Tosiaikajärjestelmät – Luento 6: Luotettavuus ja turvallisuus

Tiina Niklander

Burns & Wellings luku 5 tai
Krishna & Shin luvut 7 ja 8

Luotettavuus ja vikasietoisuus

- Yleistä:
  - Vika, virhe, häiriö
  - Vikatyyppejä
  - Mitä voidaan tehdä: estää, havaita, sietää, korjata
  - Vian havaitseminen
  - Vian sieto
- Monentaminen ja toisintaminen
- Turvallisuus ja luotettavuus
- Riippuvuus (dependability)
- Esimerkkejä

Vika, virhe, häiriö

- Virhe (error, bug, mistake)
  - Ohjelmiston poikkeaminen määrittelystä
- Vika (fault)
  - Virheellisen ohjelmakohdan suoritus
- Häiriö (failure)
  - Viasta aiheutuva ulkoisesti havaittava poikkeama määrittelystä
  - Kaikki virheet ja viat eivät välttämättä johda häiriöön

Mistä häiriöt syntyvät?

- Määrittely- tai suunnitteluvirheet
  - Puutteellinen määritelmä
  - *Suunnitelmassa (tai toteutuksessa) virheitä*
- Komponenttiviat
  - Vanheneminen, huono valmistuserä, tms.
- Ympäristön vaikutus
  - Elektromagneettinen häiriö
  - Liikaa lämpöä, kiihtyvyyttä, tärinää, yms.

Vikatyyppejä

- Pysyvä vika (permanent)
  - Poistuu järjestelmästä vasta korjauksen jälkeen
- Tilapäinen vika (transient)
  - Ilmenee hetken ja häviää
  - Syynä esimerkiksi elektromagneettinen säteily
  - Muita nimiä: soft error
- Toistuva vika (intermittent)
  - Ilmaantuu satunnaisesti aika-ajoin
  - Irtonainen johto, lämpöherkkä komponentti

Häiriökäyttäytyminen (failure mode)

- Fail-silent
  - Aina oikea tulos, tai ei vastausta lainkaan
  - Helppo malli teoreettisissa tarkasteluissa
- Fail-consistent, value-error
  - Laskennan tulos voi olla väärä, mutta sama tieto kaikille kommunikointiin osallistuville
- Byzantine, arbitrary, malicious
  - Laskennan tulos voi olla erilainen eri vastaanottajille.

Mitä voidaan tehdä?

- Pysyvä vika
  - Varataan laitteistoon ylimääräistä kapasiteettia, jotta toiminta voi jatkua
  - Korjataan myöhemmin
- Tilapäinen (tai satunnainen) vika
  - Toivutaan viasta, laitteisto on yleensä edelleen ehjä
  - Havaitaan ja tehdään vain laskenta uudelleen
- Tyypillisesti tutkimuksissa on valittu vikaantumisen malliksi joko
  - Pysyvä vika + fail silent tai
  - Tilapäinen vika + fail consistent tai byzantine

