



IMPLEMENTATION OF AUTOMATIZATION AS A WAY TO DECREASE THE PROGRAMMING TIME

12th October 2018, Warsaw
Igor Dedikov

PAREXEL[®]
YOUR JOURNEY. OUR MISSION.[®]

AGENDA

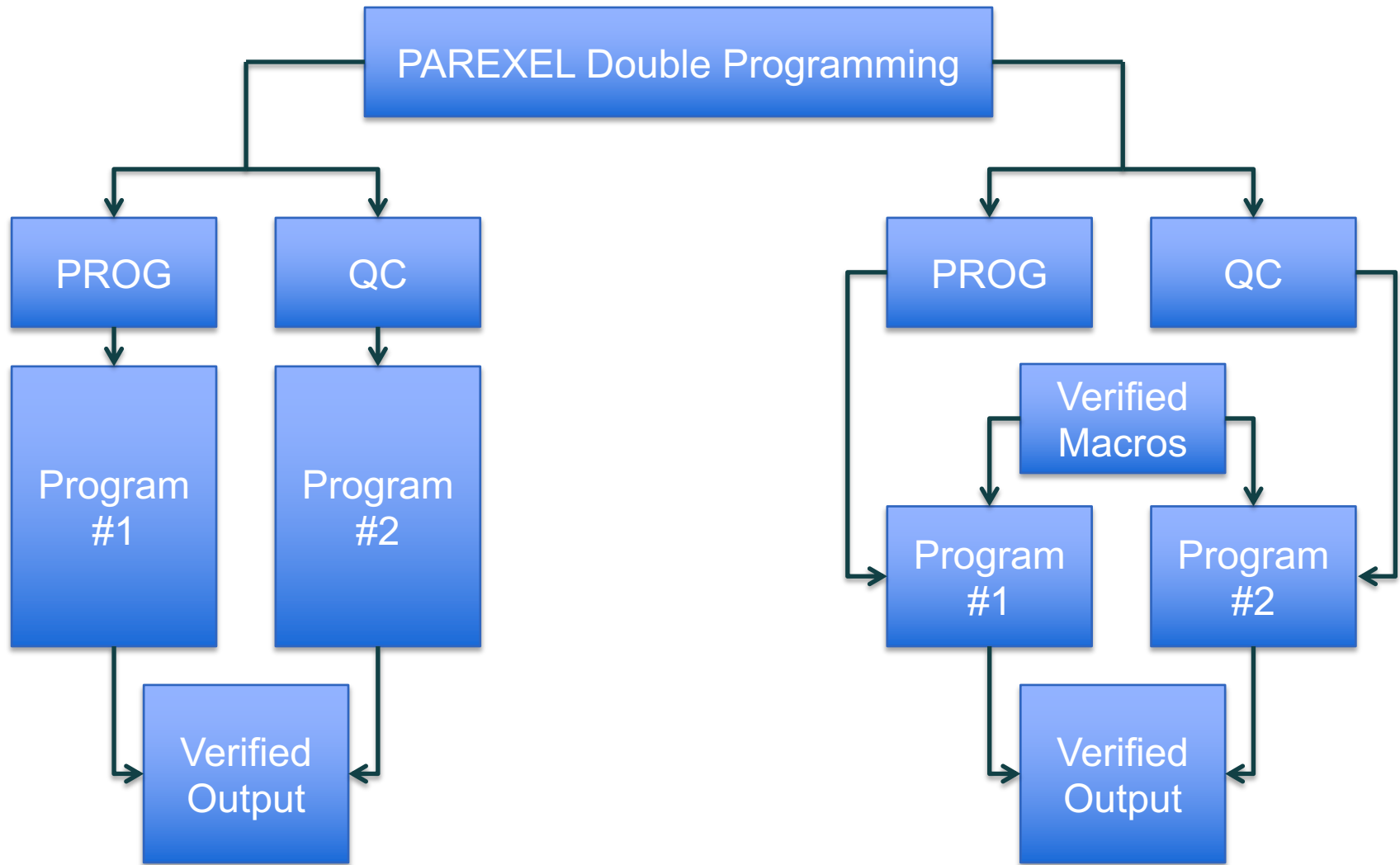


- Introduction
- Key principles of automatization
- Tests and Results
- What's next?

INTRODUCTION



INTRODUCTION - I



Time consumers in non-core outputs:

- Different templates for the same outputs;
- The lack of SAS macro code in programs:
 - Different amount of treatments;
 - Different column labels and by-line labels;
 - Scaling problems (it is easier to rewrite than to update the macros);
- The lack of testing:
 - Null data case;
 - Big amount of treatments.

