

Käytön laadun ja tuotelaadun väliset riippuvuudet eli

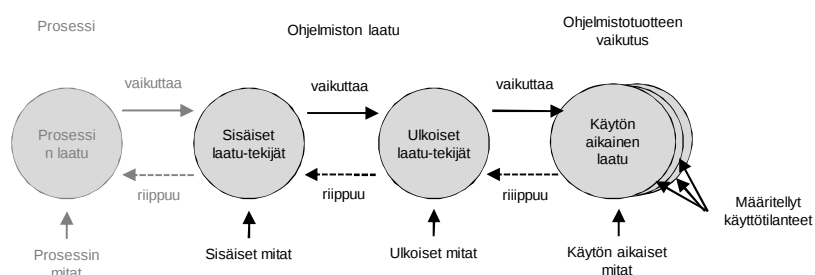
LAADUN VAIKUTUSKETJU

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

50

Laadun vaikutusketju



[Nev13] Risto Nevalaisen esitys "Järjestelmän ja ohjelmiston mittaaminen".
<http://www.sfsedu.fi/materiaalit> (kts. IT-Standardit)

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

51

Laadun vaikutusketju

- Lähtökohtana ovat käyttäjän tarpeet ja käytön tilanteet
- Esimerkki: joukkoliikenteen aikataulutiedon hakeminen on-line palvelusta mobiililaitteella
- Käyttäjän tarpeet ja odotukset (käytön aikaisen laadun tärkeät piirteet):
 - ◆ *Vaikuttavuus, tyytyväisyys* - Tietojen pitää olla täsmällisiä ja ajan tasalla
 - ◆ *Tehokkuus, tyytyväisyys* - Halutun linjan ja vuoron tietojen löytäminen pitää olla intuitiivista ja nopeaa

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

52

Laadun vaikutusketju

- Käytön laatuun vaikuttavat palvelun ulkoiset laatupiirteet (järjestelmän *black-box* tarkastelu)
 - ◆ *Käytettävyys* - esimerkiksi
 - Kuinka monen askelen (klikkauksen, muun syötteen) takana halutut tiedot ovat?
 - Kuinka selvän ja nopean palautteen käli antaa käyttäjän toiminnoista?
 - ◆ *Suorituskyky (tehokkuus)* - esimerkiksi
 - Kuinka kauan *koko* käyttötapaus kestää?
 - Kuinka kauan yksittäisten tietonäyttöjen tai sivujen latautuminen kestää?
 - Kuinka nopeasti sovellus käynnistyy?
 - Kuinka nopea tietoliikenneyhteys tarvitaan jne.
 - ◆ *(Data quality – ei käsitellä tällä kurssilla)*

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

53

Laadun vaikutusketju

- Sisäinen laatu mahdollistaa (tai estää!) halutun ulkoisen laadun saavuttamisen (järjestelmän *white-box* eli lasilaatikkotarkastelu)
 - ◆ Suorituskyky (esim. tiedonhakujen kesto) määräytyy pitkälle ohjelmiston (ja laitteiston) *arkkitehtuurista* ja *teknologiavalinnoista* sekä tekniikoiden oikeasta käytöstä
 - ◆ Käytettävyyteen vaikuttavat suoraan *käyttöliittymän suunnittelu* (esim. useimmin tarvittujen toimintojen nostaminen aloitusnäyttöön sekä niiden suorittamiseen tarvittavien vuorovaikutusten määrän minimointi) ja epäsuoremmiin palvelun eri toimintojen käytön monitorointi- ja tilastointikyvykyys (instrumentointi, logit)

