Dezvoltare si Integrare

Andrei Bragarenco

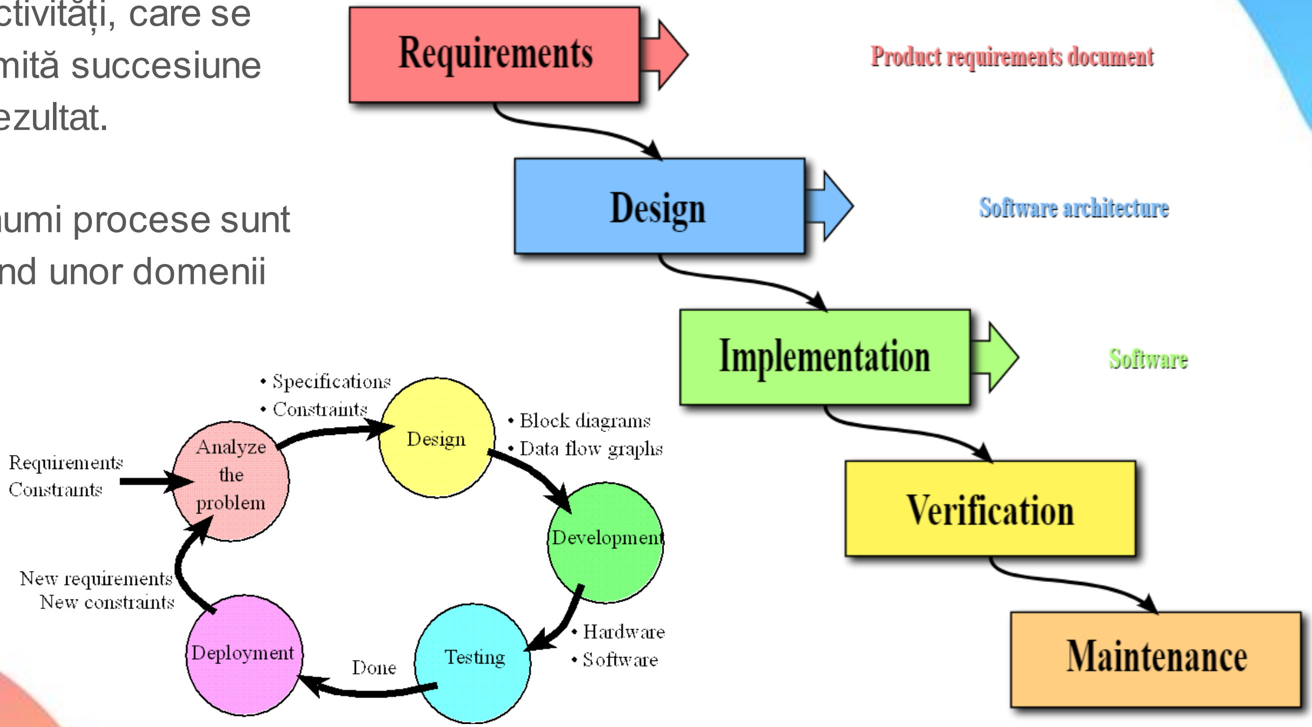
Clubul Ingineresc Micro Lab

IoT 2020 Bootcamp

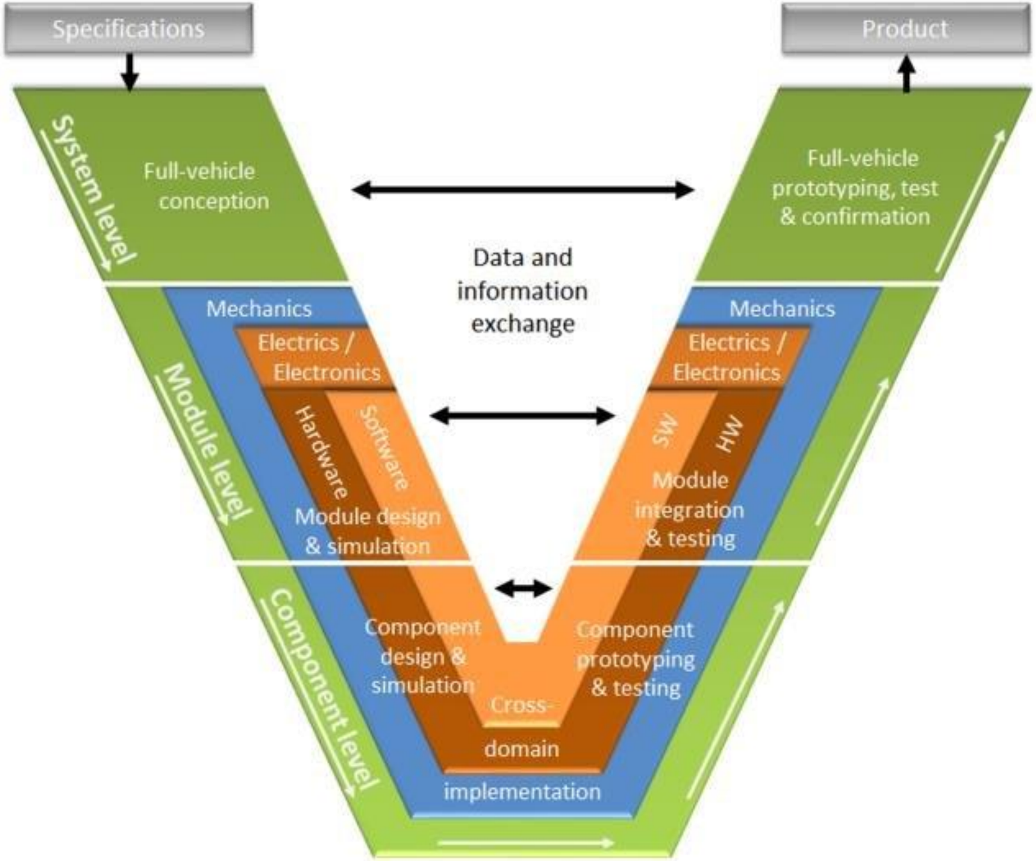
Proces de dezvoltare

