

# Software Development

ART or ENGINEERING?

# Agenda

---

- ES@Pesc
- Software Development
- Software Engineering
  - Process
  - Project
- AgileKip: SE in Practice

# SE@PESC

---

**Profa. Ana Regina** - Software Processes, Software Quality and Maturity Models (MPS.Br)

**Profa. Claudia Werner** - Software Reuse, Software Ecosystems and SE Education

**Prof. Claudio Miceli** - Internet of Things (IoT), Smart Grid and Security

**Prof. Guilherme Horta Travassos** - Empirical Software Engineering and Evidence-based Software Engineering

**Prof. Toacy C. de Oliveira** - Software Process Representation and Analysis, Process Mining and BPM

# Increasing Demand

- ❑ Software Systems are reshaping society.....
- ❑ People...
  - ❑ Are highly connected
  - ❑ Take care of their own stuff
    - ❑ Buy tickets, do banking, get paid, find places, book hotels
  - ❑ Demand more and more information (awareness, transparency)
- ❑ Organizations (Companies, Governments, NGOs,...)
  - ❑ Are also highly connected
  - ❑ Need to do more with less (be efficient)
  - ❑ Need to be transparent and responsible
- ❑ Both people and organizations generate tons of information!
- ❑ Things need software 😊

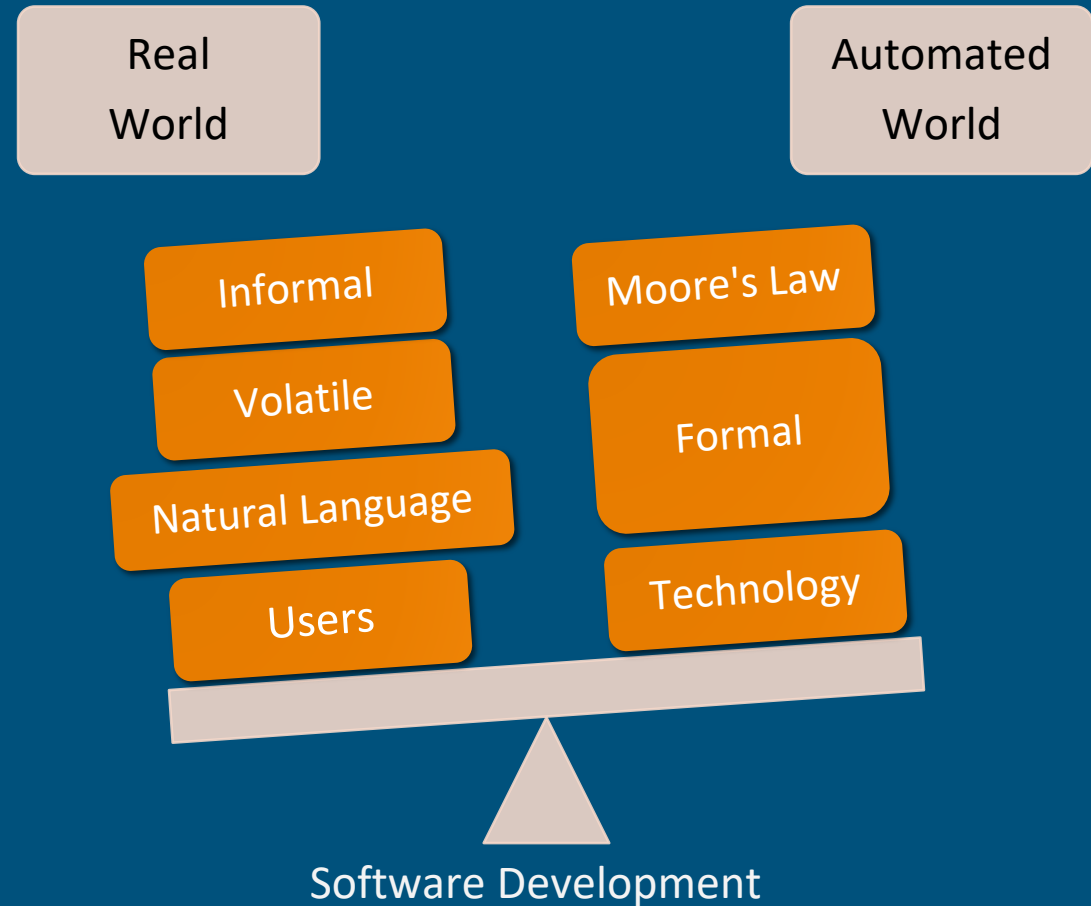
Internet  
Of  
Things

Smart  
Things

Wearable  
Things

# Software Development: Context

In practice, Software Development practitioners battle between the "Real World" and the "Automated World" to create bespoke (semi-)automated systems that usually mimic real world procedures.



