

# **Low-Code vs. Model-Driven Architecture**

---

Markus Reiter

January 12, 2021

supervised by Prof. Dr. Ruth Breu

# Outline

- Motivation
- Model-Driven Architecture
- Low-Code Architecture
- Criticisms
- Evaluation of Low-Code Tools
- Assessment of Low-Code Tools
- Findings & Future Work
- Conclusion

- What are the advantages and disadvantages of low-code tools do?
- Are they a viable alternative to model-driven or traditional development?
- When to choose one approach over the other?

# Model-Driven Architecture

- provides a set of guidelines for the structuring of specifications
- standardise on models in a given domain to reduce code duplication and speed up development
- code (fully or partially) generated from models, e.g. from UML diagrams
- aimed at developers who have good understanding of underlying programming languages

Example: Code Generation using Swagger

- API specification given in OpenAPI format
- API client is generated for the specified programming language
- support for new languages/frameworks can be added by implementing a new generator
- very easy to provide API clients for many languages with virtually no development effort

# Low-Code Architecture

- provides pre-built application components
- graphical user interface for creating both the application logic as well as the user interface
- typically aimed more at end-users rather than developers
- no-code: targeted completely towards end-users

- Model-Driven Architecture
  - UML diagrams lack details included in the code itself
  - “the Code is the design” - Should models be derived from code instead of code from models?
- Low-Code Architecture
  - unsuitable for implementing scalable and mission-critical applications
  - increase in unsupported applications built by shadow IT, i.e. applications which are not controlled by a company's IT department
- Do these approaches actually make development easier and cheaper?

# Evaluation of Low-Code Tools

- Find low-code tools in the following categories:
  - open-source
  - developed by well-known company
  - developed by unknown company
  - old/well-established platform
  - new/unestablished
- Set up each tool
- Build a test application (TODO List) with each tool
  - add/edit items
  - mark items as done



## Open Standard Business Platform (OSBP)

- open-source
- plug-in for the Eclipse IDE developed since 2016
- community version of the commercial OS.bee product developed by COMPEX
- latest version over one year old
- does not work with latest version of the Eclipse IDE

## Corteza Low Code

- open-source
- part of the Corteza Project initiated by Crust Technology in 2019
- the Corteza Project includes a CRM solution built on top of Corteza Low Code, among other things
- web-based platform
- test by signing up with a GitHub or Google account or by deploying it locally using Docker

- made for building record-based management applications
- process for building the test application:
  1. create application namespace
  2. create module for TODO List records, specifying the necessary fields (status, title, body)
  3. create page and add a list block linked to the module

# Corteza Low Code: Editor

Corteza

Messaging

CRM Suite

+

CRM Modules Pages Charts

Lenny Horstink

Public pages

Create a new page:

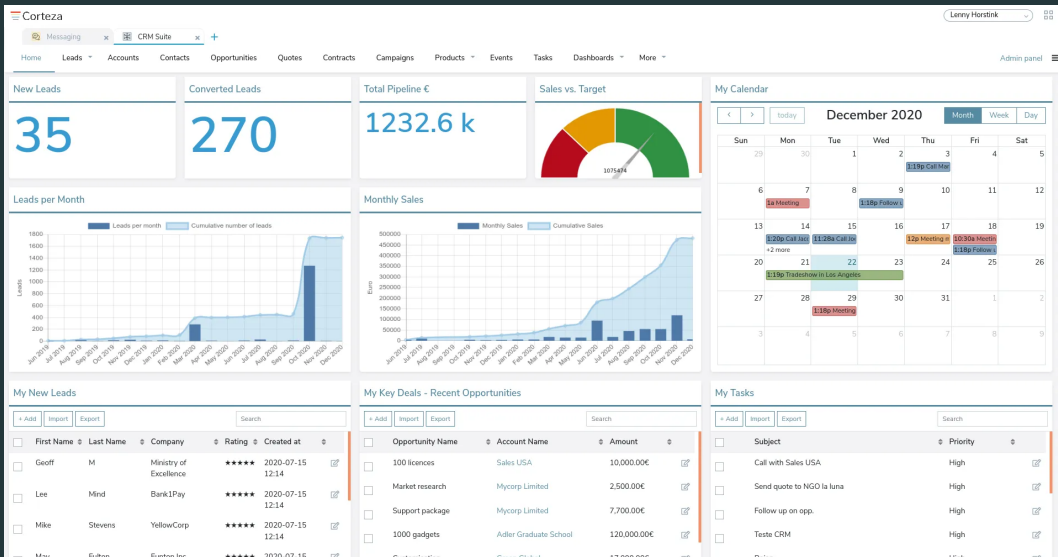
Page title

Create

List of pages

Home	Visible	Page builder	
Leads	Visible	Page builder	
Record page for module "Lead"	Record page for Lead	Page builder	
All New Leads	Visible	Page builder	
My Leads	Visible	Page builder	
Accounts	Visible	Page builder	
Record page for module "Account"	Record page for Account	Page builder	
Record page for module "AccountContactRole"	Record page for AccountContactRole	Page builder	
Case Studies	Not visible	Page builder	
Record page for module "CaseStudies"	Record page for CaseStudies	Page builder	
Contacts	Visible	Page builder	
Record page for module "Contact"	Record page for Contact	Page builder	
Opportunities	Visible	Page builder	
Record page for module "Opportunity"	Record page for Opportunity	Page builder	
Record page for module "OpportunityLineItem"	Record page for OpportunityLineItem	Page builder	
Record page for module "OpportunityCompetitor"	Record page for OpportunityCompetitor	Page builder	
Record page for module "OpportunityContactRole"	Record page for OpportunityContactRole	Page builder	
Record page for module "Offer files"	Record page for	Page builder	

# Corteza Low Code: Application



## Oracle APEX (Application Express)

- commercial
- initially released as Oracle Flows in 2000
- web-based platform
- test by signing up for an Oracle Cloud account or by requesting an APEX workspace

# Oracle APEX (Application Express)

- process for building the test application:
  1. create database table for TODO List items  
(requires basic SQL knowledge)
  2. create new blank application
  3. add list view to home page and select  
the corresponding database table
  4. add new form page (dialog style) and select  
the corresponding database table
  5. add a button to the home page that opens the form page
  6. add a dynamic action that refreshes the list view  
when the form dialog is closed

# Oracle APEX: Page Designer

The screenshot displays the Oracle APEX Page Designer interface for Application 81611. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', 'Team Development', and 'App Gallery'. A green status bar at the top right indicates 'Changes saved' and shows the user 'Markus R.'. The main workspace is titled 'Page Designer' and shows a page layout with a 'Home' region containing a 'Todo List' region. The 'Todo List' region has a 'CREATE' button. The right sidebar contains configuration options for the selected region, including 'Identification', 'Layout', and 'Appearance'. The 'Layout' section shows the 'Sequence' as 10 and the 'Region' as 'Todo List'. The 'Appearance' section shows the 'Button Template' as 'Text' and the 'Hot' status as 'On'. The bottom of the interface shows a 'Regions' tab with a grid of available components like Breadcrumb, Calendar, Cards, Chart, Column Toggle Report, Faceted Search, Form, Help Text, Interactive Report, List, List View, PL/SQL Dynamic Content, Region Display Selector, Static Content, Tree, and URL.

**Navigation Bar:** APEX | App Builder | SQL Workshop | Team Development | App Gallery

**Status Bar:** ✓ Changes saved | Markus R. | X

**Page Designer:** Application 81611 \ Page Designer

**Left Sidebar:**

- Page 1: Home
  - Pre-Rendering
  - Regions
    - Content Body
      - Todo List
        - Columns
          - Column Groups
          - Saved Reports
          - Region Buttons
            - CREATE
          - Dynamic Actions
            - New task - Dialog closed
              - True
                - Refresh
              - False
    - Post-Rendering

**Page Layout:**

- Home
  - PAGE HEADER
  - PAGE NAVIGATION
  - BREADCRUMB BAR
  - BEFORE CONTENT BODY
  - CONTENT BODY
    - Todo List
      - PREVIOUS
      - ITEMS
      - REGION CONTENT
        - RIGHT OF INTERACTIVE REPORT SEARCH BAR
          - CREATE
      - SUB REGIONS
      - NEXT
  - FOOTER
  - INLINE DIALOGS

**Right Sidebar:**

- Button
  - Filter
- Identification
  - Button Name: CREATE
  - Label: New task
- Layout
  - Sequence: 10
  - Region: Todo List
  - Button Position: Right of Interactive Report Search Bar
- Appearance
  - Button Template: Text
  - Hot: On
  - Template Options: Use Template Defaults
  - CSS Classes
  - Icon

**Regions:**

- Breadcrumb
- Calendar
- Cards
- Chart
- Column Toggle Report
- Faceted Search
- Form
- Help Text
- Interactive Report
- List
- List View
- PL/SQL Dynamic Content
- Region Display Selector
- Static Content
- Tree
- URL



# Oracle APEX: Application

≡

TODO List

me@reitermark.us

Q

Go

Actions

New task

Status	Title	Body
Done	Write paper	
Done	Create presentation	
Outstanding	Proof-read paper	

