



HHS Public Access

Author manuscript

Health Innov Point Care Conf. Author manuscript; available in PMC 2018 November 01.

Published in final edited form as:

Health Innov Point Care Conf. 2017 November ; 2018: 56–59. doi:10.1109/HIC.2017.8227583.

Agile Co-Development for Clinical Adoption and Adaptation of Innovative Technologies*

Vaishnavi Kannan, Mujeeb A. Basit, Josh E. Youngblood, Trenton D. Bryson, Seth M. Toomay, Jason S. Fish, and Duwayne L. Willett

University of Texas Southwestern Medical Center, Dallas, TX 75390 USA

Abstract

Even the most innovative healthcare technologies provide patient benefits only when adopted by clinicians and/or patients in actual practice. Yet realizing optimal positive impact from a new technology for the widest range of individuals who would benefit remains elusive. In software and new product development, iterative rapid-cycle “agile” methods more rapidly provide value, mitigate failure risks, and adapt to customer feedback. Co-development between builders and customers is a key agile principle. But how does one accomplish co-development with busy clinicians? In this paper, we discuss four practical agile co-development practices found helpful clinically: (1) User stories for lightweight requirements; (2) Time-boxed development for collaborative design and prompt course correction; (3) Automated acceptance test driven development, with clinician-vetted specifications; and (4) Monitoring of clinician interactions after release, for rapid-cycle product adaptation and evolution. In the coming wave of innovation in healthcare apps ushered in by open APIs to EHRs, learning rapidly what new product features work well for clinicians and patients will become even more crucial.

I. Introduction

Even the most innovative healthcare technologies provide patient benefits only when adopted by clinicians and/or patients in actual practice. When developing a new technology or feature, how can we optimize clinician and patient initial adoption, and continue to adapt to their complex, evolving environment, in order to help achieve the greatest positive impact on health? In new product development, iterative rapid-cycle “agile” methods prove more effective than a traditional “waterfall” approach. Such products provide value earlier, mitigate risks sooner, and incorporate customer/user feedback to iteratively adapt and evolve, becoming more fit for real-world use. Agile methods should be equally applicable to innovative healthcare technology development projects. Co-development between builders and customers is one key agile concept for iteratively better meeting customer/user needs. But how does one accomplish co-development with busy clinicians, and adaptively develop products willingly adopted into their practices? In this paper, we propose 4 agile practices helpful clinically, and demonstrate their feasibility and usefulness.

*Research supported by NIH Grant: 5UL1TR001105-05 UT Southwestern Center for Translational Medicine
Corresponding author phone: 214-648-1303; fax: 214-648-1033; Vaishnavi.Kannan@UTSouthwestern.edu).

A. Agile Methods for New Product Development

Since the publication of the Manifesto for Agile Software Development in 2001 [1], agile practices have come to dominate software development [2]. Large-scale evidence now supports long-standing anecdotal support that such practices work better than traditional prescriptive project management approaches, across all sizes of software projects [3]. Agile methods work better than traditional waterfall development primarily in complex situations, where the optimal solution and product design is not knowable at the outset, and for new product development in general, not just software [4], [5]. Agile methods have now been applied to EHR-based and other healthcare innovation [6–8].

Co-development with customers is one of the core values in the Agile Manifesto (Fig. 1):

One challenge in healthcare innovation is achieving “customer collaboration” with often-busy clinicians and with patients.

B. Agile Co-Development with Clinicians: 4 Practices

Collaborative co-development can be fostered by multiple techniques, not mutually exclusive, which can play complementary roles during product evolution. Four techniques potentially useful with clinicians include:

- “User Stories” for lightweight requirements [9].
- Time-boxed development with available clinicians, for collaborative design, rapid feedback and early course correction [5].
- Automated acceptance test driven development, with clinician-understandable “executable requirements” and detailed specifications [10].
- Monitoring of clinician interactions after release, for rapid-cycle product adaptation and evolution [11].

Though separate, the techniques offer synergies. For instance, short time-boxed iterations (1 or 2 weeks) become feasible through use of lightweight initial requirements, further elaborated through customer collaboration in creating automated acceptance tests as more detailed specifications. Also, adding automated acceptance tests to grow a regression test suite provides a “safety net” for subsequent rapid-cycle product adaptation driven by ongoing monitoring of clinician behavior.

In this paper, we use examples to empirically demonstrate feasibility and usefulness of these candidate practices.

II. Methods

A. Location and Clinicians

All activities in this report took place at the University of Texas Southwestern Medical Center in Dallas, Texas. Clinicians involved in agile co-development were all faculty or house staff practicing in one of UT Southwestern’s hospitals and clinics. This work was determined not to be human subjects research and not to require IRB oversight.

