Paper TT04

Twice As Much Ain't Twice As Good: Is Double Programming Really The Gold Standard We Think It Is?

Vitali Gering, Chrestos GmbH, Essen, Germany

ABSTRACT

The ultimate goal of conducting clinical trials is to bring life-saving treatments to patients as quickly as possible, while ensuring safety and efficacy. To achieve this, for over two decades our industry has relied on double programming as the gold standard of quality control. Yet, we are the only industry that relies so heavily on a method that is not only resource intensive but also inefficient. Its effectiveness depends on the availability of perfectly accurate specifications.

In this presentation, we share both the successes and challenges of rethinking validation beyond traditional double programming. We demonstrate how code traceability and the latest advances in artificial intelligence have enabled us to build a new model for quality control. We share how such an approach might be better suited to the evolving demands of statistical programming and the urgency of delivering new therapies to patients with greater speed and confidence.

INTRODUCTION

In double programming (DP), two statisticians or statistical programmers write code independently to create datasets, tables, listings or figures (TLFs). These results are then compared and any discrepancies discussed until an agreement is reached. DP therefore serves to reduce human errors, such as coding mistakes or misinterpretation of requirements, and agencies such as the FDA and EMA expect robust quality control measures to be implemented in clinical trial submissions. What has been considered as best practice widely throughout the industry, may require critical scrutiny nowadays. Although the very goal of DP is to guarantee correctness, it relies heavily on the assumption that if two independent individuals produce the same results, they must be correct. This statement must be treated with caution.

Why has DP been established as the current gold standard? What prerequisites need to be met, and what can go wrong if they are not? What are the potential alternatives and promising future trends that are worth considering? This paper addresses these questions and aims to encourage readers to be sceptical and question the status quo of today's quality control (QC).

THE GRASS IS ALWAYS GREENER ON THE OTHER SIDE

After graduating from university, I started my first job in the pharmaceutical industry as a statistical programmer. I was quickly introduced to the concept of independent DP, which I accepted without question. Lacking experience and the thinking outside the box, I accepted this unfamiliar validation process anyway. As I gained expertise in my field and related roles in the pharmaceutical industry, and through conversations with friends and acquaintances, I learned that DP is rather unique to our industry. I took the opportunity within the scope of this paper and immersed myself in the world of other professions just enough to see and roughly understand what they do – and why we don`t. Below are the two examples that I picked.

SOFTWARE DEVELOPMENT

Common quality control practices in software development include automated unit, integration, system and acceptance testing. Ongoing changes are also built and tested automatically to catch errors early. Peer reviews are also commonplace, helping junior developers by having their code reviewed by more experienced peers. Although this approach provides a steep learning curve, especially for newcomers, I have not seen or heard of it being followed too often. Should we follow it? With the latest developments in agentic AI, the concept of peer reviews is likely to change anyway and influence our work. However, the pharmaceutical industry tends to be risk averse, particularly with regard to regulatory affairs, and with a proven method such as independent DP available, why change something that works?

AEROSPACE

Would you trust an aeroplane whose software was developed using double programming? Well, aerospace engineers certainly would not. Instead, they rely on a broad, layered QC approach. This includes formal safety analyses, environmental and qualification testing, model-based simulation, and hardware-in-the-loop testing. For critical software, independent verification and validation (IV&V) is used. Although IV&V might sound similar to DP, as both aim to

minimise risk and ensure compliance with stringent quality standards, there are major differences. IV&V involves a separate team that evaluates the entire software system - from requirements and design to code and testing - to confirm that it meets the specifications and intended use. In short, while DP checks outputs through replication, IV&V checks systems through independent assessment.

THE EVOLUTION TO THE GOLD STANDARD

Prior to the 1990s, single programming and manual checks were commonly practiced. This approach relyied heavily on the expertise of the individual assigned to work on a clinical study. Although far from efficient, manual reviews or spot checks by the validator were standard practice. This limited validation process was prone to oversight, particularly regarding complex analyses or large datasets.

