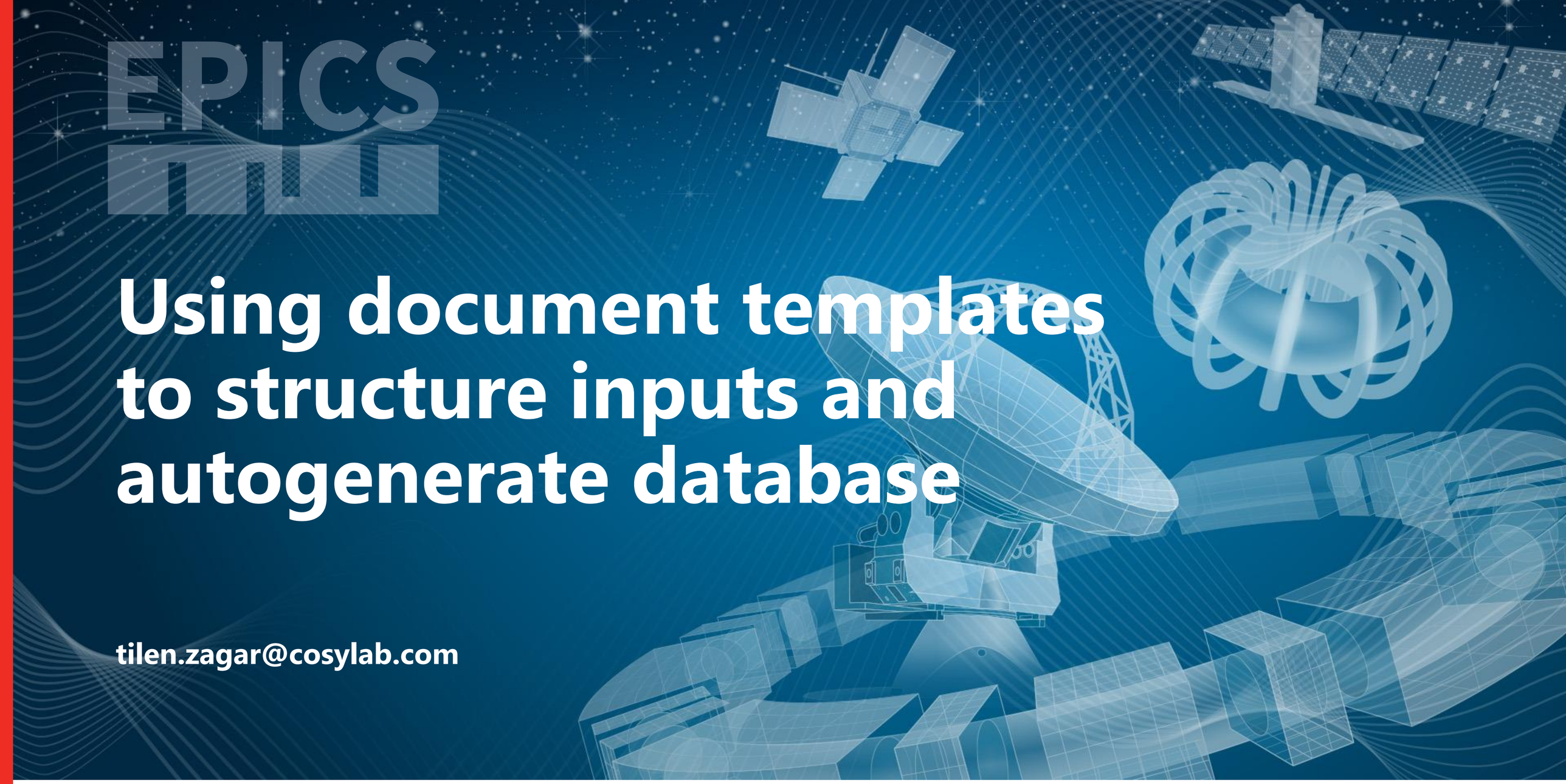


Using document templates to structure inputs and autogenerate database