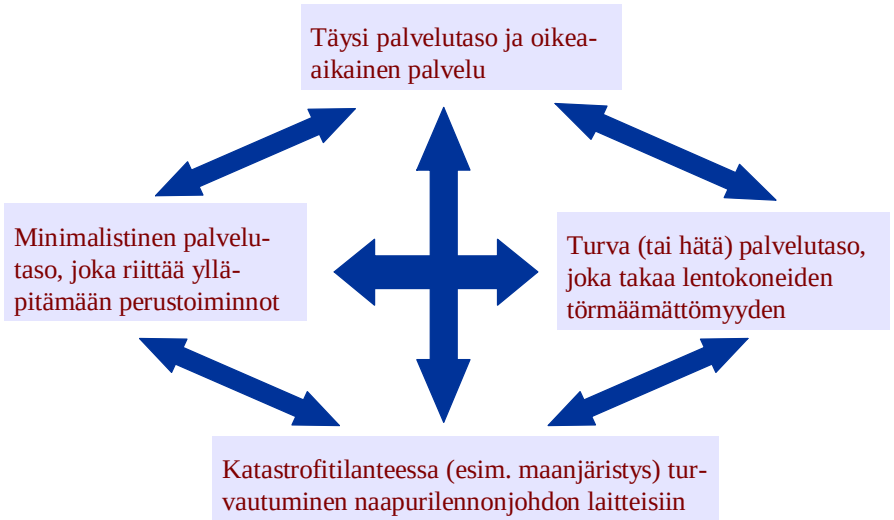
Vian havaitseminen

- Vian havaitseminen edellyttää sekä tietoa nykyisestä tilasta että suunnitellusta tilasta
- Suunnitellun tilan tieto voidaan saada
  - Joko etukäteen pääteltynä (a priori)
  - Tai vertailutietona rinnakkaisista järjestelmistä
- Prosessorin häiriökäyttäytyminen voidaan yrittää havaita
  - Laittomasta toiminnasta: esim. suorituskäskyn nouto data-alueelta, virheellinen opkoodi, liian pitkään kestänyt suoritus

Vikaantumisen sieto

- Perustuu aina monentamiseen (redundancy)
- Useita vaihtoehtoisia palvelutasoja
  - Täysi vikasietoisuus: järjestelmä tarjoaa vioista riippumatta saman pysyvän palvelutason
  - Alentunut palvelutaso: järjestelmä tarjoaa edelleen palvelua, mutta sen suorituskyky on alentunut vikojen seurauksena
  - Turvallinen pysähtyminen (fail safe): Järjestelmä ei enää tarjoa palvelua, mutta varmistaa, että ‘pysähtyminen’ tapahtuu turvallisesti

Esimerkki: lennonvalvonta



Vikasietoisuus

- Järjestelmä on *k-vikasietoinen*, jos se toimii määritelmänsä mukaan vielä, kun *korkeintaan k komponenttia* on vikaantunut
- Hierarkkinen rakenne (järjestelmä voi olla komponentti ylemmän tason järjestelmälle)

Luotettavuus ja vikasietoisuus

- Yleistä
- **Monentaminen ja toisintaminen**
  - k-vikasietoisuus, n-versiointi
  - Laitteisto, ohjelmisto vai aika
  - Äänestys, konsensus, toipuminen
- Turvallisuus ja luotettavuus
- Riippuvuus (dependability)
- Esimerkkejä

Monentaminen on vanha keksintö

- *”The most certain and effectual check upon errors which arise in the process of computation is to cause the same computations to be made by separate and independent computers; and this check is rendered still more decisive if their computations are carried out by different methods.”* D. Lardner, Edinburg Review, 1825

Laitteisto

- Lisätään järjestelmään laitteistoa, jota ei tarvittaisi, jos mikään ei koskaan vikaantuisi
- Lyhytaikainen käyttötapa
  - Häiriön havaitsemiseen
  - Kaikki komponentit suorittavat samat operaatiot ja tuloksesta äänestetään
- Pitkäaikainen käyttötapa
  - Korvataan vikaantunut järjestelmän osa vastaavalla toimivalla varaosalla

Laitteisto

- Monentaminen on kallista, koska tarvitaan ns. 'turhia' osia valmiiksi järjestelmään
- Avaruusluotaimiin on pakko asentaa valmiiksi riittävä määrä varaosia, joita voidaan ottaa käyttöön tarvittaessa
- Huollettavammassa järjestelmässä (esim. auto, lentokone) voidaan käyttää äänestysmenettelyä ja vikaantuneet osat korjataan, kun niitä havaitaan

Kuinka monta toisintoa tarvitaan?

- Kun halutaan sietää k eri virhettä, tarvitaan
  - k+1 toisintoa, jos virhe pysäyttää toiminnan (fail-silent tai fail-stop)
  - 2k+1 toisintoa, jos johdonmukainen vikaantuminen eli virhe on sama kaikille (fail-consistent)
  - 3k+1 toisintoa, jos Bysanttilainen vikaantuminen eli 'pahantahtoinen' toiminnallisuus, vikaantunut komponentti tuottaa satunnaisia tuloksia (malicious, arbitrary)

Äänestys ja konsensus

- Vähintään kolme samanlaista voi äänestää ja käyttää enemmistöpäätöstä
- Kaksi voi havaita vian, mutta ei päättää oikeaa tulosta (fail silent mahdollinen)
- Miten lähellä tulosten pitää olla toisiaan? (Pyöristyserot eri laitteistoilla)
  Onko 1.400978 sama kuin 1.400984 ?
- Likimääräinen samuus sallittava, jos syötteillekin sallitaan pientä vaihtelua

