

An aerial photograph of a city, likely Helsinki, showing a dense urban landscape with various buildings, streets, and green spaces. The sky is clear and blue. A red rectangular overlay is present in the bottom right corner, containing the 'digia' logo.

# Digia Oyj

– Näkemyksellinen kumppani ja  
muutoksen mahdollistaja

Digian OpenEdge –pohjaisten tuotteiden modernisointi

digia

# Olli Havilehto

Olli on 25-vuotias ohjelmistosuunnittelija Suomen Tampereelta. Olli on työskennellyt erilaisten Web-, mobiili- sekä palvelinsovellusten parissa ja omaa toimialatuntemusta valmistavasta teollisuudesta terveydenhoitoalan järjestelmiin sekä kotimaisiin ERP-ratkaisuihin.

Tällä hetkellä Olli työskentelee osana Digia Enterprisen tuotekehitystiimiä. Työn ohessa hän viimeistelee tietojenkäsittelyopin maisteriopintojaan Tampereen Yliopistossa. Progress & Telerik - tuotteista Ollilla on neljän vuoden työkokemus.



# Sisällys

1. Digia
2. Arkkitehtuuriuudistus
3. Taustajärjestelmä
4. Edusta
5. Välineet



# Digia lyhyesti

- Kasvava digitaalisiin palveluihin keskittyvä IT-palveluyritys - näkemyksellinen kumppani ja muutoksen mahdollistaja
- Perustettu 1997
- Toimipisteet Helsingissä, Jyväskylässä, Lahdessa, Oulussa, Raumalla, Tampereella, Turussa ja Vaasassa sekä Ruotsissa Tukholmassa



**1000**

ammattilaista  
kahdessa  
maassa



**96,2 M€**

liikevaihto



**1,8 M€**

liikevoitto



**Pörssiyhtiö**

NASDAQ  
Helsinki

Palveluiden kehittäminen

Toiminnan ohjaaminen

Tiedon hyödyntäminen

\*) luvut 2017 lopun tilanne

digia

# Rakennamme sujuvaa digiarkea



## Palveluiden kehittäminen

Saat digitaalisilla  
palveluilla kilpailuetua

Sähköinen asiointi  
Verkkokaupankäynti  
Sähköinen työympäristö  
Asiakaskokemuksen kehittäminen  
Palveluiden mobilisointi  
Palvelunhallinta



## Toiminnan ohjaaminen

Saat parempaa tehokkuutta  
koko arvoketjuun

Toiminnanohjaus  
Henkilöstönohjaus  
Asiakkuuden hallinta



## Tiedon hyödyntäminen


Saat tukea  
päätöksien tekoon

Integraatio  
Analytiikka  
Asioiden internet  
Ydintiedon hallinta  
Tiedon turvaaminen



# Lisäämme asiakkaidemme valovoimaa digiarjessa





*"Täällä panostetaan henkilöihin. Kuunnellaan ihan oikeasti sitä, että mitä sä haluat ja mihin sä haluat mennä."*

*- Turo, projektipäällikkö*

*"Oppimisen mahdollisuudet ovat lähes rajattomat, vaihtoehtoja löytyy oman mielenkiinnon."*

*- Eero, harjoittelija*

*"Jatkuva kehittyminen ja pyrkimys aina parempaan innostaa tekemään."*

*- Petri, ohjelmoija*

*"Täällä on hyvät työkaverit, jotka kirittää, ja toisaalta meillä on semmonen organisaatio, joka rakentuu sille, että jokainen tekee aina parhaansa."*

*- Mika, tiiminvetäjä*



# Henkilöstömme

## – tärkein voimavaramme

SUKUPUOLI- JA IKÄJAKAUMA		HENKILÖMÄÄRÄ TOIMIPISTEITTÄIN		TYÖSUHTEEN PITUUS	
Naisia	26%	Helsinki	559	0-1	20 %
Miehiä	74%	Jyväskylä	137	1-5	27 %
		Tampere	123	6-10	21 %
20-29	9 %	Rauma	33	11-15	13 %
30-39	36 %	Oulu	6	16-20	13 %
40-49	34 %	Turku	5	20 >	6 %
50-59	18 %	Vaasa	12		
60-	3 %	Tukholma	10		

(12/2016 tilanne)



# Digia

- 1 Suomalainen yritys, riittävän suuri haastamaan kansainväliset kilpailijamme
- 2 Paikallinen päätöksenteko ja ketterät toimintatavat
- 3 Matala organisaatio ja rento ilmapiiri
- 4 Kovaa osaamista, mahdollisuus oppia parhailta asiantuntijoilta
- 5 Runsaasti vaihtoehtoja erilaisiin urapolkuihin ja itsensä kehittämiseen
- 6 Itseohjautuvat tiimit, vapaus ja vastuu omasta työstä, laadukas esimiestyö
- 7 Merkityksellistä tekemistä, mukana rakentamassa digitalisoituvaa yhteiskuntaa
- 8 Kilpailukykyiset työsuhte-edut, työn ja vapaa-ajan tasapaino, joustot

# Arkkitehtuuriuudistus



# Uusiutumisen lähtökohdat ja eteneminen

- Lähtökohdat
  - Asiakkaiden toive selaimella toimivista järjestelmistä
  - Liiketoiminnan tarve uudistaa tuotteita ja luoda uusia tuotteita tehokkaasti
  - Hyvät kumppanit ja varusohjelmisto, jolla pystyttiin tunnistamaan tapa uudistua
- Eteneminen
  - Tuotteiden ja kehityspolun arviointi
  - Yhteistyökumppanien ja kehitysvälineiden valinta
  - POC kehitysvälineistä, uudesta toimintatavasta ja menetelmistä
  - Pilottiprojekti

