Egészségügyi beágyazott rendszerek verifikációja

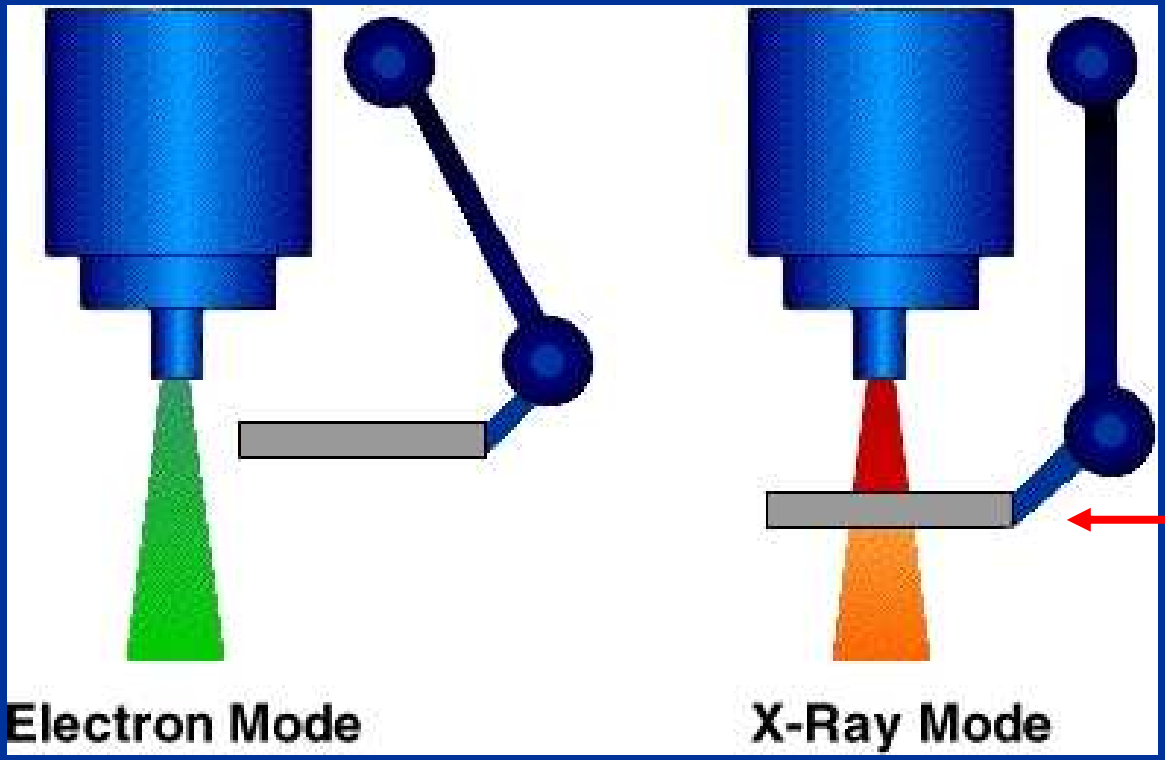
Kertész Zsolt

2011. december 8.

Miért verifikálunk és validálunk?

Esettanulmány 1
Therac25 sugárterápiás rendszer

Kétféle üzemmód a behatolási mélységnek megfelelően:



(5 MeV - 25 MeV)

(> 25 MeV)

„Incidensek” 1985-87

1. 61 éves nőbeteg, mellrák utókezelés után elveszítette a karját
2. 40 éves nőbeteg, A kezelő „No Dose” hibaüzenetet kapott, de a program lehetőséget adott arra, hogy „P” gombbal folytassa a kezelést.. A beteg 13 000 - 17 000 rad sugárzást kapott, 3 hónap után elhunyt. (összehasonlításképpen: az egész testet érő kb. 1000 rad már halálos dózis)
3. Egy beteg égési sérülés tüneteit mutatta, később meggyógyult.
4. Férfibeteg a besugárzás után heteken belül elvesztette mindkét lábát és bal karját. Öt hónappal később ő lett az első áldozat, aki közvetlenül és bizonyíthatóan kapcsolatba hozható a Therac-25 rendszerrel.
5. Egy férfibetegnél sugárkezelés arcon, a beteg súlyos idegrendszeri károsodást szenvedett, kómába esett, 3 hét múlva meghalt.
6. Egy újabb esetben a gép ismét hibaüzenettel leállt, de a technikus a „P” gombbal folytat(hat)ta a kezelést. Sugárbetegség tünetei után a beteg 3 hónap múlva elhunyt.

Mi történt?