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

54

Laadun vaikutusketju

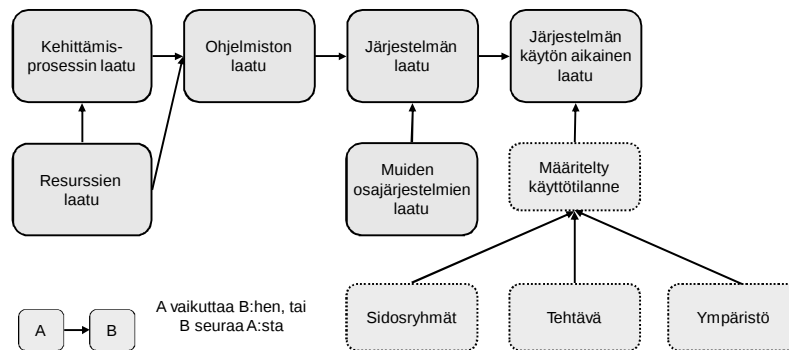
- Palvelun toteutuksessa käytettävät resurssit ja menetelmät vaikuttavat kaikkiin laadun alueisiin
 - ◆ Käytettävyyssuunnittelijoiden ja ohjelmisto-arkkitehtien sekä –kehittäjien *tiedot*, *taidot* ja *kokemus* - ja ennen muuta asenne (myös johdon)
 - ◆ Aikataulu ja budjetti suhteessa toim. määrään
 - ◆ Käyttäjien *osallistaminen* kehitykseen ja käyttäjien *palautteen* systeemaattinen *kerääminen* ja *hyödyntäminen*
 - ◆ Laatuavoitteiden esillä pito ja jatkuva arviointi

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

55

Laadun vaikutusketju



[Nev13] Risto Nevalaisen esitys "Järjestelmän ja ohjelmiston mittaaminen".
<http://www.sfsedu.fi/materiaalit> (kts. IT-Standardit)

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

56

Käyttäjäryhmät ja laatu

ISO/IEC 25010 tunnistaa 3 käyttäjäryhmää

1. Pääasiallinen käyttäjä (primary user)
 - Henkilö, joka on järjestelmän kanssa vuorovaikutuksessa järjestelmän pääkäyttötarkoituksen mukaisesti
2. Toissijainen, pääkäyttöä tukeva käyttäjä (secondary user)
 - Sisällön tuottaja, järjestelmävalvoja, turvallisuusvalvoja
 - Ylläpitäjä, asentaja, suunnittelija (analyst), testaaja
3. Epäsuora käyttäjä (indirect user)
 - Vastaa järjestelmän tuottamia tulostietoja, mutta ei itse ole vuorovaikutuksessa sen kanssa

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

57

Käyttäjäryhmät ja laatu

- Jokaisella käyttäjäryhmällä on omat tarpeensa, jotka tulee ottaa huomioon järjestelmän määrittelyssä, suunnittelussa ja toteutuksessa
- Tuotelaadun* piirteet ovat eri tavalla tärkeitä eri käyttäjäryhmille
- Ohjelmiston ja sitä suorittavan järjestelmän ominaisuudet määräävät tuotelaadun eri käyttöyhteyksissä (ja eri käyttäjäryhmille)
 - Taulukko seuraavalla dialla

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

58

Käyttäjäryhmät ja laatu

Tuotelaatu-piirre	Vaikuttaa pääkäyttäjän kokemaan käytön laatuun	Vaikuttaa ylläpitotehtävissä koettuun laatuun	Muiden sidosryhmien näkemä laatu
Toiminnallinen sopivuus	X		
Tehokkuus	X		X
Yhteensopivuus		X	
Käytettävyys	X		
Luotettavuus	X		X
Turvallisuus	X		X
Ylläpidettävyys		X	
Siirrettävyys		X	

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

59

Sisäisen ja ulkoisen laadun piirteet

Toiminnallinen sopivuus	Tehokkuus	Yhteensopivuus	Käytettävyys	Luotettavuus	Turvallisuus	Ylläpidettävyys	Siirrettävyys
Toiminnallinen kattavuus	Vasteaika Resurssien käyttösuhte	Rinnakkaiselo	Soveltuvuuden selkeys	Ohjelmistotuotteen kypsyy	Tietosuoja	Rakenteellinen selkeys	Sovitettavuus
Toiminnallinen oikeellisuus	Kapasiteetti	Yhteensopivuus	Opittavuus	Saatavuus	Koskemattomuus	Uudelleenkäytettävyys	Asennettavuus
Toiminnallinen soveltuvuus			Helppokäyttöisyys	Vikasietoisuus	Kiistämättömyys	Analysoitavuus	Korvattavuus
			Käyttövirheiden estäminen	Toipumisvalmius	Aitous	Muunneltavuus	
			Käyttöliittymän miellyttävyys		Todenperäisyys	Testattavuus	
			Matala kynnys				

