

# End-User Development for Human-Robot Interaction: Approaches and Possibilities

**GIACOMO VAIANI**

*University of Pisa*  
*PhD in Computer Science*

**Mauriana Pesaresi seminar series**

February 02, 2024

# What is End User Development?


EUD is a set of methods, techniques, and tools that empower non-technical users to create, modify, or extend software artifacts.

# End User Development concepts



## End User Development

set of methods, techniques, and tools that empower non-professional users to create, modify, or extend software artifacts



# End User Development concepts



## End User Development

set of techniques that empower non-professional users to create or modify software artifact: it spans the **entire development**

## End User Programming

enables end-users to independently **create** programs, emphasizing their coding

# End User Development concepts



## End User Development

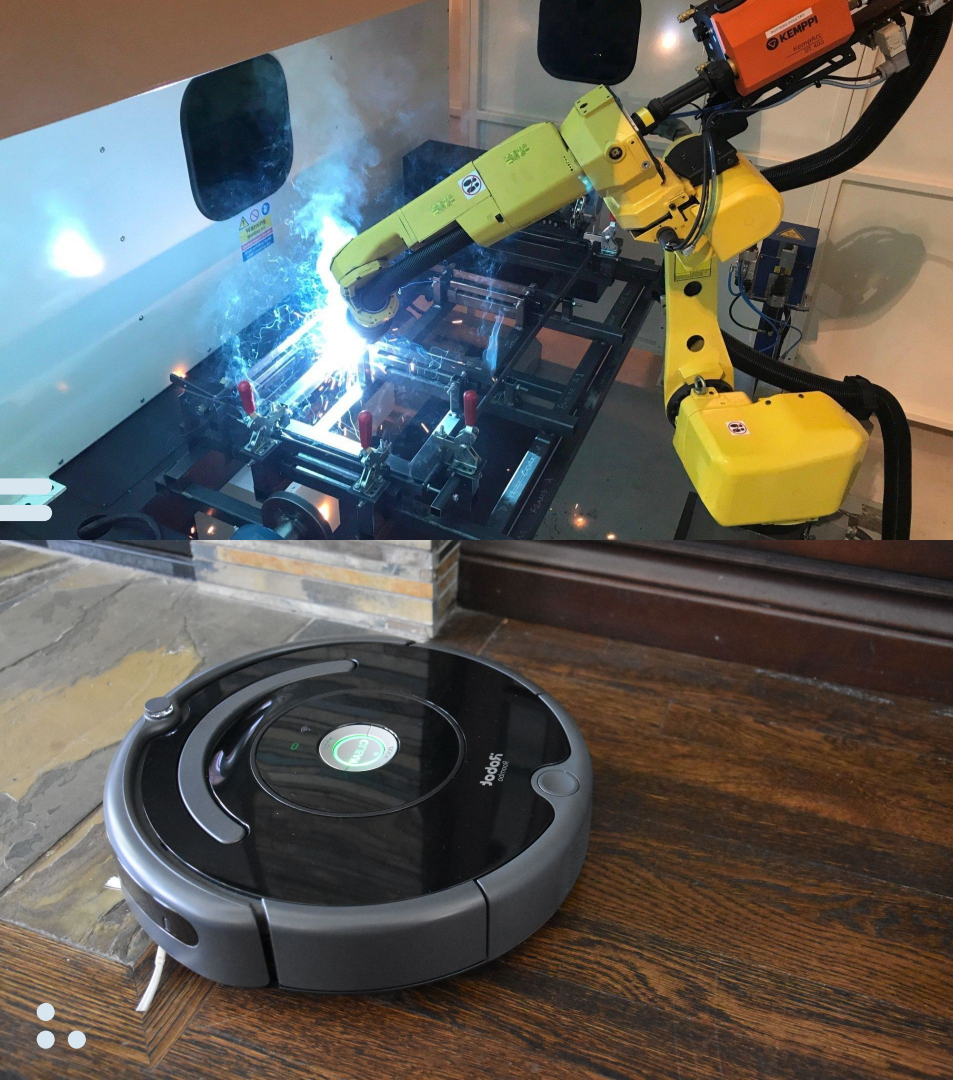
set of techniques that empower non-professional users to create or modify software artifact: it spans the **entire development**