- Proces - un grup de activități, care se desfășoară într-o anumită succesiune pentru a produce un rezultat.

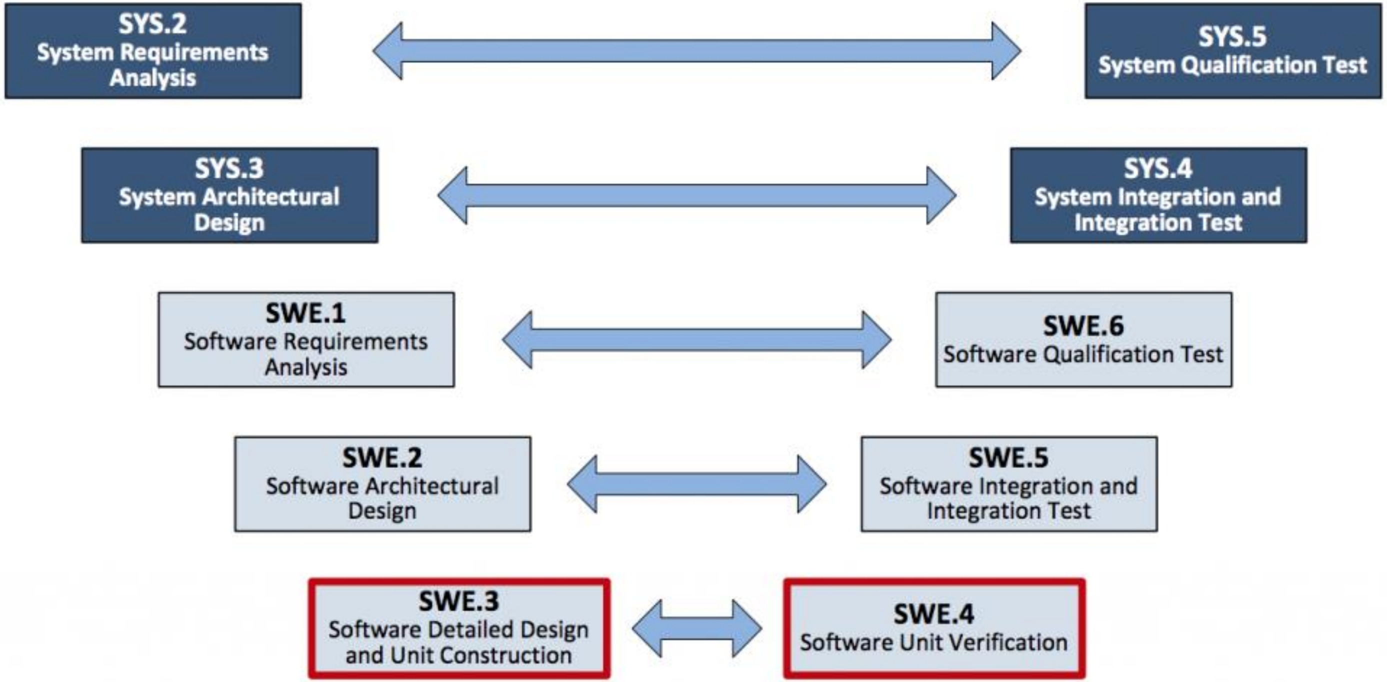
- Acțiunile care se pot numi procese sunt foarte variate, apațiând unor domenii foarte diferite.



