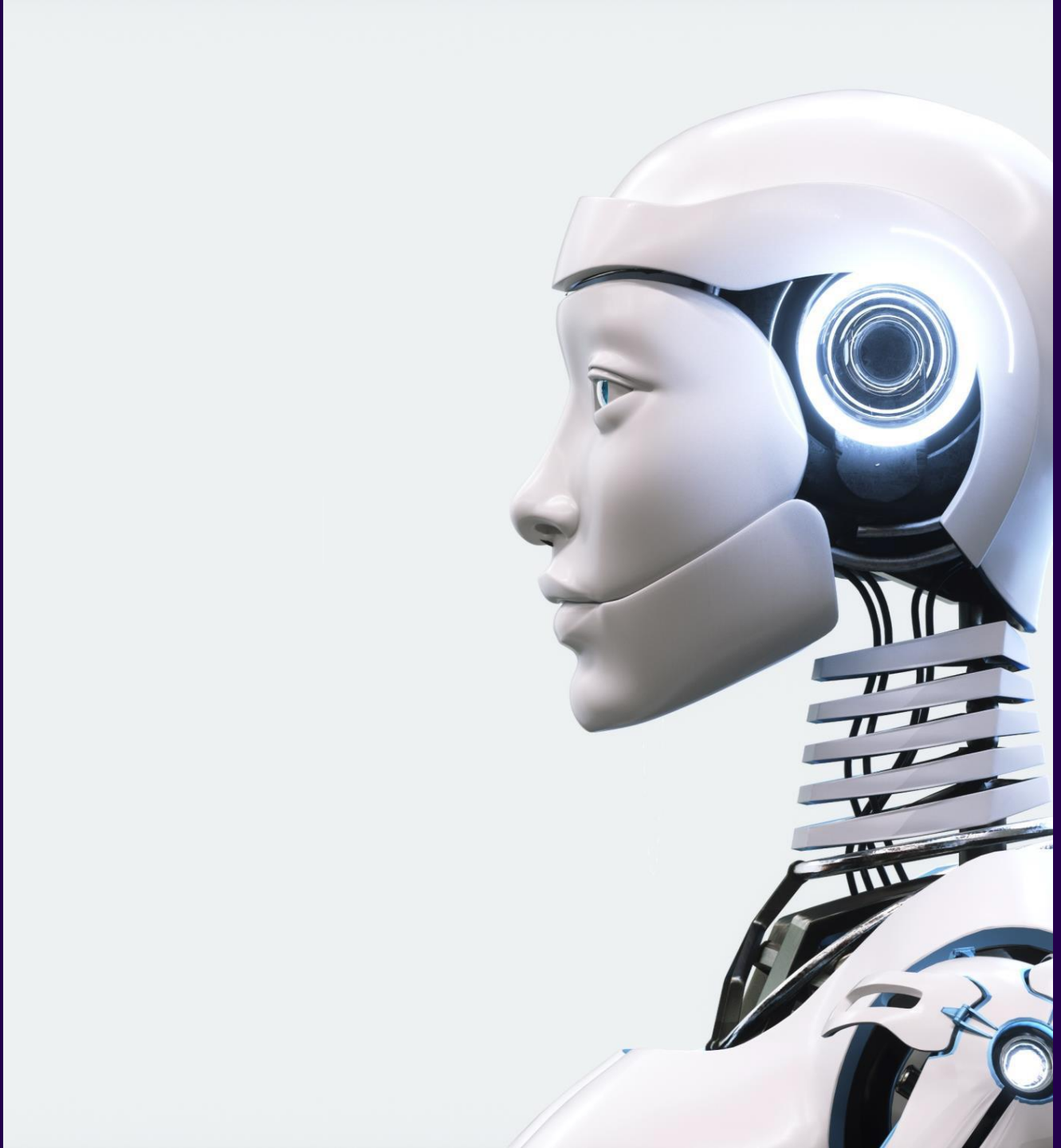


sanofi

Exploration of Code Generation using Generative AI

Sufang Huang

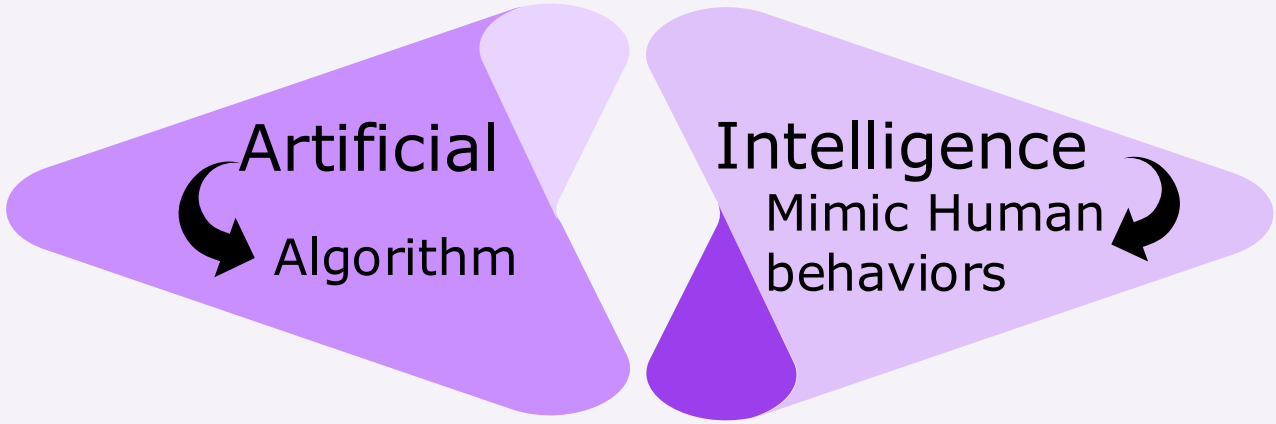
Group Head, Statistical Programming and Analytical Reporting
Sanofi

Nov 2024

What is Artificial Intelligence (AI) ?

