

The Value of Monitoring Clinical Decision Support Interventions

Eileen Yoshida¹ Shirley Fei² Karen Bavuso³ Charles Lagor⁴ Saverio Maviglia^{3,5,6}

¹ EBSCO Health, EBSCO Information Services, Ipswich, Massachusetts, United States

² Cambridge Health Alliance, Somerville, Massachusetts, United States

³ Smedy, Cambridge, Massachusetts, United States

⁴ Precision Health AI, New York, New York, United States

⁵ Department of Medicine, Brigham and Women's Hospital, Boston, Massachusetts, United States

⁶ Harvard Medical School, Boston, Massachusetts, United States

Address for correspondence Eileen Yoshida, BScPhm, MBA, EBSCO Health, EBSCO Information Services, 10 Estes Street, Ipswich, MA 01938, United States (e-mail: eyoshida@ebSCO.com).

Appl Clin Inform 2018;9:163–173.

Abstract

Background Well-functioning clinical decision support (CDS) can facilitate provider workflow, improve patient care, promote better outcomes, and reduce costs. However, poorly functioning CDS may lead to alert fatigue, cause providers to ignore important CDS interventions, and increase provider dissatisfaction.

Objective The purpose of this article is to describe one institution's experience in implementing a program to create and maintain properly functioning CDS by systematically monitoring CDS firing rates and patterns.

Methods Four types of CDS monitoring activities were implemented as part of the CDS lifecycle. One type of monitoring occurs prior to releasing active CDS, while the other types occur at different points after CDS activation.

Results Two hundred and forty-eight CDS interventions were monitored over a 2-year period. The rate of detecting a malfunction or significant opportunity for improvement was 37% during preactivation and 18% during immediate postactivation monitoring. Monitoring also informed the process of responding to user feedback about alerts. Finally, an automated alert detection tool identified 128 instances of alert pattern change over the same period. A subset of cases was evaluated by knowledge engineers to identify true and false positives, the results of which were used to optimize the tool's pattern detection algorithms.

Conclusion CDS monitoring can identify malfunctions and/or significant improvement opportunities even after careful design and robust testing. CDS monitoring provides information when responding to user feedback. Ongoing, continuous, and automated monitoring can detect malfunctions in real time, before users report problems. Therefore, CDS monitoring should be part of any systematic program of implementing and maintaining CDS.

Keywords

- ▶ clinical decision support
- ▶ CDS monitoring
- ▶ automated pattern detection (from MeSH)
- ▶ knowledge management
- ▶ alert fatigue

received
October 31, 2017
accepted after revision
January 8, 2018

Copyright © 2018 Schattauer

DOI <https://doi.org/10.1055/s-0038-1632397>.
ISSN 1869-0327.

Background and Significance

When properly designed and implemented, clinical decision support (CDS) has the potential to facilitate provider workflow, improve patient care, potentiate better outcomes, and reduce costs.¹⁻⁴ However, sometimes CDS is suboptimally deployed by health care organizations, including those that have implemented commercial electronic health record (EHR) systems.⁵ CDS may fire inappropriately for the wrong patient or provider, at the wrong time in the workflow, or may not fire when it should.⁶ Poorly performing CDS can negatively affect patient safety.⁶ Incorrect or overalerting may increase the risk of alert fatigue and cause providers to ignore important CDS interventions, leading to provider dissatisfaction and high override rates.^{2,7}

Many factors figure into the success of a CDS program, including effective and representative governance, robust knowledge management policies and processes, focus on workflow, and data-driven assessments.^{8,9} Even at institutions which have robust and comprehensive CDS programs, CDS malfunctions are common and often not detected before they reach end users.⁶ CDS malfunctions have been defined as events where CDS interventions do not work as designed or expected.^{6,8} A recent study by Wright et al defines a taxonomy for CDS malfunctions based on the answers to four questions: (1) What caused the malfunction?; (2) How was the malfunction discovered?; (3) When did the malfunction start?; and (4) What was the effect of the malfunction on rule firing? The authors conclude that a robust testing and monitoring program is essential to ensure more reliable CDS.⁸

At Partners HealthCare, the Clinical Informatics team led the development and maintenance of computerized CDS interventions utilized within a commercial enterprise EHR. These interventions consist of various types of CDS, targeting a variety of recipients at different points within the workflow. CDS development occurs in ongoing phases from request to implementation, known as the “CDS lifecycle.” CDS testing and monitoring are two phases of the CDS lifecycle which are intended to prevent and detect malfunctions. While testing typically is the phase in which malfunctions are identified and corrected, monitoring CDS in the production environment while in a silent status (invisible to the end user), also provides an opportunity to identify malfunctions prior to activation. Ongoing monitoring after activation can ensure that CDS continues to function correctly as the underlying EHR configuration and CDS dependencies change or become obsolete.⁶ Ultimately, the goal of a robust CDS monitoring program is to systematically, efficiently, and consistently review CDS fire rates and patterns to guide decisions about when to activate, revise, or retire CDS interventions.

Objective

The objective of this article is to describe our experience with implementing a program to create and maintain properly functioning CDS by systematically monitoring firing rates and patterns. We define in operational terms both pre- and

postactivation CDS monitoring, including a continuous automated monitoring system. Illustrative examples of each type of monitoring are provided, categorized by a CDS malfunction taxonomy.⁸ The purpose and place within the larger CDS lifecycle of each of these types of monitoring is described. Data about the effectiveness of these types of monitoring activities are also presented.

Methods

Study Setting and Electronic Health Record System

Partners HealthCare is a large integrated delivery system in Boston, Massachusetts. The system includes two large academic medical centers as well as several community hospitals, primary care and specialty physicians, a managed care organization, specialty facilities, community health centers, and other health-related entities. In 2012, the organization decided to transition to a single instance of a vendor EHR in a multiyear, phased roll-out. The EHR in use is Epic (Epic Systems, Verona, Wisconsin, United States) Version 2014 (later upgraded to Version 2015 in October 2016). Compared with the prior EHR systems at Partners HealthCare, Epic provided new functionalities to monitor the utilization of both silent and active CDS interventions. These functionalities were extended to form the basis of a formal program of CDS monitoring aimed at improving the quality of released CDS.

Clinical Decision Support Interventions

This study focuses on the monitoring of all point of care alerts/reminders targeted to health care providers who interact with the EHR while providing care to patients in all care settings. These CDS interventions are essentially production rules which are triggered when prespecified logical criteria are satisfied, in which case one or more recommended actions are suggested and facilitated.⁹ Focusing on preventing alert fatigue, we chose first to monitor alerts and reminders which interrupt the provider workflow (compared with those which do not interrupt user workflow).

