nedeljainformatike v2.0

Velike softverske katastrofe

Organizatori Nedelje informatike

Matematička gimnazija
Nedelja informatike v2.0

30. septembar 2015.

OPERATOR ERROR

MOLTEN CORE WARNING

An opperator error exception has occured at FISSREAC0020093:09
FISSREAC0020077:14 FISSREAC0020023:17 FISSREAC0020088:22
neutron multiplication rate at spikevalue 99999999

* Press any key to vent radiological emissions into atomosphere.
* Consult reactor core manual for instructions on proper reactor core maintenance and repair.

Press any key to continue

Award Modular BIOS v6.00PG, An Energy Star Ally
Copyright (C) 1984-99, Award Software, Inc.

BIW1M/BIW2M BIOS V1.3

Main Processor : PENTIUM II 910MHz
Memory Testing : 131072K OK + 1024K Shared Memory

Award Plug and Play BIOS Extension v1.0A
Copyright (C) 1999, Award Software, Inc.

Trend ChipAwayVirus(R) On Guard Ver 1.64

Press **DEL** to enter SETUP, **ALT+F2** to enter AWDFLASH
09/21/2000-i810-W83627HF-6A69MPNAC-00


Microsoft
Microsoft
Windows95
Microsoft Internet Explorer


My Computer


Inbox

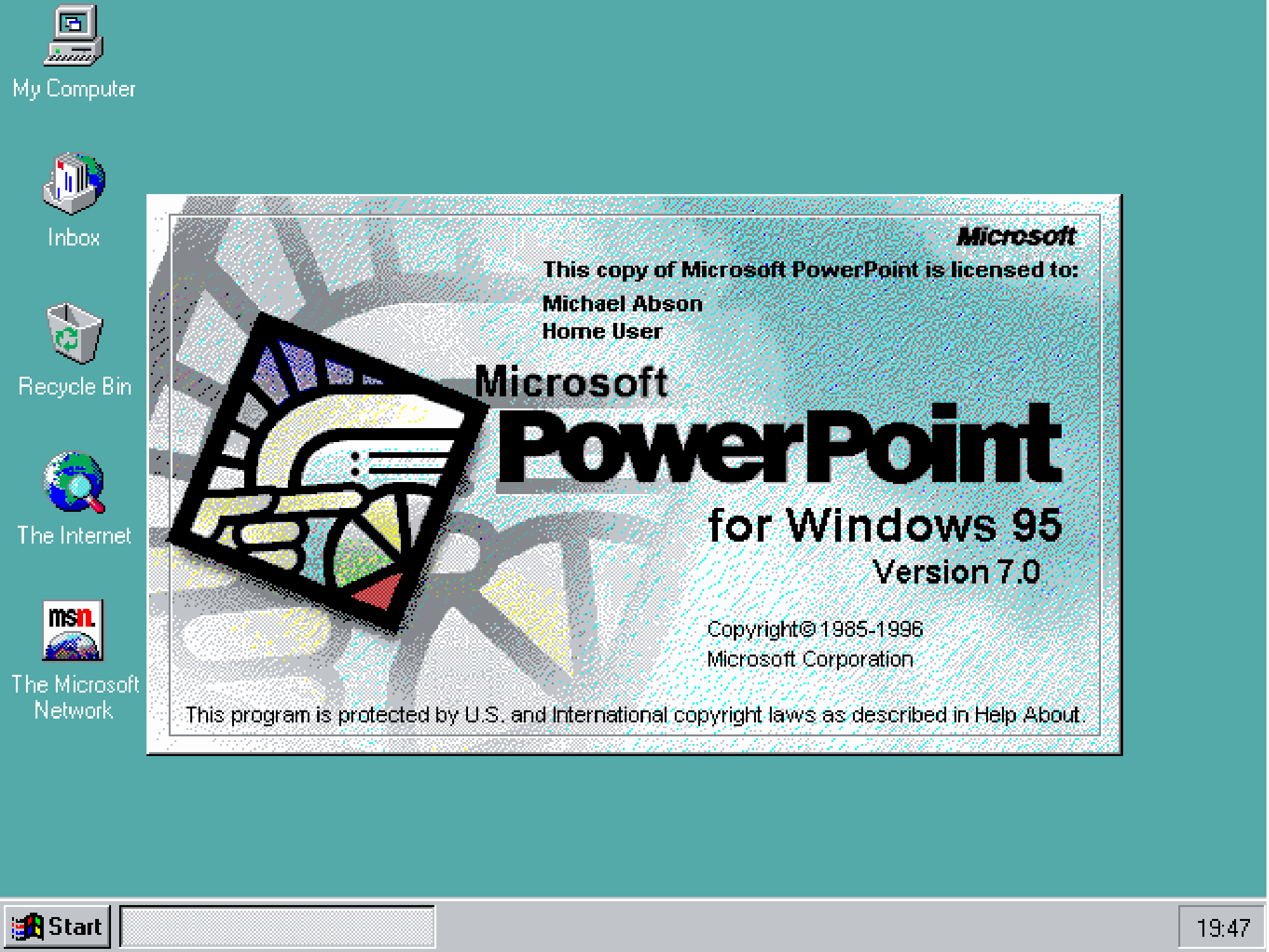

Recycle Bin

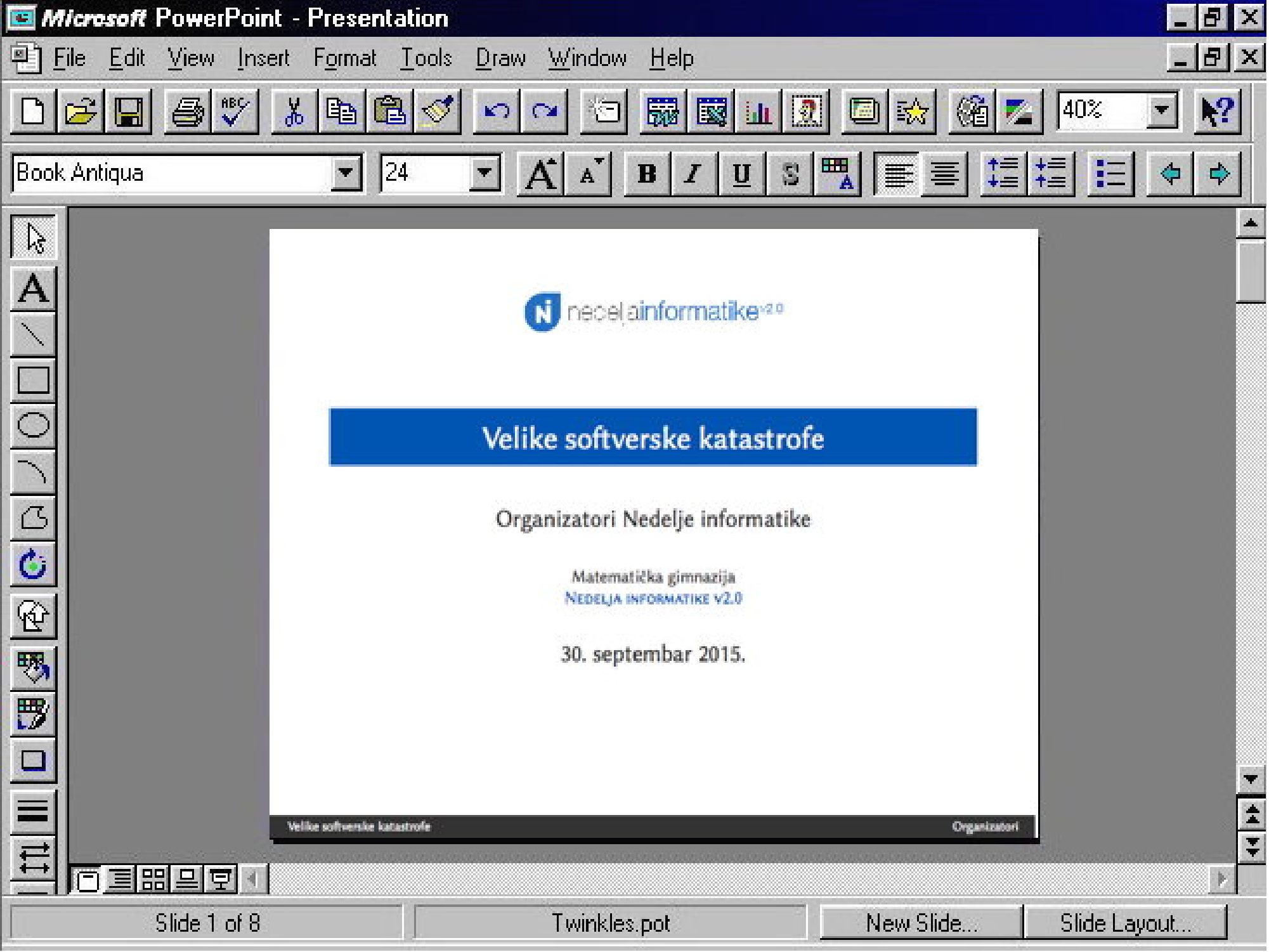

The Internet

The Microsoft Network

Start

19:47


My Computer
Inbox
Recycle Bin
The Internet
The Microsoft Network
Microsoft
This copy of Microsoft PowerPoint is licensed to:
Michael Abson
Home User
Microsoft
PowerPoint
for Windows 95
Version 7.0
Copyright© 1985-1996
Microsoft Corporation