# Vaatimukset

Kuka	Tarve
Asiakkaat	<b>Käyttö oltava mahdollista mistä tahansa ja miltä laitteelta tahansa.</b> Selainkäyttöliittymä ja rajapinnat mahdollistamaan uudet käyttöliittymät
Asiakkaat	<b>Moderni asiakaskohtainen käyttöliittymä.</b> Käytettävyys ja muokattavuus
Asiakkaat	<b>Uutta toiminnallisuutta.</b> Mahdollisuus liittää olemassa olevat toteutukset ja uudet tuotteet. Jatkuvuus järjestelmäkehityksessä, mahdollisuus päivittää nykyinen järjestelmä uuteen selainkäyttöliittymään, vaihtamatta taustajärjestelmää.
Digia	<b>Tehokkuus toteutuksessa.</b> Muutosten tekeminen nopeammin, laadukkaammin ja mielekkäämmin.
Digia	<b>Tehokkuus toimittamisessa.</b> Versiopäivitysten kehittäminen ja toimitusten automatisointi.
Digia	Oltava kyky luoda <b>uutta liiketoimintaa</b> , muokkaamalla järjestelmiä asiakaskohtaisesti riittävässä laajuudessa.
Digia	<b>Sovellukset palveluna</b> , asennukset, jota voidaan käyttää useilla asiakkailla.



# Ratkaisu

Käyttö selaimella

Uudessa ohjelmistoarkkitehtuurissa

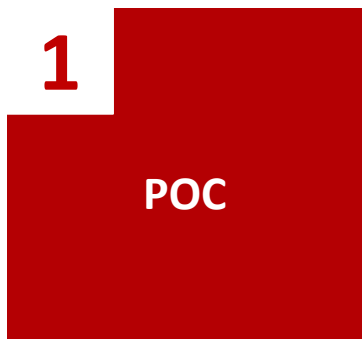
Uusilla työkaluilla

Uudella toteutustavalla

Uusilla mahdollisuuksilla

... mutta tukien myös jo olemassa olevia toiminnallisuuksia

# Eteneminen



2016 – määrittely ja ensimmäiset toiminnot



Ensimmäinen toimintokokonaisuus



Rajattu joukko toimintoja selaimessa sekä perinteisessä sovelluksessa



Asiakkaalle myytävissä olevat tuotteet



# Organisoituminen - Pilotti

Tiimi koottiin eri yksiköistä tulevilla, erilaisen osaamisen ja taustat omaavista henkilöistä. Projektia lähdettiin viemään eteenpäin ketterillä menetelmillä ja DevOps – ajatuksilla

Pohjana onnistumiselle johdon tuki ja osaavat, motivoituneet resurssit

## 1 Palvelumuotoilu

- Toteutettavat toiminnallisuudet
- Käyttöliittymän suunnittelu

## 2 Tausta

- Taustan toteutus
- Arkkitehtuurin muutos
- Uudet välineet ja työtavat

## 3 Edusta

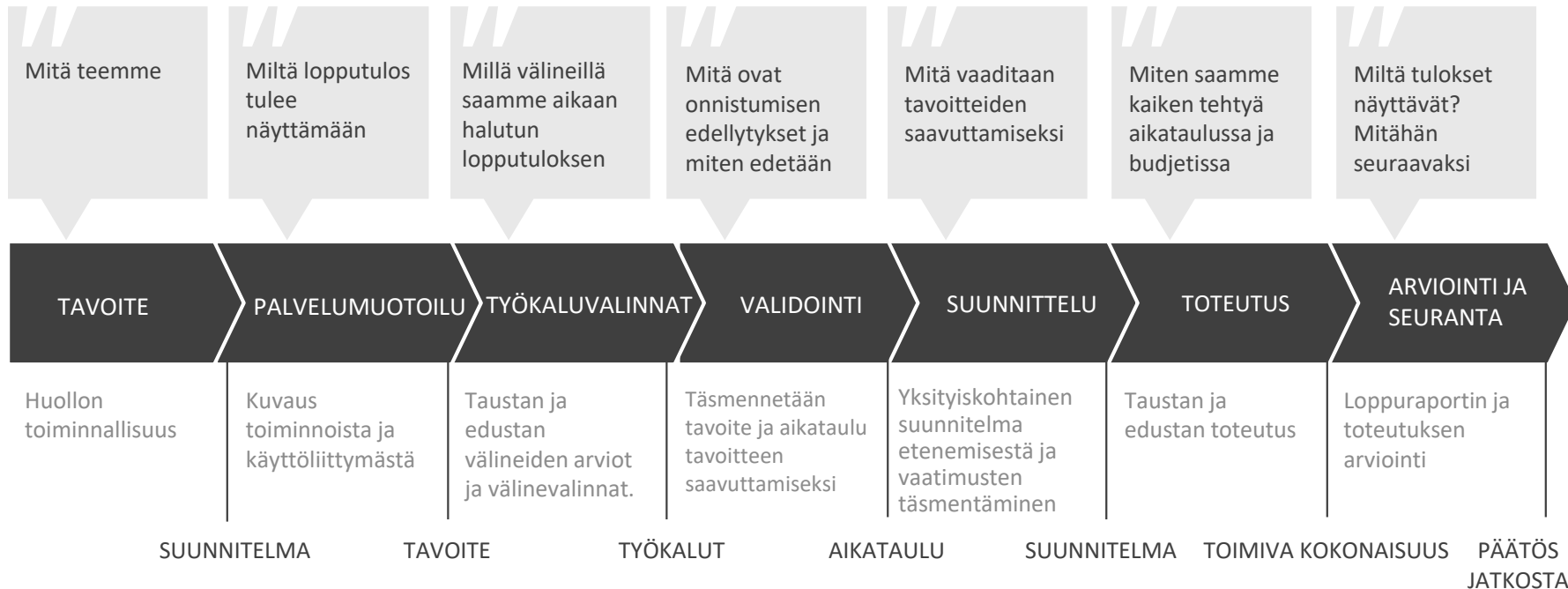
- Käyttöliittymän toteutus
- Uusi käyttöliittymäkirjasto
- Uudet välineet ja työtavat

