Next Generation Technologies eBook Series

Part I

Next-Generation IWR/IVRS

BIOCLINICA® Trial Blazers

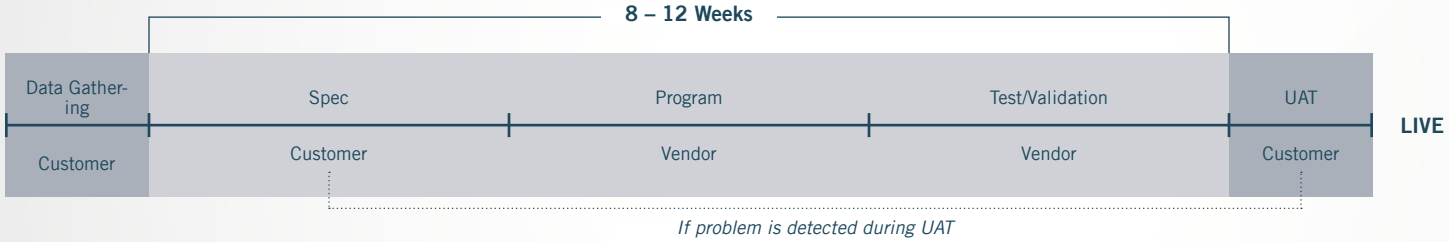
The IWR/IVRS/IRT class of clinical trial support systems is centered around patient randomization and clinical supply management and dispensing. Such systems commonly include other features such as screening, break blind and patient status management, but the greatest value and primary justification in using them tends to come from the dramatic savings in clinical supply relative to not using such systems. It is simply no longer feasible to run a large, blinded multi-center trial without IVRS.

This eBook uses concepts which will be key to understand. Some of them are technical in nature, but all result in dramatic real-world impacts and therefore should be laid out clearly. Please click on the highlighted words throughout the paper to link to the definitions in Glossary.

Please note: this eBook will sometimes use the three terms—IWR, IVRS, IRT—interchangeably to describe a type of system in a general sense, unless a specific interface is being discussed.

Legacy IVRS — The Reality

The reality of the IVRS market today is this: sponsors design a trial, select a vendor, and then set out to write a large and complex specification detailing how the trial should be conducted in IVRS. It is not uncommon for these specifications to be hundreds of pages long, and yet they must be authored in the abstract (before the system exists) by sponsor staff who are not necessarily system designers and may not even have a passion for the work of writing specifications. Using the specification, a programming team then builds a system, which then becomes the IVRS for the study in question. This system is tested and validated, and if all goes well it can be "turned on" in 2–3 months.



All does not always go well, however. It is difficult for a vendor to get every line of a 250-page IVRS specification right, and so errors in specification or even in programming may occur. Even more commonly, the trial design may change, even slightly, but after the specification has been written and programming has begun. Regardless of the cause, changes to the IVRS design have the potential to start the whole process over again, with the added penalty of change order and associated cost from the vendor. In the worst case, the 3-month clock starts ticking all over again.

All of this can be a slow, painful and expensive process for sponsors to endure. At the same time, the chances are good that the study start up date may be very important to the sponsor (considering drug patent life and all) and yet, this is also put at risk. On top of that, the rather high demand for clinical trials in IVRS around the world combined with the slow, resource-intensive process described above has put pressure on IVRS vendor resources to such an extent that it is not unheard of for existing vendors to literally turn away business (or at least not to pursue it).

Sponsors endure this reality because the case for clinical supply savings is utterly compelling. In many cases, there is just no other way to support a large trial besides the use of IVRS. The costs, workload, stress and even delays to the trial itself are all accepted because in the end there is really no other choice.