This program is protected by U.S. and International copyright laws as described in Help About.
Start
19:47


Microsoft PowerPoint - Presentation
File Edit View Insert Format Tools Draw Window Help
40%
Book Antiqua
24
nedeljainformatike v2.0
Velike softverske katastrofe
Organizatori Nedelje informatike
Matematička gimnazija
NEDELJA INFORMATIKE V2.0
30. septembar 2015.
Velike softverske katastrofe
Organizatori
Slide 1 of 8
Twinkles.pot
New Slide...
Slide Layout...

O čemu ćemo pričati?

- U okviru prethodne Nedelje je bilo i predavanje koje se ticalo računarske bezbednost, u okviru kog smo se bavili OpenSSL *Heartbleed* bagom, odnosno veoma ozbiljnim sigurnosnim propustom.
- Posledice *Heartbleed* baga su bile drastične.
- U okviru ovog kratkog predavanja ćemo vam predstaviti još neke katastrofe izazvane veoma trivijalnim greškama.

Steam briše sve na disku?

- U januaru 2015. godine, korisnici *Steam* online servisa za igre koji su koristili Linux OS su počeli da prijavljuju da su im brisani svi fajlovi sa diska na kom je i operativni sistem.
- Brisanje se dešavalo isključivo ukoliko je menjan instalacioni direktorijum.
- Srećom, ovo je bilo ispravljeno izuzetno brzo.

Koji je bio uzrok ovoga?

- Brisanje se dešavalo isključivo ukoliko je menjan instalacioni direktorijum.
- Ključ je u jednom redu skripte `steam.sh`:

```
rm -rf "$STEAMROOT/"
```

- Instrukcija `rm -rf` briše sve što se nalazi u nekom direktorijumu i svim poddirektorijumima.