- Az első eset után tagadás a gyártó részéről!
- A második eset után az Atomic Energy of Canada Limited (AECL) nem tudta rekonstruálni a hibát, de apróbb kiigazításokat csinált a hardveren.
- A 4. és 5. eset helyén sikerült először rekonstruálni a hibajelenséget és a "Hibás működés 54" hibaüzenetet:

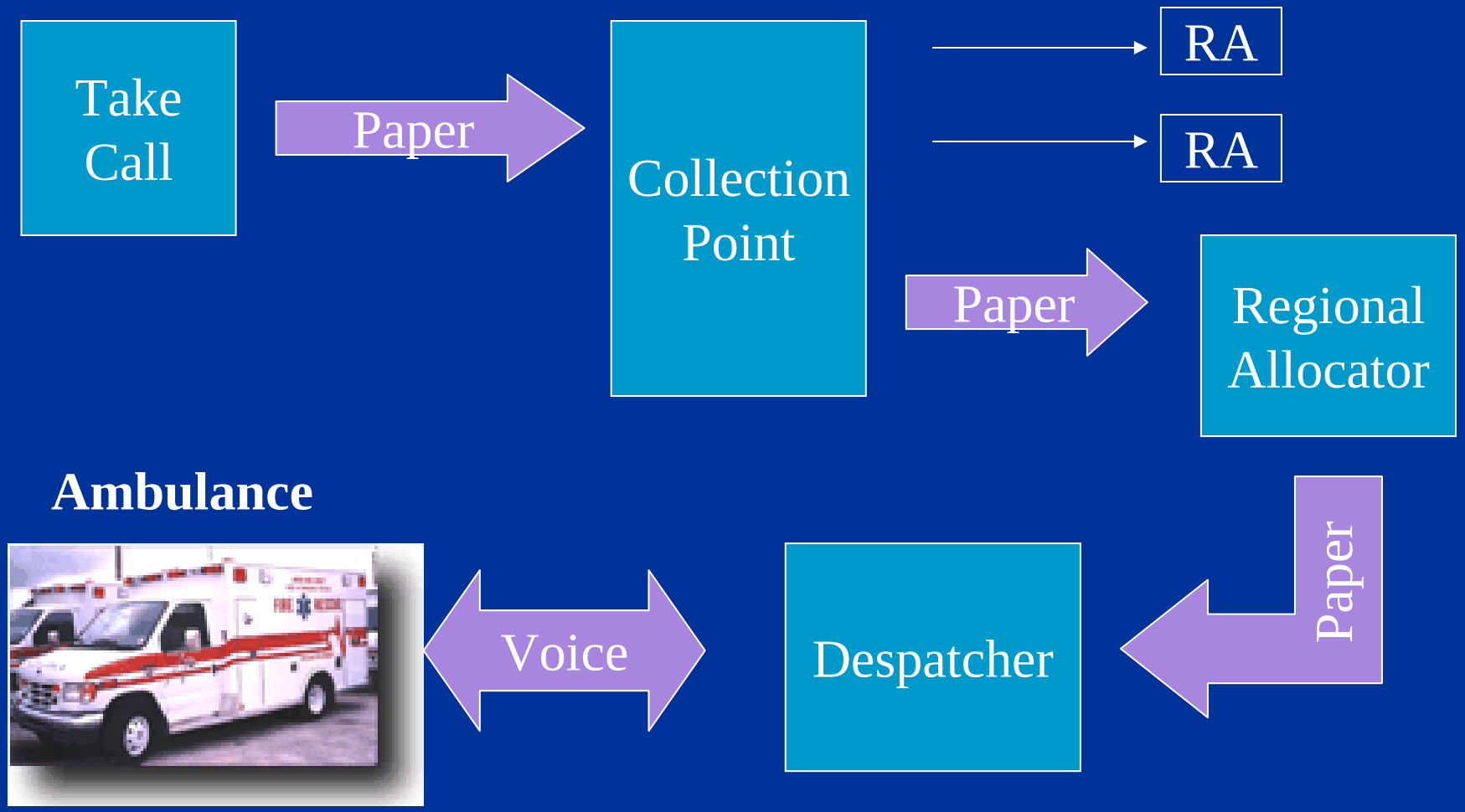
  Ha a technikus gyorsan váltott a terminálon a kurzorral a két üzemmód között, és 8 másodpercen belül nyomott Enter-t, akkor előjött a hiba!

Esettanulmány 2
London Ambulance Service

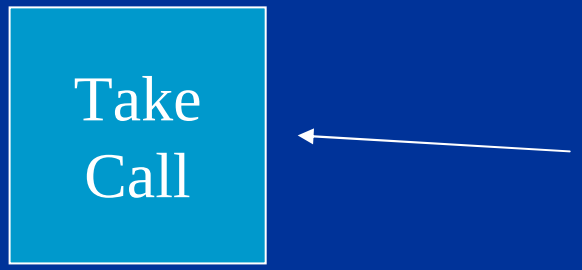
A legnagyobb mentőszolgálat a világon

- Az eset idején 6,8 millió lakos tartozott hozzá
- 5000 beteg egy nap
- Több mint 2000-2500 hívás naponta
- 2700 teljes munkaidős dolgozó

