# Test Data Factory – Phase 2

**The idea of this document is to allow offline discussion and to capture the thought process and arguments.**

[Test Data Factory – Phase 2 1](#_Toc532661786)

[Background 1](#_Toc532661787)

[Open questions 1](#_Toc532661788)

[Thoughts and details 2](#_Toc532661789)

[On Deliverables 2](#_Toc532661790)

[On Scope 3](#_Toc532661791)

## Background

TDF – Phase 1 was focused on updating the existing CDISC pilot datasets (SDTM and ADaM) to be in compliance with newer versions of the standards. The objectives were to provide some tangible results quickly and to gain experience with processes and tasks around creating test datasets.

Phase 2 is supposed to allow users to generate datasets that didn’t previously exist. The vision for this phase is to provide a “system” that supports the following steps:

* Collect user input about what datasets are required and about the required content and characteristics of the datasets.
* Create the specified datasets and make them available to the user.

## Open questions

At the starting point of Phase 2, the team needs to make some decisions – including setting priorities for required features. As a starting point, I see the following areas for decision making. Note that in the next section I am describing my thoughts or answers there is space for comments and discussion.

* Expected or planned deliverables: The TDF Phase 2 vision calls for some means to create test datasets based on user input. In a perfect world, this would probably be a “program” (commercial or open source) that allows the user to input dataset specification and that would then create the datasets accordingly. However, lower hanging fruits can be envisioned: For example, a viable concept for specifying test datasets and example scripts that generate datasets could be enough to enable users to implement their own scripts by adopting and modifying the sample scripts.
* Scope: This concerns which datasets should the user be able to create and which characteristics the datasets should display. The scope decisions include determination of the standard (like SEND, SDTM, ADaM), which SEND or SDTM domains or ADaM datasets, and questions such as “dirty or clean data” or whether a define.xml should be created or not.
* Process related issues: This concerns questions how the project will assure that the deliverables are actually usable: Documentation of deliverables, quality control for scripts, validation of generated test datasets against standard specification.

## Thoughts and details

First, I think that the process-related issues are very much driven by decisions on scope and deliverables. Therefore, they are not included in the detailed discussion in this section.

I don’t think that we need to make all decisions now at the beginning of the TDF Phase 2 project. Just the opposite, at this point, I think we need to focus on a scope that is reasonable and most likely provides value to other groups and decide on deliverables that are achievable. In other words, I believe we should strive for a way to follow an “agile approach”. What I mean is that that whatever the starting point will be (which will be something small), create a “simple and limited deliverable” that provides some value to a user – even it the value is rather small. Then in each “iteration” we should add to the existing solution.

### On Deliverables

From my point of view this is really a tricky question.

* The spectrum of possible deliverables ranges from a “ready-to-use” solution (like a commercial or open source program or SaaS product) to a concept or white paper about how to generate test data sets. Somewhere in this range I can see the team to provide examples of scripts that users can adopt and modify to meet their own needs.   
  Let’s look closer at three alternatives
* Concept paper:
* Description: A concept paper could be very generic (and from my point of view not very useful) to very specific with details about which domain to create, which user input would be required, which characteristics of the test data would be supported, how to represent (and maybe even how to capture) user input, and more details. I would envision that the team would start on a generic level and as the work progresses, decide how specific the paper is going to become.
* Peter: I think that much of the work for a concept paper needs to be done regardless of the final deliverable. Therefore, it seems that the team can start working on a concept without making a final call about the delivery at the end of the project.
* Dante: I would start with an "Analysis" deliverable. What are the most basic, common e-submission datasets? Start with the top 5 or 6. What are the most basic types of variables / data collected? Start with the top 8 or 10 data types. Work on simulating these most basic clinical trial data types. Model these data types to create a framework / syntax for users to specify their database (tables, vars, value characteristics). A YAML type file or JSON may work for this.
* Jessica: To me, I think a concept paper would only make sense alongside actual examples. Also, in order to come up with the details for the concept, I think we would need to try it out ourselves to be able to really see what type of issues would come up. So, I agree with you that the concept paper will need to be done regardless as we figure out how to create actual scripts or a ready-to-use solution. Basically, we would need some kind of requirements doc or spec to guide what we are trying to program
* Sample scripts:
* Description: In some sense, we would follow the idea of the “Repository Content and Delivery” (RCD) project: Start with a ‘white paper’ and implement scripts that are shared across the industry. These scripts would be published in the script repository (the one that Hanming describes in the recently completed white paper.
* Peter: As attractive as this sounds, I have my doubts that this is the most promising path. I agree that the idea of sample scripts to prove the concept is a very good one. However, the uptake of the RCD scripts has been low – close to zero. I believe there are reasons (the ‘not invented here’ issue and the perception that the effort for integrating these scripts into some existing solution might be high (true or not doesn’t matter) – just to name two issues that come to mind.) So, I believe that as useful as sample scripts are to get started and to get some code running – think of demos and presentations, as well – we should not stop there but move towards a ‘ready-to-use’ solution.
* Jessica: In order to create a ready-to-use solution, wouldn’t we still need to first create some scripts/programs that would then need to be pieced together for a complete solution?
* “Ready-to-use” solution
* Description: details need to be determined (such as desktop or SaaS solution, how much user interface, how much custiomization, etc.) Essentially, we would need to run a “product development” operation including product management, development, testing, and delivery.
* Peter: I am afraid that the effort for testing and releasing an open source or even worse a commercial product is rather high and goes very likely way beyond the scope and means of the TDF project team. It might also not even be what PhUSE wants.

