

# Reuse

Value and importance for increased efficiency and faster time-to-value

English





# Re-Use

Value and importance for increased efficiency and faster time to value

November 2024 | Marco Duizer, Technical Solutions Director



# Our clients have a lot to say...

## Rabobank/Bank

"This new approach has improved communication within the team, enhancing mutual understanding. As a developer, I also feel more comfortable collaborating inside App Studio rather than relying solely on paper or whiteboards."

## Lex Ruijter/LeasePlan

"The promise appears in your face. The Modules together with the use of app studio and Constellation are a dream come true. Less work leading to shorter Time to market, more consistency in design and reuse of patterns you get instant ROI, instant UI and instant business value. Governance is easier at a lower cost."

## ING/Bank

"With the Low Code Enterprise training, Pega has helped ING to understand what is possible with Pega and how we need to change our way of working so that we can deliver on ING's digital ambitions in a faster way."

## Car Insurance

"I was skeptical first that we could deliver this new application with Pega in 7 months, now I think we can deliver the application within 3 to 4 months. (And they did, way before the deadline!)"

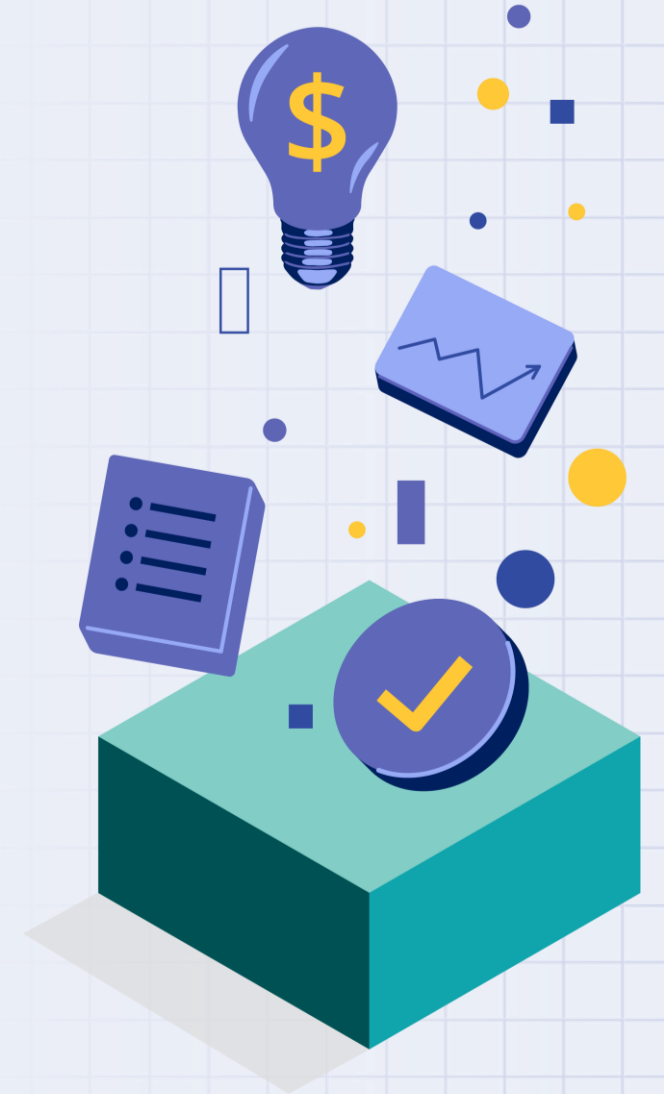
## Telenet/Telco

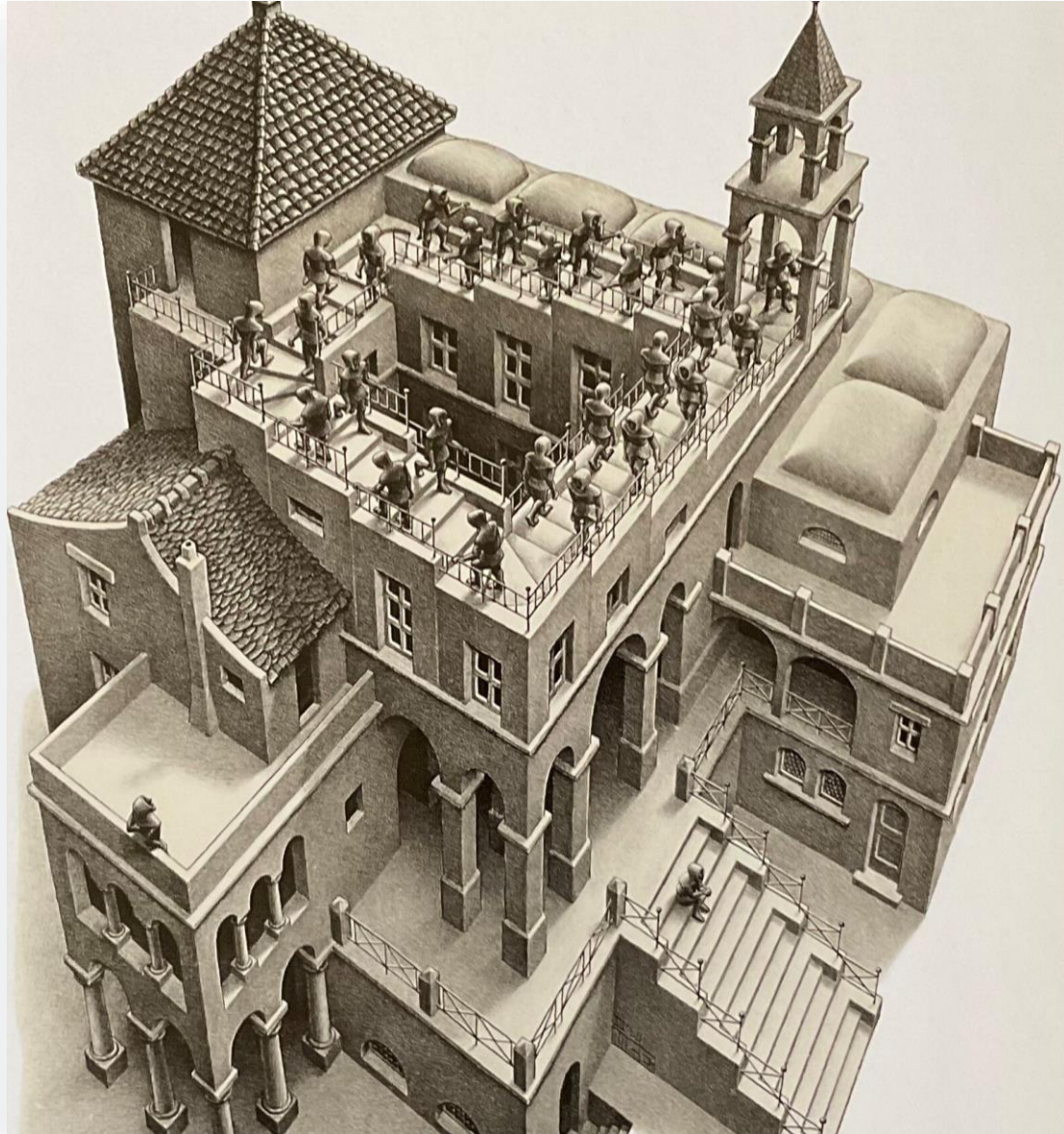
"The pace and quality of the first version of the application was an eye opener for the participants, showing the true potential of low-code development in Pega and translating the theory into practice."





# Why Re-Use?





## Design challenge



# Modular approach



Enable



Build

# Using Fusion Teams

A fusion team is a **cross-functional** approach to low-code application delivery that involves **collaboration** between **citizen developers**, **IT**, and other **experts** to mitigate security, operational, regulatory, reputational, and financial risk. With a fusion team, applications can be built quickly and at **lower cost**, without introducing unnecessary risk.

## Who makes up a fusion team?



### Project manager

Ensures project is completed on time and budget



### Business analyst

Gathers and translates application requirements



### Citizen developer

Develops and maintains app with low code



### Subject matter expert(s)

Represents the business function for the app



### Professional developer

Provides coaching and advanced configuration



### IT professional(s)

Manages the CI/CD pipeline, testing, and tuning

# How to work as a Fusion Team

- Work together and use each other strengths
- Enable the business team to quickly create new business value
- Support from a COE for governance and oversight
- Knowledge sharing between the teams
- Enablement for team members
- Have a shared way of working throughout the organization

We use **Low Code Enterprise Development** to enable and guide the teams in their fusion team journey!





# Work Together

**Document driven development**, System Architect doing the work and get instructions (by documents) from the other disciplines (Business, UX, PO, SME...)



