# Corran! Se me colgó el Programa!

## Catástrofes originadas por Errores en el Software

Carlos G. Lopez Pombo,
Dr. en ciencias de la computación
Departamento de Computación, FCEyN, UBA

clpombo@dc.uba.ar

# El Software…

- El software (programas) en la actualidad está presente en casi todas nuestras actividades.
  - En electrodomésticos
  - Equipos de música
  - Teléfonos
  - Automóviles

# ... puede fallar!

- Principalmente por:
  - errores de programación
  - errores de uso

# Problema vs. Programa

- Se usan programas para resolver problemas.
- Es necesario describirle al programador el problema a resolver de manera apropiada.

# Especificación

- La especificación es esa descripción.
- Puede ser informal o formal.

# Informal vs. Formal

- Informal:
  - Requiere menos entrenamiento del programador
  - Menos precisa y por lo tanto más proclive a aceptar errores
- Formal:
  - Requiere mayor formación de parte del programador
  - Se puede verificar formalmente la ausencia de ciertos errores

# Catástrofes por Errores de Programación

# Therac-25 (1985)

- Máquina para realizar tratamientos con radiación a pacientes con cáncer.
- Error al asumir que ciertos programas se ejecutarían siempre en un cierto orden.
- 6 casos documentados donde la máquina administró sobredosis de radiación. 4 fatales.

# Arianne 5 (1996)

- Cohete europeo para lanzamiento de satélites.
- Costo de desarrollo: U$S 7.000.000.000
- Costo del cohete y satélites que llevaba a bordo: U$S 500.000.000
- Error en conversión de un dato de 64 bits a 16 bits.

# Arianne 5 (30 segundos)

- A los 30 segundos del despegue se la veía así:

# Arianne 5

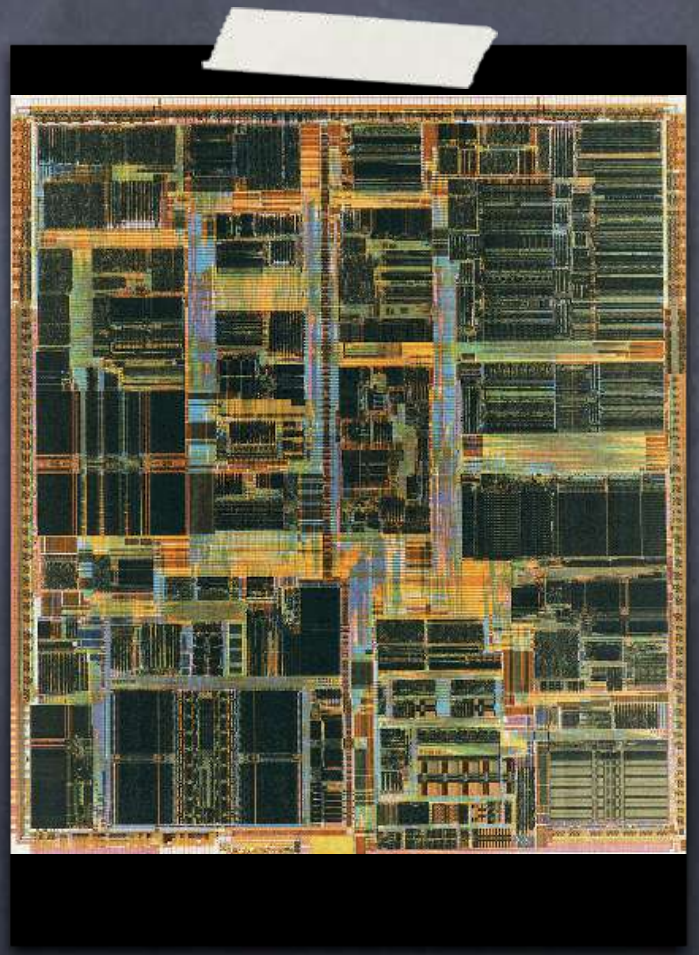
- 40 segundos

# Cohete Cryosat (2005)

- Mediría el cambio en las capas de hielo de la tierra.
- Costó Euro 135.000.000
- Cayó al mar por problemas en el software de control de vuelo.

# Error en el Procesador Pentium (1994)

- Al mejorar el algoritmo de división del chip (con respecto al del procesador 486), olvidaron setear unos valores necesarios en tablas internas del procesador.
- Costo del error:
  U$S 500.000.000



# Orbitador Climático de Marte (1998)

- Error al confundir medidas en unidades inglesas con americanas (libras versus kilos).
- Costo: U$S 300.000.000
- Se acercó demasiado a la atmósfera de Marte y se perdió contacto con el satélite.

# Corte de Luz en USA y Canadá (2003)

- Afectó a 50.000.000 de personas en 8 estados y Canadá.
- Se debió a un error de software en un sistema de alarmas de una central energética. Los operarios nunca recibieron alarmas que indicaran fallas.

# Bombardero F-22

- 12 se quedaron sin sistemas en el aire por fallas en el software.
- Llegaron a salvo siguiendo de forma visual a los aviones para recarga de combustible.

# Toyota Prius (2005)

- Error en el software hacía que el auto se detuviera sin motivo.
- El software de 160.000 autos debió ser modificado.

# Errores en el Uso del Software

# Contratos

- La forma en la que se debe utilizar un programa puede ser explicada formalmente mediante un contrato.
- El contrato describe el contexto en el cual se debe utilizar el programa (lo que el programa “requiere”) y describe lo que el programa “garantiza”.

# Catástrofes por Violaciones de Contratos

# El Sheffield en Malvinas

- El sistema de alarma del radar estaba programado para reconocer a los misiles Exocet como “amigos” por tenerlos los ingleses en su arsenal.

# Falla en defensa “Patriot”

- Un misil Scud iraquí no fue detectado por las defensas de Patriots.
- El contrato de los misiles los hacía eficaces para trackear objetos por debajo de la velocidad de los Scud, y para ser utilizados por hasta 14 horas seguidas antes de resetear el reloj. Los usaron por 3 días.
- 28 muertos, 98 heridos.

# Por qué hay errores en el software?

- Para eso tenemos que entender cómo es el proceso de desarrollo del software.
- Para procesos informales se “testea” que los programas resultantes funcionen de la forma esperada.

# Testeo de Programas

- Idealmente sería deseable chequear que un programa se comporta adecuadamente para todas las entradas posibles que se le pueden dar.
- Este conjunto de datos puede ser inmensamente grande, entonces se chequea el programa con unos pocos datos representativos.

# Contras del Testeo

- No hay criterios lo suficientemente precisos para garantizar que los programas no fallarán con los datos en los que no fueron testeados.

# Alternativa: Uso de técnicas formales de análisis de software

# Requerimientos de manera formal

- Los programas sirven para resolver problemas.
- Cómo hacemos para garantizar que el programador entendió bien el problema?
- Usamos herramientas para analizar formalmente lo que el programador entendió.

# Análisis Formal

- Herramientas para requerimientos
- También hay herramientas para analizar los programas.
- Problema: No son lo suficientemente poderosas (aún) para tratar grandes sistemas.

# Dónde aprender más del tema?

- Carrera de Licenciatura en Ciencias de la Computación, Facultad de Ciencias Exactas y Naturales, UBA.
- Visítenos! (Pabellón I, Ciudad Universitaria)

Muchas gracias!