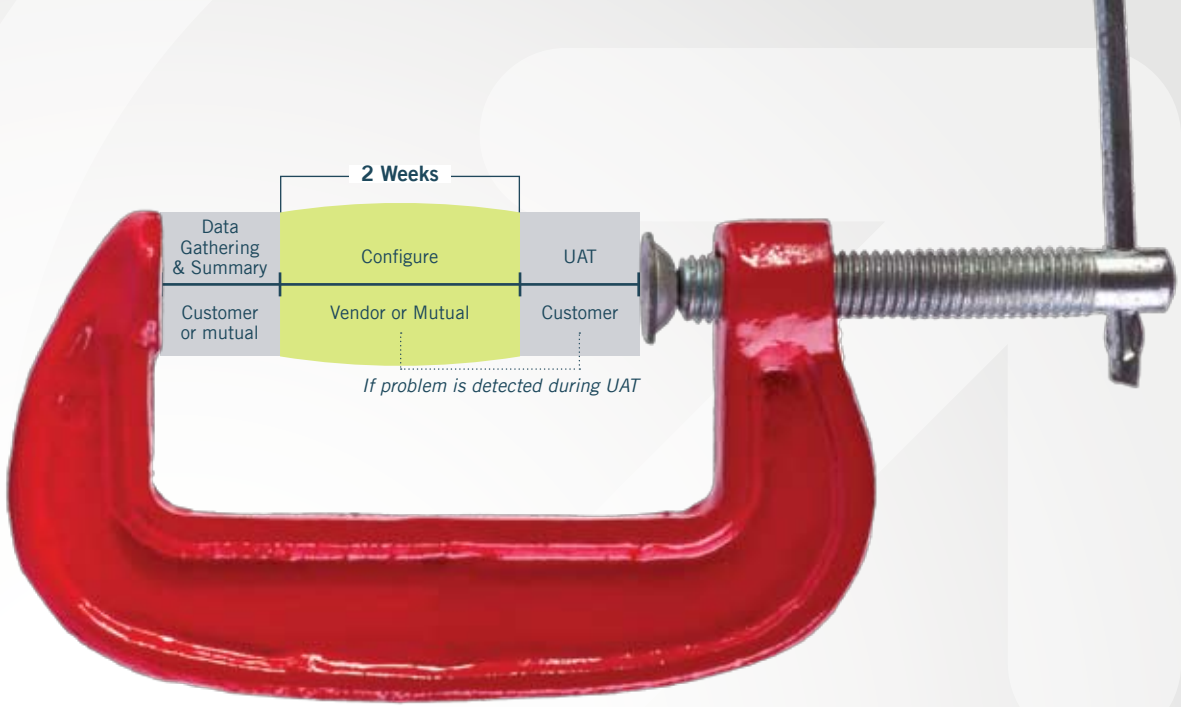
Forward thinking vendors have been working on solutions to this problem, not only in order to leapfrog the competition but in many cases simply to be able to take on the amount of business that is available to be had—if only the vendor could do all the work. Some vendors have built partially configurable systems. Some have easily reusable code libraries, and others have added neat graphical tools with which to speed the process of building an IVRS. In some cases, the "industry standard" 12-week lead time can be cut down to 8 weeks, with some vendors even claiming to be able to bring up an IVRS from start to finish within 6 or sometimes even 4 weeks (although these claims are difficult to verify and study complexity may become an issue). In any case, it has been a focus for the vendor community to try to improve "the situation" with respect to IVRS development resource intensiveness, cost and ultimately speed.

Next-Generation IWR/IVRS

"Next-generation" IWR, IVRS or IRT is a tag that is being used more and more frequently by vendors coming into the market with new products. In some cases, this refers to advances that were mentioned in the paragraph above (e.g. partially parameter driven); in others it refers to new technology that a company truly feels is the next big thing; and in still others, it merely refers to the latest release of vendor software, and/or a marketing campaign.

While everyone is entitled to their own opinion about what truly constitutes "next-generation" software, for our purposes, we will define it somewhat narrowly: A Next-Generation IWR/IVRS must have at least the following three characteristics:

- It must be parameter-driven: meaning that studies must be configurable by business experts rather than programmed by programmers.
- It must be robust: meaning that a majority of modern study designs, especially around randomization and dispensing, can be supported without any customization
- It must contain some facility: to allow fast and minimal customization when necessary, such as custom code plugins.



The reason that we have defined next-generation IVRS so precisely is because these features, above all others, will fundamentally transform the global IVRS market—largely solving the problems listed above in the current state and replacing them with fast and flexible timelines and empowered sponsors. These three characteristics will enable a sea of change in IVRS above all others.

In addition to the primary characteristics, other features which might be associated with next-generation IVRS include:

- Modern web-oriented UI
- No call flow (context aware presentation)
- Branching Q&A capabilities
- Native drug pooling support
- Powerful and transparent site resupply algorithms
- Powerful reporting
- Study and supply administration by sponsor

- IWR and IVR from the same logic (no separate setup)
- Text to speech / automatic IVR

These features are all powerful—and sometimes transformational—in their own rights. But without the big three aforementioned features, it would be difficult to transcend the problems of the current market.