# Who is winning?

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

| SIZE                 | METHOD    | SUCCESSFUL | CHALLENGED | FAILED |
|----------------------|-----------|------------|------------|--------|
| All Size Projects    | Agile     | 39%        | 52%        | 9%     |
|                      | Waterfall | 11%        | 60%        | 29%    |
| Large Size Projects  | Agile     | 18%        | 59%        | 23%    |
|                      | Waterfall | 3%         | 55%        | 42%    |
| Medium Size Projects | Agile     | 27%        | 62%        | 11%    |
|                      | Waterfall | 7%         | 68%        | 25%    |
| Small Size Projects  | Agile     | 58%        | 38%        | 4%     |
|                      | Waterfall | 44%        | 45%        | 11%    |

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

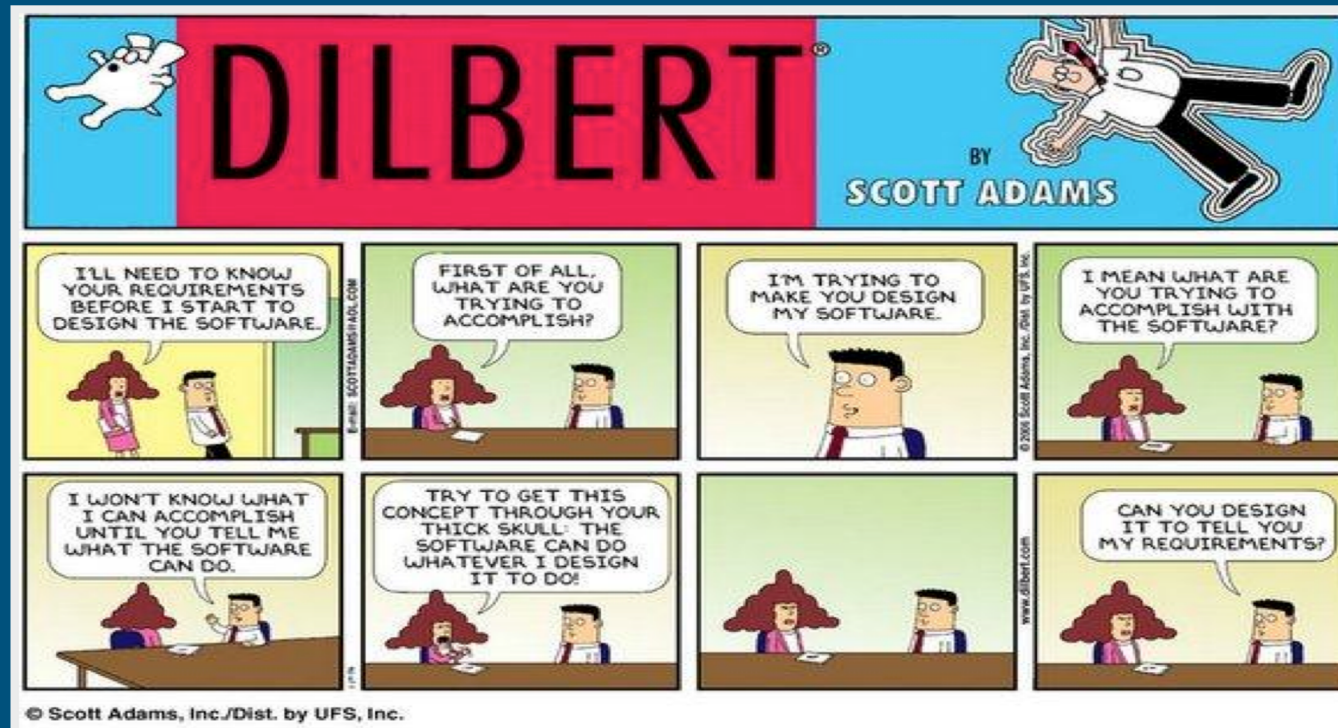
# Why developing software is so complex ?

---

USER VS TECHNOLOGY VS "US"

# User!

It might be not so obvious to understand their needs... ..and their domains (banking, insurance, HR, education, logistics, healthcare, etc..).



