



Mikko Holmberg  
CodeBakers Oy  
ICT-arkkitehti  
Technology Lead – Java



Perttu Karvinen  
CodeBakers Oy  
ICT-arkkitehti  
Technology Lead – Menetelmät



Joni Moilanen  
CodeBakers Oy  
IT-konsultti  
Technology Lead – Microsoft

# Kollaboraatioalustat sovelluskehityksessä – mikä muuttuu?

**Perinteisesti sovelluskehitysprosessissa työkalut ovat vastanneet yksittäisistä tehtävistä, kuten projektihallinnasta, ohjelmoinnista, mallinnuksesta, kommunikaatiosta, vaatimusten hallinnasta jne. Ongelmana ei pidetä näiden työkalujen toimintaa vaan niiden yhteistoimintaa. Ohjelmistojen hyvä yhteistoimintakaan ei nykypäivänä riitä vaan mukaan pitää saada tuki prosesseille ja kommunikoinnille. Näihin haasteisiin vastaavat uudenlaiset sovelluskehityksen kollaboraatioalustat.**

## Johdanto

Kollaboraatioalustoja on ollut tarjolla ja käytössä jo pitkän aikaa. Yleisimmin niillä tarkoitetaan alustoja, joissa toimistosovellukset on integroitu yhdeksi käyttäjäkokemukseksi ja jotka tarjoavat yhteistoimintaan sekä viestintään liittyviä ominaisuuksia kuten sähköpostit, kalenterit, tiedostojen jako, tehtävien hallinta, tavoitettavuustiedot, pika-viestit, blogit, rss-syötteet jne. Tällaisia ovat mm. IBM Lotus - ja MS Exchange/SharePoint -alustoihin pohjautuvat ratkaisut ja ne ovat nykyisin laajassa käytössä yrityksissä. Tällaiset alustat helpottavat ja tehostavat ihmisten jokapäiväistä työtä, mutta eivät tarjoa sellaisenaan tarpeeksi ominaisuuksia sovelluskehitysprojektien tarpeisiin.

Viime vuosina on puhuttu paljon ALM:sta eli Application Lifecycle Managementista. Yleisesti voidaan sanoa, että ALM-järjestelmä:

- koordinoi sovelluskehitykseen liittyviä toimintoja, mm. vaatimusten keräämistä ja analysointia, mallintamista, toteutusta ja testausta
- hallitsee sovelluskehityksen eri toiminnoissa tuotettavia väli- ja lopputuloksia sekä niiden välisiä suhteita, mm. jäljitettävyyttä
- ohjaa käyttäjän toimintaa järjestelmään konfiguroiduilla prosesseilla
- tuottaa raportteja tehtävien edistymisestä ja projektin tilasta

## Nykytilanne

Nykyisin yleisesti käytössä olevat ohjelmistot toteuttavat vain pienen osan edellä mainituista kollaboraatioalustojen ominaisuuksista. Joillakin pienemmillä toimittajilla on ratkaisuja, jotka vastaavat artikkelin kollaboraatioalustoja, mutta niiden suosio on vielä ollut marginaalista.

Ohjelmistojen yhteistoiminta on tyypillisesti saatu aikaan löyhällä integraatioilla eli toisesta ohjelmasta päästään ehkä näkemään toisilla ohjelmilla tehtyjä lopputuloksia, mutta ei kenties muuttamaan niitä. Prosessituki on yleensä vain kunkin ohjelmiston sisäistä, ohjelmistojen välille

se ei yllä. Projektinhallintaan liittyvät integroinnit ovat ehkä kaikkein huonoiten hoidettu. Projektin tehtävien hallintaa ja toteutustyötä ei ole yleensä integroitu millään tavalla.

Nykyisten järjestelmien suurin ongelma on se, että ne eivät tue jokapäiväistä työtä. Eri rooleilla ei ole näkyvyyttä toistensa tekemisiin, joten perinteisen kommunikointiin (face-to-face, sähköposti) menee paljon aikaa. Kaikille rooleille, esim. liike-toiminnan edustajalle, ei ole tarjota muuta kuin perinteisiä dokumentteja ja ehkä jotain generoituja web-sivuja yksittäisistä lopputuloksista. Eri vaiheiden lopputulosten linkitystä toisiinsa ei tueta riittävän hyvin.