Az eligazító rendszer a ’80-as években



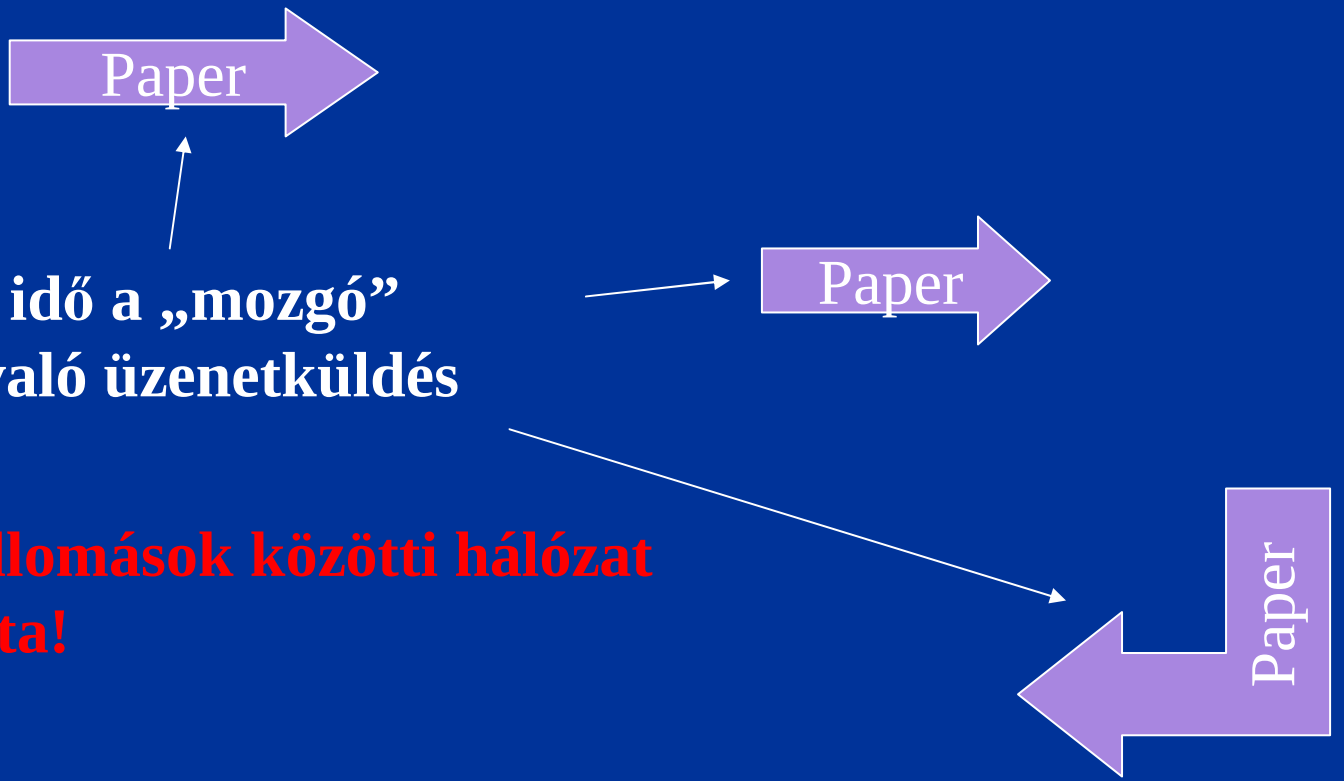
Computer Aided Despatch projekt célok 1.



A baleset helyét megtalálni nehéz volt és időigényes

A nyilvános telefonok helyét egy szoftver automatikusan találja meg!

Computer Aided Despatch projekt célok 2.

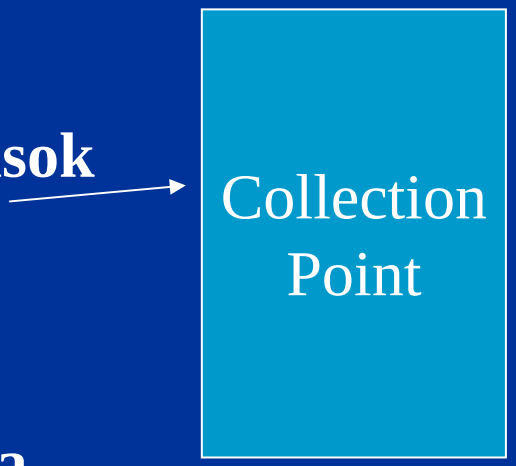


Rengeteg idő a „mozgó” papíron való üzenetküldés

Munkaállomások közötti hálózat használata!

Computer Aided Despatch projekt célok 3.

Az ismételt hívások felismerése a diszpécser memóriájától függött és hibára volt hajlamos



Mesterséges Intelligencia segítse az ismételt hívások felismerését!

Computer Aided Despatch projekt célok 4.