# Technology!

---

## Client Side Options

---



# +Technology .....

## Server Side Options



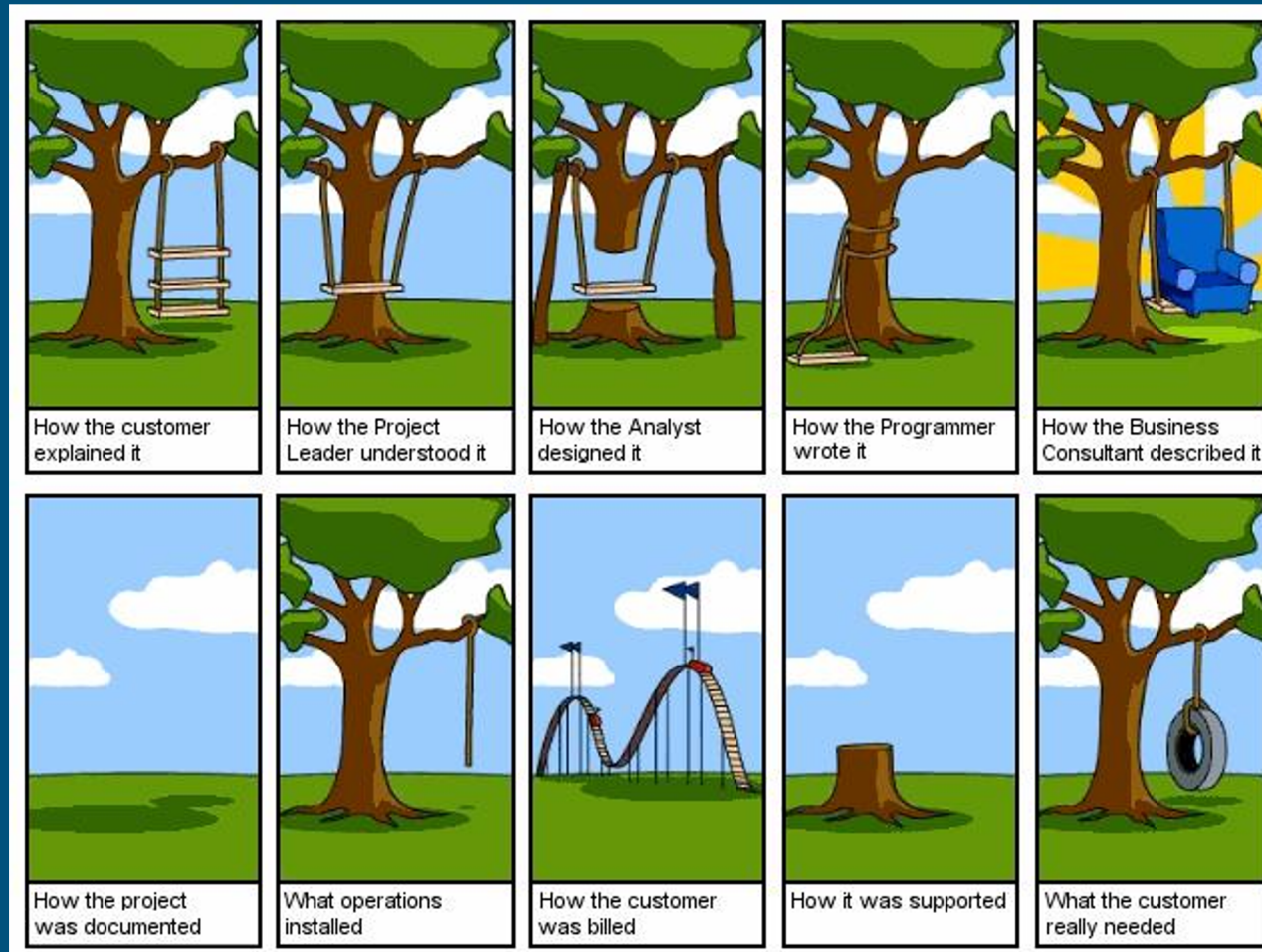
## Deployment Options



## CI/CD Options



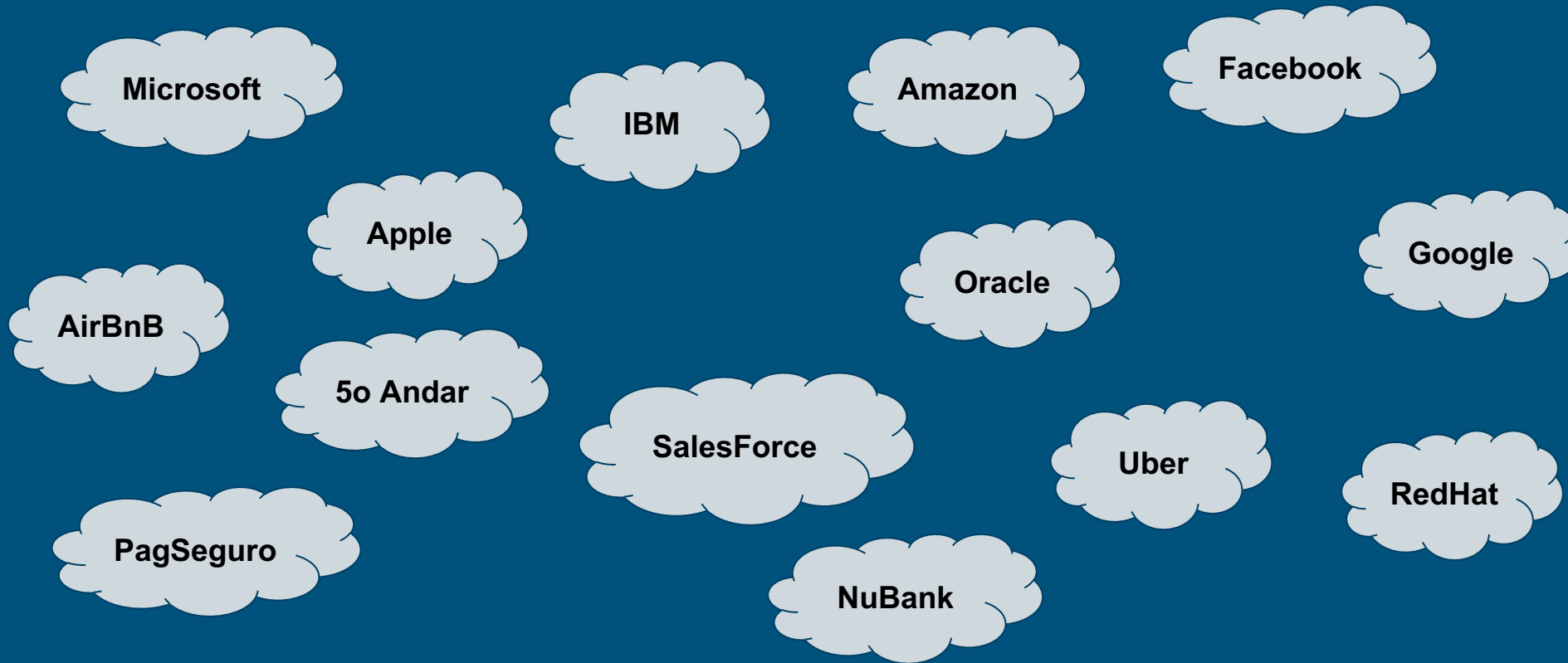
# Us!!



Project Leader  
Analyst  
Programmer  
Business Consultant  
Operations

# It's not all lost!!!

---



# The Solution!!

---

**Software Engineering** — According to IEEE's definition, software engineering can be defined as the application of a **systematic, disciplined, quantifiable** approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software.

