

Ohjelmistojen suunnittelu

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 154

Ohjelmistojen suunnittelu

- “Software design is a creative activity in which you identify software components and their relationships, based on a customer’s requirements.” –Sommerville 2010
- Suunnittelussa päätetään ohjelmiston loogisesta rakenteesta
- Rakenne kuvataan eri *abstraktiotasoilla*.
 - Mitä korkeampi abstraktiotaso, sitä selkeämmin näkyy järjestelmän yleiskuva ja sitä vähemmän näkyy yksityiskohtia
 - Korkeimmalla abstraktiotasolla on arkkitehtuurisuunnitelma, matalimmalla on pseudokoodi.

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 155

Suunnittelun luonne

- Suunnittelu on luovaa toimintaa, jota ei välttämättä kannata rajoittaa liikaa prosesseihin
- Toisaalta myös selkeästi strukturoituja suunnitteluprosesseja olemassa (mikä ei tietysti tarkoita luovuuden puuttumista)
- Ei ole helppoa suoraviivaista tapaa suunnitella ohjelmisto. Jokainen suunnittelu on ainutkertainen
- Suunnittelun lähtökohdانا voi käyttää esimerkiksi seuraavan kalvon prosessia

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 156

Strukturoitu suunnittelu

- Perustuu standardoituihin notaatioihin
 - Erilaisten graafisten mallien tuottaminen järjestelmästä
 - Toteutus seuraa luontevasti vähitellen kohti yksityiskohtia jäsenyvästä suunnittelusta
 - Useimmiten suunnittelu ja toteutus osittain rinnakkaisia

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 157

Oliopohjainen suunnitteluprosessi

- ”Ilmentymä” yleisestä suunnitteluprosessista
- Strukturoitu oliopohjainen suunnitteluprosessi pitää sisällään erilaisten järjestelmää kuvaavien suunnittelumallien kehitystyön
 - UML, esim. pakkaus-, aktiviteetti-, luokka-, olio-, sekvenssi-, kommunikaatio-, tilakaaviot
- Suunnittelumallien kehittäminen ja ylläpito vaativat paljon työtä. Pienten järjestelmien kohdalla ei välttämättä kustannustehokasta
- Suunnittelumallit tärkeä kommunikointitapa suurten järjestelmien kehitystyössä, erityisesti kun useita ryhmiä kehittämässä järjestelmää

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 158

Kevyt suunnittelu

- Ketterissä menetelmissä painotetaan suunnitteluprosessin keveyttä ja yksinkertaisuutta
- Suunnitellaan vain se mikä on aivan välttämätöntä, ei ennakoida liikoja
 - Arkkitehtuurisuunnittelu
 - (Arkkitehtuurisuunnittelun jälkeen) ei juuri erillisiä suunnitteludokumentteja
 - Suunnittelu ja toteutus toisiinsa kietoutuneina, inkrementaalisesti lyhyissä sykleissä

© Juha Taina, Marko Salmenkivi ja Kjell Lemström, 2006-2011 581259 Ohjelmistotuotanto 159

XP ja suunnittelu

- Ohjelmakoodin pitäminen selkeänä ja yksinkertaisena
- Refaktorointi
- Järjestelmän suunnittelu joustavaksi, muutosten mahdollistaminen
- Painotetaan sitä, että suunnittelukaaviot ovat *kommunikoinnin apuvälineitä*
jos kaaviot eivät palvele kommunikointia, niitä ei pidä käyttää (ja tietysti päinvastoin)
- Ohjelmakoodin ensisijaisuus myös suunnittelun liittyvän kommunikoinnin välineenä

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

160

Arkkitehtuurisuunnittelu

- Suuret järjestelmät ositetaan aina keskenään yhteistyötä tekeviksi *osajärjestelmiksi*
- *Arkkitehtuurisuunnittelu:*
 1. Osajärjestelmien määrittäminen
 2. Osajärjestelmien välisen kommunikaation suunnittelu
 - arkkitehtuurityylit
- Arkkitehtuurisuunnittelun lopputuloksena saadaan järjestelmän arkkitehtuurikuvaus
- Arkkitehtuurisuunnittelu on linkki määrittely- ja suunnitteluvaiheen välillä
- Usen toteutaan osin rinnakkain määrittelyvaiheen kanssa