Ketterien menetelmien ja toimintatapojen käyttöönotto on entisestään lisännyt tarvetta uudentaisille ohjelmistoille, jotka tukevat valittuja menetelmiä ja ohjaavat alusta alkaen tekemistä ja avointa kommunikointia. Tällaisia ohjelmistoja onkin tullut tarjolle viime vuosina ketterien menetelmien suosion kasvaessa, mutta ne ovat tyypillisesti tarkoitettu pienille yksittäisille tiimeille ja keskittyvät lähinnä ketterien tiimien toiminnan tehostamiseen.

Agile-prosessien avuksi löytyy tuotteita, jotka panostavat yhteistyön ja kommunikoinnin helpouteen, mutta eivät varsinaisesti ole sovelluskehitysvälineitä. Joitain näistä sovelluksista tarjotaan myös SaaS-palveluina. Tällaisia tuotteita ovat mm. ThoughtWorksin Mingle, Danuben Scrum Works ja Zimbra Collaboration Suite.

Seuraavaksi esittelemme kaksi erilaista ohjelmistoa, joissa edellä mainittuja ongelmakohtia on ratkaistu: IBM Rational Team Concertin (RTC) ja Microsoft Team Foundation Serverin (TFS). Molemmat ovat uusia tuotteita ja tarjoavat jo aika paljon tavoiteltuja ominaisuuksia. Toki muitakin ratkaisuja on olemassa ja tulossa markkinoille. Esimerkkinä voidaan mainita TechExcelin DevSuite.

## Ratkaisuvaihtoehtoja

### **Rational Team Concert**

IBM Rational julkaisu toukokuussa uuden Jazz-ohjelmistoalustansa ja tähän ensimmäisen sovelluksen Rational Team Concertin. RTC tarjoaa työkaluja pääasiassa sovelluskehitysprosessin implementaatiovaiheeseen. Se on lähinnä sovelluskehittäjän ja projektipäällikön tai esimerkiksi Scrum Masterin työkalu.

RTC:iin kuuluvat Eclipse-pohjainen asiakasohjelma, rajoitetumpi web-käyttöliittymä sekä itse Jazz-palvelinohjelmisto. Siihen voidaan integroida

chat-asiakasohjelmia lisäämään kollaboraatio-ominaisuuksia. Se ei itsessään tarjoa työkaluja erillisten dokumenttien hallintaan. Se sisältää myös build-moottorin ja tämän hallintatyökalut, joilla buildeja voidaan hallita ja niiden tuloksia raportoida.

RTC:n toiminta perustuu tehtävien hallintaan. Niitä voidaan osoittaa tietyille käyttäjälle ja aikatauluttaa sitomalla ne hankkeen iteraatioihin. Käyttäjät pitävät yllä tehtävien tilaa ja projektipäällikkö voi kätevästi seurata tehtävien edistymistä. Tehtävien tyypit ja iteraation rakenne riippuu projektin käytössä olevasta menetelmästä. Välineessä tuettuja menetelmiä ovat mm. Scrum ja "Eclipse way", mutta niitä voi toki luoda myös itse. Tavallisimpia tehtävien tyyppejä ovat yleinen työtehtävä, virhe, muutospyyntö ja käyttäjätarina.

Tehtävien käsittely on varsin monipuolista. Käyttäjä voi esimerkiksi liittää tehtävään koodipätkän tai dokumentin ja siirtää sen toiselle käyttäjälle. Lisäksi tehtävien etsintä on helppoa. Käyttäjä esimerkiksi näkee yhdellä silmäyksellä kaikki hänelle määrättyt tehtävät.

Monet toiminnot ovat roolipohjaisia. Esimerkiksi projektipäällikkö voi muokata iteraatiosuunnitelmaa, mutta tavallisella käyttäjältä onnistuu vain sen katsominen. Roolit riippuvat taas käytetystä menetelmästä. Tavallisimmat roolit ovat kehittäjä, projektipäällikkö sekä build-vastaava.

RTC:n käyttöönotto on suhteellisen suoraviivaista. Alkuun pääsee kolmen käyttäjän ilmaisversiolla. RTC on paketoitu ilmaisen Express-C:n lisäksi Express- ja Standard-paketteihin. Lisenssit ovat käyttäjälisenssejä, eikä ohjelmisto vaadi

Kuva 1:  
Esimerkki iteraatio-  
suunnitelmasta.

The screenshot displays the 'Iteration 1' view in the RTC application. The top bar shows the team area 'Development Project Team' and the current iteration '1.12 (13.8.2008 - 27.8.2008)' with 0 closed and 7 open items. The main content is divided into two sections: 'Top Items' and 'Defects & Enhancements'.