KEY PRINCIPLES OF AUTOMATIZATION

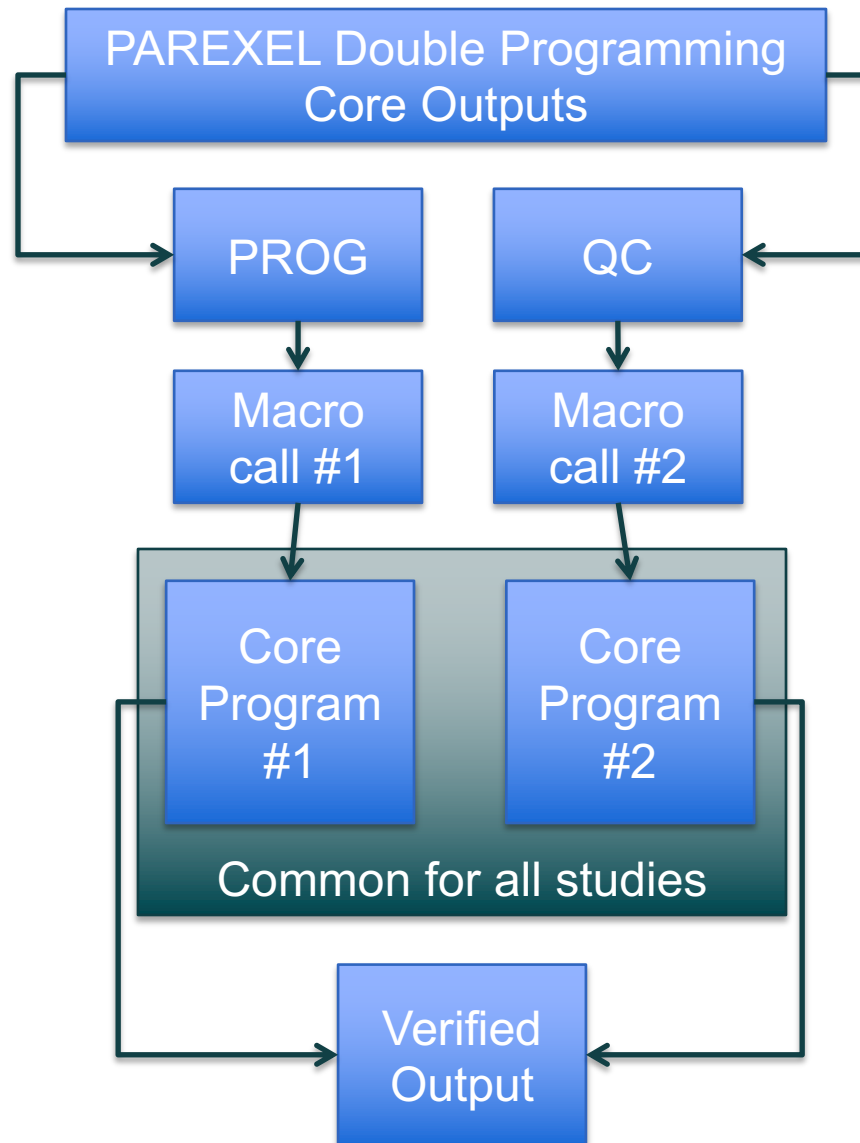


KEY PRINCIPLES OF AUTOMATIZATION - I

Key ideas for automatization of programming process:

- To develop and harmonize with the customer the templates for unified outputs;
- To split the program into 2 parts:
 - Core part. This code is common for all projects. It works like an output factory – you give data to the input and you get standardized output;
 - Macro call part. The setup part of code that prepares data to them into core macros. It is project specified part of the code;
- To create production and testing areas for core macros:
 - Core macros should be tested on any study using any data before they could be used by everyone;
 - To update / fix core macros that are in use we need to created an instrument to test the changes on the majority of the projects.

KEY PRINCIPLES OF AUTOMATIZATION - II



The contents of macro call:

- The amount of treatment groups for the outputs (could be configured for the whole study in the setup file);
- Names for columns and labels for by-line groups (could be configured for the whole study in the setup file);
- Preparation of input datasets (could be configured for the whole study with a help of adjuvant macros);
- Configuring macro variables for the core macros;
- Macro call itself with correct macro parameters.

A wide-angle photograph of the Boston skyline at sunset. The water in the foreground reflects the colorful buildings and the sky. A large ship is docked on the left, and a chain is visible in the immediate foreground. The text "TESTS AND RESULTS" is overlaid in white, italicized font, underlined.

TESTS AND RESULTS

Test process for usage of core macros:

- Before the implementation new core macros is checked in the testing area on real data from the study. Outputs made by core macros and by project specific program should match. Only when core macro gives the same result on several projects it is introduced into the production area of core macros;
- There are special data cases that help test new macros on zero data;
- The testing process for core macros that are already in use:
 - Test changes on 2-3 studies to test new updates;
 - Test changes on 4-5 deliveries on different studies. A special environment helps to recreate an area with data from specified delivery and run there the updated core macros. This step helps us to understand that changes won't cause the crash for the current performance of core macros in production area.

Macro revision system:

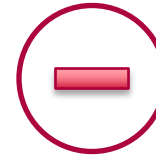
- When there is a update of core macros all programming Leads are informed about the changes;
- Before the macros update the whole production area is copied and named by the revision number;
- If there are any troubles with new core macros and there is an urgent delivery, the programmer Lead can easily use any previous revision of core macros.

TESTS AND RESULTS - III

Core Macros Pro and Contra



- Core macros were implemented into more than 29 studies and programming time for a big part of the outputs (from 30% to 50% of the whole study) decreased at least twice;
- Programmers who developed core macros got a good programming practice - it can be considered as a non-standard task and it can be used to motivate programmers who are interested in technical tasks.



- Late response – to update the macro there is a need to test it on different studies.
- The support of core macros could be a bit difficult for new people;
- As programming of algorithms is done in core macro, programming of macro calls doesn't contribute into programming experience and general clinical trial understanding.

WHAT'S NEXT?



The future of core macros in PAREXEL:

- The customer is ready to enlarge the amount of core outputs and PAREXEL is preparing new macros to become core;
- The development of test environment:
 - The changes are tested on more studies;
 - There are plans about an automatizing of the output check system (automatically compare outputs made by core macros from production and testing area);
- During creating the core macros new programming standards were created. Old core macros need to be reworked to fulfill those standards.

THANK YOU