

Chapter 14 / Capítulo 14

Primum non nocere: Software engineering
for medicine
ISBN: 978-9915-9876-8-2
DOI: 10.62486/978-9915-9876-8-2.Ch14
Pages: 225-245



© 2026 The authors. This is an open access
book chapter distributed under the terms of
the Creative Commons Attribution (CC BY)
4.0 License.

Chapter 14. Study Design for Medical Software

Annier Jesús Fajardo Quesada  

Eduardo Antonio Hernández González 

René Herrero Pacheco 

Rolando Javier Álvarez Pérez 

1 Granma University of Medical Sciences. Granma, Cuba.

2 University of Medical Sciences of Pinar del Río. Pinar del Río, Cuba.

“A clinical trial for a drug and a study for medical software are like open surgery and a laparoscopic procedure: they pursue the same goals, but the methods, risks, and ways of evaluating them are radically different. Confusing them is not only a methodological error but *also* a source of delays, cost overruns, and, worst of all, insufficient evidence to make informed decisions.”

Considering the following case, before implementing it, we want to demonstrate that it actually improves patient outcomes. We designed a randomized clinical trial, similar to those used for drugs, where half of the patients received care using the algorithm and the other half did not. However, problems soon arise: Doctors cannot be blinded to the treatment because they know whether they are using the algorithm or not. Patients may be at risk of bias because the same physician treats patients from both groups. The outcome measure, mortality, may be influenced by many factors in addition to the algorithm. The required sample size is enormous because the algorithm is only an aid and not a direct treatment. This scenario illustrates the difficulties of applying traditional evaluation methods to digital interventions and the need for specific approaches for medical software.

The evaluation of medical software poses unique methodological challenges that distinguish it from drug evaluation. Software is not ingested, has no pharmacological effects, and is not metabolized. Its impact is always indirect and mediated by the decisions of the clinicians who use it. The same algorithm can be highly effective in the hands of some doctors and useless in the hands of others, depending on how they integrate it into their practice. It can improve outcomes in some contexts and worsen them in others, depending on the population, resources, and local culture. Evaluating medical software involves evaluating not only a technology but also a complex sociotechnical system.

In this chapter, we explore the fundamentals of the study design for medical software. We will begin with the fundamental differences between clinical trials of drugs and studies of digital interventions. We will continue with prospective and retrospective designs and their advantages and limitations in this context. We address the practical aspect of how to speak with department heads to secure their collaboration, an often overlooked but essential aspect. Finally, we discuss the preparation of applications for ethics committees, a process that can be particularly complex when the intervention is software. This is not an exhaustive guide to research methodology but rather a practical reflection for those who need to demonstrate that their software actually serves a purpose.

14.1 DIFFERENCES FROM DRUG CLINICAL TRIALS

Modern medicine has been built on a methodological pillar that seems unshakable: randomized clinical trials. For decades, this design has proven its worth distinguishing between an effective drug and an inert drug and between a genuine cure and a placebo effect. The logic is seductive in its simplicity: we randomize, compare results, and randomize the rest of the work, balancing known and unknown factors between the groups. However, when the intervention comes not in a vial but on a screen, when it is not a molecule but an algorithm, that methodological elegance collides with a much more complex reality. Medical software does not behave like a drug and does not act like a drug; therefore, it cannot be evaluated exactly like a drug. This assertion is the starting point for developing more appropriate, more honest methodologies.^(1,2)

After all, regulators are familiar with them, journals publish them enthusiastically, and scientific prestige seems guaranteed. However, this approach can lead to studies that are brilliant in appearance yet deeply misleading in substance. A trial demonstrating that an algorithm “works” under controlled conditions can tell us very little about whether it will work in the operating room of a regional hospital three times in the morning, with an exhausted doctor and a computer system that crashes every ten minutes. External validity—the ability to generalize results beyond the study’s context—becomes the true methodological challenge. To address this issue, we must first understand precisely how digital intervention differs from chemical intervention.

14.1.1 THE HUMAN INTERMEDIARY: WHY THE DOCTOR IS PART OF THE MEDICINE

The first methodological obstacle is so obvious that it is often overlooked: a drug acts directly on the patient’s physiology, regardless of who prescribes it. The active ingredient in metoprolol reduces heart rate whether it is prescribed by an expert cardiologist or administered by a resident

on their first shift. Medical software, on the other hand, is completely dependent on the human intermediary. A decision-support algorithm does not diagnose the patient; it suggests to the physician that they consider a certain diagnosis. It does not prescribe treatment; it presents options that the clinician can accept, modify, or ignore entirely. This human mediation introduces enormous variability such that traditional trials are not designed to capture or analyze adequately.

Let us imagine a hypothetical study of an early sepsis alert system in the emergency department. The algorithm analyzes data in real time and notifies the physician when it detects a suspicious pattern. In the intervention group, the early diagnosis rate improved dramatically. Does this mean the algorithm is effective? It depends. If we breakdown the data, we might discover that young, digitally native physicians act quickly on the alert and improve their outcomes. However, veteran physicians, who rely more on their clinical experience, may systematically ignore notifications that they consider to be false alarms. The average shows success, but the reality is a mix of success and failure that depends on user characteristics, not the software. Traditional clinical trials, with their focus on intention-to-treat, would lose this crucial heterogeneity in terms of a deceptively positive average.

This variability is not a software design flaw; it is its inherent nature. A drug is a physical entity that can enter the body and trigger predictable biochemical reactions. Software is a tool that amplifies, modifies, or distorts the clinical judgment of the user. Therefore, evaluating software without characterizing its users is similar to evaluating a scalpel without considering the surgeon's skill. The relevant question is not just "does the algorithm work?" but "with what kind of physicians, in what contexts, and with what additional training does it work best?" Classical experimental designs, which seek to eliminate variability between subjects, can be counterproductive here because we precisely need to study that variability to understand where and how to implement the tool.

Furthermore, the relationship between the clinician and the algorithm evolves over time. During the first few weeks of use, the clinician may be cautious and verify each recommendation. With familiarity, they may develop overconfidence, accepting suggestions without initial scrutiny. Alternatively, the opposite may occur: after several false alerts, the doctor may become desensitized and start ignoring the system, a phenomenon known as "alert fatigue." Drug trials assume that the effect is relatively constant over time, except for pharmacokinetics. Software trials must incorporate longitudinal analyses that capture this learning and adaptation curve, which may be more decisive than the technical characteristics of the algorithm itself.

The practical implication is that studies must collect detailed data on users' years of experience, specialties, attitudes toward technology, and workloads during on-call shifts when the system is used. This allows for subgroup analyses that reveal for whom the intervention actually works. A trial showing a null effect in the general population but a large effect in first-year residents remains valuable; it simply tells us that the software needs to be implemented selectively or accompanied by mentoring programs. Without this granularity, we run the risk of discarding useful tools or, worse still, of massively implementing systems that benefit only a minority.

14.1.2 THE BROKEN MIRROR: THE IMPOSSIBILITY OF BLINDING AND THE KNOWLEDGE EFFECT

In pharmacology, a blind trial is the methodological shield against expectation biases. If neither the patient nor the physician knows who is receiving the active treatment, differences in outcomes must be attributed to the drug and not to the psychology of expectations. With medical software, this shield inevitably cracks. It is impossible for a physician to be unaware of whether a decision-support algorithm is available or whether they are practicing with their usual resources. Knowledge of group assignment is complete, and that knowledge inevitably alters behavior in subtle but potentially decisive ways for the interpretation of results.

Consider the phenomenon of compensatory effort. Physicians assigned to the control group, aware that they do not have the "modern" tool at their disposal, may redouble their clinical attention to demonstrate that they do not need the technology. They may consult more literature, order more diagnostic tests, and devote more time to each patient. This extra effort, motivated by the knowledge of being in the placebo group, can level the differences between groups, making the software appear ineffective when in reality it is being masked by a predictable psychological reaction. Conversely, physicians in the intervention group may experience a novelty effect that temporarily improves their performance, regardless of the algorithm's actual quality.