- systematic: *def.* done or acting according to a fixed plan.
- disciplined : *def.* showing a controlled form of behavior or way of working.
- quantifiable : *def.* express or measure the quantity.

# The Solution in practice 😊

---

1. Apply Divide and Conquer
2. Apply Abstraction
3. Gather a bunch of well educated and experienced people with different skills and backgrounds...and don't forget the users!!!
4. Understand, Implement, Test, Document, Deploy an increment ( set of divided bits)
5. Goto 1 until run out of budget or the boss says it is done!

Do all this with a plan in mind!!!

# Software Development Processes

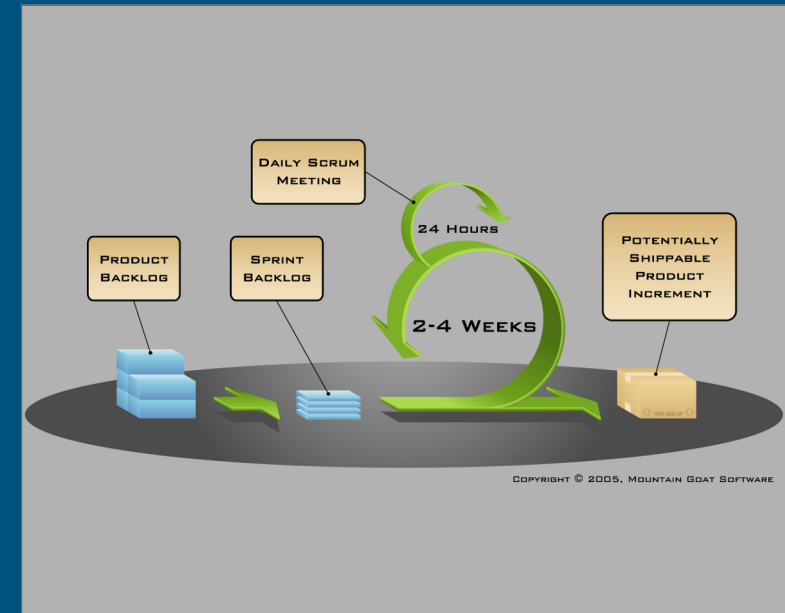
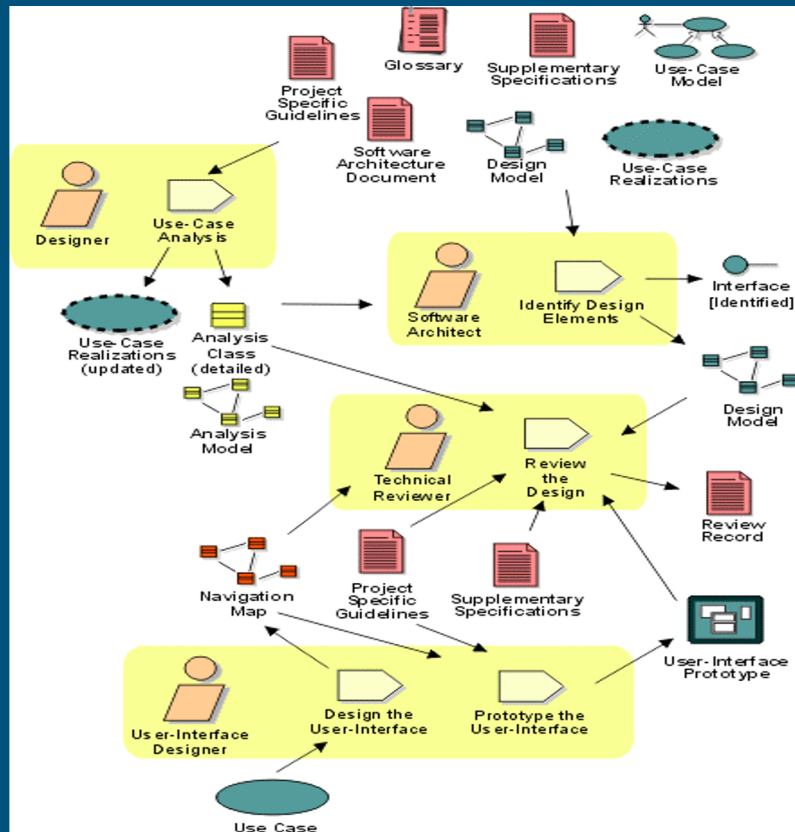
---

A software process is a set of interrelated **activities** and **tasks** that transform **input work products** into **output work products**. At minimum, the **description** of a software process includes required inputs, transforming work activities, and outputs generated.

-Swebok, 2014

Software Development Processes (SDPc) attempt to represent common and successful practices that together facilitate how software systems are built or maintained. Processes typically come from independent "entities" such as ISO, CMM, MPS.Br and SCRUM Manifesto or from companies such as the Rational (Rational Unified Process).

