

Micro-Decisions, Macro Impact

Cultivating an Agile Mindset in Every Line of Statistical Code

Eswara Guniseti, PHUSE EU Connect 2025, Hamburg
Stream: Coding Tips & Tricks (CT)
Presentation No. CT14

Disclaimer

The slides and views expressed in this presentation are those of speakers and are not representative of Cytel's views or opinions.

Agenda

- Introduction
- Think Agile
- Agility in action
- Conclusion
- Q & A



Introduction

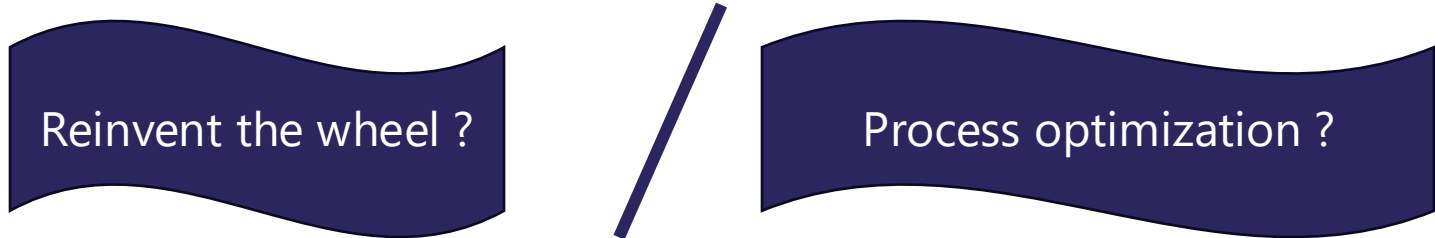
An agile mindset is a way of thinking that prioritizes **adaptability**, **flexibility**, and **continuous learning**.

In Clinical Statistical Programming we focus on specifications, datasets and outputs, yet all those lie in the micro decisions that we make everyday such as,

How we name a variable, structure a dataset and design a macro.

Each of these choices may seem small but together they shape how agile our work becomes.

To achieve better results how do we choose the best **path forward** ?



Reinvent the wheel ?

Process optimization ?

Reinventing the wheel - Total reset

Process optimization – Enhancing existing systems by sticking to what we already learnt.

Process Optimization

Kaizen Model & Toyota lean model principles

- Continuous improvement
- With a culture of cooperation
- Eliminating waste

Think Agile

Agility isn't only a team principal, **it's a personal practice**

For statistical programmers, this means:

- ❑ Writing with empathy
- ❑ Building with change in mind
- ❑ Prioritizing clarity and consistency
- ❑ Favoring adaptability over perfection

Agility in Action

Few impactful areas where micro-decisions can embody the Agile mindset

- ✓ **Intentional Naming**
- ✓ **Smart Commenting**
- ✓ **Scalable Macros**
- ✓ **Readable Structure**
- ✓ **Built-in Quality**

1. Intentional Naming

Names that Clarify, Not Confuse

Compare Result : LST with more than one compare result

*Modify the comparing dataset names
more meaningful for better understanding*

1st Compare result

```

The COMPARE Procedure
Comparison of WORK.PROD with WORK.FINAL_TAB
(Method=EXACT)

Data Set Summary

Dataset              Created              Modified      NVar      NObs
WORK.PROD             11SEP25:08:31:46    11SEP25:08:31:46      5        209
WORK.FINAL_TAB        11SEP25:08:31:46    11SEP25:08:31:46      5        209

Variables Summary

Number of Variables in Common: 5.

```

2nd Compare result

```

The COMPARE Procedure
Comparison of WORK.PROD with WORK.FINAL_FIG
(Method=EXACT)

Data Set Summary

Dataset              Created              Modified      NVar      NObs
WORK.PROD             11SEP25:08:31:48    11SEP25:08:31:48      5        132
WORK.FINAL_FIG        11SEP25:08:31:47    11SEP25:08:31:47      5        132

Variables Summary

Number of Variables in Common: 5.

