

**Pikaopas**

# **Devops Platform**

**Eficode**

# Liikkeenjohdolle

03 - 14

# Asiantuntijoille

15 - 27

# Devops Plattform

Liikkeenjohdolle

**Kiihtyvän digitaalisen kehityksen myötä reagointikyky  
innovaatioihin ja asiakkaiden tarpeisiin on noussut  
merkittäväksi menestystekijäksi.**

**Tämä edellyttää muutosta, joka saavutetaan yhdistämällä  
toimintatavat, työkalut ja ympäristöt yhdeksi hallittavaksi  
ekosysteemiksi.**



# Kaikki on koodia

Palveluiden digitalisoituminen ja automaation tuomat mahdollisuudet muovaavat yritysten toimintakenttää kiihtyvällä vauhdilla. Tuotteiden tekninen paremmuus ei ole enää merkittävä kilpailutekijä, vaan yritysten täytyy pystyä täyttämään asiakkaiden tarpeet kokonaisvaltaisemmin kuin ennen – käyttäen uusia kanavia ja toimintamalleja.

**M**uutoksen vauhti on kasvanut ja reagoitukyky asiakkaiden tarpeiden muutoksiin on noussut merkittäväksi menestystekijäksi monella toimialalla. Tätä muutosta kuvastaa se, että useat perinteistä tuote- tai laitemyyntiä harjoittaneet yritykset seuraavat viimeaikaisten kansainvälisten menestyjien kuten AirBnB:n ja Uber'in, esimerkkiä tuottaen uusia, asiakaslähtöisiä digitaalisia palveluja.

**Liiketoiminnan kyky vastata ja toteuttaa asiakkaiden muuttuvia tarpeita on suoraan riippuvainen yrityksen ohjelmisto- ja IT-osaamisen tasosta.**

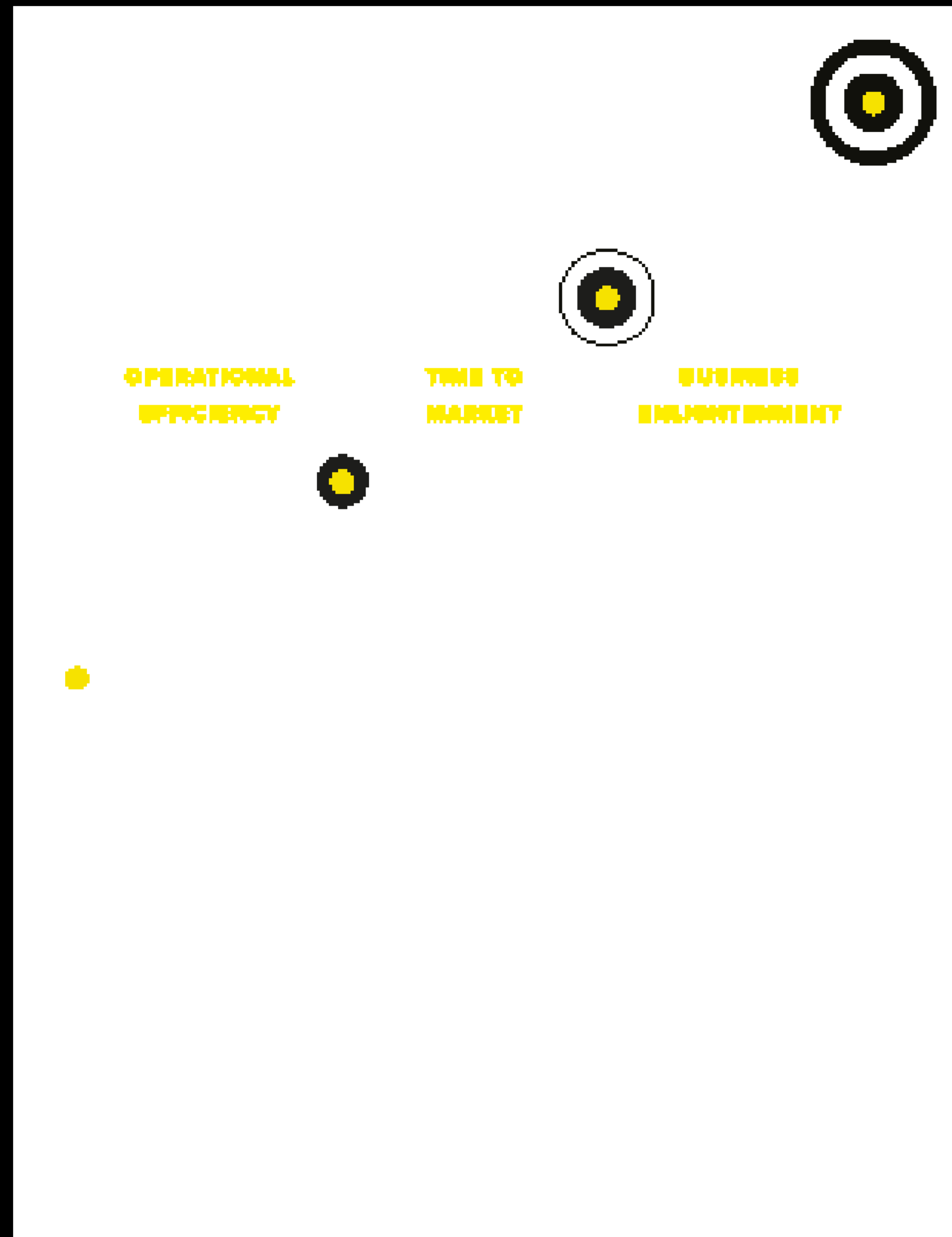
Ohjelmistokehitys ja sen järkevä toteutus on siirtynyt organisaation sisäisiä tarpeita täyttävästä toiminnosta liiketoiminnan mahdollistajaksi, joka antaa avaimet uusien kilpailukykytekijöiden kehittämiseen ja nopeaan reagointiin markkinoiden muuttuessa.

Perinteinen tuotekehitysmalli, jossa ohjelmistotuotantoa johdetaan raskailla prosesseilla ja se tehdään kokonaisvaltaisena hankkeena, on jäämässä historiaan. Tilalle on tullut malleja, joilla menestystä haluavat yritykset voivat rakentaa helposti skaalautuvia ohjelmistoja ja mikropalveluita jotka keskittyvät toteuttamaan rajattuja vastuita kevyesti ja tehokkaasti. Tämä mahdollistaa uusien liiketoimintamallien ja ideoiden viemisen markkinoille kevyellä rakenteella nopeasti ja samalla voidaan validoida niiden toimivuus ja kannattavuus loppuasiakkailla.

Tämän uuden mallin perustana on ohjelmistotuotannon, IT-hallinnan ja liiketoiminnan saumaton yhteistyö. Devops-toiminta tähtää tämän yhteistoiminnan rakentamiseen ja jatkuvaan kehittämiseen. **Sen tavoitteena on mahdollistaa ketterä ja laadukas ohjelmistotuotanto, joka vastaa asiakkaiden tarpeita ja mahdollistaa automatisoidun toimintamallin.** Devops-viitekehys on kokoelma työkaluja ja toimintatapoja joita hyödyntämällä yritykset saavat reaaliaikaisen kuvan liiketoiminnastaan. Se näyttää miten asiakkaat kohtaavat yrityksen tuotteet ja miten yrityksen eri toiminnot näistä kohtaamisista syntyviin tarpeisiin pystyy vastaamaan. Tämä kokonaisvaltainen toimintamalli mahdollistaa ongelmakohtien helpon identifioinnin sekä auttaa liiketoimintaa päätöksenteossa ja menestyvän strategian luomisessa. **Datalla johdettu yritys on valmiimpi ja nopeampi vastaamaan kilpailukentän tuomiin muutoksiin ja luomaan täysin uusia ansaintalogiikoita eri arvoketjuissa ja niiden osissa.** Tämän mallin perustana on, että kaikki mahdollinen saatavilla

oleva informaatio yrityksen omasta toiminnasta ja asiakkaiden käyttäytymisestä kerätään ja analysoidaan jatkuvana toimintana päätöksentekoa varten.

Yritysten pitää tottua ajatukseen siitä, että olemassa olevan liiketoiminnan tehostaminen ja uuden luominen on jatkuva prosessi jota voidaan kehittää. Parhaimmat toimijat yritys kentässä omaksuvat mallit ja käytännöt jotka mahdollistavat jatkuvan kehittymisen, ja haluavat, että jatkuvasta kehittämisen toiminnasta on tullut osa yrityksen kulttuuriperimää. On tärkeää huomata, että jatkuvaan kokeiluun perustuva, asiakaslähtöinen palvelukehitys määrittelee ne työkalut, joilla asiakkaiden ongelmat saadaan ratkaistua. **Moderni ja menestyvä yritys kokeilee ja ottaa käyttöön uusia teknologioita ja malleja jatkuvasti ja samalla hylkää nopeasti ne, jotka eivät tuo asiakkaalle riittävää lisäarvoa. ■**

DEVOPS-MATKADEVOPS-MATKA

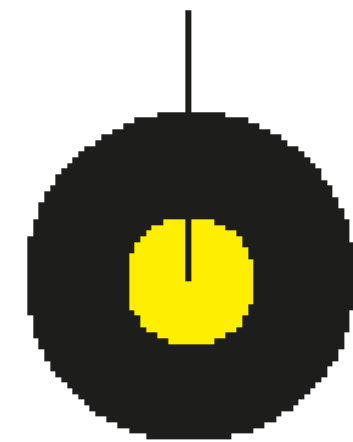
Liiketoiminnan vaatimukset ketterämpään ja hajautetumpaan digitaaliseen palvelukehitykseen aiheuttaa haasteita IT-toiminnoille.

Ympäristöistä tulee monimutkaisia kokonaisuuksia. Palvelujen, tuotteiden ja käyttäjien määrä lisääntyy ja samalla käytävissä olevien työkalujen valikoima monipuolistuu. Näiden vaatimusten toteuttaminen ja muutoksen läpivieminen on toteutettava hallitusti.

