

# Ohjelmiston laadulliset metriikat

Kasper Hirvikoski

*“As software engineers and designers, it’s easy to lose sight of the very human impact of the bugs we ship.”*

– MICHAEL JUREWITZ

# Kehityksen haasteet

- Testaamisen ja käytön välillä on suuri kuilu
- Virheiden korjaaminen jälkikäteen on kallista
- Virheitä on mahdoton välttää
- Kehittäminen on *haastavaa*!

# Mitä laatu on?

Asiakkaan ja käyttäjien asettamat toiveet ja tarpeet.

– ISO & IEEE

# Käyttäjä kokee ulkoisen laadun



*Hups! Ohjelmisto  
kaatui.*

## Sisäinen laatu kertoo toteutuksesta

```
10 i = 0
20 i = i + 1
30 PRINT i; " neliö = "; i * i
40 IF i >= 10 THEN GOTO 60
50 GOTO 20
60 PRINT "Valmis!"
70 END
```

# Miten arvioida laatua?

- Lasketaan julkaisun jälkeisten virheiden määrä (ulkoinen laatu)
- Arvioidaan mm. lähdekoodin laatua, monimutkaisuutta ja riippuvuuksia (sisäinen laatu)

=

*Metriikat* arvioivat ohjelmistoa ja  
etsivät siitä kriittisiä (virheherkkiä)  
komponentteja.

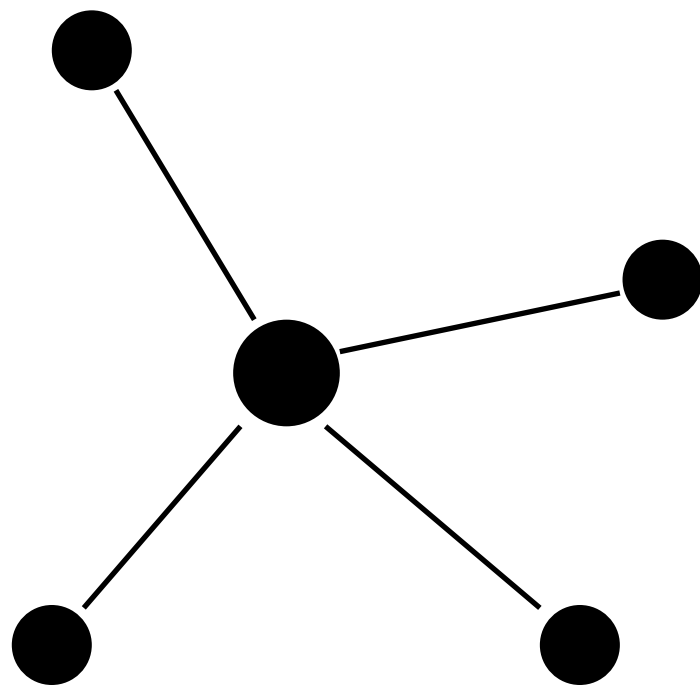


**Muutokset lisäävät monimutkaisuutta**

+ 4257

- 1631

**Komponenteilla on usein riippuvuuksia**



**Virheet eivät paljastu ilman testejä**

```
assertTrue(true);
```

1/1631 riviä katettu testeillä

# Perinteiset metriikat

- Perinteisimmät metriikat pohjautuvat koodirivien määrään (< 1990)
- CK-metriikat keskittyvät metodien, ylliluokkien ja lapsien lukumäärään sekä tutkivat komponenttien riippuvuuksia, vastuita ja yhtenäisyyttä (1994)
- Muodostavat muiden metriikoiden pohjan

## Koodikirnu tutkii muutoksia

- Versionhallintajärjestelmistä saadaan tieto kuinka monta riviä on *lisätty*, *poistettu* tai *muutettu* edelliseen versioon nähden
- Ehdottomat mitat eivät kerro totuutta
- *Suhteutetaan* nämä muutokset aikaan

