Proposing Regulatory-Driven Automated Test Suites for Electronic Health Record Systems

Patrick Morrison, Casper Holmgreen, Aaron Massey, Laurie Williams
Department of Computer Science
North Carolina State University
Raleigh, NC
{pjmorris, cmholmgr, akmassey, lawilli3}@ncsu.edu

***Abstract*—In regulated domains such as finance and health care, failure to comply with regulation can lead to financial, civil and criminal penalties. While systems vary from organization to organization, regulations apply across organizations. We propose the use of Behavior-Driven-Development (BDD) scenarios as the basis of an automated compliance test suite for standards such as regulation and interoperability. Such test suites could become a shared asset for use by all systems subject to these regulations and standards. Each system, then, need only create their own system-specific test driver code to automate their compliance checks. The goal of this research is to enable organizations to compare their systems to regulation in a repeatable and traceable way through the use of BDD. To evaluate our proposal, we developed an abbreviated HIPAA test suite and applied it to three open-source electronic health record systems. The scenarios covered all security behavior defined by the selected regulation. The system-specific test driver code covered all security behavior defined in the scenarios, and identified where the tested system lacked such behavior.**

Keywords- Behavior-Driven-Development; Healthcare IT; Regulatory Compliance; Security; Software Engineering; Software Testing

I. INTRODUCTION

In regulated domains such as finance and healthcare, organizations must ensure their software systems comply with applicable laws and regulations. Failure to comply often carries financial, civil and even criminal penalties. While systems vary widely among organizations, they must all check compliance against the same regulatory requirements.

Approaches to compliance vary across industries, but typically include elements such as staff training, manual and automated monitoring, internal and external audits, and software certification. Compliance is also a concern over the entire software lifecycle, from requirements [1] [2] to ongoing maintenance [3].

The goal of this research is to enable organizations to compare their systems to regulation in a repeatable and traceable way through the use of Behavior-Driven-Development (BDD) [4]. A test suite built from standard scenarios that depict behavior specified by a regulation can help to confirm that important issues have been addressed. An organization can obtain indications of how their system will respond to external audits through use of the test suite. At an industry level, a common test suite of these standard scenarios provides a target for implementers and a basis for comparison among systems. Such a test suite can be shared among all organizations that must adhere to a regulation.

BDD is a software development practice that organizes development effort around the creation of scenarios that illustrate desired system behavior in terms of the vocabulary used by system stakeholders [4]. Scenarios are descriptions of system behavior expressed in system user vocabulary. These scenarios are then automated through the creation of system-specific test driver code that binds each scenario to the system. Each scenario, combined with the system-specific test driver code, serves as a test of the system's behavior. The collection of scenarios forms an acceptance test suite for the system. They can be automatically executed to verify system behavior. Frameworks that support this style of development include FIT [5], FitNesse[1], JBehave[2] and Cucumber [4]. The typical use case for BDD is in custom software system development. The scenarios and the system-specific test driver code are both associated with a single custom software system [6][7].

We propose the use of BDD scenarios as the basis of an automated compliance test suite for standards such as regulation and interoperability. Such test suites could become a shared asset for use by all systems subject to these regulations and standards. Each system, then, need only create their own system-specific test driver code to automate their compliance checks. System owners and auditors can gain confidence in the compliance of a system by running the compliance test suite on the system. This paper presents our proposal for using BDD technology to implement reusable test suites for regulations by demonstrating a partial HIPAA test suite on three electronic health record

[1] http://fitnesse.org/
[2] http://jbehave.org/

(EHR) systems; iTrust[3], OpenEMR[4] and Tolven[5], and gives an overview of our initial evaluation of this proposal.

The remainder of the paper is organized as follows; Section 2 provides background for BDD, Regulation, Certification and test suites. Section 3 describes our development process. Section 4 presents our evaluation and Section 5 presents a summary and next steps.

II. BACKGROUND

A growing body of research examines how to link regulations and software requirements [8][1][9]. Within that, there has been some focus on how to measure the performance of running systems [1][10] against a requirements baseline by implementing a custom monitoring system. Our approach treats the BDD test suite as the monitoring system, based on commonly available BDD technology. Each BDD scenario is written in terms of the applicable regulation rather than a requirements specification. For each EHR system, system-specific driver code implements the scenario.

Given the significant consequences of not addressing regulatory compliance issues, attention has been paid by the requirements engineering community to eliciting requirements from legislation [9], [11].

In the United States, healthcare organizations must comply with the HITECH and Health Insurance Portability and Accountability Act (HIPAA) Acts, among others. HITECH regulations stipulate that failure to protect personal health information can lead to fines of up to $50,000 per violation and imprisonment for up to one year. A successful test suite implementation could provide assistance in evaluating systems for this high stakes regulatory concern.