- `$STEAMROOT` predstavlja promenljivu koja ima vrednost direktorijuma u kom se nalazi instalacija. Međutim, ako ta lokacija ne postoji, onda se brišu svi nesistemski fajlovi na disku jer je tada to ekvivalentno sa:

```
rm -rf "/"*
```

Eksplozija prvog lansiranja *Ariane 5* rakete

- *Ariane 5* je aktuelni model raketa koje lansira Evropska svemirska agencija (ESA).
- Iako su pre prvog lansiranja vršene veoma detaljne simulacije, prvi pokušaj 1997. godine je trajao svega četrdesetak sekundi, nakon čega je raketa eksplodirala.
- Šteta je bila procenjena na 500 miliona dolara.
- Bilo je neophodno svega par dana da bi se otkrio uzrok problema.

Snimak eksplozije

Uzrok eksplozije

- Interni program je čuvao sve važne fizičke veličine u vidu 64-bitnih *floating point* realnih brojeva, što daje ogroman opseg.
- U toku leta, jedan potprogram je izvršio pretvaranje brzine u 16-bitni označeni ceo broj. 16-bitni celi brojevi ne mogu da pamte vrednosti veće od 32767.
- Kako je broj bio veći od 32767, ovo je aktiviralo jedan fleg. Podrazumevano podešavanje rakete je bilo takvo da se, ukoliko dođe do nepredviđenih grešaka, pokreće sistem za samouništenje.

Raketa *Patriot*

- *Patriot* je familija anti-balističkih raketa koje koristi vojska SAD.
- 25. februara 1991. godine, u toku Zalivskog rata, jedna ovakva raketa je promašila iračku raketu koju je trebalo da uništi. Iračka raketa je pogodila američku vojnu bazu, usled čega je stradalo 28 vojnika i ranjeno 98.

