

Ohjelmistokehitys

Suunnitellen toteuttaen
JOT2008 Jonne Itkonen

Näkökulmia

- * Ensin suunnittelu, sitten toteutus.
- * Suunnittelu ja toteutus käsi kädessä.

Suunnittelun apuja

- * Mallit ja mallikielet
 - * suunnittelumallit (design patterns)
- * Periaatteet
- * Toisten tekeleet
- * Testilähtöinen kehitys

Mallit ja mallikielet

- * <http://hillside.net/patterns/>
- * <http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html>
- * <http://esis.pace.edu/~bergin/patterns/PatternCourseSlides.pdf>
- * <http://g.oswego.edu/dl/pd-FAQ/pd-FAQ.html>

Suunnittelumallit

- * GoF: Olio-ohjelmoinnin mikroarkkitehtuureja
- * Ei ihan näin:
 - * [http://en.wikipedia.org/wiki/Design_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))
 - * Lukekaa kirjana mielummin! (Ensin ne alkukappaleet, sitten vasta mallit.)

Mallikielet

- * <http://hillside.net/patterns/patternslanguages.htm>
- * <http://users.rcn.com/jcoplien/Patterns/Top10OrgPatterns.html>
- * <http://www.laputan.org/mud/>

Periaatteet ja refaktorointi

- * <http://www.objectmentor.com/resources/publishedArticles.html>
- * valitse "Design Principles"
- * <http://www.refactoring.com/catalog/index.html>
- * PS Nämä eivät ole malleja, vaikka sellaisia niistä voisi kirjoittaa!

Toisten tekeleet

- * ...the best way to prepare is to write programs, and to study great programs that other people have written. In my case, I went to the garbage cans at the Computer Science Center and I fished out listings of their operating system.

-Bill Gates
Read the rest in *Programmers at Work*, Susan Lammers, 1985

<http://www.cafeaulait.org/quotes2004.html>

- * "You've got to be willing to read other people's code, then write your own, then have other people review your code."
-Bill Gates

- * "I still think that one of the best tests of programming ability is to hand the programmer about 30 pages of code and see how quickly he can read through and understand it." -Bill Gates

<http://norvig.com/luv-slides.ps>

Testilähtöinen kehitys

- * Test-driven development
 - * Kirjoita ensin testi uudelle ominaisuudelle.
 - * Kehitä ominaisuus testin välillä suorittaen, kunnes testi menee kokonaan läpi.
 - * Siisti (ja refaktoroi) koodi.
 - * Toista, kunnes ohjelmisto(n osa) on valmis.

Kehityksen apuja

- * Testilähtöinen ohjelmistokehitys
- * FIT / Fitnesse, hyväksyntätestit

Validointi ja verifointi

- * Validation: Are you building the right thing?
- * Verification: Are you building the thing right?
- * Ohjelmiston ja ohjelmistotyön laadun varmistamista: katselmoinnit, testaus, laatuikäytännöt, auditointi.

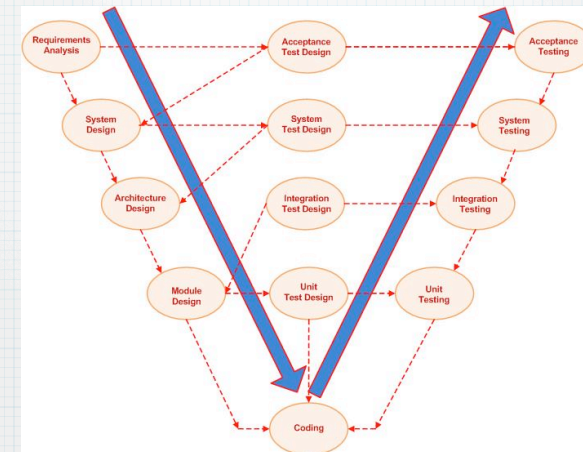
Katselmoinnit

- * Tutustutaan yhdessä kohteeseen ja kirjataan huomiot.
- * Jokainen tutustuu kohteeseen ensin yksin ja kirjaa huomionsa.
- * Käydään yhdessä kohde läpi kirjaten huomiot muistiin.
- * Asianosaiset reagoivat huomioihin.

- * Ei pohdita tai kehitetä ratkaisuja ongelmiin (huomioihin)!
- * Valmistautuminen tärkeää, jos joku on jättänyt sen väliin, katselmointi siirretään myöhemmäksi.
- * Sopii kaikenlaisille kohteille, mm. dokumentaatiolle ja lähdekoodille.

Testaus

- * http://www.bucanac.com/documents/The_V-Model.pdf



[http://en.wikipedia.org/wiki/V-Model_\(software_development\)](http://en.wikipedia.org/wiki/V-Model_(software_development))

Hyväksyntätestaus

- * Fit & Fitness
- * <http://fit.c2.com/>
- * <http://fitnessse.org/>

Yleistä testauksesta

- * Suunnitellaan testaaminen ensin.
- * Käytä testausta helpottavia työkaluja: automatisoi testaus, nauhoita testit, käytä mock-uppeja...
- * Arvosta testaajaa, vaikka hän kertoisikin ikäviä uutisia.