Process V-cycle



Process SW Development



SYS.2
System Requirements Analysis

SYS.5
System Qualification Test

SYS.3
System Architectural Design

SYS.4
System Integration and Integration Test

SWE.1
Software Requirements Analysis

SWE.6
Software Qualification Test

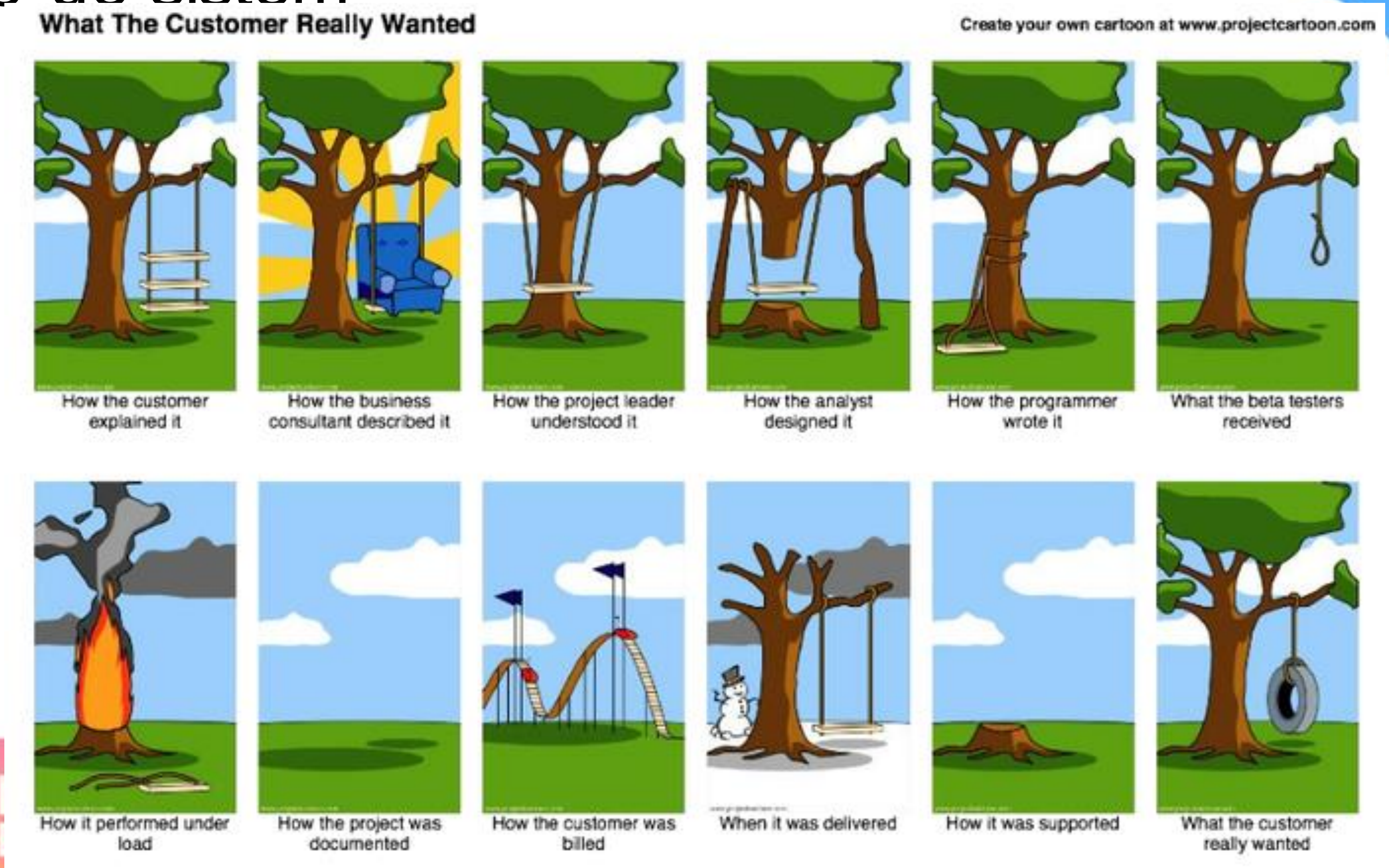
SWE.2
Software Architectural Design

SWE.5
Software Integration and Integration Test

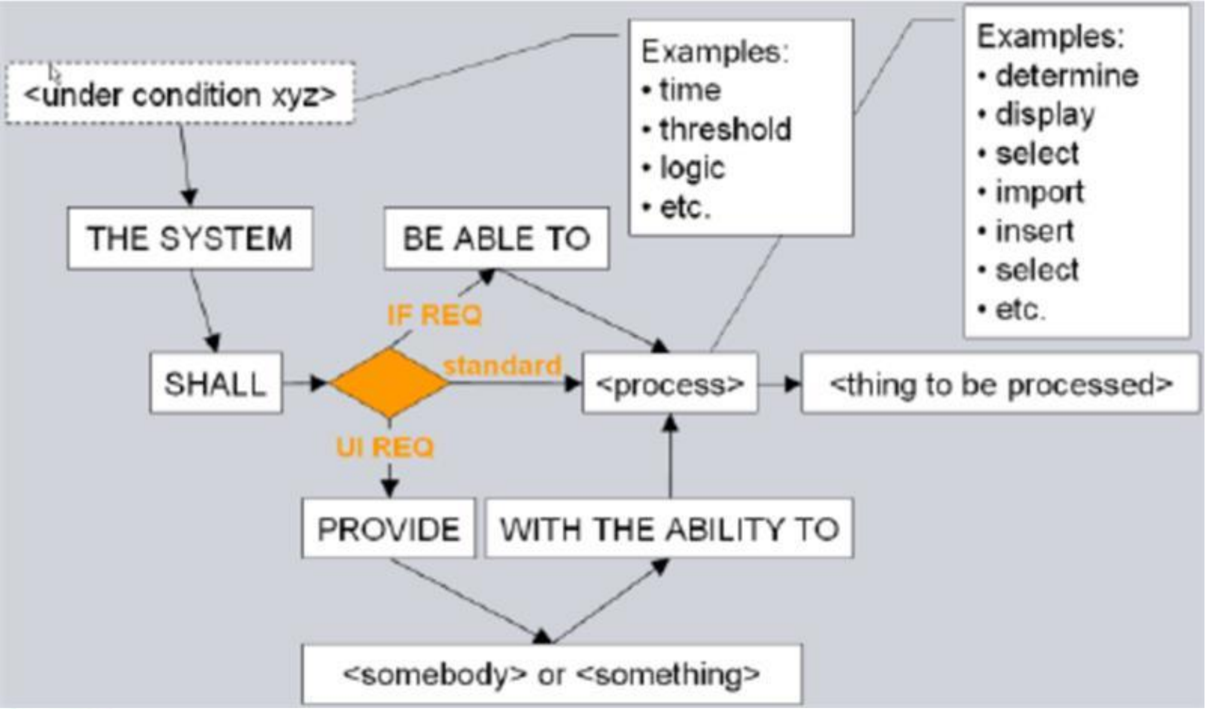
SWE.3
Software Detailed Design and Unit Construction