Opet *floating point* aritmetika

- Srž problema je bila u sistemu koji je vodio računa o vremenu. Periodično je interni tajmer uvećavao aktuelno vreme za 0.1.
- Međutim, 0.1 u dekadnom zapisu odgovara beskonačnom periodičnom binarnom zapisu:

$$0.000110011001100\ldots$$

- Kako se pamti samo konačan broj binarnih cifara, 0.1 ne može tačno da se predstavi u računaru. Zato pri sabiranju dolazi do gubitka preciznosti.
- Ne pomaže ni sistem kako se pamte realni brojevi:

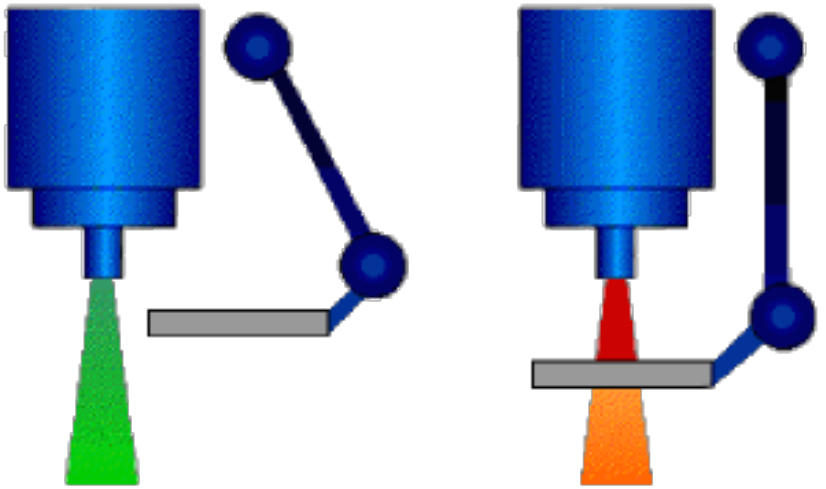
z | eeeeeeee | mmmmmmmmmmmmmmmmmmmmmmm

- Tako, na primer, izraz $0.1 + 0.1 + \ldots + 0.1$, gde ima ukupno $5 \cdot 10^6$ sabiraka, daje rezultat 499999.999955.

Therac-25

- Treća u seriji *Therac* mašina za radijacionu terapiju, razvijena od strane *AECL* (*Atomic Energy of Canada Limited*).
- Između 1985. i 1987. najmanje šest pacijenata primilo **potencijalno smrtonosne doze radijacije**!

Način funkcionisanja



- *Therac-25* ima dva predviđena načina emitovanja radijacije:
  - *Terapija elektronskim snopom*, koja emituje male doze snopa elektrona u kratkom vremenskom periodu;
  - *Terapija X-zracima*, koja emituje X-zrake koji se dobijaju sudarom elektrona visoke energije sa “metom”.

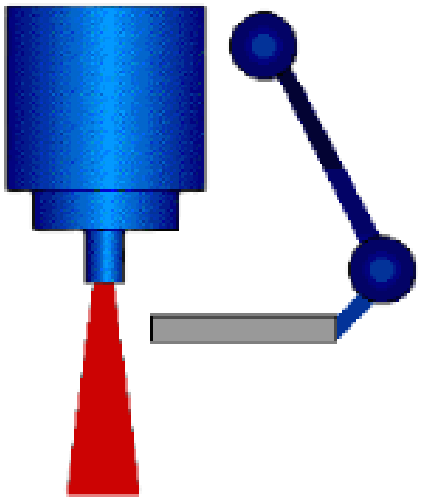
Anatomija katastrofe, deo prvi

- *Therac-25* je imao loše dokumentovan *konkurentni softver*:
  - Više funkcija mogu da se izvršavaju u isto vreme.
  - Ove funkcije mogu da pristupaju i manipulišu istim delovima memorije istovremeno, izazivajući neočekivano stanje sistema;
  - U ovom slučaju, “problematično stanje” je ispaljivanje jakih elektrona bez pravilno postavljene mete.
  - Doktori mogu izazvati prelaz u ovo stanje ukoliko dovoljno brzo unesu određen niz komandi.