```

Standards-based/sensible naming makes your code understandable and future-ready.

2. Smart Commenting

Comment the **Why**, Not the **What**

Scenario:

(i) /* **Sort data by subject and visit** */
proc sort data=advs; by usubjid avisit; run;

(ii) /* **Align visits in chronological order for exposure-response analysis** */
proc sort data=advs; by usubjid avisit; run;

The Agile comment explains **why** the sort matters in context

3. Scalable Macros

Write Once, Use Many Times

```
data adam_;  
  set adam;  
  if paramcd="HR" then aval=aval*60;  
run;
```

```
%macro unit_convert(ds=, param=, factor=);  
  data &ds._;  
    set &ds;  
    if paramcd="&param" then aval=aval*&factor;  
  run;  
%mend;  
  
/* Call for multiple parameters */  
%unit_convert(ds=adam, param=HR, factor=60);  
%unit_convert(ds=adam, param=WTKG, factor=0.001);
```

Agility - you adapt once instead of fixing the same bug in 10 places.

- ✓ **Easily adaptable across the studies**
- ✓ **Automate the treatment column numbers and macro parameter values basing on data than predefining them**
- ✓ **Use 'Do Looping' or other SAS functionality for better flexibility**

4. Readable Structure

Code that Explains Itself: By consistent in indentation, alignment & logical grouping of steps.

```
/*Hard to read*/  
data adsl;set raw;age=round(age,1);if sex='M' then group='Male';else group='Female';run;
```

```
/*Cleaner version*/  
data adsl;  
  set raw;  
  age = round(age, 1); /* Round to nearest year */  
  if sex = 'M' then group = 'Male';  
  else group = 'Female';  
run;
```

- ✓ Well-structured code is self-documenting. Format for **readability**, not just **execution**
- ✓ The cleaner version takes the same seconds to run, but **minutes less to review** and hand over.

5. Built-in Quality

QC Starts in the Code :

Program with defensive checks for anticipated common issues/errors/warning prevents downstream rework.

```
/* Check for unmatched records before merge */
```

```
proc sort data=adsl ; by usubjid; run;  
proc sort data=advb ; by usubjid; run;
```

```
data adsl_advb;  
  merge adsl (in=a) advb (in=b);  
  by usubjid;  
  if a and not b then put "WARNING: Subject in ADSL not found in ADVS: " usubjid=;  
  if b and not a then put "WARNING: Subject in ADVS not found in ADSL: " usubjid=;  
run;
```

"Agile code doesn't just assume the merge works,
- it **checks, flags, and communicates** issues early, so errors don't snowball later."

Benefits at Team-Level

- ✓ **Shorter turnaround times** due to cleaner, reusable components.
- ✓ **Easier collaboration** when logic and assumptions are transparent.
- ✓ **Better risk management** through modular, testable code.
- ✓ **Healthier teams** with less rework and greater code confidence.

Leadership's Role in Embedding Agility

As a team lead or project manager, to cultivate the Agile mindset,

- ✓ **Normalize code reviews as learning sessions**, not fault-finding missions.
- ✓ **Create shared libraries** built on Agile-friendly practices.
- ✓ **Set expectations that code should explain itself**, not require decoding.
- ✓ **Celebrate reusability and clarity**, not just speed.

Conclusion

- At its core, code is communication.
- Every variable, macro, and comment is a message to a collaborator, a regulator, or your future self.
- Support the programmer that spent thinking deeply, naming carefully, structuring smartly. That's where true Agile excellence starts.

REFERENCES

1. “White Paper - Clinical Trial SAS Programming Needs a Makeover: Agile Leads the Way” - <https://cytel.com/perspectives/moving-to-agile-a-new-approach-to-statistical-programming/>
2. “Looking at statistical programming as a software development process” https://www.lexjansen.com/phuse/2024/li/PAP_LI08.pdf

Questions ?



Thank you

Contact: eswara.gunisetti@cytel.com

