Assessing the Safety of Custom Web-Based Clinical Decision Support Systems in Electronic Health Records: A Case Study

Jeritt G. Thayer[1] Jeffrey M. Miller[1] Alexander G. Fiks[1,2] Linda Tague[3] Robert W. Grundmeier[1,2]

[1] Department of Biomedical and Health Informatics, Children's Hospital of Philadelphia, Philadelphia, Pennsylvania, United States
[2] Department of Pediatrics, Perelman School of Medicine at the University of Pennsylvania, Philadelphia, Pennsylvania, United States
[3] Department of Information Services, Children's Hospital of Philadelphia, Philadelphia, Pennsylvania, United States

Address for correspondence Jeritt G. Thayer, BA, Department of Biomedical and Health Informatics, Children's Hospital of Philadelphia, 2716 South Street, 15th Floor, Philadelphia, PA 19146, United States (e-mail: thayerj@email.chop.edu).

Appl Clin Inform 2019;10:237–246.

Abstract

Background With the widespread adoption of vendor-supplied electronic health record (EHR) systems, clinical decision support (CDS) customization efforts beyond those anticipated by the vendor may require the use of technologies external to the EHR such as web services. Pursuing such customizations, however, is not without risk. Validating the expected behavior of a customized CDS system in the high-volume, complex environment of the live EHR is a challenging problem.

Objective This article identifies technology failures that impacted clinical care related to web service-based advanced custom CDS systems embedded in the complex sociotechnical context of a production EHR.

Methods In an academic health system's primary care network, we performed an inventory of incidents between January 1, 2008 and December 31, 2016 related to a customized CDS system and performed a targeted review of changes in the CDS source code. Additional feedback on the root cause of individual incidents was obtained through interviews with members of the CDS project teams.

Results We identified five CDS malfunctions that impaired clinical workflow. The mechanisms for these failures are mapped to four characteristics of well-behaved applications: (1) system integrity; (2) data integrity; (3) reliability; and (4) scalability. Over the 9-year period, two malfunctions of the customized CDS significantly impaired clinical workflow for a total of 5 hours. Lesser impacts—loss of individual features with straightforward workarounds—arose from three malfunctions, which affected users on 53 days.

Discussion Advanced customization of EHRs for the purpose of CDS can present significant risks to clinical workflow.

Conclusion This case study highlights that advanced customization of CDS within a commercial EHR may support care for complex patient populations, but ongoing monitoring and support is required to ensure its safe use.

Keywords

- clinical decision support
- electronic health records and systems
- clinical information systems
- safety
- error management

received
November 29, 2018
accepted after revision
February 13, 2019

DOI https://doi.org/10.1055/s-0039-1683985.
ISSN 1869-0327.

Background and Significance

Some research suggests that clinical decision support (CDS) can improve clinical care.[1–5] CDS is featured in multiple stages of the Meaningful Use Recommendations, and has been incorporated into the majority of commercial electronic health record (EHR) systems; however, these systems are often aimed at medication ordering tasks that apply to the majority of patients such as drug–drug or drug–allergy interactions.[6,7] To realize the full potential of EHRs to enhance care and improve outcomes for more specific and complex patient populations, advanced customization of these systems by individual health organizations is often required.

Pursuing such customizations, however, is not without risk. Validating the expected behavior of a customized CDS system in the high-volume, complex environment of the live EHR is a challenging problem.[8,9] A growing body of evidence has shown that even native EHR configured CDS malfunctions are widespread and difficult to detect, and can lead to a variety of unintended consequences and safety issues.[8–14] EHR version upgrades, updates to clinical coding schemes, workflow changes, and ongoing configuration efforts may all disrupt the behavior of these interconnected systems.

With the widespread adoption of vendor-supplied EHR systems, CDS customization efforts beyond those anticipated by the vendor may require the use of technologies external to the EHR such as web services.[15–17] Although the EHR typically receives the highest level of support (e.g., dedicated staff available 24 hours per day, minimal downtime, rapid response to help desk calls), components that are accessible via the EHR but reside externally may receive less attention within an organization. To maintain safe and effective use, it is important for these systems to adhere to common standards, including safety, security, privacy, reliability, scalability, data integrity, and system integrity.[18,19] To explore challenges that may need to be addressed to ensure the successful implementation of customized CDS systems within an EHR, we present five incidents that occurred at one academic health system that delivered 20 different CDS modules during the evaluation period. These incidents are illustrative of the range of problems that can occur with web service-based advanced customization of CDS integrated within an EHR.

Objective

This article identifies technology failures that impacted clinical care related to web service-based advanced custom CDS systems embedded in the complex sociotechnical context of a production EHR.

Methods

Study Setting

The failure modes described below originated from Children's Hospital of Philadelphia's (CHOP) primary care network, which treats more than 200,000 children annually at 33 practice sites spanning urban, suburban, and rural regions of Pennsylvania and New Jersey, and includes both academic and nonacademic practices. These sites are linked by a common EHR (Epic Systems Inc., Verona, Wisconsin, United States) and utilize a homegrown CDS framework, the Care Assistant, to provide targeted decision support.[20]

The Care Assistant is an application framework that enables CHOP to extend the functionality of its commercial EHR by utilizing a web services-based approach to CDS similar to the approach that was later implemented by SMART through its use of the Fast Healthcare Interoperability Resources standard.[15–17,21–23] As interest in using the EHR to support complex decision-making tasks at CHOP increased, we switched to web services for CDS in 2008. The first project to use the new framework was an implementation of asthma treatment guidelines.[24] The Care Assistant fully integrates into the physician workflow as an embedded section within the EHR alongside other functionality and appears to the clinician to be a part of the EHR itself.