SWE.4
Software Unit Verification

Cerinte de sistem


What The Customer Really Wanted
Create your own cartoon at www.projectcartoon.com
How the customer explained it
How the business consultant described it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
What the beta testers received
How it performed under load
How the project was documented
How the customer was billed
When it was delivered
How it was supported
What the customer really wanted

Reguli definire cerinte

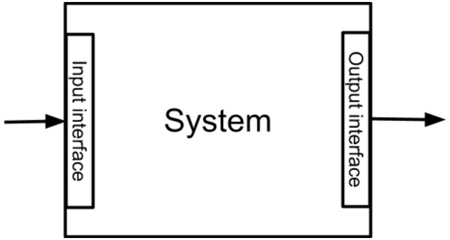


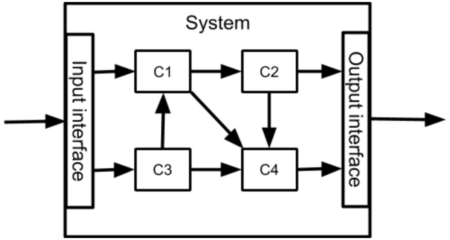
Exemple cerinte sistem

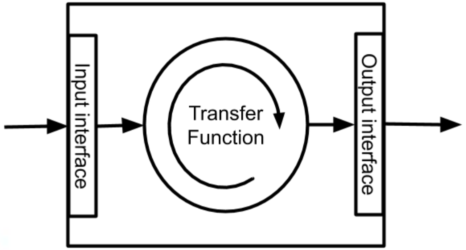
- Creinte Functionale
  - Sistemul **trebuie** sa colecteze date despre temperatura mediului.
  - Sistemul **trebuie** sa transmita datele catre Cloud.
  - Sistemul **trebuie** sa fie capabil sa activeze/dezactiveze incalzitorul.
  - Sistemul **trebuie** sa realizeze functia de control a temperaturii in mediu.
  - Sistemul **trebuie** sa fie capabil sa primeasca comenzi de control si configurare de la Cloud

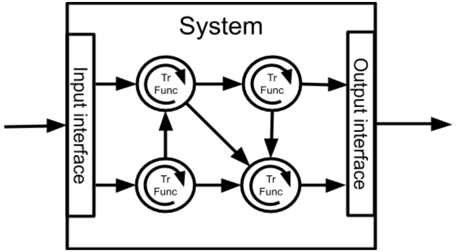
- Cerinte NE functionale
  - Sistemul trebuie sa fie distribuit in cadrul unei case de locuit