Anatomija katastrofe, deo drugi

- Zaštita implementirana u *Therac-25* da bi se sprečilo neželjeno stanje je bila slaba:
  - Postojao je jedan 8-bitni registar (opseg $[0..255]$) koji kontroliše da li je aktivna zaštita (aktivna ukoliko je različit od nule).
  - Bezbednosni sistem je "aktivirao" ovaj registar tako što ga je **uvećavao za 1**, a ne postavljao na 1!!!
  - Ukoliko doktor izazove prelaz u neželjeno stanje u tačnom momentu kada ovaj registar doživi *overflow* nazad u 0, bezbednosni sistemi ovo neće korigovati!
  - Sistem bi, ipak, uspeo da prepozna da nešto nije u redu, i obavestio doktora o tome—međutim, ovo upozorenje je bilo jako neinformativno (prikaz reči *MALFUNCTION* na monitoru) i imalo je isti format kao i pri "lažnim uzbunama".
  - Doktor bi onda preinačio ovo upozorenje, time izazvavši...

Ishod



Emisiju 100x jače doze radijacije od namenjene!


YOU HAD ONE JOB

London Ambulance Service (LAS)

- 1992. godine, London pokušava da automatizuje dispečerski proces za hitnu pomoć.
- Najpoznatiji primer softverske katastrofe: London **ostao bez hitne pomoći** jedan ceo dan!

Kontekst

- Prethodni (ručni) dispečerski sistem zahtevao 3 minuta od prijema poziva do slanja vozila, i angažovao 200 ljudi.
- Automatizacija prethodno pokušana u 1980-tim godinama; sistem nije prošao testiranje.
- Veliki politički pritisci da se smanji cena i vreme izvođenja projekta (predstojeći parlamentarni izbori u UK).

Tender

- Jedna studija utvrdila da bi projekat koštao £1.9M i zahtevao bar 19 meseci.
- Tender održan u februaru 1991. sa zadatim rokom od 12 meseci. Većina kandidata na tenderu istakli da je ovaj rok nerealan.
- Pobednik na tenderu je bio konzorcijum od tri firme (*Systems Options Limited, Apricot, Datatrak*), koje nisu imale nikakvog iskustva sa softverom ovih proporcija, za £940K (£700K jeftinije od drugoplasiranog kandidata!)

Anatomija katastrofe, deo prvi

- Na kontrolnom sastanku posle 5 meseci, utvrđena velika količina nepravilnosti:
  - Potcenjene proporcije projekta;
  - Nedostatak ikakve metodologije i principa razvoja softvera;
  - Nepostojeća komunikacija sa budućim korisnicima sistema;
  - Oslanjanje na usmene “garancije” firmi, bez direktne kontrole.

- Rad na “dizajnu” “gotov” posle 6 meseci; posle 11 meseci, odlučeno da se u prvoj verziji automatizuje samo manji deo sistema.

Anatomija katastrofe, deo drugi

- Centralni sistem nikad nije bio u stabilnom stanju, i često je izazivao blokade.
- U takvim okolnostima, dodatno se implementira sistem automatskih radio poruka; nastaju problemi sa mestima slabog dometa i preopterećenjem radio kanala.
- Međutim, i pored velikih upozorenja od strane nezavisnih kontrolora, menadžment londonske hitne pomoći odlučuje da pokrene sistem, 8 meseci posle prvobitnog roka.
- Dispečerska kancelarija preuređena da koristi računare; svi prethodni alati su izbačeni. Nije bilo *back-up* sistema…