A tentative* definition of Artificial Intelligence :

A machine's ability to perform cognitive functions, such as learning, reasoning/predicting and making decisions (output). It is a branch of computer science, mathematics/statistics, and engineering that uses algorithms or models on training and/or operational data (input) to perform tasks and exhibit behaviors, learn and improve over time.



sanofi

* No current consensus on a given definition as of today

Artificial Intelligence (AI)

Narrow AI

- Rule-based system
- Symbolic reasoning
- Knowledge representation & reasoning
- Planning

Machine Learning (ML)

Decision Tree
e.g. Random forest

Bayesian Algo
e.g. Bayesian Network

Instance based algo
e.g. SVM, kNN

Artificial Neural Network (ANN)

Deep Learning (DL)

- Image recognition
- Speech recognition
- Recommendation systems
- Natural Language Processing (NLP)
- Medical diagnosis
- Game Playing

Generative AI
(Large Language Models or Generative Adversarial Networks)

- Text Generation
- Text Summarization
- Question answering
- Code generation
- Image generation, translation, edition

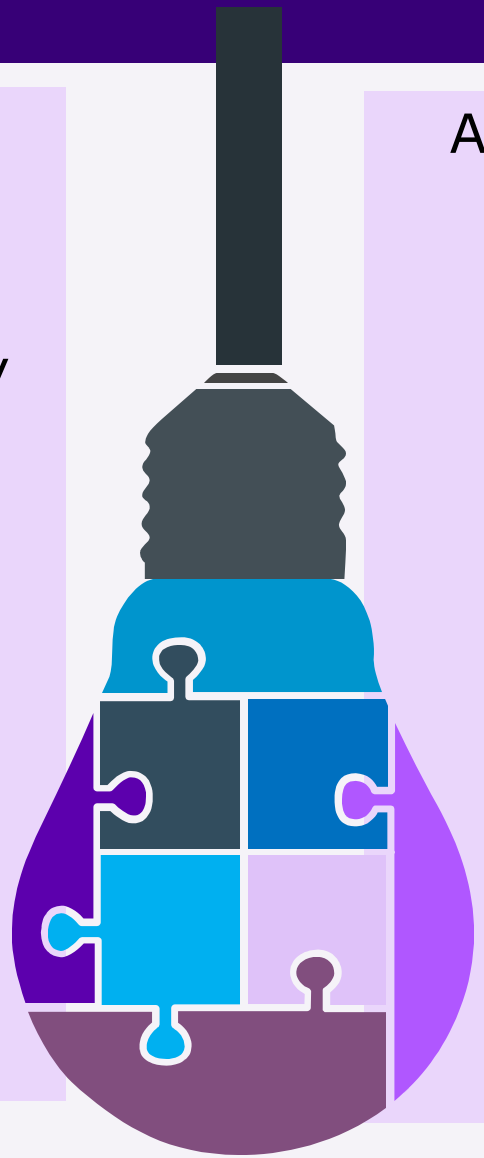
Supervised
- Image classification
- Natural Language Processing
- Speech Recognition

Unsupervised
- Clustering
- Dimension reduction
- Anomaly detection

Reinforced
- Self-driving car
- Robotic

When to consider AI ?

- Automate repetitive, tedious or error-prone tasks (Incl data pre-processing)
- Improve efficiency and productivity (notably by making better & faster decisions, predictions, …)
- Gain insights from data (generate new ideas, identify risks…)
- Solve complex problems (simulations…)
- Automatically improve solutions over time (through learning)



And specifically on Generative AI :

- Text content generation with specific length, tense and tone
- Summarization from structure/unstructure datasource, simplification
- Classification/extraction of content for specific use : by feeling, topic, …
- Code generation: code generation, comments, checks,…

sanofi

Why?| Competition is here and now

Pfizer uses AI-enabled software tools to facilitate knowledge sharing throughout the company, and to accelerate documentation writing

GSK uses AI to increased patient safety through better signal detection

J&J opened 969 new artificial intelligence job positions since October 2020

Lilly developed a platform that handles complex data from wearables. It provides a visual computing solution for large-scale data, live tracking, and monitoring of digital endpoints in several indications.