Äänestystapoja

- Enemmistöpäätös (majority vote)
  - $x,y \in P_i$, joss $d(x,y) \leq \varepsilon$, valitaan $\max(P_i)$
  - Valitaan yksi arvoista $x \in P_i$
- k-enemmistöinen päätös (k-plurality vote)
  - Valitaan joku P_i, jossa vähintään k jäsentä
- Keskivertopäätös (median vote)
  - Valitaan vastauksista arvoltaan keskimmäinen

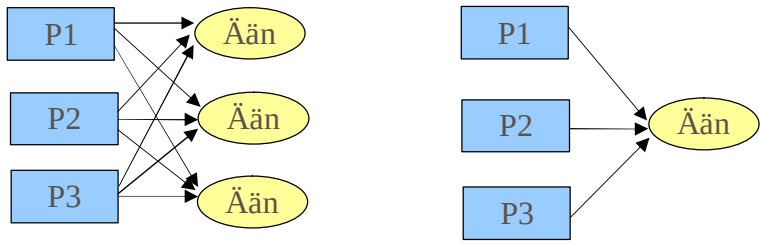
Arkkitehtuurivaihtoehdot

- Kaksi synkronoitua prosessoria (lock-step dual processor)
  - Verrataan tuloksia, jos eroja -> fail-silent
  - Havaitsee kaikki viat
- Kaksi erillistä prosessoria (loosely-synchronized dual processor)
  - Havaitsee virheellinen tuloksen,
  - Tarvitsee erillisen vian etsinnän
- Kolme prosessoria (TMR - triple modular redundant)
  - Äänestyksen jälkeen havaitaan virhe ja tiedetään viallinen komponentti
- Uusi idea (2003): Dual Lock-Step
  - Oikeasti siis neljä prosessoria

n-modulaarinen monennus (NMR)

- Tyypillinen laitteistotason ratkaisu
- N prosessoria ja äänestetään tulos
- N yleensä pariton
- Tarvitaan 2m+1 prosessoria, jotta selvitään m:sta vikaantumisesta
- TMR - triple modular redundant

NMR kaavakuvia



PLC (ohjelmoitava logiikkakontrolleri)

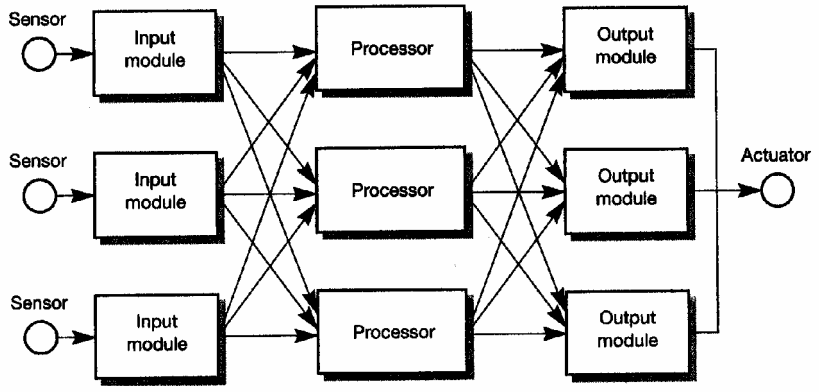


Figure 6.21 A typical arrangement for a high-reliability PLC.

Storey: Safety-critical computer systems

Dual Lock Step

- Tähän tuo kaavakuva artikkelista
- (tai koko asia pois)

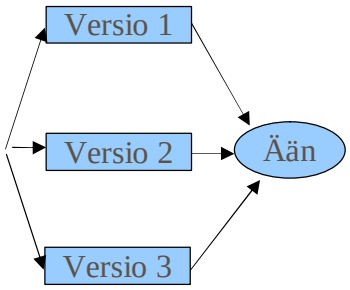
Baleani et.al: Fault-Tolerant Platforms for Automotive Safety-Critical Applications. CASES´03, s. 170-177, ACM.