## 4 Testaus

- Toiminnalliset automaatiotestit
- Testausmenetelmät ja testauksen kehittäminen
- Suorituskykytestit

Osallistujina projektin aikana myös muiden osa-alueiden osaajia tuomassa projektiin oman erikoisosaamisensa.

# Kehitysprosessi ja projektin eteneminen



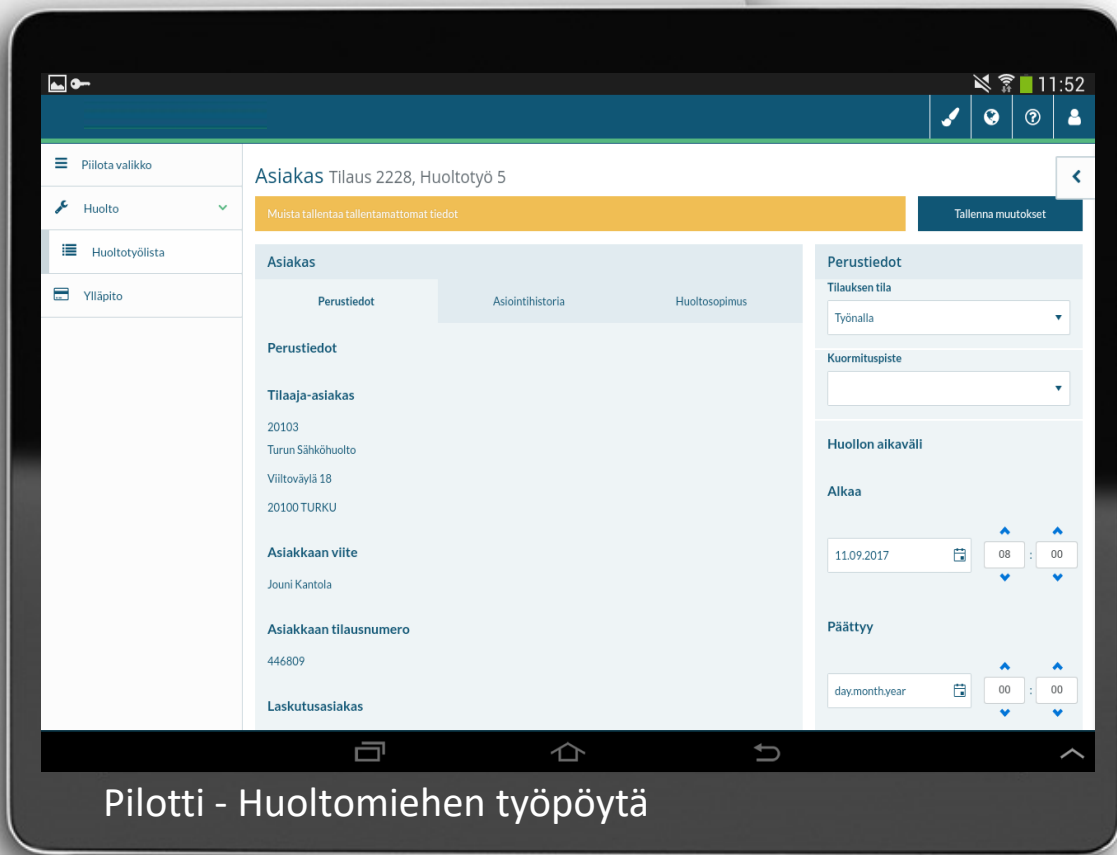
# Pilotin havainnot

## Tausta

- Hieman erilaista ideaa, SCL vs. JSDO
- Ohjelmistoissa monimutkaiset toteutukset
  - Tietojen suojaukset, sessiokäsittelyt
  - Käyttäjätietoja useissa paikoissa
  - Uudella arkkitehtuurilla ei voida korjata tietokannan ongelmia, mutta ne voidaan piilottaa
- Sovelluksissa paljon logiikkaa UI:llä kuten tallennustarkistuksia jne.

## Edusta

- JSDO - Dokumentaatio puutteellista, "ei nykymaailmaa"
- JSDO –hierarkisen datan käyttö ongelmallista
- CD-toteutukset mahdollisia pääasiassa pilveen tai palvelukeskukseen



*”Sen on helvetti paras olla hyvä”*

Stefan Hellstrand, Johtaja, Digia

*”Jos et osaa, kaikki on vaikeaa”*

Marko Kiviniemi, Pienempi Johtaja, Digia

digia



A night street scene with blurred lights and a wet pavement reflecting the light. The text "Taustajärjestelmä" is centered in the image.