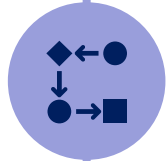
**Model Driven development**, work together on the application, deliver in the App studio designs that needs detailing by system architects, clear vision from the PO on direction for the team. Less documents more Pega models.



# What is Low Code Enterprise Development (LCED)



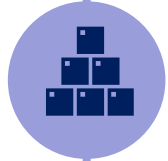
**Business driven designs** that are designed for re-use on business level



Enablement of the Business Architect to work from **App Studio** and get more done (Assemble the Blocks)



System Architects build Features (the blocks) in App studio and Dev studio **to enable the Business Architect in App studio**



A Business architecture that supports and **enables re-use**



The way to **scale, speed up and save money** on application development for the Enterprise

# Background on a sample implementation



## Challenges



Full End-to-End lease process, from quote to sell the car



32 countries worldwide



Private Individuals  
Small/Medium  
Enterprise (SME)  
Corporate clients



Direct sales  
Dealer network (partners)  
White label

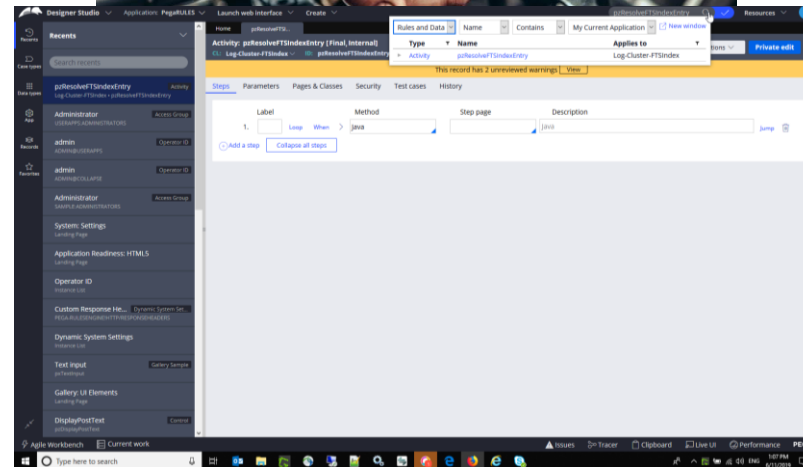


1,5 year

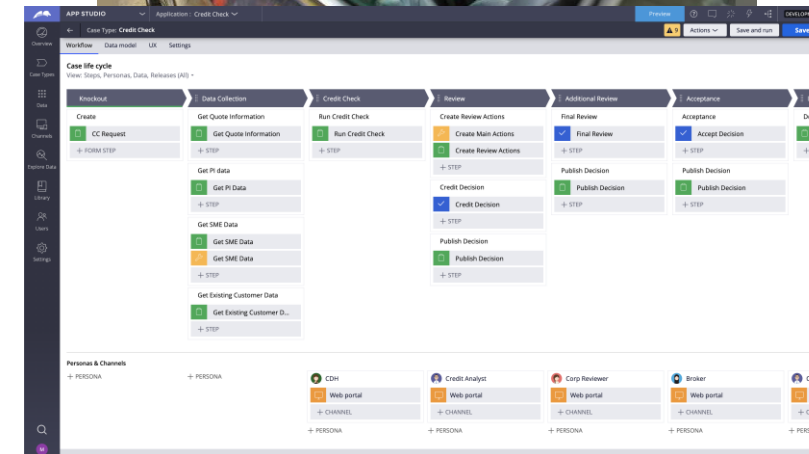
# Skills needed changed over the years!