There is also the risk of excessive dependence or paralyzing mistrust. Doctors who blindly trust the algorithm may stop exercising their own critical judgment, accepting inappropriate recommendations in atypical cases where the algorithmic pattern fails. Another physician, who is distrustful of the technology, may have the algorithm available but systematically ignore it, or worse still, may feel insecure and make more conservative decisions than usual. These behavioral responses are not methodological artifacts to be eliminated; they are an integral part of the “effect” of the intervention. However, they greatly complicate causal attribution: if outcomes improve, is it because of the algorithm or the greater caution it induces? If they worsen, is it due to software failures or the complacency it generates?

Methodological designs must incorporate strategies to detect and quantify these biases. One approach is to measure intermediate processes, not just final outcomes. If the algorithm recommends a certain treatment, how often does the physician follow it? Does this adherence vary depending on the complexity of the case? Another strategy is to include washout or follow-up periods after discontinuing the intervention to determine whether the effects persist (suggesting genuine learning) or disappear (suggesting a novelty effect). Qualitative interviews can also be used to explore physicians’ subjective experiences, revealing mechanisms invisible in quantitative data.

Transparency regarding these limitations is essential. An article reporting on an open-label software trial must explicitly discuss how knowledge of the assignment may have influenced the results. Readers, many of whom are clinicians who intuitively understand these dynamics, will value an honest study about its limitations more than one that pretends to have unattainable methodological purity. Science advances not by denying complexity but by acknowledging it and systematically incorporating it into the analysis.

14.1.3 INVISIBLE CONTAMINATION

One of the most insidious threats to the internal validity of software trials is cross-group contamination. In a pharmacological study, contamination is relatively rare and generally involves administration errors that can be detected and corrected. In a software trial, contamination is almost inevitable if the same professional treats patients from both groups, and it operates through cognitive mechanisms that are impossible to control through the protocol. When a physician uses an algorithm for some patients, the knowledge acquired cannot be selectively compartmentalized; it becomes integrated into their pattern of clinical reasoning and is applied, consciously or unconsciously, to all their patients.

The mechanism is simple yet powerful. Over weeks of use, the physician exposed to the algorithm learns to recognize patterns that have been previously overlooked. A pneumonia diagnosis support system can teach them to evaluate certain subtle radiological findings, to combine clinical data in novel ways, and to consider differential diagnoses they would typically rule out. This learning, which is precisely one of the desirable goals of the technology, becomes a methodological problem when the physician applies these new criteria to patients in the control group. The difference between groups is diluted not because the software is ineffective but because the control group has been “contaminated” by indirect exposure.

The magnitude of this effect depends on the nature of the software. Systems that provide factual information, such as drug databases, are less prone to contamination because the knowledge is not internalized in the same way. However, machine learning algorithms that identify complex patterns, especially in medical images or continuous monitoring data, modify clinicians’ perceptions in a lasting way. A radiologist who has used a lung nodule detection system for months will develop an “algorithmic vision” that will persist even when analyzing studies without computer assistance.

Strategies to minimize bias are costly but necessary. The most robust is cluster randomization, where entire hospitals or complete departments are assigned to one group or another. This eliminates cross-exposure at the individual level, although it introduces new complexities: hospitals differ in many ways beyond the intervention, and much larger samples are needed to maintain statistical power. Another approach is a phased design, where all centers eventually receive the intervention but at different randomized times. During the period prior to implementation at each center, they function as controls, and the comparison between centers in different phases allows us to estimate the effect.

It is also possible to measure contamination directly. If we suspect that clinicians are applying knowledge of the algorithm to patients in the control group, we can analyze whether

clinical decisions in the control group change over the course of the study, converging toward the algorithmic recommendations. If we find this trend, we can statistically adjust the analysis or, at the very least, discuss it as a limitation. The worst thing a researcher can do is ignore the possibility of contamination, assuming that the groups remain isolated when clinical reality suggests otherwise.

Contamination has an interesting philosophical implication: it challenges the dichotomy between intervention and control. In pharmacology, placebo and treatment are distinct entities. In software, the intervention may be a process of knowledge transformation that, once initiated, cannot be selectively reversed. Perhaps we should think less about evaluating “the software” as a static object and more about evaluating “the introduction of the software” as a dynamic process that irreversibly changes clinical practice. This perspective opens the door to study designs that embrace, rather than combat, the inevitability of learning and adaptation.

14.1.4 TIME RUNS FASTER THAN THE PROTOCOL

Perhaps no difference is characteristic of software because of its ability to change during the study itself. Once manufactured and encapsulated, a drug remains chemically stable for years, allowing for long-term trials that evaluate exactly the same substance from start to finish. Software, on the other hand, is inherently mutable. Developers may detect critical errors during the trial and feel an ethical obligation to correct them. They may implement improvements on the basis of user feedback. They may adapt to changes in operating systems or hospital integrations. Every update, however small it may seem, raises an uncomfortable methodological question: are we evaluating the same intervention as at the start?

This volatility is not a minor flaw; it is a fundamental characteristic that challenges one of the basic assumptions of experimental methodology: the constancy of the intervention. When a software clinical trial lasts twelve months, the version deployed at month six is likely to differ substantially from the initial one. Machine learning algorithms can be retrained with new data, changing their recommendation patterns. User interfaces can be redesigned to improve usability. Workflows can be optimized. From the end-user’s perspective, these improvements are desirable. From a methodological perspective, they introduce variability that complicates the interpretation of results.⁽³⁾

The question of when to “freeze” the software for evaluation purposes has no single answer. Freezing at the start of the trial ensures temporal comparability but may mean evaluating an obsolete or defective version. Allowing updates maintains clinical relevance but makes it difficult to know which specific version is responsible for the observed results. Some studies adopt a compromise: they freeze the software during specific evaluation periods, allowing updates between these periods and restarting data collection after each major change. Others meticulously document all modifications and perform sensitivity analyses to determine whether the results vary depending on the version used.

The speed of software evolution has implications for external validity that go beyond trial design. Even if we demonstrate that a specific version of the software is effective under controlled conditions, how long will that evidence remain valid? An algorithm trained on 2022 data may become less accurate by 2024 because of changes in epidemiology, diagnostic practices, or the population served. This suggests that software evaluation should not be a one-time event culminating in regulatory approval but rather a continuous process of postmarket surveillance, more akin to pharmacovigilance but adapted to the speed of technological change.

For developers, this means that trial design must incorporate flexibility from the outset. The protocol must clearly specify which types of changes are considered minor (correction of typos in the interface) versus major (modification of the core algorithm), and this establishes procedures for each case. The exact version used by each participant at every point in the study must be documented. A per-protocol analysis comparing results among users of different versions must be planned, in addition to the conventional intention-to-treat analysis. This added complexity is the price of evaluating an intervention that, unlike a drug, does not remain inert in a vial.

14.1.5 GAME-CHANGING CONTEXTS. WHY IS ONE HOSPITAL NOT THE SAME AS ANOTHER?

Generalizing results has always been a challenge in clinical research, but in medical software, it takes on qualitatively different dimensions. A drug administered at equivalent doses

should produce similar biological effects in different populations, with similarities in metabolism and comorbidities assumed. Software, on the other hand, is deeply embedded in complex sociotechnical systems where context is not a neutral backdrop but an active determinant of performance. The same algorithm can thrive in a tertiary care hospital and fail spectacularly in a regional center, not because of technical flaws but because of differences in infrastructure, organizational culture, and workflows.

Consider an electronic prescribing system with drug interaction alerts. In a hospital with robust information systems, technical support staff available around the clock, and a safety culture that values reviewing alerts, the system can significantly reduce medication errors. In a hospital with intermittent connectivity, where alerts take a long time to load and doctors are accustomed to ignoring intrusive pop-ups, the same system may increase frustration without improving safety, or worse, it may foster complacency if clinicians assume that the system will catch all errors. The software's effect is entirely mediated by these contextual characteristics.

This contextual heterogeneity has profound methodological implications. Single-center trials, while easier to conduct, have limited external validity. A study conducted at an elite teaching hospital may overestimate the effectiveness of the software by failing to reflect the resource constraints of most facilities. Even traditional multicenter trials may fail to capture this variability if they select hospitals that are too similar to one another. Deliberately heterogeneous designs are needed that include centers of different sizes, resource levels, and organizational cultures, allowing for moderation analyses that identify in which contexts the intervention works best.