Jatkuvaan ja tehokkaaseen Devopsiin pohjautuvaan digitaaliseen liiketoimintaan on kolme vaihetta: julkaisu (deliver), seuranta (track) ja ohjaus (drive).

LEVEL 1

**CORE**



**DELIVER**

**Operational Efficiency**



LEVEL 1

Toiminnallisen tehokkuuden tasolla yritys standardoi käytettävän koodin, siihen liittyvän dokumentaation ja työn hallinnan, yhdistäen ne keskitettyjen käyttäjätunnuksien taakse. Tarpeelliset kehityskäytännöt pitää harmonisoida ja niitä myös valvotaan eri työkalujen välillä automaattisesti.

Tämä luo vankan pohjan seuraavalle vaiheelle Devops-matkalla ja toiminnan mielekäs tehostaminen muuttuu mahdolliseksi.

Tässä vaiheessa yksittäisen tuotteen kehityksen seuranta on tyypillisesti hyvin projektikohtaista, jolloin historiallisen informaation haku perustuu pitkälti omaan aktiivisuuteen.



DELIVER

Ensimmäinen kehitysaskel lähtee tuotekehitykseen käytettävien työkalujen harmonisoinnista yhdeksi hallittavaksi kokonaisuudeksi, eli Devops-ympäristöksi. Tämä tarkoituksena on oikeanlaisen informaation kerääminen, jakaminen ja sen toteuttaminen yksittäisen kehittäjän lisäksi kaikille tahoille, joille se on tarpeellista.

Kun tuotekehityksen eri vaiheissa käytettävät työkalut on yhdistetty tehokkaasti toisiinsa, voidaan parhaiten seurata myös julkaisuun vaadittavien tarpeiden toteutumista käytännössä. Tällä vältetään kehityksen ohjaaminen vain yksittäisten henkilöiden välisellä kommunikaatiolla. Samoin julkaisua varten toteutettavien ominaisuuksien valmistumista ja kehitykseen kuluva aikaa voidaan seurata luotettavasti Devops-ympäristöön kerätystä tiedosta hallitusti. **Työkalujen harmonisoinnilla saavutetaan toiminnallinen tehokkuus (operational efficiency), joka mahdollistaa uusien tuotteiden tai päivitysten julkaisemisen ennustettavasti ja luotettavasti.**

**Edut**

Johtavassa roolissa toimiville kuvattu ympäristö mahdollistaa kyvyn hallita käyttäjiä eri työkaluissa keskitetysti. Tästä seuraa, että **projektin ajalliset ja rahalliset aloituskustannukset, henkilöstövaihdokset ja muut vastaavat kulut on helppo minimoida**. Kun projektit noudattavat harmonisoitua työtapaa, ei projektissa tuotetun lisäarvon löytäminen, hallitseminen ja seuraaminen ole enää haasteena.

Kehittäjät ja testaajat pystyvät keskittymään olennaiseen paremmin, kun toimintatapa on selkeä ja looginen – projektista riippumatta. He saavat jatkuvasti palautetta omasta työstään testi-, build- ja deploy-automaatiosta, pystyen siten ohjaamaan työtään tuottavammaksi. Automaatio tuo lisää tehokkuutta ja hoitaa vakioidut välivaiheet, jolloin ohjelmistoprojektin laatua on myös helpompi valvoa ja parantaa. Laatupoikkeaman paljastuessa on helppoa palauttaa edellinen versio tuotantoon automaattisesti.

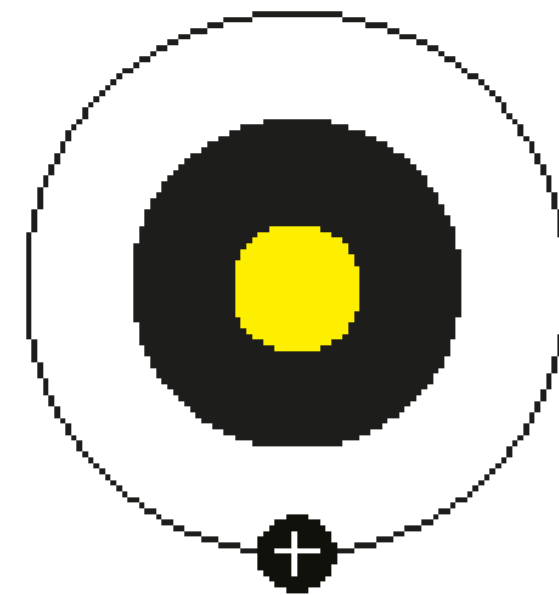
CASE**Valtion laitos**

Asiakkaamme hyödyntää liiketoiminnassaan runsaasti erilaisia pitkälle erikoistuneita ohjelmistoja, joiden kehitys on ulkoistettu yhteistyökumppaneille. **Halutessaan toimia tehokkaammin ja ketterämmin, he huomasivat, että houkuttelevien digitaalisten palveluiden jatkuva julkaisu vaatii, että eri projektien välillä ei ole suuria eroavaisuuksia perustyökaluissa.**

Vauhdittaakseen ohjelmistokehityksen siirtymistä Devops-kulttuuriin, yritys tarjoaa jatkossa yhteistyökumppaneilleen alustan joka kannustaa kumppaneita kehittämään työmenetelmiään yhteensopiviksi ja mahdollistaa toiminnan tehokkuuden ja automatisoinnin. Alusta tarjoaa Devops-menetelmien mukaiset työkalut ja käyttäjähallinnan yhtiön kaikille projekteille.

Asiakkaalle itselleen ympäristö tarjoaa mahdollisuuden ohjata ohjelmistokehitysmenetelmien kehitystä organisaatiossaan ja seurata projektien tuottamaa lisäarvoa helposti. Ympäristö mahdollistaa paremman näkyvyyden projektien seurannassa ja tuo kustannussäästöjä hajautettuihin työkaluihin verratuna. Yritys haluaa palveluiden omistajana ja tilaajana panostaa ohjelmistokehityksen läpinäkyvyyteen ja laatuun. Uusi alusta mahdollistaa tämän läpi koko organisaation ja hankintaketjun.

LEVEL 2



**TRACK**

Time-to-Market

LEVEL 2

Nopeampaan läpimenoaikaan pyrkiessään yritys liittää julkaisuprosessiinsa myös binäärien hallinnan, jolla mahdollistetaan useiden kymmenien ympäristöjen hallinta samanaikaisesti.

Kehityksestä ja tuotannosta kerättävä tieto on reaaliaikaista ja eri mittausmenetelmiä käyttäen päästään toteuttamaan liiketoimintaan liittyvää kehitystä joka parantaa operatiivista toimintaa.

Kehityksen nopeuden kasvaessa myös kommunikaatiota tehostetaan.



TRACK

Toisessa vaiheessa muodostetaan organisaation ydinliiketoiminnan onnistumista ja riskienhallintaa kuvaavia mittareita, joiden pohjana toimii Devops-ympäristöön kerätty tieto. Toiminnan mittaamisessa siirrytään käsin kerätyistä raporteista reaaliaikaisen informaation hyödyntämiseen. Koska Devops-ympäristö toimii laadun ja julkaisun tukijalkana, siitä kerättävä informaatio kertoo täsmällisesti toiminnan tehokkuuden.

Mukaan tuodaan myös uusia työkaluja, joilla laajempien ja monimutkaisempien kokonaisuuksien hallinta helpottuu. Tällöin pystytään automatisoimaan myös useista pienemmistä järjestelmistä koostuvia jatkuvaan julkaisuun tähtäviä palveluita. **Tämä tarkoittaa liiketoiminnan kannalta nopeampaa reagointikykyä markkinoiden ja asiakkaiden tarpeiden muutoksiin, eli nopeampaa läpimenoaikaa (time-to-market).**

**Edut**

Liiketoiminnan edustajalle kuvattu ympäristö tarjoaa työkalut, joilla mitataan kehitystä, tuotantoa ja projektin loppukäyttäjien reaktioita. **Tätä tietoa voidaan eri mittausmenetelmiä hyödyntäen käyttää toiminnan parantamisessa ja ohjaamisessa.**

Kehittäjät ja testaajat puolestaan pystyvät hallitsemaan laajempia ohjelmistokokonaisuuksia ja keskittymään monimutkaisten build- ja deploy-prosessien sijaan uuden kehittämiseen. Reaaliaikainen palaute auttaa heitä kohdentamaan työtään paremmin ja tarjoaa mahdollisuuden kehittyä työssään.

**Tuotteiden laatua ja turvallisuutta voidaan parantaa, kun tehdyt mittaustavat kertovat reaaliaikaisesti mitä ohjelmistossa oikeasti tapahtuu ja missä seuraava mahdollinen ongelma tai kehityskohta on.**

CASE**Pohjoismainen pankki**

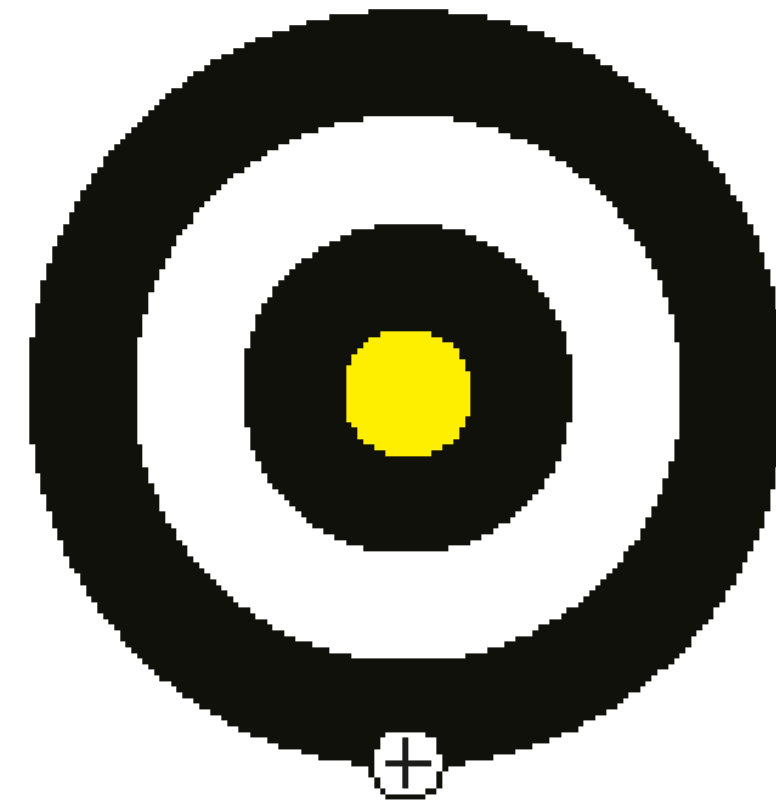
Asiakkaamme on tunnistanut, että uuden palvelun menestys on usein riippuvainen siitä, kuinka nopeasti ja laadukkaasti ensimmäinen tuoteversio saadaan loppukäyttäjille validoitavaksi.