Two distinct advantages of our framework are its ability to write data to the patient's medical record as well as the ability to have external services perform computational logic on the patient's data (►Fig. 1). These features allow our hospital to provide enhanced CDS using data that may reside outside the EHR (e.g., teacher surveys for children with attention deficit hyperactivity disorder collected in a third-party system) and execute complex decision logic that may be impractical to implement using native EHR functionality (e.g., determining whether a child is at high risk for severe complications of respiratory syncytial virus).[25,26]

The development of this framework has supported several successful research studies as well as provided meaningful decision support across a wide variety of clinical efforts including administration of routine childhood immunizations, primary care for premature infants (►Fig. 2), otitis media management, short stature evaluation, and shared decision-making activities.[24,25,27–32] Since its inception, the Care Assistant has expanded its reach to cover additional specialty clinics, the emergency department, and inpatient units.[33]

Incident Identification

We performed a systematic review of Care Assistant errors using our hospital's former and current incident reporting systems (ServiceNow, Santa Clara, California, United States, and BMC Service Desk Express, Houston, Texas, United States, respectively) to identify major incidents. To ensure a comprehensive evaluation, we examined Care Assistant errors from the time Care Assistant first became operational in January 2008, through December 2016 (the last full calendar year of data available prior to our evaluation efforts). Major incidents at CHOP are distinguished from routine incidents based on how they are created and the severity of their impact on clinical workflow. Major incidents are created manually by our organization's Information Services team and are created only after an EHR incident has been reported by five or more distinct users. Our search for major incidents included technical terms specific to the Care Assistant framework as well as clinical terms related to CDS modules that were supported by the framework (see ►Appendix A). The list of major incidents ($N = 318$) was

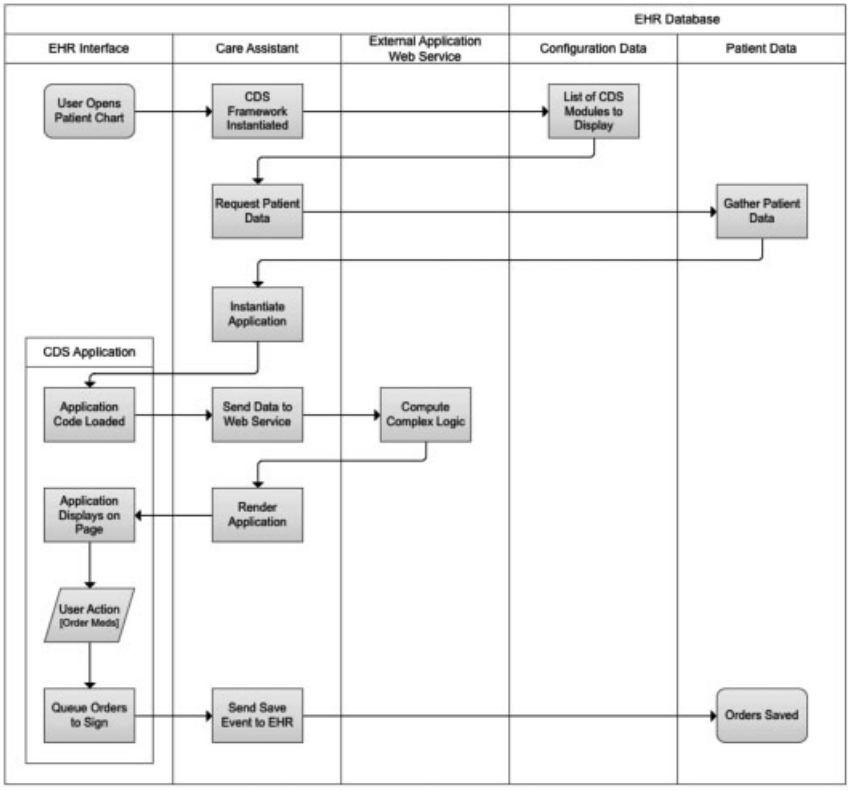

Fig. 1 Electronic health record (EHR) integration of Care Assistant.