Clinical Decision Support Lifecycle

At our institution, CDS within Epic is developed and maintained following a formalized CDS lifecycle composed of successive phases of activities, each performed by a specialized team or set of actors, using specific tools, policies, and processes. Over the first 2 years of the EHR roll-out, the lifecycle matured to include eight distinct phases as depicted in **Fig. 1**. Briefly, the lifecycle begins with a request, which can be submitted by leadership groups or individuals. Requests are formally approved by the CDS Committee, a governing body which consists of representatives from each institution within our organization and different clinical disciplines. Approved requests are then prioritized by a subset of CDS Committee members. Clinical knowledge engineers collaborate with clinical subject matter experts to design the CDS. During the build phase, the design specification is given to the application coordinator to build the CDS using the editors and tools supplied by the EHR vendor. Formal test scripts are executed by the knowledge

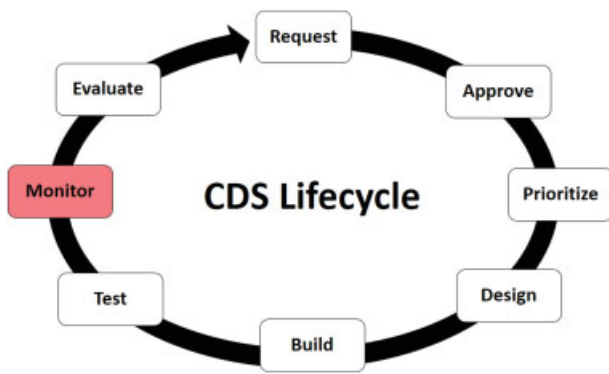


Fig. 1 Clinical decision support (CDS) lifecycle.

engineers during the testing phase. Once CDS passes testing, it moves to the monitoring phase, which is the observation, measurement, and tracking of the rate and pattern of CDS firing. Evaluation is the final phase of the lifecycle, which results in requests for new or revised CDS, or to retire CDS. The lifecycle repeats indefinitely. While monitoring is about ensuring correctness, evaluation is about measuring the effectiveness of CDS interventions. The focus of this article is on the monitoring phase of the lifecycle.

Clinical Decision Support Monitoring Program Description

CDS monitoring examines when and why alerts fire (or do not). It focuses on what happens until the time an alert is triggered to determine if it is firing as intended, to detect malfunctions, and to minimize false positive and false negative alerts. Using semiautomated extract-transform-load (ETL) processes, data are moved from the EHR's production environment into a proprietary web-based CDS monitoring portal where knowledge engineers review and analyze the monitoring data. CDS fire rates are monitored because they are readily available, observable, and measurable. A change in a fire rate or pattern can signal the need for more intensive investigation to determine the underlying cause and guide appropriate steps for remediation.

There are four types of monitoring within the CDS monitoring phase. The first type is *PreActivation Monitoring*. During preactivation monitoring, CDS is released for at least 2 weeks in silent mode (not visible to users) prior to activation. If the firing rate and pattern during this period is significantly different (higher or lower) than what would be expected based on the volume of patients seen with the targeted condition, or if selected chart reviews surface false positive or false negative firings, then a more thorough analysis is initiated, otherwise, the CDS moves to postactivation monitoring. Two weeks is the minimum duration of preactivation monitoring because the firing pattern tends to vary with the day of the week (especially weekdays compared with weekends), so at least two consecutive weeks are monitored to confirm stable firing patterns. This timeframe also provides a reasonable volume of data and time to complete random chart reviews of actual silent firings. The target number of charts to review is predetermined based on

statistical power calculations of how many consecutive, randomly selected, error-free firings need to be observed to conclude that the false positive rate is below a desired threshold, as shown in **Appendix**. In addition to detecting malfunctions missed during the testing phase, preactivation monitoring also accomplishes the following important goals:

1. Establishes a baseline alert firing profile—detection of statistically significant deviance from this pattern is the basis of automated postactivation monitoring (described later).
2. Allows measurement of the baseline or inherent compliance rate for the alert. This measurement is technically part of the evaluation phase of the CDS lifecycle, but is made possible by silent preactivation monitoring. The providers who would have seen an alert or reminder but did not because the intervention was in silent mode constitute a convenient control group to estimate how often the intended action is performed regardless of the intervention. This baseline compliance rate should be discounted from the measured compliance rate when the CDS is active to measure the true effectiveness of the intervention.
3. Identifies potential subgroups of providers who may serve as pilot users when the CDS is turned on—this is particularly useful for potentially “noisy” or controversial alerts or reminders, to reduce the risk of activating CDS which is unacceptable to end users, even if technically correct.

The second type of monitoring is *PostActivation Monitoring*. Here, CDS is monitored for at least 2 weeks following activation. Again, if the firing rate shows unexpected results or if selected chart reviews surface false positive or false negative firings, then a more intensive analysis occurs, otherwise, the CDS is deemed stable. The goal of postactivation monitoring is to confirm that the CDS and all its dependencies have been successfully migrated to the production environment in the active state. Migration between environments has been identified as causing a significant number of malfunctions.⁸

The third type is *Ad Hoc Monitoring*. For stable active CDS, this monitoring occurs as needed, usually when a user reports a problem with an alert. The reason for an unexpected firing can often be ascertained just from reviewing the firing frequency and pattern, much like an electrocardiogram (EKG) can be used to diagnose a patient-reported symptom such as “palpitations.” Other reasons to perform ad hoc monitoring are during significant EHR events, like go-lives or system software upgrades.

The final type is *Continuous Automated Monitoring*. This type of monitoring occurs continuously and indefinitely after postactivation monitoring ends, and utilizes automated algorithms described in the “Clinical Decision Support Monitoring Tools” section below. Sometimes the deviations are expected (“false positive”), such as might occur with a system upgrade, or the addition of a new practice site, or closing of a preexisting site. In other cases, the change is not expected but caused by an inadvertent change in an upstream or downstream dependency, such as the addition

or retirement of a diagnosis, procedure, or medication code. These “true positive” signals are malfunctions that should be corrected promptly.

Clinical Decision Support Monitoring Tools

To monitor CDS interventions released into the EHR, we implemented a CDS data analytics infrastructure consisting of a relational SQL Server database (Microsoft, Redmond, Washington, United States) Version 2014, and Visual Studio reports (Microsoft) Version 2015.