Ishod, deo prvi

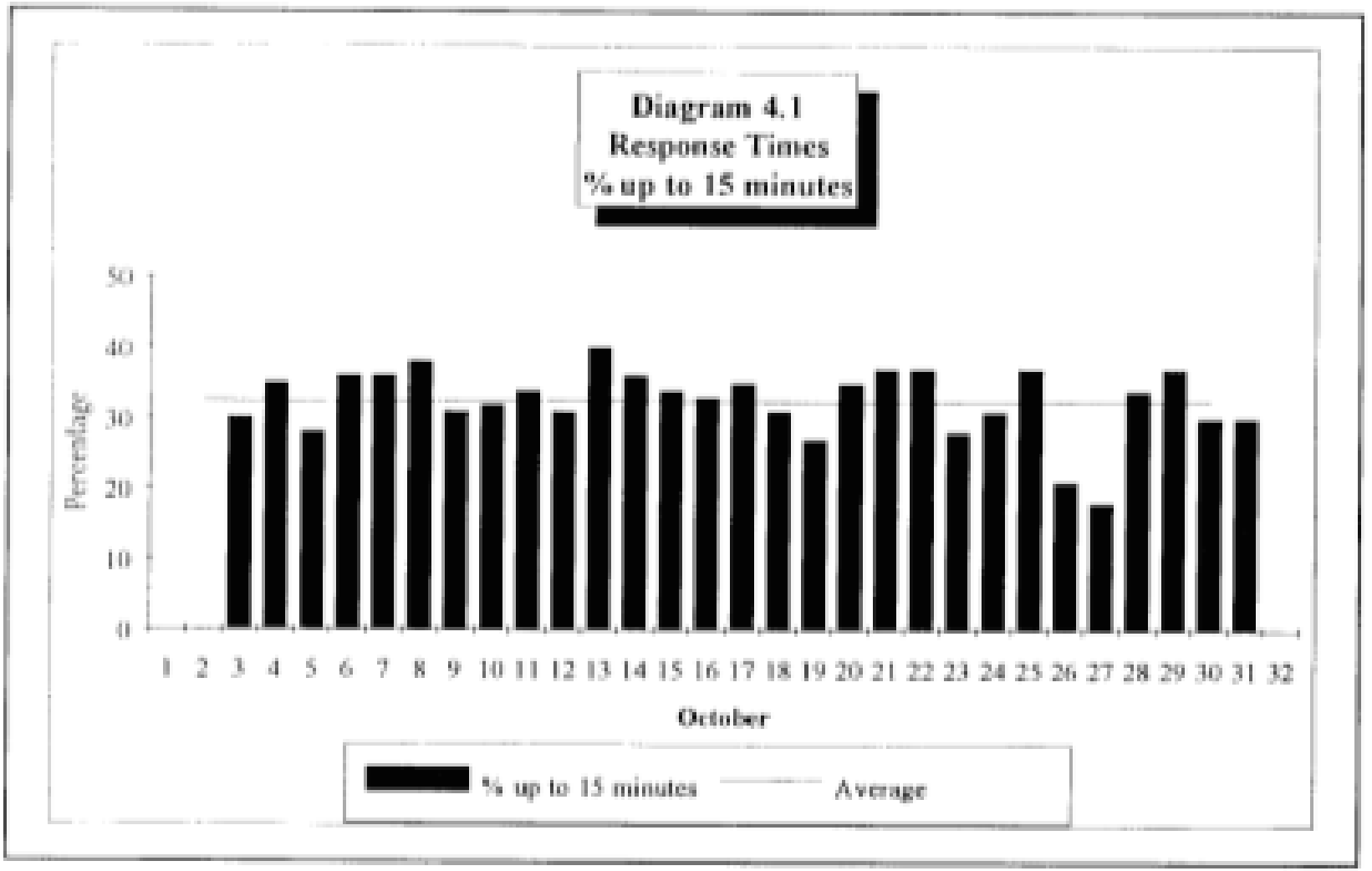
- Sistem sve češće gubi podatke o pozicijama vozila;
- Poruke upozorenja na računarima su bile toliko česte da nisu mogle da se razreše pre nego što bi nestale sa ekrana;
- Pacijenti su pozivali hitnu pomoć više puta, izazivajući dodatno zagušenje sistema;
- Zbog neiskustva u načinu korišćenja sistema, dešavalo se da se na jedan incident pošalje više vozila nego što treba, ili nijedno.

