

Lars Wirzenius <liw@iki.fi> wrote:

Kaikki, mitä ikinä tarvitsee tietää ohjelmoinnista, lukuunottamatta joitakin lähes yhdentekeviä yksityiskohtia ohjelmointikielten syntaksista ja semantiikasta, tietokoneiden toimintaperiaatteista, ynnä muista sellaisista turhuuksista.

21.9.2005

Minkälainen

on hyvä
ohjelma?

Mitä
tarkoittaa
ohjelman
laatu?



Ohjelma on hyvälaatuinen, jos se...

- tekee mitä sen on tarkoituskin tehdä
- ei tee ylimääräisiä asioita
- kestää virhetilanteita järkevästi
- on helppo käyttää, alussa ja pidemmän päälle
- on vaivaton paketoida, asentaa, päivittää, poistaa
- ei ole kohtuuttoman virheellinen ja virheet on helppo korjata
- on riittävän tehokas

Laatu = ei yllätyksiä

(paitsi välillä joskus positiivisesti)

Minkälainen on hyvä
ohjelmoija?

Hyvä ohjelmoija...

- osaa perusasiat: algoritmit, tietorakenteet, analyysin; paradigmat (imperatiivinen, olio, funktionaalinen); suunnittelu- ja toteutusmenetelmät (top-down, bottom-up, vesiputous- ja silmukkamallit, XP/Agile, TDD, UML, refaktorointi, design patterns); tarpeelliset kielet (C, C++, C#, Java, Lisp, Perl, Python, PHP, HTML, XML/XSLT/DTD), työkalut (SQL, Eclipse, CVS, SVN, Emacs, Word, Excel) ja protokollat (HTTP, FTP, SMTP, TCP/IP, SMPP, XMLRPC/SOAP, RFC2822/MIME); tietoturvan

Ei pidä uskoa kaikkea roskaa...

- Asenne ratkaisee!
 - pitää ohjelmoinnista (harrasteohjelmoi!), oppii mielellään uutta, näkee isot kuviot eikä vain nippeleitä
- Tiedot ja taidot ovat hyödyllisiä, välttämättömiä, mutta eivät ratkaisevia
- Mitä ei osata, se opitaan
 - jos ei haluta oppia, on asennevika ja iso ongelma

Kyllä tietotaitoakin tarvitaan...

- Ohjelmointi = maailmankaikkeuden luomista pelkällä ajatuksen voimalla
- Tärkeää osata: ajattelun tehostajat
 - opi uusia tapoja ajatella
 - esimerkki: Lisp räjäyttää tajunnan
- CV:ssä on hyvä olla oleelliset lyhenteet, jotta HR-osasto osaa tunnistaa hyväksi tyypiksi
 - tietysti pitää osata se, mitä työssään tarvitsee

Miten kehittyä?

- Kirjoita ohjelmia, myös harrastuksena
 - erilaisia ohjelmia
- Ylläpidä muiden koodia (tai edes lue)
- Lue ylipäätään, paljon, monenlaista
 - alan kirjallisuutta
- Keskustele, väittele, polemisoi, ajattele
- Kirjoita ohjelmoinnista, pidä esitelmää

Miten hyviä ohjelmia
tehdään?

P.t.o.t.m. on ratkaisu!

P.t.o.t.m. = Parhaiten toimiva ohjelmistotuotantomenetelmä

- Palkataan hyviä ohjelmoijia, jotka tulevat hyvin toimeen keskenään
- Kerrotaan mitä ohjelman tarvitsee tehdä, riittävällä tarkkuudella
- Annetaan riittävät resurssit, tarpeeksi aikaa, pyydettäessä palautetta ja TYÖRAUHA

P.t.o.t.m. tiivistettynä

- Hyviä tyyppejä
- Työrauha

Tuottavuus = hack mode

- "Flow"
 - syvän keskittymisen tila
 - vaatii ainakin 15 min työrauhaa alkaakseen
 - katkeaa helposti häiriöihin
- Ilmankin pärjää, jos on moninkertaisesti aikaa
 - mutta se ei ole ollenkaan yhtä hauskaa
 - hack mode = nirvana

March of Cambreadth (Heather Alexander)

Axes flash, broadswords swing,
Shining armour's piercing ring
Horses run with polished shield
Fight those bastards 'till they yield

Midnight mare and blood red roan
Fight to keep this land your own
Sound the horn and call the cry
How many of them can we make die?