## End User Programming

enables end-users to independently **create** programs, emphasizing their coding

## End User Software Engineering

involves activities that address **software quality** issues, ensuring the long-term sustainability of the software



**Why End-User  
Development for  
Robots?**







## ☐☐☐ GROCERY SHOP SCENARIO

activities like restocking items can be exploited by an **automated solution**

EUD in robots  
environments:  
use case example ☐☐☐



## ☹️ = GROCERY SHOP SCENARIO

activities like restocking can be exploited by an **automated solution**

due to possible variations, **human customization** should be possible

**EUD in robots  
environments:  
use case example**





# EUD for robots: use domains



## Industrial

manufacturing tasks and  
production optimization



## Non-Industrial

Social, domestic,  
education, assistive, etc.



# EUD Design Space

## Platform

Web, desktop, mobile, tangible, etc.

## Domain

Teaching, assistive, home, collaborative, etc.

## Event

and Event composition.  
User, environment, etc.

## Metaphor

Components, timelines, rules, puzzle etc.

## Style

Visual, demonstration, template, language, etc.

## Action

and Action composition.  
Interface, features, etc.

# EUD Design Space

## Programming Style

How the EUD environment enables users to create or modify an application.

some examples:

- demonstration
- natural language
- mix/aug. reality
- spreadsheet
- template
- text
- trigger action
- visual



# EUD Design Space

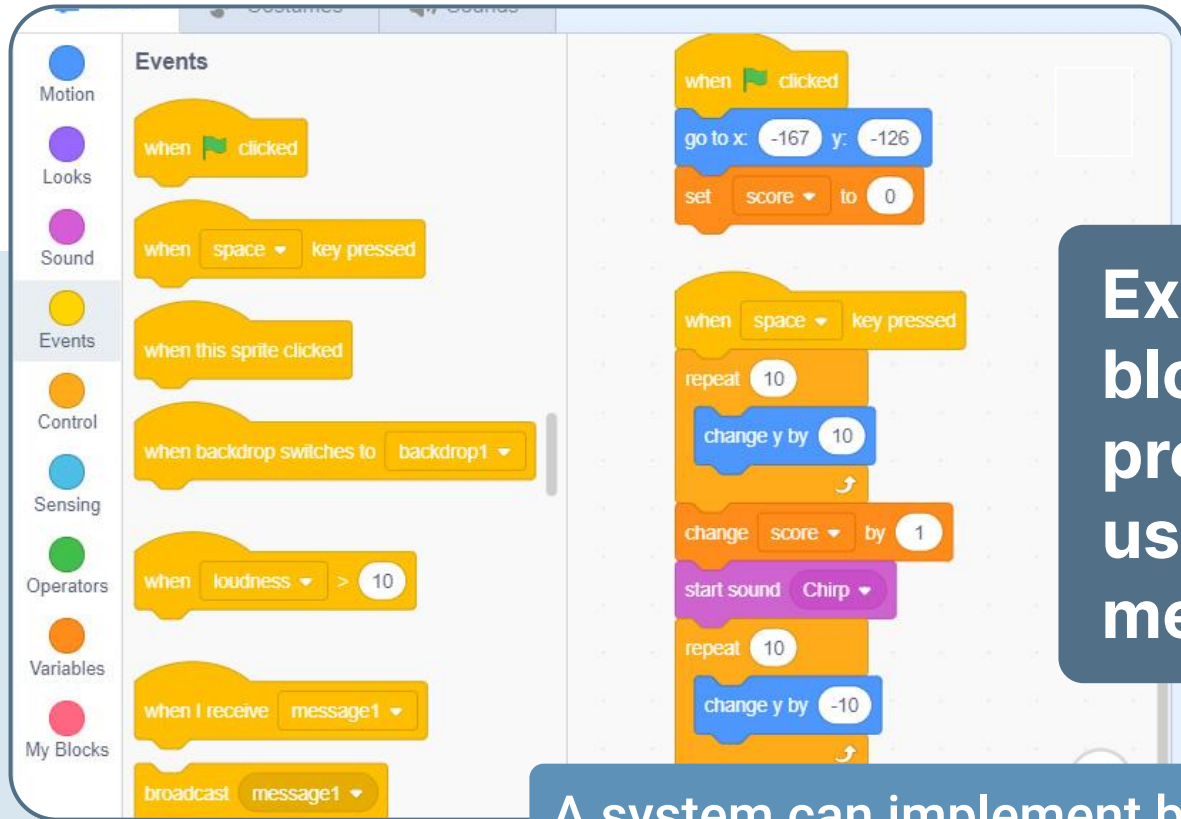
some examples:

- cards
- icons
- rules
- puzzle
- timeline
- tree

## Metaphor

Represent complex programming concepts providing users with understandable hints.