**Top Items:** Shows two open items. The first is 'Muuta myyntiartikkelin hintaa' (Medium priority, assigned to Kari Kehittäjä, 40 hours). The second is 'Lisää myyntiartikkeli' (Medium priority, assigned to Timo Tunari, 37 hours). Progress is 0/0 h, Estimated 100%.

**Defects & Enhancements:** Shows five open items. The first is 'Hintojen tarkistus' (Medium priority, assigned to Timo Tunari, 39 hours). The second is 'Punaisen boksen pitää vilkkua vihreänä' (Medium priority, assigned to Pauli Päällikkö, 34 hours). The third is 'Päivitä tietokanta' (Medium priority, assigned to Kari Kehittäjä, 38 hours). The fourth is 'Suunnitteluitemi' (Unassigned, assigned to Pauli Päällikkö, 36 hours). The fifth is 'Joo, ei mikään taas toimi' (High priority, assigned to Kari Kehittäjä, 35 hours). Progress is 0/116 h, Estimated 80%.

The bottom section shows a table of work items for the iteration:

| I  | Status      | P      | S      | Summary                                | Owned By        |
|----|-------------|--------|--------|----------------------------------------|-----------------|
| 39 | New         | Medium | Yellow | Hintojen tarkistus                     | Developer       |
| 38 | New         | Medium | Yellow | Päivitä tietokanta                     | Other Developer |
| 35 | In Progr... | High   | Red    | Joo, ei mikään taas toimi              | Other Developer |
| 34 | New         | Medium | Yellow | Punaisen boksen pitää vilkkua vihreänä | Project Manager |

muuta maksullisia sovelluksia kuten tietokantoja. Palvelinsovellus toimii Windows- ja Linux-ympäristöissä ja sen lisensointi on Open Source -pohjainen. Express- ja Standard-versiot voidaan asentaa myös WebSphere sovelluspalvelimelle ilmaisen Tomcat-alustan sijasta. Lisäksi ne voidaan konfiguroida käyttämään DB2- ja Oracle-tietokantoja.

### Microsoft Team Foundation Server

Microsoft Visual Studio Team System on Microsoftin vastaus ALM-markkinoille. Team System koostuu useasta osasta, jotka ovat sekä palvelin-, että asiakaspäässä. Kaiken ytimenä toimii juuri SP1-versioon päässyt Team Foundation Server 2008, johon työasemat ovat roolista riippuen yhteydessä eri asiakasohjelmilla.

Microsoftin tyyliin kokonaisuus on koottu joukosta valmistajan omia tuotteita, jotka myös integroituvat hyvin yhteen. Tästä syntyvän vahvan riippuvuuden vastapainoksi saadaan toimiva kokonaisuus, jossa projektin eri osapuolet voivat käyttää itselleen tuttuja välineitä.

Team Systemissä prosessi alkaa projektin luonnilla, mikä perustaa SharePointiin automaattisesti projektisivuston, jonne voi koostaa keskitetysti kaikki dokumentit ja muun tarvittavan oheismateriaalin. Mukana tulee myös joukko raportteja, joista näkee silmäyksellä muun muassa projektin etenemisen, bugien määrän ja korjausasteen.

Projektipäälliköt voivat määritellä tehtävät ja vastuuhenkilöt tutulla Excelillä tai Projectilla. Sovelluskehittäjät saavat kyselyä tehtävänsä

lisenssit SharePointiin, SQL Serveriin ja minimivaatimukset täyttäviin Visual Studioihin. Palvelinsovellukset voidaan hajauttaa usealle palvelimelle tai keskittää yhteen, mikäli käyttäjämäärät ovat pieniä. Gold Certified-tason Microsoft-kumppanit tosin säästävät lisenssikustannuksissa huomattavasti.

### Uudet haasteet ja mahdollisuudet

Uusien työkalujen myötä saadaan vanhoja haasteita: miten suuri joukko käyttäjiä saadaan sujuvasti käyttämään uutta työkalua tai ympäristöä? Miten koulutus järjestetään ja kuka sen hoitaa? Miten käytössä oleva prosessi taipuu uusille välineille? Kuinka kivulias siirtymävaihe ylipäättensä on? Miten päivitykset hoidetaan? Kollaboraatioalustan valinta johtaa myös jonkinlaiseen vendor-lock-tilaan.