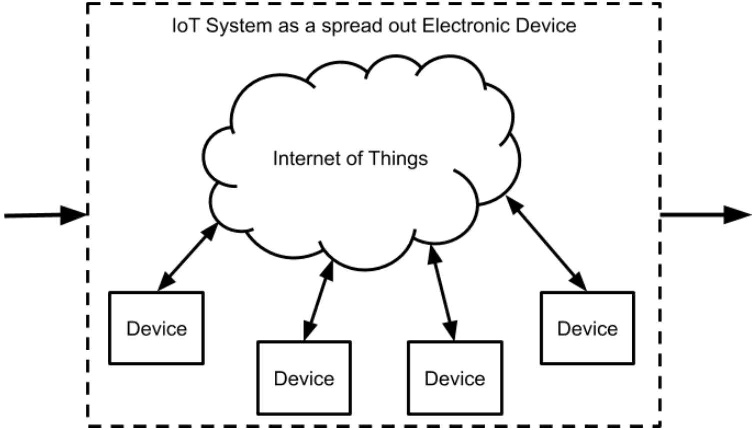
Arhitectura sistem



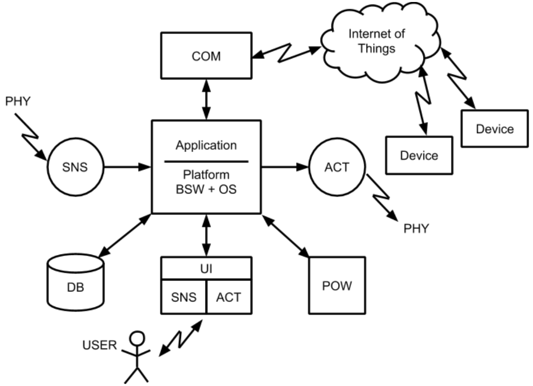


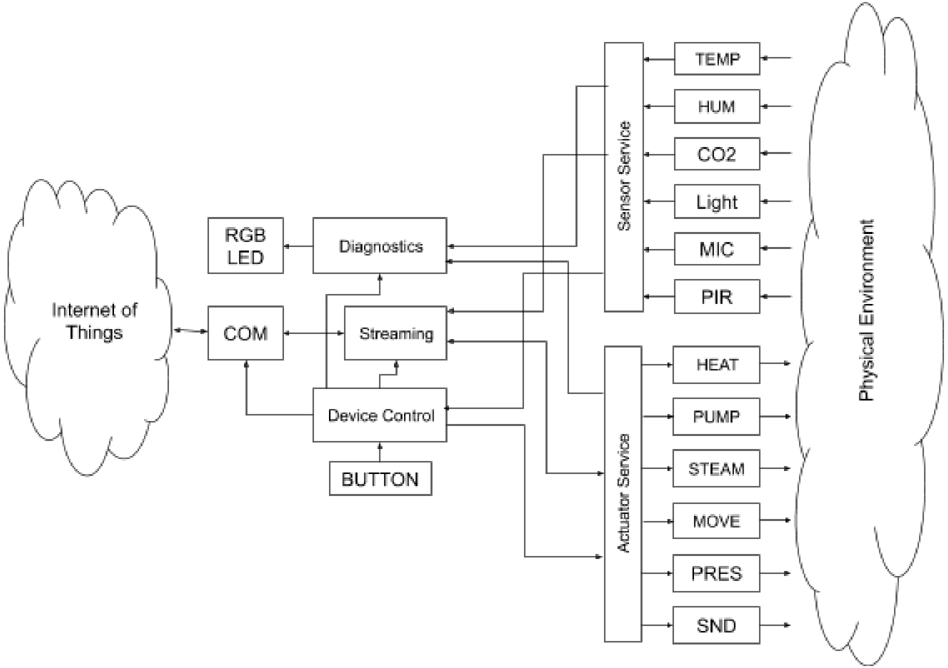






Arhitectura "Thing"