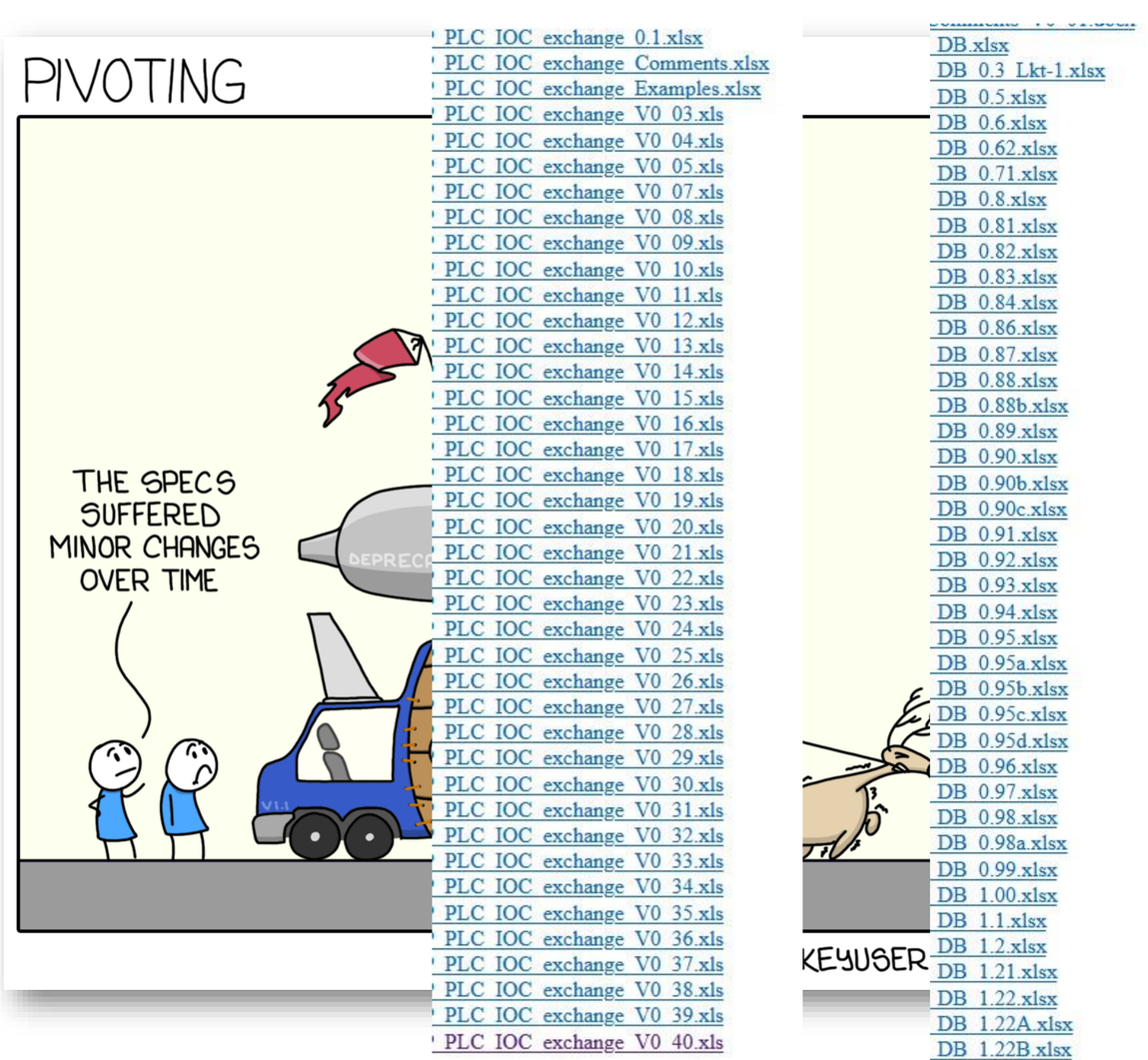
tilen.zagar@cosylab.com

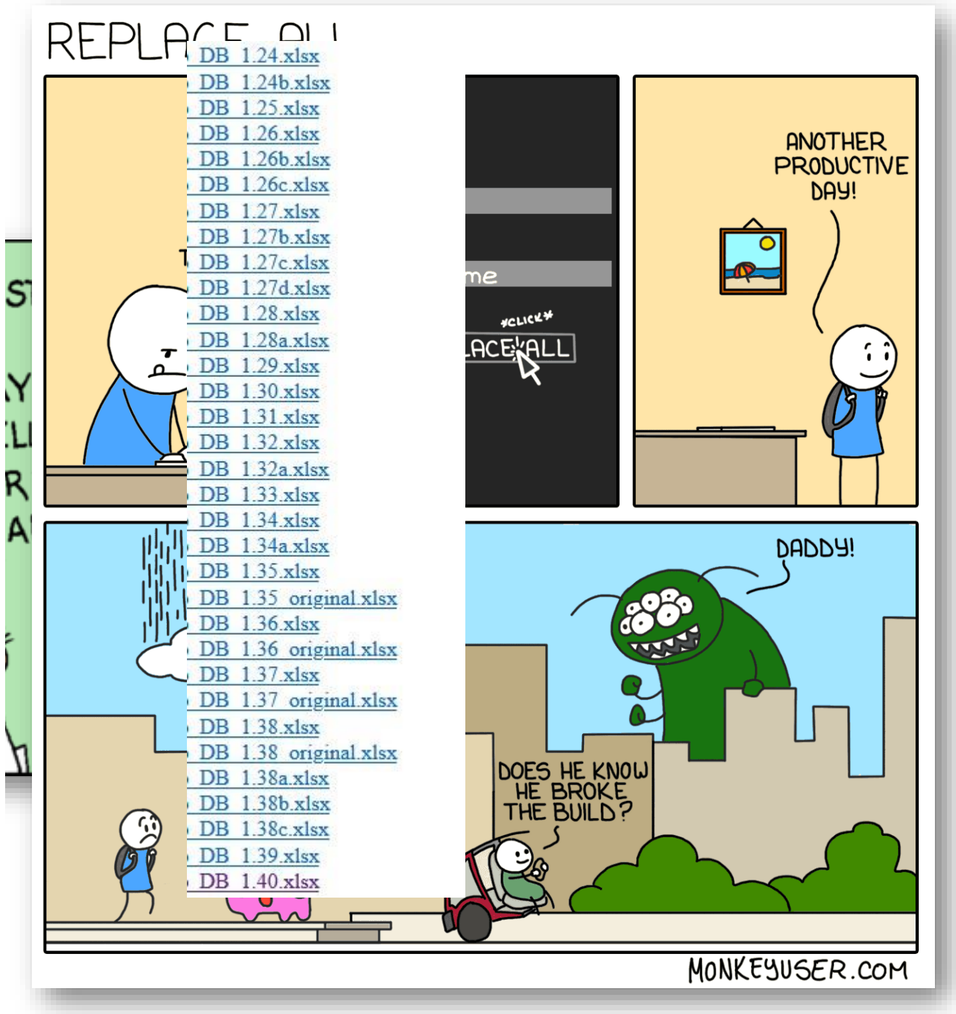
EPICS Collaboration Meeting 2022

EPICS

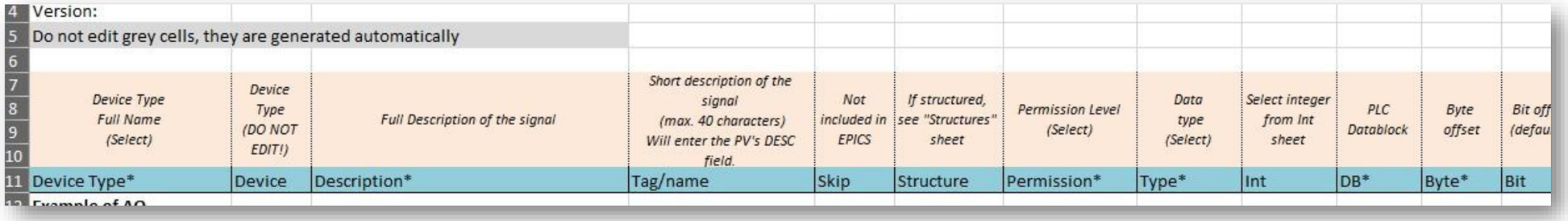
COSYLAB

Challenges of developing (control system) software?





