Good Programming Practice – Code Review

Introduction

Section content first added by: Nicky Fahers 2010-11-19

How do you know if you and your colleagues are actually following Good Programming Practice?

The main aim of code reviews are to make sure that all SAS programs are following Good Programming Practice.

Benefits of Coding Reviewing

Help everyone to write more efficient programs

- Less codeLess time to run a programLearning/development process when and how to apply techniques to improve code quality consistency and maintainability

Help everyone to write programs that adhere to the Good Programming Practice

- Produce code that is fit for use. Produce code that new users can easily follow and adapt (so helping the rest of the department- reduces time new users have to spend figuring out a program that only needs a minor update)

Implementation of a Code Review Process

All companies will tackle code review in different ways and some will not perform any at all. Here are a few examples of how code review has been performed in other companies:

Scenario 1

Everyone in the department who writes SAS programs will have their code reviewed by the Lead programmer of the study. The code reviewer will review the code and give informal feedback to the Lead programmers code will be reviewed by the code review team.

- **Note all comments are confidential and will NOT be fed back to anyone else.**

Ideally it would be code you have written from scratch If not:

Code you have changed substantially.

or

A section of code that you have added or adapted. Highlighting clearly what you have added or changed.

Common findings are presented in department meeting and training is given accordingly.

Scenario 2

Code reviews are only performed on people with less than 6 months experience.

Common Findings

General	Efficiencies	Advice

• No Program Header • Header with incorrect details or not enough detail • No comments throughout the program • Over writing datasets over and over again	• Lots of unnecessary proc sorts • Lots of unnecessary data steps • Incorrect merging of datasets (learning how to use proc sql) • Use arrays and proc transpose- know when to use them	• Use SAS functions • Use standard macros • Create a macro for repeating code • Simple is best- do not over complicate things **Plan and structure your program before you start**