**Mitä aikaisemmin julkaisu voidaan tehdä – erityisesti kehityksen vielä ollessa alkuvaiheessa – sitä paremmin kehityksen suuntaa voidaan ohjata ja epäonnistumisen riski vähenee.**

Tästä syystä kaikki Java-pohjaisen ohjelmistotuotannon osat aina koodin toimittamisesta laadunvarmistukseen ja dokumentaatioon toteutetaan kaikille osapuolille yhteisessä ympäristössä. Koska yli 3000 ohjelmoijaa eri maantieteellisillä alueilla käyttää ohjelmistotoimituksen ympäristöä yhtä aikaa, on palvelun toimittava saumattomasti ja tehokkaasti. Ympäristö sisältää SVN- ja Git -versionhallintapalvelut, vaatimustenhallintajärjestelmän, dokumentaatio- ja identiteetinhallintajärjestelmän sekä CD-palvelimet ja artifaktivarastot. Kaikki yhtenä kokonaisuutena.

Keskitetty ympäristö mahdollistaa projektien työtapojen ripeän kehittämisen ja automatisoinnin lisäämisen. Tämä tarkoittaa myös sitä, että monien erilaisten integraatioiden ja automatisointien käyttöönotto voidaan toteuttaa ilman, että asialla tarvitsee vaivata ylläpitoa. Samalla ympäristö tarjoaa hyvän näkyvyyden projekteihin, nopeat palautekierrokset ja tuo joustavuutta kehitysprosesseihin. Ympäristö on käytettävissä 99,95 prosenttia ajasta, kellonajasta ja viikonpäivästä riippumatta.

LEVEL 3



# DRIVE

Business Enlightenment



LEVEL 3

Uutta liiketoimintaa innovoidakseen yrityksen ohjelmisto- ja IT-tuotanto virtualisoi käytössä olevat ympäristöt, palvelimien asennuksien ollessa täysin automaattista. Yritys on kehittänyt IT-infrastruktuurinsa sille tasolle, että se toimii kokeilulähtöisen kulttuurin (experiment-driven culture) perusteilla.

Tällä tasolla oleva yritys on yleensä johtavassa markkina-asemassa ja luo uusia ekosysteemejä, valtaa uusia markkinoita ja mullistaa alalla vallitsevia toimintatapoja ja -malleja. Voidaan sanoa, että yritys on saavuttanut liiketoiminnallisen valaistumisen (business enlightenment).

DRIVE

Yrityksellä on nyt mahdollisuus kokeilla viimeisimpiä teknologioita ja luoda uutta liiketoimintaa. Tämä voi olla esimerkiksi uusien pilvipalveluiden käyttöönottoa tai Big Dataan perustuvia ratkaisuja. Ohjelmiston infrastruktuurin virtualisointi esimerkiksi konttien avulla mahdollistaa automaattisesti uudelleen luotavat ympäristöt ja antaa yritykselle mahdollisuuden nousta asiakkaan palvelemisessa johtavien toimijoiden kuten Googlen tai Amazonin tasolle.

**Edut**

Harmonisoitu, keskitetty ympäristöjen ja työkalujen hallinta takaa ennakoitavissa olevat kustannukset, luvattujen palvelutasosopimusten pitämisen, sisäisten prosessien kehityksen ja resursoinnin helpottumisen. Liiketoiminnan päättäjät voivat keskittyä asiakkaan tarpeisiin.

Kehittäjillä on puolestaan käytössä kaikki tarpeellinen varustus ratkaista kaikki tekniset ongelmat tehokkaasti: koodikatselmukset, automaattinen testaus, staattinen koodin analysointi, sekä välittömästi käyttöönotettavat kehitysympäristöt helpottavat kehitystyötä, kun rutiinitehtävät on automatisoitu. Samoista eduista nauttivat myös testaajat. He voivat luoda testiympäristöjä tarpeen mukaan itsenäisesti ja pystyvät standardisoimaan hyväksyntä-, integraatio-, turvallisuus- sekä suorituskykytestauksen ja automaattisen raportoinnin. Kun testaajat päivittävät tarvitsemansa ympäristön, mahdollistetaan parempi ohjelmistojen laatu ja parannetaan sekä yhteisymmärrystä että kommunikaatiota kehittäjien ja testaajien välillä.

Projektin kokonaisvaltainen testaus lisää laadun lisäksi myös sen turvallisuutta. Ohjelmistojen tietoturvan kohdalla siirrytään haavoittuvuuksien ja niiden vakavuuksien arvioimisesta ennaltaehkäisemään niitä hyvillä toimintatavoilla. Koodianalytiikka, automatisoidut deployment- prosessit, sekä mahdollisuus ajaa automaattisia tietoturvatestejä mahdollistavat sovitun tietoturvatason saavuttamisen. Virtualisoidut, uudelleenluotavat ympäristöt tarkoittavat, että eri alihankkijoiden hallitseminen on mahdollista, vähentäen toimittajaloukkuja.

CASE**Case Sinä?**

Kuinka monta edellä mainittua, globaalilla tasolla toimivaa yritystä Suomessa on? Yksi merkittävimmistä tekijöistä on kykenemättömyys tutustua ja adoptoida esiin nousevia teknologioita jonka avulla voitaisiin luoda uutta liiketoimintaa – kuten isot pelurit tekevät. Tämä johtuu siitä, että elämme vielä ajassa, jolloin IT- ja ohjelmistokehitys ovat erillisiä osastoja siinä missä muutkin yksiköt. Tämä siloutuminen estää tehokasta kommunikaatiota ja uudenaikaista innovaatiota.

Uusien palveluiden ja tuotteiden julkaisuaika markkinoille mitataan viikoissa, jotta voidaan validoida niiden toimivuus loppukäyttäjillä ja asiakkailla. Tämän jälkeen palautteen perusteella ohjataan tuotteen jatkokehitystä ja priorisoidaan uusien ominaisuuksien julkaisu. Tämä ei onnistu, jos odotetaan osastojen välisiä palaverieita. Sen sijaan jokainen projektitiimi pitäisi luoda siten, että projektihenkilöstöön kuuluvat kaikkien osa-alueiden edustajat (cross-functional teams). Projektitiimin jäseniksi otetaan tarpeiden esiin tullessa lisää kompetenssia tai sitä vähennetään kun jotain erikoisosaamista ei enää tarvita dynaamisesti skaalaten.

Oleellisena osana edellä kuvattua projektien mahdollistamista on IT-infrastruktuuri. Kolmansien osapuolien palvelujen hyödyntäminen nousee keskiöön, jotta aikaa ei tarvitse käyttää oman laiteinvestointien kanssa puljaamiseen. Samalla on kuitenkin varmistuttava siitä, että eri alojen regulaatiot ja lakien tuomat vaatimukset täytetään.

RATKAISU?

Uusien liiketoimintamallien toteuttaminen, asiakkaiden palvelumallien muuttuminen ja ekosysteemien rakentaminen vaatii IT-järjestelmien harmonisointia saumattomiksi. Yhtenäinen, järkevä ja hallittava kokonaisuus on välttämätöntä yrityksen kilpailukyvyn kannalta. Yritysten on tehtävä päätöksiä: miten tämä toteutetaan ja kuka valitaan kumppaniksi?

**Eficode Root tarjoaa loistavan ratkaisun yrityksille, jotka haluavat keskittyä oman liiketoiminnan kehittämiseen ja partneroitua alan johtavan toimijan kanssa.** Ulkoistamisella päästään vaikuttamaan kustannuksiin vähentävästi uusimpien välineiden ja toimintatapojen käyttöönoton kautta sekä tarjoamaan liiketoiminnalle mahdollisuuden lisätä liikevaihtoa uusien palveluiden muodossa nopeammin ja luotettavammin. Ratkaisumme tarjoaa kaikki modernit työkalut yhdessä paikassa ja räätälöintimahdollisuuden tilaajalle sopivaksi. Voit tilata Eficode Rootin joko omana yksityisenä pilvipalveluna avaimet-käteen-pakettina tai vaihtoehtoisesti yrityksesi palomuurin taakse. Toimitusaika on tyyppillisesti noin 4-6 viikkoa.

Yritysten toimintatavat, kulttuurit ja organisaatorakenteet eroavat toisistaan ja niiden tehtävänä on vastata asiakkaiden tarpeisiin parhaalla mahdollisella tavalla. Tämän vuoksi on tärkeää, että keskitetty työkalujen ja ympäristöjen hallinta tuodaan organisaatioon hallitusti sen toimintatavat huomioon ottaen. **On suositeltavaa tehdä Devops-ympäristön käyttöönotto ensiksi vain valituissa tuote- tai palveluosioissa riippuen organisaation vastaanottokyvystä.** Eficode Rootin toiminnallisuus on rakennettu siten, että eri tasoille siirtyminen on luontevaa kehitystä, organisaation lähtötaso huomioiden. Tämä mahdollistaa aina ajan tasalla olevan seurannan Devops-käytäntöjen jalkautumisesta organisaatioon.

