Software Development

ART or ENGINEERING?
Agenda

● ES@Pesc
● Software Development
● Software Engineering
  ○ Process
  ○ Project
● AgileKip: SE in Practice
SE@PESC

Profa. Claudia Werner - Software Reuse, Games in SE 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 😊
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?

<table>
<thead>
<tr>
<th>SIZE</th>
<th>METHOD</th>
<th>SUCCESSFUL</th>
<th>CHALLENGED</th>
<th>FAILED</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Size Projects</td>
<td>Agile</td>
<td>39%</td>
<td>52%</td>
<td>9%</td>
</tr>
<tr>
<td></td>
<td>Waterfall</td>
<td>11%</td>
<td>60%</td>
<td>29%</td>
</tr>
<tr>
<td>Large Size Projects</td>
<td>Agile</td>
<td>18%</td>
<td>59%</td>
<td>23%</td>
</tr>
<tr>
<td></td>
<td>Waterfall</td>
<td>3%</td>
<td>55%</td>
<td>42%</td>
</tr>
<tr>
<td>Medium Size Projects</td>
<td>Agile</td>
<td>27%</td>
<td>62%</td>
<td>11%</td>
</tr>
<tr>
<td></td>
<td>Waterfall</td>
<td>7%</td>
<td>68%</td>
<td>25%</td>
</tr>
<tr>
<td>Small Size Projects</td>
<td>Agile</td>
<td>58%</td>
<td>38%</td>
<td>4%</td>
</tr>
<tr>
<td></td>
<td>Waterfall</td>
<td>44%</td>
<td>45%</td>
<td>11%</td>
</tr>
</tbody>
</table>

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

Client Side Options

- HTML5
- CSS3
- Bootstrap
- TypeScript
- Angular
- React
- Vue
- Redux
- Websockets
- Yarn
- Webpack
- Sass
- Browsersync
- Jest
- Protractor
+Technology .....!
Us!!

Project Leader
Analyst
Programmer
Business Consultant
Operations

Toacy C. Oliveira – agilekip.com
It's not all lost!!!
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
- 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 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)

- 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

Support for:
- CRUDs
- User Management
- Rest Endpoints
- Internationalization
- Database Management
- Microservices support
- & more...

Entities

Generate

From

To

Create or edit a Travel Plan Process

Planning a Trip

1. Buy Flight Tickets
2. Book a hotel
3. Rent a car

Task Execution

Buying the Flight Tickets

1. Book Early for Cheap Flights
2. Check Fares
3. Be Flexible
4. Book a Connecting Flight
5. Use a Travel Credit Card
6. Use an Airline Credit Card to Avoid Baggage Fees
7. Use Award Miles
8. Use Your Credit Card Travel Credits

Support for:
- CRUDs
- User Management
- Rest EndPoints
- Internationalization
- Database Management
- Microservices support
- & more...
The Platform
Specifying Entities

Create or edit a Book

<table>
<thead>
<tr>
<th>Name</th>
<th>Book1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Pub Year</td>
<td>2002</td>
</tr>
<tr>
<td>Cover</td>
<td></td>
</tr>
<tr>
<td>Author</td>
<td>admin</td>
</tr>
<tr>
<td>Publisher</td>
<td>Book1</td>
</tr>
</tbody>
</table>

Generate

BuyBook
- purpose: String
- description: String
- date: LocalDate
- deliveryHome: Boolean
- deliveryAddress: String
- paymentDetails: String

StartForm
- purpose: String
- description: String
- date: LocalDate

TaskSelectBook
- purpose: String
- description: String

TaskSelectDelivery
- purpose: String
- deliveryHome: Boolean
- deliveryAddress: String

TaskAddShippingInfo
- purpose: String
- deliveryAddress: String

TaskSelectPickUpStore
- purpose: String
- deliveryAddress: String

TaskPayBook
- purpose: String
- paymentDetails: String

TaskHandleOrder
- purpose: String
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

---

Toacy C. Oliveira – agilekip.com
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

@toacyoliveira

www.linkedin.com/in/toacyoliveira