# Example – A Process



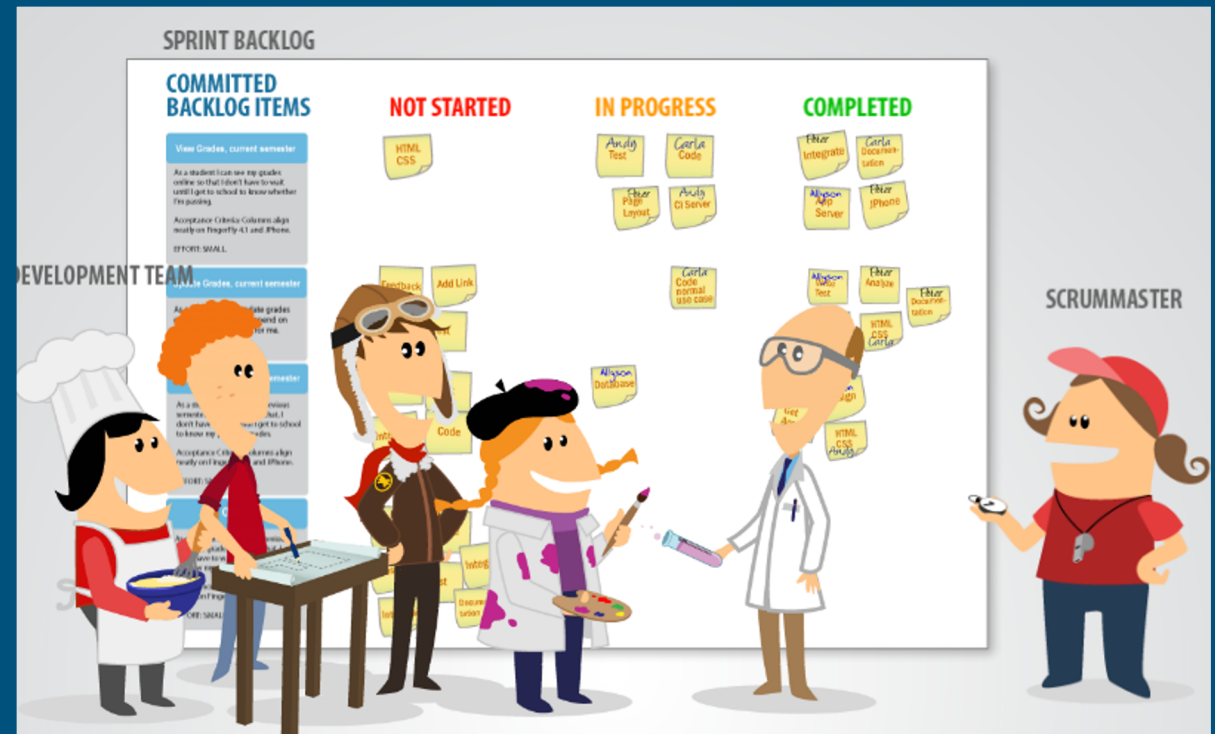
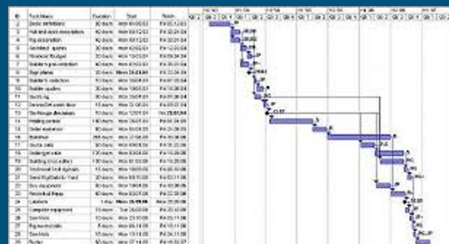
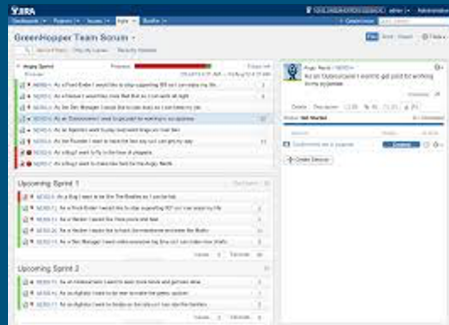
# Software Development Projects

---

Software Development **Projects (SDPj)** using Software Engineering principles and concepts:

- Leverage on **plans** (to be systematic)
- Measure (time, budget, satisfaction, errors, etc.) what is happening
- Control (execution, implementation,...) what is happening
- ...and...
- Gather volatile and unstable information from users and the like (manuals, procedures, laws, etc.)
- Convert informal information to formal representations (NL => UML, BPMN, Java, )
- Add technology to the mix (web, mobile, tv, cloud, etc.)

# Example – A Plan in SDPjs



# Art or Engineering?

---

Software Development is still a mix of Art and Engineering, and it will stay like this as long as people play the central role.