# Taustajärjestelmä

# Tausta-arkkitehtuuri ja välineet

## Tarve

- Käyttöliittymäriippumaton tausta
- Multitenancy / SaaS
- Olemassa olevan koodin hyödyntäminen
- Uusi, tehokkaampi tapa tehdä työtä

## Hyödyt

- Monikerrosarkkitehtuuri
- Jatkuvuus tekemiselle
- Standardi toteutustapa
- Toteutuskerrosten vastualueet selkeiksi

## Käytännössä

- Kumppaniksi Consultingwerk ja SmartComponentLibrary monikerrosarkkitehtuurin luomiseen
- PAS (Progress Application Server)
- SonarQube linttauksen ja testikattavuuteen (Gilles Querret)

# OERA-malli ja taustan uusi arkkitehtuuri

OpenEdge Reference Architecture

Asiakkaan prosessi ja palvelutarpeet

Selain



Mobiili



Analytiikka



3. osapuolen  
järjestelmä



Palvelurajapinta

Business Entity (BE) -  
Liiketoimintalogiikka

Sovellus  
Logiikka

Data Access (DA) - Tietokantarajapinta

Tietokanta – Pilvi / MS SQL / OpenEdge / ...

Smart  
Component  
Library (SCL)  
ja  
Digia  
Framework

digia

# Taustan Ohjelmointi

## Tarve

- Laadun parantaminen ja nopea oppiminen
- Kolmannen osapuolen välineiden hyödyntäminen

## Käytännössä

- Olio-ohjelmointi
- OE Studio + SCL:n työkalut liiketoimintalogiikan ja tietokantarajapinnan generointiin
- SCL-työkalujen kustomointi ja DigiaFramework
- Testausvälineet

## Hyödyt

- Uusien projektien toteutukset yksinkertaisempia toteuttajille
- Välineet ohjaavat toteuttajia hyvään laatuun
- Tekeminen voidaan jakaa osaamisen mukaan

## Työtavat

- Sprintit – ketterä kehittäminen
- Tiimin kommunikointi – jatkuvaa kommunikaatiota ja yhteisiä tilaisuuksia
- Testit: Yksikkö- ja toiminnalliset testit
- Koodin katselmointi



# Taustan CI

## Tarve

- Muutosten integrointi ohjelmakirjastoihin
- Vakaan taustajärjestelmän tarjoaminen edustakehittäjien käyttöön
- Yksikkötestien Commit -perusteinen suoritus

## Käytännössä

- DevOps: Git, Jenkins 2, SonarQube, Robot Framework
- Muutokset testien onnistuttua kehitys-, QA- ja robottitestiympäristöihin ja julkaisu tarpeen mukaan

## Hyödyt

- Virhetilanteiden välttäminen
- Automatisoitu julkaisu

# Käyttöliittymä näkymäkuvauksesta

## Tarve

- Näkymät taustajärjestelmästä
- Helppo käyttöliittymien suunnittelu / ylläpito

## Hyödyt

- Yksi frontend toteutus, näkymät tarpeen mukaan
- Tausta ohjaa täysin edustan käyttöliittymän muodostusta

## Käytännössä

- Näkymien generointi taustan ylläpito näkymistä (perusylläpidot)
- Näkymien suunnittelu uudelle toiminnallisuudelle
- Asiakaskohtaiset/käyttäjakohtaiset näkymät
- Selainpohjainen näkymien ylläpito

A person is working on a laptop outdoors. The laptop is open on a wooden desk. To the right of the laptop, a hand is holding a red pen, writing on a white cup. In the foreground, a white smartphone lies on the desk. To the left of the laptop, there are several colored pencils and a notebook. The background is a blurred view of green trees and sunlight, creating a bokeh effect. The word "Edusta" is written in white text in the center of the image.

Edusta

# Edusta - selainkäyttöliittymä

## Tarve

- Päätelaiteriippumaton ja moderni, palvelumuotoiltu käyttöliittymä
- Taustan rajapintojen käyttö
- Tehokkuus ja luotettavuus