The interaction between software and context is not static; it evolves over time. A system implemented in a hospital may perform poorly initially because of resistance to change, improve as users adapt, and subsequently deteriorate if training is not maintained in the face of staff turnover. Capturing this dynamic trajectory requires longitudinal study designs with multiple measurement points, moving away from the static view of the classic clinical trial. It also requires mixed methods that combine quantitative outcome data with qualitative assessments of contextual factors, using theoretical frameworks such as implementation science to interpret why the same software produces different results in different settings.

For clinicians involved in these studies, understanding this contextual variability is essential for critically interpreting the literature. An article reporting positive results should be read with the following implicit question: "In what type of hospital was this study conducted, and does it resemble mine?" If the answer is no, the evidence remains valuable but should be applied with caution, perhaps if local adaptation or pilot reevaluation is needed before large-scale implementation.

14.1.6 COMPARATORS, METRICS, AND THE NATURE OF SUCCESS

Choosing the right comparator is a critical decision in any clinical trial, but in medical software, there are no obvious answers. In pharmacology, the natural comparator is usually a placebo or the standard of care. In software, what constitutes the equivalent of a placebo? Standard practice without the algorithm? Standard practice plus additional training to match the time spent on the intervention? Another competing software product? Each choice answers a different question and yields conclusions with varying practical implications, and none is inherently correct.

If we compare against standard practice without software, a positive effect demonstrates that the algorithm adds value relative to the status quo. However, it does not tell us whether the benefit is due to the specific technology or simply to the additional care involved in its implementation. If we compare against an equivalent training intervention, we can isolate the algorithm's specific added value. However, this design may underestimate the actual value if, in practice, intensive training without software is never implemented, making the comparison artificially unfavorable to the technology. The choice should be guided by the relevant clinical question: do we want to know if this software is better than nothing or if it is better than other ways of spending the same resources?

The outcome metrics present parallel dilemmas. In pharmaceuticals, the endpoints are typically hard clinical outcomes: mortality, myocardial infarction, and cancer progression. In software, direct effects on these endpoints are rare and difficult to attribute. An algorithm that improves diagnostic accuracy will improve mortality only if subsequent treatment is adequate and timely; if the limitation lies in the availability of treatments, a better diagnosis will not save lives. Therefore, many software trials use the following process endpoints: time to diagnosis, adherence to clinical guidelines, and interobserver variability in test interpretation. These are more

sensitive but less compelling to patients and payers, who naturally care about health outcomes, not intermediate processes.

Causal attribution is complicated by the multiplicity of intervening factors. A decision support system may recommend the optimal treatment, but if the patient does not adhere to the medication for social or economic reasons, the clinical outcome will not improve. If the system is implemented in a hospital where surgical wait times are long, diagnosing a surgical condition more quickly will not accelerate the definitive intervention. This does not invalidate the software, but it does require that trials measure not only the direct effect but also the mediators and moderators of the effect. Mediation analyses can decompose the total effect into the portion due to diagnostic improvement versus the portion due to changes in treatment, revealing exactly where the technology is adding value.

The sample size required to detect software effects is typically larger than that for potent drugs because the effects are smaller and variability is greater. This makes traditional trials prohibitively expensive, driving interest in alternative designs. Interrupted timeseries studies, which compare trends before and after implementation, may be more feasible. Before-and-after designs with historical controls, although vulnerable to selection bias, can provide sufficient evidence for implementation decisions when randomized trials are not feasible. The methodology must serve the clinical question, not the other way around, and sometimes the best methodology is the one that produces reproducible evidence rather than the one that maximizes internal validity.

14.1.7 BEYOND THE CLASSIC TRIAL: TOWARD ADAPTIVE METHODOLOGIES

The fundamental differences between software and drugs should not lead us to methodological discouragement but rather to creative innovation. The scientific community is developing designs specifically tailored to the peculiarities of digital interventions—designs that maintain rigor while embracing the complexity inherent in medical software. These methodologies are not concessions to scientific weakness but are sophisticated responses to complex questions that traditional designs simply cannot adequately address.

Cluster randomization, mentioned earlier, is perhaps the most established adaptation. Assigning entire hospitals or departments rather than individual patients eliminates cross-group contamination and allows for the evaluation of effects at the organizational level. However, this requires special considerations in statistical analysis, as the unit of observation (the patient) is nested within the unit of randomization (the hospital), violating the assumption of independence of observations. Mixed-effects methods or generalized estimation equations are necessary to obtain valid standard errors, and the number of clusters must be sufficiently large to maintain statistical power, typically more than twenty per group.

Staged or phased implementation designs offer an elegant compromise between rigor and feasibility. All the participating centers eventually receive the intervention, but at randomized times. During the period when a center has not yet received the software, it serves as a control for the centers that already have it. This design is politically appealing because no site is permanently disadvantaged, and it allows for the continuous accumulation of evidence, enabling the study to be stopped early if the results are convincing or extended if more data are needed. Staged randomization also facilitates the analysis of implementation effects, examining how outcomes vary according to the time elapsed since introduction.

Hybrid effectiveness-implementation studies represent perhaps the most promising evolution. They recognize that technically excellent software can fail if not implemented properly and that mediocre software can have a disproportionate impact in the right context. These designs simultaneously evaluate clinical outcomes and the contextual factors that influence adoption and effectiveness. They do not settle to ask “does it work?”, but rather investigate “does it work, for whom, under what circumstances, and why?”. This approach requires multidisciplinary teams that bring together clinical experts, implementation scientists, health economists, and software developers, reflecting the inherently interdisciplinary nature of digital medicine.

For software developers, understanding these methodologies is just as important as knowing the technical requirements. An algorithm designed without considering how it will be evaluated may prove impossible to study rigorously. For example, if the system requires extensive customization for each hospital, a multicenter trial becomes methodologically problematic because the intervention differs substantially across sites. If the system generates usage data that cannot be exported for privacy reasons, analyses of adherence and mechanisms are limited. Early collaboration between developers and research methodologists can lead to the design of

systems that are not only clinically useful but also scientifically evaluable.

For clinicians, familiarity with these alternative methodologies expands their ability to contribute to the evidence base. Not all physicians have access to the resources of a traditional randomized clinical trial, but many can participate in timeseries studies or implementation evaluations. Their knowledge of the local context is invaluable for interpreting why a software program performs differently in their hospital than in others do and for identifying local modifications that could improve its effectiveness. Research on medical software should not be the exclusive prerogative of large academic centers; adaptive methodology allows meaningful contributions from everyday clinical practice.

The methodological horizon continues to expand. Adaptive designs, where the protocol can be modified during the study on the basis of interim results, are particularly relevant for rapidly evolving software. Platform trials, where multiple interventions are tested on a common infrastructure, allow for more efficient comparisons. The use of electronic health data for real-world effectiveness observational studies, although with well-known limitations, can complement controlled trials by providing evidence on long-term effects in diverse populations. The methodology is alive, evolving alongside the technology it seeks to evaluate.

14.2 PROSPECTIVE VS. RETROSPECTIVE STUDIES

Clinical research has built an entire cosmology around a seemingly simple distinction: do we look forward or backward? Do we plan data collection before events occur, or do we analyze existing information that was accumulated for other purposes? This dichotomy between prospective and retrospective studies is not merely technical; it is philosophical, economic, and temporal. It defines which questions we can answer, with what confidence we can answer them, and at what cost. When the subject of study is medical software, this distinction takes on additional layers of complexity because the object of study changes faster than the study itself and because its effects are so intertwined with the human context that the temporal direction of our gaze determines not only what we see but also how we interpret it.

The choice between the two approaches is not neutral. A prospective study allows us to design data collection with surgical precision, defining exactly what to measure, how to measure it, and when. A retrospective study forces us to accept what the medical record chose to document—with its omissions, inconsistencies, and eloquent silences. However, this prospective precision comes at a price: time. While the prospective study progresses slowly and methodically, the software being evaluated may have been updated three times, changed versions, or withdrawn from the market because of obsolescence or competition. Rigorous evidence runs the risk of reaching the world as a postmortem diagnosis of a product that no longer exists. This tension between methodological rigor and temporal relevance is the central dilemma that every digital health researcher must navigate.