Our approach: structure, guidance and automation

1. Identify stakeholders / domain experts and agree about roles / expectations
2. Milestones with dates for each step – helps to keep the goal in mind
3. Help beyond just the scope of “coding”, but …
4. … know your audience!
5. 

4	Version:
5	Do not edit grey cells, they are generated automatically
6	
7–10	*Device Type Full Name (Select)* / *Device Type (DO NOT EDIT!)* / *Full Description of the signal* / *Short description of the signal (max. 40 characters) Will enter the PV's DESC field.* / *Not included in EPICS* / *If structured, see "Structures" sheet* / *Permission Level (Select)* / *Data type (Select)* / *Select integer from Int sheet* / *PLC Datablock* / *Byte offset* / *Bit off (defau*
11	Device Type* / Device / Description* / Tag/name / Skip / Structure / Permission* / Type* / Int / DB* / Byte* / Bit

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Type	No.	Relay Name	Assignment	EPICS Record Name	I/O comment	ON	OFF	ON Severity	OFF Severity	HIGH	Default	SCAN	Archive	Alarm
2	Shared Relay	1	E1	CPU1	KLY:SW:TEMP:ILK	Klystron thermo switch	OK	Interlock	NO_ALARM	MAJOR			1 second	Monitor	NO_ALARM
3	Shared Relay	2	E2	CPU1	KLY:IP1:PRS:ILK	Klystron ion pump 1 interlock	OK	Interlock	NO_ALARM	MAJOR			1 second	Monitor	NO_ALARM
4	Shared Relay	3	E3	CPU1	KLY:IP2:PRS:ILK	Klystron ion pump 2 interlock	OK	Interlock	NO_ALARM	MAJOR			1 second	Monitor	NO_ALARM

6. Automate “everything”!

Tools are already out there

- No need to reinvent the wheel
  - We tried Visual Basic …
  - … and we stick to Python ☺

shared_memory_map_template.xlsm - Module1 (Code)

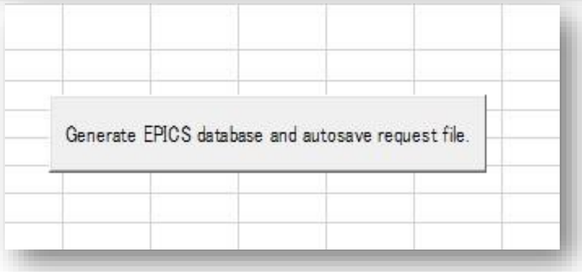
(General) | MakeBase

```
Sub MakeBase()
    ' Create text files
    Dim dbName As String
    Dim reqFile As String

    ' Obtain version number of this signal list
    Worksheets("History").Activate
    colVersion = Rows(1).Find("Version", LookIn:=xlValues, LookAt:=xlWhole).Column
    latestRow = Columns(colVersion).End(xlDown).Row
    latestVersion = Cells(latestRow, colVersion).Value

    Worksheets("Basic").Activate
    dbName = ActiveWorkbook.Path & "\" & Cells(5, 2) & ".db"
```

```
617
618         try:
619
620                 # Error check: duplication of byte offsets
621                 if line[plcdb_i] in DBs:
622                         if ((line[byte_i] in DBs[line[plcdb_i]][1]) and not line[type_i] == 'BIT'):
623                                 raise ValueError('Byte offset ' + line[byte_i] + ' in ' + line[plcdb_i] + ' already exists!')
624                 else:
625                         DBs[line[plcdb_i]] = ['', []]
626
627                 # Assign record type
628                 if (args.mode == 'r'):
629                         if (line[type_i] == 'BIT'):
```




```
11  os.system('python parse_signals.py -i %s -m s' % sp_pth)
12  os.system('python parse_signals.py -i %s -m r' % rb_pth)
13
14  os.system('python generate_RB_to_SP_database.py')
15
```