## Käytännössä

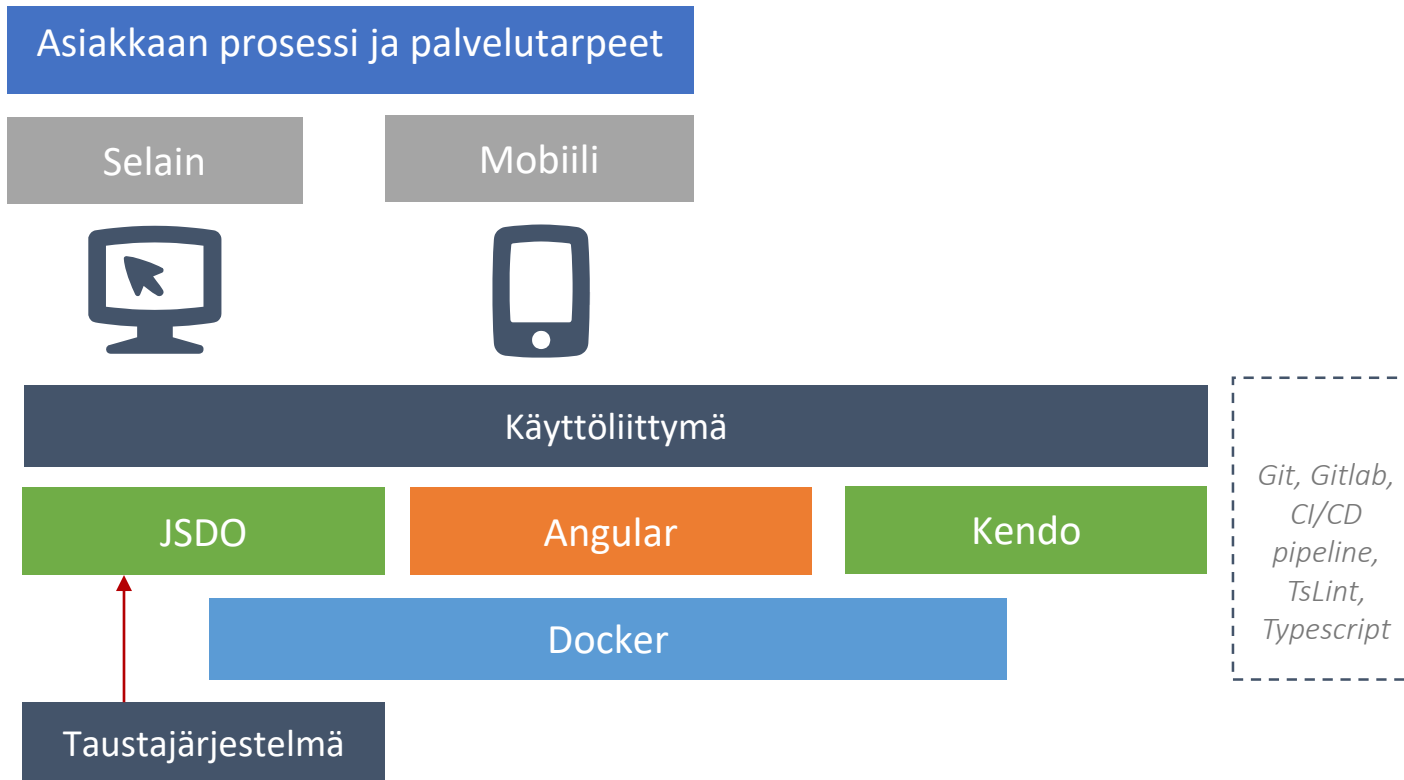
- Angular -käyttöliittymä
- Kehitys Scrum-mallilla, Test Driven Development
- Mikrokomponentteja joista rakennetaan käyttöliittymä
- Tausta kuvaa käyttöliittymän JSON API-formaatissa

## Hyödyt

- Käyttöliittymän skaalautuvuus
- Komponenttien uudelleenkäytettävyys
- Työ tehdään edustassa vain kerran
- Helppokäyttöisyys, asiakastyytyväisyys
- Helpompi uusasiakashankinta
- Varmatoimisuus ja nopeampi kehitys
- Tarjoillaan pilvessä
  - Automaattinen provisiointi
  - SaaS-palvelumalliin siirtyminen



# Edusta - Arkkitehtuuri



# Edusta - Testaus

## Tarve

- Nopeampi iteraatiovauhti
- Toimintavarmuus
- Koodin testikattavuus ja laadunvarmistus

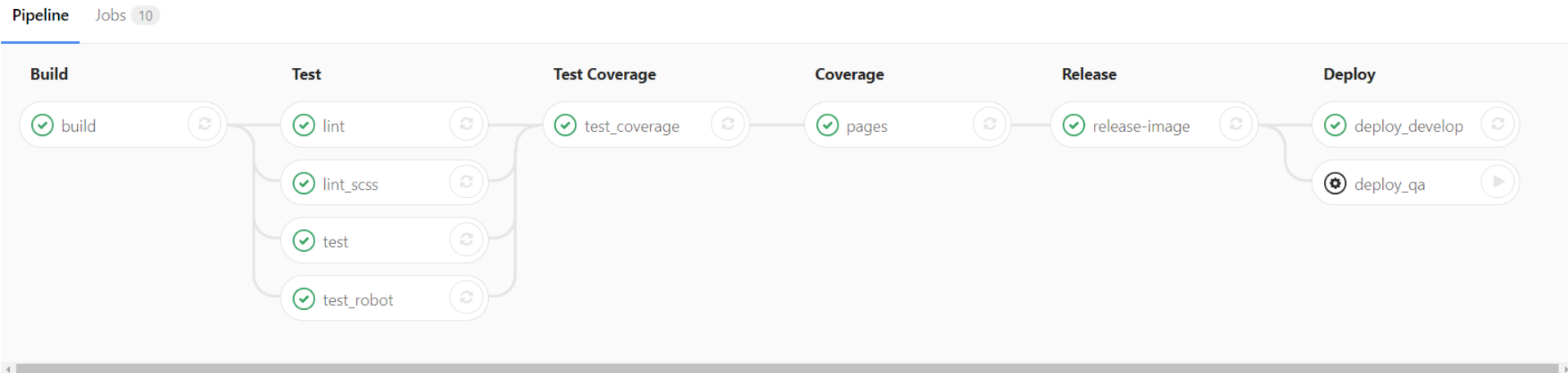
## Hyödyt

- Toteuttamisen kulttuurin muutos
- Nopea ja toimintavarma kehityssykli ideasta tuotantoon
- Parempi laatu

## Käytännössä

- Koodikattavuuden tavoite 80% (yksikkötestit)
- Integraatiotestaus Robot Frameworkilla, Suorituskykymittaus JMeter
- Testien ajaminen läpi CI-pipelinessä jokaisella commitilla