One remark though: As you know, my company has developed a platform called Vicos, that would allow to publish scripts as SaaS product with very little additional effort once the scripts work. This applies without any license restrictions to R and Python scripts. I can imagine a scenario, where the TDF team would make steps towards a product (concept, specification sample scripts, etc.) but ‘productization’ would be done my VCA-Plus resources. In this scenario, PhUSE and VCA-Plus need to discuss and clarify issues like IP rights, commercialization, etc – in short, all things “Product Management”.

* Dante: I would support a fully functional application, but with limited initial capabilities. Get something very basic working and available to all, and people will see the beauty of extending :). This would be supported by an agile approach to our process.

Initial basic application should not be too hard to validate and could be an interesting project. I can imagine simply visualizing or summarizing variable contents – which should match basic input (categorical vars, real with some distribution). Should not be too hard.

* Jessica: I’m not sure if a full comprehensive product is needed, but I think it would be outside the scope of PhUSE.

### On Scope

From the discussions that I had over the last weeks and months I see the following alternatives:

1. From the white paper project, I heard several times that they have the need for test data frequently. Most recently, the Lab data were mentioned several times, from earlier I know that some people were looking for AE data. Details to be clarified and Mary would be willing to help us.
2. From the work on the ADaM-NCA team, I know that there are no good pharmacokinetic (“PK”) test data available. PK data are in some way different from other data because they represent concentration-time profiles and for analysis (even the simplest NCA) the concentration-time profile data need to be merged with exposition and vital sign data. In some cases, even more domains come into play. So, a broadly useful concept for creating “observed” PK datasets could quickly become complex. On the other hand, I believe that these concepts for PK data will get quite a bit of attention.
3. When I looked at available scripts in the working group, I realized that we have all the scripts provided by the FDA from their Jumpstart services, but there are no datasets for these scripts. The required datasets are DM, EX, AE, DS, and LB – so it is a limited set of datasets, includes the LB domain (see option 1). So, a deliverable that generates these datasets could be a very useful and natural result of the TDF Phase 2 project.
4. Finally, we know that the non-clinical working group has started an effort to create SEDN datasets. So, from my point of view it wouldn’t make sense to focus on SEND (though from my understanding the “Jumpstart datasets”, see option 3) would be SEND datasets).

| Name | Comments |
| --- | --- |
| Peter | Contemplating the different aspects, my conclusion would be to go for SDTM datasets for the domains that are used as input in the FDA provided scripts (option 3).  My reasoning:   * I believe these datasets have a level of complexity that we can manage. * A solution that can create these 5 domain datasets will be broad enough to be useful for users. I would also expect that this kind of solution can be extended to cover additional domains with very reasonable effort. * Targeting SDTM and not SEND and sharing experience with the SEND effort (option 4) will help to avoid duplication of work or even competition between teams. * Targeting SDTM and not ADaM seems to be a better starting point, because companies might have solutions for creating ADaM datasets from existing SDTM domains. Maybe we can leverage these kinds of solutions. |
| Nancy |  |
| Jessica | I think SDTM datasets would make more sense over ADaM as ADaM datasets have much more study specific requirements to take into account. I would wonder if the datasets we had already updated could be used with the FDA scripts. If not, what about them would need to be updated to be useful with the FDA scripts?  Even if people are not really using the RCD scripts, I still think it would be useful to have one example script for one dataset that someone could adjust. Even though I remember seeing your example for creating DM, without knowing the program used behind it, not sure how easily adaptable it is. I think since people specifically required LB and AE test data, it would be good to know those requirements and see if we could create something specific for that known need. |
| Dante | About the “white Paper approach”: I've heard this for so long! Yet people don't know where to begin and always revert to cobbling together some (poor) historical data. The TDF is a better approach! |
| Cynthia |  |