The database reconciles daily CDS alert data feeds from the EHR with data from a CDS documentation system implemented in JIRA (Atlassian, San Francisco, California, United States) version 7.3.2. It contains the following data elements: alert instance identification number (ID), date and time of alert instance, name and ID of CDS intervention, type of CDS intervention (interruptive, noninterruptive, or CDS that is not seen by providers), CDS monitoring status (silent monitoring that is not seen by providers versus active release), patient ID, provider ID, location of service, and provider interaction with an interruptive alert (so-called “follow-up action”).

Reports generated from the alert data are presented via an intranet web site accessible by knowledge engineers and other authorized users. The main graph for CDS firing rates shows plots of daily alerted patient counts (*y*-axis) against days (*x*-axis). The daily alerted patient count helps knowledge engineers assess whether a CDS intervention is performing as designed. Another graph for CDS firing rates plots daily alert counts (*y*-axis) against days (*x*-axis). The daily alert counts assess the alert burden of a particular CDS intervention on providers. The data analytics infrastructure also contains graphs for the CDS evaluation phase, which help assess provider compliance with CDS interventions.

All graphs can be configured by users to show selected subsets of CDS interventions, time periods, and locations. In addition, the data analytics infrastructure contains a report that contrasts alerted patient counts over a selected time period of one site versus another, as well as a report that contrasts alerted patient counts from one time period versus another for a selected site. These reports are particularly helpful when a site goes live with the EHR.

Continuous automated monitoring is achieved by an algorithm implemented in R version 3.3.2.¹⁰ The algorithm processes the alert data of individual CDS interventions for a given timeframe. It fits an exponential smoothing predictive model using the Holt Winters function, which is part of the “stats” package in R. The function calculates a parameter α , which ranges from 0 to 1. The closer the α value is to 1, the more each model relies on recent observations to make forecasts of future values. Thus, high α values are more likely to occur in data that displays changing trends. The algorithm flags those CDS interventions whose alert data sample has α values greater than a given threshold. Under the current threshold setting of 0.3, the algorithm has a sensitivity of 75.0%, a specificity of 96.2%, a positive predictive value of 30.0%, and a negative predictive value of 99.4%, when validated against a set of 380 graphs (adjudicated by S.M. and

C.L.). The algorithm runs weekly and its results are presented as a dynamic report in the data analytics intranet site.

Results

The four monitoring processes described in the “Clinical Decision Support Monitoring Program Description” section above were implemented stepwise beginning shortly after our first go-live in 2015. At the time of submission of this article, 2 years later, we have processed 248 CDS interventions, either new or revisions, through pre- and postactivation monitoring. Of these, 92 (37%) were noted during preactivation monitoring to have a malfunction or significant opportunity for improving the sensitivity or specificity of the triggering logic. Immediately after activation, 45 (18%) CDS interventions were observed to have problems during postactivation monitoring—these were almost always caused by data or configuration issues related to the migration from the testing environment to the production environment, rather than malfunctions that required revising the CDS. Many of these issues were not appreciated prior to implementing postactivation monitoring, and have been effectively eliminated by changing the process of how CDS-related assets are migrated across environments.

Additional CDS interventions have undergone ad hoc monitoring, either subsequent to user feedback (this occurs for approximately 3–5% of released CDS), or to a system event (such as when we upgraded to the 2015 version of the Epic software). Continuous automated monitoring now occurs with all active CDS, regardless of when it was activated (~570 CDS interventions as of June 2017).

The following sections provide illustrative examples of each type of monitoring, categorized according to a CDS malfunction taxonomy.⁸

Preactivation Monitoring

One example of preactivation monitoring which resulted in revision of CDS prior to release is CDS that reminds providers to add a principal problem to the problem list to address a Meaningful Use (MU) hospital measure (→ Fig. 2). The original version of this CDS was only restricted to hospital encounters. In silent mode, we noticed the firing rate was high (> 1,000 patients/day). Chart reviews revealed that the CDS fired frequently in emergency departments and perioperative areas, which prompted us to further restrict the CDS to admitted inpatients, excluding procedural areas. The fix for this CDS resulted in ~80% decreased firing rate and more targeted interventions due to increased specificity without compromised sensitivity (→ Table 1: CDS example 1A).

A second example of successful preactivation monitoring is CDS that recommends an EKG when a patient is on two or more QTc prolonging medications (→ Fig. 3). Our monitoring data analysis and chart review in silent mode surfaced a malfunction in the complex set of medications for which the alert was supposed to fire, which led to a significant modification to the medication value set build. As with the prior example, the fix eliminated false positive alert instances not identified during testing (→ Table 1: CDS example 2).