Clubul Ingineresc Micro Lab

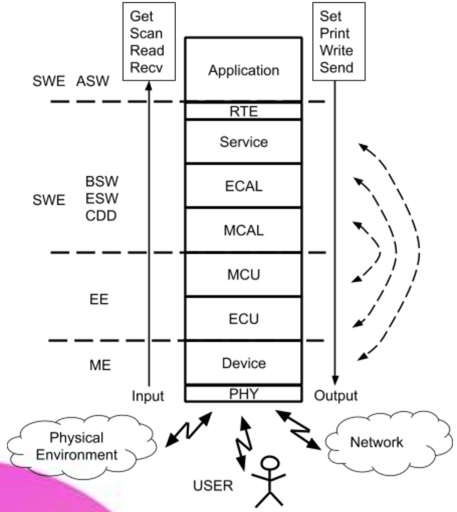
IoT 2020 Bootcamp

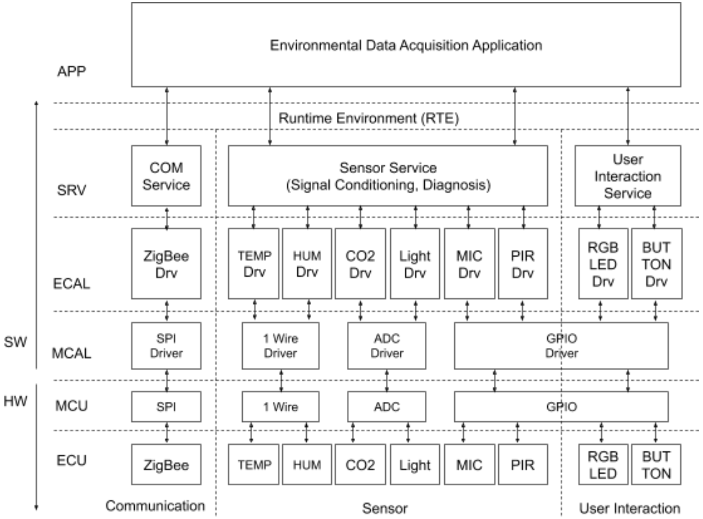
Exemple cerinte Software

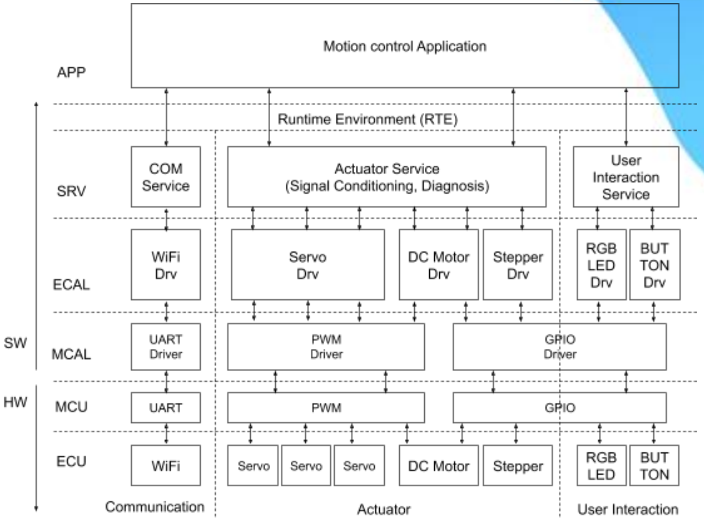
- Creinte Functionale
  - SW Sensor trebuie sa ofere o interfata pentru citirea Temperaturii
  - SW actuator trebuie sa ofere o interfata pentru activare/dezactivare incalzitor.
  - SW de sa realizeze controlul temperaturii dupa functia ON/OF cu histerezis
- Cerinte NE functionale
  - SW trebuie realizat dupa principiul Modular