Incorrect dose calculations and misreported efficacy data are examples of major mistakes that led to costly delays and regulatory actions in the 1990s, resulting in increased regulatory scrutiny. At the same time, the complexity of clinical trials increased, undermining the credibility of single programming and demanding higher standards of data quality. Guidelines were also introduced to help ensure data integrity and statistical rigor: ICH E6 (Good Clinical Practice) in 1996 and ICH E9 (Statistical Principles for Clinical Trials) in 1998. Let us look at what exactly was specified in terms of double programming.

ICH E6 (GOOD CLINICAL PRACTICE)

Section 5.5.3 (Trial Management, Data Handling and Record Keeping) from ICH E6 (R2) says

"When using electronic trial data handling and/or remote electronic trial data systems, the sponsor should ensure and document that the electronic data processing system(s) conforms to the sponsor's established requirements for completeness, accuracy, reliability, and consistent intended performance (i.e., validation).".

ICH E6 does not mention the term DP at all. However, sponsors are urged to adhere to concepts such as **reliability** which can be achieved through the use of DP.

ICH E9 (STATISTICAL PRINCIPLES FOR CLINICAL TRIALS)

To harmonize statistical principles for clinical trials on a global scale, ICH E9 has been introduced back in 1998. **Reliable** and **interpretable results** are key, so making sure trials are well-designed, properly analyzed as well as accurately reported is of high importance. Section 5.1 (Prespecification of the Analysis) states that

"The statistical analysis should be described in detail in the protocol or statistical analysis plan. Any deviations from the planned analysis should be described and justified in the clinical study report."

There are a clear emphasis and need for a pre-specified and detailed Statistical Analysis Plan (SAP). Statistical Programming shall strictly follow this plan, and deviations must be justified. Independent DP naturally strengthens the confidence of adherence to the SAP. Moreover, section 5.5 (Reproducibility and Sensitivity Analyses) states that

"The analysis should be reproducible and the results should be robust to alternative assumptions.",

hence requiring **reproducibility** and **robustness**, which again is supported by independent DP. While ICH E9 does not explicitly mention DP, the adherence to fundamental concepts outlined in this guideline are implicitly supported by the use of DP.

U.S. FOOD AND DRUG ADMINISTRATION (FDA)

The FDA highlights a strong focus on documentation, **reproducibility**, **reliability** of software, submission of programs and appropriate testing and/or validation. However, it does not explicitly require independent DP to validate statistical programs. It rather recommends the sponsor to provide analysis programs and to document how datasets and outputs were created to that reviewers can understand it and reproduce results, if needed. Often, the term "should" is used rather than "must", showing that the FDA provides guidance to facilitate submissions instead of forcing sponsors to

follow certain approaches. During my research, I also came across a couple of quotes I would like to share here that may or may not be clear to some of you, too:

"FDA does not require use of any specific software for statistical analyses."

Looking at all the discussions currently (e.g. SAS vs R), I find it worth mentioning that the FDA actually does not mandate a specific software. The FDA however expects

"to be able to reconstruct a study"

as well as

"Sponsors should provide the software programs used to create all ADaM datasets and generate tables and figures associated with primary and secondary efficacy analyses."

and that

"The main purpose of requesting the submission of these programs is to understand the process by which the variables for the respective analyses were created and to confirm the analysis algorithms and results.".

The above quotes imply that outputs submitted to authorities should be **reproducible** and **traceable**. Independent DP in fact is exactly a reproducibility/verification technique.

EUROPEAN MEDICINES AGENCY (EMA)

Analogues to the FDA, the EMA also does not explicitly require independent DP to be practiced. It states

"The sponsor should ensure that appropriate and documented quality control of statistical programming and data analysis is implemented.",

and

"The sponsor should ensure the traceability of data transformations and derivations during data processing and analysis."