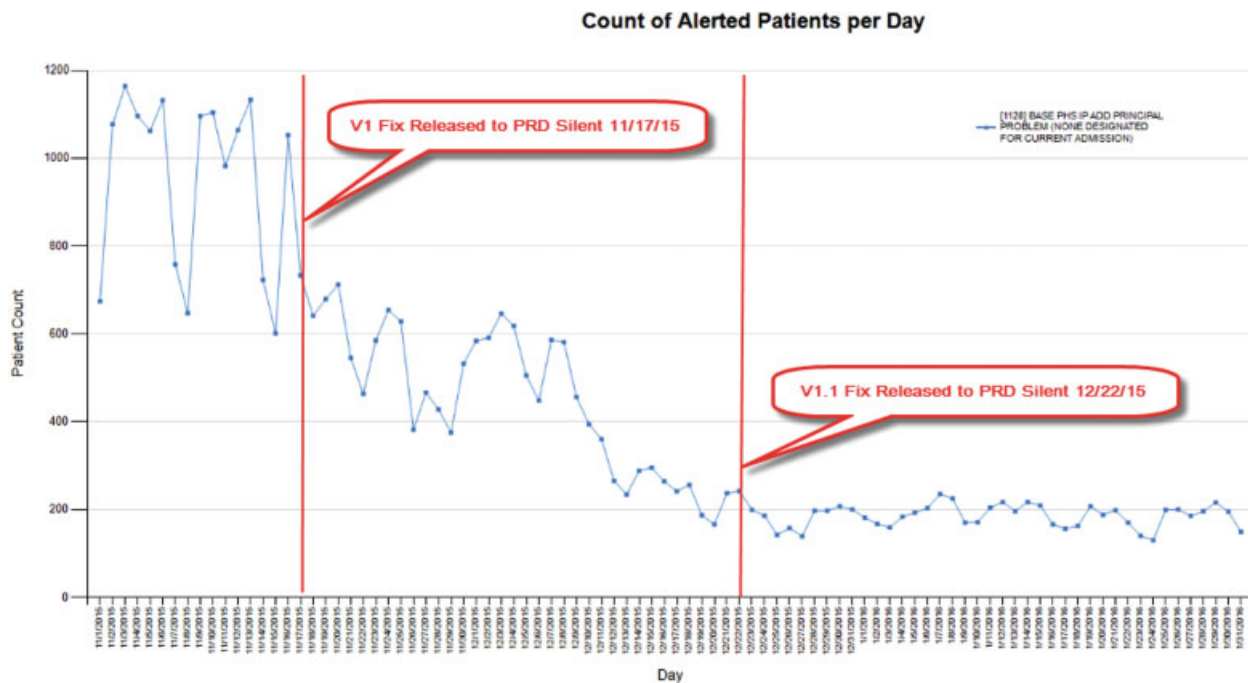


Fig. 2 Preactivation monitoring: clinical decision support that reminds physicians to add a principal problem.

Postactivation Monitoring

The same CDS that reminds providers to add a principal problem to the problem list also illustrates the value of postactivation monitoring. Soon after this CDS was released to users in production, we saw an increased firing which correlated temporally with our second major go-live, as expected. However, because of postactivation monitoring chart reviews, we further fine-tuned the triggering criteria by excluding emergency departments and urgent care specialty logins, and also by excluding scenarios where a principal problem had already been resolved. As seen in [Fig. 4](#), these changes eliminated an additional of ~50% of firings. The final fire rate of the CDS stabilized at < 100 patients per day across the entire enterprise ([Table 1](#): CDS example 1B).

Ad Hoc Monitoring

One example of the value of ad hoc monitoring is CDS that informs nurses when a medication dose is given too close to a previous dose as documented on the medication administration record (MAR) ([Fig. 5](#)). In this example, a user reported that a student nurse was not alerted for this scenario. We quickly determined that although nursing students were included in the provider type of the CDS build, due to user configuration complexities, this CDS did not surface upon MAR administration for student nurses, nor did it surface for nurse preceptors upon cosigning their students' MAR administrations. We resolved this problem by using workflow restrictors rather than provider type restrictors. In this case, the update resulted in a slight increase in the overall firing, because false negatives were eliminated ([Table 1](#): CDS example 3).

A second example of successful ad hoc monitoring is CDS which reminds providers to order venous thromboembolism

(VTE) prophylaxis within 4 hours of admission ([Fig. 6](#)). False positive firings were found after we customized our codes of reasons for not prescribing VTE prophylaxis. Despite the fix, the monitoring data indicated no change in firing rate. It was only later, during ad hoc monitoring prompted by the v2015 upgrade in which Epic corrected how it computed procedure exclusion logic that we noticed a large drop in firing because the previous fix began to work properly. Since then, we further fine-tuned this CDS by adding exclusion procedure look back (which slightly increased firing) and excluding hospital outpatient departments (which slightly decreased firing), therefore, we observed a relatively unchanged rate of firings overall, but nonetheless increased both sensitivity and specificity ([Table 1](#): CDS example 4).

Continuous Automated Monitoring

An example of updating CDS due to continuous automated monitoring is CDS that provides guidance for ordering radiology procedures via a web service call to a third-party vendor ([Fig. 7](#)). Our change detection algorithm flagged an unusual pattern where the firing suddenly stopped. Upon investigation, it was determined that the vendor web service was down unexpectedly. To the end user, there was no warning or error message, only the failure to display an alert, which might have gone unnoticed (or even have been a welcome change). Without automated monitoring, the malfunction could have persisted indefinitely. Another drop in firing of this CDS (this time incomplete and transient) was detected on July 4th, but upon investigation this was deemed to be a false positive change, caused only because the volume of ordered procedures dropped significantly enough due to the holiday to trigger the automated change detector to flag the incident ([Table 1](#): CDS example 5).

Table 1 Examples of CDS malfunctions categorized by Wright's CDS malfunction taxonomy⁸

CDS examples	Type of monitoring	Cause	Mode of discovery	Initiation	Effect on rule firing
1A: CDS that reminds providers to add a principal problem to the problem list	Preactivation	Conceptualization error	Review of firing data	From its implementation in production	Rule fires in situations where it should not fire
1B: CDS that reminds providers to add a principal problem to the problem list	Postactivation	Conceptualization error	Review of firing data	After deployment into production	Rule fires in situations where it should not fire
2: CDS that recommends an EKG when a patient is on two or more QTc prolonging medications	Preactivation	Build error	Review of firing data	From its implementation in production	Rule fires in situations where it should not fire
3: CDS that informs nurses when a medication dose is given too close to a previous dose as documented on the medication administration record (MAR)	Ad hoc	Conceptualization error	Reported by an end user	After deployment into production	Rule does not fire in situations where it should
4: CDS which reminds providers to order venous thromboembolism (VTE) prophylaxis within 4 h of admission	Ad hoc	1. New code, concept or term introduced, but rule not updated 2. Defect in EHR software	1. Reported by an end user 2. Review of firing data	After deployment into production	Rule fires in situations where it should not fire
5: CDS that provides guidance for ordering radiology procedures via a web service call to a third-party vendor	Continuous automated	External service issue	Review of firing data (automatic process)	After deployment into production	Rule does not fire in situations where it should

Abbreviations: CDS, clinical decision support; EHR, electronic health record; EKG, electrocardiogram.