Pega 5.X



Pega 7.X



Infinity '24

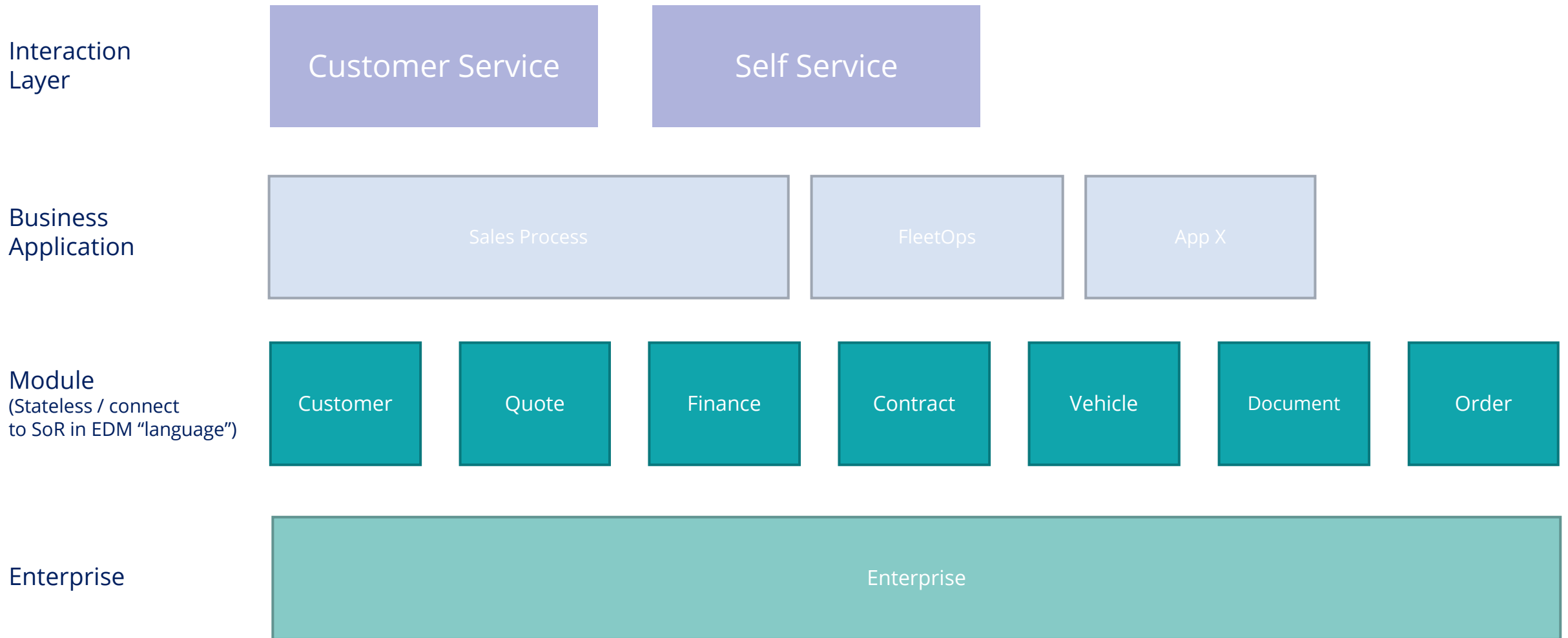


# How to implement



# Example Business Architecture

Not complete, meant for explanation only



# Handle Multiple Countries/Labels and Innovation

Not complete, meant for explanation only

Configuration /  
Innovation Layer

Sales Process LPUK

Sales Process LPNL

Sales Process ARUK

Sales Process LPPT

Innovation XX

Business  
Application

Sales Process

Module  
(Stateless / connect  
to SoR in EDM "language")

Customer

Quote

Finance

Contract

Vehicle

Document

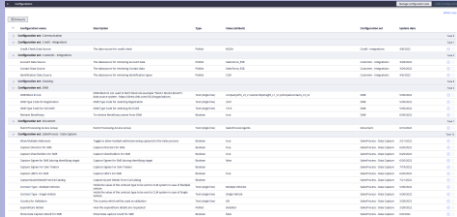
Order

Enterprise

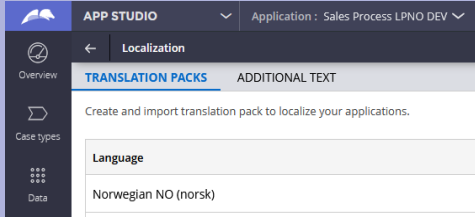
Enterprise

# Business Architecture Implementation

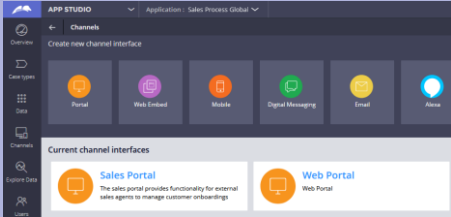
### Configuration/Innovation (application rule)