A mentőautó hozzárendelés kézzel, az elosztó ügyességére támaszkodott

Az egyszerűbb esetekben a mentőautó hozzárendelést végezze számítógép!

Computer Aided Despatch projekt célok 5.

Hang alapú kommunikáció

Legyen digitális kommunikáció az elosztó és a járművek között!

Mentőautó

Voice

Despatcher

Mi történt?

1992. október 26. A menedzsment úgy dönt, hogy be kell kapcsolni a teljes CAD rendszert.

Mindezt úgy, hogy

- A rendszer nem volt tesztelve
- Ismert volt néhány súlyosnak minősített hiba

A rendszer kezdetben megfelelően működött, kisebb hibák jelentkeztek, de ezeket a munkatársak kézben tudták tartani.

Mi történt?

A tartósan jelenlévő hibák miatt a rendszert visszaállították félautomata üzemmódba.

November 4-én a rendszer teljesen lefagyott.

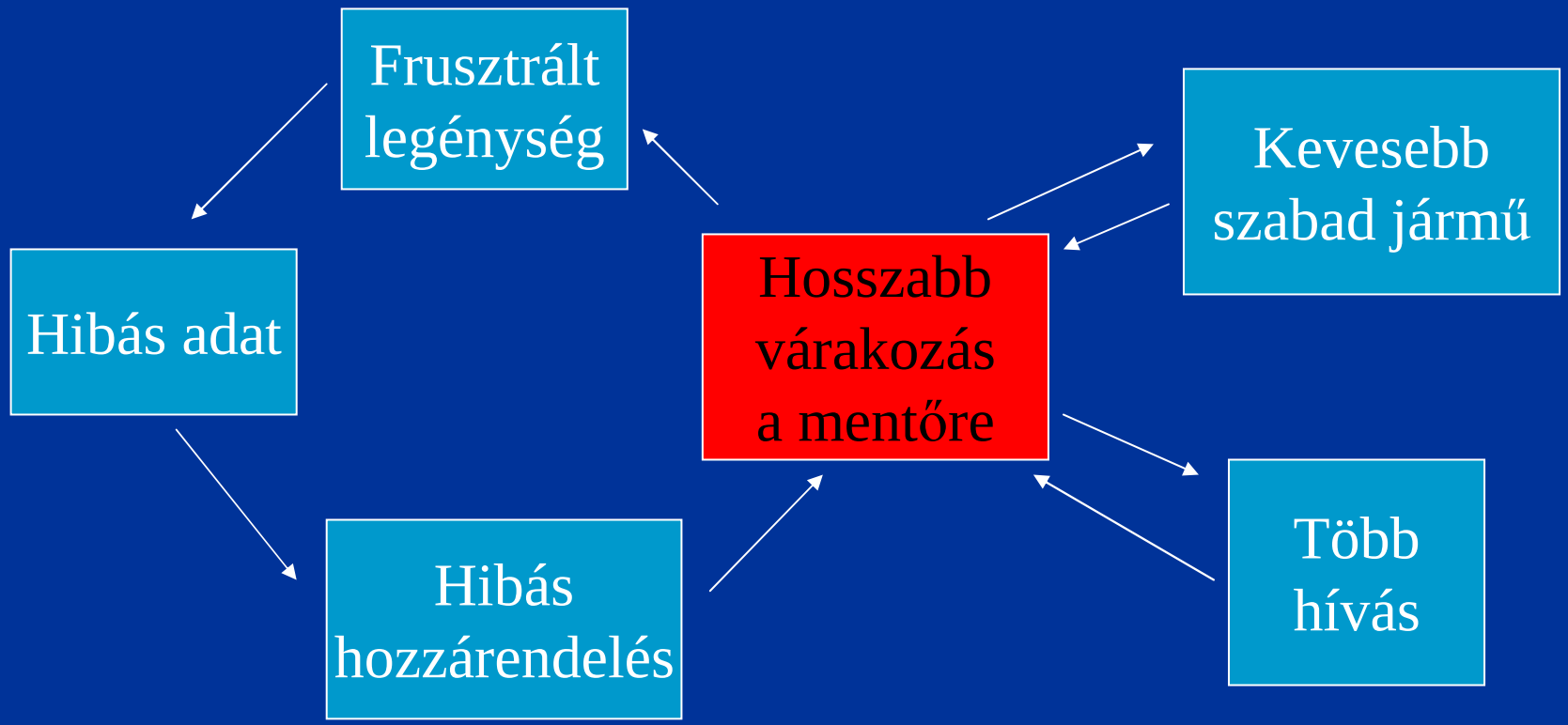
A hiba során a szerver „out of memory” üzenete jelent meg.

Ugyan létezett egy backup rendszer, de azt a teljes CAD rendszerhez tervezték.

A következmények

- A segélykérő hívásoknál több mint 10 perces várakozás, mire valaki felvette a telefont.
- Mivel a híváslista hosszúra nyúlt, több hívás elveszett.
- Ezután a mentők 80%-a több mint 15 perc múlva tudott reagálni.