**But who cares!!!  
Software is needed everywhere and  
this discussion is pointless.**

# AgileKip -

A "small" example of  
engineering in practice

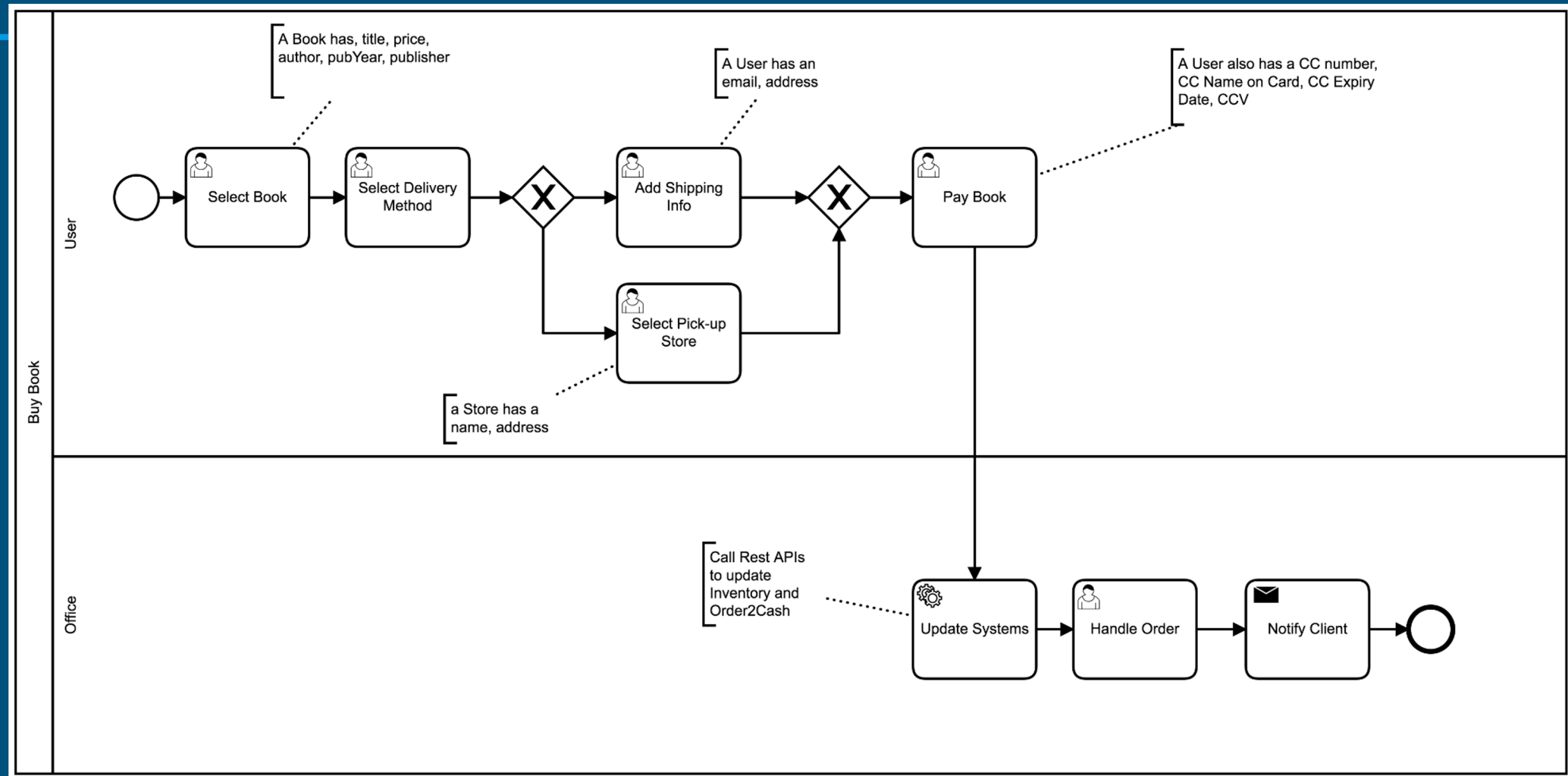
AgileKip – A R&D Agenda intended to investigate, develop and communicate state-of-the-art research and also create an ecosystem based on open-source tools to help practitioners navigating the Agile and Knowledge-Intensive Process Lifecycle ([agilekip.com](http://agilekip.com))

- Project 1 – AgileQube – Agility Assessment
- Project 2 - PACA – Process Aware Conversational Agent
- Project 3 - Intelligent Process Automation Platform

# Process Automation

- ❑ The adoption of software systems to support the execution of normative and explicit workflows.
- ❑ Promote a standardized way to handle "things".
- ❑ Examples
  - ❑ Procurement
  - ❑ Order2Cash
  - ❑ Employee Onboarding
  - ❑ Planning a Trip
- ❑ It is bootstrapped by a process model, typically BPMN.

# Process Model - BuyBook

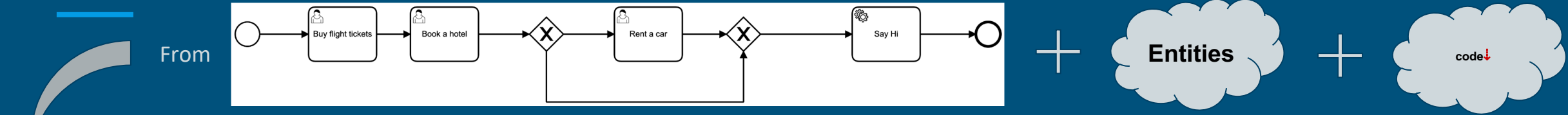


# Problem

---

- ❑ How to simplify developing Process Aware Information Systems?
  - ❑ avoiding vendor lock in;
  - ❑ based on state-of-the-art and proven technologies;
  - ❑ using an approach where most of the system is automatically generated;
  - ❑ using a reference architecture that is highly configurable;
  - ❑ that can be available on premise or as a service;
  - ❑ allowing bespoke extensions (see What's next).

# Solution Rationale



Generate

To

Create or edit a Travel Plan Process

#1001 - Travel Plan Process

### Planning a Trip

*Planning a Trip*

Planning a trip can feel like both the most exciting AND the most overwhelming part of travel. Sure, creating a Pinterest board full of gorgeous pictures is fun, but then you start trying to figure out the actual logistics and it hits you ... oh god, this is SO MUCH WORK. Cue panic attacks. *Wow, I totally just described our wedding planning process, too. Anyone else?*

But planning a trip doesn't have to be the source of constant anxiety and overwhelm! Through the years, I've perfected my trip planning process, and I've got it down to a fine science.

