

Ohjelmistokehitys

Johdanto

Ohjelmistokehitys - Johdanto

- ☐ Ohjelmistotuotanto/-tekniikka alana
- ☐ Ohjelmistotuotannon sovellusalueet
- ☐ Prosessimallit
- ☐ Ohjelmistotuotteen elinkaari
- ☐ Ohjelmistotyö projektina: haasteet, ongelmat, erityispiirteet
- ☐ Vaatimusmäärittely
- ☐ Dokumentointi, katselmointi, testaus
- ☐ Ohjelmistotuotteen laatuun vaikuttavat seikat
- ☐ UML

Ohjelmistokehitys

Historia

- ☐ Automaattinen tietojenkäsittely (ATK) osaksi jokapäiväistä byrokratiaa 1960-70-luvulla
- ☐ Tietokonepelit yleistyvät 1980-luvulla
- ☐ Ohjelmistojen käyttö osana erilaisia laitteita – ns. sulautetut järjestelmät
- ☐ Monet hyödyntävät ohjelmistoja huomaamattaan, esim. TV kaukosäädin, matkapuhelin jne.
- ☐ Ohjelmistotekniikkaa tarvitaan kokonaisten järjestelmien ohjauksessa: Lentokone, ydinvoimala..
- ☐ Ohjelmistotekniikka on tärkeä menestystekijä myös muilla aloilla vs. Ohjelmistoteollisuus
- ☐ Ohjelmistotuotanto on tieteenala, joka käsittelee miten tietokoneohjelmia valmistetaan
- ☐ Tarkoituksena löytää ja tunnistaa ohjelmistojen valmistamiseen soveltuvia peruseriaatteita esim. parempaa laatua, ylimääräisten kustannusten karsiminen

Ohjelmistokehitys

Software Engineering

- ❑ Englanninkielinen termi "Software Engineering" suomennettu: Ohjelmistotekniikka tai ohjelmistotuotanto
- ❑ Tarkoittaa ohjelmistotyötä jonka tuloksena järjestelmät:
 - täyttävät käyttäjien kohtuulliset toiveet ja odotukset
 - valmistuvat laadittujen aikataulujen ja kustannusarvioiden puitteissa
- ❑ "Software Engineering" käsittää kaikki ohjelmiston tuotantoprosessiin liittyvät osa-alueet: laatujärjestelmä, projektinhallinta, dokumentointi, tuotteenhallinta, laadunvarmistus, testaus, määrittely, suunnittelu, toteutus, käyttöönotto ja ylläpito
- ❑ Termi "Engineering" suomennetaan: "tekniikka", tarkoitetaan tieteellisen tiedon järjestelmällistä soveltamista käytännön ongelmiin
- ❑ "Software" ohjelmistotyön tuloksena syntyvä materiaali, mm. dokumentaatio, tietokantoja ja itse ohjelmia -> suomennetaan: Ohjelmisto