A memóriaszivárgás oka: Visszacsatolási probléma



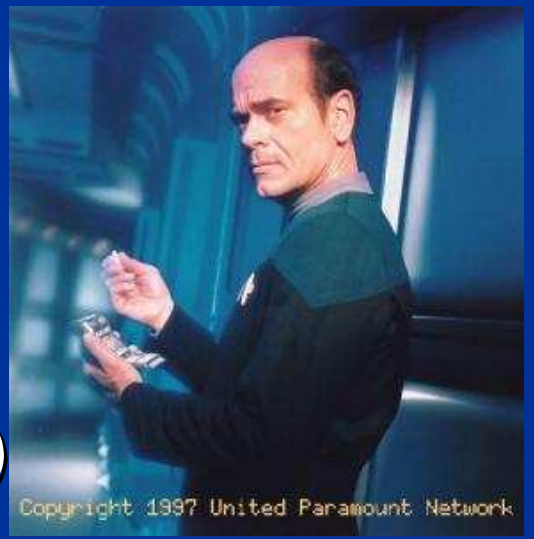
A jelen és a jövő

Az egészségügyben ’90-es évek óta hatalmas fejlődés.

Beágyazott rendszerek mindenütt:

- Infúziós pumpák
- Pacemakerek
- Hordozható EKG, ultrahang, PET szkennerek
- Ágy melletti monitorozó rendszerek
- Nagy orvosi képalkotó rendszerek
- Adatgyűjtők

(Qualcomm tricorder fejlesztő verseny)



Egy rendkívül biztonságkritikus példa
Robotic Surgery

Robotic Surgery
Fatal error...

Egy másik robotikai példa
Reharob projekt (videó)

Minőségbiztosítás

Vajon alkalmazhatóak-e a klasszikus minőségbiztosítási eljárások?

Nagy Gábor: Hibaokok és hatások analízise diszkrét gyártás területén előadás (december 7.)

- 6σ

„Az a vállalat, amely képes hat szigma minőséggel gyártani, az a legyártott 10 000 000 db terméke közül legfeljebb 34 db nem megfelelő terméket állít elő, tehát a gyártása 99,99966%-ban hibamentes terméket eredményez.”

GE: Naponta több ezer röntgenvizsgálat a világban

Létfontosságú orvosi berendezéseknél inkább NE alkalmazzuk a 6σ eljárást!

Szabályozások

- FDA (Food and Drug Administration) (Egyesült Államok)
- MHW (Ministry of Health and Welfare) jóváhagyás (Japán)
- ISO 9000
  - ISO 9001: Tervezés és gyártás
  - ISO 9002: Kizárólag gyártás
  - ISO 9003: Ellenőrzés és vizsgálat
- CE (Conformité Européenne = európai megfelelőség) jelölés (Európa)
  - kiállító szervezetek:

    Technischer **Ü**berwachungs-**V**erein

    British **S**tandards **I**nstitution

    Société **G**énérale de **S**urveillance

Szabályozások

Magyarországon a CE jelölés megszerzésének folyamata:
http://ce-jeloles.hu/

Termékkategória	Vonatkozó irányelv száma	Vonatkozó magyar jogszabály
Orvostechnikai eszközök	93/42/EGK	4/2009. (III. 17.) EüM r.
Aktív beültethető orvostechnikai eszközök	90/385/EGK	4/2009. (III. 17.) EüM r.
In vitro diagnosztikai orvostechnikai eszközök	98/79/EK	8/2003. (III. 13.) ESZCSM r.

Vizsgálati módszerek

Nagy megbízhatóságú orvosi rendszer példák:

- Napjaink Pacemaker-ei akár 80 000 kódsor
- Infúzió pumpákban akár 170 000 kódsor

Ezeknél a rendszereknél a verifikáció folyamata történelmileg:

1. Kód áttekintés egy másik szakértő által
2. Statikus analízis (mintaillesztéssel, futási idejű hibák nélkül)

   hibás pozitív versus hibás negatív detektálás
3. Dinamikus tesztek (valamennyi működési feltétel vizsgálata lehetetlen)

FDA által erre a célra alapított szervezetek:

- Center for Devices and Radiological Health (CDRH)
- Office of Science and Engineering Laboratories (OSEL)

CDRH és OSEL vizsgálatok alapján a leggyakoribb hibák

- Nem inicializált változók
- Tartományon kívüli tömb címzés
- Null pointer
- Hibás számítások, például nullával osztás
- Elosztott adatokhoz konkurens hozzáférés
- Típuskonverziós hibák
- Memóriaszivárgás
- Bizonyos feltételeknél nem leálló ciklusok

Formális módszerek

- Absztrakt interpretáció a hibás negatív találat csökkentésére

-4586 - 34,985 - 2389 = ?