Er, well, maybe not a science, but sort of like a general step-by-step outline. **Breaking down each of the steps to planning a trip helps me cut through the confusion and tackle stuff in a practical, down to earth way.**

So, this travel plan process aids you in this mission and consists of the following tasks:

1. Buy Flight Tickets
2. Book a Hotel
3. Rent a Car

Name

Start Date

End Date

Status: **ACTIVE**

Camunda Deployment Id: 1

Camunda Process Definition Id: TravelPlanProcess:1:3

Bpmn Process Definition Id: TravelPlanProcess

Task Execution

#1306 - Buy flight tickets **Assigned**

### Buying the Flight Tickets

Airfare can easily be the largest expense of your trip. Expensive plane tickets mean you need to choose a more affordable destination or spend less money at your vacation stop to stay within your spending limit. If the idea of booking cheap flights sounds appealing to you, these 10 tips will help your air travel become more reasonable.

1. Book Early for Cheap Flights
2. Set Price Alerts
3. Be Flexible
4. Book a Connecting Flight
5. Consider Discount Airlines
6. Use More Than One Travel Portal
7. Browse Airline Specials
8. Use an Airline Credit Card to Avoid Baggage Fees
9. Use Award Miles
10. Use Your Credit Card Travel Credits

Name:

Start Date:

End Date:

Airline Company Name:

Airline Ticket Number:

Created at: Tuesday, May 4, 2021, 5:01 PM

Assignee: admin

Process Definition: [Travel Plan Process](#)

Process Instance: [TravelPlan#1253](#)

Task Definition Key: TaskFlight

Camunda Task Id: 1759

Camunda Execution Id: 1756

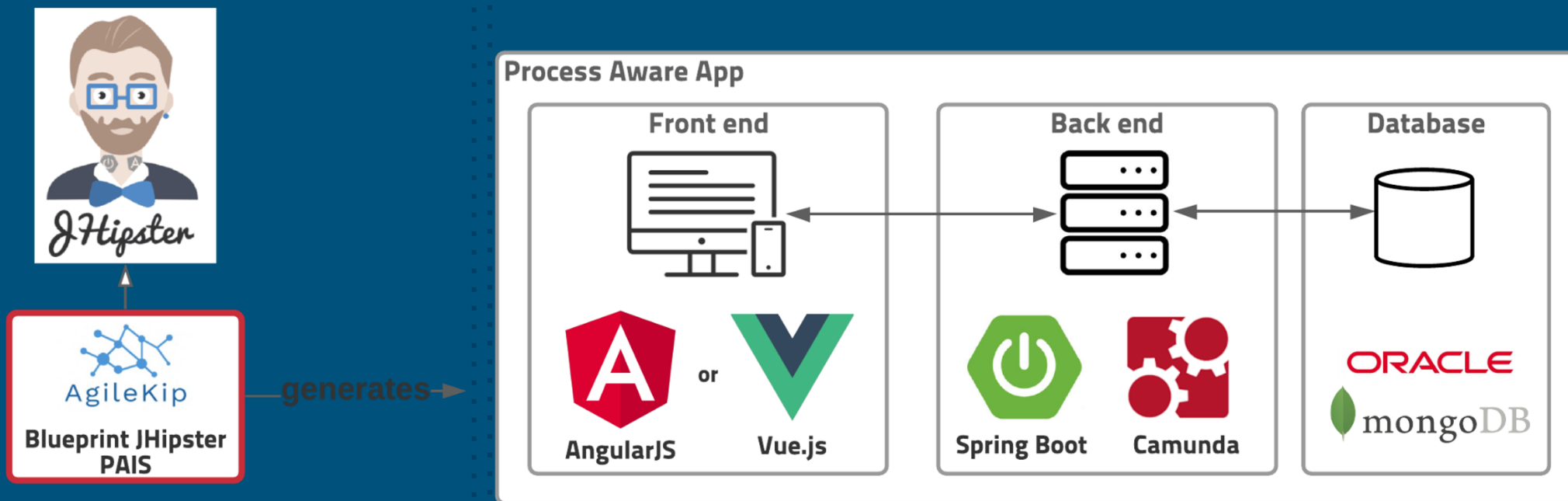
Support for:

- CRUDs
- User Management
- Rest EndPoints
- Internationalization
- Database Management
- Microservices support
- & more...