**Liiketoimintajohto pääsee heti käyttöönoton jälkeen seuraamaan oleellisia parametrejä valmiilta työpöydältä, joka kertoo tärkeimmät tiedot ja numerot yksittäisistä projekteista koko niiden kehitysajalta.** Samalla jokaisen projektin hallittu siirtyminen Devops-malliseen kehitykseen on seurattavissa. Työpöytä näkymästä löytyy kaikki projekteihin liittyvä tieto helposti ja vaivattomasti, jolloin kehityksen seuranta helpottuu. **Lopputuloksena yrityksesi omat ja ulkoistetut ohjelmistokehittäjät voivat keskittyä parhaan mahdollisen koodin toteuttamiseen ja liiketoiminta uusien palvelujen kehittämiseen.**

Eficode Rootin yhtenä tavoitteena on vapauttaa pitkällä aikavälillä resurssit liiketoiminnallisesti kriittisiin infrastruktuuri- sekä ohjelmistoprojekteihin, sekä tarjota päättäjille parhaan mahdollisen näkymän ja ennustettavuuden ohjelmistokehityksen eri vaiheisiin. Tämän lisäksi Root on rakennettu modulaariseksi, jotta sitä voidaan edelleen kehittää uusia työkaluja lisäämällä ja muokata vaivattomasti yritysten tarpeita vastaavaksi. **Testatut ja toimiviksi todetut teknologia- ja työkaluvalinnat takaavat joustavan, toimittajariippumattoman ympäristön, joka vastaa yrityksen tarpeisiin nyt ja tulevaisuudessa.**

Eficode on sitoutunut Rootin jatkuvaan kehittämiseen ja uskomme, että asiakasyrityksemme saavat palvelustamme liiketoiminnallista ja mitattavaa hyötyä, niin kustannusnäkökulmasta kuin liiketoiminnan kehittämisen kannalta. **Devops ja Eficode Root kehittyvät yhdessä tuoden yrityksille kyvyn vastata toimintaympäristön jatkuviin muutoksiin, vähentää kustannuksia ja tuoda uutta liikevaihtoa.**

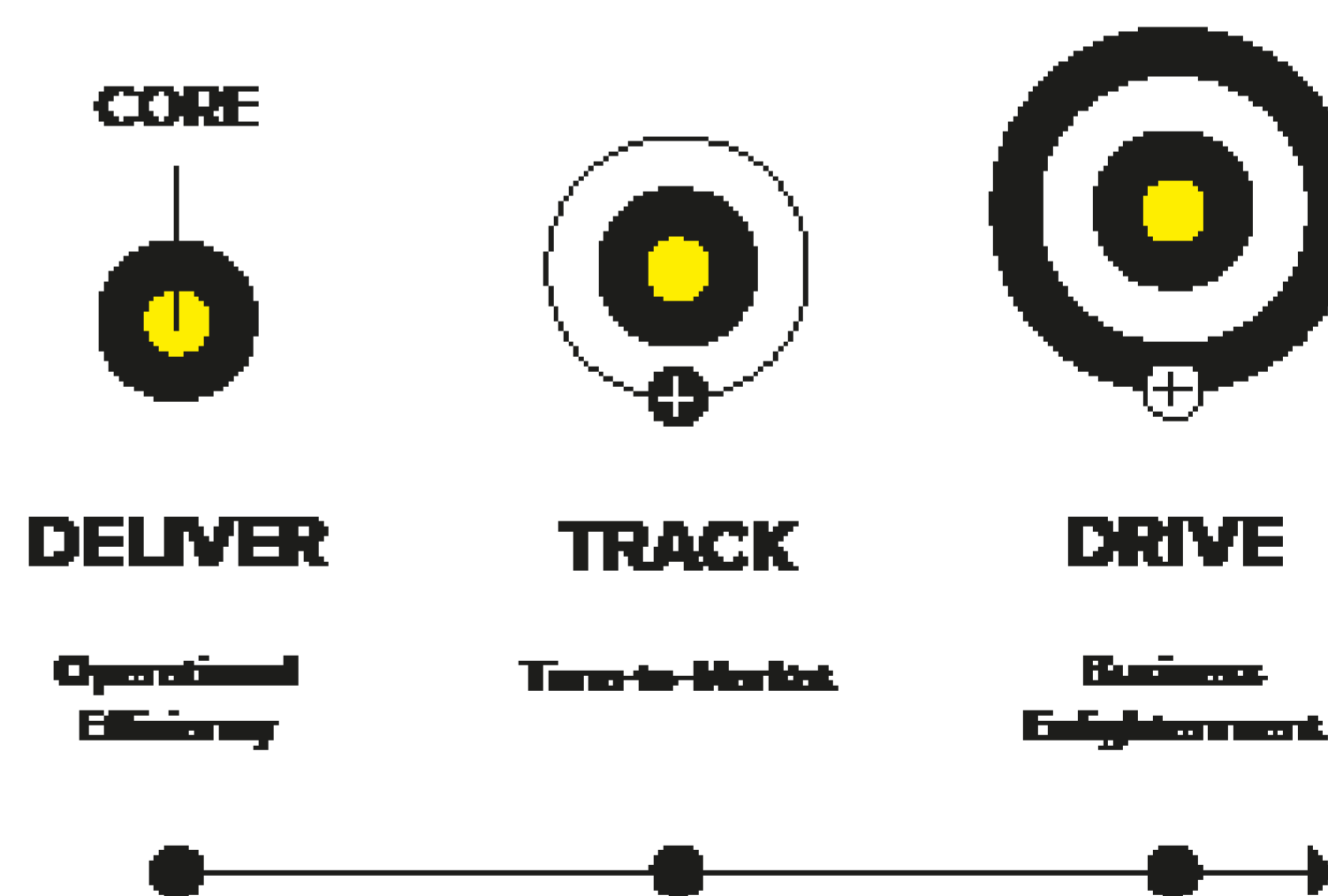
# Devops Plattform

Asiantuntijoille

ASiantuntijoille

Modulaarinen, skaalautuva ja ennen kaikkea tarpeenmukainen.

Hyvä Devops-ympäristö koostuu tasoista, jotka vastaavat Devops-matkan kolmea vaihetta: Deliver, Track & Drive





LEVEL 1 - TOIMINNALLINEN TEHOKKUUSLEVEL 1 - TOIMINNALLINEN TEHOKKUUS

Ensimmäisellä tasolla ohjelmistokehitys stabiloidaan ja harmonisoidaan kaikkien projektien välillä. Näin otetaan projektien

IPR:t haltuun sekä luodaan yhtenäiset toimintatavat, jotka perustuvat automaatioon.

Samalla sekä dokumentaation että työtehtävien hallinta yhtenäistetään, jolla varmistetaan ymmärrys siitä mitä on tehty ja mitä tullaan tekemään

Kaikki edellä mainitut työkalut yhdistetään käyttäjähallinnalla niin, että projektin jäsenet pääsevät yksillä käyttäjätunnuksilla kaikkialle. Työkalut myös integroidaan niin, että siirtyminen projektin eri tietojen välillä on saumatonta myös työkalujen välillä.

### Keskitetty käyttäjähallinta

Kaikkien työkalujen ja järjestelmien tulisi toimia yksillä käyttäjäkohtaisilla tunnuksilla. Näin mahdollistetaan tehokkaampi käyttöaste sekä helppo, ryhmäperustainen käyttäjähallinta. Ryhmäperustaisella käyttäjähallinnalla käyttäjät saavat oikeudet järjestelmiin ilman, että oikeuksia pitäisi erikseen jokaisen käyttäjän kohdalla määritellä jokaiseen palveluun. Kokonaisuusien hallinta selkeytyy ja muutoksia henkilöstössä on helpompi kontrolloida. Ryhmäperustainen käyttäjähallinta kannattaa aloittaa määrittelemällä ensiksi kolme peruseriä: järjestelmäylläpitäjät, kehittäjät ja pelkästään lukuoikeudet omaavat.

**Ilman helppoa ryhmäperustaista, keskitettyä käyttäjähallintaa on vaikea tarjota tarpeeksi joustava ympäristö muun Devopsin mukaisen kulttuurin mahdollistamiseksi.**

### Versionhallinta

Versionhallinta takaa yritykselle kestävästä kehityksestä, kun kaikki ohjelmistokoodi löytyy yhdestä järjestelmästä. **Yrityksen omistuksessa olevat IPR:t pysyvät yrityksensä itsensä kontrollissa** ja versionhallinta mahdollistaa aina helpon paluun mihin tahansa edelliseen versioon. Lisäksi versionhallinta mahdollistaa yhtäaikaisen tekemisen tiimissä ilman vaaraa, että jonkun tiimiläisen tekemä ohjelmointityö katoaisi.

Jokaisessa versionhallintatyökalussa on omat tapansa hallita ohjelmistokoodia. Tästä johtuen on oleellista ymmärtää, miten työkalussa käsitellään loogisia kokonaisuuksia – esimerkiksi käsitelläänkö kokonaista uutta toiminnallisuutta, vai yhtä koodimuutosta? Työkalua tarkoituksenmukaisesti käyttämällä tehtyjen muutoksien löytäminen on helppoa myöhemmin. Versionhallintatyökaluissa on myös omat tapansa hallita tilanne, jossa kaksi kehittäjää ovat muuttaneet samaa tiedostoa, helpottaen tällaisten konfliktien ratkaisua.

Toimiva versionhallinta tuo mukanaan kyvyn määritellä hyödyllisiä prosesseja, esimerkiksi koodikatselmuksiin ja julkaisunhallintaan liittyviä hyviä tapoja. Versionhallinnan on oltava tietoturvallinen ja pitää huoli siitä, että kehittäjien työ ei vahingossakaan huku.