Limitations & Challenges

The limitations of next-generation IWR/IVRS relative to legacy systems are primarily centered around legacy systems' abilities to literally program whatever a sponsor can imagine, exactly to spec. In some cases these can be critical features that are required in order to run the trial. In other cases they may be matters of convenience or strong preference, or they may simply be artifacts of sponsors' comfort with the old paradigm. In any case, anything that cannot be natively supported by a next-gen IWR/IVRS must be addressed in one way or another, or the sponsor will be forced to choose a legacy offering.

In the end, most customization concerns can be mitigated by custom pluggable code. Also, any significant or somewhat common feature missing from a next-generation system should be added as a main system feature for a subsequent release of the product. So, while the challenge of customization will always be present, it is strongly mitigated by systems, which are designed to offer very robust protocol support, and ultimately in the majority of odd cases, by custom code plugins.

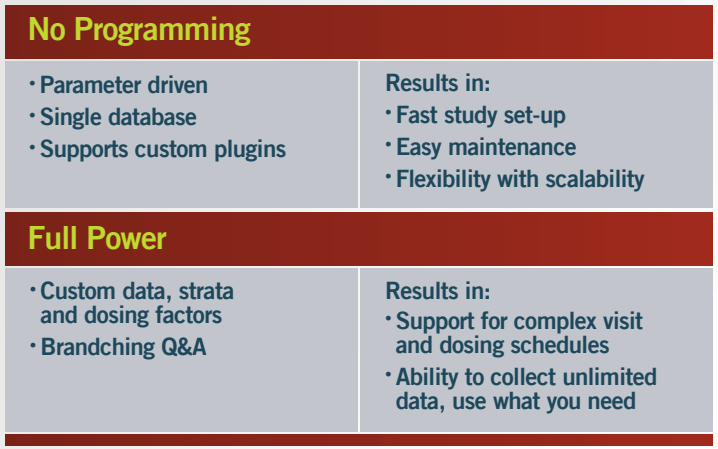
Modern IWR	Legacy IVRS
Fast, easy set-up	Programming can take months
PM/Business participation	Business users are isolated from system programmers
UAT by same group	Software testers, another disconnected group
De-pressurized resources	Inadequate supply of IVR resources industry wide

Another challenge associated with next-generation IWR/IVRS worth mentioning is that the systems themselves are very difficult to design, build and test, especially relative to legacy IVRS.

Imagine the difficulty of building and testing all of the possible functions of a single study design in IVRS. Now, imagine building all possible study designs in one product, and testing all meaningful (i.e. cross-impacting) features independently and as a group. Designers and builders of a truly robust next-generation IWR/IVRS must plan to support most or all varieties of modern multi-center study designs—and this in an era where study designs continue to become more esoteric! This is perhaps the main reason why next-generation systems remain rare: given the level of difficulty involved, vendors have not been successful in bringing such systems to life. Perhaps this contributes to some degree of market skepticism that such a solution is even possible.

Benefits of Next-Generation IWR/IVRS

The practical impact of next-generation IWR/IVRS as defined in this eBook is dramatic: instead of specification writing, programming, testing, and repeat, the process can be simplified to a few quick build-and-review iterations based on the protocol document and some other supporting information. In the fastest case, it should be possible for sponsor and vendor to work together via WebEx and configure a draft IWR/IVRS for the study—from scratch—in just a few hours. The speed with which this can be done enables many rapid iterations to take place, as the system is built, honed and demonstrated to other sponsor personnel. In practice, however, only a few iterations are usually needed. The ability to do rapid changes and iterations completely mitigates the perennial issue of late breaking protocol changes.



In summary: next-generation IWR/IVRS takes a multi-group, multi-step process that is usually 8–12 weeks—perhaps 6 weeks at best—and combines it into a business-experts-only, single-group, fast-iteration process that should really take no longer than 2 weeks. The lion's share of the effort to set up a study rarely takes more than a day or two of cumulative effort. So even if a study takes a few weeks to set up (due to lack of information from the sponsor, perhaps), the same vendor staff can be simultaneously setting up multiple other studies for other sponsors due to the ease and speed of doing so in a next-gen IWR. And at the end of the process, the really neat IWRs can spit out a configuration specification as a printed document for signature. Just to put a cherry on top, the dreaded IVRS specification has been turned from an input into an output.