Ishod, deo drugi

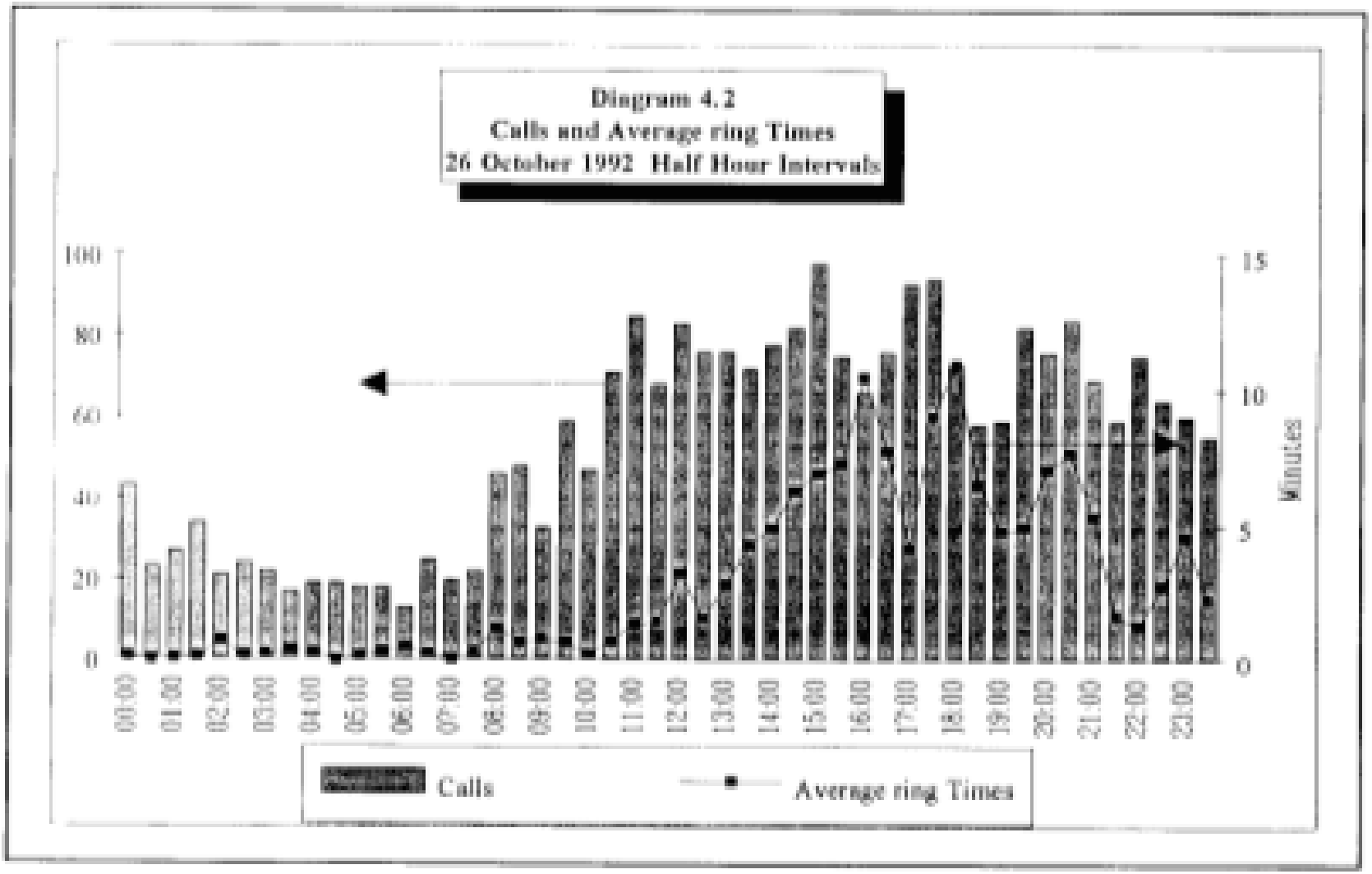
- Kompletan sistem, u nedostatku rezervnog sistema ubrzo postao potpuno **haotičan**:
  - Jedna ambulantna kola stigla na lice mesta i zatekla kako mrtvog pacijenta odvodi pogrebna služba;
  - Druga ambulantna kola stigla sa 11 sati zakašnjenja na lokaciju gde je prijavljen 'šlog', 5 sati nakon što je pacijent sam došao do bolnice.

- *Nemoguće je proceniti koliko smrtnih slučajeva je izazvano ovim incidentom.*

Ishod, deo treći



Ishod, deo četvrti



Nekoliko kratkih zaključaka...

- Razvoj softvera je jedinstven tip inženjerstva:
  - Potencijalno čuva nepredvidljive *složenosti*; testiranje ne može dokazati odsustvo bagova!
  - Za razliku od hardvera, koji se često pravi od gotovih komponenti, softver se uglavnom pravi "od nule";
  - Dugotrajni projekti mogu stalno "juriti pokretnu metu" (u vidu nejasnih, stalno promenljivih zahteva);
  - Menadžment koji veruje u *vile i vilenjake*; nerealni rokovi...
  - Dodatne komplikacije kod kritičnog softvera: *"A fail-safe system fails by failing to fail safe"*...
- Nadamo se da vas je današnje predavanje inspirisalo da, kao budući softverski inženjeri, uvek imate gorenavedeno u vidu.
- *Hvala na pažnji!*