### Työn hallinta

Työn seuraamisen ja ohjaamisen kulmakivi on kyky nähdä, analysoida ja hallita työtä. Paras lähestymistapa tähän on luoda yksittäisistä työtehtävistä kirjauksia sellaiseen järjestelmään, jolla voidaan hallita suurempia kokonaisuuksia sekä eri tavoin visualisoida esimerkiksi toteutuneita ja keskeneräisiä työtehtäviä jonkin ajanjakson aikana. **Tuottavuus paranee, kun voidaan järjestelmällisesti tunnistaa pullonkauloja ja kohdistaa resursseja sinne, missä edistyminen on kriittistä muiden, rinnakkaisten työtehtävien edistymisen suhteen.** Hallitsemalla työtehtävien välisiä riippuvuuksia voidaan optimoida tehdyn työn määrä ja minimoida se aika, jonka työntekijät käyttävät asioiden odottamiseen.

Työnhallintajärjestelmän tulisi taipua yleismaailmallisen luonteensa takia moneen asiaan. Sen avulla pitäisi onnistua niin ohjelmistokehityksen kuin sekä IT-hankintojen seuraaminen. Kaikesta kerätystä informaatiosta pitäisi voida tuottaa ymmärrettävää raportointia niille, jotka sellaista tarvitsevat. Kun yleiskuva työstä on nähtävillä, saadaan uusi näkökulma siihen miten työnteko yllipäättään sujuu ja millä tavoin sitä voitaisiin parantaa.

Työnhallintajärjestelmät tulisi tallentaa myös selkeä jälki siitä mitä on sovittu ja miten asian kanssa on edetty. Mustaa on vaikea väittää valkoiseksi, kun on selkeästi nähtävillä historia siitä miten asiat sovitulla etenemistavoilla ovat oikeasti kehittyneet.

### Dokumentaation hallinta

Hyvä dokumentointijärjestelmä tarjoaa keskitetyn sijainnin, johon voidaan tallettaa kaikki merkityksellinen tieto. Se tarjoaa valmiit, muokattavissa olevat pohjat toistaiseen tiedon luomiselle sekä mahdollistaa helpon käyttöoikeuksien hallinnan ja projektikohtaisen tiedon kapseloinnin. Jokaisen uuden projektin yhteydessä pitää olla helppoa luoda projektikohtainen työtila tiedon tallentamiseen ja jokaisen yksittäisen projektin tulee pystyä eväämään pääsy mahdollisesti salaiseen tietoon projektin ulkopuolisilta henkilöiltä. Kokousmuistiinpanot sekä kalenterimerkintöjen tekeminen ja niiden jakaminen tietyn käyttäjäryhmän sisäisesti on tärkeä osa tiimien sisäistä interaktiota. Järjestelmän pitäisi myös sopia sellaisenkin tiedon tallentamiseen, joka on kaikille merkityksellistä – esimerkiksi ohjeistus yrityksen ulkoiseen viestintään. **Dokumentointijärjestelmän tulee siis olla kanava sekä tiimien sisäiselle että organisaation laajuiselle viestinnälle.**

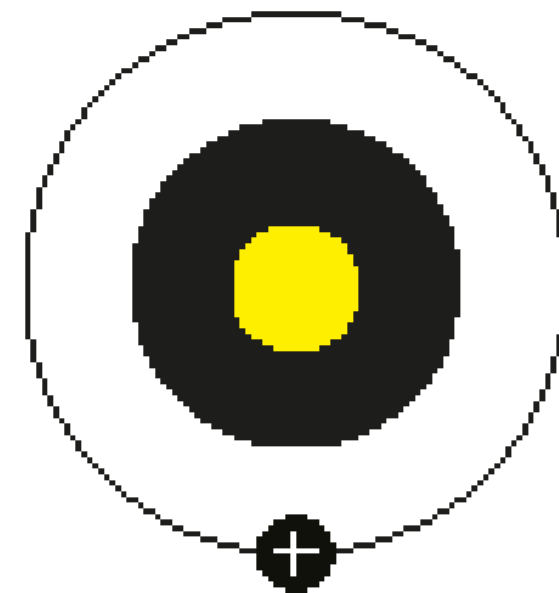
Dokumentointijärjestelmää pitäisi myös voida helposti käyttää suunnitteluun ja viestintään organisaation ulkopuolisten työntekijöiden kanssa. Tässäkin on huomattavaa, että organisaation sisäiseksi tarkoitettu tieto pysyy nimenomaan organisaation sisäisenä.

### Jatkuva julkaisu

Jatkuva integraatio (Continuous Integration, CI) on nykyään kehittynyt yhä laajemmaksi käsitteeksi tuoden mukanaan tuen myös jatkuvalle julkaisulle. Tästä seuraa uusia haasteita kehitykselle, kun ei enää pyritä standardoimaan vain kehityksen build- ja testausvaiheita, vaan automaation on yletyttävä myös valmiin ohjelmistopakettien säilöntään, ympäristöjen provisiointiin ja tuotantojulkaisuun. Kehityssykliden lyheneminen liiketoiminnan tarpeiden johdosta pakottaa kehityksen automatisoimaan testauksen, buildauksen ja julkaisun mahdollisimman pitkälle.

Jatkuvan julkaisun järjestelmien kehitys on oltava jatkuvaa ja pystyttävä vastaamaan kehityksessä esiin nouseviin vaatimuksiin. Tämä vaatii alati kehittyvää kykyä julkaista ohjelmisto useisiin eri ympäristöihin. Esimerkiksi webissä toimivilta palveluilta oletetaan nykyään toimivuutta eri selainversioilla, kukin jokaisella käyttöjärjestelmällä. Toinen jatkuvan julkaisun kasvava trendi on skaalautuvat build- ja deploy-ympäristöt, jotka määrittellään koodin avulla dynaamisesti aikaisemman manuaalisen, etukäteen tapahtuvan asennustyön sijaan.

Hyvä jatkuvan julkaisun järjestelmä tukee yrityksen sen hetkistä tarvetta ja kehittyy tarpeen mukaan. Kun jatkuva julkaisu jää jälkeen siitä mikä on tarve, ei voida edellyttää modernia, ketterää ohjelmistokehitystä.



# TRACK

Time-to-Market



Tämä taso rakentuu edellisen varaan, tuoden kasvavaa kykyä kommunikoida ja seurata kehityksen tilaa.

Projektien välinen harmonisointi jatkuu ja ohjelmistokehityksen repertuaariin tuodaan mukaan reaaliaikaista analytiikka ja niistä jalostettua metriikkaa.

Analytiikasta ja metriikasta voidaan johtaa projektien välistä vertailua, mahdollistaen hyvien toimintatapojen jalkauttamisen koko yrityksen käyttöön.

### Binäärienhallinta

Projekteissa syntyy erinäisiä tallennusta tarvitsevia tiedostoja. On järkevää, että myös nämä hallitaan keskitetysti, jotta versiointi olisi helppoa. Tällä tavalla sama binääri voidaan viedä testistä tuotantoon asti vaikka vuosien jälkeen. Myös uudelleenkäytettävät kirjastot, joita tarvitaan useissa projekteissa, on helppo säilöä, versioida ja jakaa muun organisaation käyttöön. **Binäärienhallinta on versionhallinnan lisäksi keskiössä, kun puhutaan IPR:ien säilyttämisestä.**

Binäärihallintajärjestelmää valitessa tulee ottaa huomioon tuetut tiedostotyypit ja ohjelmistoprojekteissa käytetyt teknologiat, jotta ikäviä yllätyksiä ei synny myöhemmin. Lisäksi tulee huomioida, kuinka paljon levytilaa tälle järjestelmälle varataan ja miten sen varmuuskopiointi toteutetaan.

### Sisäinen viestintä

Sähköposti viestivälineenä on tänä päivänä turhauttavan hidas. Sisäinen viestintä on syytä järjestää pikaviestinjärjestelmällä (Instant Messaging, IM) kun tuotetta kehitetään nopealla tahdilla – vastauksien pitää olla saatavilla minuuteissa, ei tunneissa. On oleellista, että **kaikki projektin sidosryhmät ovat samassa järjestelmässä aina projektin johdosta alihankkijoihin, jotta kommunikaatio on nopeata ja informaatio kulkee ketterästi.**

Toimivassa, modernissa ohjelmistokehityksessä on oleellista, että erilaiset toimijat bisneksen, myynnin ja IT:n välillä voivat keskustella vaatimuksista ja toiminnallisuuksista reaaliajassa. Jatkuvan integraation ja julkaisun järjestelmät pitää voida integroida pikaviestimiin tuottamaan kaikille jaettua, ajantasaista tietoa automatiikan tilasta.