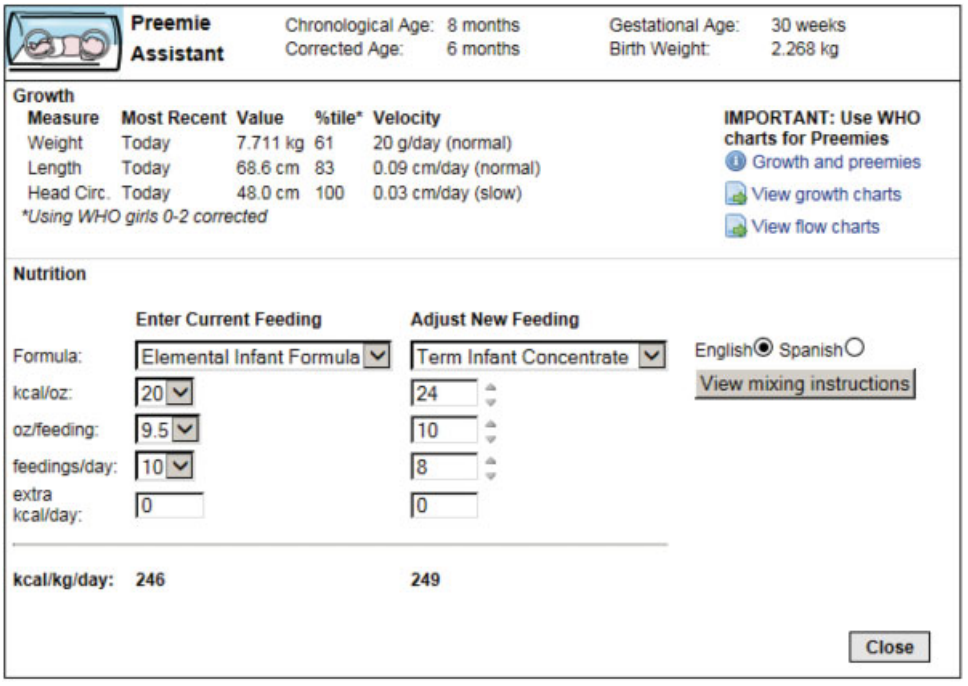

Fig. 2 The Preemie Assistant Module.

manually reviewed by two authors (J.T. and R.G.) to identify errors that met Sittig and Singh's definition of EHR-related errors, which occur "anytime health information technology (IT) is unavailable for use, malfunctions during use, is used incorrectly by someone, or when health IT interacts with another system component incorrectly, resulting in data being lost or incorrectly entered, displayed, or transmitted."[34] Additional taxonomies exist for the purpose of identifying errors relating specifically to CDS recommendations.[13] However, while these taxonomies provide valuable information related to CDS malfunctions, the focus of our work was on assessing the safety of

integrating external decision support into a commercial EHR. This includes identification of any potential degradation of native EHR features and the unintended effects on workflow involving either the CDS system or EHR. As such, we chose to utilize the more general classification of EHR-related errors proposed by Sittig and Singh as described above. Incidents identified in the search, but considered irrelevant (e.g., incidents that related only to native EHR functionality), were omitted from further review leaving three for analysis.

We also conducted interviews with six Care Assistant developers, one EHR analyst, and one EHR database administrator to provide qualitative feedback on the software development life cycle for both our custom framework and its supported modules. At the time of the interviews, six interviewees had five full years of experience with the Care Assistant, and two interviewees had three full years of experience. The interviewees represented all steps within the software development life cycle of our web services-based CDS framework as well as all levels of support for CDS development within the EHR. The interviews were conducted by a single author (J.T.). Interviewees were asked to describe the incidents they personally recalled related to the Care Assistant. They reported their recollection of each incident's timing, apparent cause, and impacts (technical or clinical). The interviewer recorded detailed notes, which were reviewed by all authors.

Finally, a review of our version-controlled source code records (GitHub, San Francisco, California, United States) for our custom framework's code base was performed to identify bug fixes and releases attributed to incident reports and patient safety issues. This review of source code allowed us to identify additional "near misses" that were caught during testing and removed before being deployed in the production environment. These "near miss" events not migrated to production were also included in our review as we feel they offer valuable insight into additional pitfalls that may arise from the complex efforts to customize CDS within a vendor-supplied EHR.

Incident Coding

A final list of five incidents was compiled using data from the three collection methods and was reviewed by three authors (J. M., J.T., and R.G.) for relevance. The expertise of the authors coding the incidents included two software developers (J.M. and J.T.) and one physician/programmer (R.G.). All three had thorough knowledge of the Care Assistant and the associated EHR infrastructure, and were involved in both the testing and implementation of various Care Assistant modules. Each incident was evaluated against the principles of well-behaved applications and mapped to a respective concept. Disagreements regarding the root cause of each incident were discussed until there was complete consensus. Incidents that could be mapped to multiple concepts were discussed among the team until a single concept was identified as the best fit.

Results

Over the 9-year study period, review of help desk incident reports revealed three major incidents attributed to the Care Assistant. Our review of source code identified two incidents, one of which was a near miss that was caught and corrected before being deployed to production. Interviews with the Care Assistant team revealed no additional incidents but provided valuable feedback on the root cause of individual incidents. These five malfunctions were related to four principles of well-behaved applications: (1) system integrity—an application does not cause errors in other systems; (2) data integrity—an application does not corrupt data; (3) reliability—an application is stable and predictable; and (4) scalability—an application performs the same regardless of user load. Two malfunctions of the customized CDS significantly impaired clinical workflow for a total of 5 hours. Lesser impacts—loss of features for a small subset of users or loss of features that have alternative implementations supported by the EHR (e.g., a function designed to increase provider efficiency, but maintain the same general workflow)—arose from three malfunctions, which affected 53 days (►Table 1). These malfunctions are described in detail below. Of note, EHR failures unrelated to CDS have had much more substantial impacts on our health organization (e.g., data center downtimes on two separate occasions that were unrelated to any customization efforts) during the period considered in this study.

Failure Mode 1: System Integrity—Disabling Native EHR Functionality

Incident 1: Loss of Keyboard Shortcuts

During the pilot run of an intervention in the ambulatory workflow, participating clinicians noticed that they were unable to use keystroke shortcuts to select options from "pop-up" documentation lists during their routine note taking process. After careful analysis of the incident by two authors (J. T. and R.G.), it was discovered that one module in our custom framework was responsible for the loss of this EHR functionality due to the faulty declaration of a feature—specifically for a clickable button—within the module's Hypertext Markup Language (HTML) content. Our EHR displays certain components of its medical record system using Microsoft's Internet Explorer web browser and uses HTML element tags to drive many functions. While the custom intervention and native EHR components displayed appropriately upon loading, the construction of an incomplete HTML tag had cascading effects to our EHR's event handling system, which inadvertently prevented EHR logic from fully executing, causing the keystroke issue the clinicians experienced.