Számolás nélkül is tudjuk, hogy csak negatív lehet!

Formális módszerek

Absztrakt interpretáció példa

- Futási időben jelentkező tulajdonságokra lehet következtetni a forráskód alapján:

Például:

$$X = X / (X - Y)$$

Futásidejű hibák:

1. Változók nincsenek inicializálva
2. Túl vagy alulcsordulás az osztás műveleténél
3. Túl vagy alulcsordulás a kivonás műveleténél
4. Nullával való osztás (ha X = Y)
5. Túl vagy alulcsordulás az X-hez rendelt érték függvényében

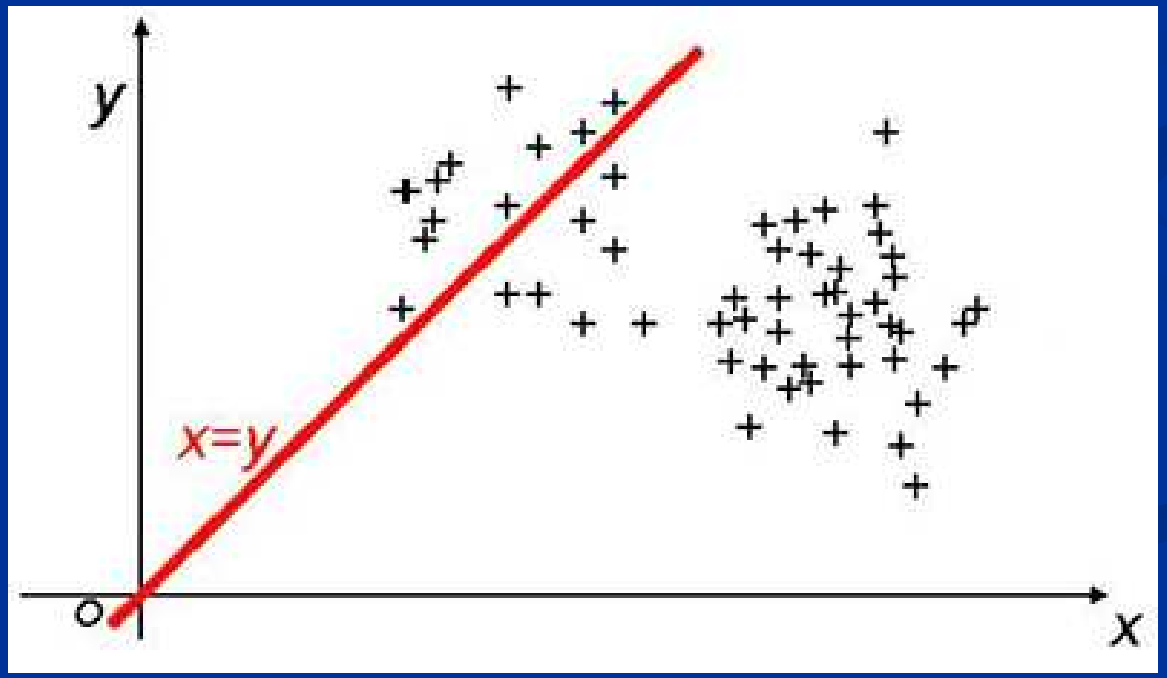
Formális módszerek

Absztrakt interpretáció példa

- Futási időben jelentkező tulajdonságokra lehet következtetni a forráskód alapján:

Például:

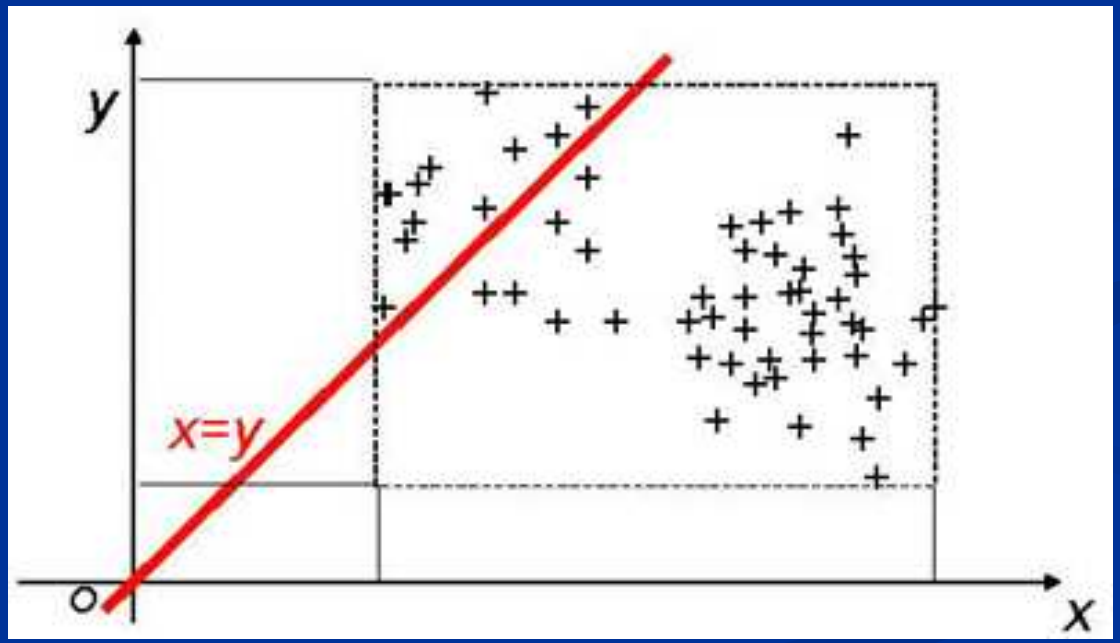
$$X = X / (X - Y)$$



Formális módszerek

- Típus analízis

X és Y értékkészletének vizsgálata hogyan függ össze a fenti feltétellel!

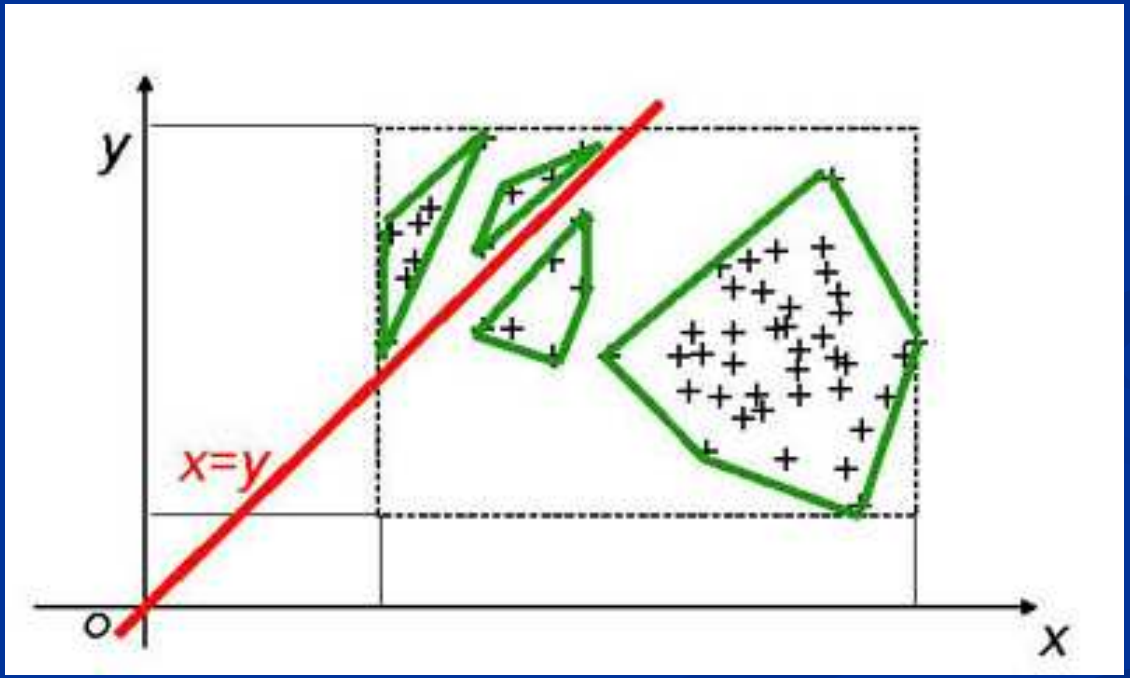


Túl pesszimista, nem reális X és Y értékeket is vizsgál!

Formális módszerek

- Az értékkészlet pontosabb reprezentációja

Számos programozási kontrukció megváltoztatja X és Y értékkészletét (aritmetika, ciklusok, if-then ágak, stb)



Az adatok poliéder tartományokba csoportosíthatóak!

FDA előírásoknak megfelelő Medical Device Software Development

- **Static code analysis** - coding standards, data flow, metrics.
- **Dynamic analysis** - unit/component testing, integration testing, functional testing, memory error detection, continuous regression testing.
- **Coverage analysis** - Multiple coverage metrics
- **Peer review** (and document review) process automation