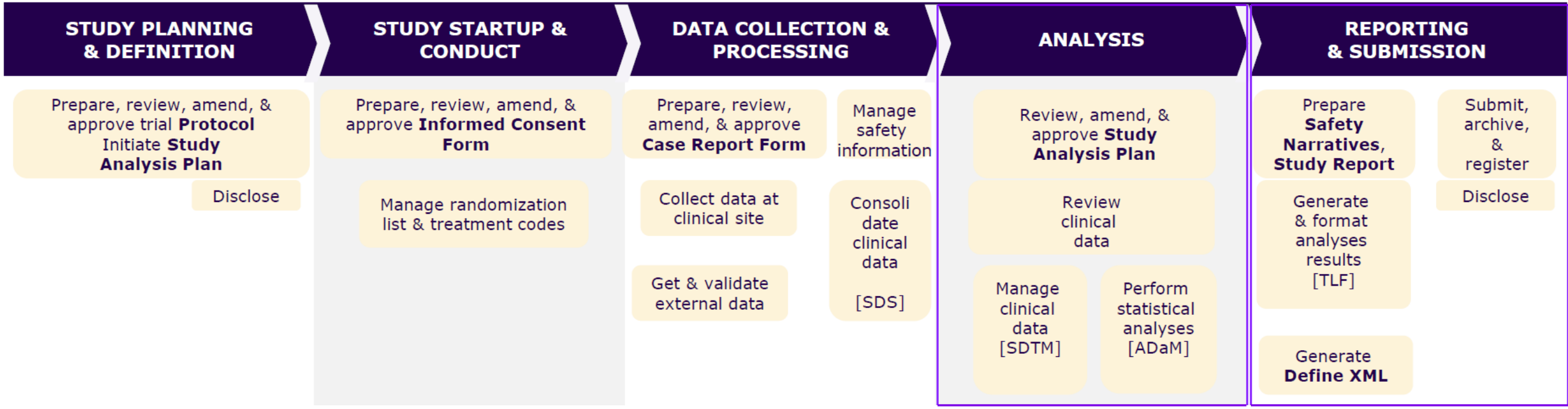
Bayer harnesses NLP*, to revolutionize medical coding, translating physicians' case reports into standardized terms and categories. It enables clinical trial acceleration and medical writers to focus more on interpretation of study results.

Roche uses AI to search expert across the company (provider : StarMind)

*NLP = natural language processing

At Sanofi: Opportunities to accelerate submission via GenAI /LLMs

Manual steps across clinical data treatment, statistical programming, and document authoring can be partially automated with LLMs based solutions.



Protocol Digitalization

End to End Data Flow: RAPID Program

Automated Document Creation

Modenized Data Review Automated Data Integration

Metadata Assisted Programming

Gen AI for CSR

sanofi

Metadata Assisted Programming Services (**MAPS**)

Capabilities

Study Metadata Manager

- Streamline study-level metadata preparation for SDTM and ADaM data creation
- Leverage the use of SDTM & ADaM standards (global level and study level) and versioning

Code Generator

- Automate R script creation (generate a draft program), Rule-based

Report Designer

- Facilitate design and generation of TLF programs (Tables, Listings and Figures)

Project Main Objectives

- Support B&P statistical programming teams to generate high quality draft R programs for non-efficacy related SDTM, ADaM and TLF deliverables
- Automation of programming activities (from metadata management through code generation to TLF design and programming), which can be standardized and treated using pre-defined rule-based logic

MAPS offers a **unique opportunity to transform clinical statistical programming & gain efficiency** by simplifying and automating the existing manual process

sanofi

None Rule-based Code Generator: Generative AI assisted code generation

Perform **study & analysis metadata preparation**

Completed

Study Metadata Manager

Create study SDTM & ADaM variables and their related mapping rules (including specific CTs)

Move from template-based to **semi-automated clinical statistical programming**

Ongoing

Code Generator

Generate SAS/R scripts (drafts) from the metadata that will be used to convert the clinical data in SDTM & ADaM standards
Generate draft R scripts for the TLF creation

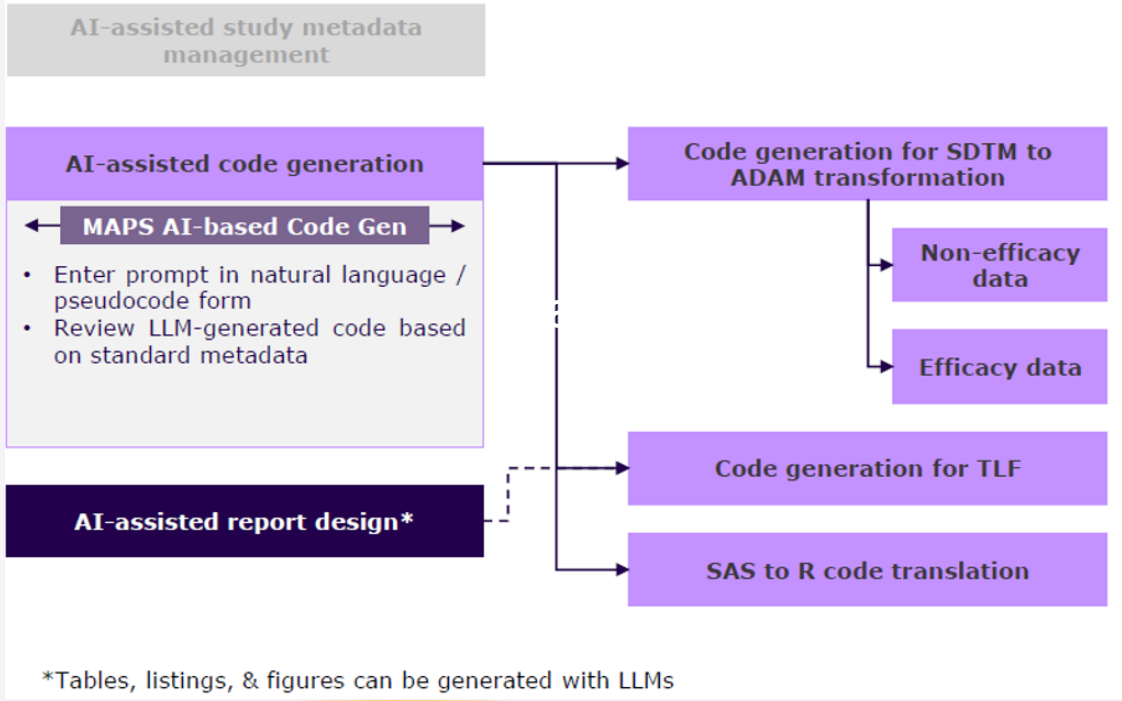
GenAI can be applied for enhancing Code Generator scope