**Example of  
block-based  
programming  
using a puzzle  
metaphor**

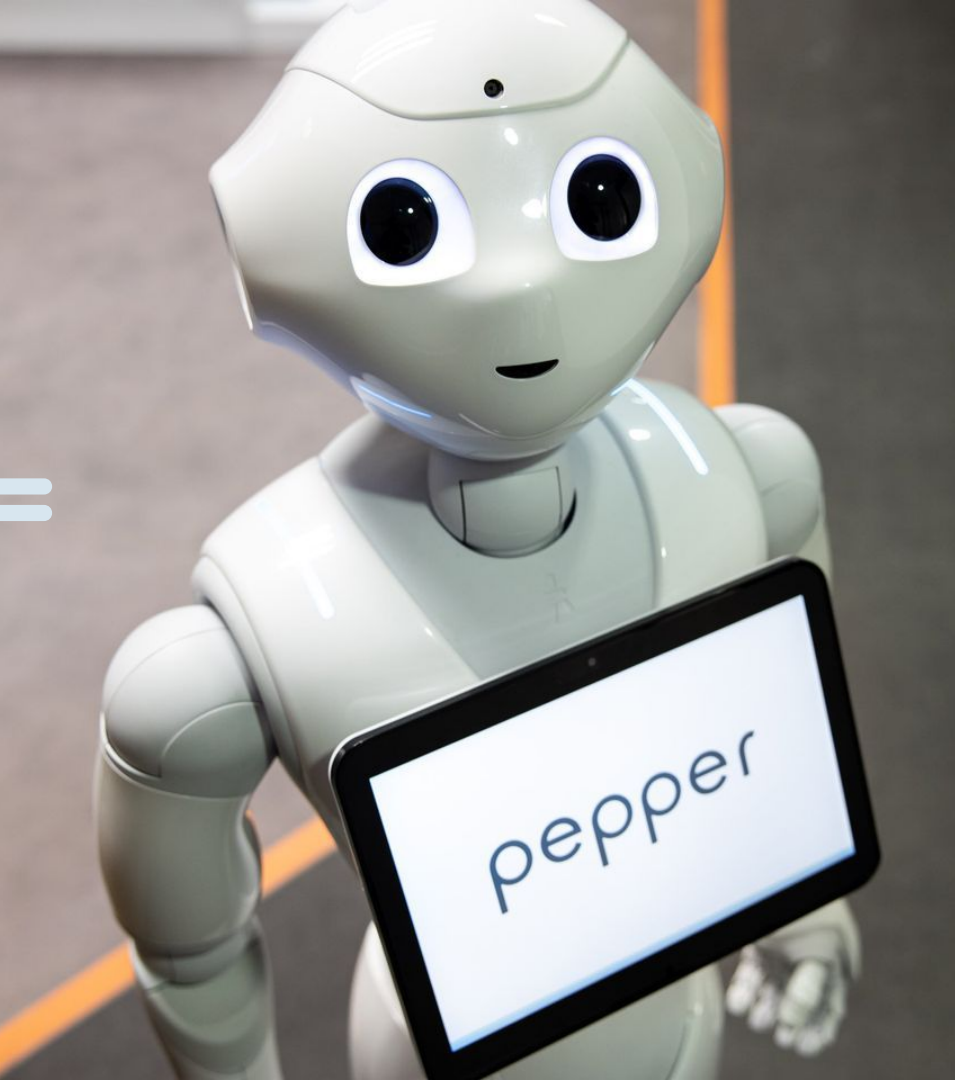
**A system can implement both a single style  
or a combination of them (multi-modal).**