Clubul Ingineresc Micro Lab IoT 2020 Bootcamp

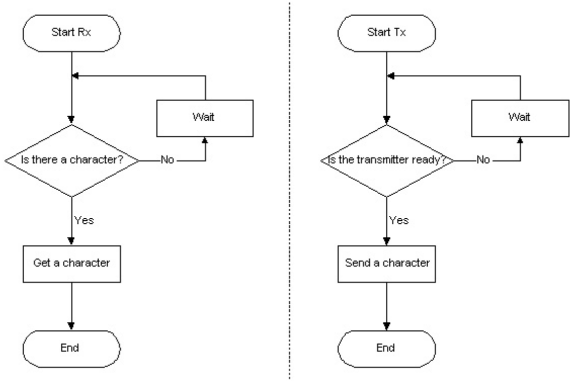
Arhitectura SW

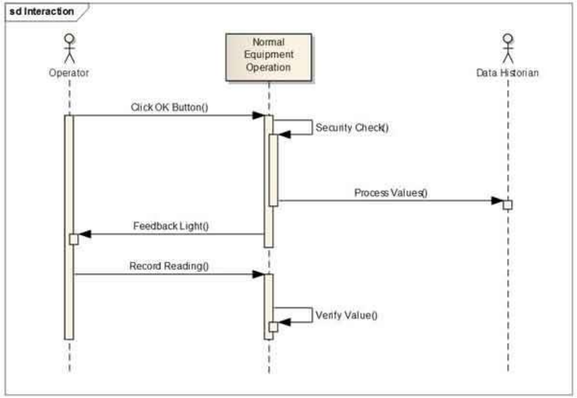


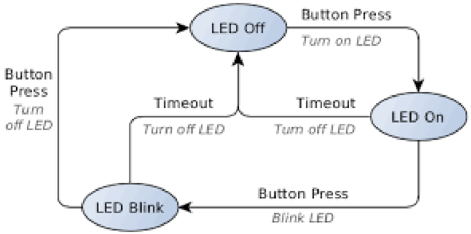


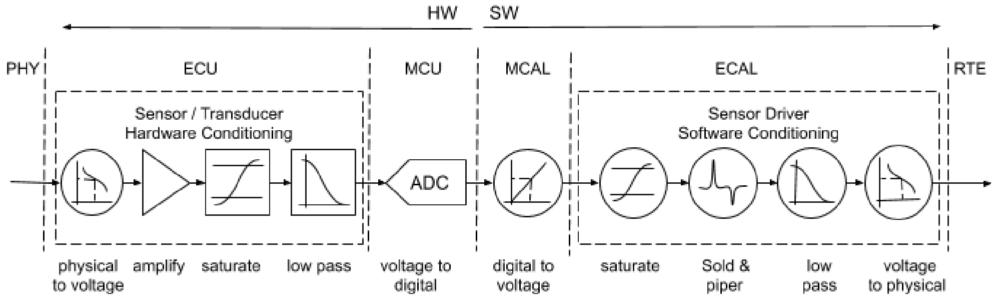


SW Detailed design

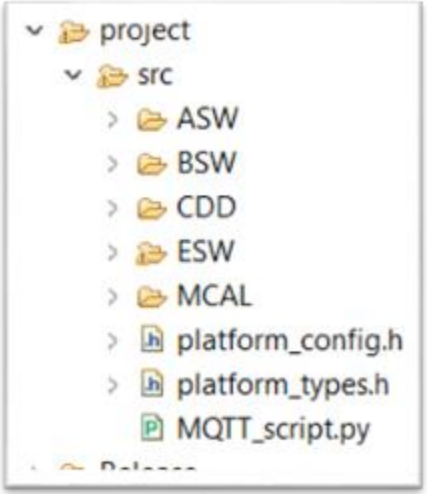




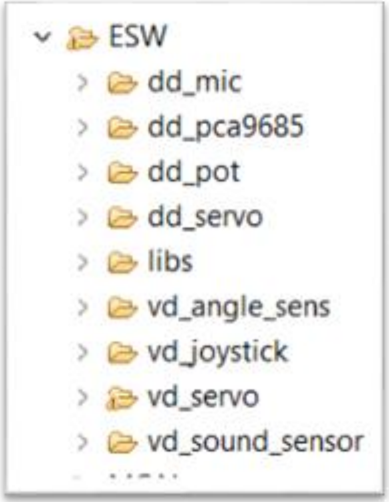




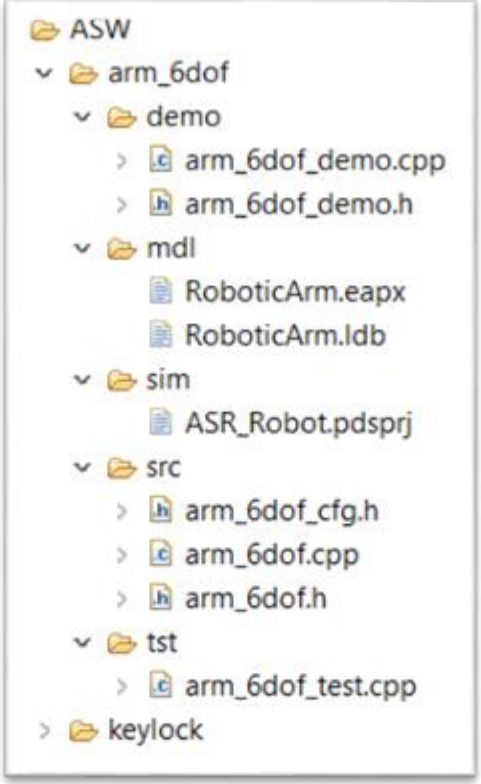
Implementare - Structura proiect



```
⌄ project
  ⌄ src
    › ASW
    › BSW
    › CDD
    › ESW
    › MCAL
    › platform_config.h
    › platform_types.h
      MQTT_script.py
  › Release
```

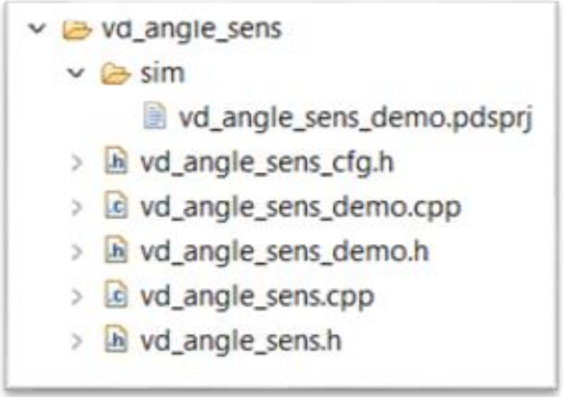


```
⌄ ESW
  › dd_mic
  › dd_pca9685
  › dd_pot
  › dd_servo
  › libs
  › vd_angle_sens
  › vd_joystick
  › vd_servo
  › vd_sound_sensor
```



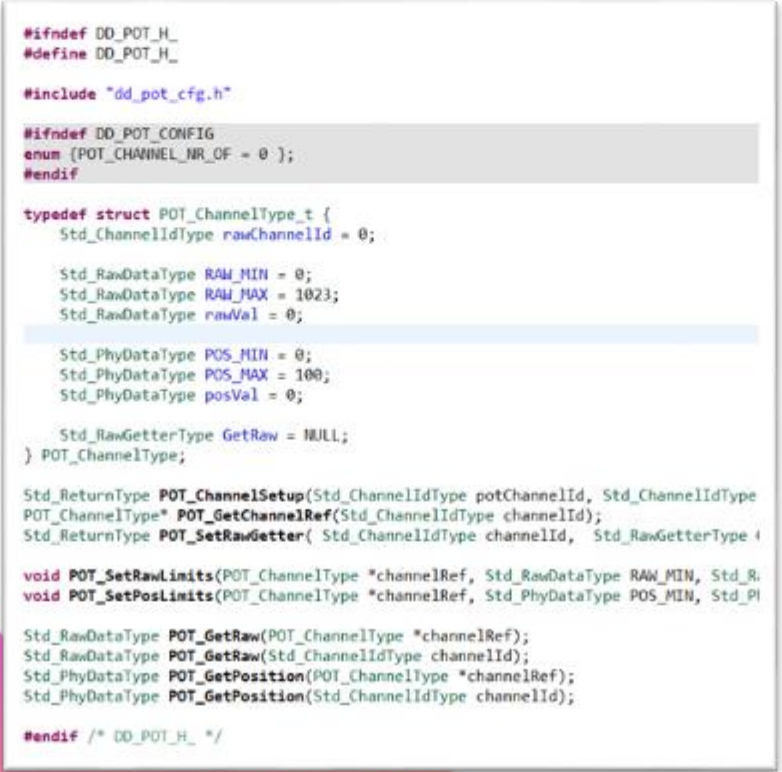
```
ASW
⌄ arm_6dof
  ⌄ demo
    › arm_6dof_demo.cpp
    › arm_6dof_demo.h
  ⌄ mdl
      RoboticArm.eapx
      RoboticArm.ldb
  ⌄ sim
      ASR_Robot.pdsprj
  ⌄ src
    › arm_6dof_cfg.h
    › arm_6dof.cpp
    › arm_6dof.h
  ⌄ tst
    › arm_6dof_test.cpp
› keylock
```