Follow orders as you're told
Make their yellow blood run cold
Fight until you die and drop
A force like ours is hard to stop

Close your mind to stress and pain,
Fight 'till you're no longer sane
Let not one damn cur pass by,
How many of them can we make die?

Guard your women and children well,
Send these bastards back to hell
We'll teach them the ways of war,
They won't come here any more

Use your shield and use your head,
Fight 'till every one is dead
Raise the flag up to the sky,
How many of them can we make die?

Dawn has broke, the time has come
Move your feet to the marching drum
We'll win the war and pay the toll
We'll fight as one in heart and soul

Hyviä tapoja välttää hack mode

- Puhelimet
- Kokoukset, kysymykset, kahvitauot, keskeytykset
 - meteli ylipäättään
 - karsinat ("avokonttori")
- Meili, IRC
- Kiire

Ei makiaa mahan täydeltä

- Hack mode ei pelasta optimistista aikataulua
 - ei toimi: päivällä sekoillaan miten sattuu, illalla ja yöllä koodataan kovasti
 - tällä tavalla pääsee sairaalaan...
- Normaalit työpäivät, mutta ainakin puolikkaita päiviä pelkästään yhteen asiaan keskittyen
 - ei vain koodausta: kokouksetkin voivat päästä hack modeen, kimpassa se on vain vaikeampaa, kaikkien pitää olla mukana

Tuottavuus = oikeat työkalut

- Ammattilainen tunnetaan työkaluistaan
 - oikea väline kuhunkin hommaan
- Väärä työkalu => missä on ammattitaito?
 - copypaste-ohjelmointi, toistuvat bugit
- Jos ei ole hyvää valmista, tehdään itse
 - "Never underestimate the power of a small tactical Lisp interpreter"

Ohjelmointi = puutarhanhoitoa



Hyvät ohjelmat kasvatetaan pienestä surkeasta siemenestä, vähän kerrallaan.
Täydellisyys on tavoite, ei alkutila.

Hyvän tiimin tunnusmerkki nro 1

- Heistä on HAUSKAA tehdä töitä
 - nimenomaan tätä projektia
 - nimenomaan yhdessä
 - työt ovat hauskoja, ei kortinpeluu kahvitauoilla
- Jos hauskuus katoaa...
- Miten saada aikaan hyvä tiimi?

Joitakin väitteitä
ohjelmakoodin
kirjoittamisesta

KISS: Yksinkertainen koodi on parasta

- Ei tarvitse kommentoida tai dokumentoida
 - jos koodi tarvitsee kommentteja, jotain on väärin
- Helpompi saada toimimaan
- Helpompi korjata
- Helpompi muuttaa, kun vaatimukset muuttuvat
- Myönnetään: helppo = tylsää
 - kikkailu on kivaa
 - mutta vaikeat bugit ovat pahempia kuin pieni tylsyys

Testattavuus todella tarpeellista

- Mitä helpompi ohjelman eri osat, isot ja pienet, on testata erillään, sen paremmin se tulee testattua
 - metodi, olio, olijoukko, alijärjestelmä, koko ohjelma...
 - yksinkertaisuus auttaa
 - funktionaalisesta ohjelmoinnista hyvä ottaa vaikutteita
- Helposti testattava ohjelma myös yleensä helposti debugattava

Koodia, ei kommentteja

- Jos asian voi ilmaista koodilla, tee niin
 - `/* Length must be at least 2. */`
 - `assert(length >= 2);`
- Koodi on tarkistettavissa automaattisesti

Dataa, ei koodia

- Ei näin:

```
if (name == "Lars") id = ID_LARS;  
else if (name == "Lauri") id == ID_LAURI;  
...
```

- Vaan näin:

```
struct { char *name; int id; } tab[] = {  
    { "Lars", ID_LARS },  
    { "Lauri", ID_LAURI },  
    ....  
};
```