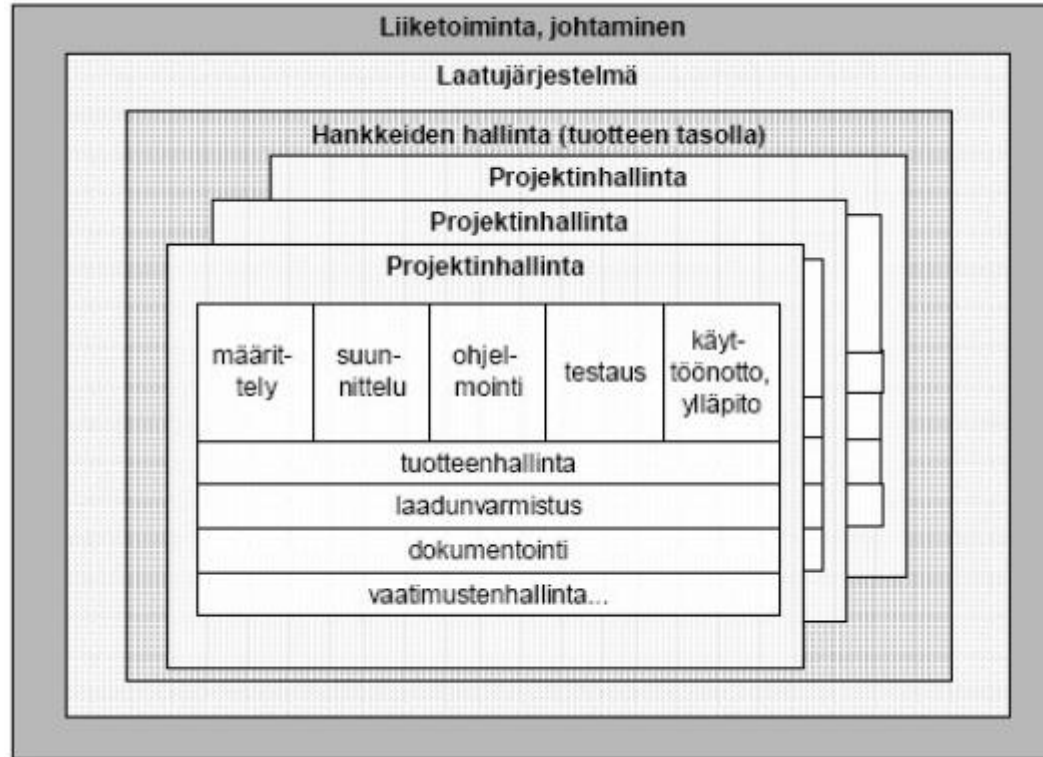
Ohjelmistokehitys

Ohjelmistojen ominaisuuksia

- ❑ Ohjelmistot voidaan jaotella ominaisuuksien mukaisesti erilaisiin kategorioihin
- ❑ Pyritään luonnehtimaan ohjelmistoa periaatteellisella tasolla -> auttaa ymmärtämään tärkeimpiä haasteita
- ❑ Kuvastetaan ohjelman luonnetta havainnollisemmin termein; esimerkkinä seuraavia ominaisuuksia:
 - Ohjelmiston koko: Miljoonan rivin ohjelma, kehittämässä satoja ihmisiä <-> tuhannen rivin ohjelma, suunnittelusta/toteutuksesta vastaa yksi ohjelmistosuunnittelija
 - Reaaliaikaisuus: Järjestelmät joiden toiminta riippuu aikaan sidotusta vaatimuksesta <-> suoritus päättyy kohtuullisen ajan kuluessa
 - Hajautus ja sulautus: Hajauttaminen johtaa rinnakkaisiin suorituksiin -> ohjelman sisäisiä ongelmia. Sulautetuissa järjestelmissä ei selvää rajaa missä laitteiston ominaisuudet loppuvat ja ohjelmistojen ominaisuudet alkavat -> toteuttaminen monimutkaisempaa
 - Luotettavuus: Mikä ohjelman tulee toipua virhetilanteista tai jatkuva häiriötön käyttö vaaditaan, voidaan tarvittaessa rakentaa varajärjestelmiä(backup) tai panostetaan järjestelmän laatuun
 - Skaalautuvuus: Järjestelmä käytettävissä erilaisissa ympäristöissä; kokoonpanot/ suorituskyky. Suunnitteluun panostettava enemmän kuin totetutettaessa yhteen kokonpanoon
 - Tuotteistus- ja standardointiaste: Esim. ohjelma joka pohjautuu aiemmin toteutettuun, hyvin testattuun järjestelmään, voi toteutuksesta selvitä pienellä työllä. Standardoituissa voidaan keventää toteutusta, saatavilla valmiita komponentteja ja ohjelmistotyössä tarvittavia työkaluja

Ohjelmistokehitys

Ohjelmistotuotannon
osa-alueet



(Haikala-Märijärvi 2002: 35)

Ohjelmistokehitys

Esimerkkijärjestelmiä

- ☐ Miten ohjelmistoja hyödynnetään erilaisia järjestelmiä rakennettaessa
- ☐ Monesti ohjelmat eroavat paljon toisistaan
- ☐ Tuottamiseen liittyvät menetelmät ja ratkaisut suurelta osin samoja