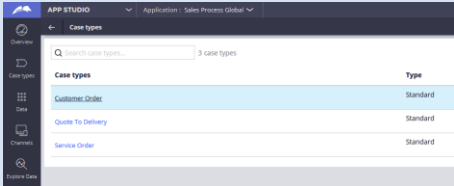
### Localization



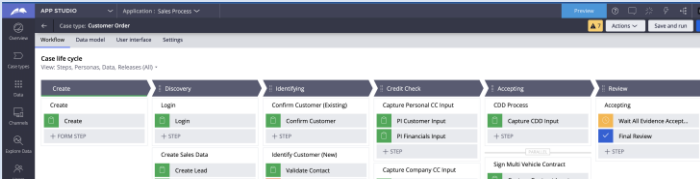
### Channel



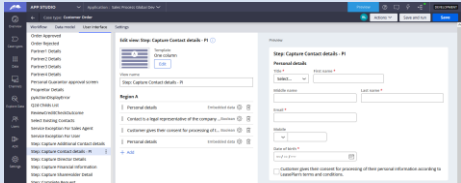
### Business Application (application rule)



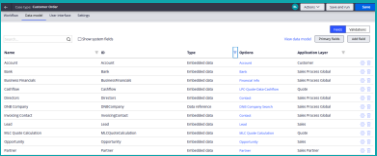
### Orchestration



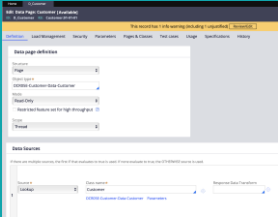
### (Assignment) Views



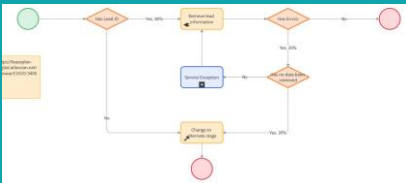
### Module Application (application rule)



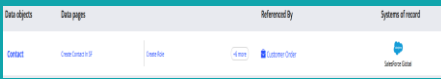
### Data Pages



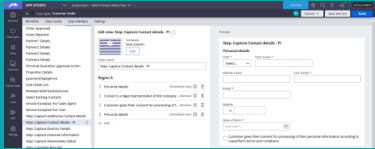
### Smart Shapes (Features)



### Integrations



### (Data) Views



External Services







# Design for Re-Use



# From Need to Application

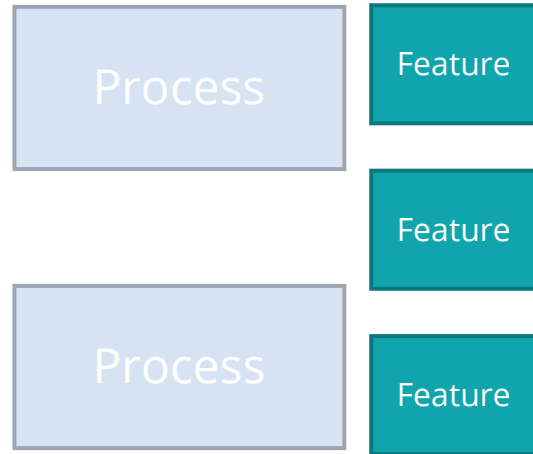
## Requirements



PO



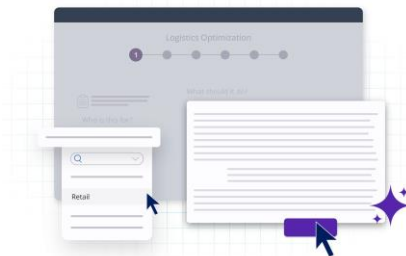
## Feature Map



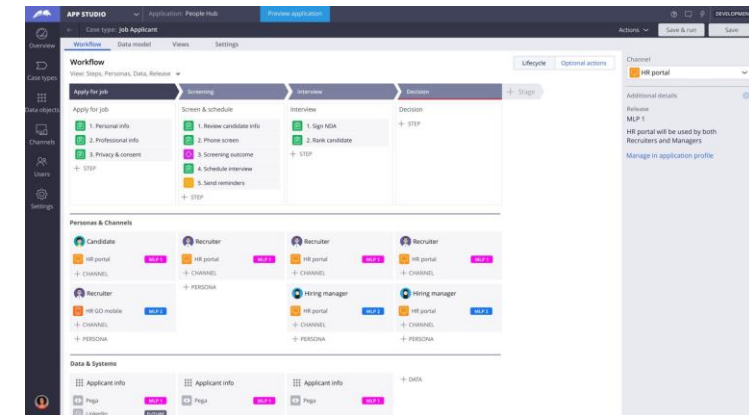
BA



## Pega GenAI Blueprint



## App Studio detailing



SA



# Implement features



- User story
- Refinement
- Build
- Test feature
- Test Process
- Deploy

- ~~User story~~
- ~~Refinement~~
- ~~Build~~
- ~~Test feature~~
- Test process
- Deploy