Ulkoinen, pääasiassa ulkoinen, sekä sisäinen että ulkoinen, pääasiassa sisäinen

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

60

Mistä ne tulevat?

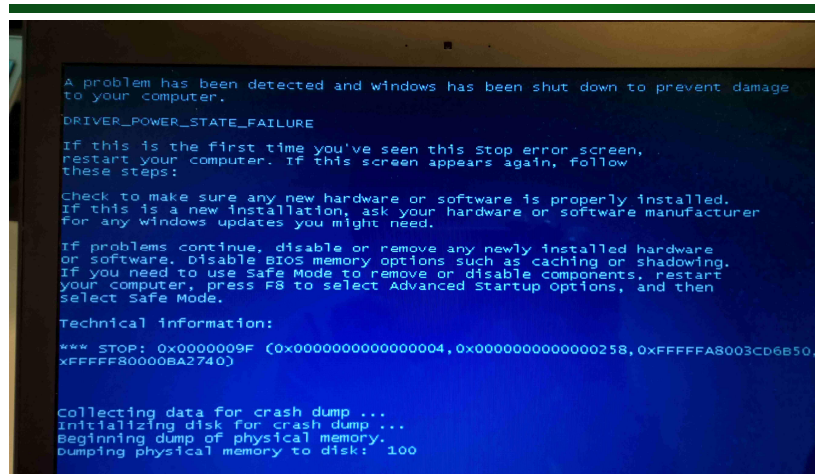
OHJELMISTOJEN VIOISTA

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

61

"Blue screen of death"



Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

62

Vioista

- *Vikojen vähäinen määrä* on perinteisesti ollut yksi tärkeimpiä laadun indikaattoreita
- Esimerkiksi ohjelmistojen laadun ja kehitysprojektien tuottavuuden parissa pitkän päivätyön tehnyt Capers Jones on määritellyt^[Jon08] laadun seuraavasti
 - ◆ "The absence of defects that would make software either stop completely or produce unacceptable results. Defects can be traced to requirements, to design, to code, to documentation, or to bad fixes of previous defects."
- Tällä tavoin määriteltä laatu on kvantifioitavissa melko suoraviivaisesti (sekä *mittaamista* että *ennustamista* varten), mutta on muuten kovin suppea
 - ◆ Jones on myöhemmin laajentanutkin laadun määritelmäänsä kattamaan mm. käytettävyyden
 - ◆ Aina ei myöskään ole selvää, mitkä asiat tulkitaan vioiksi ja miten ne kirjataan (käyttäjät, testaajat ja kehittäjät voivat tulkita asian eri tavalla)

[Jon08] Capers Jones, *Applied Software Measurement*. McGraw-Hill, 2008.

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

63

Virhe – vika - häiriö

- Ohjelmistojen laadun kannalta on oleellista erotella toisistaan (kehitys-) *virhe* eli *erehdys* (error, mistake), (ohjelmisto-) *vika* (fault) ja (toiminta-) *häiriö* (failure/defect).
 - ◆ Koodin kirjoittajat tekevät *virheitä*. Virhe voi koskea rakennetta (esim. viittaus väärään muuttujaan) tai koodin logiikkaa (esim. vaatimus on tulkittu väärin, käytetään jotain APLa väärällä tavalla, väärä ehtolauseen vertailuoperaattori jne).
 - ◆ Kun erehdyksessä kirjoitettu koodi saa ohjelmiston toimimaan väärin eli tapahtuu *häiriö*, kyse on (ohjelmisto-) *viasta*. Kaikki viat eivät tee näin, sillä vikaa seuraavat koodirivit saattavat "neutraloida" vian vaikutuksen, tai vika voi peittyä toisten vikojen vaikutuksesta.