## Koodikirnu tutkii muutoksia

1. Käsiteltyt koodirivit / Koodirivien määrä
2. Poistettut koodirivit / Koodirivien määrä
3. Käsiteltyt tiedostot / Tiedostojen määrä
4. Muutosten määrä / Käsiteltyt tiedostot

## Koodikirnu tutkii muutoksia

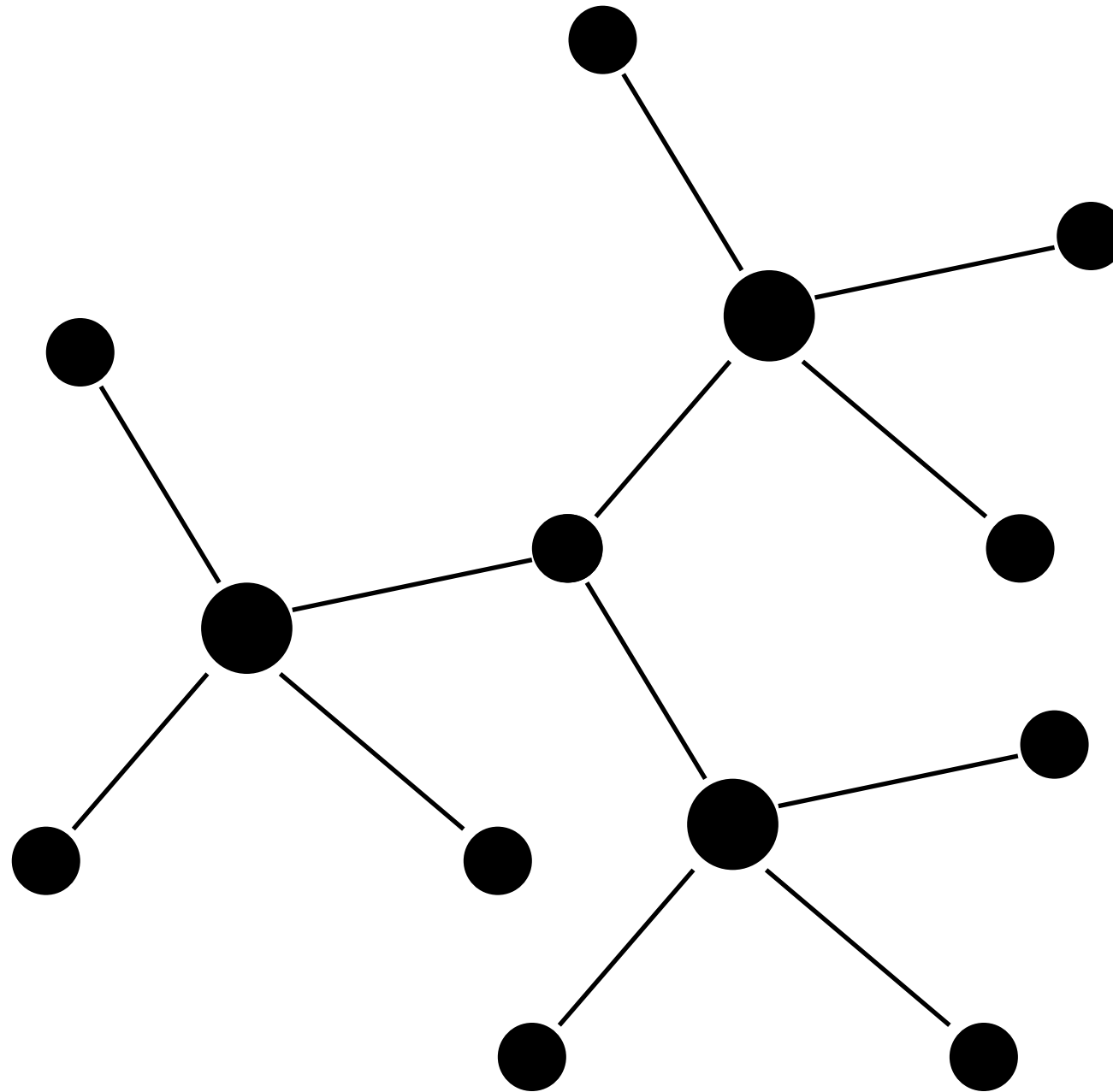
5. Muutosten ajanjakso / Tiedostojen määrä
6. Käsiteltyt ja poistetut koodirivit / Muutosten ajanjakso
7. Käsiteltyt koodirivit / Poistetut koodirivit
8. Käsiteltyt ja poistetut koodirivit / Muutosten määrä