The module that caused the problem was a pilot project involving the delivery of public health alerts (e.g., infectious disease outbreaks) that was only available to three participating clinicians. Consequently, this malfunction was not identified through the incident review process, but was deemed relevant during the source code review and developer interviews. This incident was identified and resolved within 1 day; however, we attribute the speed of this resolution to the error being identified serendipitously by a clinician familiar with the project.

Incident 2: Inability to Secure Workstations

Our CDS framework integrates directly within the user's typical EHR workflow, with features including navigation

Table 1 Care Assistant failure modes

Failure mode	Issue summary	Impact duration	Impact severity
System integrity	Incident 1–Faulty HTML displayed by the Care Assistant prevented clinicians from using standard keyboard shortcut features for writing progress notes	1 day	Low – Only three pilot clinicians affected
	Incident 2–Removal of an event listener from the Care Assistant framework prevented users from using the "secure workstation" feature	0 days	None – Identified and removed in test environment
Data integrity	Premature querying of a Care Assistant module interface caused a pop-up exception and eventual shutdown of the user's workstation	52 days	Medium – Close encounter feature from the message inbox was removed for all clinicians until a full patch to the Care Assistant was available
Reliability	Routine updates to the web service for the Care Assistant's Immunizations module caused a dramatic increase in its execution time. During peak hours, the web server became saturated with requests and thus unresponsive causing the EHR to freeze for the user	2 hours	High – Complete loss of access to the EHR for all users (both clinical and administrative) in primary care office locations
Scalability	Rapid growth in the number of Care Assistant modules deployed at the hospital resulted in an unsustainable configuration management process. As a result, configuration meant for a future update was accidentally moved into the production environment	3 hours	High – Nearly all Care Assistant modules affected for all clinicians

Abbreviations: EHR, electronic health record; HTML, Hypertext Markup Language.

within the chart as well as writing data to the patient's chart. This unique ability is a result of custom programming logic that interacts at the lower levels of the EHR's technology stack (i.e., automatically saving data when the EHR workstation "times out" due to inactivity). While advantageous in many respects, this level of customization requires a deep understanding of the EHR as well as ongoing maintenance for version upgrades and routine vendor patches that go beyond the scope of a typical EHR upgrade.

In preparation for an EHR upgrade, a particular section of the framework's code base, written in Visual Basic 6, was updated to be compatible with the new version. The new code was tested in a development environment based on common workflows and then loaded into our test environments. Thirty-four days later, an author (J.T.) was testing a new CDS module when he noticed that the EHR feature to secure a workstation was not working in certain situations. The problem was noted to only occur after our framework loaded into the patient's chart. A review of the recent changes to the framework's source code showed that the framework was no longer correctly handling a native EHR event, which inadvertently disabled a user's ability to secure their workstation for the next user. The delay in identifying this error was determined to be a result of the fact that the "secure workstation" feature, while used heavily in production clinical workflows, is used only rarely in test scenarios. This incident was identified and removed before reaching the production environment. While the incident never reached the production EHR, this "near miss" event was included since it offers valuable insight into the complex sociotechnical environment of the EHR and into the additional pitfalls that may arise from the complex efforts to customize CDS within a vendor-supplied EHR.

Failure Mode 2: Data Integrity—Unexpected EHR Workflow

Features of our framework related to asynchronous access to remote data and delayed evaluation of complex decision logic came to heightened attention in June of 2016 when a new feature was implemented by our information services team that allowed users to sign encounters directly from their EHR message inbox. Shortly upon deployment to the production environment, several incidents were filed due to an interruptive "pop-up" error alert from the module and eventual forced closure of the EHR when attempting to use the aforementioned feature for certain patients. It was not immediately clear why the framework was causing an error and it was difficult to reliably replicate the error.

An analysis of the EHR error logs by two developers with a deep understanding of both the EHR and our custom CDS framework (J.M. and J.T.) identified the issue to be a result of a data "race condition" originating in the framework. The new EHR feature—intended to help clinicians sign and close encounters more quickly—was implemented similar to a "macro" in that it imitated the actions a clinician would normally perform to sign an encounter in a series of rapid automated steps: (1) open the encounter; (2) navigate to the "sign encounter" activity; (3) perform a series of checks to verify if the encounter was complete; and (4) sign the encounter if all checks passed successfully.

In a typical workflow, when an encounter is opened our framework begins to load its modules and sends patient data to any required external services (e.g., a service to execute complex decision logic). When the clinician exits an encounter (as happens when signing a chart), the framework gives each module an opportunity to save data to the patient's chart. However, when a clinician accesses an encounter, the framework does not allow its modules to render visibly on the screen until data returns from any external services. The rapid navigation in and out of the encounter by the EHR's new "sign encounter" feature created the possibility that the framework could instruct a module to save its data prior to actually rendering it. As a result, the modules would, in some cases, attempt to extract information from fields in a user interface that did not yet exist, causing an exception (►Fig. 3).

The sign encounter feature was removed from production for 52 days until a full patch of the framework was implemented. During this time, no patient care was directly affected and all other features of the framework and EHR were fully operational. This error was particularly difficult to track down since it was not consistently reproducible. This was, in large part, due to the combination of the EHR and external service's response times, which are affected by many factors, such as the number of users simultaneously accessing the EHR.