### Operationaalinen analytiikka

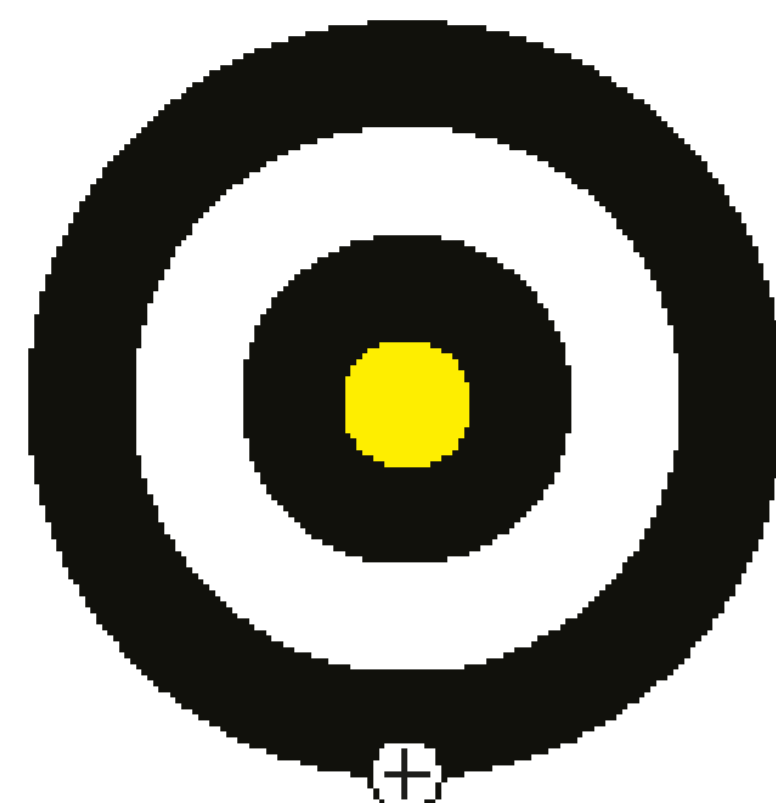
Tuotannossa olevien projektien määrä kasvaa koko ajan ja näin ollen tarve monitoroida niitä lisääntyy. On oleellista pystyä reaaliaikaisesti tietämään, että kaikki palvelut ovat pystyssä ja välittömiä ongelmia ei ole havaittu. Lisäksi monitorointijärjestelmä tarjoaa mahdollisuuden koostaa raportteja koko palvelun elinajalta, minkä perusteella voidaan tehdä liiketoiminnallisia päätöksiä esimerkiksi siitä, mitä seuraavaksi projektissa tulisi refaktoroida.

**Toimiva monitorointi ei vain huomaa tilanteita kun ne ilmenevät, vaan osaa varoittaa jo etukäteen.** Näin tilanteisiin voidaan varautua ja huoltokatkot voidaan tehdä hallitusti toisin kuin reagointiin perustuvassa toiminnassa. Toimiva monitorointi on kivuton ottaa käyttöön, tarjoaa perusmetriikat ja niihin pohjautuvan raportoinnin automaattisesti, sekä tukee laajentamista helposti

### Kehitysanalytiikka

Kehityksen aikana on kiinnostavaa nähdä, miten ohjelmisto täyttää eri laatustandardit. Lisäksi on pystyttävä vertailemaan eri projektitiimejä keskenään, jotta voidaan löytää mahdolliset pullonkaulat sekä ne parannuskohteet, joilla kaikkien projektien sisäistä laatua nostetaan. Kehitysanalytiikan työ on tukea ja auttaa kehittäjiä heidän päivittäisessä työssään. Näin kehittäjät voivat auttaa itseään kehittymään kehittäjinä.

Staattinen koodianalyysi tuottaa tietoa esimerkiksi siitä miten monimutkaista koodi on, miten yleisiä kehityskonventioita on noudatettu ja missä kohdissa ohjelmistoa on potentiaalisia ongelmia tai haavoittuvuuksia. Liittämällä koodin staattinen analyysi osaksi jatkuvaa julkaisua, saadaan jatkuvasti arvokasta tietoa koodin laadusta ja voidaan seurata, miten laatu muuttuu versioiden välillä. Analyysivaiheeseen voidaan asettaa automaattisia laatuportteja, jotka esimerkiksi pysäyttävät julkaisun mikäli testikattavuus katsotaan liian alhaiseksi. **Kehityksen ohjaaminen oikeaan suuntaan ja kustannussäästöt ovat ilmeisiä, kun ongelmat huomataan jo ennen kuin ne vaativat korjausta.**



# DRIVE

Business Enlightenment



Tällä tasolla tarvitaan tapa tarjota ympäristöjä, jotka toimivat koko Internetin skaalalla ja silti hallitusti. Tärkeää on valita oikeat teknologiat ja integroida jatkuva julkaisu dynaamisiin ympäristöihin.

Esimerkiksi pilvipalveluiden käyttö kehityksessä ja testauksessa tuo tehoa ohjelmistokehitykseen, oli pilvi sitten yrityksen sisäinen tai kolmannen osapuolen palvelu.

Nyt voidaan tarjota sekä näkyvyys että ketterä tapa julkaista ohjelmistoja “on web scale”.

### Service Desk

Tarjottujen IT-palvelujen hallinnan pitää pystyä tänä päivänä tarjoamaan reaaliaikaisesti tietoa eri toimituksien tilasta ja mahdollisuudet raportoida sekä analysoida kustannuksia. IT-palvelujen hankintaprosessit pitää pystyä sisällyttämään itse ohjelmistoon, jotta niiden tilaaminen ja käsittely ei ole enää altis inhimillisille virheille.

Edellä mainitun lisäksi esiin nousee käyttäjien oikeuksien rajoittaminen varsinkin kun IT-palvelujen toimittajana on alihankkija. Palvelutasosopimusten toteutumista pitää pystyä seuraamaan alihankkijakohtaisesti. Alihankkijat saattavat myös haluta integroida omat järjestelmänsä Service Desk-ratkaisuun samalla kun on tarve liittää se muihin sisäisiin järjestelmiin. Tästä syystä integroitavuus on yksi Service Desk-ratkaisun tärkeimpiä ominaisuuksia

Service Desk-järjestelmää voidaan käyttää myös yhtenä jatkuvan palautteen kanavista ohjelmistokehityksessä erityisesti tilanteissa joissa yrityksellä on oma ohjelmistotuote, jolloin Service Desk-järjestelmä voi toimia muun ohella myös asiakaspalvelujärjestelmänä. Devopsin mukainen koko organisaation läpinäkyvyys on helppo saavuttaa kun kehittäjät saavat silmäyksen tuotteen elämään loppukäyttäjien käytössä: he pystyvät tarvittaessa suoraan ottamaan yhteyttä loppukäyttäjään, ymmärtävät liiketoiminnallista priorisointia enemmän ja kykenevät paikallistamaan vikatilanteita helpommin.

Service Desk-järjestelmän valinta on useasti haastavaa, koska eri liiketoiminnoilla ja teknisillä toimijoilla on omat toiveensa palvelun osalta. Ei riitä, että Service Desk toimii ja sitä on helppo käyttää, sen pitää tukea myös oikeanlaista raportointia. **Service Desk-järjestelmän päämääränä ei ole vaikeuttaa työtä vaan parantaa tarjottujen IT-palvelujen asiakastytyvyyttä.**

### Pilvistrategia

Pilvipalvelujen hyödyntäminen omassa liiketoiminnassa on tällä hetkellä monen yrityksen mielenkiinnon kohteena. Suurimmaksi ongelmaksi muodostuu se, miten tämä saavutetaan mahdollisimman tehokkaasti ja turvallisesti. Statiikan kerääminen, omien tuotteiden automaattinen julkaisu sekä testaaminen pilvipalveluiden yhteydessä monimutkaistuvat, jolloin pitää varmistua, että aikaisemmin saavutettu reaaliaikainen kokonaiskuva kehityksen tilasta säilytetään.

On huomattava, että pilvipalvelut ovat nopeasti muodostumassa perustavanlaatuisiksi osaksi moderneja kehitys- ja teknologiavalintoja – esimerkkinä palvelittomat arkkitehtuurit, big data tai machine learning – joista saatavaa hyötyä ei voida valjastaa yrityksen liiketoiminnan käyttöön ilman jonkinlaista ratkaisua.

Pilviteknologioiden käyttöönotto yrityksessä vaatii yritykseltä harkittua pilvistrategiaa.

Hyvän strategian keskeisimmät tekijät ovat kustannusten ennustaminen, turvallisuusvaatimuksista huolehtiminen erityisesti lakeihin ja viranomaisiin liittyen, yrityksessä olemassa olevien käyttökohteiden kartoitus, etukäteen nähtävissä olevat tarpeet resurssien elastisuudelle ja skaalautuvuudelle, nykyinen markkinoiden palvelutarjoama sekä palveluiden monitorointimahdollisuudet. Edellä esitetyt tulisi muotoilla roadmapiksi, jonka jälkeen konkreettisen strategian muodostaminen on mahdollista. On huomioitavaa, että pilvistrategian muodostamisessa ketterä, iteroiva kokeilu on elinehto.

### Virtualisointi

Virtualisointi on tapa käyttää useita eri käyttöjärjestelmiä yhdellä palvelimella. Tämä mahdollistaa esimerkiksi Linux-ympäristöjen suorittamisen Windows-palvelimella. Virtualisointi on avainasemassa, kun halutaan toteuttaa skaalautuvia ja uudelleenluotavia ympäristöjä. Virtualisointi mahdollistaa valmiiden pohjien (template) luomisen, jolloin niiden kopioiminen ja luominen ohjelmallisesti osana jatkuvaa julkaisua on mahdollista. Tällä tavalla voidaan harmonisoida ympäristöt eri ohjelmistoprojektien välillä ja luoda ympäristöjä nopeasti lennossa.

Virtualisointi mahdollistaa kehityksen suoraan kohdekäyttöjärjestelmälle ilman palvelimien tilausta, odotusta ja asennusta. Testaaminen helpottuu, kun käyttöjärjestelmä on täsmälleen sama kehittäjän koneelta tuotantoon asti. Myös varmuuskopiointi helpottuu, koska esimerkiksi virtuaalikoneyökalut mahdollistavat snapshotien ottamisen koko järjestelmästä, jolloin korkea saatavuus (high availability) ja hyvä järjestelmäpalautus (disaster recovery) ovat helposti saavutettavissa. **Virtualisointi on vaatimus, kun halutaan tarjota yrityksen sisäistä pilvi-infrastruktuuria.**

### Uudelleenluotavat ympäristöt

Ohjelmistojen hallitseminen erilaisissa ympäristöissä on tällä hetkellä ehkä suurin IT-haaste. Palvelimien asennus on tällä hetkellä vielä pitkälti manuaalista työtä ja suurimpia kustannuseriä IT-organisaatioissa. **Uudet teknologiat mahdollistavat viimein ympäristöjen hallinnan sillä tasolla, että voidaan ottaa käyttöön uusia, parempia toimintatapoja.**

Oli kyseessä sitten kevyet kontit (containers), pilviympäristöt tai muut virtualisointivaihtoehdot, on tärkeää pyrkiä tarjoamaan ympäristöjä ketterästi niitä tarvitseville. On ymmärrettävä, että tarve ei ole vain kehittäjillä, vaan ympäristöjä tarvitsevat myös liiketoiminta, testaajat, myyntihenkilöstö

tai tuotantojärjestelmistä vastaavat. Myyntitilanteessa tehtävä demo ohjelmistoprojektista on tyypillinen esimerkki tilanteesta, jossa tarvitaan oma ympäristö verrattain lyhyeksi ajaksi.