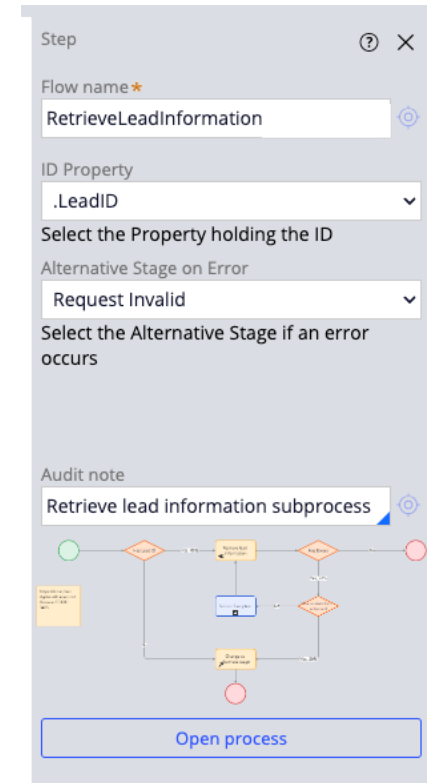
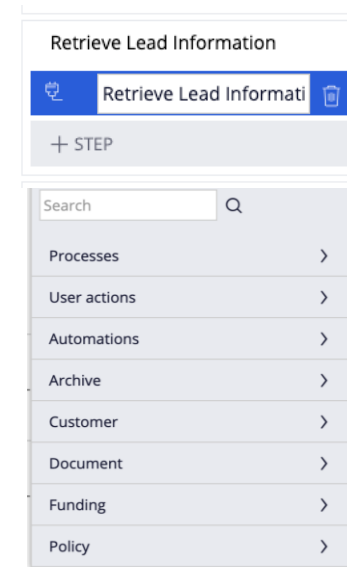
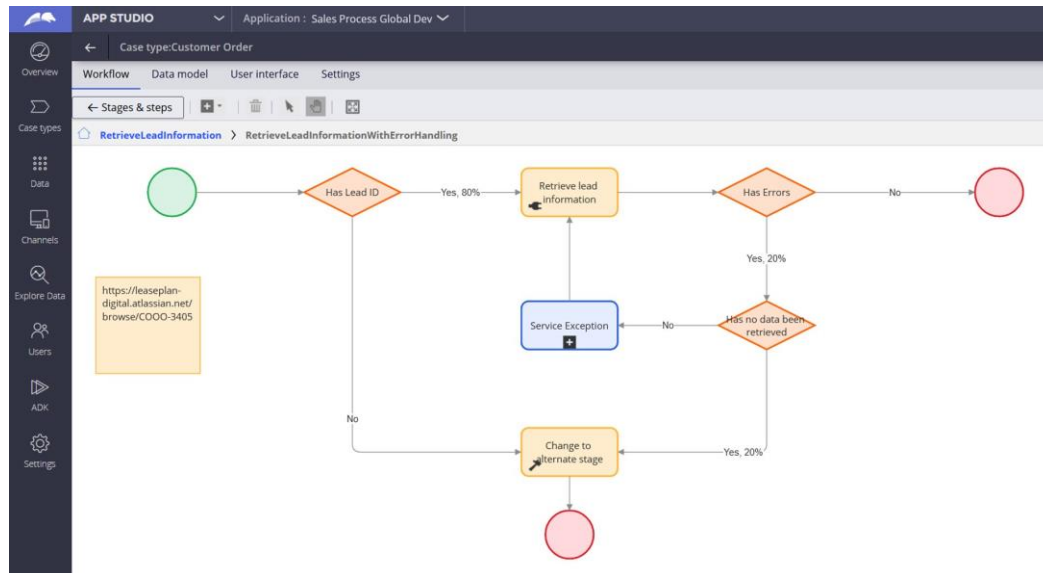
- ~~User story~~
- ~~Refinement~~
- ~~Build~~
- ~~Test feature~~
- Test process
- Deploy

# Feature Example (Design for Re-Use)

**Design reusable business features** in such a way that they can be easily configured in App Studio by the Business Architect.

