

Micro-Decisions, Macro Impact - Cultivating an Agile Mindset in Every Line of Statistical Code

Eswara Satyanarayana Guniseti, Cytel Inc. UK, London, United Kingdom

ABSTRACT

In the world of statistical programming, agility isn't just about stand-ups or sprints, it's woven into every tiny decision a programmer makes. From how we name variables, design macros, structure datasets, or write comments, every line of code either builds flexibility or bakes in future friction.

This talk explores how agile thinking: clarity, simplicity, adaptability, and value orientation can be applied directly at the code level. We'll discuss how small programming habits, like modularizing code, documenting intent (not just process), and writing for readability, have outsized impacts on reusability, QC efficiency, and audit readiness.

Through real-world examples from SDTM, ADaM, and TLF workflows, we'll show how these micro-decisions lead to macro gains: faster change implementation, clearer handovers, and a stronger sense of ownership. This is not about doing more, it is about thinking agile while doing what we already do, better.

INTRODUCTION

In today's fast-paced world of clinical research, the statistical programmer is more than a coder; they are dynamic problem solver, collaborator, and quality guardian.

It is not always wise to think about reinventing the wheel by adapting to a new process altogether. We can stick to the existing process and still achieve greater efficiency and effectiveness by enhancing the way we do the things.

Proven models like Kaizen and Toyota Lean manifest that with continuous improvement with a culture of cooperation within employees and eliminating waste we can deliver maximum value to the customers by sticking to the existing process and not letting go of what we already learnt.

As data complexity grows and timelines shrink, agility has become an essential trait, not just in workflows, but in mindset. While Agile methods like sprints and stand-ups guide our teams, the true power of "being agile" lives in the smallest, most frequent decisions made behind the keyboard.

This document explores how statistical programmers can embed Agile thinking in every coding choice, transforming micro-decisions into drivers of long-term efficiency, resilience, and impact.

AGILE THINKING FROM THE FIRST LINE OF CODE

Agility isn't only a team principle; it is a personal practice.

For statistical programmers, this means:

- **Writing with empathy:** Code isn't just for machines; it is for colleagues, reviewers, and your future self.
- **Building with change in mind:** Structure programs so they flex instead of break under evolving requirements.
- **Prioritizing clarity and consistency:** Simple, well-named variables, clear comments, and clean formatting communicate better than clever shortcuts.
- **Favoring adaptability over perfection:** Focus on delivering something valuable and refine it iteratively.

These ideas might sound subtle, but together, they create the difference between **rigid code** and **resilient code**.

MICRO-DECISIONS, MAJOR OUTCOMES

Each line of code is a choice. The sum of these choices shapes our ability to adapt, maintain, and scale.

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

1. Intentional Naming
2. Smart Commenting
3. Scalable Macros
4. Readable Structure
5. Built-In Quality

1. INTENTIONAL NAMING

Names are narratives. Properly naming the programs, datasets, macros, and outputs instantly reduces confusion.

Consider a scenario where more than one output was being compared using same validation program. Uniquely naming the datasets avoids confusion rather than keeping the same name for both comparisons.

Following is one such scenario, here two outputs were being compared using same program and at both places, name of the datasets were same. This will not compromise the comparison but confuse the reviewer.

The COMPARE Procedure Comparison of WORK.PROD with WORK.FINAL_TAB (Method=EXACT)				
1st Compare result				
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.				
The COMPARE Procedure Comparison of WORK.PROD with WORK.FINAL_FIG (Method=EXACT)				
2nd Compare result				
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.				

Points to consider:

ADaMs: While adding a non-standard variable to ADaM dataset, rather than giving a random name try to give more meaningful naming by considering its role.

- ✓ End the name with *FL if the new variable is supporting subject level or record level flagging.
- ✓ End the name with *DT/*DTM if the new variable holds date and time information.
- ✓ Use names like ANLxxFL/CRITxxL if the new variable helps to select analysis records/specific criterion.

SDTMs: While adding non-standard information to SUPPxx dataset, name the variable in more meaningful way by relating the name to its parent variable if available or resemble key text that is being collected.

TLFs: While saving production dataset for Qc/validation purposes, keep only the variables that are supporting the report with meaningful names to let reviewer understand it better.

2. SMART COMMENTING

Commenting is not about restating the code - it is about providing context.

- ✓ Context about why this approach was chosen.
- ✓ Details about assumptions/decisions made.
- ✓ Useful link/email text to support the decisions.

Example 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;
```

In (ii), comment explains **why** the sort matters in context.

Good comments reflect decisions and aid auditors, collaborators, and even your future self.

3. SCALABLE MACROS

Macros built with parameters, metadata input, and clean separation of logic are future proof.

Instead of one-off fixes, design tools that solve broader problems. Agile thinking encourages small investments today for exponential time savings tomorrow.

Example scenario:

```
/* Plain Code */  
data adam;  
  set adam;  
  if paramcd="HR" then  
    aval=aval*60;  
run;
```

```
/* Macro Code to reuse */  
  
%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);
```

Points to consider:

- ✓ Do not restrict the macro program to specific study or project.
- ✓ Implement in such a way that macro can be easily adopted across the studies/projects.
- ✓ Automate treatment columns, parameters usage.
- ✓ Use Do Looping concept to execute code snippet maximum number of times basing on data than restricting for a defined number.

4. READABLE STRUCTURES

Whitespace, section headers, indentation, this might seem cosmetic, but it tells reviewers and collaborators: “I respect your review time.” It makes validation easier and faster.

Example scenario:

```
/*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;
```

5. BUILT-IN QUALITY

Agile isn't about skipping checks for speed. It is about embedding quality throughout. Using test datasets, logging, or conditional checks as part of normal code flow reflects an Agile commitment to "done means tested."

Following is one such scenario where a simple conditional check placed in the program makes sure no subjects are missing out.

Example scenario:

```
/* Check for unmatched records before merge */  
proc sort data=adsl ; by usubjid; run;  
proc sort data=advs ; by usubjid; run;  
  
data adsl_advs;  
  merge adsl (in=a) advs (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;
```

Points to consider:

- ✓ Use PUT statement to write the results to log file to verify the data discrepancies.
- ✓ Differentiate the user defined key text like ERROR/WARNING to avoid confusion.

BENEFITS OF EMBEDDING AGILITY

A. TEAM LEVEL RIPPLE EFFECTS

When statistical programmers apply Agile principles to micro-decisions, it transforms how teams' function

- ✓ **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.

B. LEADERSHIP'S ROLE

As a team leader or project manager, cultivating this mindset takes intention:

- ✓ **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: CODE AS COMMUNICATION

At its core, code is communication. Every variable, macro, and comment is a message to a collaborator, a regulator, or your future self.

Embracing the Agile mindset is not about doing more, it's about doing better, one thoughtful, intentional decision at a time. The future of statistical programming belongs not just to those who can code fast, but to those who code with purpose.

Let every line you write carry clarity. Let every structure you build invite change. Let every decision reflect agility. That's the path from micro-decisions to macro impact.

REFERENCES

- "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/>
- "Looking at statistical programming as a software development process" https://www.lexjansen.com/phuse/2024/li/PAP_LI08.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Eswara Satyanarayana Guniseti

Company: Cytel Inc. UK

Address: Aldwych House, 71-91 Aldwych, London WC2B 4HN

Work Phone: 07586991314

Email: eswara.guniseti@cytel.com

Website: www.cytel.com

Brand and product names are trademarks of their respective companies.