Java-puolella alustat ovat yleensä avoimia ja laajennettavissa. Varsinkin Eclipse-pohjaiset ratkaisut kuten RTC mahdollistavat hyvät kustomointimahdollisuudet ja kolmansien osapuolien ratkaisut. Toisaalta Microsoft-maailman TFS on suljettu ympäristö, mutta siihen on olemassa rajapinnat, joiden avulla voidaan myös tarjota kolmansien osapuolien tuotteita. Edellä esitellyt ratkaisut ovat isojen ohjelmistotalojen tuotteita, joten voidaan olettaa, että niille on olemassa hyvä enterprise-tasoinen tuki.

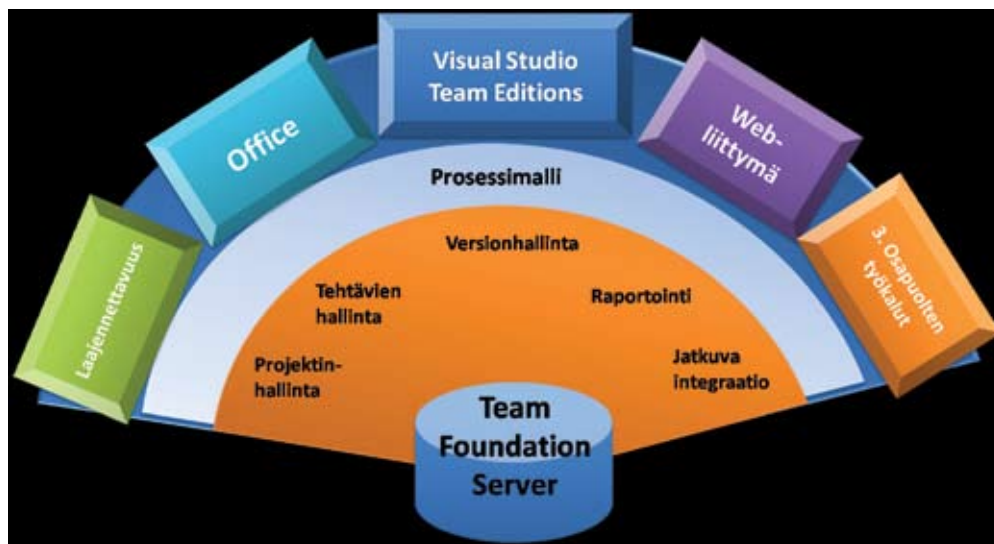
### Yhteenveto

Onko aika kypsä tällaisten kollaboraatioalustojen käyttöönotolle? Mielestämme on, teknologiamielessä ainakin. Monipuolisten kommunikointitekniikoiden ja ryhmätyöominaisuuksien lisääminen ALM-tyyppisiin sovelluskehitysympäristöihin tekee ympäristöistä niin paljon mielekkäämpiä käyttää, että on ihme ettei näitä ominaisuuksia ole yhdistetty jo aikaisemmin.

Se että kaikki sovelluskehityksessä mukana olevat henkilöt näkevät koko ajan projektin tilan ja sen kehittymisen sekä se että kommunikointi on helppoa, tehostaa tiimin toimintaa ja parantaa laatua. Tällaisia ohjelmistoja voidaan pitää myös riskienhallinnan välineinä, koska virheet huomataan nopeasti ja niihin voidaan puuttua välittömästi.

Ketterien menetelmien käyttöönotto on lisännyt ymmärrystä siitä, minkälaisia kehitysympäristöjen tulisi olla ja mitä toimintoja niiden pitäisi tukea. Kollaboraatioalustat vastaavat hyvin tähän huutoon. Yhtä asiaa kannattaa kuitenkin korostaa: varsinainen toteutustyö ja testaus eivät muutu mitenkään, vaikka käytettäisiin näitä uusia ohjelmistoja.

Lopuksi voisi todeta että mikä tahansa ympäristö on juuri niin hyvä kuin sen käyttäjät ovat eli ympäristöt eivät itsessään ratkaise yhtään ongelmaa. Toisaalta huono ympäristö voi pilata paljon.



Kuva 2.  
Team Foundation  
Server

suoraan kehitysympäristöön, josta ne on helppo löytää ja kuitata tehdyiksi. Team Foundation Serverissä on lisäksi versionhallinta sekä tuki jatkuvalle integraatiolle ja automaattisille testeille. Palvelimessa on myös paljon avoimia web service -rajapintoja, joiden avulla myös omia tai muiden ratkaisuja on helppo integroida osaksi kokonaisuutta. Tästä huolimatta kokonaisuus toimii helpoiten Microsoft-alustaan panostavilla tiimeillä.

Team Systemin käyttöönotto ei tule halvaksi, sillä se vaatii oman lisenssin lisäksi palvelimen ja