Failure Mode 3: Reliability—Unresponsive Server

Another incident underscored two major themes in CDS errors: (1) software complexity caused by unintended component interaction; and (2) the difficulty in testing multiple components under conditions representative of a production EHR. In this incident, two separate components—an external web service and a web control used to make external web requests—conspired to cause the EHR workstation to hang indefinitely. This led to a service outage lasting 2 hours.

The incident was caused by a code update to a web service that provided immunization decision support. Unknown to the software developers, the update significantly increased the execution time required to analyze each patient's immunization record. When confronted with a heavy user load during peak clinic time, the web server became overloaded and slow to respond. Alone, this problem would have caused a relatively minor inconvenience, affecting only the immunization module. However, since the web service did not completely fail, the software handling the web request on the client was left in a pending state. When the clinician inevitably tried to close the medical record while the web operation was still pending, an undocumented issue in a software component of the workstation's operating system caused the entire EHR client application to freeze.

Failure Mode 4: Scalability—Configuration Management

The operational complexity of supporting multiple web services across a large health system is significant. Trigger criteria, which determine when CDS appears, change over time. New modules come online (e.g., to access new sources of clinical data), old modules are retired, and the location of web services and code change. Additionally, we manage several different EHR environments aside from production, including development, test, and training environments. All of this information was tracked in a single Excel spreadsheet that was processed into an EHR-specific format and pushed to production during planned service windows. In the beginning, with a single Asthma module supported by a single informatics researcher,

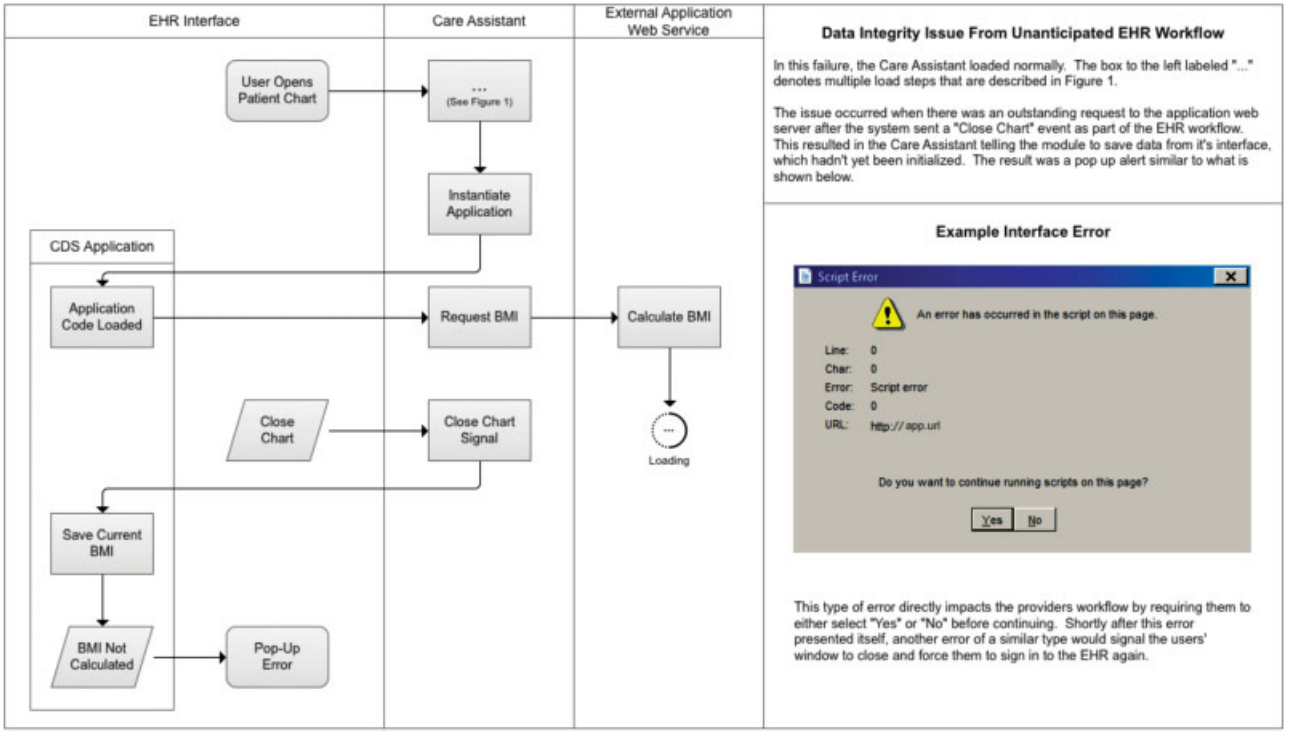

Fig. 3 Data integrity issue from unexpected electronic health record (EHR) workflow.

this was a manageable solution. This situation quickly changed as more modules were created and managed by multiple people across two major hospital departments. In particular, it became difficult to clearly see what changes had been made to the configuration since the last update and manage these changes as they propagated through the environments on the path to production.

In this specific scenario, we were preparing to migrate the hosting of the main Care Assistant framework code to our hospital information services team, but this update was accidently and prematurely included in one of a series of monthly maintenance updates. This caused the EHR to request the framework code from servers that were not yet configured. The result was 3 hours of downtime for nearly all custom CDS modules. Fortunately, this did not degrade other parts of the medical record, but the custom modules that clinicians rely upon were temporarily unavailable during a busy time of day.