Integrate **TLF design and programming** with metadata

Ongoing

Report Designer

Provide design facilities for the TLF preparation and mapping management



GenAI Code Generator PoC: Goals and Design

AI-Powered application expected to provide SAS code based on a prompt written in natural language

PoC Goals:

- To **explore feasibility** of using LLM to generate SAS code for study data preparation where standard mapping is not available
- To **enhance** MAPS (Metadata Assisted Programming Services) for the **non-rule-based** feature
- To focus on **non-standard efficacy** data (ADaM variable derivation and TFL development), as well as code translation from SAS to R.
- To **transform** clinical statistical programming & gain efficiency by simplifying and automating the existing manual process

PoC Design: Two Stages

Stage I: ✓

- Develop one LLM to build simple SAS code transforming SDTM to ADaM
- Scope: ADTTE PFS & OS of Oncology Multiple Myeloma Studies

Go-No-Go Evaluation

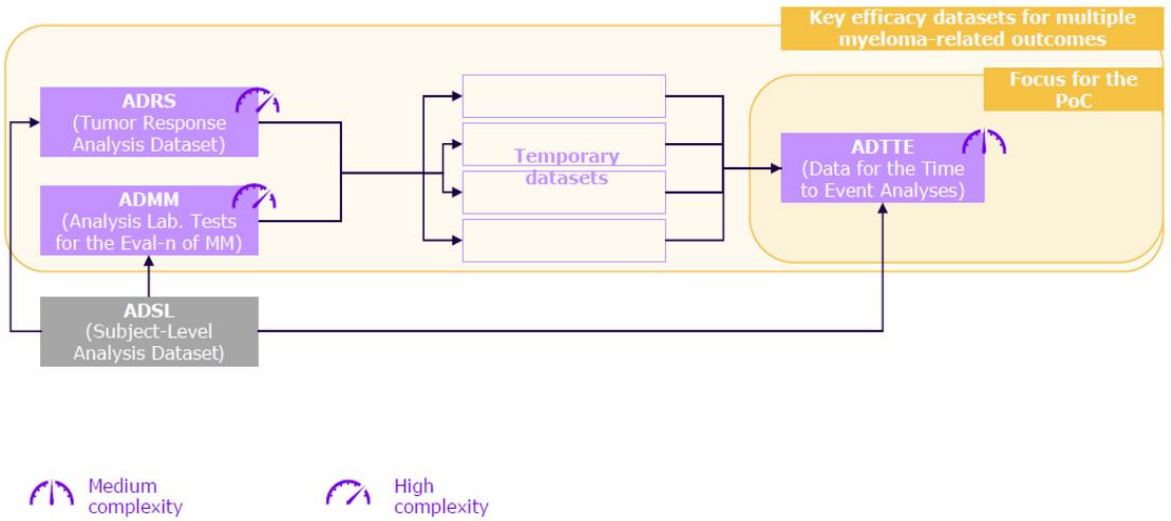
Stage II:

- Establish the model **stability**
- Assess **generalization/repeatability** (new endpoints and studies)
- Refine generated **code quality** methods
- Scope: ADTTE PFS of Solid Tumor studies
  Selection of additional endpoints: BOR
  Other TA

sanofi

Stage I: Oncology Efficacy Datasets are selected for PoC

ADTTE presents to a lower derivation complexity, as compared to ADRS and ADMM datasets



- PoC will be rolled out in 5 sprints of 3 weeks (a total of 15 weeks), with a Go/No Go decision gate after 3 sprints.
- A set of 9 KPIs across code relevance, speed, and green IT is proposed for evaluation.

Objective	Key results	KPI
RELEVANCE GenAI generates usable code	**Code similarity** Generated code is similar to human-written one	Perplexity
		Code acceptance score
	Code quality The code follows "gold standard" of programming	Code statistics
		Code quality score
		User confidence score
SPEED GenAI model replies fast enough to keep end use engaged	**Time to get relevant code** The model provides correct response to prompt fast	Average model inference time
		Average number of iterations
GREEN IT Greenhouse gas emissions associated with GenAI model are limited	**Infrastructure impact** The infrastructure impact is the lowest possible	Average daily CPU load
		Average monthly machine up time

GenAI Code Generator: Prompt driven

Derivation prompt example (STARTDT, OS)

prompt

System Prompt

1. Your task is to answer the user request below by generating functional, efficient and documented SAS code with good syntax.

User Prompt

2. User Input:
Create variable **STARTD** ... ADaM dataset for OS endpoint

Automated retrieval of metadata based on variables names

Metadata/Context

3. Use the following instructions:
- **STARTDT:** The origin... for the time-to-event ...ysis. This is generally the time at which a subject is first at risk for the event of interest evaluation (as defined in the protocol or SAP). [...]

User Prompt

4. User instructions (...tional):
```
**Specific instructions** that is not pulled from automatic retrieval.
```

Metadata/Context

5. <<<

Input code

```
/* create a dataset named os from
add.ADSL dataset */
data os;
  set add.ADSL;
  if ittfl = "Y";
  keep STUDYID USUBJID TRT01P  [...]

/*Merge all information*/
data os;
  /* Deriving startdt */
```

>>>

Output

Built prompt

```
Your task is to answer the user request below by generating
functional, efficient and documented SAS code with good
syntax.

### Instruction:
Create variable startdt of ADTTE ADaM dataset for Overall
survival endpoint:

Use the following instructions to create variables:
- startdt: The original date of risk for the time-to-event
analysis. This is generally the time at which a subject is
first at risk for the event of interest evaluation (as defined
in the protocol or SAP). For example, this may be the
randomization date or the date of first study therapy
exposure.  This may often be Parameter dependent (e.g.
Randomization date of efficacy parameters and date of first
study drug for safety parameters).
- USER INTRUCTIONS

###
Response:
```SAS
/* create a dataset named os from add.ADSL dataset */
data os;
  set add.ADSL;
  if ittfl = "Y";
  keep STUDYID USUBJID TRT01P TRT01PN TRT01A TRT01AN DTHDLDT
[...]