B. Procedures

1) User Stories—Lightweight initial feature requirements were elicited with User Stories, using the common template of “As a <type of user>, I want <something>, so that <benefit to be achieved>” [9, 12, 13]. The user stories were consistently written in the voice of the clinician or patient interacting with the feature, and not in the voice of the feature’s requestor or builder. That is, the <type of user> denotes the type of clinician or patient interacting with the feature, and the <benefit to be achieved> represents a benefit appreciated by that clinician or patient. For clinicians, this benefit was framed as directly to them (e.g. improved efficiency), to their patients (e.g. improved care, outcomes), and/or to their organization (e.g. improved financial stability). Most often development team members initially drafted the user stories, then validated and revised them in conversations with the feature customer and product owner. With experience, some product owners and customers began submitting new feature requests in this “user story” format.

User Stories were elaborated with Acceptance Criteria as needed to unambiguously define what successful development of the feature would look like. Acceptance criteria were represented in the simplest possible way, commonly initially as bulleted lists, but occasionally augmented with more detailed specification formats and/or manual or automated acceptance tests (see Results).

2) Time-Boxed Iterations and Iteration Reviews—Two-week iterations were employed [5], with end-iteration reviews including demos of working product features developed during the iteration, attended by members of the team and by customers or their representatives. For clinician-facing features, clinician accessibility determined the iteration in which that user story would be scheduled, so the clinician(s) could clarify design issues during development, and provide clinical feedback at the end-iteration demo and feature review. Agile modeling with customer collaboration was used selectively for detailed design decision-making during iterations [14, 15]

3) Automated Acceptance Test Driven Development—Acceptance tests were written in table format, listing expected system output(s) for each specified set of inputs. Tables were constructed with reader-friendly terms, so that the specifications could be vetted with clinicians. Table-based tests were implemented in FitNesse (see Apparatus and Software) as part of automated testing suites [16]. Tests were expected to fail initially (be “red”) before work on the feature started, then pass (turn green) in the development and testing environments once work was successfully completed [17–19]. Following release of the feature, the same test was added to the automated regression test suite for the production environment. Any subsequent failures of regression tests in production triggered a new entry in the incident management system for investigation and resolution.

4) Monitoring of Clinician Interactions—Clinician and patient interactions with EHR-based features are logged in our electronic health record, and available in standard relational database tables for querying and analysis. We constructed refreshable data cubes and graphical interactive displays of clinician interactions with EHR features (e.g. clinical decision support tools), for analysis of trends in clinician response and comparisons among

sub-groups. Well-described iterative methods for development of dimensional models and analytics were employed [20, 21]. Analyses of clinician interactions were made available for decisions to adaptively evolve the configuration of EHR features, to retire features for ineffectiveness, and to broaden the scope of more effective feature configurations.

C. Apparatus and Software

User stories were documented initially on paper cards or as simple text files. For online management of many user stories and iterations across groups, we used agile project management software: CA Agile Central (formerly Rally), from Computer Associates. Automated testing employed the open-source testing software FitNesse, based on the Framework for Integrated Testing (Fit), along with the dbFit extension for querying databases [16, 22, 23]. Electronic health record software at UT Southwestern is from Epic (Verona, WI). Data querying, analysis, and presentation were performed with SQL Server, Analysis Services, and Power BI, all from Microsoft (Redmond, WA).

III. Results/Examples

The following sections provide example applications of each of the four techniques, with several spanning development of a new clinical decision support (CDS) alert.

A. User Stories

Regardless of development methodology, “doing the right thing” remains at least as important as “doing things right”, and requirements remain relevant. User stories provide a lightweight way to elicit requirements, succinctly capturing the “who, what, and why” for a product or product feature. More detailed acceptance criteria can be elaborated, even just-in-time prior to beginning development.

An example of a high-level user story for an entire EHR-based registry “product” is:

- *As a* physician caring for patients with kidney cancer, *I want* to know the status of each patient’s cancer including staging, recurrence, and/or progression, *so that* I can provide optimal clinical care to each individual patient, and continuously improve our overall care to this entire patient group.

Acceptance Criteria for the entire registry were initially displayed as a “Feature Breakdown Structure” showing the component features envisioned as defining successful staged delivery of the product (Fig. 2). Individual features were then elaborated with their own user stories.

An example of a more detailed feature-level user story for a CDS alert to prevent spinal hematoma in patients with a post-surgical indwelling epidural catheter for pain control is:

- *As a* hospital physician caring for post-operative patients, *I want* to be alerted when ordering an anticoagulant or antithrombotic medication for a patient with a post-operative epidural catheter, *so that* my patient avoids risk for developing a spinal hematoma and resulting neurologic impairment.

Acceptance criteria here are in the form of a bulleted list, co-formulated by the clinicians involved and the EHR team:

- Alert fires for anyone with Order Entry privileges, in Hospital Encounters only (inpatient, observation, and ambulatory settings)
- It should query for EHR documentation of either presence of the epidural catheter OR any active medication ordered to infuse via epidural.
- In those patients, it should then look for the presence of any anticoagulant except a) BID or TID 5000 units SQ heparin alone OR b) aspirin alone. If both of those two meds are present, however, still fire the alert.
- It should then warn the provider ordering the anticoagulation that this form of anticoagulation is incompatible with an epidural catheter and direct them to contact the Pain Management Service.