- ☐ Pieni sulautettu järjestelmä
 - Useimmiten laite jota ei tunnista tietokoneeksi
 - Voidaan ajatella että jos käyttäjä tunnistaa tietokoneeksi -> suunnitteluvirhe
 - Esim. teollisuudessa käytettävät venttiilit ja muut pienet laitteet
 - joukko ulkopuolisia yksiköitä, jotka keräävät dataa tai ohjelma toiminnallaan ohjaa
 - Keskeinen ongelma miten voidaan taata oheislaitteiden ohjaus riittävän nopeasti
 - Osa-ongelmina miten ohjelman vuorovaikutus oheislaitteiden kanssa tapahtuu esim. virran kulutus, yhteysmuoto jne
 - Pieni sulautettu vaatii vähän muistia -> ohjelmat pieniä mutta vaativia
 - Raskaat luotettavuusvaatimukset, esimerkiksi ydinvoimalan venttiilien ohjaus

Ohjelmistokehitys

Työkoneen ohjausohjelmisto

- ☐ Auto- ja koneteollisuus, usean tietokonejärjestelmän verkko
- ☐ Yksittäiset verkon komponentit ovat itsenäisiä sulautettuja järjestelmiä
- ☐ Järjestelmät toteutettu omalla tietokoneyksiköllä joka ohjaa ohjelmistoja
- ☐ Yksiköt kommunikoivat yleisesti käyttäen jonkinlaista väylää
- ☐ Työkoneiden tapauksessa väylänä jokin vikasietoinen väylä, useimmiten CAN-väylä(Controller Area Network), alun perin autojen johdotusten välttämiseksi
- ☐ Ohjelmistot kasvavat mittaviksi sillä kokonaisuudessa useita yksiköitä
- ☐ Jokaisen yksikön koordinointi haastava suunnittelutehtävä
- ☐ Järjestelmän tehtävänä sulautettujen järjestelmien tapaan laiteohjaus
 - Reaaliaikaisuus tärkeää
- ☐ Hajautus väistämätön osa tämäntyyppisessä järjestelmässä
 - Vaikeuttaa ohjelmointia eri yksiköiden tehdessä itsenäisiä päätöksiä
- ☐ Luotettavuus olennaista, koko järjestelmä käyttökelvoton ongelmien sattuessa

Ohjelmistokehitys

Laajan automaattijärjestelmän ohjelmisto

- ❑ Edellä mainittu ohjausjärjestelmä voidaan liittää osaksi suurempaa kokonaisuutta
- ❑ Tehtävänä ohjata tuotantoprosessia ja myös ohjata kaikkea siihen liittyvää toimintaa
- ❑ Integraatio jossa joukko pieniä järjestelmiä yhdistetään isommaksi kokonaisuudeksi
→ kaikki järjestelmät yhdistyvät toistensa kanssa tavalla tai toisella
- ❑ Yleistermi trendistä: järjestelmien järjestelmä (system of systems)
- ❑ Suurten integroitujen ohjelmistojen rakentamisessa tarvitaan erityyppisiä ja kokoisia ohjelmistoja
→ ominaisuudet vaihtelevat paljonkin
- ❑ Ohjelmistoista osa on pieniä itsenäisiä kokonaisuuksia
- ❑ Voi olla myös pieniä ohjelmia jotka helpottavat suuremman kokonaisuuden rakentamista
→ Tällaiselle järjestelmälle tyypisiä ns. esiinnousevat toiminnot (emergent behavior),
määrittelevät miten järjestelmä integroitu mutta kukaa ei etukäteen suunnitellut
→ Ohjelma kasvaa helposti todella monimutkaiseksi, hajautetuksi kokonaisuudeksi
→ → Toimintaa ei kukaan yksittäinen suunnittelija pysty hallitsemaan
- ❑ Hallittavuuden kriteeriksi, miten tietyt perusvaatimukset kuten ohjelmien välinen kommunikointi on suunniteltu ja toteutettu; rajapinnat eivät ole yhteensopivia → sidos- tai välikerroksia