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

64

Virhe – vika - häiriö

- ◆ Kun viallista koodia suoritetaan, syntyy häiriö. Kaikista ohjelmistovioista ei seuraa toimintahäiriöitä. Vika saattaa olla hautautuneena koodiin sellaiseen paikkaan, että sitä ei suoriteta koskaan (esimerkiksi harvinaiseen poikkeuskäsittelijään tai käytännössä mahdottomaan parametrien kombinaatioon).
- Puhekielen *bugi* (bug) tarkoittaa yleensä virhettä tai vikaa
- *Defekti* (defect) voi tarkoittaa myös vikaa

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

65

Virheet, viat ja häiriöt

- Vain osa kaikista virheistä näkyy vikoina ja osa vioista häiriöinä loppukäyttäjälle asti (ohjelmiston tuotantoversiossa)
 - ◆ Suuri osa häiriöistä tulee ilmi jo kehitysaikana testauksen ja muun laadunvarmistuksen ansioista
- Kuitenkin mikä tahansa koodiin jäänyt passiivinen vika saattaa koodia muutettaessa aktivoitua ja mikä tahansa vika saattaa sopivassa käytössä aiheuttaa häiriön.
- Kehitystiimille laadukas tuote tarkoittaa vähäistä vikojen määrää. Loppukäyttäjän kannalta se tarkoittaa vähäistä häiriöiden määrää.

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

66

Virheet, viat ja häiriöt

- Yleensä ohjelmistoista raportoitujen häiriöiden määrä on välillä 0,1 – 5,0 häiriötä tuhatta koodiriviä kohden
- Capers Jones on kerännyt vuosikymmeniä asiakkaidensa projekteista (n. 13000 kpl) tietoja *häiriöpotentiaaleista* (defect potentials) ja *häiriöiden poistamistehokkuudesta* (defect removal efficiency)
- Jonesin data^[Jon13] osoittaa keskimääräiseksi häiriöpotentiaaliksi 5,0 häiriötä per toimintopiste (~ 50-60 riviä Javan tasoista ohjelmointikieltä) ja keskim. häiriöiden poistamistehokkuudeksi 85%
 - ◆ Eli noin 15 häiriötä per 1 KLOC
 - ◆ Parhailla projekteilla luvut ovat 2,5 ja 96% eli 2,6 per 1 KLOC
 - ◆ Kts. diat 23 – 25 [Jon13]

[Jon13] Capers Jones, *Software Quality: A Survey of the State of the Art*, 2013.
<http://namcookanalytics.com/software-quality-survey-state-art/> (haettu 16.3.2015)

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

67

Ohjelmistojen virhetyypit 1

- Koska ohjelmistohankkeissa tehty virheet vaikuttavat ohjelmiston laatuun, on tärkeää tietää, minkä tyyppisiä virheitä tehdään.
- Galin^[Gal04] listaa yhdeksän virhetyyppiä ja antaa kustakin esimerkkejä:
 1. Väärin määritellyt vaatimukset
 - § Vaatimusten määrittelytavassa on virheitä
 - § Puuttuvia vaatimuksia
 - § Epätäydellisiä vaatimuksia
 - § Ylimääräisiä vaatimuksia
 2. Asiakkaan ja kehittäjän väliset kommunikaatiokatkokset
 - § Asiakkaan vaatimukset ymmärretään väärin
 - § Asiakkaan muutostoiveet ymmärretään väärin
 - § Asiakkaan vastaukset kysymyksiin ymmärretään väärin

[Gal04] Daniel Galin: Software Quality Assurance - From Theory to Implementation. Pearson Education, 2004