Component & Unit

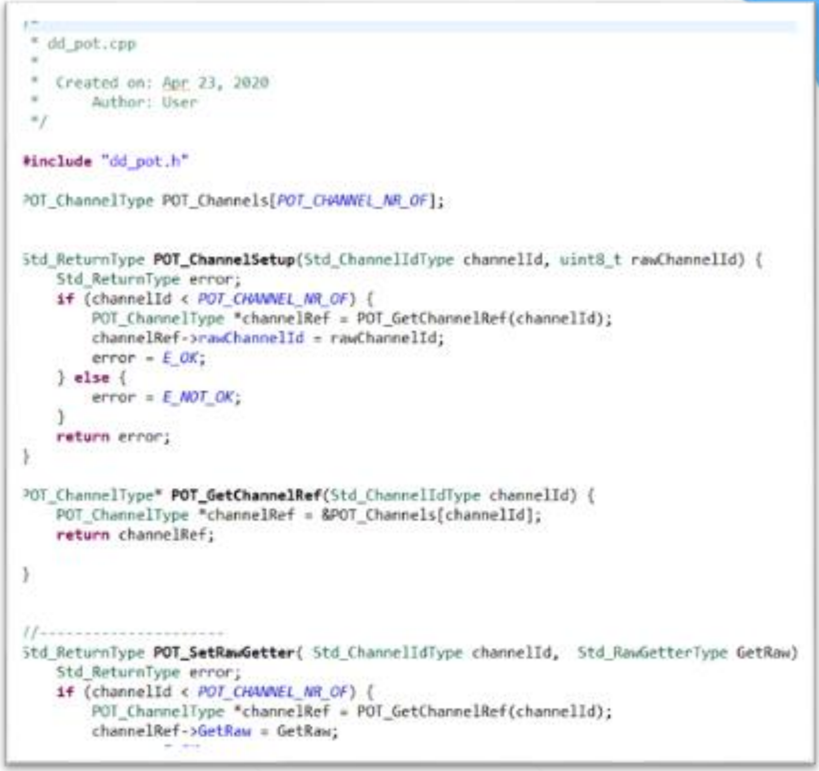


```
⌄ vd_angle_sens
  ⌄ sim
      vd_angle_sens_demo.pdsprj
  › vd_angle_sens_cfg.h
  › vd_angle_sens_demo.cpp
  › vd_angle_sens_demo.h
  › vd_angle_sens.cpp
  › vd_angle_sens.h
```

Implementare - strucutura unit



```
#ifndef DD_POT_H_
#define DD_POT_H_

#include "dd_pot_cfg.h"

#ifndef DD_POT_CONFIG
enum {POT_CHANNEL_NR_OF = 0 };
#endif

typedef struct POT_ChannelType_t {
    Std_ChannelIdType rawChannelId = 0;

    Std_RawDataType RAW_MIN = 0;
    Std_RawDataType RAW_MAX = 1023;
    Std_RawDataType rawVal = 0;

    Std_PhyDataType POS_MIN = 0;
    Std_PhyDataType POS_MAX = 100;
    Std_PhyDataType posVal = 0;

    Std_RawGetterType GetRaw = NULL;
} POT_ChannelType;

Std_ReturnType POT_ChannelSetup(Std_ChannelIdType potChannelId, Std_ChannelIdType
POT_ChannelType* POT_GetChannelRef(Std_ChannelIdType channelId);
Std_ReturnType POT_SetRawGetter( Std_ChannelIdType channelId,  Std_RawGetterType

void POT_SetRawLimits(POT_ChannelType *channelRef, Std_RawDataType RAW_MIN, Std_R
void POT_SetPosLimits(POT_ChannelType *channelRef, Std_PhyDataType POS_MIN, Std_P

Std_RawDataType POT_GetRaw(POT_ChannelType *channelRef);
Std_RawDataType POT_GetRaw(Std_ChannelIdType channelId);
Std_PhyDataType POT_GetPosition(POT_ChannelType *channelRef);
Std_PhyDataType POT_GetPosition(Std_ChannelIdType channelId);

#endif /* DD_POT_H_ */
```



```
/*
 * dd_pot.cpp
 *
 *  Created on: Apr 23, 2020
 *      Author: User
 */

#include "dd_pot.h"

POT_ChannelType POT_Channels[POT_CHANNEL_NR_OF];


Std_ReturnType POT_ChannelSetup(Std_ChannelIdType channelId, uint8_t rawChannelId) {
    Std_ReturnType error;
    if (channelId < POT_CHANNEL_NR_OF) {
        POT_ChannelType *channelRef = POT_GetChannelRef(channelId);
        channelRef->rawChannelId = rawChannelId;
        error = E_OK;
    } else {
        error = E_NOT_OK;
    }
    return error;
}

POT_ChannelType* POT_GetChannelRef(Std_ChannelIdType channelId) {
    POT_ChannelType *channelRef = &POT_Channels[channelId];
    return channelRef;

}


//---------------------
Std_ReturnType POT_SetRawGetter( Std_ChannelIdType channelId,  Std_RawGetterType GetRaw)
    Std_ReturnType error;
    if (channelId < POT_CHANNEL_NR_OF) {
        POT_ChannelType *channelRef = POT_GetChannelRef(channelId);
        channelRef->GetRaw = GetRaw;
```

Integrare

Aplicatie practica …..

Mulțumesc pentru atentie

Întrebari?