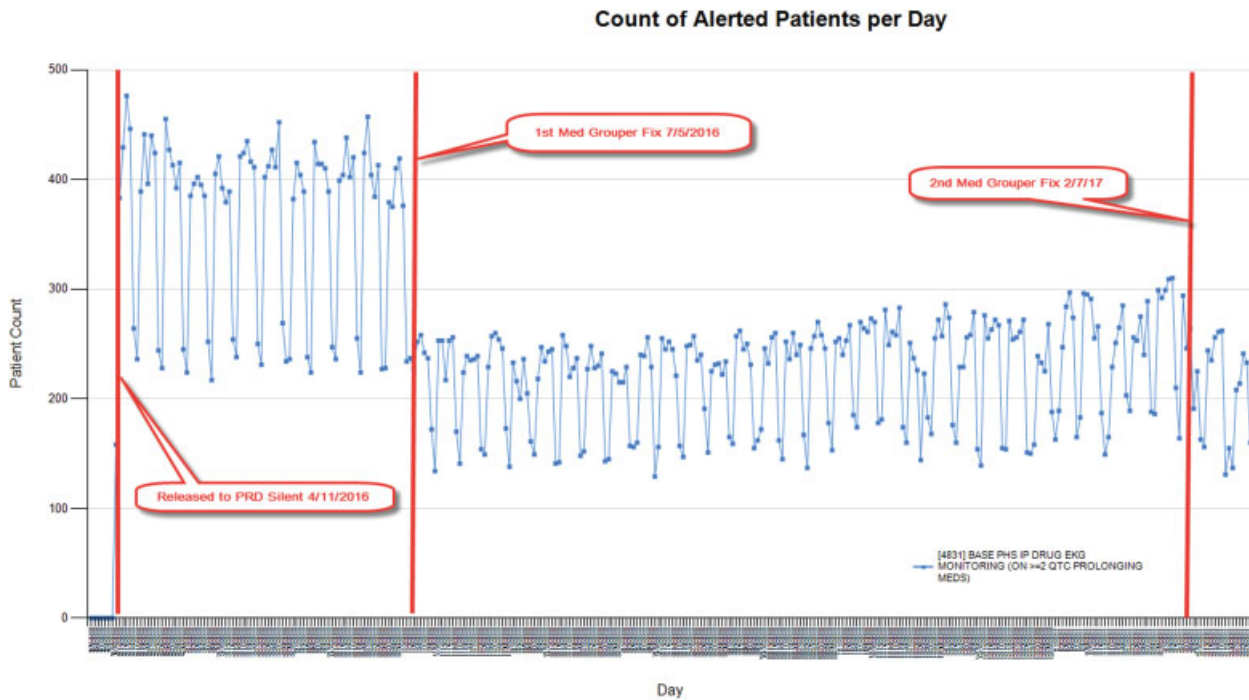


Fig. 3 Preactivation monitoring: clinical decision support that recommends electrocardiogram when patient on ≥ 2 QTc prolonging medications.

From November 2016 through September 2017, our tool detected a total of 128 instances of alert pattern change. Of the 97 instances that were manually reviewed by a knowledge engineer, 5 were true positives where the CDS had to be revised; 63 were true positives but no action was required; 24 were deemed false positives; and 5 are currently under investigation. These findings were used to increase the sensitivity and specificity of the detection tool itself.

Discussion

CDS monitoring is often included as a success factor for CDS programs.^{6,11-13} In this article, we describe our experience in implementing a comprehensive CDS monitoring program, including both pre- and postactivation monitoring activities. To our knowledge, this is the first report in the literature of such activities implemented in a systematic way. In addition,

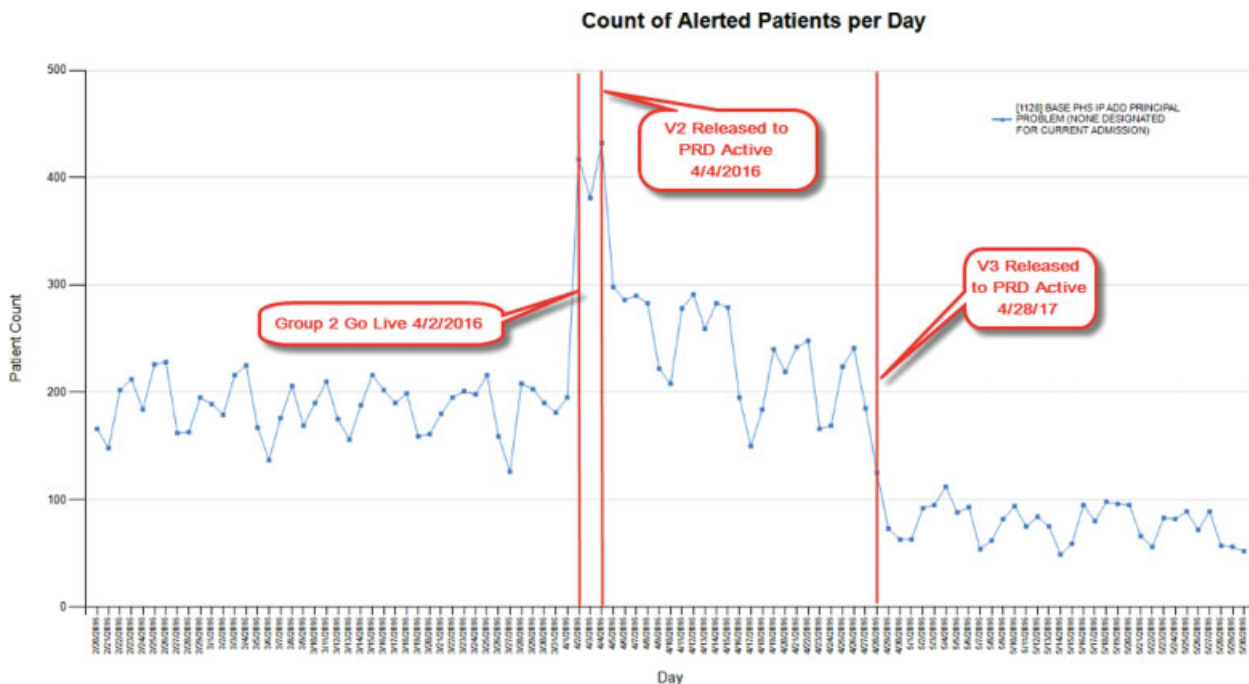


Fig. 4 Postactivation monitoring: clinical decision support that reminds physicians to add a principal problem.