# Edusta – CI-pipeline / GitLab



## CI-pipeline

- N-kappaletta vaihteita, joita voi ryhmitellä kokonaisuuksiksi
- Konfigurointi versioidun määrittelytiedoston avulla (CI pipeline as code)
- Koodin käänös ja validointi, tyylien validointi, yksikkötestit, robottitestit
- Koodin testikattavuus ja julkaisu, sekä asennus

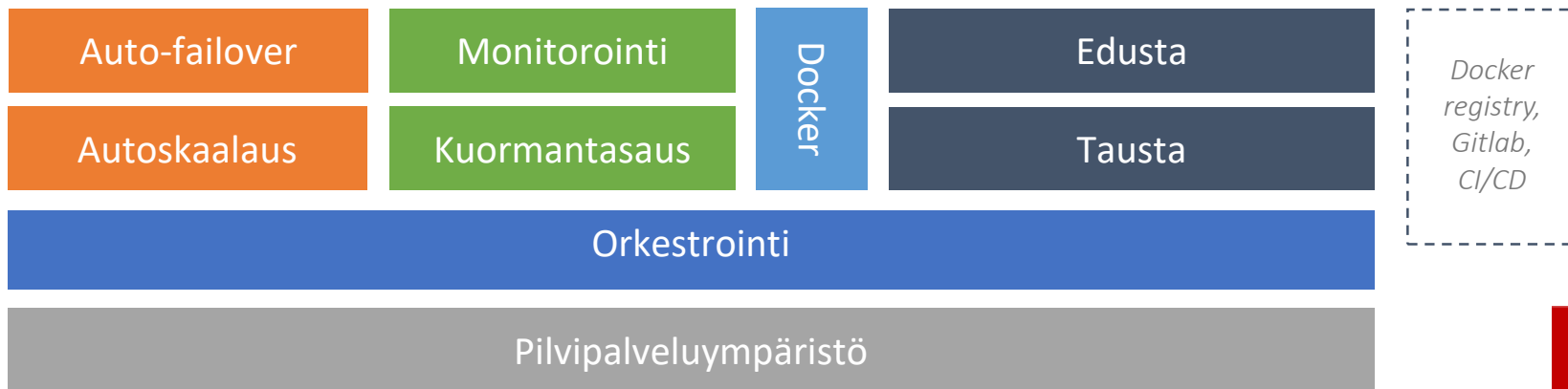
# Edusta – Pilvipalveluympäristö

## Tarve

- Toimituksen ja versiopäivitysten tehokkuus
- Ei asennuksia asiakkaille, monistettavuus
- Sovellus palveluna

## Hyödyt

- Automatisoitu palveluympäristö → tehostettu toiminta
- Joustavuus, skaalautuvuus, kustannus käytön mukaan, tietoturva





A night street scene with blurred lights and a wet cobblestone path leading towards the viewer. The path is made of dark, wet cobblestones that reflect the ambient light. In the background, there are numerous out-of-focus lights in various colors (yellow, white, blue, red, green) that create a bokeh effect. The overall atmosphere is dark and moody.

# Työkaluja

# OE Studio

The screenshot displays the OpenEdge ABLUnit development environment. The main window shows the source code for the `ServiceOrderBusinessEntity` class. The code includes a class declaration, inheritance, and a constructor. The constructor sets the `DataAccessName` property. There are also comments describing the purpose and notes for the class and its methods.

```
32
33 CLASS Digia.Erp.Service.ServiceOrder.ServiceOrderBusinessEntity
34 ....INHERITS DigiaBusinessEntity
35 ....USE-WIDGET-POOL:
36
37 .....{ Digia/Erp/Service/ServiceOrder/dsServiceOrder.i }
38
39 ...../**
40 .....* Purpose: Constructor of the ServiceOrderBusinessEntity class
41 .....* Notes: Passes the handle of the dataset instance to the base class and
42 .....* sets the default DataAccessName
43 .....*/
44 .....CONSTRUCTOR PUBLIC ServiceOrderBusinessEntity():
45 .....SUPER (DATASET dsServiceOrder:HANDLE).
46
47 .....THIS-OBJECT:DataAccessName = "Digia.Erp.Service.ServiceOrder.ServiceOrder
48 .....END CONSTRUCTOR.
49
50 ...../**
51 .....* Purpose: Provides a hook to modify data in the ProDataset after Read and
52 .....* Update operations (i.e. population of aggregated values)
53 .....* Notes: Invoked during FetchData () and SaveChanges ()
54 .....*/
55 .....METHOD OVERRIDE PUBLIC VOID ReceiveData():
56
57 .....END METHOD.
58
```

The interface includes a Project Explorer on the left showing a tree view of the project structure, including folders like `DigiaFramework`, `Erp`, and `Tools`. The bottom status bar shows `Runs: 0/0`, `Errors: 0`, and `Failures: 0`.