=

Koodi joka *muuttuu* useasti  
ennen julkaisua on virheherkkää.



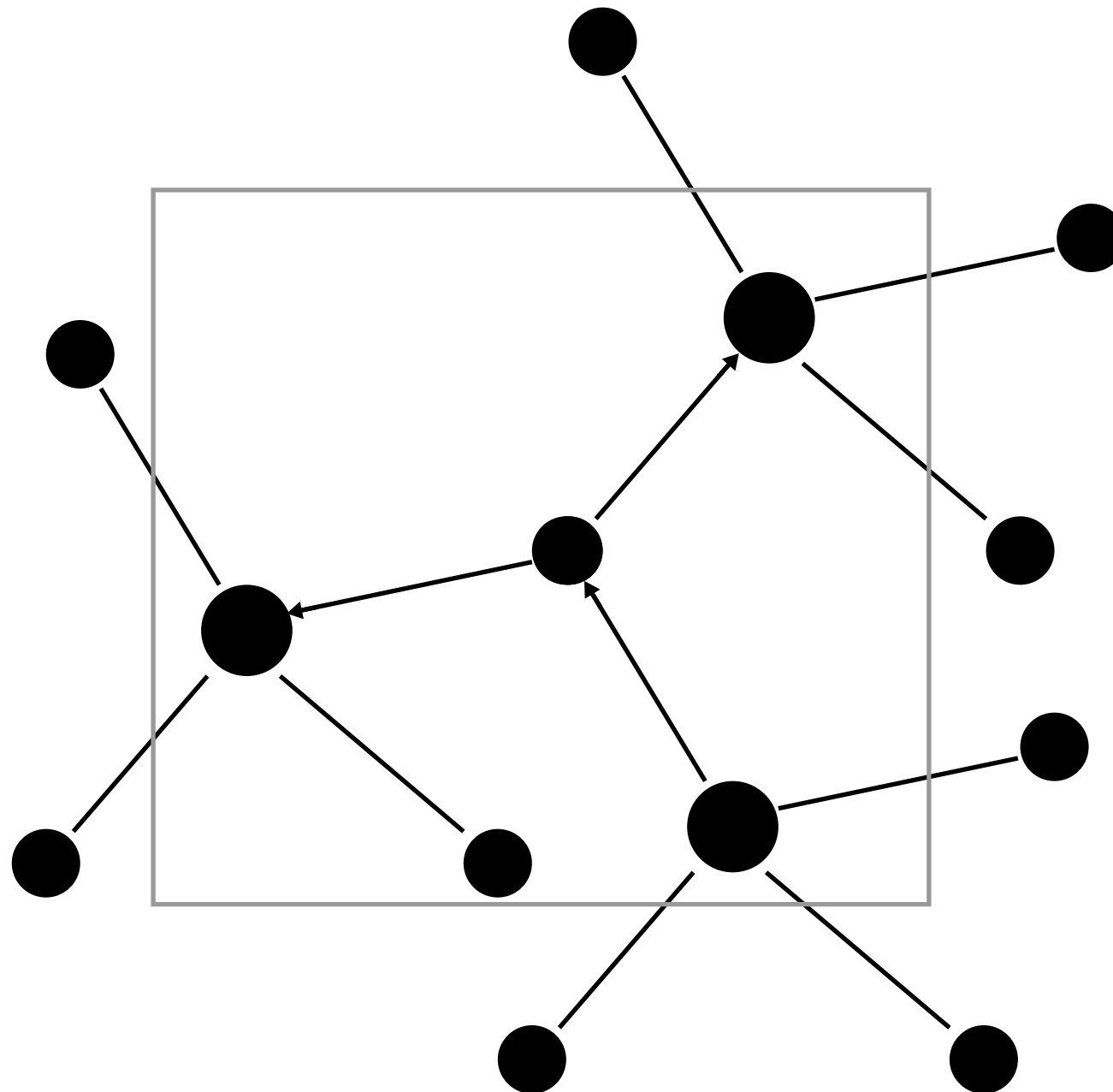
# Verkkoanalyysi tutkii riippuvuuksia



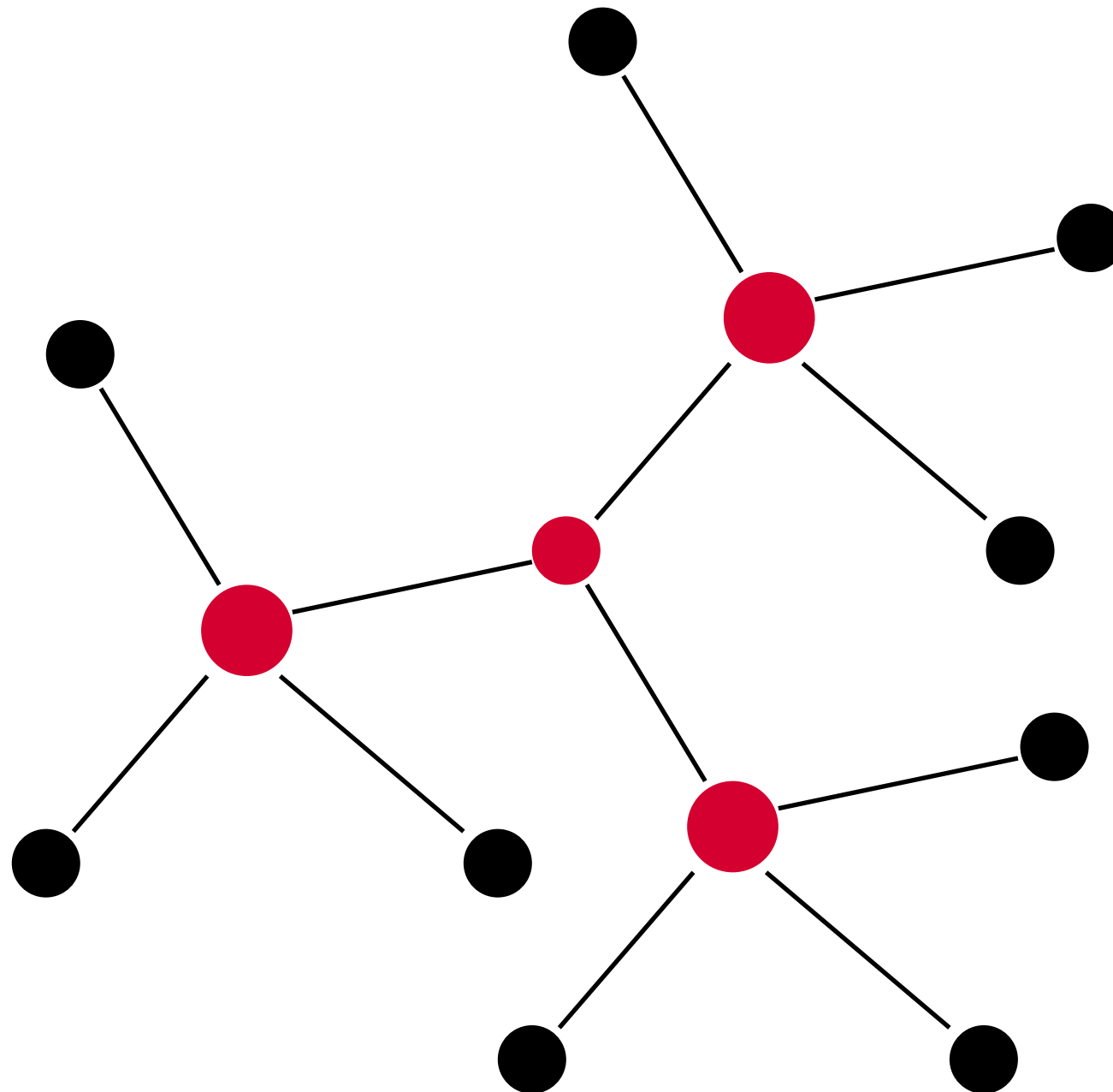
# Verkkoanalyysi tutkii riippuvuuksia

- Komponenttien keskinäiset suhteet paljastuvat verkoista
- Yhteydet kertovat kuinka paljon työtä tarvitaan yhteyksien ylläpitämiseen