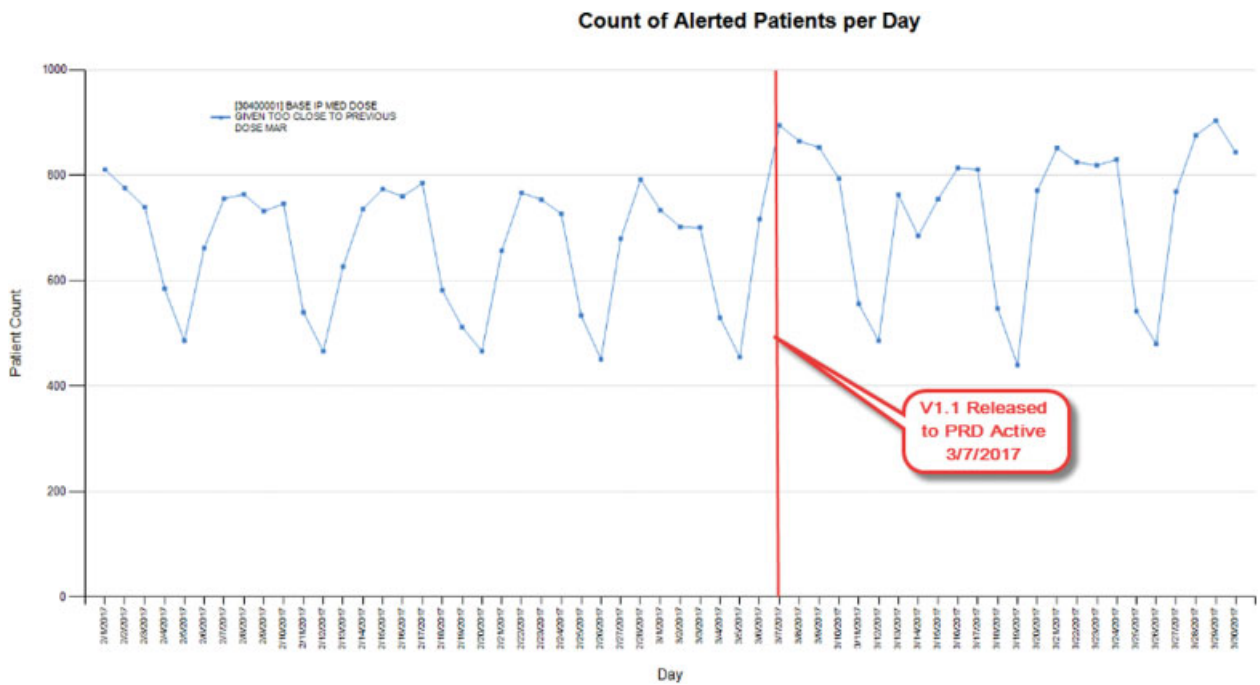


Fig. 5 Ad hoc monitoring: clinical decision support that informs nurses when a medication is given too close to a previous dose on medication administration record.

the use of preactivation monitoring activities to decrease the risk of CDS malfunction has not been previously reported. Two previously reported studies have described the challenges in trying to determine the cause of nonuser reported malfunctions, particularly when the cause occurred many months before detection.^{5,6} In both cases, the authors proposed the development of a real-time dashboard such as our continuous automated monitoring dashboard. Continuous

automated monitoring provides an additional/alternative strategy to testing in the live environment after any CDS-related change and after major EHR software upgrades as recommended by others, particularly when the number of CDS is large and resources are limited.⁶

Campbell et al identified nine types of clinical unintended adverse consequences resulting from computerized physician order entry implementation (CPOE).¹⁴ The authors

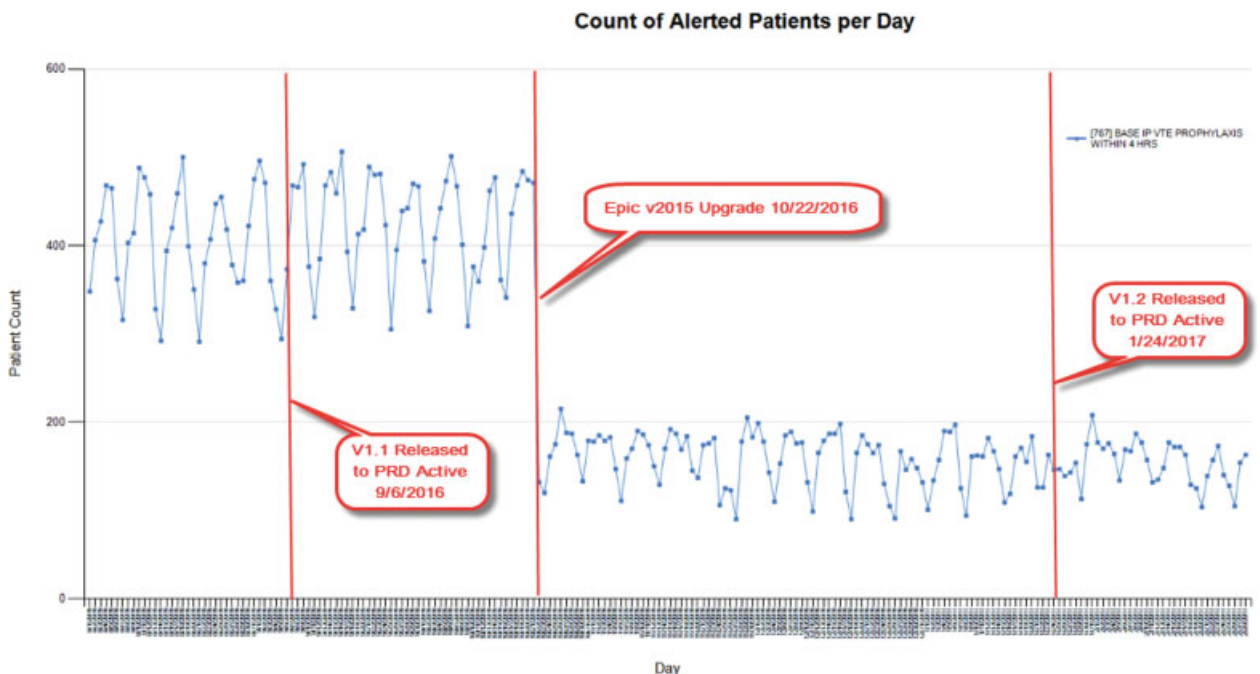


Fig. 6 Ad hoc monitoring: clinical decision support that recommends venous thromboembolism prophylaxis within 4 hours of admission.