Discussion

This article reports five incidents that contributed to four modes of failure experienced as a result of efforts to deliver advanced CDS using a web service-based architecture over a 9-year period. Four of these incidents (Incidents #1 and #3–5, See ►Table 1) were directly related to our web-based infrastructure and one (Incident #2) could be attributed to general custom development within a commercial EHR. These failure modes included problems with (1) system integrity; (2) data integrity; (3) reliability; and (4) scalability. Significant disruptions to clinical workflow only occurred with two incidents, but were relatively brief (5 hours total). All but one of the five malfunctions affected other areas of the medical record beyond the customized CDS.

In addition to the individual failure modes described above, our study identified three themes that occurred across multiple failures. One such theme was the difficulty of adequately preventing and/or detecting these failures from reaching the end user. Recent studies on native EHR CDS alerts have shown that CDS malfunctions are common and often undetected.[9,11,13,14] These malfunctions can manifest themselves in the form of false-alarms, as is the case with misfired CDS, which can lead to a mistrust of the system.[35–38] Our findings also show the potential for programming errors to propagate to the user's workspace forcing a disruption of their current workflow. These malfunctions are interruptive to the physician workflow, which has been shown to decrease the effectiveness of clinical care in specific workflows.[39–41] To combat this issue, we have implemented a more robust logging infrastructure along with alerts to immediately notify relevant developers of the errors before they reach the end user, an approach that institutions seeking to implement customized CDS features might find helpful. The alerts are configured to identify common error messages from the application and to identify spikes or lulls in use that could help point to potential CDS misfires.

Our results also suggest that it is difficult to effectively test a customized CDS intervention in a nonproduction environment. Individual software components, fairly simple alone, may interact with one another in complex and unpredictable ways.[8,42–47] Typically used monitoring and testing procedures may be inadequate to detect malfunctions before impacting physician workflow, which can lead to workflow fragmentation and a higher likelihood of mistakes.[8,48] Typical software testing often relies on unit tests, where parts of an application are individually tested for proper operation. While useful, these tests are rarely (if ever) allowed within a production EHR for security and safety reasons. As such, unit testing fails to capture the complex ecosystem of the EHR, and unique clinical workflows are inevitably left untested. Possible solutions to this issue, which are already in place at most organizations, include creating a copy of a production environment. This helps to overcome the limited complexity of synthetic data in test environments. However, given the distributed nature of the health care environment and multitenant EHRs, certain workflows that involve multiple users and external systems will likely remain untested. Software that can better simulate or automate complex clinical workflows may help to further increase the effectiveness of CDS testing. For example, it may be feasible to record actual production transactions related to multiuser workflows. These transactions could then be "replayed" against a copied environment to better simulate the pace and complexity of clinical care for purposes of CDS testing.

Our study also pointed out the importance of effective configuration management that spans multiple teams. Many of our web services-based CDS projects were developed and piloted in the research arm of our organization. However, as the projects became successful, our web services-based CDS framework was requested for use by additional teams within the hospital. Had all web services development efforts remained within a single team, it is possible some of the incidents we experienced may not have occurred, which highlights that the technology itself is not always the "root cause" of health IT failures. In our case, the collaboration with additional teams highlighted weaknesses in our initial configuration management efforts, which consisted of a single spreadsheet stored in a shared drive. Our teams began to rely too heavily on e-mail communication with inconsistent lists of recipients to orchestrate updates, which eventually caused a 3-hour downtime of all custom CDS modules. In response, we now use a custom web application to hold and update our configuration. This enforces mandatory comments on all configuration changes and provides a more manageable visual interface to the configuration, including a full history of all changes. The application also automatically archives a serialized text copy of each version of the configuration into a version control system where it can be tagged with release numbers and examined using traditional code management tools. Such approaches are likely to prove helpful to others working in settings with similar complexities.

Limitations

Our study was limited to a single institution that helped support its research mission by adopting web service

approaches to implement CDS a decade before standard toolkits such as SMART became available. It is possible that certain incidents (e.g., complete failure of the EHR) experienced by our health system are unlikely to reoccur now that approaches similar to ours are becoming more standard. Also, one of our incidents (#2) was specifically caused by software written in Visual Basic and would be unlikely to occur in other settings. However, other issues such as the challenges related to maintaining a large portfolio of customized CDS will very likely remain important for the foreseeable future. Additionally, our sources of safety information were incident reports from end users, interviews with those familiar with the Care Assistant framework, and Care Assistant source code. There were likely additional actual and potential incidents beyond those that were identified through our sources of data.

Conclusion

Although advanced customization of CDS within a commercial EHR is increasingly desirable to promote safe and effective health care, these customization efforts are not without risk. We encourage teams engaged in this line of work to be constantly vigilant both in their testing, configuration management, and postimplementation monitoring activities.

Clinical Relevance Statement

Advances in health information exchange have made external clinical decision support (CDS) applications feasible. This capability has also introduced the potential for programming errors to propagate to the user's workspace and disrupt clinical workflow. Individual software components within a complex sociotechnical environment can interact with one another in unpredictable ways. Institutions that customize CDS should ensure robust testing and monitoring infrastructures are in place.

Multiple Choice Questions

1. The customization of clinical decision support (CDS) using external software components is difficult for which of the following reasons?
   a. CDS alert colors are hard to pick.
   b. External CDS modules are difficult to fully test.
   c. CDS developers are too busy.
   d. No external systems are needed.

   Correct Answer: The correct answer is option b. External clinical decision support development poses many challenges, but when done correctly offers a variety of benefits. Choosing colors for an application, while potentially important, is a generally straightforward task. Testing of an external CDS application within an EHR is extremely difficult given the complex workflows and interactions among human and machine. Developers can be stretched for time, but this is not a reason by itself to create difficulties during CDS development. EHR vendors provide support for a variety of useful tools; however, to care for a more complex patient population, external systems are often required.