/*Merge all information*/
data os;
  /* Deriving startdt */
```

User inputs
Extracted informations
Template
```

GenAI Code Generator: Detailed view of POC steps & Go / NoGo gates

Evaluate ability to generate a Draft SAS code for SDTM to ADaM transformation for efficacy data in oncology (*incl.* code quality evaluation)

Decision gate 1

Based on evaluation metrics:

- If none of the LLM performs well on simple transformations → **stop the PoC**
- If WizardCoder, Code-Llama Instruct or OctoCoder perform well on simple transformations and show potential to perform well on medium ones → **go to Phase 2**
- If WizardCoder performs well on simple & medium transformations → **go to MVP**

Decision gate 2

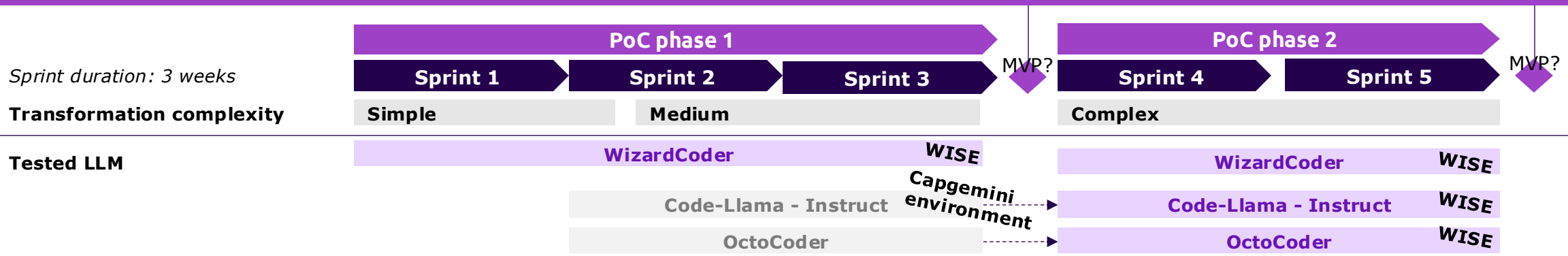
Based on evaluation metrics:

- If at least one model performs well or shows potential to perform well after prompt engineering on complex transformations → **go to MVP**

Disclaimer : PoC Phase 2 content depends on PoC phase 1 results

◆ Go/NoGo

PoC rolled out in 5 sprints of 3 weeks (a total of 15 weeks), with a Go/No Go decision gate after 3 sprints



GenAI Code Generator: Status

PoC Phase I: Tested on Oncology ADTTE OS (Simple Transformation) and PFS (Medium transformation)

- Five Studies were used as input:
  - ❖Protocol + SAP
  - ❖SDTM + ADaM metadata
  - ❖Complex Algorithm
  - ❖Program
- Tested and evaluated the 3 pre-selected LLMs based on objective and human-evaluation (SME).
- Selected one of the LLMs for Phase 2
- Created demo of preliminary SAS code

Good elements

- Knows and respects SAS syntax
- Has a good understanding of the task
- Creates efficient code when transformation is simple
- Provides interesting and relevant explanations about the code

Issues to be addressed at the end of PoC Phase 1:

- Will make up information if it is missing or unknown (halluciation)
- Logic can be off
- SAS code is fragmented
- Context is operated in blackbox

GenAI Code Generator: Status

PoC Phase I example

System prompt

Your task is to answer the user request below by generating functional SAS code with good syntax.

User prompt

Derive variable CNSR of ADTTE ADaM dataset for Overall survival endpoint.

Metadata / context

*Unstructured copy-paste of Note 1100K from **ComplexAlgorithm.pdf***

ADTTE.PARAMCD=OS
Algorithm used to derive the overall survival (OS).
Case #
Situation
Censor (CNSR)
Analysis Date (ADT)
Event or Censoring Description (EVNTDESC)
Censor Date Description (CNSDTDSC)
[...]

Output

Here's the SAS code to derive the variable CNSR of ADTTE ADaM dataset for Overall survival endpoint:

```SAS
/* Create a new variable CNSR */
data ADTTE;
  set ADTTE;
  if ADSL.DTHDLDT <= EFFCD then CNSR = 0;
  else if ADSL.FUEDT > EFFCD then CNSR = 1;
  else if ADSL.FUEDT <= EFFCD - 8 then CNSR = 1;
  else if ADSL.FUEDT > EFFCD - 8 then CNSR = 1;
  else if ADSL.FUEDT < EFFCD - 8 then CNSR = 1;
  else CNSR = 1;
run;
```