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

68

Ohjelmistojen virhetyypit 2

3. Tahalliset poikkeamat vaatimuksista
 - § Ohjelmistossa uudelleenkäytetään koodia ilman kunnollista analyysia vaadittavista muutoksista
 - § Osa vaatimuksista jää toteuttamatta budjetin tai aikataulun ylittymisen johdosta
 - § Kehitystiimi muuttaa vaatimuksia kysymättä asiasta asiakkaalta
4. Ohjelmiston logiikan suunnitteluvirheet
 - § Väärien tai tilanteeseen sopimattomien algoritmien käyttö
 - § Tilasiirtymävirheet
 - § Raja-arvovirheet
5. Koodausvirheet
 - § Painovirheet
 - § Kielioppivirheet
 - § Vaatimusten tai suunnittelun tulkintavirheet

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

69

Ohjelmistojen virhetyypit 3

6. Dokumentointi- ja koodausohjeiden noudattamatta jättäminen
 - § Vaikealukuiset dokumentit
 - § Vaikeaselkoinen ohjelmakoodi
7. Testausprosessissa oikomisiet
 - § Vajaa testaus suunnitelma
 - § Löydettyjen vikojen ja virheiden keuhno dokumentointi
 - § Tiukan aikataulun johdosta tapahtuva puutteellinen vikojen ja virheiden korjaus
8. Toimintatapavirheet
 - § Väärin tulkittu ohjelmiston tarkoitettu käyttötapa (käyttötapaus)
9. Dokumentointivirheet
 - § Dokumentoimattomat toiminnot
 - § Virheelliset toimintaohjeet
 - § Ylimääräiset ja puuttuvat toiminnot

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

70

Häiriöiden lähteet Jonesin mukaan

- Capers Jones on kerännyt vuosikymmeniä asiakkaidensa erityyppisistä projekteista tietoja häiriötiheyksistä ja niiden synty-yhteyksistä (eli virhetyypeistä)
 - ◆ Katso dia 58 esityksessä [Jon13]
- Jones & Bonsignour^[JoB11]
 - ◆ “For large systems, *requirements defects*, *architectural defects*, and *design defects* are the main sources of quality problems” (Ch. 2)

[Jon13] Capers Jones, *Software Quality: A Survey of the State of the Art*, 2013.
<http://namcooanalytics.com/software-quality-survey-state-art/> (haettu 16.3.2015)

[JoB11] C. Jones, O. Bonsignour, *The Economics of Software Quality*. Addison-Wesley Professional, 2011
Kevät 2016 Ohjelmistoprosessit ja ohjelmistojen laatu 71

Vikojen estäminen ja poistaminen

- Monet ohjelmistokehityksen menetelmät, työtavat ja työkalut tähtäävät vikojen *syntymisen estämiseen* tai vikojen tehokkaaseen *poistamiseen*
- Tehokkuus, jolla kehityksen aikana poistetaan vikoja ohjelmistosta, on perinteisesti ollut keskeinen mittari kehitysprosessin laadulle
 - ◆ Tehokkuuden mittaaminen vaatii kirjanpitoa havaituista vioista, niiden vakavuudesta ja lähteestä/syntyvaiheesta
 - ◆ Nopeaa reagointia korostavissa web-projekteissa painotetaan pikemmin häiriöiden nopeaa havaitsemista tuotannossa ja korjausten pikaista julkaisua
- Tästä lisää myöhemmin kurssilla

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

72

Show me the money

OHJELMISTOJEN LAADUN TALOUDELLINEN MERKITYS

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

73

Laatu, kustannukset ja tuotot

- Ohjelmistoja tuottavia ja käyttäviä organisaatioita kiinnostaa *taloudellisesta näkökulmasta laadun vaikutus* seuraaviin asioihin^[JoB11]
 - ◆ Kehityksen, ylläpidon, parannuksen ja tuen kustannukset.
 - ◆ Kehityksen, ylläpidon, parannuksen ja tuen aikataulut
 - ◆ Ohjelmiston suorat myyntitulot
 - ◆ Liitännäispalveluista ja –tuotteista saatavat tulot
 - ◆ O:n käytön oppimiskäyrä
 - ◆ O:n käytöstä seuraavat operationaaliset säästöt
 - ◆ Uudet liiketoimintamahdollisuudet, jotka o:n käyttö tuo mukanaan
- Liiketoiminnan ja yrityksen johdon näkökulmasta on tärkeää, että käytetty laadun määritelmä ja laatumittarit mahdollistavat laadun vaikutusten *kvantifioinnin*