# Research Direction and Open Problems

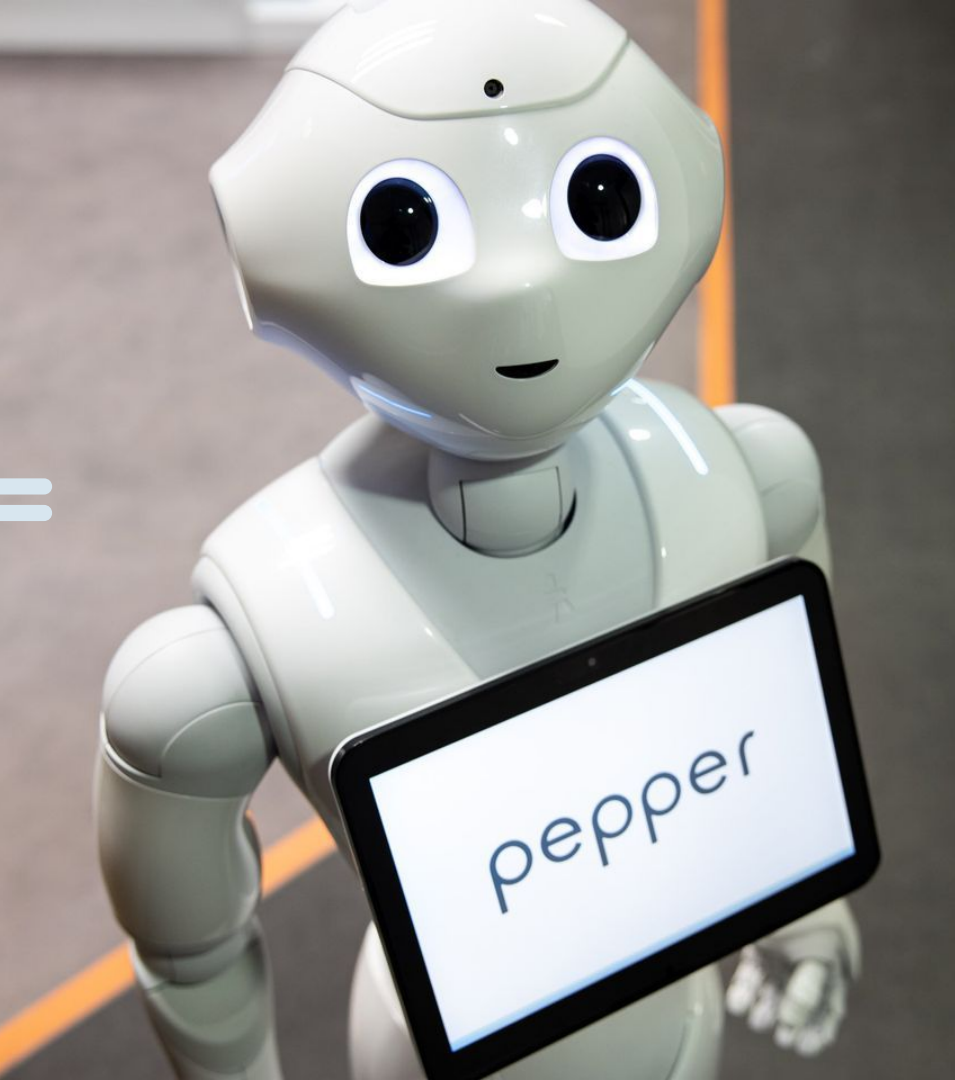




## Research Direction

- **Pepper** - Humanoid Robot





# Research Direction

- Pepper - Humanoid Robot



**focus on  
a novel EUD  
approach**



# Research Direction

focus on  
a novel EUD  
approach

✓ aim to **democratize programming**

! non-technical users **may not care about programming** itself

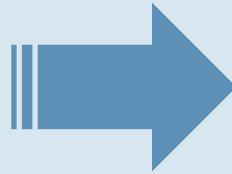


# Research Direction

focus on  
a novel EUD  
approach

✓ aim to **democratize  