Two sources of guidance in the EHR domain are the Certification Commission for Health Information Technology (CCHIT)[6], and the National Institute of Standards and Technology (NIST)[7]. NIST has published a suite of test procedures targeting the regulations and guidelines established by HITECH[8]. CCHIT provides certification of EHR systems according to a set of internally developed criteria and test scripts. CCHIT makes these criteria and test scripts available on the web[9]. This certification process requires significant manual effort to execute each time, and further effort to review the results. The scripts exercise a wide range of functionality, however they do not necessarily cover all aspects of EHR security [12]. The NIST-developed test procedures form the basis of our BDD scenarios, as there are explicit, documented links made between the NIST procedures and the regulations they are designed to check. Test suites are collections of test cases organized around some unifying purpose.

Morgan Stanley built a BDD test framework for validating financial time series data [13], although the test suite was applied to a single application rather than the multiple applications we propose. In the telecomm domain, a set of test suites for various network interoperability standards was built based upon TTCN-3, a telecomm industry standard for test specification.[10]

In the domain of programming languages, validation suites consisting of executable acceptance tests establish conformance for a given language implementation to its specification. For example, Plum Hall[11] builds compiler validation test suites for C and C++. RubySpec is an open-source executable specification for the Ruby programming language. Java's Technology Compatibility Kit 3 serves a similar function for the Java language.

III. DEVELOPMENT PROCESS

To build a BDD test suite with which multiple organizations can evaluate their systems against a regulation in a repeatable and traceable way, we must, at a minimum, perform the following tasks:

1. Identify Regulations - Identify the regulation(s) for which BDD scenarios will be built. In general, identifying requirements in regulatory texts is a difficult problem that requires not only engineering expertise but legal advice [11]. A readable test suite could assist in interpretation of the regulations being addressed by each test scenario.
2. Develop Scenarios - A scenario, a step-by-step test procedure, must be associated with each tested regulation to validate its achievement. Figure 1 shows an excerpt from the scenario file that implements NIST 170.302(t), which is a test procedure for HIPAA regulation, CFR section 170.302(t).
3. Automate Scenarios - Automate the scenarios as executable tests by a combination of structured natural language and system-specific files. Figures 2 and 3 show the system-specific details for executing the scenario from Figure 1on two different EHRs.

Completing these tasks for all or part of a regulatory text establishes a baseline for the development of the acceptance test suite.

IV. EVALUATION

We evaluated our test suite by measuring and reporting on our execution of the methodology tasks against seven regulations on three EHR systems (iTrust, OpenEMR and

[3] http://agile.csc.ncsu.edu/iTrust/wiki/doku.php

[4] http://www.oemr.org/

[5] http://home.tolven.org/

[6] http://www.cchit.org/ CCHIT is a federally chartered certification bureau in the US..

[7] http://www.nist.gov NIST develops and publishes standards across a wide range of industries and topic areas in the US.

[8] http://healthcare.nist.gov/use_testing/index.html

[9] https://www.cchit.org/cchit-certified

[10] http://www.ttcn-3.org

[11] http://www.plumhall.com/

```
Feature: Authentication

NIST Â§170.302(t) Authentication

Background:
       * Using the Vendor-identified EHR function(s), the Tester shall create a new user
account and assign permissions to this new account

Scenario: Verify authorization

DTR170.302.t 1:  Verify authorization     evaluates the capability to verify that a person
or entity seeking access to electronic health information is the one claimed and is
authorized

* Using the new user account, the Tester shall login to the EHR using the new account
* The Tester shall perform an action authorized by the assigned permissions.
* The Tester shall verify that the authorized action was performed
```

Figure 1: Cucumber feature file excerpt, Authentication

Tolven). We have tabulated our results for each task and each system in Table I.

Each row represents one of the scenarios implemented for the test suite, including the section of the regulation that is addressed, the test procedure used, the name of the feature file in which the scenario is implemented, and a score for each EHR system. Systems receive one point each for presence of the functionality tested in the scenario, step code for executing the scenario, connection between the feature file and the step files, and successful execution of the step code testing the functionality.

V. Summary and Future work

We propose the creation and use of BDD acceptance test suites to support checking of regulatory requirements.

Two natural next steps would be to add the remainder of the Meaningful Use regulations to the test suite, and to pursue implementations of the test suite for other EHR systems. A survey of testing procedures and experiences among certification bureaus, developers of EHRs and user organizations (e.g. hospitals, doctor's practices) should be conducted to form a basis for this comparison.

One possible expansion of the current effort would be to

```
Given /^Using the Vendor\-identified EHR
function\(s\), the Tester shall create a new
user account and assign permissions to this new
account$/ do
    @user = default_hcp
    login(driver,@user)
    @new_user =
          create_new_patient(driver,ITrust
          ::User.new(first_name:'Ted',
          last_name:'Nugent',
          email:'ted@nugent.com'))
    driver.find_element(link:'Logout')
        .click
end

Given /^Using the new user account, the Tester
shall login to the EHR using the new account$/
do
    reset_password(
        driver,@new_user,
        'password')
    @user = @new_user
    login(driver,@user)
end

Given /^The Tester shall perform an action
authorized by the assigned permissions\.$/ do
    driver.find_element(
        link:'My Demographics')
        .click
end

Given /^The Tester shall verify that the
authorized action was performed$/ do
    driver.title.should
        == 'iTrust - Edit Patient'
end
```