Ohjelmisto

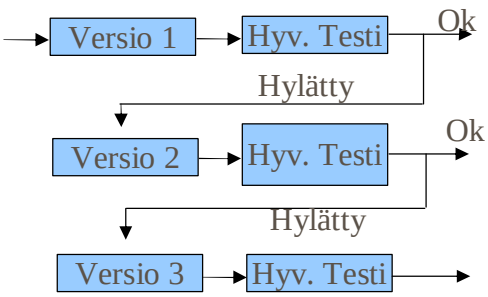
- Laitteistotason monentaminen mallina
- N-versiointi
  - Tehdään N eri toteutusta samasta ohjelmasta
  - Tavoitteena eri virheet eri ohjelmissa
- Toipumislohkot (recovery block)
  - Toistettavissa oleva ohjelman suorituksen osa
  - Toistossa voidaan käyttää samaa ohjelmakomponenttia tai jotain toista versiota
  - Toistetaan vain, jos ens. suoritus virheellinen

Ohjelmiston monentaminen

N-versiointi



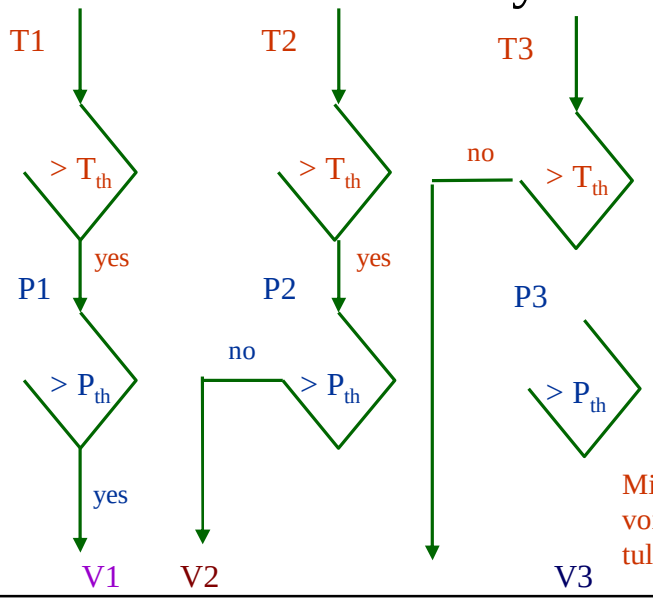
Toipumislohkot



N-versiointi

- Kallista
- Riippumattomat työryhmät, ei yhteistoimintaa
- Vaikuttavia tekijöitä:
  - Vaatimusten spesifiointi (entä jos tässä virhe)
  - Ohjelmointikieli ja Algoritmi
  - Työkalut, koulutus ja ohjelmointitaito

Vertailun tai äänestyksen ongelma



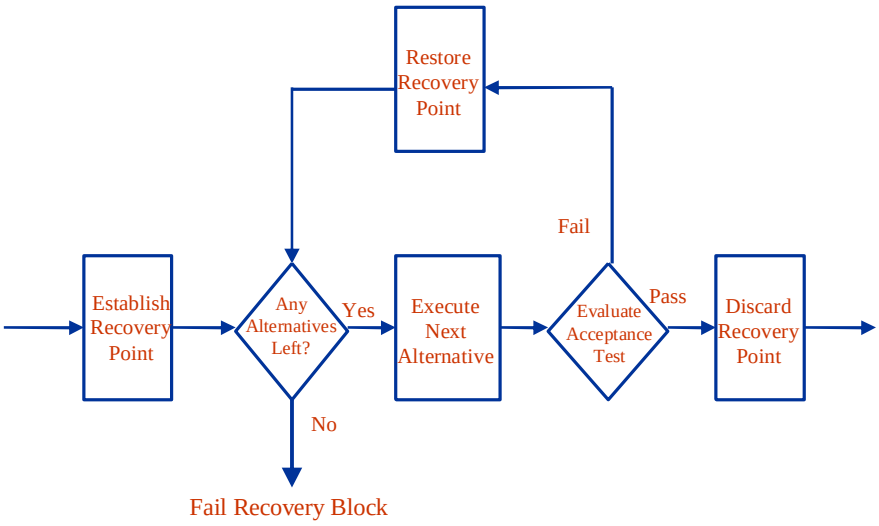
Erilainen, mutta oikea tulos kaikissa versioissa

Millaisella vertailulla voidaan päätellä tuloksen samuus?