# OEDT

## Hyötyjä

- Outline näkymä suoraan Projekti-puussa
- Tietokannan navigointi (DB navigator) ilman SQL-yhteyksiä
- Paljon ohjelmointiavusteita, jotka **nopeuttavat** ohjelmointia ja **vähentävät virheiden** mahdollisuuksia
  - Näyttää metodien kuvaukset
  - Näyttää metodien input-parametrit ja niiden selitteet
  - Näyttää käyttämättömät muuttujat
- Osaa näyttää **helposti** ohjelmat, jotka käyttävät valittua include-tiedostoa (include muutos laukaisee käännöksen sitä käyttäville kooditiedostoille)
- Debug Listing file compile

# Business Entity Designer

The screenshot displays the Business Entity Designer application window. The title bar reads "ServiceOrderBusinessEntity - Business Entity Designer (C:\enterprise\product\shelves\stara\Erp\Erp\Service\ServiceOrder\ServiceOrderBusinessEntity.bedm)". The interface includes a menu bar (File, Business Entity, Temp Table, Data-Relation, Generate), a toolbar with icons for various actions like Update, Save, Cancel, Pointer, Relation, Auto Layout, Import Table, Paste ABL, Custom Business Entity Properties, Import SDO, Import Rollbase Object, Relation Fields, Generate, Compile, Edit Source, Business Entity Tester, Filter, Clear, History, Expand all, Collapse all, Generate Viewer, Generate Window, Generate TabPage, Default fields, and SmartFramework, and a Toolbox on the right containing various Smart\* classes and database types (PGDB, POWDB, DICTDB).

The main design area is a grid with several entities and their fields:

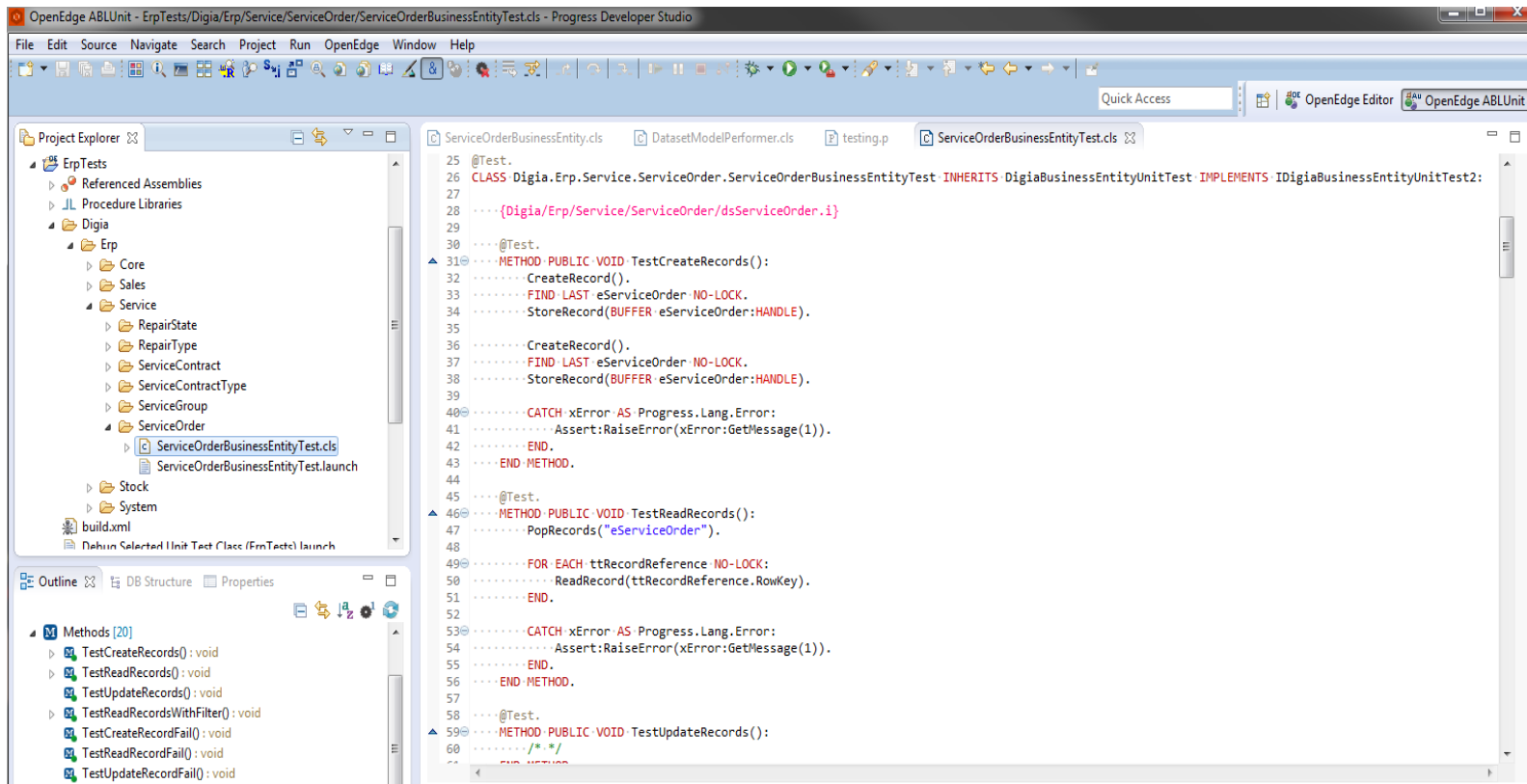
- Address (PU)**: Fields include Country, SecurityKey, SecurityField, Active, InfoText, State, EdiPartyId, and Indexes.
- eDeliveryAddress**: Fields include RecordNum, StreetAddress1, StreetAddress2, PostCode, and PostOffice.
- Table (PU)**: Fields include IrExtSympton, IrSection, IrRepair, RefNum, PCB, Qty, ItemCode, Description, IrDefect, IrCondition, TableName, IrLineIrCodeRelation, SecurityKey, IrisCodeKey, and Indexes.
- Table (PU)**: Fields include Urgency, BegDate, SerContNum, ChargeCode, OrderDate, OrderTime, ContactNum, ItemCode, Version, SerialNbr, SCustNum, WarrantyBegDate, SCustName1, WarrantyEndDate, SerConLineKey, Project, ItemName, FwNum, PromDate, PromTime, ReqTime, SerContract, Info, DivDate, and DivTime.

