Vaishnavi Kannan, DuWayne Willett
Agile Clinical Decision Support Development
Agenda for this Experience Report

- **Background**
  - Clinical Decision Support (CDS): What it is, challenges
  - Specialty Patient Registry project

- **Our Story**
  - Problems we encountered
  - What we did:
    - Agile Methods for CDS Development
    - Agile Modeling for CDS Development
    - Automated Testing; Monitoring

- **Results**

- **What We Learned**
Clinical Decision Support (CDS)

Types of CDS

Alerts
- Clinical practice alerts
- Med alerts
- Duplicate alerts

Health maintenance reminders

Order questions

Order sets

Reference information

Reports

Conclusion:
“CDSS malfunctions occur commonly and often go undetected. Better methods are needed to prevent and detect these malfunctions.”
Specialty Patient Registry Project

- Principle: Need to prove the quality of care we deliver, not just assert it
- Directive: work with each of 30+ specialties to define conditions and quality measures
- Most registries require:
  - EHR data collection tools
  - Clinical decision support tools
  - Registry reports: care gaps, performance measures
Our Story: Problems
Challenges faced in creating Clinical Decision Support tools in general

- Differing understanding of intended solution among stakeholders

- Complexity:
  - Of clinical environment into which being released
  - Of solution elements → difficulty keeping track of all components

- Evolving requirements:
  - Release to production EHR yields new understanding of desired CDS tool behavior
Problems we faced on this project

- Our CDS Tool and Registry development track record historically was poor
  - long turnaround time
  - unintended behavior in production (defects)
- Large number of high-quality CDS tools needed
- Short time-line, so needed all hands on deck, not just current experts
- Need for shared communication about tools being built
  - Among analysts/developers
  - With customer
Our Story: What we did

1. Adopted agile project management (APM) principles and methods
2. Adopted agile modeling (AM) principles and methods
3. Began gaining experience with test-driven development and with monitoring for rapid-cycle redesign
Why We Chose Agile Project Management for CDS

- promote *shared understanding* among multiple stakeholders
- support *rapid-cycle time-boxed development* of CDS tools
- promote agility in *responding to evolving requirements* as stakeholders interact with delivered EHR-based tools.
Benefits of Agile Project Management for CDS

- **Background using Agile at UT Southwestern**
  - Data Warehouse/BI Team: monthly sprints, quarterly releases
  - Electronic Medical Record (EMR) Team: 2-week sprints & releases

- **This project:**
  - 4 groups involved: Quality, Clinic Operations, EMR Team, Analytics
  - Chose EMR Team methodology

- **Key beneficial agile practices on this project:**
  - Time-boxed iterations
  - End-iteration demos
  - User Stories and acceptance criteria for requirements gathering
Lightweight Requirements Gathering with User Stories

- **Some advantages:**
  - Short, easy to read
  - Understandable to developers, stakeholders, and users
  - Focuses on customer and value to be delivered
**User Story**

As a specialist caring for patients with osteoporosis, **I want to** be alerted when such a patient is not on a bisphosphonate medication **so that** I can increase the percentage of my patients on optimal medical treatment for osteoporosis, and thus improve their bone health.

**Acceptance Criteria**

- Alert within EHR to prompt ordering of a bisphosphonate on patients with osteoporosis
- Alert fires for patients with either Osteoporosis or Osteopenia on their Problem List in the EHR (based on SNOMED)
- Alert does not fire if a bisphosphonate medication is already on their active medication list
- Alert allows designation of reasons why the patient is an exception
- Tipsheet to train on use
Agile Modeling

Why Model?
• To understand
• To communicate

Agile Modeling goal:
high value for degree of model creation effort expended

Agile Modeling core practices include:
• Active Stakeholder Participation
• Model With Others
• Apply the Right Artifact(s)
• Use the Simplest Tools
• Model in Small Increments
• Create Simple Content
• ...

Copyright 2003-2006 Scott W. Ambler
aKama
GEORGIA
July 25-29, 2016
Domain Model (Class Diagram) for Osteoporosis Registry

Purpose:
Define “nouns” of system—depict the major terms and concepts (objects, classes) relevant to the project, and show how they are related to each other.
Purpose:
Catalog “verbs” of system—depict the major activities people in various roles can accomplish when using the system.
Decision Tree for a Clinical Alert

Purpose:
Unambiguously define the logic ("business rules") involved when evaluating one or more input conditions to derive a result, such as whether or not to display an alert to a physician or nurse.
Purpose:
Provide a chance to anticipate what the system will look like, and to visualize interacting with it when reading a Use Case Text, described next.
Solution Object Diagram

Purpose:
Depict the software objects (records) to be built or employed in our EHR, and how they relate to one another, as a road-map for understanding the detailed design.
Agile Modeling Summary

- Complexities of EHR-based Clinical Decision Support Tools can result in misunderstood requirements and product defects.

- Requirement changes—especially for CDS tools—are frequent and inevitable, as users experience them in their workflow.

- “Agile modeling” can foster understanding and shared communication about CDS requirements and design, at a relatively low cost in time/effort.