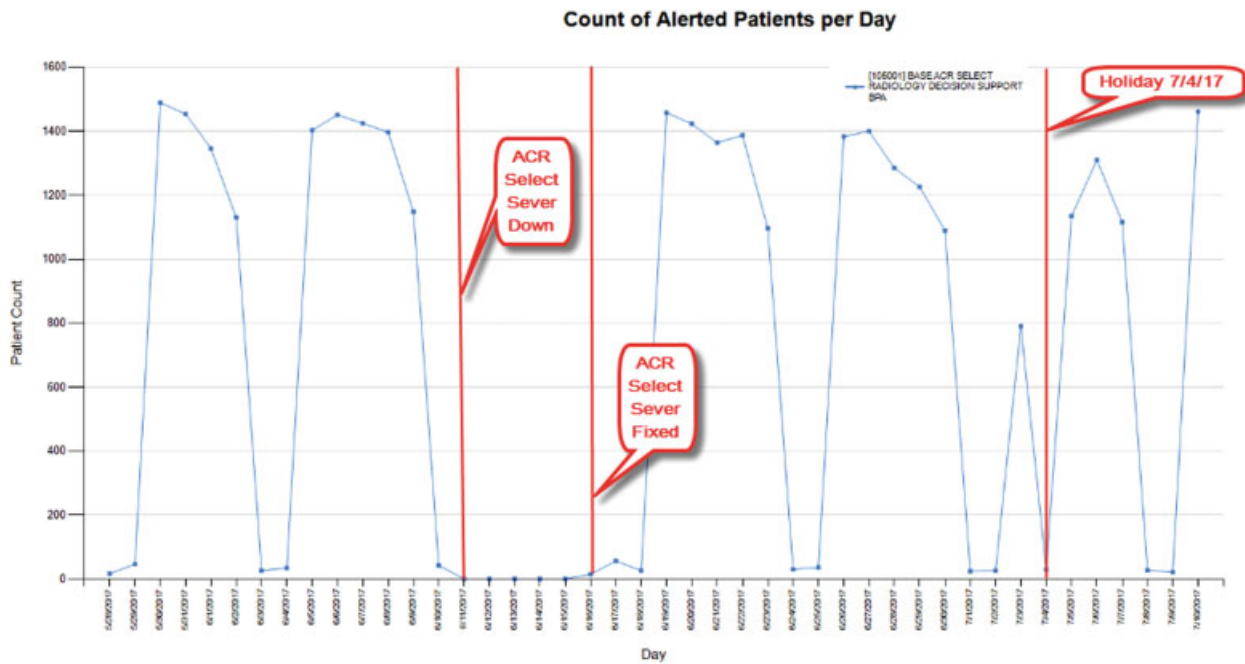


Fig. 7 Continuous automated monitoring: clinical decision support that calls a third-party vendor via web service to provide radiology decision support.

concluded that CDS features introduced many of these unintended consequences. Monitoring can help reduce some unintended consequences related to CPOE by identifying ways to improve the alert (via pre- and postactivation monitoring) and by detecting when alerts stop firing appropriately due to “never-ending system demands” (via ad hoc and continuous automated monitoring).

Monitoring is an important phase of the CDS lifecycle because it identifies many errors that would not have otherwise been detected prior to activation to the end user. We identified malfunctions or significant opportunities for improvement in 37% of CDS reviewed during preactivation monitoring and in 18% of CDS reviewed during postactivation monitoring. This is significant because at our institution, CDS must pass formal integrated testing before monitoring begins.

With preactivation monitoring, CDS malfunctions are resolved before users see them, and with postactivation and continuous automated monitoring, malfunctions are detected and resolved quickly, before users might notice or report a problem. Correcting CDS malfunctions before they are noticed is an important way to promote faith in the EHR. Relying on users to report malfunctions potentially undermines their trust of CDS, which can foster general discontent and degrade overall acceptance of the entire EHR implementation.

CDS monitoring has also helped improve the design of CDS going forward, by increased recognition of the types of malfunctions that can be missed by formal testing, and by greater understanding of how sensitivity and specificity of logical criteria used to trigger CDS can be increased. From this experience, including the examples described in this article, a great deal has been learned about how to: (1)

optimally include or exclude patient populations (by encounter, department, specialty, or admission/discharge/transfer (ADT) status); (2) appropriately suppress CDS to minimize false positives; (3) scope CDS so that all provider types and even atypical workflows are considered during the request phase; (4) track upstream and downstream CDS dependencies; and much more. We added these lessons to our design guide as we learned them. The design guide is used to train new knowledge engineers, so they quickly become proficient at designing high-quality CDS and avoid recommitting errors made by their predecessors, including those who have left the organization. In this sense, CDS monitoring is one way to actualize a learning health system.¹⁵

Robust testing and monitoring are important phases of the CDS lifecycle that promote the release of correct CDS, that is, CDS which fires when it should, and does not fire when it should not. But even infinite testing and monitoring will not guarantee effective CDS, which is CDS that results in improved outcomes or decreased costs. While CDS monitoring focuses on everything *up to* the triggering of CDS, CDS evaluation focuses on everything that happens *after* the alert fires—was the alert overridden, was the recommended order signed and implemented regardless of whether the alert was overridden, and did the clinical outcome change for the better. Nevertheless, CDS monitoring can promote CDS effectiveness indirectly by reducing false positive and false negative alerting through more sensitive and specific triggering criteria; this in turn can help prevent alert fatigue, which should increase compliance with all alerts. Further, process and outcomes data collected during preactivation monitoring provide the control or baseline effectiveness with which postactivation should be compared. Therefore, robust CDS

monitoring is a prerequisite for robust CDS evaluation, the last phase of the CDS lifecycle.

Limitations

There are some limitations with our approach. The findings reported here reflect the experience of one organization focusing on one type of CDS; the implementation should be replicated at other sites using the same type of CDS to confirm our findings. We believe the benefits of routine monitoring can be achieved with other types of CDS and additional studies should be conducted to confirm our hypothesis.

In addition, the introduction of the CDS lifecycle processes and tools was done while the organization was in the midst of a multiyear EHR implementation; finite resources were allocated to configuring the system, rolling out the software to new sites, and supporting novice users, at the same time that CDS was being implemented and maintained. When there is fierce competition for resources, the relative benefit of CDS monitoring must be compared with that of alternative activities. Ideally, resources for CDS monitoring should be anticipated and allocated from the beginning of an implementation project.

We were able to implement preactivation monitoring because our EHR vendor had the capability to release CDS to production in silent mode. We recognize that not all EHR vendors have this capability. We developed our own CDS data analytics and monitoring portal due to limitations in functionality with the vendor-supplied tools. We recognize that not all institutions may have the resources to do this.

And finally, at our site dedicated knowledge engineers and clinical informaticians participate in the design and implementation of CDS. These resources typically have a clinical background and are also formally trained and/or have experience in knowledge management and informatics and may have additional expertise in designing and implementing CDS at scale. Such highly skilled resources may not be available in all health care delivery systems.