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

74

Laadun taloudellinen merkitys

- Tiivistettynä, laatu vaikuttaa neljään taloudellisesti merkittävään alueeseen ohjelmistojen tuotannossa ja käytössä
 1. Ohjelmistojen kehittämisen ja ylläpidon kustannukset ja työmäärä
 2. Kaupallisten ohjelmistojen kannattavuus
 3. Inhimillisen työn tuottavuus
 4. Ohjelmistojen rooli uusissa ja innovatiivisissa tuotteissa ja palveluissa

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

75

Huonon laadun vaikutus kehitykseen ja ylläpitoon

● Huono laatu

- ◆ Venyttää testausta ja aiheuttaa epävarmuutta toimitusaikatauluun
- ◆ Korjaukset ja uudelleen tekeminen aiheuttavat suurimman kustannuspaineen
- ◆ Aiheuttaa ylitöitä ja budjetin pettämisen
- ◆ Vaatii julkaisun jälkeen kallista asiakastukea
- ◆ Lisää julkaisun jälkeisen ylläpidon kustannuksia
- ◆ Voi viedä oikeudenkäyntiin sopimusrikkomuksesta

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

76

Hyvän laadun vaikutus kehitykseen ja ylläpitoon

● Hyvä laatu

- ◆ Lyhentää testaukseen käytettyä aikaa ja parantaa toimitusaikataulun ennustettavuutta
- ◆ Vähentää korjausten ja uudelleen tekemisen määrää jopa 50%
- ◆ Vähentää yllättäviä ylitöitä ja pienentää kuluylityksiä
- ◆ Vähentää julkaisun jälkeistä asiakastuen tarvetta
- ◆ Vähentää julkaisun jälkeisen ylläpidon kustannuksia
- ◆ Pienentää oikeudenkäyntien riskiä

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen laatu

77

Huonon laadun vaikutus inhimillisen työn tuottavuuteen

● Huono laatu

- ◆ Lisää toimintahäiriöistä johtuvaa odottelua ja toimettomuutta
- ◆ Voi hidastaa toimenpiteiden suoritusta ja pienentää työpanoksen tuottavuutta
- ◆ Voi johtaa onnettomuuksiin tai transaktiovirheisiin
- ◆ Häiriöiden seurausten korjaaminen aiheuttaa ylimääräistä työtä
- ◆ Lisää myös väärin häiriöraporttien määrää
- ◆ Johtaa välillisiin vahinkoihin ja kalliisiin liiketoiminnan ongelmiin

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

78

Hyvän laadun vaikutus inhimillisen työn tuottavuuteen

● Hyvä laatu

- ◆ Vähentää toimintahäiriöitä ja niistä johtuvaa toimettomuutta
- ◆ Auttaa optimoimaan työntekijöiden työsuorituksia
- ◆ Vähentää onnettomuuksien tai transaktiovirheiden todennäköisyyksiä
- ◆ Pienentää häiriöiden seurausten korjaamistarvetta
- ◆ Vähentää väärin ja aiheettomien häiriöraporttien määrää
- ◆ Vähentää välillisiä vahinkoja ja liiketoiminnan ongelmia

Kevät 2016

Ohjelmistoprosessit ja ohjelmistojen
laatu

79

"Quality is free"

● Philip Crosby

- ◆ "It is always cheaper to do the job right the first time"
- ◆ "Quality is free, but no one is ever going to know it if there isn't some sort of agreed-on system of measurement"
- ◆ <http://www.philipcrosby.com/25years/read.html>
- ◆ http://en.wikipedia.org/wiki/Philip_B._Crosby