14.2.1 PROSPECTIVE STUDIES AS THE GOLD STANDARD

Prospective studies occupy the highest rung in the hierarchy of medical evidence, and this privileged position is not arbitrary. By planning data collection before the events of interest occur, the researcher exercises almost artisanal control over the process of knowledge generation. They can precisely define the variables they consider relevant, standardize measurement instruments to ensure interobserver reliability, and establish collection protocols that minimize unwanted variability. In the context of medical software, this meticulous planning allows us to go beyond the final results to explore intermediate mechanisms: how clinicians interact with the interface, how often they follow the algorithm's recommendations, and whether there are unanticipated effects that only direct observation can reveal.

Considering the following case, a prospective design allows us to install specific sensors to capture high-frequency physiological data, train nursing staff in the standardized use of the system, and define composite endpoints that combine mortality with intermediate events such as transfers to intensive care units. We can collect process measures, such as the time between the alert and the medical intervention, which help us understand why the system works or fails. This wealth of information is impossible to obtain retrospectively, where we depend on what the clinical record deemed worth noting, typically at discrete moments and often insufficient to capture the dynamics of acute deterioration.

The ability to establish causal relationships with greater confidence is perhaps the greatest strength of prospective studies. By controlling the assignment of the intervention—whether through randomization or careful quasiexperimental designs—we can argue more strongly that the observed differences are due to the software and not to confounding factors. If we randomize hospitals to implement the algorithm in a phased manner, we can compare departments that are similar in both observable and unobservable characteristics, thereby strengthening causal inference. This methodological strength is especially crucial when the intervention is complex and its effect is mediated by multiple contextual factors that would be impossible to statistically adjust for in a retrospective analysis.

However, prospective studies have their drawbacks. The financial cost of a well-designed prospective study can be prohibitive for startups or academic researchers without industry funding. The resources required to recruit participants, monitor data quality, and maintain protocol integrity over months or years scale rapidly. Furthermore, the time required from study conception to publication of results may be incompatible with the pace of the technology market. A machine learning algorithm trained on 2022 data may require retraining in 2023 because of changes in epidemiology or clinical practices, making the results of the prospective study—however rigorous—applicable only to a historical version of the software.

External validity represents another persistent concern. Prospective studies, in their quest for methodological control, often implement software under conditions that are more ideal than those in real-world practice are. Technical support is more intensive, staff training is more comprehensive, and integration with existing systems is more carefully planned. When the same software is subsequently deployed in hospitals without these research resources, the results may be less favorable. This gap between efficacy (what the software can achieve under optimal conditions) and effectiveness (what it achieves in real-world practice) is particularly pronounced in digital interventions, where implementation is as critical as the technology itself.

14.2.2 THE RETROSPECTIVE VIEW

Retrospective studies offer an irresistible temptation: the promise of quick and inexpensive answers by tapping into the hidden treasures in existing health records. Instead of waiting years for events to occur, the researcher can analyze thousands of patients in a matter of months, explore associations and generate hypotheses at a speed that prospective designs cannot match. For medical software, this approach is especially valuable in the early stages of development, when it is necessary to quickly evaluate an algorithm's technical performance, compare it with historical practice, or identify subgroups where it might be particularly beneficial or harmful.

The main strength of retrospective studies is their ability to reflect real-world clinical practice in terms of all its variability and imperfection. Electronic health record data capture the authentic noise of everyday medicine: constant interruptions, decisions made with incomplete information, and deviations from protocols for clinical or logistical reasons. An algorithm that performs well on these “dirty” data is more likely to work in the real world than one validated solely on clean, curated databases. Furthermore, compared with prospective studies, retrospective studies can include much larger and more diverse populations, allowing for an assessment of the generalizability of the software and the detection of rare effects that emerge only in massive samples.

Let us consider a concrete example. A team developed an algorithm to predict the risk of readmission in patients with heart failure. A retrospective study analyzed five years of records from ten hospitals, including tens of thousands of patients with various demographic characteristics and comorbidities. It could compare the algorithm's predictions with actual discharge decisions, identifying cases where the algorithm would have suggested a prolonged hospitalization that turned out to be unnecessary, or vice versa. This large-scale evaluation provides invaluable insights into the software's performance under real-world conditions—insights that would be prohibitively expensive to obtain prospectively.

However, retrospectivity involves systemic vulnerabilities that must be honestly acknowledged. Data quality is the Achilles' heel of these studies. Clinical records were designed for patient care, not for research, and reflect the priorities, habits, and limitations of those who completed them. Diagnoses may be coded inconsistently, using different classification systems or levels of specificity. Important variables may be systematically missing, such as the patient's socioeconomic status, their actual adherence to medication, and their treatment preferences. Timestamps may be documented approximately, with events grouped by date without the hourly granularity required for certain analyses.

Selection and information biases are particularly insidious in retrospective studies of medical software. Selection bias arises when the included patients do not adequately represent the target population. For example, if we evaluate a sepsis detection algorithm using only patients who were ultimately diagnosed with sepsis, we are ignoring the false positives it would have generated in the general emergency department population. Information bias occurs when the way data are recorded is related to the exposure or outcome. A physician using a diagnostic support algorithm may record certain symptoms in greater detail precisely because the algorithm suggested them, creating an artificial appearance of agreement between prediction and reality.

Establishing causal relationships from retrospective data requires considerable statistical sophistication. Techniques such as propensity score matching, discontinuity regression, or instrumental variable models attempt to simulate the randomization we lacked, but all rely on assumptions that are rarely perfectly met. The researcher must assume that they have correctly identified and measured all important confounding factors—a bold assumption in the complex clinical context. Therefore, retrospective studies are excellent for generating hypotheses and evaluating technical performance but are limited in demonstrating that the software actually improves health outcomes in a causal manner.

14.2.3 HYBRID AND ALTERNATIVE DESIGNS

The rigid dichotomy between prospective and retrospective has given way to more flexible designs that combine elements of both approaches, seeking to maximize strengths while minimizing weaknesses. These hybrid designs are particularly relevant for medical software, where the need for rigorous evidence competes with time-to-market pressures and where the intervention itself may be modified during the study. Understanding these methodological alternatives expands the researcher's repertoire and allows us to address questions that no single design could adequately answer.

Before-and-after studies with historical controls represent a widely used intermediate category. In this design, we implement the software at a specific point in time and compare the results with those of the same service prior to implementation or with those of a similar service that has not received the intervention. This approach is more feasible than a randomized trial because it does not require randomization or simultaneous parallel groups, but it introduces specific vulnerabilities. Temporal changes in the quality of care, the epidemiology of diseases, or the composition of the served population can be confounded by the effect of the software. If we implement a COVID-19 diagnostic algorithm in 2021 and compare it with that in 2020, we compare periods that are radically different in terms of medical knowledge and healthcare pressure.

Discontinuous timeseries designs offer an additional level of sophistication that makes them particularly appealing for evaluations at the service or hospital level. Rather than comparing simple pre- and postimplementation averages, these designs analyze the trend of the indicator of interest over time, looking for changes in the level or slope of the trend coinciding with the implementation of the software. If the medication error rate gradually decreased before the algorithm and continued to decline at the same rate afterward, the software likely had no additional impact. However, if there is a sharp decrease at the time of implementation or if the rate of improvement accelerates, we have more robust evidence of a causal effect. This design controls for preexisting trends and can detect effects that simple before-and-after comparisons would overlook.

Staged randomization, mentioned in the previous section, represents another elegant synthesis. All participating centers eventually receive the intervention, eliminating ethical objections regarding the deprivation of potential benefits, but they do so at randomized times. During the period when some centers already have software and others do not, there is a rigorous prospective comparison. As time progresses, more centers enter the intervention group, leading to the accumulation of ongoing evidence that allows the study to be stopped if the results are convincing or expanded if more precision is needed. This design is particularly useful when the intervention is politically popular and it is difficult to retain controls or when contamination between groups is inevitable if they coexist in the same time and space.