Conclusion

CDS has the potential to facilitate provider workflow, improve patient care, promote better outcomes, and reduce costs. However, it is important to develop and maintain CDS that fires for the right patient and the right provider at the appropriate time in the workflow. Implementation of robust CDS monitoring activities can help achieve this and is a valuable addition to an organization's CDS lifecycle. Monitoring can be done before CDS is released active to end users, as well as postactivation. Automated alert detection can help proactively detect potential malfunctions based on historical patterns of firing.

Clinical Relevance Statement

CDS is a valuable tool leveraged within an organization's EHR that can improve patient safety and quality of care. CDS monitoring is a critical step to identify errors after formal

testing and prior to activation in the production environment. The monitoring processes and experiences described in this article can assist organizations in the development and implementation of a formal CDS monitoring process to support a successful CDS program.

Multiple Choice Questions

1. "Identifying potential subgroups of providers to serve as pilot users" is one of the goals of which type of CDS monitoring?
 - a. Preactivation monitoring.
 - b. Postactivation monitoring.
 - c. Ad hoc monitoring.
 - d. Continuous automatic monitoring.
2. What is the final phase of the CDS lifecycle, which results in requests for new/revised CDS, or to retire CDS, before the CDS lifecycle repeats indefinitely?
 - a. Request.
 - b. Design.
 - c. Monitoring.
 - d. Evaluation.

Correct Answer: The correct answer is "a, preactivation monitoring." Preactivation monitoring is the only phase where the CDS is being monitored in silent mode. This is the perfect time to determine if a CDS, particularly a potentially "noisy" or controversial one, should be released to selected pilot users first, and who might serve as the pilot users. It is important to reduce the risk of activating CDS which is unacceptable to end users, even if it is technically correct.

Correct Answer: The correct answer is "d, evaluation." Request and Design are the earlier phases of a CDS lifecycle. After CDS is released to users, the Monitoring phase focuses on ensuring its correctness. After the Monitoring phase, the Evaluation phase focuses on measuring the effectiveness of CDS interventions.

Note

All the authors were affiliated with Clinical Informatics, Partners HealthCare System Inc., Boston, Massachusetts, United States at the time of the study.

Protection of Human and Animal Subjects

Neither humans nor animal subjects were included in this project.

Conflict of Interest

None.

Acknowledgments

We thank Mahesh Shanmugam and Mark Twelves at Partners HealthCare for their technical contributions to the CDS data analytics infrastructure, as well as Dave Evans at Partners HealthCare for providing the data feeds from the EHR.

References

- 1 Mishuris RG, Linder JA, Bates DW, Bitton A. Using electronic health record clinical decision support is associated with improved quality of care. *Am J Manag Care* 2014;20(10):e445–e452
- 2 Beeler PE, Bates DW, Hug BL. Clinical decision support systems. *Swiss Med Wkly* 2014;144:w14073
- 3 Kassakian SZ, Yackel TR, Deloughery T, Dorr DA. Clinical decision support reduces overuse of red blood cell transfusions: interrupted time series analysis. *Am J Med* 2016;129(06):636.e13–636.e20
- 4 Procop GW, Yerian LM, Wyllie R, Harrison AM, Kottke-Marchant K. Duplicate laboratory test reduction using a clinical decision support tool. *Am J Clin Pathol* 2014;141(05):718–723
- 5 Kassakian SZ, Yackel TR, Gorman PN, Dorr DA. Clinical decisions support malfunctions in a commercial electronic health record. *Appl Clin Inform* 2017;8(03):910–923
- 6 Wright A, Hickman TT, McEvoy D, et al. Analysis of clinical decision support system malfunctions: a case series and survey. *J Am Med Inform Assoc* 2016;23(06):1068–1076
- 7 McCoy AB, Thomas EJ, Krousel-Wood M, Sittig DF. Clinical decision support alert appropriateness: a review and proposal for improvement. *Ochsner J* 2014;14(02):195–202
- 8 Wright A, Ai A, Ash J, et al. Clinical decision support alert malfunctions: analysis and empirically derived taxonomy. *J Am Med Inform Assoc* 2017. Doi: 10.1093/jamia/ocx106
- 9 Wright A, Sittig DF, Ash JS, et al. Governance for clinical decision support: case studies and recommended practices from leading institutions. *J Am Med Inform Assoc* 2011;18(02):187–194
- 10 R core team. R: A language and environment for statistical computing. [Internet]. R Foundation for Statistical Computing, Vienna, Austria; 2013. Available at: <http://www.r-project.org/>
- 11 Health Care IT Advisor. Enhancing CDS Effectiveness: Monitoring Alert Performance at Stanford Keys to Success; 2017:17–19
- 12 Ash JS, Sittig DF, Guappone KP, et al. Recommended practices for computerized clinical decision support and knowledge management in community settings: a qualitative study. *BMC Med Inform Decis Mak* 2012;12(01):6
- 13 Wright A, Ash JS, Erickson JL, et al. A qualitative study of the activities performed by people involved in clinical decision support: recommended practices for success. *J Am Med Inform Assoc* 2014;21(03):464–472
- 14 Campbell EM, Sittig DF, Ash JS, Guappone KP, Dykstra RH. Types of unintended consequences related to computerized provider order entry. *J Am Med Inform Assoc* 2006;13(05):547–556
- 15 Institute of Medicine. The path to continuously learning health care. Vol. 29, Issues in Science and Technology. Washington, DC: National Academies Press; 2013:18

Appendix : Statistical Model to Guide Chart Reviews (Partial Sample)

No. of consecutive error-free chart reviews	Confidence that the false positive rate is less than:				
	1%	5%	10%	15%	20%
5	4.9	22.6	41.0	55.6	67.2
10	9.6	40.1	65.1	80.3	89.3
15	14.0	53.7	79.4	91.3	96.5
20	18.2	64.2	87.8	96.1	98.8
25	22.2	72.3	92.8	98.3	99.6
30	26.0	78.5	95.8	99.2	99.9
35	29.7	83.4	97.5	99.7	100.0
40	33.1	87.1	98.5	99.8	100.0
45	36.4	90.1	99.1	99.9	100.0
50	39.5	92.3	99.5	100.0	100.0
55	42.5	94.0	99.7	100.0	100.0
60	45.3	95.4	99.8	100.0	100.0
65	48.0	96.4	99.9	100.0	100.0
70	50.5	97.2	99.9	100.0	100.0
75	52.9	97.9	100.0	100.0	100.0