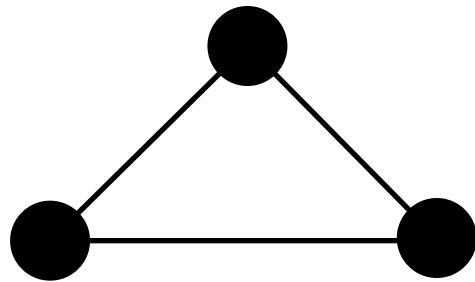
# Yksittäisen komponentin riippuvuudet



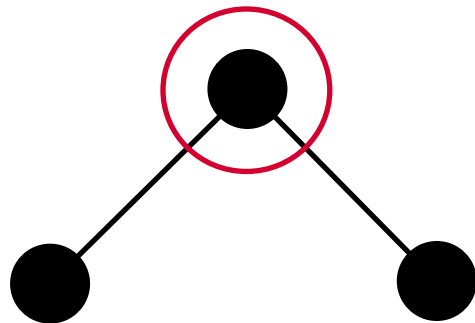
# Komponenttien tärkeys



# Rakenteelliset puutteet

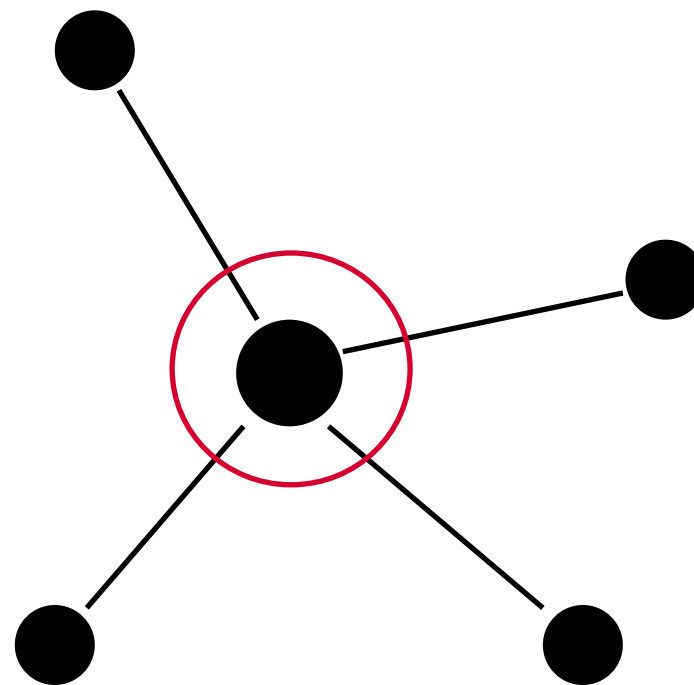


Tasapaino  
komponenttien  
välillä

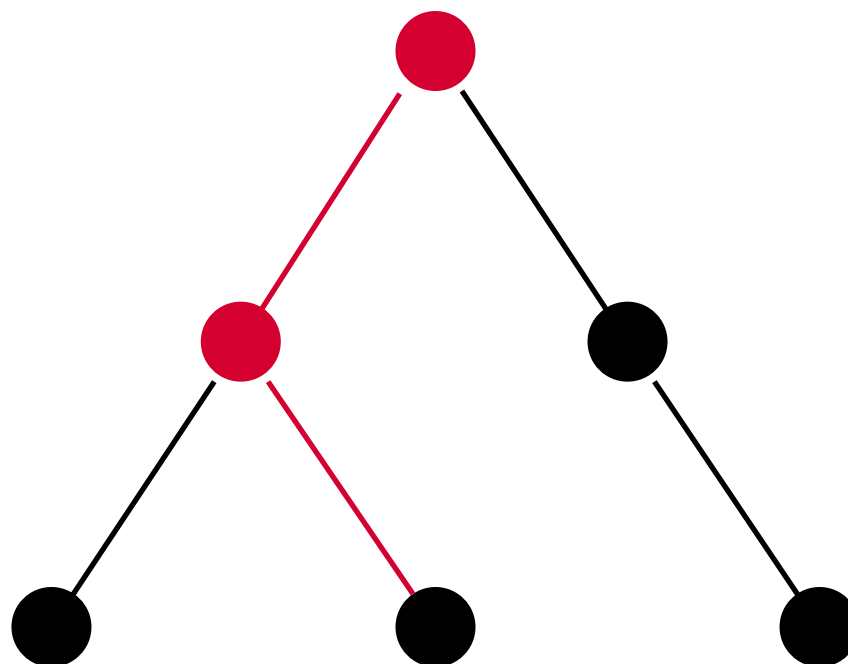


Välittäjäsolmu

# Komponentin keskeisyys



# Muutokset leviävät riippuvuuksiin



=

*Keskeiset* komponentit ovat  
usein virheherkkiä.



# Testikattavuus varmistaa laatua

- Suurempaa testikattavuutta seuraa pienempi määrä virheitä
- Testikattavuuden saavuttaminen kasvaa eksponentiaalisesti, mutta virheet vähenevät vain lineaarisesti
- Mielenkiintoista on tietää kuinka laadukkaita testit ovat

## Mutaatiotestaus arvioi testien laatua

- Simuloi kehittäjän aiheuttamia virheitä tekemällä alkuperäiseen lähdekoodiin pieniä syntaksimuutoksia
- Mutanttien ei pidä mennä testeistä läpi
- Haasteet: mutanttien valtava määrä ja yhtäläisyys

=