This value proposition is worth repeating: next-generation IWR/IVRS *solve* the painful problems of current state IVRS, while enabling lightning fast, flexible setup of even the most complex clinical trials. Late breaking changes are no problem. Urgent study starts are no problem. On average, a sponsor could reasonably expect IVRS setup time for a clinical trial to decrease from 12 weeks to just 2 weeks.

If this is all true, then it would seem that in the long run—or perhaps even in the "medium run"—one logical conclusion might be that there will be little reason for legacy IVRS to continue to exist.

Some might question whether a picture this good could possibly be true. Skeptics might note, however, that vendors equipped with true next-generation IWR/IVRS regularly show up for sponsor IVRS demonstrations with studies already built (based on protocol documents provided only a few days earlier) and/or undertake to build or modify the study during the demonstration itself. If, at the end of the demonstration, the IWR and IVRS are set up to accurately run most or all of the study in question, then the point is already proven. The IVRS has been built before the vendor selection is even made. Rather than to start writing a specification to be programmed, tested and validated over a 12-week cycle, the sponsor can choose to simply press "print specification," and sign, and proceed to user acceptance testing.

Read Part II: The Truth About Forecasting Clinical Trial Supplies.

Glossary

Clinical Supply Forecasting / Simulation

Clinical trials involve a variety of unknowns: where in the world will subjects enroll? At what rate? What treatment group will they randomize to? What titration path will they take within a treatment group? What about dropout rates? What progress through unusual visit schedules will occur? What about expiring drug?

Despite these unknowns, it is always necessary to create a clinical supply plan in order to properly supply, and thus run, a clinical trial.

In the simplest case, it is usually possible to calculate baseline drug consumption for a clinical trial by multiplying the number of intended patients in each group by the number and types of intended doses for that group. However, this ignores (among other things) the complexities of how much "overage" to place at sites and depots in advance, in order for the trial to run. Because drug cannot be delivered instantly to sites where patients enroll, it usually has to be pre-positioned. This fact, and many other complexities of modern trials, may drive the necessary or required overage for a trial — that is, the amount drug that you will definitely not use in the trial but you need anyway.

There are approximately four different methods that people use to calculate overage for a given clinical trial. In order of effectiveness, they are:

- Basic math on paper or in a simple web application (actually does not produce overage)
- Intuition
- Excel®
- Simulation

Since the middle of the last decade, simulation has emerged as the leading and most accurate method of forecasting clinical supply requirements, although all four methods are still in use depending on the sponsor company, complexity of the trial in question, and tools at hand for the person creating the supply plan.

IWR/IVRS/IRT

In the world of clinical trials, "IVRS" is one of several fairly imperfect names for a very specific type of technology. This phone-based technology (literally: "Interactive Voice Response System") was originally built to support patient randomization over the phone, but quickly evolved to include management of clinical supply from depots to sites to dispensing to subjects. Other common functions of IVRS include screening / registration (i.e. generation of the patient ID), changing a subject or kit status (e.g. to screen fail for a subject or lost or damaged for a kit), emergency break blind, and advanced dispensing (e.g. titration, investigator choice). All along, clinical supply algorithms have been improving and becoming more complex, with most serious IVRS now including complex predictive algorithms for just-in-time delivery of future patient dosing. In summary IVRS — originally conceived of as randomization systems — have evolved into advanced clinical supply management and dispensing systems.

Accordingly, the primary driver for companies to use IVRS is the huge amount of savings in drug supply that it provides relative to non-IVRS trials, and therefore most IVRS companies spend a good deal of effort attempting to do this well.

As IVRS became more advanced, several began to migrate to the web, either as an option included with IVRS or as a separately built system. For lack of a better term, these systems were dubbed "IWR" for "Interactive Web Response." Yet another common acronym is "IRT" which means "Interactive Response Technology" or sometimes "Interactive Randomization Technology." This eBook commonly uses the terms IVRS and IWR/IVRS, either of which should be taken to mean all systems that fall into the category regardless of their presentation on web, phone of both.