Relationships are shown with arrows: "eServiceOrder" connects Address and Table; "eServiceOrderServiceLineRelation" connects Table and another Table; "eServiceOrder" connects Table and another Table.

At the bottom, there are three property windows:

- Entity Properties**: Business Entity Name: ServiceOrderBusinessEntity, Business Entity Purpose: Business Entity for ServiceOrder.
- Index Properties**: Index Name: LineNum, Unique: .
- Field Properties**: Field Name: OrderNum, Data Type: INT64, Validation Settings: Extent: 0.

# Yksikkötestaus



The screenshot displays the Progress Developer Studio interface. The Project Explorer on the left shows the project structure, with the file `ServiceOrderBusinessEntityTest.cls` selected under the `ServiceOrder` folder. The Outline pane at the bottom left lists the methods defined in the test class: `TestCreateRecords()`, `TestReadRecords()`, `TestUpdateRecords()`, `TestReadRecordsWithFilter()`, `TestCreateRecordFail()`, `TestReadRecordFail()`, and `TestUpdateRecordFail()`.

The main editor window shows the source code for `ServiceOrderBusinessEntityTest`. The code is as follows:

```
25 @Test.  
26 CLASS Digia.Erp.Service.ServiceOrder.ServiceOrderBusinessEntityTest INHERITS DigiaBusinessEntityUnitTest IMPLEMENTS IDigiaBusinessEntityUnitTest2:  
27  
28 .....{Digia/Erp/Service/ServiceOrder/dsServiceOrder.i}  
29  
30 .....@Test.  
31 .....METHOD PUBLIC VOID TestCreateRecords():  
32 .....CreateRecord().  
33 .....FIND_LAST eServiceOrder NO-LOCK.  
34 .....StoreRecord(BUFFER eServiceOrder:HANDLE).  
35  
36 .....CreateRecord().  
37 .....FIND_LAST eServiceOrder NO-LOCK.  
38 .....StoreRecord(BUFFER eServiceOrder:HANDLE).  
39  
40 .....CATCH xError AS Progress.Lang.Error:  
41 .....Assert:RaiseError(xError:GetMessage(1)).  
42 .....END.  
43 .....END METHOD.  
44  
45 .....@Test.  
46 .....METHOD PUBLIC VOID TestReadRecords():  
47 .....PopRecords("eServiceOrder").  
48  
49 .....FOR EACH ttRecordReference NO-LOCK:  
50 .....ReadRecord(ttRecordReference.RowKey).  
51 .....END.  
52  
53 .....CATCH xError AS Progress.Lang.Error:  
54 .....Assert:RaiseError(xError:GetMessage(1)).  
55 .....END.  
56 .....END METHOD.  
57  
58 .....@Test.  
59 .....METHOD PUBLIC VOID TestUpdateRecords():  
60 ...../* */  
61 .....END METHOD.
```



# Kehittäjädokumentaatio

- Luokkadokumentaatio koodista
- Wiki
  - Sekä edusta että tausta
  - Ympäristökuvaukset
  - Työkalujen käyttöönotto
  - PAS konfiguointi
  - Testaus
  - Paljon muuta

The screenshot shows a code documentation tool interface. On the left is a tree view of the project structure, and on the right is the details for the `DigiaHybridRealm` class.

**Tree View:**

- Digia
  - Erp
  - Exception
  - Framework
    - Authentication
      - DigiaHybridRealm**
      - HybridRealm
    - Authorization
      - DigiaRequestAuthorizationProvider
    - Server
      - DigiaRestServerSessionActivator
      - SessionActivator
    - Session
    - System
  - OERA
  - Service
  - Tools
  - Util

**Class Details:**

Namespace: `Digia.Framework.Authentication`

## Class DigiaHybridRealm

Parent classes: `Consultingwerk.Web2.SmartFramework.Authentication.SmartHybridRealm`  
`Digia.Framework.Authentication.DigiaHybridRealm`

Inherits: `Consultingwerk.Web2.SmartFramework.Authentication.SmartHybridRealm`

File: `DigiaHybridRealm`  
Purpose: Validate user id and password  
Description: \_User table must have values in user\_number-field  
Author(s): toperata  
Created: Tue Nov 22 15:44:10 EET 2016

Navigate to: [Methods](#) | [Constructors](#) | [Events](#) | [Properties](#)

### Method Summary

Options	Name	Purpose
[-]	# LOGICAL ValidatePassword (integer, character)	Validate password
[-]	# INTEGER ValidateUser (character)	Validate user

### Constructor Summary

Options	Name	Purpose
[-]	# DigiaHybridRealm ()	Constructor for the DigiaHybridRealm class

### Method Detail

**PROTECTED LOGICAL ValidatePassword (integer, character)**

Purpose: Validate password

# Yhteenveto

Nopeampaa kuin koskaan ennen

- Kehitys
- Toimitus
- Ylläpito



Kiitos!

#digiarki

Digia – Tähdellisten tehtävien tekemiseen

[www.digia.com](http://www.digia.com)

digia