Virheitä ei pystytä havaitsemaan  
jos niitä aiheuttavaa koodia ei ole  
*testattu* vähintään yhdellä testillä.

# Metriikoiden haasteet

- Vaikka metriikoita on tutkittu jo pitkään, niiden täydellistä potentiaalia ei ole vieläkaan valjastettu käyttöön
- Metriikat ovat perimmiltään vielä osittain tieteellisiä eikä niitä arvioivia kehittyneitä työkaluja ole helposti saatavilla
- Työkalut tulee saada kehittäjien käsiin!

# Koodin kehittäjänä ihminen

commit bc2d230ae4d125f657309bb486259d0640e0904d

Author: Arto <arto@afterlife.Elisa>

Date: Mon Dec 10 00:52:01 2012 +0200

asdasd

commit e4d8a591ff06aeac7c00ae1a30deccf4793a3ca2

Author: Arto <arto@afterlife.Elisa>

Date: Mon Nov 19 00:44:39 2012 +0200

dadaidaam

=

Lopulta *laadukkaan*  
ohjelmiston tekee laadukkaat  
kehittäjät.

=

*Ulkoinen ja sisäinen laatu* lähtee  
siitä, että ohjelmisto  
suunnitellaan,  
toteutetaan ja testataan käyttäen  
hyväksi todettuja  
ohjelmointitapoja ja malleja.