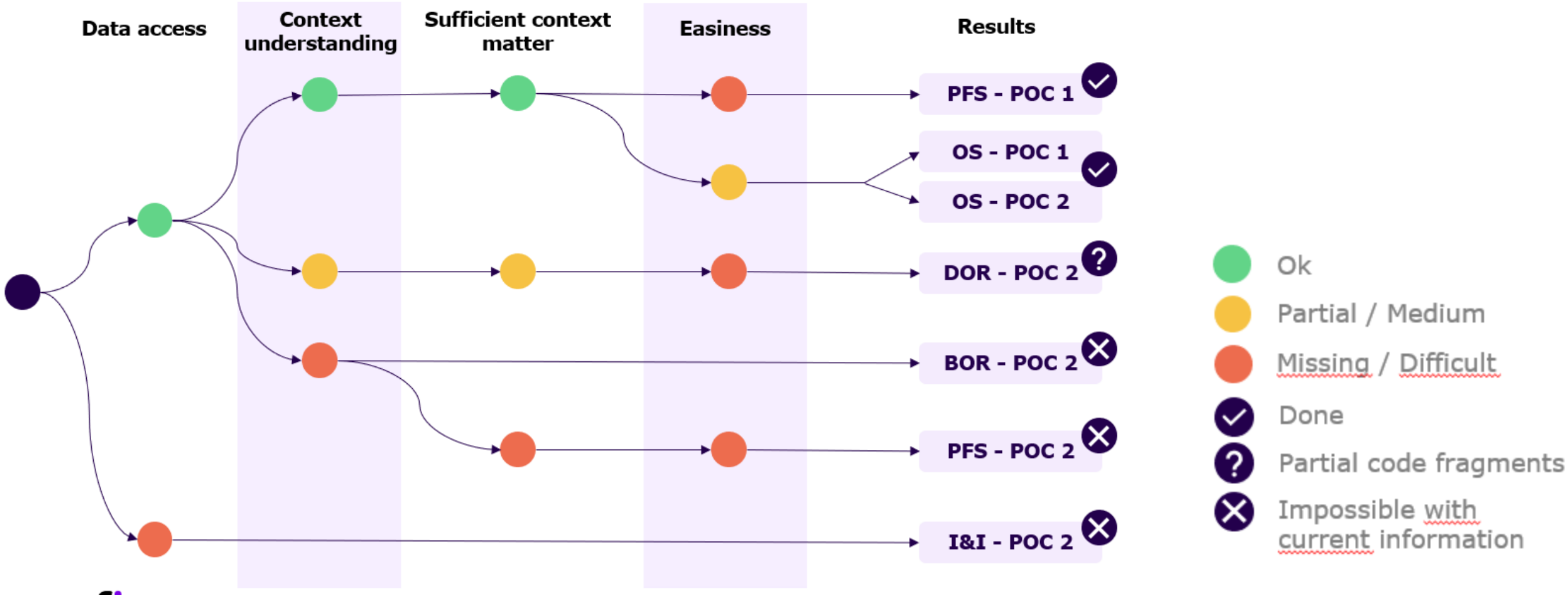
Explanation:

- The code first creates a new variable CNSR in the ADTTE dataset.
- The if-else statements are used to assign values to the CNSR variable based on the conditions specified in the context.
- If the death date (ADSL.DTHDLDT) is on or before the efficacy cutoff date (EFFCD), then the patient is considered to have died and CNSR is set to 0.
- If the last contact date (ADSL.FUEDT) is after the efficacy cutoff date (EFFCD), then the patient is considered to have survived and CNSR is set to 1.
- If the last contact date (ADSL.FUEDT) is between the efficacy cutoff date – 8 weeks and the efficacy cutoff date, then the patient is considered to have survived and CNSR is set to 1.
- If the last contact date (ADSL.FUEDT) is before the efficacy cutoff date – 8 weeks or missing, then the patient is considered to have survived and CNSR is set to 1.
- If the last contact date (ADSL.FUEDT) is missing, then the patient is considered to have survived and CNSR is set to 1.

GenAI Code Generator: Status

PoC Phase II: Generalization and extension on Solid tumor indications and I&I/Neuro