Toipumislohko

- Lisätään ohjelmaan toipumispisteitä (recovery point)
  - Tällainen voi olla esim. checkpoint, jossa ohjelman koko tila tallennetaan
- Vikatilanteessa suoritus voidaan peruuttaa edelliseen pisteeseen ja jatkaa siitä uudelleen
- Toipumispisteestä seuraavaan voidaan käyttää jotain vaihtoehtoista menetelmää tai versiota

Toipumislohko

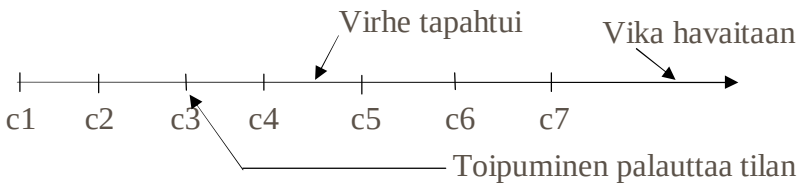


Vertailu: N-versio / toipumislohko

	N-versio	**Toipumislohko**
Toimintamalli	Staattinen	Dynaaminen
Kehityskustannus	Versiot + ohjaus	Moduulit + hyv-testi
Suorituskustannus	N resurssia	Toipumispisteet
Virheen hav.	Äänestys, joustava	Hyväksymistesti
Atomisuus	Kommunikointi ohjaimen kautta	Taaksepäin toipuva, vähäinen yleiskuorma

Aika

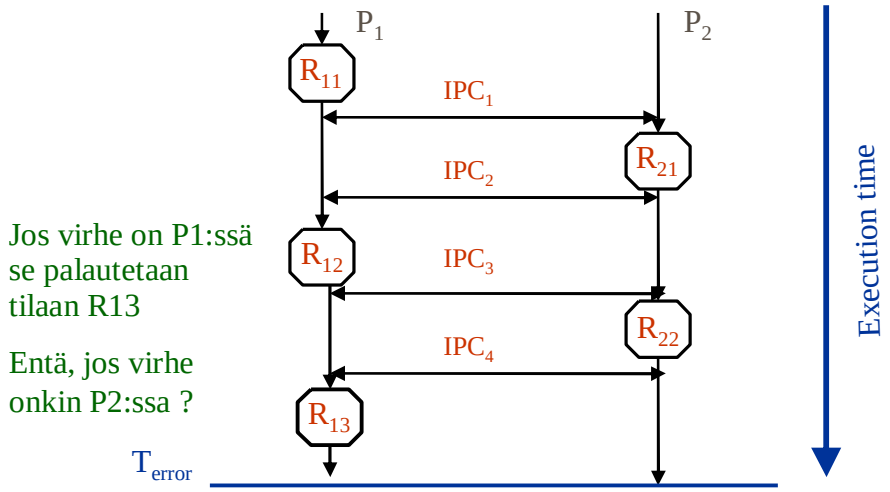
- Ajan toisintaminen -> laskennan toistaminen myöhemmin
- Edellytys: aikarajat sallivat kaksinkertaisen suorituksen ainakin vikatilanteissa
- Toipumislohkojen käyttö perustuu tähän



Kaksi tapaa toipua ajassa

- Toipuminen eteenpäin
  - Siirretään järjestelmän tilaa eteenpäin turvalliseen tilaan
  - Tilat määrättävä jokaiselle järjestelmälle erikseen
- Toipuminen taaksepäin
  - Palautetaan turvallinen aiemmin suoritettu tila
  - Yleisempi lähestymistapa

Dominoefekti kommunikoivat prosessit



Informaatio

- Koodataan lisätietoa, jolla voidaan havaita ja jopa korjata virheitä
  - Tallennetaan sama tieto kahteen kertaan
  - Pariteettibitti
  - Tarkistussumma
- Hamming etäisyys
  - Käytetään arvioimaan bittijonojen samuutta

