ticbeat.com/tecnologias/kit-para-retirar-samsung-galaxy-note-7-incluye-cajas-y-guantes-ignifugos/>
- <http://www.bbc.com/news/business-38714461>
- <https://www.cnet.com/news/5-biggest-takeaways-from-samsungs-note-7-battery-fire/>
- <https://www.techradar.com/news/samsung-galaxy-note-7-battery-fires-heres-why-they-exploded>