Project in numbers

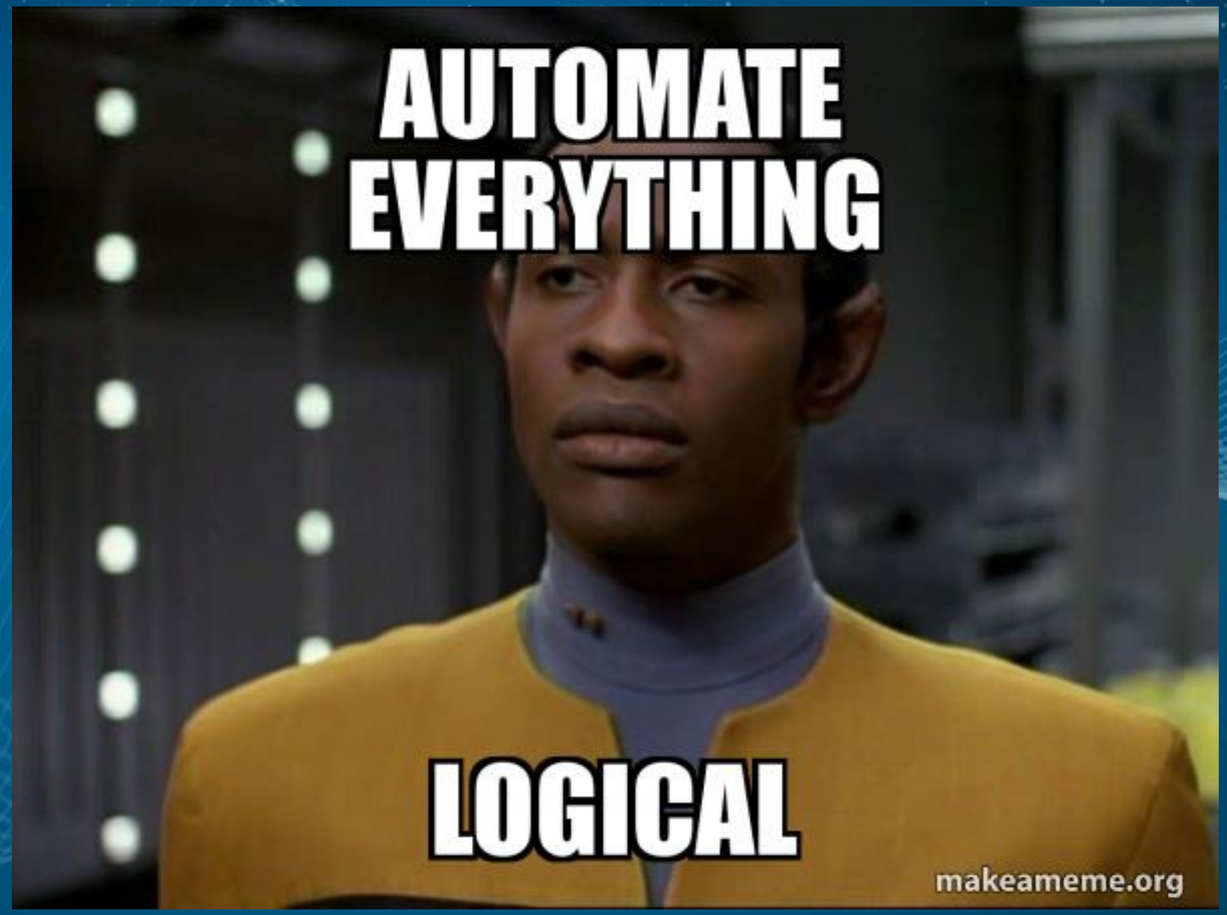
- Almost 50 versions after the “final” version of signals :D
- 9000 variables / records

- Scripts to generate and error check **database, archiver and alarm server configuration took 10% of effort**
- In **half year** we managed to define interfaces, implement all the business logic required on the EPICS level, prepare IOC app, create GUIs, configure central services …

Why are we doing it?

- Single interface / document between field experts and control system engineers
- Clear roles and expectations
- One input, multiple outputs in a single, automated step – **saves a lot of time**
- Added features: error checking, statistic …
- “To err is human”, so if no human is involved = “**no” errors in the database**

Rules, structure and possible



Thank you

EPICS

COSYLAB