**Build reusable business features** in the shareable layer of the architecture. (Module Application).

**Result: higher re-use** and **accelerated builds**.



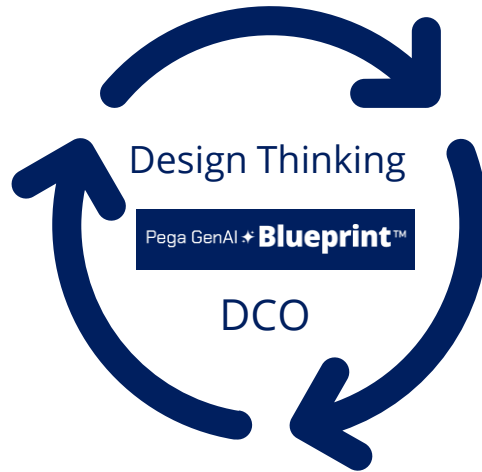
# Deliverables



Backlog Review/Design Period



- Prioritization
- Refinement



- Feature Map
- Solution Design
- Case Type(s)
- Planning (Estimator)
- Data Model



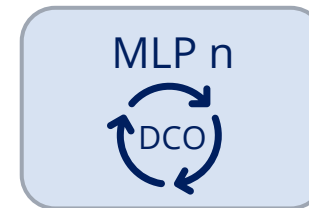
Build Period



- Detail features
- Integration
- Insights



- Channels
- Configuration Sets
- Testing



- Production Support



# Return on Investment



# How do you get your ROI?



## During Design

- Design features once and reuse many times
- Focus on the important part of the business process
- More efficient design sessions



## During Build

- Only new features need to be built
- Build is easier because atomic features have clear inputs and outputs
- QA is simplified since features can be easily tested without running a whole case type



## During Maintenance

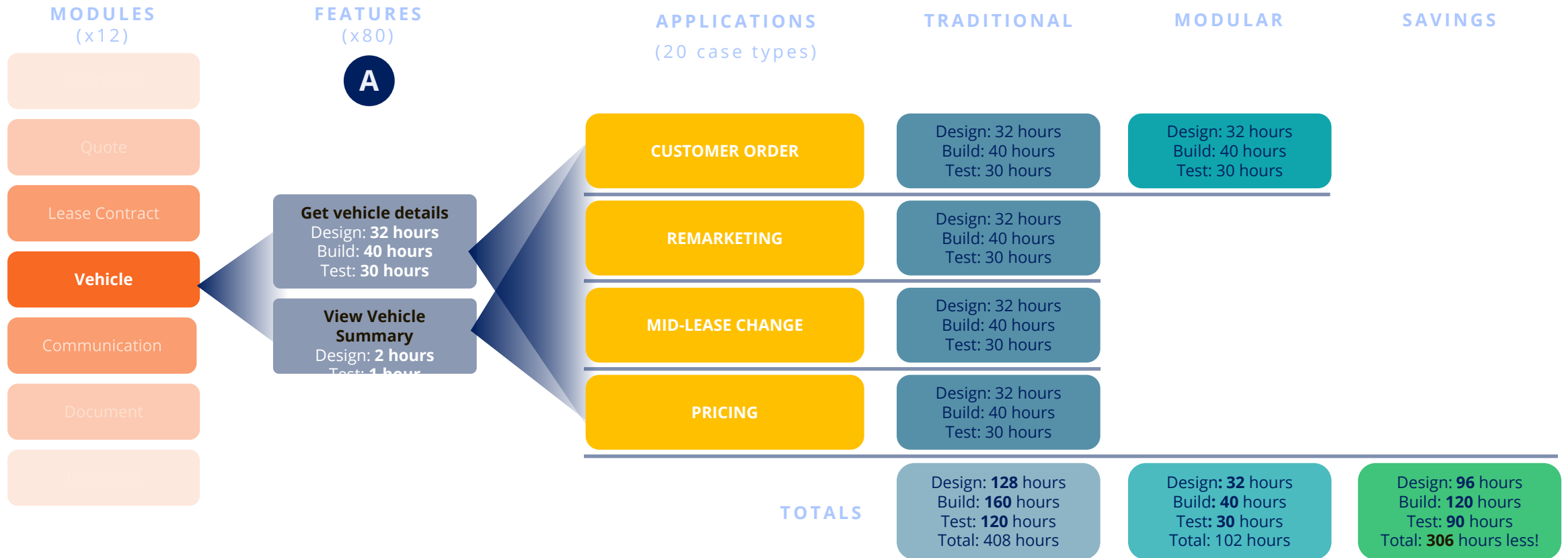
- New version of an integration can be implemented in one place, and applications can consume the new version as needed
- Fix can be applied once in a Module and consuming applications automatically benefit



## Overall

- Faster time to market
- Richer MLP for new case types and apps because of features reuse
- Reduced dependency on on highly experienced System Architects; shift to Business Architects

# ROI example with Enterprise Reuse



**B**

Average saving will be **102 hours per feature** on each subsequent use.