Hybrid effectiveness-implementation designs go a step further, recognizing that technically excellent software can fail if not implemented properly and that mediocre software can have a disproportionate impact in the right context. These studies simultaneously evaluate clinical outcomes and the contextual factors that influence adoption: organizational culture, local leadership, available resources, and barriers perceived by users or im. They do not settle to ask

whether the software works but rather investigate for whom it works, under what circumstances, and why. This approach requires mixed methods that combine the quantification of outcomes with the qualitative assessment of processes, but it produces evidence that is more accessible to those who must decide on the adoption and scaling of the technology.

Understanding these alternative designs is crucial for the strategic planning of the evidence program. In the early phases, retrospective studies can validate technical performance and justify investment in more costly studies. Timeseries or staggered randomization designs can provide sufficient evidence of effectiveness for initial regulatory approval, while broader prospective studies are planned for the postmarketing phase. The key is to align the methodological design with the specific question that needs to be answered at each stage of the product lifecycle, recognizing that different audiences (regulators, payers, clinicians) may require different types of evidence.

14.2.4 PRACTICAL IMPLICATIONS: SELECTION, COMBINATION, AND COMMUNICATION

The choice between prospective and retrospective approaches, or between their hybrid variants, should not be an abstract decision based purely on methodology but rather a pragmatic strategy aligned with the study's objectives, available resources, and the software's stage of development. For teams developing medical technology, this choice has direct consequences for commercial viability, regulatory acceptance, and, ultimately, patient health. A poorly designed evidence strategy can doom technically excellent software to failure, whereas a smart strategy can accelerate patient access to beneficial tools.

In the early stages of development, when the algorithm is still being refined, retrospective studies are appropriate and efficient. They allow for rapid iteration, testing different versions of the software against historical data, identifying edge cases where the algorithm fails, and adjusting parameters without putting patients at risk. This internal validation phase is comparable to preclinical drug trials: it is necessary but insufficient. A common pitfall is to remain in this phase, publishing articles on the algorithm's excellent performance in curated databases, without moving toward evaluations under real-world clinical conditions.

The transition to prospective studies must be carefully planned, typically when the software has reached a level of technical maturity that justifies the investment and when clinical added value needs to be demonstrated for regulatory approval or reimbursement. This transition does not need to be abrupt; hybrid designs can serve as a bridge. A prospective study with a systematic collection of primary data can be complemented by retrospective registry analyses to expand the sample size or evaluate long-term effects. The integration of research platforms into electronic health records systems facilitates this combination, allowing prospectively collected data to be fed into registries that can be analyzed retrospectively in the future.

For clinicians who participate in these studies—whether as principal investigators, collaborators, or simply as users whose work generates the data—understanding these methodological distinctions empowers their critical engagement. They can assess whether the study design is appropriate for the research question, identify potential sources of bias that researchers may have overlooked, and contribute their tacit knowledge of real-world clinical practice. A physician who knows that emergency department records systematically underestimate alcohol consumption can point out this limitation in a retrospective study. A clinician familiar with team dynamics in their department can suggest that the implementation of the software coincided with a change in the department head, confounding the effect of the intervention.

The communication of results must be equally sophisticated. Researchers have a responsibility to be transparent about the limitations of their design, avoiding the temptation to present a retrospective study as if it were prospective or to extrapolate causal conclusions from observational associations. Readers, many of whom are busy clinicians who rely on summaries and abstracts, must be educated to always ask, “Was this a prospective or retrospective study? What type of data was used? How were confounding factors handled?” The quality of evidence in digital medicine depends as much on the sophistication of those who produce it as on the critical literacy of those who consume it.

Relationships with regulators add another layer of complexity. Drug and medical device agencies are developing specific guidelines for software evaluation, and these guidelines often distinguish between evidence requirements on the basis of the device's risk. With respect to low-risk software, well-designed retrospective studies may be sufficient. For high-risk devices

that support critical decisions, prospective studies are typically needed. Understanding these regulatory thresholds from the earliest stages of development allows for the design of an evidence program that efficiently moves toward approval, avoiding the costly mistake of having to repeat studies because the initial design did not meet regulatory standards.

The future points toward increasingly close integration between these two worlds. Electronic health records are becoming more standardized and comprehensive, narrowing the quality gap between retrospective and prospective data. Federated machine learning platforms allow algorithms to be trained and validated across multiple institutions without centralizing sensitive data, facilitating large-scale retrospective studies that maintain privacy. Postmarket continuous surveillance systems, where the software itself collects data on its performance in real-world use, blur the line between research and clinical practice. In this emerging landscape, the distinction between prospective and retrospective data may become less relevant than the distinction between high-quality, contextualized data and poor or decontextualized data.

One persistent question must be echoing: once we have generated this evidence, whether looking forward or backward, how do we translate it into action? How do we convince a skeptical department head that it is worth altering their workflows to incorporate an algorithm? How do we navigate the complex ethical and regulatory procedures that protect patients but often seem to hinder innovation? Methodology tells us what works, but implementation confronts us with the question of how to make it work in the real world, with its organizational resistance, limited budgets, and imperfect humans.

14.3 HOW TO TALK TO A DEPARTMENT HEAD TO COLLABORATE

In the world of clinical research with medical software, there is an invisible threshold that separates projects that thrive from those that sink into administrative oblivion. It is not the threshold of ethical approval, however, that is crucial. It is not that of funding, although without it nothing moves forward. It is the moment when someone—usually a young, enthusiastic researcher—sits down with a department head to ask for something seemingly simple: permission to conduct a study within their department. That conversation, which may last twenty minutes or an hour, determines whether a project secures the most powerful ally it can have or whether it becomes just another folder piled on a desk already overflowing with unfulfilled good intentions. The difference between these two outcomes rarely depends on the technical quality of the algorithm but almost always on the human ability of the person seeking access to connect with the real concerns of the person holding the key.

Hospital department heads face a reality of converging pressures that those coming from the software industry often underestimate. They are not merely dealing with bureaucratic hurdles or passively receiving academic proposals; they are managers of extremely scarce resources, ultimately responsible for the quality of care, coordinators of complex teams, trainers of the next generation of specialists, and often clinicians who maintain their own practice while working grueling shifts. In this context, a research project is not a brilliant opportunity that lights up its desk; it is an additional burden that competes with life-threatening emergencies, endless waiting lists, residents who need supervision, and quality targets that must be met quarter after quarter. Understanding this ecosystem of pressure is the first step toward designing a conversation that has any chance of success.

14.3.1 FROM THE OTHER SIDE OF THE TABLE: UNDERSTANDING THE CLINICAL MANAGER

The most common mistake when a department head is approached is to treat them as if they were fellow researchers who are interested in the methodological nuances of the study design. Nothing could be further from the truth. The department head is, above all, a manager whose success is measured by indicators of productivity, quality, and team stability. Their primary concern is not whether the algorithm uses transformer architectures or convolutional networks but whether the study's implementation will make their department function better or worse during the coming months. This fundamental understanding must shape every aspect of the preparation and execution of the initial conversation.

To connect with these priorities, it is necessary to translate the project into the language of its tangible benefits for the department. It is not enough to say that the algorithm has a sensitivity of ninety-five percent; one must explain that this means detecting three more cases of sepsis per

week, which could translate into two fewer ICU admissions and a reduction in the average length of stay. It is not enough to mention that the study will include 200 patients; you must highlight that this represents coauthorship on a high-impact publication for the department, visibility at international conferences, and priority access to technology that other centers do not yet have. Department heads value academic prestige, but they value it even more when it is clearly linked to improvements in daily clinical practice and public recognition of their team.

Concerns about staff are particularly sensitive. Department heads live with the constant fear of their physicians burning out, the turnover of experienced nurses, and the difficulty in retaining talent. Any project they perceive as an additional source of stress for a team already at its limit will be rejected, regardless of its scientific merit. Therefore, the proposal must explicitly address how staff will be protected: by hiring dedicated support staff for the study, by integrating data collection into existing workflows without duplicating efforts, and by scheduling data collection and collection times that avoid peak hours of patient care. This study demonstrates that the team's well-being builds more trust than any promise of an impending technological revolution does.