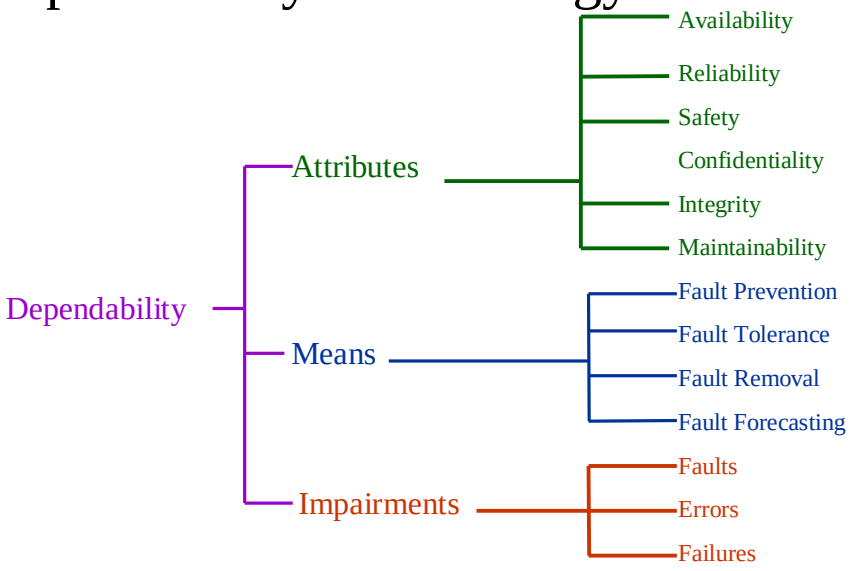
Luotettavuus ja vikasietoisuus

- Yleistä:
- Monentaminen ja toisintaminen
- **Turvallisuus ja luotettavuus**
- **Riippuvuus (dependability)**
- Esimerkkejä

Turvallisuus ja luotettavuus

- Turvallisuus: Järjestelmä ei aiheuta hengenmenoja, loukkaantumisia, sairauksia, omaisuuden tai ympäristön vaurioitumista.
- Luotettavuus: mittari, joka kertoo kuinka hyvin järjestelmä vastaa toiminnankuvausta.
- Turvallisuus on siis todennäköisyys, että järjestelmän toiminta ei johda vahinkoon toimipa järjestelmä suunnitelmansa mukaan tai ei.

Dependability Terminology



Virheen käsittely

- Virheen havaitseminen
  - äänestys, hyväksymistesti
- Virheen (tai vahingon) eristäminen
- Virheestä toipuminen
  - Konsensus, toipumislohko
- Virheen korjaus
  - Korjauksen jälkeen palautetaan taas toimintaan

Luotettavuus ja vikasietoisuus

- Yleistä:
- Monentaminen ja toisintaminen
- Turvallisuus ja luotettavuus
- Riippuvuus (dependability)
- **Esimerkkejä**
  - Avaruussukkulan tietokonejärjestelmä
  - Prosessin ylösnousemus (process resurrection)

Avaruussukkula

- Viisi tietokonetta, joista neljä samassa ryhmässä
- Viidennessä koneessa on muista täysin riippumaton ohjelmistototeutus lähinnä varalla
- Kaikki koneet kytketty yhteen viidellä rinnakkaisella väylällä

Avaruussukkula

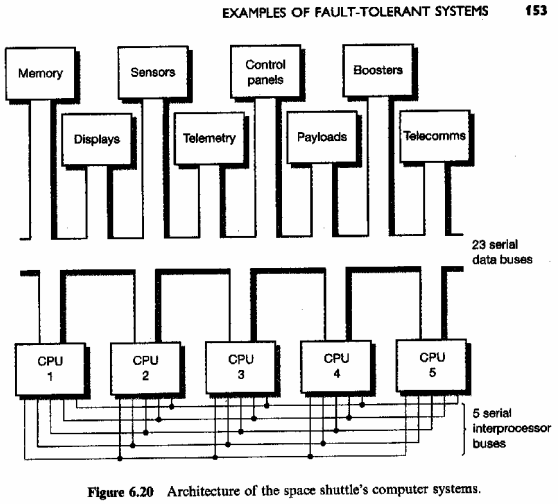


Figure 6.20 Architecture of the space shuttle's computer systems.

Kuva 6.20 teoksesta: N. Storey Safety-critical computer systems. Prentice Hall 1996

Prosessin ylösnousemus

- Artikkeli
  - Kihwal Lee ja Lui Sha: Process Resurrection: A Fast Recovery Mechanism for Real-Time Embedded Systems. Proc of RTAS'05, IEEE.
- Idea:
  - Tehdään uudelleenkäynnistys niin nopeaksi, että uudelleenkäynnistetty prosessi ehtii valmistua alkuperäisen aikarajan sisällä!

Prosessin ylösnousemus

- Ratkaisuidea:

Artikkelin kuva 2

Yhteenveto

- Virheiden ja vikojen luokat ja mallit
- Toisinnetaan
  - Laitteisto: N-modulaarinen, äänestys
  - Ohjelmisto: N-versiointi, toipumislohko
  - Aika: Toivutaan eteen- tai taaksepäin
  - Informaatio: tuplaus, lisäkoodaus, tarkistussumma
- Prosessi: virheen havaitseminen, eristäminen, toipuminen, korjaaminen