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

161

Osajärjestelmät ja ei-toiminnalliset vaatimukset

- Osajärjestelmiin jako riippuu ei-toiminnallisista vaatimuksista:
 - Jos *suorituskyky* on tärkeää, kannattaa käyttää mahdollisimman pientä lukumäärää laajoja osajärjestelmiä
 - Jos *turvallisuus* on tärkeää, kannattaa käyttää sisäkkäisiä arkkitehtuureja, joista kaikkein sisimmällä tasolla ovat turvallisuuskriittisimmät toiminnot
 - Jos *saatavuus* on tärkeää, samat palvelut kannattaa toistaa monessa osajärjestelmässä
 - Jos *ylläpidettävyyden* on tärkeää, kannattaa käyttää suurta määrää pieniä osajärjestelmiä

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

162

Arkkitehtuurisuunnitelma

- Arkkitehtuurisuunnittelun lopputuloksena saadaan järjestelmän arkkitehtuurisuunnitelma
 - koko järjestelmän kuvaus
 - järjestelmästä tehdyt kaaviokuvat selityksineen
 - järjestelmän jako osajärjestelmiin ja osajärjestelmittain komponentteihin
 - osajärjestelmien välinen yhteydenpito

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

163

Osajärjestelmien ositus

- Arkkitehtuurisuunnittelun jälkeen kukin osajärjestelmä ositetaan *moduuleiksi* (module).
 - Moduuli on järjestelmän komponentti, joka käyttää muiden moduuleiden palveluja ja tarjoaa muille moduuleille palveluja
 - Moduuli ei ole itsenäinen, kuten osajärjestelmä, vaan tiiviissä yhteistyössä muiden moduulien kanssa toteuttaa osajärjestelmän toiminnot

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

164

Rajapintasuunnittelu

- Rajapintasuunnittelu on ehkä suunnittelun tärkein työvaihe
- Jokaisen osajärjestelmän tai moduulin rajapinta muihin osajärjestelmiin tai moduuleihin suunnitellaan ja dokumentoidaan yksityiskohtaisesti.
- Rajapintojen pitää olla sellaiset, että tarjottuja palveluja voidaan käyttää tietämättä niiden toteutuksesta mitään

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

165

Rajapintojen määrittely

- Rajapintojen määrittely tulee tehdä yksiselitteisiksi ja ristiriidattomiksi
- Rajapintakuvauksesta tulee näkyä:
 - Rajapinnan palvelut
 - Kunkin palvelun syötteiden ja tulosteiden tyypit ja arvoalueet
 - Kunkin palvelun kohdalta ehdot, joiden on oltava voimassa ennen palvelun kutsua ja ehdot, jotka palvelu takaa kutsun jälkeen.
- Rajapinnoissa oikea abstraktiotaso on tärkeä

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

166

Oliopohjaisen suunnittelun perustehtävät

1. Järjestelmän ympäristön ja erilaisten käyttötapojen ymmärtäminen
 - Vaatimusten ymmärtäminen
2. Järjestelmäarkkitehtuurin suunnittelu
3. Tärkeimpien olioiden tunnistus
 - Oliot voivat olla eritasoisia osajärjestelmästä palvelun tai toiminnon yksityiskohtaan.
4. Suunnittelumallien kehitys
 - Silta vaatimusmäärittelyn ja toteutuksen välillä, eri abstraktiotason mallit
5. Olioiden rajapintojen määrittely
 - Rajapintasuunnittelua

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

167

- Vaiheet eivät ole käytännössä suoraviivaisesti peräkkäisiä, vaan vaikuttavat toisiinsa
- Käytännössä ongelmia tulee esiin eri vaiheissa, jolloin on syytä palata päivittämään edellisiä vaiheita
- Luova prosessi: hahmotellaan ratkaisuja ja tarkennetaan niitä kun lisää informaatiota on käytettävissä

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

168

Olioluokkien määrittäminen

- Usein vaikea osa oliopohjaista suunnitteluprosessia, johon ei ole olemassa suoraviivaisia toimintaohjeita
- Iteratiivinen prosessi
- Sovellusalueen käsitteistö luokkarakenteen mallina
- Skenaariot: kunkin skenaarion analysoiminen ja siihen liittyvien toimintojen tarkentaminen olioiksi, attribuuteiksi ja metodeiksi

© Juha Taina, Marko
Salmenkivi ja Kjell
Lemström, 2006-2011

581259 Ohjelmistotuotanto

169