It is also crucial to understand the internal power dynamics of departments. The department head must answer hospital management, quality committees, and patient associations. A project that could be interpreted as experimental, risky, or likely to generate complaints represents a political burden that many prefer to avoid. Therefore, the proposal must include elements of security and risk management: clear protocols for adverse situations, data security committees, and patient communication plans that minimize misunderstandings. The department head needs to feel that they can defend the project to their superiors with solid arguments regarding security and potential benefits.

The timing of the approach is also strategic. Approaching the department head immediately after a tense meeting with management or during a period of peak clinical pressure is a surefire way to reject the proposal because of burnout. Conversely, identifying moments of relative calm, or of particular interest in innovation (for example, following an audit visit that highlighted areas for improvement), can significantly increase the chances of success. This contextual intelligence requires prior networking and informal conversations with department members who can provide insight into the head's current mood and priorities.

14.3.2 ART OF TRANSLATION

The language gap between software developers and clinicians is deep and mutually bewildering. For those coming from the tech world, it is natural to discuss microservice architectures, RESTful APIs, and performance metrics such as the AUC or F1score. For the clinician, this vocabulary is white noise that obscures the essential message: does this help my patients or not? The inability to translate technical concepts into tangible clinical impact is among the main causes of failure in these conversations, and overcoming it requires a deliberate effort at empathy and clarification.

Effective translation is not trivial simplification; it is conceptual reconstruction. It is not about omitting technical complexity but about reconnecting it to outcomes that the clinician values. Instead of saying “our algorithm has a ninety-two percent accuracy rate in detecting intracranial hemorrhages,” one could say “this means that out of every hundred scans the algorithm flags as urgent, ninety-two actually require immediate intervention, reducing the false alarms that currently overwhelm your radiology department.” Instead of mentioning “training with fifty thousand MRI images,” one can explain, “the system has learned from more cases than any radiologist would see in their entire career, identifying subtle patterns that human experience might overlook during night shifts.” Technical metrics become clinical benefits; data volume transforms into superior accumulated expertise.

This translation must extend to the entire project narrative. The methodological design, which for the researcher is the heart of the study, must be presented as a plan to minimize disruptions. Data collection, which for the methodologist is crucial for validity, must be explained as something that will occur in the background, without interfering with patient care. Quality controls, which ensure scientific rigor, must be framed as safeguards for the safety of the service's patients. Every technical element must pass through the filter of immediate clinical utility.

Preparing for this translation requires preliminary work. The researcher must study the target department, read its recent publications, understand its prioritized lines of research, and be familiar with the clinical problems most frequently mentioned in meetings or internal

communications. This field research allows for personalizing the message: if the department has published on emergency department workload, emphasize how the algorithm reduces decision time; if they have highlighted the importance of training, underscore how the project will include training sessions for residents. A generic message, however technically brilliant, will compete with dozens of similar proposals; a personalized message demonstrates respect and genuine commitment.

It is also important to anticipate the questions the department head will ask, which are rarely technical. “How much time will this take away from my doctors?” “What happens if the algorithm makes a mistake and we don’t notice?” “How will this affect our relationship with patients?” “Who takes responsibility if there is a problem?” These questions reveal concerns about workflow, safety, clinical ethics, and legal liability. Having prepared, honest, and detailed answers makes the difference between a professional interlocutor and a seller of empty promises. It is particularly important not to downplay potential risks; acknowledging them and explaining how they are mitigated builds more trust than implicitly denying them.

14.3.3 THE ECONOMICS OF THE EXCHANGE

A successful research project cannot be an extractive relationship where the clinical service provides labor, patients, and data, while external researchers take the publications, patents, and commercial benefits. This perception of inequality, which is too common in the history of biomedical research, is poisons for sustainable collaboration. Building a genuinely reciprocal relationship requires explicitly designing what the department receives in exchange for its participation and ensuring that these benefits are valued and tangible.

Reciprocity can take many forms, and the key is to identify which forms align with the department’s specific priorities. For some, coauthorship on high-impact publications is a crucial academic asset for departmental accreditation or the promotion of team members. For others, priority access to developed software—potentially before its general commercial release—represents a tangible competitive advantage. Some departments value the specialized training the project can offer their residents, exposing them to research methodologies or emerging technologies. Others may need funding to hire additional staff to alleviate the clinical workload or IT infrastructure to enhance their capabilities beyond the specific project.

It is important that these considerations are not vague future promises but rather concrete commitments documented from the outset. A memorandum of understanding specifies who will be a coauthor on which publications, what access the department will have to the software poststudy, what training will be provided and for how long, serves to align expectations and avoid future conflicts. This document does not need to be a complex legal contract, but it must be clear and signed by both parties. Formalizing the agreement demonstrates seriousness and protects both sides from misunderstandings that may arise months later, when the original participants in the conversation are no longer available.

Reciprocity also operates on an emotional and relational level. The department head and their team must feel respected as intelligent and valuable partners, not as mere executors of a protocol designed elsewhere. This involves genuinely involving them in the study design, soliciting their opinion on the feasibility of procedures, adapting the protocol to their suggestions, and publicly acknowledging their contribution in presentations and publications. A researcher who treats the clinician as an equal collaborator fosters loyalty that extends beyond the individual project, opening doors for future collaborations and creating advocates for the work within the clinical community.

It is also wise to consider institutional reciprocity, not just individual reciprocity. Agreement with the department head should include provisions that benefit the hospital: recognition in press releases, participation in outreach events, and networking opportunities with other centers participating in the study. These institutional benefits strengthen the department head’s support with hospital management and contribute to an organizational culture favorable to research that extends beyond the specific project. A hospital that perceives research as providing prestige and resources is more willing to facilitate future studies.

14.3.4 THE DANCE OF FLEXIBILITY: ADAPTING TO MOVE FORWARD

Approaching the conversation with a rigid, unchangeable protocol is a strategy guaranteed

to fail. Department heads know the nuances of their own reality better than any external researcher does, and their suggestions for adaptation are not capricious obstacles but are invaluable contributions of local insight. Methodological flexibility is not a scientific weakness; it is intelligent pragmatism that increases the study's feasibility and often improves its quality. A researcher who actively listens and adapts builds trust; one who imposes resistance.

This flexibility can manifest in multiple dimensions. Temporally, the department head may suggest that certain study procedures be conducted at specific times of the day or week to minimize conflicts with peak clinical activity. Methodologically, they may propose excluding certain patient subgroups for ethical or logistical reasons that the researcher had not considered. Technically, they may point out limitations in the IT infrastructure that require modifications to the software integration. In all these cases, the appropriate response is not to defend the original protocol at all costs but to assess whether the proposed adaptation seriously compromises the scientific objectives or, conversely, improves external validity by aligning with actual practice.

Anticipating objections is an advanced skill that distinguishes experienced researchers from novices. Instead of passively waiting for the department head to raise issues, the prepared researcher brings them up first, demonstrating that they have considered the project from the department's perspective. "I know that one of your concerns will be the additional burden on residents; that is why we have budgeted for the hiring of a dedicated research coordinator to handle all administrative tasks." "I understand that the emergency department is extremely busy on weekends; the protocol allows for excluding those days from intensive data collection." This foresight demonstrates empathy and professionalism and shifts the conversation from a defensive negotiation to a constructive collaboration.

However, flexibility has ethical and scientific limits that must not be crossed. Adaptations that compromise patient safety, introduce unacceptable systematic biases, or violate fundamental regulatory principles must be politely but firmly rejected. The key is to explain why certain elements are nonnegotiable while offering creative alternatives that address the underlying concern without sacrificing the study's integrity. For example, if the head of a department wishes to exclude the most severely ill patients out of fear of complications, one can explain that this would skew the results but offer a specific intensive monitoring protocol for that subgroup that ensures safety without excluding them from the analysis.