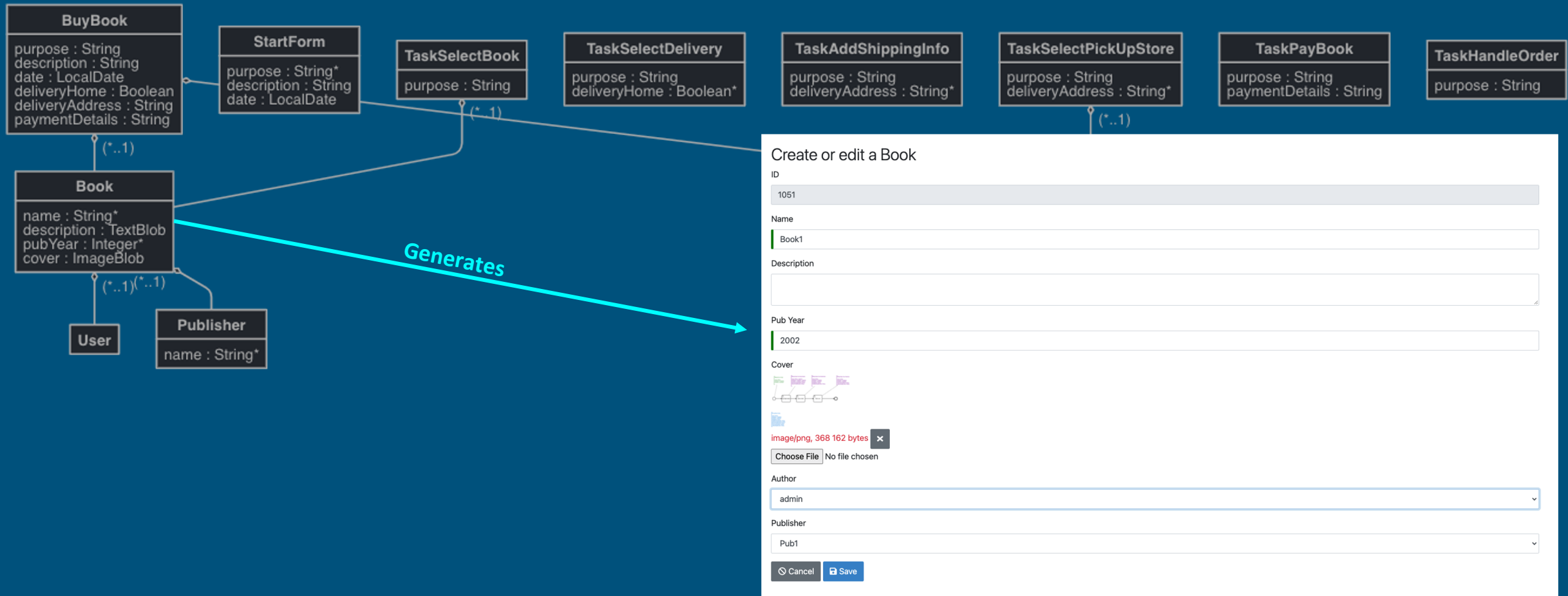
# The Platform

PAIS JH-Blueprint

Reference Architecture



# Specifying Entities



# Specifying Forms

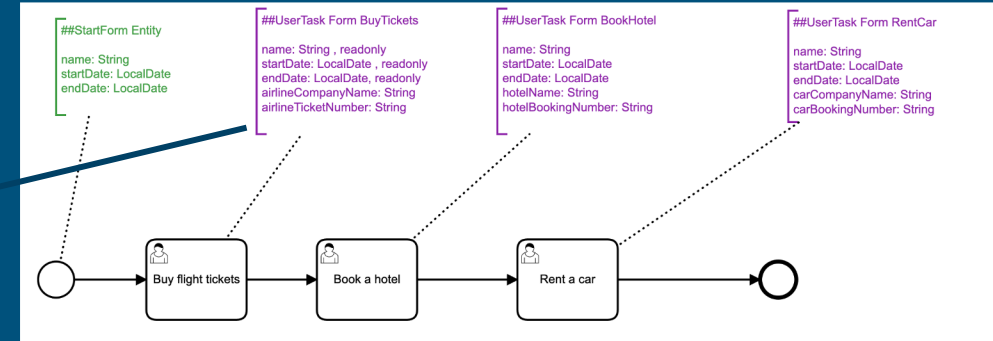
## ##TaskForm

**name:** String , readonly  
**startDate:** LocalDate , readonly  
**endDate:** LocalDate, readonly  
**airlineCompanyName:** String  
**airlineTicketNumber:** String

## #TravelPlan Entity

**name:** String  
**startDate:** LocalDate  
**endDate:** LocalDate  
**hotelName:** String  
**hotelBookingNumber:** String  
**airlineCompanyName:** String  
**airlineTicketNumber:** String  
**carCompanyName:** String  
**carBookingNumber:** String

Generates



## Task Execution

#1306 - Buy flight tickets **Assigned**

### Buying the Flight Tickets

Airfare can easily be the largest expense of your trip. Expensive plane tickets mean you need to choose a more affordable destination or spend less money at your vacation stop to stay within your spending limit. If the idea of booking cheap flights sounds appealing to you, these 10 tips will help your air travel become more reasonable.

1. Book Early for Cheap Flights
2. Set Price Alerts
3. Be Flexible
4. Book a Connecting Flight
5. Consider Discount Airlines
6. Use More Than One Travel Portal
7. Browse Airline Specials
8. Use an Airline Credit Card to Avoid Baggage Fees
9. Use Award Miles
10. Use Your Credit Card Travel Credits

**Assignee**  
admin

**Process Definition**  
[Travel Plan Process](#)

**Process Instance**  
[TravelPlan#1253](#)

**Task Definition Key**  
TaskFlight

**Camunda Task Id**  
1759

**Camunda Execution Id**  
1756

**Name**  
Trpi1

**Start Date**  
2021-03-03

**End Date**  
2021-03-04

**Airline Company Name**

**Airline Ticket Number**

Created at: Tuesday, May 4, 2021, 5:01 PM

[Back](#) [Complete](#)

TLDR

# Demo

# Take aways

- Creating Modern Software Systems is hard.
    - ◆ It involves several participants, with different skill sets.
    - ◆ It involves ever changing technology.
    - ◆ It involves discovering domains that engineers are not used to.
  - Software Engineering is about organizing how software is built and maintained.
    - ◆ It defines practices on how information should flow between participants (engineers, customers, stakeholders, etc.).
    - ◆ It defines responsibilities that should be followed (manager, scrum master, tester, etc.).
    - ◆ It defines a workflow (or a ritual) to organize collaborative work.
-

# Take aways

---

Software Engineering is not a silver bullet but working in an organized way typically outweighs working in an ad-hoc manner.



# Thanks



[agilekip.com](http://agilekip.com)

[toacy@cos.ufrj.br](mailto:toacy@cos.ufrj.br)



[@toacyoliveira](https://twitter.com/toacyoliveira)



[www.linkedin.com/in/toacyoliveira](https://www.linkedin.com/in/toacyoliveira)