- A small subset of model/diagram types can prove most helpful for designing CDS tools.
What we did:
Exploration of Acceptance Test-Driven Development and Monitoring
CDS Acceptance Test Suite: Pre-Build

Test System: fit:D:\fitnesse\fitsharp\Runner.exe

Specify a CDS Alert to display if a patient with CKD lacks correctly-staged CKD on their Problem List

<table>
<thead>
<tr>
<th>Query</th>
<th>Alert Name</th>
<th>Alert ID?</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTSW CKD DX WITHOUT CORRECT CKD STAGE BASE missing</td>
<td>Alert Name</td>
<td>Alert ID?</td>
</tr>
</tbody>
</table>
CDS Acceptance Test Suite: Initial Build

Test System: fit:D:\fitnesse\fitsharp\Runner.exe

Specify a CDS Alert to display if a patient with CKD lacks correctly-staged CKD on their Problem List

<table>
<thead>
<tr>
<th>Query</th>
<th>Alert Name</th>
<th>Alert ID?</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTSW CKD DX WITHOUT CORRECT CKD STAGE BASE</td>
<td>Alert ID?</td>
<td>1589</td>
</tr>
</tbody>
</table>
Specify restrictions on when this alert should apply:

<table>
<thead>
<tr>
<th>Query</th>
<th>SELECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Encounter Type?</td>
</tr>
<tr>
<td>1</td>
<td>Office Visit</td>
</tr>
<tr>
<td>2</td>
<td>Office Visit</td>
</tr>
<tr>
<td></td>
<td>Nephrology actual</td>
</tr>
</tbody>
</table>

Specify what triggering actions should prompt evaluation of the alert logic:

<table>
<thead>
<tr>
<th>Query</th>
<th>SELECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triggering Action?</td>
<td></td>
</tr>
<tr>
<td>General BPA section</td>
<td></td>
</tr>
<tr>
<td>Enter order</td>
<td></td>
</tr>
<tr>
<td>Open Orders</td>
<td></td>
</tr>
<tr>
<td>Sidebar surplus</td>
<td></td>
</tr>
</tbody>
</table>
CDS Acceptance Test Suite: All Passing

Test System: fit:D:\fitnesse\fitsharp\Runner.exe

Specify a CDS Alert to display if a patient with CKD lacks correctly-staged CKD on their Problem List

Specify Alert Name (ID will display when record is built)

<table>
<thead>
<tr>
<th>Query</th>
<th>Alert Name</th>
<th>Alert ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTSW CKD DX WITHOUT CORRECT CKD STAGE BASE</td>
<td>Alert ID?</td>
<td>1589</td>
</tr>
</tbody>
</table>
Test-Driven Development for CDS: Conclusions

- “Manual acceptance testing is expensive, error prone, and not repeatable”
- FitNesse or other tools can be used to create and execute table-based acceptance tests.
- Table-based acceptance tests can be written by analysts, and imported into the testing tool.
- A single automated table-based test serves 4 purposes:
  - Requirements specification
  - Automated acceptance test
  - Automated regression test once in Production
  - Documentation of design and business rules (guaranteed to be in synch with actual production implementation)

http://www.slideshare.net/ajitkoti/fitnesse-testing-framework
Goals of CDS Monitoring

- Help detect Anomalies (defects) in CDS behavior, e.g.
  - Alerts firing inappropriately
  - Alerts not firing when appropriate

- Opportunities for CDS tool refinement
  - Assess clinicians’ actions in response to CDS
  - Assess clinician burden – reduce risk of “alert fatigue”
  - Focus rapid-cycle redesign
Example of CDS Alert Monitoring

Encounter Type: Hospital Encounter
Background Fire YN: N
Pt Dept Type: IP Nursing Unit
Proc Category: Nurse
Filter Reason: Shown to User
Alert Date, Year-Mo: 2016-03
Alert Name: TRAVEL AND EXPOSURE SCREENING NOT PERFORMED
Alert Trigger Category: Interruption

Triggered Qty

<table>
<thead>
<tr>
<th>Specific Override Reason</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defer</td>
<td>32,137</td>
</tr>
<tr>
<td>Agree</td>
<td>23,604</td>
</tr>
<tr>
<td>No Associated Reason</td>
<td>3,752</td>
</tr>
<tr>
<td>Information Not Available</td>
<td>511</td>
</tr>
<tr>
<td>Follow-up Action Not Taken</td>
<td>343</td>
</tr>
<tr>
<td>Grand Total</td>
<td>60,347</td>
</tr>
</tbody>
</table>

29
Results
Results to-date: Quantitative

<table>
<thead>
<tr>
<th>Item</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specialties</td>
<td>30</td>
</tr>
<tr>
<td>Registries</td>
<td>43</td>
</tr>
<tr>
<td>EHR Tools Built</td>
<td>111</td>
</tr>
<tr>
<td>Measures</td>
<td>163</td>
</tr>
<tr>
<td>Dashboards</td>
<td>32</td>
</tr>
<tr>
<td>Unique patients tracked on registries</td>
<td>&gt;60,000</td>
</tr>
<tr>
<td>Patient-reported outcome questionnaires completed</td>
<td>&gt;2,100</td>
</tr>
</tbody>
</table>

Project recognized with Healthcare Informatics magazine's Innovator Award for 2016, published at:
What We Learned:

- Agile methods are effective in CDS development
  - Value of time-boxed, iterative development
  - Value of “user stories” and acceptance criteria
  - Agile modeling → shared understanding

- A subset of diagrams proved most useful for CDS development

- Automated testing and TDD possible using an open-source tool (FitNesse)

- Monitoring CDS activity useful for rapid-cycle redesign
Acknowledgements

- Our wonderful Experience Report shepherd: Sue Burk
- Our UT Southwestern colleagues:
Email us at:

- Vaishnavi.Kannan@UTSouthwestern.edu
- Duwayne.Willett@UTSouthwestern.edu

Thank you!