Documenting agreed-upon adjustments is essential. Following the conversation, a written summary outlining the agreed-upon points, modifications to the original protocol, mutual commitments, and specific next steps helps prevent misunderstandings and provides a reference as the project progresses and individual memories fade. This document should be sent promptly, ideally the day after the meeting, along with a personalized thank-you email that reinforces the rapport established. Professionalism in these administrative details is interpreted by the department head as an indicator of how the project as a whole will be managed.

14.3.5 BUILDING TRUST: CREDIBILITY, PATIENCE, AND PERSISTENCE

Department heads have witnessed too many projects that promised much and delivered little, researchers who disappeared after obtaining the data and technologies that never materialized. This history of broken promises has fostered a healthy skepticism that the researcher must overcome through credible signs of seriousness and the ability to execute. Credibility is not declared; it is built through tangible evidence that this project is different and that the person proposing it has the resources, experience, and commitment to see it through to a successful conclusion.

Signs of credibility include previous results from similar projects, preferably documented in publications or presentations that the department head can consult. Support from other prestigious departments or institutions that have already approved the project serves as social validation: if the general hospital has accepted it, perhaps it is worth considering. Secured funding, demonstrable through documentation from the funding agency or the sponsoring company's contract, eliminates concerns that the project will run out of resources halfway through. A visible multidisciplinary team, with clear roles and documented experience, demonstrates the capacity to address the project's various technical, clinical, and regulatory dimensions.

Internal endorsement is particularly powerful. If someone from within the department itself—someone respected by the head—can attest to having participated in the project's design and can vouch for its viability, barriers to mistrust are significantly broken down. For this reason, prior networking with clinicians in the department, building informal relationships, and

participating in scientific meetings where preliminary ideas are presented are investments that greatly facilitate formal conversation with the head. An internal clinical “champion” who opens the door and accompanies the presentation is an invaluable asset that justifies the time spent cultivating these relationships.

Patience is an essential virtue in this process. Department heads have schedules filled with unforeseen emergencies, and a research proposal, no matter how interesting, is rarely the top priority. It may take weeks to receive a response, and during that time, the researcher must maintain polite and regular contact that keeps interest alive without coming across as a nuisance. A reminder every two or three weeks, with some relevant progress (new funding secured, ethics committee approval, recent related publication), demonstrates that the project is moving forward and keeps the conversation on the recipient's mind. Patient persistence trumps aggressive pressure.

When the response finally arrives, whether positive or negative, the researcher's reaction makes all the difference for true interactions. An acceptance should be celebrated with genuine gratitude and immediately followed by the specific steps agreed upon, demonstrating efficiency that validates the trust placed in you. A rejection should be accepted professionally, requesting feedback on what could have been done differently and leaving the door open for future collaborations when circumstances change. Burning bridges with a frustrated department head close not only that project but also potentially an entire network of contacts within that institution.

For software developers, mastering these communication and relationship management skills is just as crucial as any technical competency. A perfect algorithm that never makes it out of the lab because the necessary clinical partnerships were not established is, from the patient's perspective, equivalent to not existing. Investing time in understanding the hospital environment, in cultivating authentic human relationships, and in learning the language and priorities of those who lead the departments is among the highest-return investments a healthcare innovation team can make. Technology does not spread on its own merit; it spreads because specific people decide to trust other specific people to implement it.

14.4 INSTITUTIONAL REVIEW BOARDS (IRBS): HOW TO PREPARE AN APPLICATION

For those in the world of technology development, accustomed to the breakneck speed of iterative cycles and the “fail fast, learn fast” culture, the first encounter with a Drug Research Ethics Committee (CEIm) can feel like a journey through alien territory. Deadlines measured in weeks rather than hours, seemingly endless forms, questions that challenge assumptions the team takes for granted—all of these conspires to generate frustration and, at times, disdain. The temptation to treat ethical approval as a bureaucratic formality to be overcome with the least effort is understandable, but it represents a catastrophic mistake that can paralyze projects for months or, worse still, expose patients to avoidable risks. Understanding that behind every committee requirement lies a history of human protection—of past abuses that society decided not to repeat—transforms the experience from an obstacle into an opportunity for professional growth. (4)

Ethics committees are not abstract entities designed to stifle innovation; they are the institutional embodiment of principles forged after decades of unethical research. The Nuremberg Code emerged from the trials of doctors who experimented on prisoners without consent; the Declaration of Helsinki consolidated international standards following decades of abuses in developing countries; and the Belmont Report established the principles of respect for persons, beneficence, and justice that continue to guide contemporary research. When a committee requires detailed informed consent, it is not being nitpicky; it is ensuring that no person is reduced to a mere means for the ends of others. When it requests a rigorous analysis of risks and benefits, it does not create unnecessary obstacles; it is compelling researchers to confront the real consequences of their work on human lives. This historical genealogy must inform every step of the preparation of an ethics application. (5,6,7)

14.4.1 THE ART OF PREPARATION: TEN RULES FOR A SUCCESSFUL APPLICATION

Start the process early, when the study is still in the conceptual design phase, rather than when the software has already been developed and all that is left is the “go-ahead” to test it. Ethics committees, with their accumulated expertise in methodology and participant protection,

can fer suggestions that modify fundamental aspects of the project: the definition of inclusion and exclusion criteria, the selection of primary outcome measures, and consent procedures. If these recommendations arrive when the protocol is already set in stone in code and procedures, the changes may be technically impossible or financially ruinous. If they arrive during the design phase, they can be organically integrated, improving the study without disruption. Many committees offer nonbinding preliminary consultations that allow for validation of the general direction before investing resources in full development; ignoring this opportunity constitutes methodological negligence.

It seems obvious but is systematically overlooked: read and follow the specific instructions of the committee to which the application is addressed. Each IRB has its own forms, document checklists, submission deadlines, and scheduled meetings. Submitting a generic application downloaded from the internet—or worse, adapted from another committee without verifying specific local requirements—guarantees unnecessary delays and gives the impression of a lack of professionalism. Taking the time to familiarize yourself with the committee's website, download the updated documentation, and understand the process flow from receipt to resolution is an investment that pays off in subsequent efficiency. If the committee offers informational sessions or has support staff available for preliminary consultations, the use of these resources demonstrates research maturity and can prevent costly errors.(4)

It requires a conceptual translation of the project into the language of medical ethics. For the development team, the focus may be on the model architecture, hyperparameter optimization, ROC curves, and confusion matrices. For the committee, these are technical abstractions that only acquire ethical relevance when connected to concrete human impacts. The application must answer fundamental questions with absolute clarity: What specific risks—does each individual participant face? How are these risks minimized through study design or safety procedures? What benefits are anticipated, who will receive them, and with what level of evidence? How is the confidentiality of sensitive data guaranteed? How will informed consent be obtained and documented? An application that does not explicitly address these ethical dimensions forces the committee to request clarifications, initiating back-and-forth cycles that consume valuable weeks.

It is essential to anticipate the difficult questions the committee will inevitably raise regarding sensitive issues. IRBs are particularly vigilant with respect to the inclusion of vulnerable people who cannot protect themselves, such as minors; individuals with significant cognitive disabilities; patients in medical emergencies where informed consent seems impossible; and pregnant women, prisoners, and individuals in situations of extreme economic dependence. If the study involves any of these groups, the application must contain a solid ethical justification for why their inclusion is scientifically necessary and cannot be replaced by less vulnerable populations, along with a detailed description of special protective measures. Equally sensitive is the use of retrospective data without specific consent, where it must be argued why obtaining consent is impractical and how the irreversibility of anonymization is guaranteed.(4)

Advocate for radical transparency regarding the study's limitations. No research design is perfect, and members of ethics committees—generally experienced researchers—are well aware of this. Attempting to conceal methodological limitations from the ethics committee, minimize potential risks, or exaggerate anticipated benefits is often counterproductive. If the committee discovers inconsistencies or omissions during its review, trust in the research team erodes, and the process drags on indefinitely. It is preferable to be proactively honest about the limitations, explain why they were accepted in the design, and argue why, despite them, the study maintains a favorable benefit–risk ratio and provides sufficient scientific or social value to justify exposing participants to residual risks.