**C**

Around **20** case types were built, and on average each feature is used **10** times by a case

**A B C**

$80 \times 102 \times (10-1)$   
**=73,440**  
hours saved!





# Governance re-use

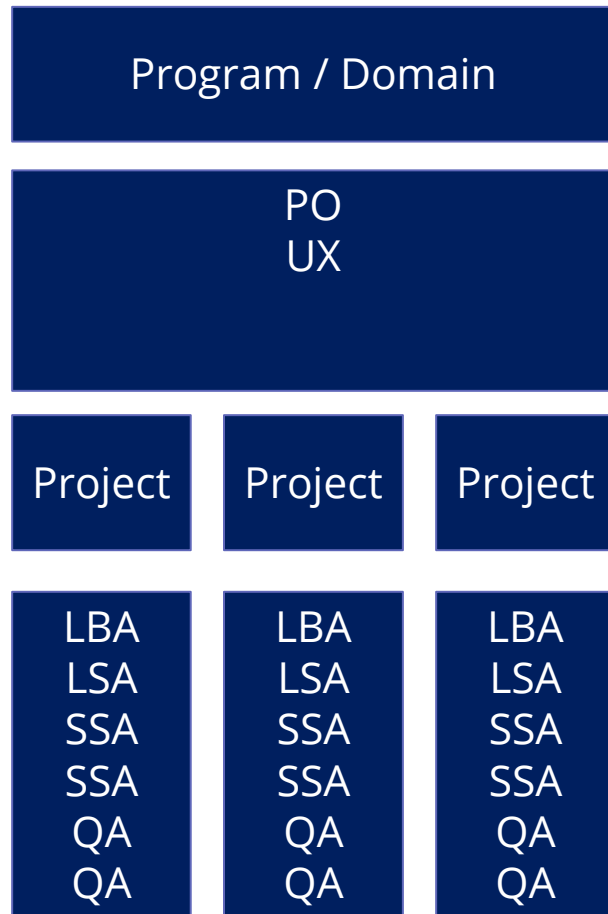


# Governance of the Re-use

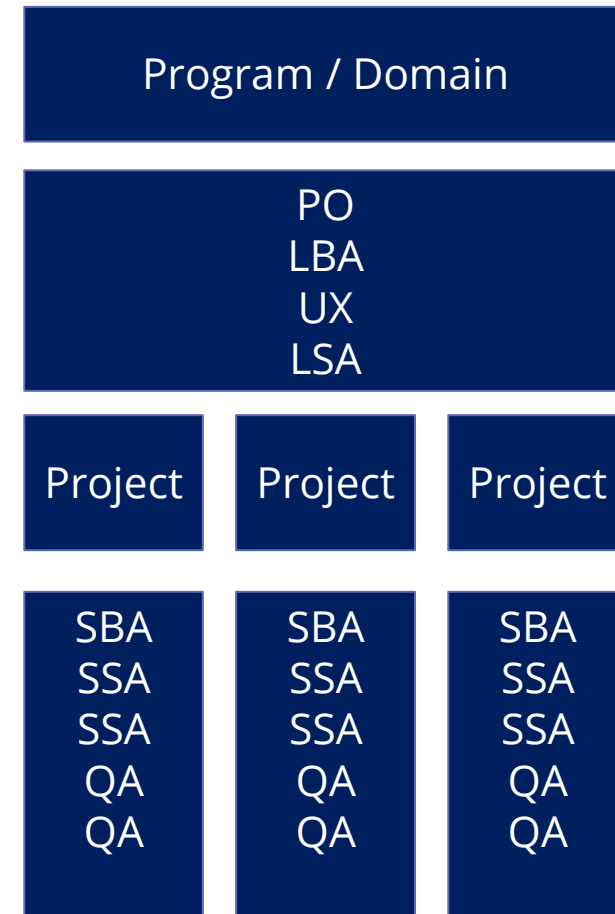
- Assign the modules to the domain with most knowledge on the topic
- All teams can contribute by developing new functionality, but merge is done by the responsible team
- Have regular meetings with the teams on the modules
- New Modules and features governed by the COE team

# Team setup with COE and Re-Use

Traditional setup



With COE and Re-Use





Questions?