As well as

"The sponsor should retain the statistical programming records that relate to the output contained or used in reports of the trial results, including quality control/validation activities performed. Outputs should be traceable to the statistical software programs, dated and time stamped, protected against any changes, and have access controls implemented to avoid inappropriate viewing of information that may introduce bias."

and finally

"The approach to validation should be based on a risk assessment that takes into consideration the intended use of the system and the potential of the system to affect human subject protection and reliability of clinical trial results.".

Once again, terms like **documentation** (also of QC) and **traceability** seem to be key. Also, a risk-based approach is recommended.

PHARMACEUTICALS AND MEDICAL DEVICES AGENCY (PMDA)

In the PMDA`s Technical Conformance Guide it states that

"Before submitting electronic study data, the applicant should perform a validation."

and

"The credibility of the numerical results of the analysis depends on the quality and validity of the methods and software.".

PMDA requires pre-submission validation of datasets and outputs as well as retention of metadata and review materials such as define.xml, **validation proportionate to risk** and an **overall verification** that the methods and software used are **valid**. However, independent DP is not explicitly mentioned.

Considering the absence of the term "double programming" in guidance documents from all the above sources, one can ask why independent DP actually became a commonly used best-practice method for validation. A simple yet practical answer to this question is: **because they could**. Meeting the expectations of authorities was possible by an approach that was (and is) resource extensive, but easy to implement technically, to document, to trace and to understand. After relying on it for decades now, technology has advanced, so it makes sense to think of ways of meeting the same expectations but following different approaches that may not be possible from a technical perspective 20 years ago. Contemporary alternatives will be discussed in a later section.

PITFALLS AND PREREQUESITES

Two individuals work independently on the same dataset, variable or output, and then compare their results. If they agree completely, the results must be correct, right? Unfortunately, this is not always the case, and things can go horribly wrong. The specifications must be very clear and precise. If two people misinterpret the specifications in the same way, they might end up with the same incorrect result. On top of being resource-intensive, this is an overall bad deal. For more complex tasks, where DP is often used for critical and complex derivations, experienced personnel should be involved. Having two junior-level programmers work on the same problem may not produce the desired result; therefore, at least one senior programmer should be assigned to prevent junior-level mistakes. This also provides a valuable learning experience for inexperienced staff. Killing two birds with one stone!

Another issue I came across in the past is the danger of reusing validation programs for several studies. Our daily work can become stressful due to tight deadlines, particularly when multiple projects overlap. It is only natural to take the path of least resistance. Why not reuse that validation program you created for the double programming of ADTTE last year for the sister study in compound X? The same endpoints are required, and at first glance, only minor updates seem necessary. This might work, but caution is needed. Even under time pressure, it is important to take the time to question the previously used code. New data may introduce cases that were not covered by the logic used in the sister study, or the logic may have been flawed without being exposed by the data. Even worse, both 'independent' programmers could start their derivations from the same ADTTE derivation program without realizing, increasing the chances of ending up with the same (incorrect) results. Finally, I would like to emphasize the importance of comprehensive documentation. Maintaining detailed records of the programming process and any discrepancies found ensures traceability and supports regulatory audits.

ALTERNATIVES AND FUTURE TRENDS

In previous chapters, we examined QC practices in other industries, the requirements of pharmaceutical regulators, the evolution of DP into the gold standard it is today, and the key considerations for implementing DP correctly. In the past, DP was a practical solution for meeting regulatory expectations. However, with the technology we have today, other approaches should be considered.

CODE TRACEABILITY

Understanding where your data comes from and how variable X used for a specific table or graphic was derived, is key to showing confidence in the results that statistical programmers create. Although DP is one way of achieving this, since regulators do not require its use, other methods can be employed. Our code interacting with data is the source of truth. Starting with the code as the basis for the analysis performed in a clinical trial, code traceability provides the foundation for a validation workflow that offers many advantages over DP when it comes to delivering high quality in less time. It captures every variable, dataset, format, codelist, result, and transformation in a study.