1 - 3

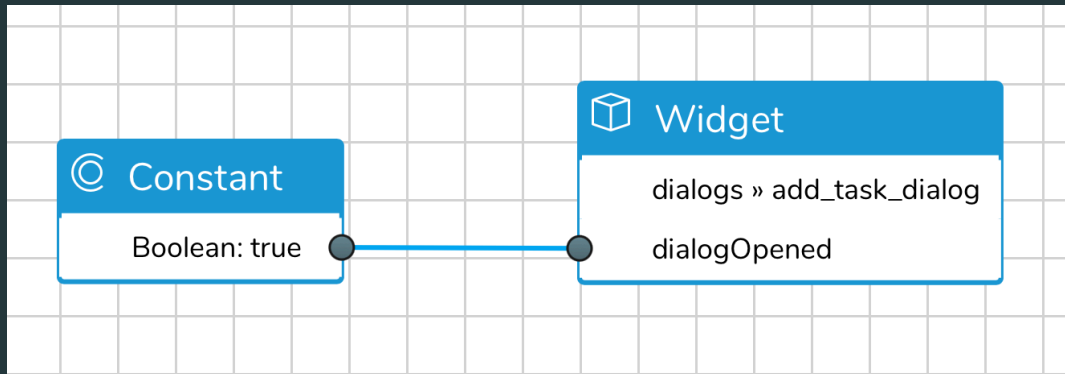
- commercial
- initially released by iTiZZiMO in 2012
- web-based platform
- test by using the Simplifier Playground (data is wiped every day) or by requesting a Simplifier test instance

- process for building the test application:
  1. create database connector (SQLite)
  2. create database schema for TODO List items and deploy to connector
  3. add list view and button to home page
  4. add new page containing a form with input fields and button
  5. create processes for
    - loading items into list view
    - submitting the form page
    - opening the form page with the button

## Simplifier: Process



## Simplifier: Process



# Simplifier: Application Editor

The screenshot displays the Simplifier Application Editor interface for an application named "Application ITIZ\_Template\_SQL\_ShoppingList". The interface is divided into several sections:

- Top Bar:** Includes the Simplifier logo, the title "Application Editor", a user profile "demo demo", and a "Verlassen" (Logout) button.
- Navigation Tabs:** "Designer" (active), "Prozess", "Data Workbench", "Tests", and "Andere".
- Left Panel (Project Explorer):**
  - Screens:** A tree view showing the application structure. The "Main" screen is selected.
  - Start Screen:** A list of available screens: "Splash", "Dialogs", and "Messages".
  - Others:** A list of other components: "custom-Header", "subHeader", "ScreenContent", "Main\_Table\_Items\_Products", "Items", "Main\_ColumnListItem\_Products", "layoutData", "dragDropConfig", "cells", "swipeContent", "headerToolBar", "Main\_Toolbar\_Products", "content", "dragDropConfig", "layoutData", "columns", "Main\_Column\_Product", "layoutData", "dragDropConfig", "footer", "header", "Main\_Column\_Amount", "layoutData", and "dragDropConfig".
- Center Panel (Design View):** A visual representation of the "Main" screen. It features a dark header bar with the title "Main", a search bar, and a table with columns "Product" and "Amount". The table is currently empty, displaying "No data".
- Right Panel (Properties):** A list of properties for the selected "Main" screen, including "General", "Appearance", "Data", "Misc", and "Others". The "Appearance" section is expanded, showing properties like "showFooter", "showHeader", "showNavBarButton", and "showSubHeader", each with a toggle switch and a link icon.

# Simplifier: Application

Main

Product

Amount

Bread

2

Apple Juice

3

Tooth Paste

1

- commercial
- founded in 2005 as a subsidiary of Siemens
- web-based platform (Mendix Studio) and Windows application (Mendix Studio Pro) with advanced features
- test by signing up for a regular account which allows hosting unlimited applications (with 1GB of memory and 0.5GB of storage per application)



- process for building the test application:
  1. create blank application
  2. add list view and button to home page, which prompts the user to select or create a new data source
  3. create data source for TODO List items
  4. add new form page linked to the corresponding data source
  5. add button to home page that opens the form page

# Mendix Studio

The screenshot displays the Mendix Studio IDE interface. The main workspace shows a 'TODO List' application. A 'New task' button is visible, and below it is a list of tasks, each with a placeholder for a title, status, and body. The right sidebar shows the 'Properties' panel for the selected 'BUTTON' widget. The 'On Click Action' section is expanded, showing a 'Create Object' action. The 'Entity' is set to 'task (MyFirstModule)' and the 'Page' is 'edit\_task (MyFirstModule)'. The 'General' section shows the 'Caption' as 'New task' and the 'Icon' as 'Select icon'. The 'Render Mode' is set to 'Button' and the 'Style' is 'Primary'. The 'Conditional Visibility' section is also visible, showing an 'Attribute-Based' condition.