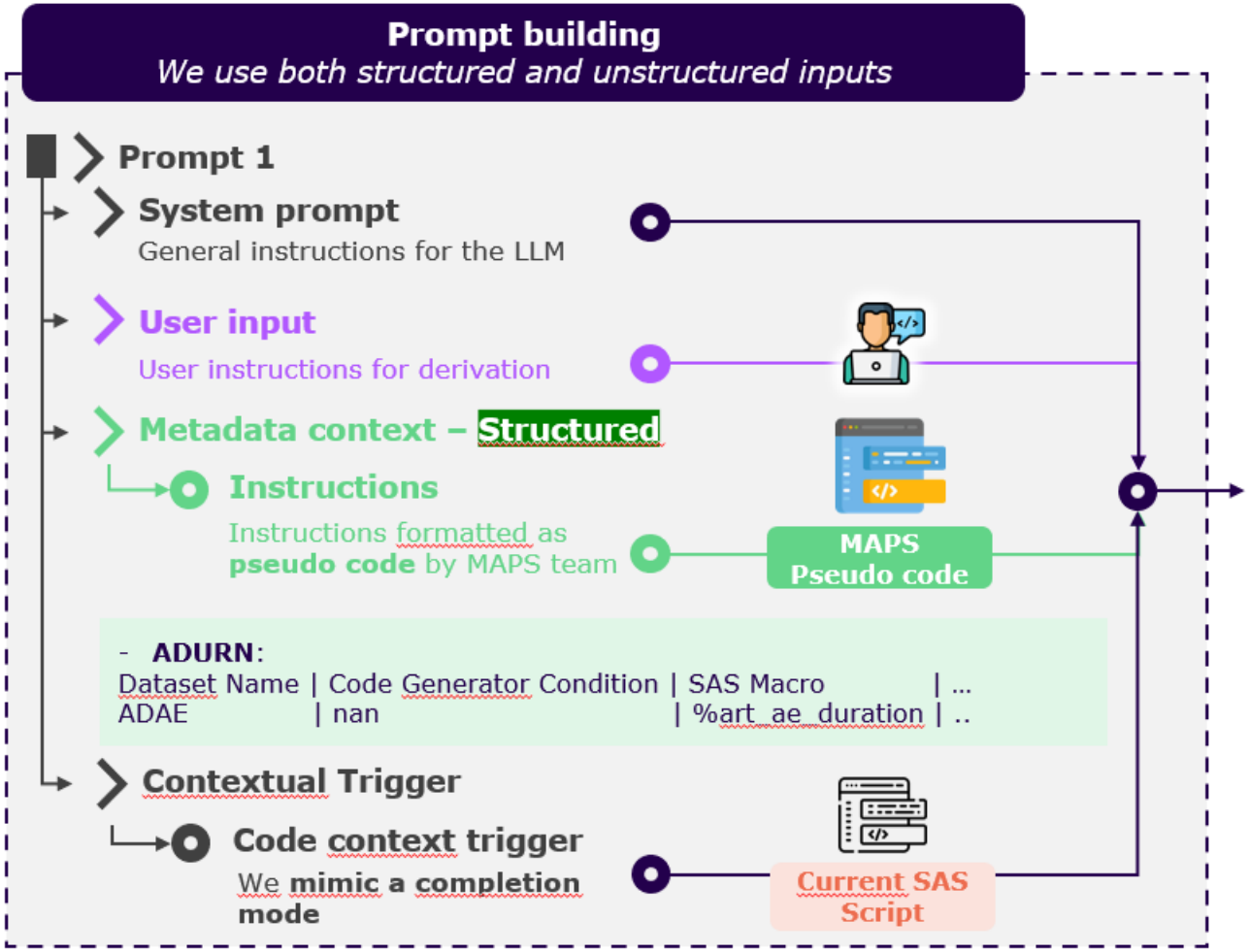
- Tested OS & PFS of 3 breast cancer studies, achieved only partial repeating of OS coding
- Discovered difficulty in understanding context or code generating for DOR and PFS
- Had no time to test on I&I and neuro studies



GenAI Code Generator: Status

PoC Phase II: experiment on Safety code generation relying on tabular metadata and pseudocode

- Tested on 3 variables per dataset: ADAE (ADURN, ADURNGR, ADURNGRN) and ADSL (AGE, AGEGR1, AGEGR1N)



Generated code

```
%art_ae_duration(
     data_in                = adae_dtc
    ,data_out               = adae_dtc
    ,art_key_list           = USUBJID AEREFID AESEQ
    ,art_key_var            = AEREFID
    ,art_duration_unit      = DAYS
    ,art_debug_flag         = N
    );

data adae_dtc;
    set adae_dtc;

    /* ADURNG */
    if missing(adurn) then adurng = .;
    else if adurn <= 3 then adurng = 1;
    else if adurn <= 7 then adurng = 2;
    else if adurn <= 14 then adurng = 3;
    else adurng = 4;

    /* ADURNGN */
    if adurng = 1 then adurngn = '1-3 days';
    else if adurng = 2 then adurngn = '4-7 days';
    else if adurng = 3 then adurngn = '8-14 days';
    else if adurng = 4 then adurngn = '>= 15 days';
run;
```

Coverage

Quality

- Code quality is excellent. The model calls the **relevant macro** for ADURN and **provide relevant derivation** for 2 other variables.*

GenAI Code Generator: Status

PoC Phase II: experiment on R

- SAS to R translation

```sas
/* create a dataset named os from add.ADSL dataset */
data os;
  set add.ADSL;
  if ittfl = "Y";
  keep STUDYID USUBJID TRT01P TRT01PN TRT01A TRT01AN
RANDTM RANDDT TRTSDT TRTSDTM FUEDT DTHDLDT RANDDTM
TRTSTM;
  rename TRT01P=TRTP TRT01PN=TRTPN TRT01A=TRTA
TRT01AN=TRTAN;
run;

/* sort by usubjid */
proc sort data=os;
  by usubjid;
run;

/* create variables paramcd, param, parqual and startdt
*/
data os;
  set os;
  paramcd = 'OS';
  param = 'Overall Survival (days)';
  parqual = 'INVESTIGATOR';
  startdt = RANDDT;
run;
```

Conversion by Mixtral

```r
os <- add.ADSL[add.ADSL$ittfl == "Y", c("STUDYID",
"USUBJID", "TRT01P", "TRT01PN", "TRT01A", "TRT01AN",
"TRTSDTM", "DTHDLDT", "RANDDTM", "TRTSDT", "FUEDT",
"RANDDT", "RANDDTM", "TRTSTM")]
names(os) <- c("STUDYID", "USUBJID", "TRTP", "TRTPN",
"TRTA", "TRTAN", "TRTSDTM", "DTHDLDT", "RANDDTM",
"TRTSDT", "FUEDT", "RANDDT", "RANDDTM", "TRTSTM")

os <- os[os$paramcd %in% 'OS', ]
os$param <- "Overall Survival (days)"
os$parqual <- "INVESTIGATOR"
os$startdt <- os$RANDDT

os <- os[order(os$USUBJID), ]artdt = RANDDT;
run;
```

Conversion by WizardCoder

```r
os <- add.ADSL %>%
  filter(ittfl == "Y") %>%
  select(STUDYID, USUBJID, TRT01P, TRT01PN, TRT01A,
TRT01AN, RANDTM, RANDDT, TRTSDT, TRTSDTM, FUEDT,
DTHDLDT, RANDDTM, TRTSTM) %>%
  rename(TRTP = TRT01P, TRTAN = TRT01AN) %>%
  mutate(paramcd = "OS",
         param = "Overall Survival (days)",
         parqual = "INVESTIGATOR",
         startdt = RANDDT)
```