One major advantage would be the ability to respond quickly to enquiries from internal colleagues or regulatory agencies. Rather than searching through many different programs to find out exactly how and where a specific variable was derived, code traceability enables us to provide all relevant dependencies for the derivation and the derivation itself much more quickly than a manual process would allow.

Coordinating resources can be challenging when dealing with multiple projects, tight deadlines and limited staff. A new validation process based on code traceability could enable much faster onboarding and increased speed. Knowledge graphs visualising the flow from raw data to SDTM, ADaM and TLF (programs) are well suited to facilitating navigation through clinical trial analysis.

Such an end-to-end traceability makes reproducibility easy, enabling transparency, trust and confidence in submissions. It does not stop at code traceability though. Linking the actual data from raw to SDTM and ADaM and tracing particular variables, allows for detection of flaws in the process. Mapping and conversion errors may lead to missing values or extreme outliers, which can be identified by tools based on code traceability. The tool of our choice is the Verisian platform. On the one hand, incorporated checks against CDISC SDTM and ADaM implementation guides help making sure the data was mapped and derived correctly or in other words – the data structure is in accordance with industry standards. On the other hand, validation summaries help identifying issues with the data content.

Another useful feature is checking the consistency of variables between SDTM and ADaM. Variables that are supposed to be the same in both standard models are auto validated, if they are actually the same in your studies' data. That alone reduces the time needed to validate mapped data. If a variable is supposed to be the same, but it is not, a warning is fired signaling that additional checks are necessary.

A PEEK INTO THE CRYSTALL BALL

In the future, statistical programmers are likely to be spending less time on writing code due to progress in achieving end-to-end automation. The focus could be more on understanding the underlying data, domains and concepts. Basically, they are going to be the ones telling the story of the data. Language agnostic programming will most probably force especially young statistical programmers to diversify their programming skill set, enabling them to choose a programming language suited best for a particular task. Instead of static outputs, dynamic and/or demand-based reporting might increase in importance.

The latest developments, including agentic AI writing code, sound promising. Nevertheless, validating and overseeing agentic AI-produced code is likely to remain the responsibility of humans. For now, it makes sense to be sceptical about AI-produced code, since AI can only use what it has been trained with. A simple yet clear example of this is asking the AI you 'trust' to show you an image of a clock displaying a time other than 10:10 am/pm. Since the internet is full of clocks showing 10:10 (as this kind of resembles a smiley face, increasing the likelihood of it being purchased by customers), it is difficult for the AI to master this task. Despite all the hype, it still has limitations that one must be aware of.

DOUBLE PROGRAMMING IS NOT DEAD, BUT…

Although regulators do not explicitly request that sponsors or CROs use DP, they do require a proper validation process to ensure the right quality. Rethinking the current status quo using the technology we already have or that is evolving rapidly leads to new QC methods that maintain or even increase confidence in results and outputs. Although DP may still be used for a while, especially in high-complexity cases, its usage may drop significantly. Combining code traceability with AI could instil confidence in AI-generated code. Even if the accuracy of such a construct is not perfect, it could improve iteratively. Having touched on the pitfalls of DP, and since probably no one knows its failure rate, the comparison might not be fair in the first place. Ultimately, it is important to understand the role of a statistical programmer. We do not ensure the quality of the data, as this is the responsibility of a data manager. Nor do we provide a full interpretation of statistical analyses, as this is the responsibility of statisticians. We bridge the gap from raw data to SDTM, ADaM and finally outputs in the highest quality we can and the way of achieving this is subject to change.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Author Name: Vitali Gering
Company: Chrestos GmbH
Address: Girardetstraße 1-5, 45131 Essen, Germany
Email: vitali.gering@chrestos.de
Website: https://chrestos.de/en/