Focus on the exceptional quality of the patient information and informed consent documents. These texts are the primary ethical interface between the study and potential participants, and committees scrutinize them with almost forensic thoroughness. They must be written in language accessible to people without medical or technical training, avoiding both incomprehensible specialized jargon and paternalism that underestimates patients' comprehension ability. They must clearly explain what participation entails in practice; what specific procedures it involves; how much time it requires; what physical, psychological, or social risks it entails; what direct or indirect benefits the participant can expect; the absolute right to withdraw at any time without penalty or impact on their regular medical care; and how their personal data will be handled. Dedication of the iterative time to refine these documents, testing them with focus groups of potential patients, is among the highest-return investments in the entire ethical approval process.

It addresses data management and privacy in an era of strict regulation and massive media

leaks. The European Union's General Data Protection Regulation and equivalent regulations in other jurisdictions have raised the bar for requirements regarding the processing of sensitive health information. The application must specifically detail which categories of data will be collected, for what legitimate purposes, who will have access at each level of identification, where and how they will be stored physically or digitally, for how long they will be retained, and what technical and organizational security measures will be implemented. If the application envisions international data transfer, legal safeguards must be specified; if cloud computing services are used, the provider's security must be justified; if data sharing with other researchers is planned, the access control mechanism must be described. A well-articulated and transparent data policy serves as a letter of credibility to the committee.

This concerns the demonstrable suitability of the research team. IRBs evaluate not only the protocol on paper but also the actual capacity of the individuals who will carry it out. The application must include updated CVs of the principal investigators, specific training in clinical research, prior experience in similar studies, and, if relevant, certifications in data protection or specialized knowledge in the clinical domain of the study. A multidisciplinary team combining medical, statistical, ethical, and technical expertise demonstrates that the project comprehensively addresses its complex dimensions. If researchers in training are involved, the supervision and mentoring system that will ensure their adequate performance must be clearly specified.⁽⁴⁾

Absolute transparency regarding funding and potential conflicts of interest is needed. Committees need to know who is funding the study to assess potential biases in the design, interpretation, or publication of results. The application must explicitly state all sources of funding, both public and private, and any commercial or academic relationships that could create a conflict of interest. If the software is developed by a company that plans to commercialize it later, a detailed explanation must be provided of how the independence of the statistical analysis and the right to publish negative or adverse results are guaranteed. The concealment of commercial interests, if discovered, can invalidate the entire application and irreparably damage the team's reputation.

Exercise patience and professional courtesy in all interactions with the committee. CEIm committees are generally composed of volunteer professionals—doctors, nurses, lawyers, and patient representatives—who devote their personal time to reviewing others' projects while maintaining their own demanding professional obligations. Their responses may not be immediate, and their questions or requests for modifications may sometimes seem disproportionate or misinformed regarding technical aspects. Responding politely, providing substantive arguments without a defensive attitude, is the only viable strategy. If the committee requests specific changes, implement them unless there are compelling ethical or scientific reasons to disagree, in which case the disagreement should be explained respectfully and constructive alternatives offered. Aggressive confrontation or disdain for the ethics process does not lead to any favorable outcomes. ⁽⁴⁾

14.4.2 BEYOND PERMISSION: ETHICS AS A COMPASS FOR DEVELOPMENT

For software developers, submitting to the scrutiny of an ethics committee represents a unique opportunity for professional reflection that rarely arises from purely technical processes. The questions the committee raises force us to consider dimensions of the project that the algorithmic approach tends to obscure: patients as people with autonomy and dignity, not as data points; psychological and social risks, not just technical risks of system failure; and the social value of the research, not just the model's metric performance. This confrontation with the ethical dimension, although initially uncomfortable, often results in software that is more mature, more responsible, and better prepared for the complexities of the real world.

Software that has survived rigorous ethical scrutiny has demonstrated its ability to justify itself against independent standards of human protection. It has had to articulate its benefits convincingly, acknowledge its limitations honestly, and demonstrate that the risks to participants are minimized and proportionate to the expected benefits. This process of public justification constitutes a form of social responsibility that distinguishes ethical research from irresponsible experimentation. The developer who goes through this process emerges with a deeper understanding that medical technology is not valued for its intrinsic sophistication but for its demonstrable contribution to human well-being.

For clinicians involved in medical software projects, their prior experience with ethics committees is an invaluable asset they can bring to the multidisciplinary team. They understand

regulatory language, anticipate the typical concerns of reviewers, and have navigated the process multiple times in their own research. They can guide developers in preparing documentation, review forms from the perspective of someone who knows what the committee truly values, and sometimes act as principal investigators submitting the formal application, lending clinical credibility to the project. Their role goes beyond merely facilitating access to patients; it includes expertly navigating the governance structures that regulate all research involving human subjects.

The future of the relationship between medical software developers and ethics committees points toward greater specialization and mutual understanding. Some institutions are creating specific IRBs or subcommittees dedicated to digital projects and technological innovation, incorporating members with training in health informatics, cybersecurity, and data science. These specialized committees can more accurately assess the specific risks of medical software—from algorithmic biases to security vulnerabilities—without falling into the trap of mechanically applying criteria designed for drug trials. Moreover, developers are developing greater ethical literacy, understanding that participant protection is not an obstacle but rather a prerequisite for the legitimacy of their work.⁽⁴⁾

As these specialized structures mature, the responsibility for preparing robust ethics applications falls on research teams. Doing this well not only speeds up the process of obtaining the necessary approval but also builds a foundation of trust with the institution that will be valuable throughout the project's lifespan. A committee that has had positive prior experience with a research team is more likely to approve future applications expeditiously, creating a virtuous cycle of collaboration. Conversely, a team that gains a reputation for superficiality or ethical opacity will face increasing scrutiny in future proposals.

If we have successfully navigated the ethical approval process, if we have demonstrated that our software respects participants' autonomy, minimizes risks, and promises proportionate benefits, we are ultimately in a position to begin data collection and clinical evaluation. However, a troubling question then arises: how do we know that the results we obtain in this study, however well-designed it may be, will reflect what will actually happen when the software is used outside the controlled research environment? How do we distinguish between efficacy demonstrated under ideal conditions and real-world effectiveness in everyday clinical practice, with all its noise, imperfections, and contextual variations?

REFERENCES

- Schulz KF, Altman DG, Moher D; CONSORT Group. CONSORT 2010 statement: updated guidelines for reporting parallel group randomized trials. *Ann Intern Med.* 2010;152(11):726-32. <https://doi.org/10.7326/0003-4819-152-11-201006010-00232>
- International Medical Device Regulators Forum. Software as a medical device (SaMD): key definitions. IMDRF/ SaMD WG/N10FINAL:2013. IMDRF; Dec 18, 2013. Available at: <https://www.imdrf.org/documents/software-medical-device-samd-key-definitions>
- U.S. Food and Drug Administration. Artificial intelligence/machine learning (AI/ML)-based software as a medical device (SaMD) action plan. Silver Spring (MD): FDA; Jan. 2021 [cited May 14, 2025]. Available at: <https://www.fda.gov/media/145022/download>
- Spain. Royal Decree 1090/2015, of December 4, regulating clinical trials with medicinal products, Ethics Committees for Research with Medicinal Products, and the Spanish Registry of Clinical Trials. BOE. Dec 24, 2015;(307):121923-64. Available at: <https://www.boe.es/eli/es/rd/2015/12/04/1090>
- World Medical Association. WMA Declaration of Helsinki – ethical principles for medical research involving human participants [Internet]. Ferney-Voltaire: WMA; Oct. 19, 2024 [cited May 14, 2025]. Available at: <https://www.wma.net/policies-post/wma-declaration-of-helsinki/>
- The Nuremberg Code. In: *Trials of War Criminals before the Nuremberg Military Tribunals under Control Council Law No. 10*. Vol. 2. Washington, DC: US Government Printing Office; 1949. p. 181–2. Available at: <https://history.nih.gov/display/history/Nuremberg+Code>
- National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. *The Belmont Report: Ethical Principles and Guidelines for the Protection of Human Subjects of Research*. Washington, DC: U.S. Department of Health, Education, and Welfare; 1979. Available at: <https://www.hhs.gov/ohrp/regulations-and-policy/belmont-report/index.html>