**TODO List**

Button task

New task

{title}  
{status}  
{body}

{title}  
{status}  
{body}

{title}  
{status}  
{body}

{title}  
{status}  
{body}

{title}  
{status}  
{body}

**Properties**

**BUTTON**

**Events**

On Click Action

Nothing Page Microflow More

Create Object

Entity

task (MyFirstModule)

Page

edit\_task (MyFirstModule)

**General**

Caption

New task

Icon

Select icon

Render Mode

Button

Style

Primary

**Conditional Visibility**

Attribute-Based

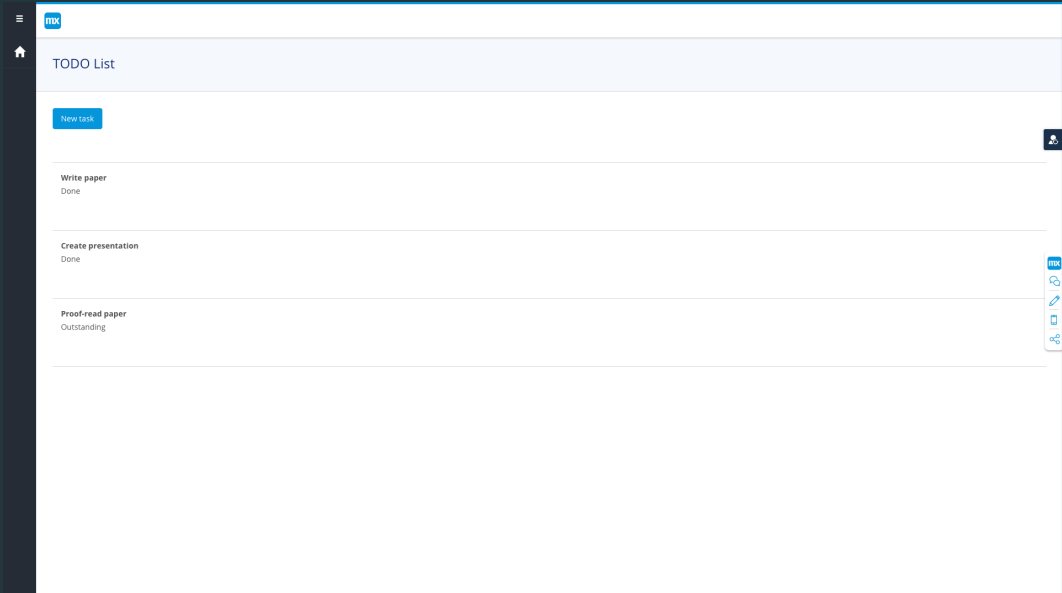
Select attribute

To use an attribute, move the widget into a data view or a list view.

Wrap with a new data view

Delete

# Mendix: Application



# Unique Features

- **Oracle APEX**
  - Creation of applications from existing data, e.g. CSV files
- **Mendix**
  - AI-assisted wizard for creating custom workflows (microflows)
  - publishing as native mobile applications for iOS and Android

# Assessment of Low-Code Tools

- ease of use
- customisability
- portability
- scalability
- suitability for mission-critical applications

# Assessment of Low-Code Tools

	<b>OSBP</b>	<b>Corteza</b>	<b>APEX</b>	<b>Simplifier</b>	<b>Mendix</b>
ease of use	N/A	high	medium	medium	high
customisability	N/A	low	medium	medium	high
portability	N/A	medium	low	medium	medium
scalability	N/A	medium <sup>1</sup>	high	high	high
mission-criticality	low	high	high	medium <sup>2</sup>	high

---

<sup>1</sup>medium in general, high for certain types of applications, e.g. management applications

<sup>2</sup>medium only due to the problems encountered, high otherwise

## Findings & Future Work

- very small number of open-source low-code tools
  - assumption: open-source community mostly consists of developers, so there is no need/demand for low-code platforms
  - future work: investigate low-code other types of platforms and compare their presence in the open-source vs. the commercial space
- more streamlined user interface in the more mature products like Oracle APEX and Mendix
- lack of portability is a valid concern for low-code tools, stored data may be the only element of a platform that is portable by using

# Conclusion

- low-code platforms are a valid alternative to model-driven development
- low-code platforms are not as flexible and limited in the ways they can be extended
- choice between low-code and model-driven architecture:
  - highly dependent on which low-code platform is used, similar to choosing a programming language or framework for a particular task
  - highly dependent on the application requirements, low-code well-suited for management applications (e.g. process management, CRM)



**Questions?**