Figure 2: iTrust step file excerpt, Authentication

```
Given /^Using the Vendor\-identified EHR
function\(s\), the Tester shall create a
new user account and assign permissions to
this new account$/ do
    login("admin","sysadmin")
    create_new_staff
    add_testaccount_to_chr
   logout
end

Given /^Using the new user account, the Tester
shall login to the EHR using the new account$/
do
    login("testaccount","twk27kox")
end

Given /^The Tester shall perform an action
authorized by the assigned permissions\.$/ do
    driver.get(base_url + "/Tolven")
    driver.find_element(:link,
        "Appointments").click
end

Given /^The Tester shall verify that the
authorized action was performed$/ do
    driver.title.should
      match /Appointments/
end
```

Figure 3: Tolven step file excerpt, Authentication

TABLE I. EVALUATION SUMMARY

Regulation	Procedure	Automation	Electronic Health Record System		
			iTrust	*OpenEMR*	*Tolven 2.1*
Access Control - CFR 170.302(o)	NIST 170.302.o	login.feature	3	3	4
Emergency Access – CFR 170.302(p)	NIST 170.302.p	emergency_access.feature	3	3	3
Automatic Logoff – CFR 170.302(q)	NIST 170.302.q	automatic_logoff.feature	4	3	4
Record actions – CFR 170.302(r)	NIST 170.302.r	audit_log.feature	4	3	4
Integrity – CFR 170.302(s)	NIST 170.302.s	integrity.feature	3	3	3
Authorization – CFR 170.302(t)	NIST 170.302.t	authentication.feature	4	4	4
Encryption – CFR 170.302(u)	NIST 170.302.u	general_encryption.feature, transfer_encryption.feature	3	3	3
Totals: 7	7	7	24/28	22/28	25/28

define a domain-specific language for specifying EHR compliance scenarios. These scenarios could illustrate unclear points in the law and serve as guidelines for system certifiers and implementers

ACKNOWLEDGMENT

The authors would like to thank the members of the Realsearch group, the members of The Privacy Place, and the students of CSC 591, section 01 Fall 2011 and CSC 591, section 007, Spring 2012 for their efforts, reviews and comments on early drafts of this work.

REFERENCES

[1] S. Ingolfo, A. Siena, and J. Mylopoulos, "Establishing Regulatory Compliance for Software Requirements," in *Conceptual Modeling – ER 2011*, vol. 6998, M. Jeusfeld, L. Delcambre, and T.-W. Ling, Eds. Springer Berlin / Heidelberg, 2011, pp. 47–61.

[2] T. D. Breaux and A. I. Anton, "Analyzing Regulatory Rules for Privacy and Security Requirements," *Software Engineering, IEEE Transactions on*, vol. 34, no. 1, pp. 5 –20, Feb. 2008.

[3] N. Chapin, "Software maintenance in complying with IT governance: A report from the field," in *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on*, 2009, pp. 499 –502.

[4] M. Wynne and A. Hellesoy, *The Cucumber Book: Behavior-Driven Development for Testers and Developers*. Pragmatic Press, 2012.

[5] R. Mugridge and W. Cunningham, *Fit for Developing Software: Framework for Integrated Tests (Robert C. Martin)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.

[6] B. Haugset and G. K. Hanssen, "Automated Acceptance Testing: A Literature Review and an Industrial Case Study," in *Proceedings of the Agile 2008*, Washington, DC, USA, 2008, pp. 27–38.

[7] G. I. Melnik, "Empirical analyses of executable acceptance test driven development," University of Calgary, Calgary, Alta., Canada, Canada, 2007.

[8] J. C. Maxwell and A. I. Antón, "The production rule framework: developing a canonical set of software requirements for compliance with law," in *Proceedings of the 1st ACM International Health Informatics Symposium*, New York, NY, USA, 2010, pp. 629–636.

[9] A. K. Massey, B. Smith, P. N. Otto, and A. I. Anton, "Assessing the accuracy of legal implementation readiness decisions," in *Requirements Engineering Conference (RE), 2011 19th IEEE International*, 2011, pp. 207 –216.

[10] W. N. Robinson, "Implementing Rule-Based Monitors within a Framework for Continuous Requirements Monitoring," in *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences - Volume 07*, Washington, DC, USA, 2005, p. 188.1–.

[11] P. N. Otto and A. I. Antón, "Addressing Legal Requirements in Requirements Engineering," *Requirements Engineering Conference, 2007. RE '07. 15th IEEE International*, pp. 5–14, Oct. 2007.

[12] A. Austin, B. Smith, and L. Williams, "Towards improved security criteria for certification of electronic health record systems," in *Proceedings of the 2010 ICSE Workshop on Software Engineering in Health Care*, New York, NY, USA, 2010, pp. 68–73.

[13] R. Salama, "A regression testing framework for financial time-series databases: an effective combination of fitnesse, scala, and kdb/q," in *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*, New York, NY, USA, 2011, pp. 149–154.