Paras lähestymistapa onkin yhdistää ympäristöjen asennus osaksi olemassa olevaa julkaisujärjestelmää. Näin käyttäjä voi julkaista itsenäisesti tuotteen uuteen ympäristöön dynaamisesti, automatiikan vapauttaen resurssit kun ympäristöä ei enää tarvita.

### Kontit (containers)

Kontit ovat verrattain uusi teknologia, jonka avulla voidaan tarjota tarjota kokonaisia ohjelmistojärjestelmiä omina, eristettyinä paketteinaan. Kontit voidaan mieltää virtuaalikoneiksi, jotka kuitenkin vievät huomattavasti vähemmän kovalevytilaa ja suoritintehoa kuin kokonainen virtuaalinen käyttöjärjestelmä.

Koska virtuaalikoneen hyödyt saadaan entistä vähemmillä resursseilla, voidaan palveluja skaalata huomattavasti paremmin. **Erityisesti nykyaikaisia mikropalveluarkkitehtuurin (microservice architecture) mukaisia palveluja ei kannata nykypäivänä rakentaa ilman jonkinlaista kontitusta.** Tämän lisäksi kehittäjät voivat kehittää projektiaan omilla työläppäreillään täsmälleen samanlaisessa ympäristössä, kuin mitä on tuotannossa ja testauksessakin, näin vähentäen ympäristöistä johtuvia bugeja. Myös järjestelmien välisiä riippuvuuksia voidaan hallita helpommin, sillä kontit on binääreihin verrattuna helpompi versioida.



### Big Data

Olemassa olevan informaation määrä on valtaisa ja sen volyymi tulee entisestään kasvamaan tulevaisuudessa. Tiedon hallinta, talletus ja analysointi on tällä hetkellä ehkä suurin hyödyntämätön mahdollisuus yrityksissä, erityisesti kun puhutaan miten sitä voitaisiin käyttää hyväksi jo ohjelmiston kehitysvaiheessa.

On huomattava, että erilaiset tiedonlähteet vaativat erilaiset tallennusmuodot. Big Datan yhteydessä, jos ei jo aikaisemmin, on tarve käsitellä niin rakenteellista tietoa perinteisillä SQL-tietokannoilla, kuin myös ei-rakenteellista tietoa NoSQL-tietokannoilla. Tulevaisuudessa tietokantaratkaisuja tulee varmasti lisää – ilman tukea näille menetetään selkeitä mahdollisuuksia ymmärtää liiketoimintaan vaikuttavia tekijöitä. Vastaavasti on huomioitava, kuinka nopeasti eri informaatiota tarvitaan. Kaiken informaation ei tarvitse olla reaaliaikaista, jopa kuukausittain saatava riittää joissakin tapauksissa. Kuitenkin **on yrityksen etu, että datan keräystä kehitetään ja "tuunataan" jatkuvasti, jotta suurimpia mahdollisuuksia ei ohiteta vahingossa.**

Yleensä etuna, jota ei huomioida Big Datan yhteydessä, on lakisäädösten noudattamisen varmistaminen ja valvominen. Kun käyttäjä haluaa tietää, mitä informaatiota hänestä kerätään, pitää pystyä tietämään kaikkien käyttämistään järjestelmistä – mielellään tämä kaikki tieto on aggregoitu eri palveluista yhdeksi käteväksi reaaliaikaiseksi tietokannaksi, josta on helppo toimittaa raportti. Big Datan avulla voidaan myös hajallaan olevaa tietoa koostaa ja analysoida markkinoinnin, jatkokehityksen ja toiminnan tehostamisen tarpeisiin. Esimerkiksi markkinointi voi datan avulla ymmärtää, miksi käyttäjät valitsevat tai eivät valitse tuotetta kilpaileviin tuotteisiin verrattuna. Sama koostettu informaatio kertoo myös, miten käyttäjät tuotetta käyttävät.

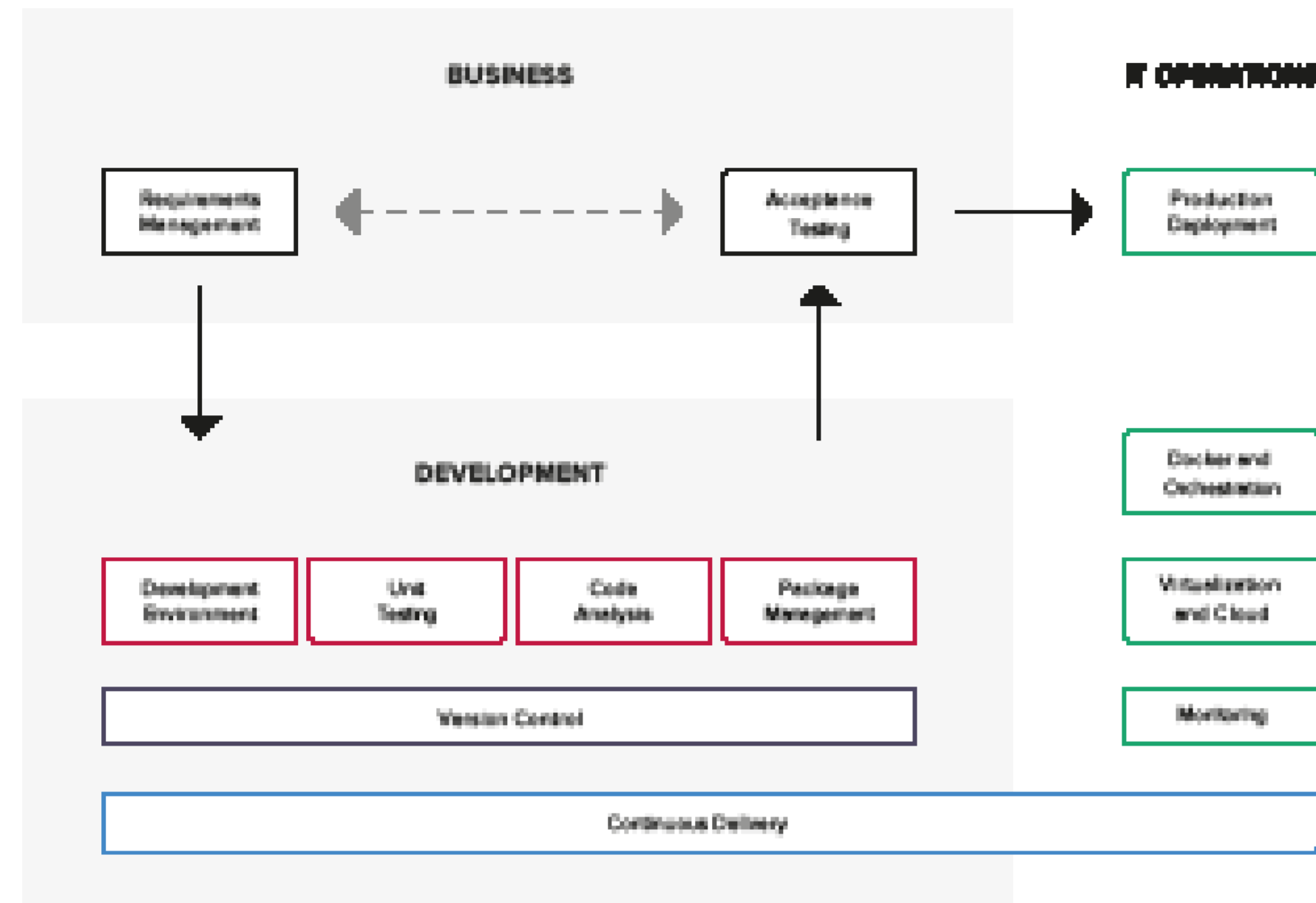
### Testidatan hallinta

**Lähitulevaisuudessa testidatan hallinta (Test Data Management, TDM) tulee monille entistä tärkeämmäksi testaamisen osa-alueeksi muun muassa EU:n GDPR-vaatimusten (General Data Protection Regulation) myötä.** On pystyttävä todistamaan, mitä yrityksen ohjelmistojen keräämä data on ja missä sitä säilytetään. Tämä ei koske vain tuotantoympäristöjä, vaan myös kaikkia muita ympäristöjä, mitä ohjelmistokehityksessä on käytetty. Kehitys-, testaus-, staging- ja muista ympäristöistä on osoitettava, että niissä käytetty esimerkiksi data ei ole suora tuotantokopio, vaan henkilön yksilöivä tieto on korvattu sellaisella generoidulla datalla, joka silti mahdollistaa mielekkään testaamisen.

Koska keskiverto ohjelmisto kerää erittäin paljon edellä mainittujen GDPR-vaatimusten alaista tietoa, ei eri tietokantojen välillä siirrettävää dataa pystytä enää järkevästi käsittelemään muutamien tietokantaspecialistien käsityönä, vaan heillä on oltava työkalut, jotka hoitavat perustoimenpiteet – kuten datan maskaamisen (masking) – automaattisesti. Näin specialistit pystyvät keskittymään datan hallitsemiseen ja automaattisen raportoinnin rakentamiseen, jotta vaivalloisia viranomaisdokumenteja ei tarvitse luoda käsin.