Summary: Exploration of Generative AI for code generation

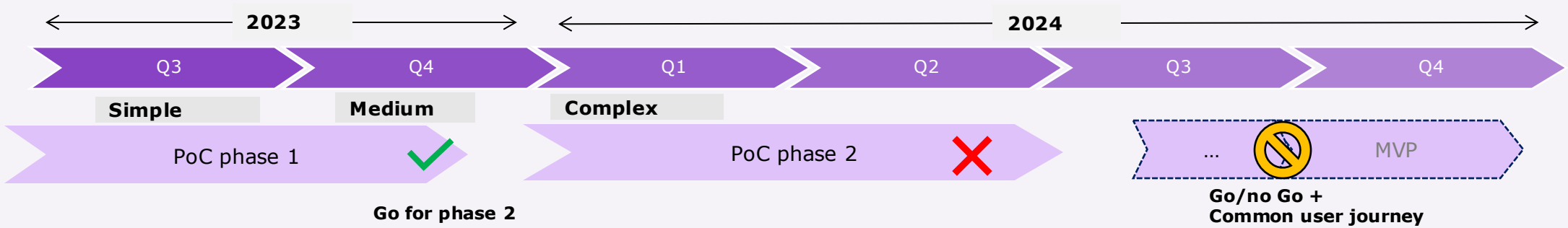
Collaborative work started in June 2023 with CapGemini Invent, B&P, and eWISE support for infra & tech.
In case of success, this could expand the automation scope of MAPS.

Primary scope & objectives

Explore **feasibility** of **using Large Language Models (LLM)** to **semi-automate the generation of SAS code for study data preparation** where standard mapping is not available (scope = efficacy data).

Exploration assessment

- **Technical** feasibility
- **Business** value evaluation
- Fit of **GenAI code generation into programmer's workflow**



Completed

Proof of Concept – phase 1

Results showed **capacity to generate a draft version of a SAS code for Onco efficacy endpoint**; promise but required refinement in addressing generalization.

Proof of Concept – phase 2

Generalization of prompting strategy with the current available samples:

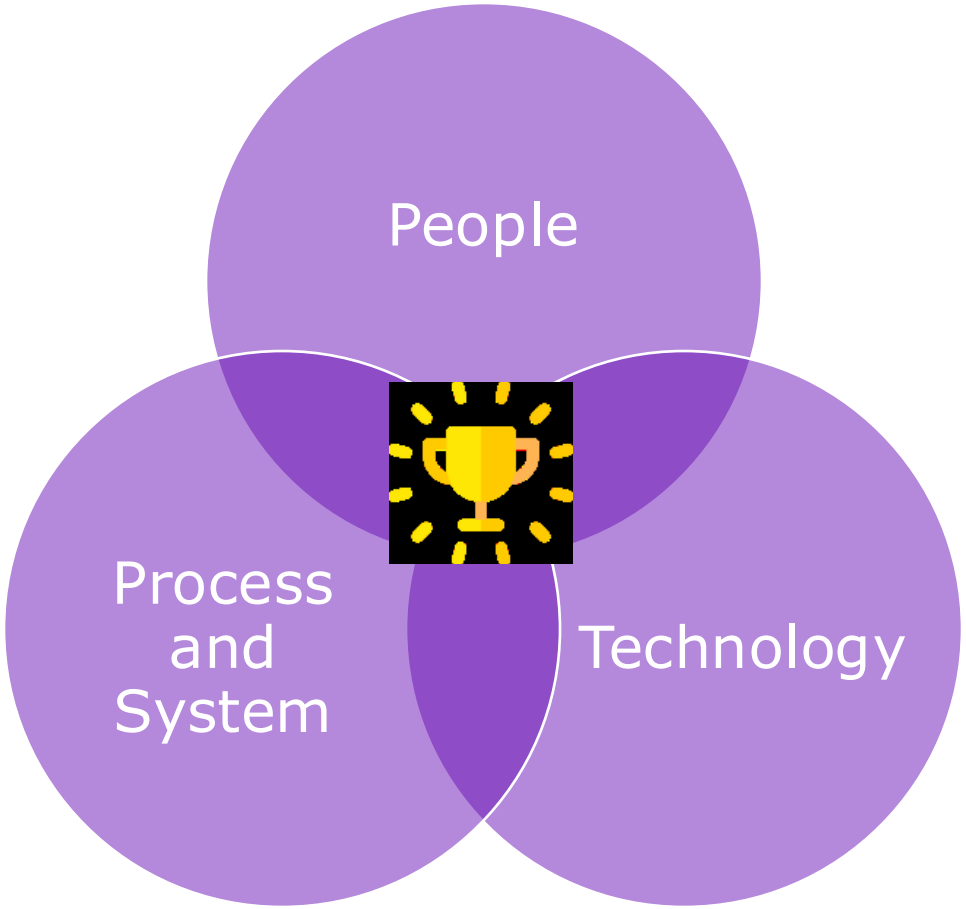
1. Extension to other TA not performed due to delays in obtaining the data
2. Generalization of Oncology efficacy endpoints to different indications was not achieved

Additional not initially planned results

1. In depth analysis of current programming process
2. Identification of potential target context format to enhance code generation
3. Initial generation of SAS code based on structured safety dataset metadata with pseudo code

sanofi

Looking Forward



- Metadata
- Code Generation
- Automation
- Quality Control
- Documentation
- Submission

sanofi

Q&A

Thank you!