2. A health care organization would like to implement a new clinical decision support (CDS) tool that is not directly provided by their electronic health record (EHR) vendor. Which of the following is the most appropriate next step to implement the new tool?
   a. Forget about the tool since it's not possible to implement it.
   b. Wait until the vendor can provide a solution.
   c. Assess available technologies to see if implementation is possible by augmenting the EHR with custom methods.
   d. Begin implementation immediately.

   Correct Answer: The correct answer is option c. While EHR vendors provide support for a variety of useful tools, it is possible that current functionality does not yet provide out of the box support for all user requirements. If this is the case, there may be other features that, when combined with custom programming, can solve a particular problem. Forgetting about the tool is not an appropriate response since it may be possible to implement the tool via standard methods. Waiting until the vendor provides a solution can take many years and there is no guarantee it will be implemented at all. It may be appealing to deploy any useful tool to your EHR, but an assessment of the feasibility and cost should always be considered, so beginning implementation immediately is not an appropriate next step.

Protection of Human and Animal Subjects

This study did not involve human or animal subjects research.

Funding

Dr. Fiks is a coinventor of the "Care Assistant" software that was evaluated in this study. He holds no patent on the software and has earned no money from this invention. No licensing agreement exists. Dr. Fiks has also received an Independent Research Grant from Pfizer from which he personally drew no support. Dr. Grundmeier is a coinventor of the "Care Assistant" software that was evaluated in this study. He holds no patent on the software and has earned no money from this invention. No licensing agreement exists. Mr. Miller is a coinventor of "Project SHARE" which is a Care Assistant module to support the care of children with attention deficit disorder. He holds no patent on the software and has earned no money from this invention. No licensing agreement exists. The other authors have indicated they have no financial relationships relevant to this article to disclose.

Conflict of Interest

Drs. Fiks and Grundmeier are coinventors of the "Care Assistant" software that was evaluated in this study. Mr. Miller is a coinventor of "Project SHARE" which is a Care Assistant module to support the care of children with attention deficit disorder. They hold no patent on the software and have earned no money from this invention. No

licensing agreement exists. Dr. Fiks has also received an Independent Research Grant from Pfizer from which he personally drew no support. The other authors have indicated they have no potential conflicts of interest to disclose.