programming**

! non-technical users **may not  
care about programming** itself

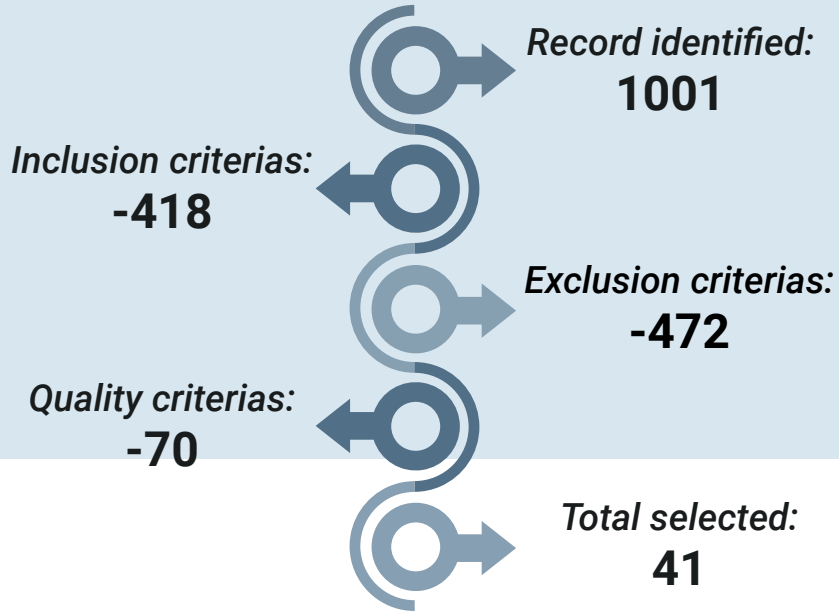


## Possible Solution

accessible  
**programming mode**  
+  
automatic problem  
solving through  
machine **knowledge**



## ☹️ = LITERATURE REVIEW



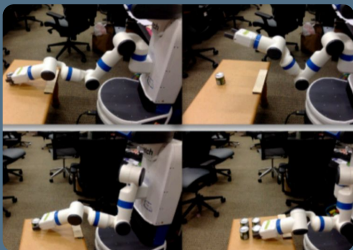
## Research Direction

### Possible Solution ☹️

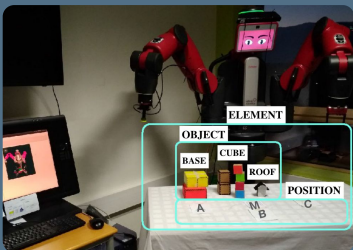
accessible  
programming mode  
+  
automatic problem  
solving through  
machined knowledge