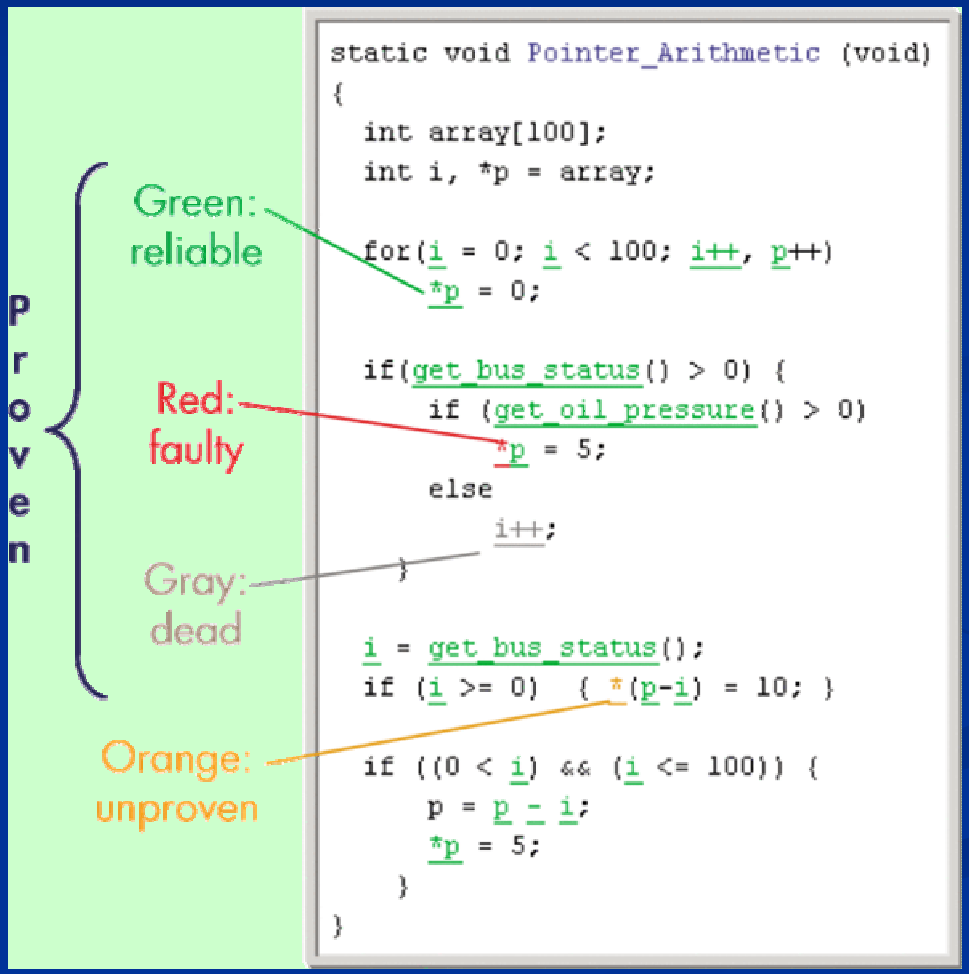
Részletesen: Galambos Róbert előadása: A Parasoft C++test bemutatása

PolySpace

- Nehezen detektálható futási idejű hibák felderítése C/C++ és Ada nyelven írt kódoknál
- MISRA C®, MISRA-C++ or JSF++ (Joint Strike Fighter Air Vehicle C++) standardok ellenőrzése
- Különböző metrikák a kód komplexitásának mérésére
- Szoftverkövetés minőségi mérőszámok generálása
- Formális módszerek a futásidejű hibák elkerülésére (absztrakt interpretáció) a kód futtatása nélkül
- Artifact-ok készítése DO-178B, IEC 61508, és ISO 26262 minősítéshez
- MathWorks Embedded Coder™ (Simulink®), dSPACE® TargetLink®, vagy IBM® Rational® Rhapsody® támogatás

PolySpace

Színkódok az egyes elemek státuszához



PolySpace példa 1.

where_are_errors.c

```
1    int where_are_errors(int input)
2    {
3    int x, y, k;
4
5    k = input / 100;
6    x = 2;
7    y = k + 5;
8    while (x < 10)
9            {
10           x++;
11           y = y + 3;
12           }
13
14   if ((3*k + 100) > 43)
15           {
16           y++;
17           x = x / (x - y);
18           }
19
20   return x;
21   }
```

A PolySpace meghatározza, hogy amikor x = 10, akkor mindig y > 10

Forrás: Paul Jones, Raoul Jetley, and Jay Abraham
A Formal Methods-based verification approach to medical device software analysis

PolySpace példa 2.

Recursive.c

```
1    void comp (int *d)
2    {
3
4    float advance;
5    *d = *d + 1;
6    advance = 1.0/(float)(*d);
7
8    if (*d < 50)
9    comp (d);
10
11   }
12
13   void bug_in_recursive (void)
14   {
15   int x;
16
17   x = 10;
18   comp ( &x );
19
20   x = -4;
21   comp ( &x );
22   }
```

x értékétől függ, hogy a rekurzív comp függvény hibás lesz-e

Forrás: Paul Jones, Raoul Jetley, and Jay Abraham
A Formal Methods-based verification approach to medical device software analysis

Legfontosabb következmény

A minősítési eljárások során a formális módszereken alapuló bizonyítást az FDA elvárja a biztonságkritikus ipari és egészségügyi beágyazott rendszerekhez!

Köszönöm a figyelmet!
Kérdések?