Site Resupply Algorithms

Clinical supply savings is a common driver for companies to adopt IVRS for multi-center trials. These savings exist because the non-IVRS method of providing clinical supply to patients it to label entire courses of medication with the patient randomization number in advance, and then ship all of that drug to sites in blocks (usually four-patient blocks at a minimum) — in advance of seeing any patients at all. This method seems incredibly wasteful compared to the IVRS paradigm of using patient interchangeable supply which is shipped to the sites just in time for patients to use it.

The IVRS vs. non-IVRS debate being long concluded, the next challenge was for IVRS to be able to supply trials most efficiently. For example, if the IVRS knows about all subjects in the world, and knows what treatment groups that they are in and what their upcoming visit schedule will require for dispensing, then it should be possible to pre-position patient kits (dispensing units) for each patient at each site just in time for the patient to use it. Add to this a small buffer for breakage or loss, and in theory it is possible to have a perfectly efficient site supply/resupply algorithm. This will perform considerably better than a simple buffer or floor/ceiling algorithm for site supply.

However, most trials have at least some element of unpredictability. Many are centrally or dynamically randomized, which means that the treatment required by a patient is not known until moments before it is required (at the randomization event). Others have elements of titration, variable dosing, variable visit schedules and so on.

Most advanced IVRS site resupply algorithms deal with this uncertainty by adding buffers or floor/ceiling values to the predictive calculations for known future demand at sites. IVRS also dabble in other advanced techniques including predict-all-options (or worst case), probability based resupply and many others.

All of these methods of figuring out how much to send to sites in advance are controlled by parameters, from the very simple (buffer, floor/ceiling) to the very complex (look-ahead time windows, trigger windows, patient counting rules, etc.).

Without going into any further detail, let us stipulate the following:

- IVRS resupply algorithms are usually complex, and controlled by complex sets of parameters
- All of this tends to be poorly understood by customers and even by many vendor personnel
- Yet, this is where clinical supply efficiency or inefficiency will be realized.

The term “IVRS resupply algorithm” may be drab and off-putting, but in most cases it represents the beating heart of the IVRS. This is where the value comes from.

Parameter Driven Systems

A “Parameter Driven” system is one that can be made to do all of its work via an administrator or other user setting the system up using purely features — switches, knobs, dropdowns, etc. — within the system itself.

By contrast, a traditional IVRS system needs to have a specification written according to the protocol design, and that specification then needs to be handed to developers who then program the system (and then testers test the system, and so forth). Because each clinical trial is different, and because fully featured parameter driven systems are very difficult to create, most IVRS systems are traditional programmed systems.

Some more modern IVRS systems rely upon validated libraries of functions and therefore are only partially programmed — "partially parameter driven."

Next-generation IWR/IVRS however are fully parameter driven — that is, all aspects of the system may be configured by a non-programmer resource, using features of the system itself in order to complete the setup of a clinical trial as an example.

Simulators face the same challenge: each protocol that needs to be modeled may be very complex, or, at least very different from the last protocol that was modeled. Therefore clinical trial simulators also fall into one of two categories: parameter driven and programmed.

The primary limitation of parameter driven systems is that designers cannot think of everything in advance, and in any case it is not practical to build configuration capability over every possible trial design, now and in the future. The difficulties of "parameterizing everything" has caused some parameter driven IWR/IVRS to remain relatively low profile in the market, running only a handful of tightly conforming studies which happen to fit within the parameters at hand. However, even the best designed parameter driven system faces this limitation, and in order to be successful they require a solution to it.

Pluggable Code

Pluggable custom code is one modern solution to the limitations of parameter driven systems. Without getting too technical, code plugins allow small pieces of programmed functionality to be inserted into a system to handle very specific aspects of specific studies, which could not be otherwise handled.

One-off items such as sending a specific shipment based on an arbitrary trigger, or performing a logical operation on data which derives some other operation (such as a dosing or stratification decision) are examples of the uses of custom code plugins. Other uses would be for any feature which is missing from the IVRS, for example a novel type of trial adaptation, or for that matter, cold chain temp tales.

Since traditional IVRS are always programmed, they always have the luxury of programming anything that a study design may call for. But parameter driven IWR/IVRS do not have this luxury — except to the extent that they implement custom code plugins or some similar technology.

The important point with plugins, however, is that the amount of specification, programming and testing is minimized as much as possible — isolated to very small bits of functionality which are "don't-have but must-have" in order to run a study.