References

1 Sim I, Gorman P, Greenes RA, et al. Clinical decision support systems for the practice of evidence-based medicine. J Am Med Inform Assoc 2001;8(06):527–534
2 Johnston ME, Langton KB, Haynes RB, Mathieu A. Effects of computer-based clinical decision support systems on clinician performance and patient outcome: a critical appraisal of research. Ann Intern Med 1994;120(02):135
3 Kaplan B. Evaluating informatics applications–clinical decision support systems literature review. Int J Med Inform 2001;64(01):15–37
4 Balas EA, Weingarten S, Garb CT, Blumenthal D, Boren SA, Brown GD. Improving preventive care by prompting physicians. Arch Intern Med 2000;160(03):301–308
5 Gardner RM, Lundsgaarde HP. Evaluation of user acceptance of a clinical expert system. J Am Med Inform Assoc 1994;1(06):428–438
6 Eligible Professional Meaningful Use Core Measures Measure 6 of 17, Stage 2, Clinical Decision Support Rule. Available at: https://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/downloads/Stage2_EPCore_6_ClinicalDecisionSupport.pdf. Accessed July 12, 2017
7 Definition of Terms Eligible Professional Meaningful Use Core Measures Measure 10 of 13. Available at: https://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/downloads/11_Clinical_Decision_Support_Rule.pdf. Accessed July 12, 2017
8 Ash JS, Berg M, Coiera E. Some unintended consequences of information technology in health care: the nature of patient care information system-related errors. J Am Med Inform Assoc 2004;11(02):104–112
9 Wright A, Hickman T-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
10 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
11 Koppel R, Metlay JP, Cohen A, et al. Role of computerized physician order entry systems in facilitating medication errors. JAMA 2005;293(10):1197–1203
12 Ash JS, Sittig DF, Poon EG, Guappone K, Campbell E, Dykstra RH. The extent and importance of unintended consequences related to computerized provider order entry. J Am Med Inform Assoc 2007;14(04):415–423
13 Wright A, Ai A, Ash J, et al. Clinical decision support alert malfunctions: analysis and empirically derived taxonomy. J Am Med Inform Assoc 2018;25(05):496–506
14 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
15 Mandl KD, Mandel JC, Murphy SN, et al. The SMART Platform: early experience enabling substitutable applications for electronic health records. J Am Med Inform Assoc 2012;19(04):597–603
16 Mandel JC, Kreda DA, Mandl KD, Kohane IS, Ramoni RB. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. J Am Med Inform Assoc 2016;23(05):899–908
17 Mandl KD, Kohane IS. No small change for the health information economy. N Engl J Med 2009;360(13):1278–1281
18 App Orchard Program Details. 2017. Available at: http://www.epic.com/. Accessed October 9, 2017
19 Gorton I. Software quality attributes. In: Essential Software Architecture. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011: 23–38. Available at: http://link.springer.com/10.1007/978-3-642-19176-3_3. Accessed October 9, 2017
20 Fiks AG, Grundmeier RW, Margolis B, et al. Comparative effectiveness research using the electronic medical record: an emerging area of investigation in pediatric primary care. J Pediatr 2012;160(05):719–724
21 Overview - FHIR v3.0.1. Available at: https://www.hl7.org/fhir/overview.html. Accessed July 13, 2017
22 Encounter - FHIR v3.0.1. Available at: https://www.hl7.org/fhir/encounter.html. Accessed August 29, 2017
23 Gordon WJ, Baronas J, Lane WJ. A FHIR human leukocyte antigen (HLA) interface for platelet transfusion support. Appl Clin Inform 2017;8(02):603–611
24 Bell LM, Grundmeier R, Localio R, et al. Electronic health record-based decision support to improve asthma care: a cluster-randomized trial. Pediatrics 2010;125(04):e770–e777
25 Utidjian LH, Hogan A, Michel J, et al. Clinical decision support and palivizumab: a means to protect from respiratory syncytial virus. Appl Clin Inform 2015;6(04):769–784
26 Fiks AG, Mayne S, Hughes CC, et al. Development of an instrument to measure parents' preferences and goals for the treatment of attention deficit-hyperactivity disorder. Acad Pediatr 2012;12(05):445–455
27 Fiks AG, Mayne SL, Karavite DJ, et al. Parent-reported outcomes of a shared decision-making portal in asthma: a practice-based RCT. Pediatrics 2015;135(04):e965–e973
28 Fiks AG, Mayne S, Karavite DJ, DeBartolo E, Grundmeier RW. A shared e-decision support portal for pediatric asthma. J Ambul Care Manage 2014;37(02):120–126
29 Fiks AG, Grundmeier RW, Biggs LM, Localio AR, Alessandrini EA. Impact of clinical alerts within an electronic health record on routine childhood immunization in an urban pediatric population. Pediatrics 2007;120(04):707–714
30 Fiks AG, Grundmeier RW, Mayne S, et al. Effectiveness of decision support for families, clinicians, or both on HPV vaccine receipt. Pediatrics 2013;131(06):1114–1124
31 Forrest CB, Fiks AG, Bailey LC, et al. Improving adherence to otitis media guidelines with clinical decision support and physician feedback. Pediatrics 2013;131(04):e1071–e1081
32 Lipman TH, Cousounis P, Grundmeier RW, et al. Electronic health record mid-parental height auto-calculator for growth assessment in primary care. Clin Pediatr (Phila) 2016;55(12):1100–1106
33 Asthma Clinical Pathway — Inpatient | Children's Hospital of Philadelphia. Available at: http://www.chop.edu/clinical-pathway/asthma-inpatient-care-clinical-pathway. Accessed July 12, 2017
34 Sittig DF, Singh H. Defining health information technology-related errors: new developments since to err is human. Arch Intern Med 2011;171(14):1281–1284
35 Wickens CD, Hollands JG, Banbury S, Parasuraman R. Engineering Psychology and Human Performance. Vol. 4, Engineering psychology and human performance. Pearson; 2013:394–395
36 Breznitz S. Cry-wolf: The Psychology of False Alarms. Hillsdale, NJ: Lawrence Erlbaum Associates; 1984:9–14
37 Sorkin RD. Why are people turning off our alarms? J Acoust Soc Am 1988;84(03):1107–1108
38 Rehr CA, Wong A, Seger DL, Bates DW. Determining inappropriate medication alerts from "inaccurate warning" overrides in the intensive care unit. Appl Clin Inform 2018;9(02):268–274
39 Drews FA. The frequency and impact of task interruptions in the ICU. Proc Hum Factors Ergon Soc Annu Meet 2007;51(11):683–686
40 Wiegmann DA, ElBardissi AW, Dearani JA, Daly RC, Sundt TM III. Disruptions in surgical flow and their relationship to surgical errors: an exploratory investigation. Surgery 2007;142(05):658–665
41 Sevdalis N, Forrest D, Undre S, Darzi A, Vincent C. Annoyances, disruptions, and interruptions in surgery: the Disruptions in Surgery Index (DiSI). World J Surg 2008;32(08):1643–1650
42 Horsky J, Phansalkar S, Desai A, Bell D, Middleton B. Design of decision support interventions for medication prescribing. Int J Med Inform 2013;82(06):492–503

43 Gaba DM, Howard SK, Small SD. Situation awareness in anesthesiology. Hum Factors 1995;37(01):20–31
44 Horsky J, Schiff GD, Johnston D, Mercincavage L, Bell D, Middleton B. Interface design principles for usable decision support: a targeted review of best practices for clinical prescribing interventions. J Biomed Inform 2012;45(06):1202–1216
45 Walker JM, Carayon P, Leveson N, et al. EHR safety: the way forward to safe and effective systems. J Am Med Inform Assoc 2008;15(03):272–277
46 Sittig DF, Classen DC. Safe electronic health record use requires a comprehensive monitoring and evaluation framework. JAMA 2010;303(05):450–451
47 Sittig DF, Singh H. Defining health information technology-related errors: new developments since to err is human. Arch Intern Med 2011;171(14):1281–1284
48 Zheng K, Haftel HM, Hirschl RB, O'Reilly M, Hanauer DA. Quantifying the impact of health IT implementations on clinical workflow: a new methodological perspective. J Am Med Inform Assoc 2010;17(04):454–461

Appendix A

Incident Reporting System Search Terms:

- ACP
- Asthma
- Asthma Care Plan
- Audiology
- Care Assistant
- Epic
- Spirometry
- EpicCare
- Screenings
- Premature Infant
- Immunizations