## ⋮ = SIMILAR APPROACHES



Knowledge based  
on 5000 images  
database



Learning from tasks  
programmed by  
end-users

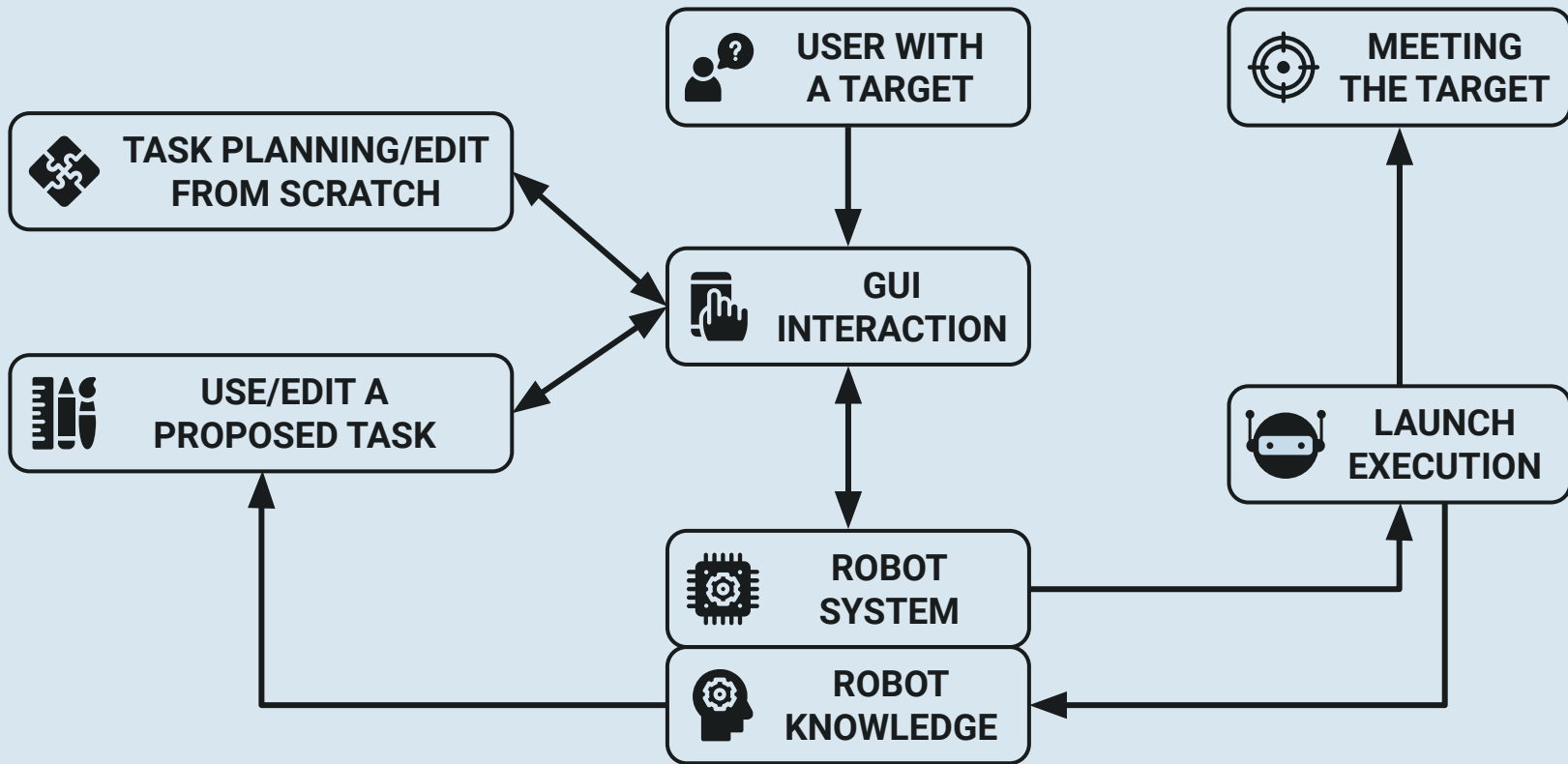
## ⋮ Research Direction

### ⋮ Possible Solution

accessible  
programming mode  
+  
automatic problem  
solving through  
machined knowledge



# Solution Overview



# Open Problems

## Usability

How to avoid issues related to adaptability and error management?

## Evaluation

Lack of standards for real and long-term evaluation.

## Learning

Define the correct method that fit an EUD approach

## Techniques

Which programming styles/metaphors are the best to use?



# Open Problem #1

## Usability

There is low consideration of the **system's usability**, especially about:

- feedback
- error preventing
- handling of errors
- adaptability

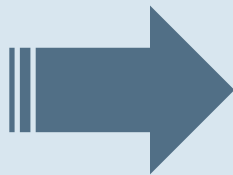


# Open Problem #1

## Usability

There is low consideration of the system's usability, especially about:

- feedback
- error preventing
- handling of errors
- adaptability



A robot equipped with its own **knowledge** could:

- suggest possible errors
- correct errors
- provide a simulation
- be more flexible to user skill
- be adapting to user habits

# Open Problem #2

## Evaluation

EUD approaches are usually tested in controlled context and not in **realistic** situations.

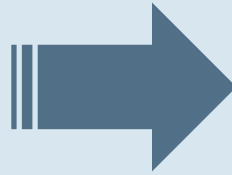
There is never a testing phase that extends in a **long-term** over the end of the design cycle.

# Open Problem #2

## Evaluation

EUD approaches are usually tested in controlled context and not in realistic situations.

There is never a testing phase that extends in a long-term over the end of the design cycle.



In addition to new standard for realistic and long-term scenarios, a robot equipped with its own **knowledge** could:

- simulate realistic scenarios
- predict long-term changing

# Open Problem #3

## Learning

- few similar studies
- no social related
- evaluation effort

Which is the best learning method to implement in an EUD approach for a robot?



# Open Problem #4

## Techniques

- multi-styles?
- need user study

Which programming styles/metaphors are the best to use?





## ☰ QUESTIONS TIME

- What is EUD?
- EUD Concepts
- Why EUD for robot?
- EUD Design Space
- Research Direction
- Open Problems
  - Usability
  - Evaluation
  - Learning
  - Techniques

Thanks for your attention!