Testaajille ja kehittäjille on oleellista pystyä testaamaan kehitettävää ohjelmistoa oikealla datalla, sillä tietyt ohjelmistovirheet, joita esiintyvät vain loppukäyttäjien käytettyä järjestelmää, liittyvät usein kehityksenaikaisiin oletuksiin. Samaan aikaan on pidettävä huolta edellä mainituista regulaatioista, joten datan pitää näyttää oikealta, muttei olla yksilöivää. Lisäksi kokonaista tuotantotietokantakopiota on vaikea siirrellä ympäristöstä toiseen, kun tietokannan koko on useita kymmeniä gigatavuja tai paljon enemmän. Hyvä testidatan hallinta mahdollistaa tietokantakopion analysoinnin ja korvaamisen osittain generoidulla, mutta oikeanmuotoisella, datalla automaattisesti. Analysoinnin tuloksena on helpompi koostaa pienempiä, mutta oikeaa dataa edustavia tietokantakopioita, joita on helppo siirrellä eri ympäristöjen välillä.

TULEVAISUUS



Ohjelmistokehitys on alati muuttuva ja kehittyvä ympäristö. On tärkeää kulkea tämän kehityksen kärjessä tarjoten uusimmat, mutta silti taistelukoetut teknologiat ohjelmisto- ja IT:n kehityksen tueksi.

Esimerkiksi uudet lakisääteiset vaatimukset tulee integroida sujuvasti osaksi kehitystä, sillä muuten niitä ei voida jälkikäteen korjata ilman suuri kustannuksia. Myös tietoturva- ja suorituskykytestaus ovat sellaisia osa-alueita, joihin vasta nyt on tulossa hyviä automaatiotyökaluja. Myös virtualisoinnin, erityisesti konttien, kehittyessä yhä laajempia kokonaisuuksia voidaan hallita automaation keinoin.

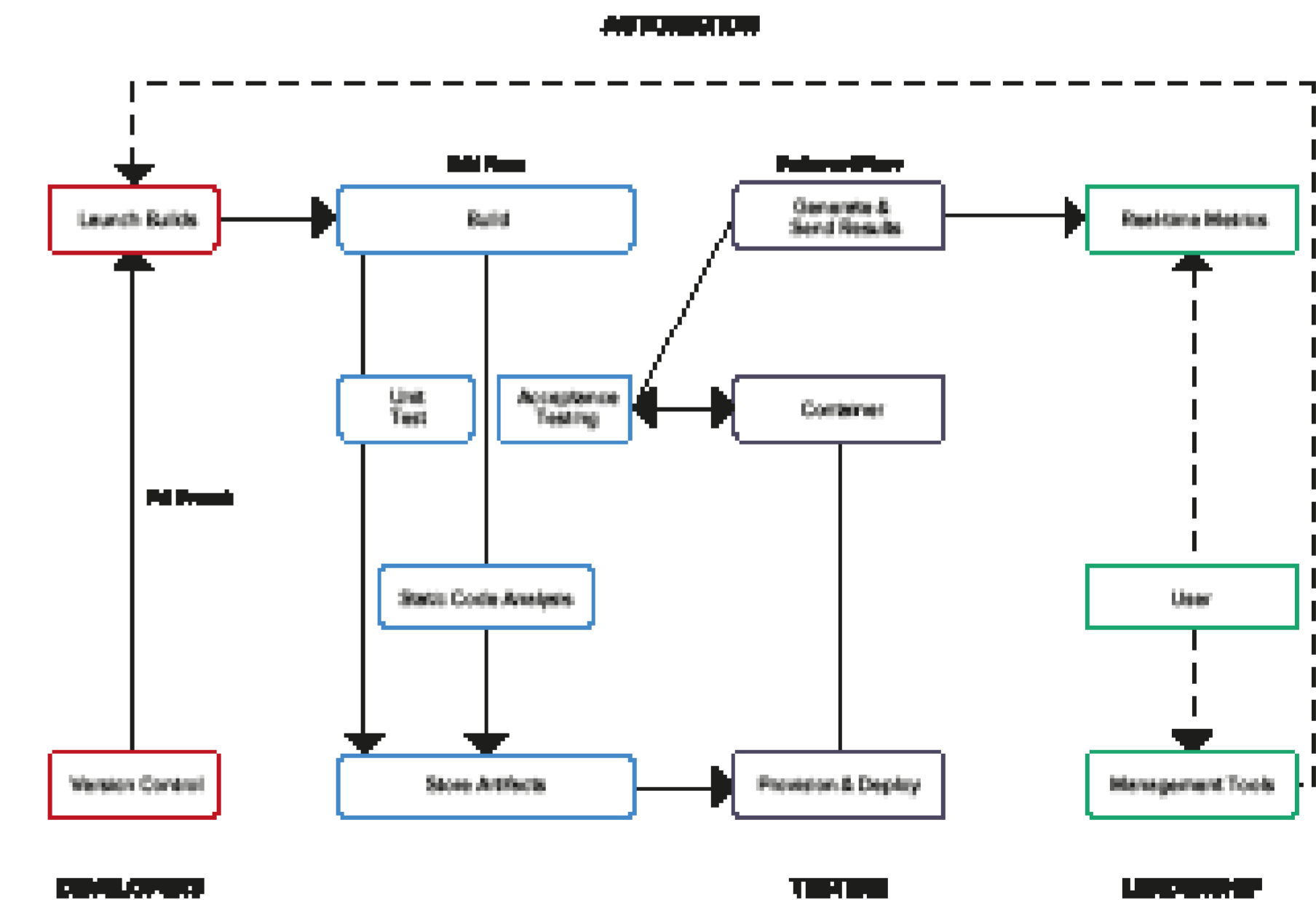
Uusien teknologioiden hyödyntäminen, prosessien parantaminen ja resurssoinin helpottaminen ei kuitenkaan koskaan lopu. Kun Devops-tyylinen ajattelumaailma on jalkautettu kokonaisvaltaisesti, on jatkuva kehittyminen ja oppiminen osa yrityksen DNA:ta.

MITEN SAAN DEVOPS-YMPÄRISTÖN?

Vaikka tämän oppaan kuvaama ympäristö on mahdollista toteuttaa itse, onko se enää yrityksesi ydinliiketoimintaa?

Eficode Root on ulkoistettu palvelu, joka tarjoaa edellä kuvatun mukaisen Devops-ympäristön ennustettavilla kuluilla sekä maan parhaalla asiakastuella.

Se tarjoaa kaikki modernit työkalut yhdessä paikassa, sekä räätälöinti-mahdollisuuden tilaajalle sopivaksi – keskity oleelliseen työkalujen sijaan.



EFICODE ROOTIN KÄYTTÖÖNOTTO ON HELPPOA

01

**ASENNUS**

Root:n asennukseen on kaksi vaihtoehtoa. Ensimmäinen vaihtoehto on yritykselle yksityinen pilviratkaisu, jolloin Root on käytettävissä neljässä viikossa. Toinen vaihtoehto on sijoittaa Root pyörimään yrityksen omassa IT-infrasassa, jolloin varaamme hieman lisää aikaa asennusvaiheeseen.

Tässä vaiheessa valitaan pilottivaiheessa käyttöön otettavat perustyökalut. Työkaluvalikoimaa voidaan kasvattaa myöhemmin tarpeiden kehittyessä.

02

**PILOTOINTI**

Pilotointivaiheessa valitut työkalut asennetaan, validoidaan ja konfiguroidaan siten, että ne tukevat yrityksen nykyistä ja tahtotilan mukaista toimintaa. Samalla koulutamme joukon avainkäyttäjiä yrityksesi henkilöstöstä. Koulutuksen jälkeen ensimmäiset projektit voivat siirtyä Rootiin. Aloitamme myös viikoittaiset palaverit, jossa pilotoinnin etenemistä seurataan.

Tästä vaiheesta siirrytään seuraavaan vasta, kun Rootiin siirrettyihin ensimmäisiin projekteihin ollaan täysin tyytyväisiä.

03

**JATKKEHITYS**

Jatkokehitysvaiheessa loput yrityksen projektit siirtyvät itsenäisesti käyttämään Rootia. Siirtymistä tuetaan tarvittaessa.

Käyttäjämäärien hallintaa skaalataan joustavasti ja pilottivaiheessa esiin tulleita kehitysideoita ja uusia työkaluja voidaan ruveta toteuttamaan ja ottamaan käyttöön – Devopsin mukaisesti myös Rootia on jatkuvasti kehitettävä.

04

**YLLÄPITO**

Käyttöönotto ja tuki eivät lopu sen jälkeen kun projektit on siirretty, vaan Root kehittyy ja päivittyy yrityksen tarpeiden mukaan.

Ylläpitosopimusta tehdessä on tärkeää, että määritellään tarpeelliset palvelutasosopimukset Rootin toiminnalle. Ei ole järkeä maksaa 24/7-ylläpidosta, jos oikeasti käyttöä on 9/5. Myös kielivaatimukset tuelle on syytä ottaa huomioon.

01

02

03

04

**For Management**

---

**Marko Klemetti**

CTO

044 522 5927

**Tatu Kairi**

Don't Stop Believin'

**Kalle Sirkesalo**

Devops Cutthroat

**Mika Aho**

Bleeding Edge

Pohjoinen Rautatiekatu 25  
00100 Helsinki

Åkerlundinkatu 11 A,  
33100 Tampere

Uplandsgade 56, 1 sal,  
2300 København S.

Birger Jarlsgatan 18A, 2tr,  
114 34 Stockholm

Marcel- Breuer- Str. 15  
80807 München

De Entree 143  
1101 Amsterdam

**For experts**

---

**Tuomas Keränen**

Leader, Devops Platform

040 530 1672

**Tatu Kairi**

Don't Stop Believin'

**Kalle Sirkesalo**

Devops Cutthroat

**Mika Aho**

Bleeding Edge