B. Time-Boxed Iterations: Clinician Availability and Iteration Reviews

During 2-week iterations, the full cycle of detailed design, build, and test is performed, with a demo of working, tested features at end-iteration. Availability of the clinician enables detailed design questions to be addressed, helping ensure what is built is what is desired. In the case of the epidural alert, involvement of the clinician was crucial in getting the decision logic precisely correct for when to display the alert, given the discrete fields available in the EHR (Fig. 3).

C. Automated Acceptance Test-Driven Development

Creating detailed specifications in the form of testable serves multiple purposes, as: (1) detailed specifications of what should be built, (2) an automated acceptance test to confirm successful build, (3) part of a regression test suite to guard against unintended future changes, and (4) online documentation of the design. In acceptance test-driven development, tests initially fail before the feature is built, then pass when built successfully, as shown for the epidural-anticoagulation alert (Fig. 4 and Fig. 5).

Automated acceptance tests during co-development with clinicians can be written to confirm the intended user interface design (e.g. wording, available actions) in addition to the underlying business/clinical logic.

D. Monitoring of Clinician Interaction

Software is rarely perfectly fit for use upon first release, but the best software evolves over time to achieve maximal effectiveness. Monitoring and assessing clinician response to new products and features in production provides insights into what designs work best. Alerts are monitored for firing frequency (e.g. by provider type and setting), and clinician response choices when alerts are presented (Fig. 6).

IV. Discussion

Agile co-development holds promise for helping optimize clinician and patient adoption of innovative technologies, and fostering rapid-cycle adaptation to maximize value in practice. Our central finding is that 4 agile co-development techniques have proven feasible to engage clinicians in iterative feedback within the new product development cycle: user stories, time-

boxed iterations with clinician demos of working features, automated acceptance testing, and monitoring of clinician real-world usage patterns.

The rise of standards-based open APIs (such as FHIR) for interacting with EHR data foretells a coming wave of innovation in healthcare apps and devices: learning rapidly what features work well for clinicians and patients (or not) will thus become even more crucial.

The goal of virtually all healthcare innovation is, at some level, improving people's health and quality of life. Realizing optimal positive impact from new technologies for the widest range of individuals who would benefit remains elusive. Making progress means solving for increasing adoption of beneficial new features by appropriate patients and clinicians, and rapidly adapting to their needs. Practical agile co-development techniques can help.

References

1. Beck K, , et al. Manifesto for Agile Software Development. Accessed on: 6/14/2010. Available: <http://agilemanifesto.org/>
2. Agile vs. waterfall: Survey shows agile is now the norm. 2017. Available: <https://techbeacon.com/survey-agile-new-norm>
3. Hastie SW, Stephanie. Standish Group 2015 Chaos Report – Q&A with Jennifer Lynch. 2015. 7/24/2017 Available: <https://www.infoq.com/articles/standish-chaos-2015>
4. Larman C. Agile and Iterative Development: A Manager's Guide. Pearson Education; 2003.
5. Rubin KS. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley Professional; 2012.
6. Hagland M. The 2016 Healthcare Informatics Innovator Awards: First-Place Winning Team—UT Southwestern Medical Center. 2016. Available: <https://www.healthcare-informatics.com/article/2016-healthcare-informatics-innovator-awards-first-place-winning-team-ut-southwestern>
7. Kannan V, et al. Rapid Development of Specialty Population Registries and Quality Measures from Electronic Health Record Data. An Agile Framework. Methods of Information in Medicine. 2017; 56:e74–e83. Open.
8. Flood D, et al. Insights into Global Health Practice from the Agile Software Development Movement. Glob Health Action. 2016; 9:29836. [PubMed: 27134081]
9. Cohn M. User Stories Applied: For Agile Software Development. Addison Wesley Longman Publishing Co., Inc; 2004.
10. Adzic G. Specification by Example: How Successful Teams Deliver the Right Software. Manning Publications Co; 2011.
11. Liu S, Wright A, Hauskrecht M. Conference on Artificial Intelligence in Medicine in Europe. Springer, Cham; 2017. Change-Point Detection Method for Clinical Decision Support System Rule Monitoring; 126–135.
12. Adzic G, Evans D. Fifty Quick Ideas to Improve Your User Stories. Neuri Consulting LLP; 2014.
13. Patton J, Economy P. User Story Mapping: Discover the Whole Story, Build the Right Product. O'Reilly Media, Inc; 2014.
14. Ambler SW. Agile modeling: effective practices for extreme programming and the unified process. John Wiley & Sons, Inc; 2002.
15. Gottesdiener E, Gorman M. Discover to Deliver: Agile Product Planning and Analysis. EBG Consulting, Inc; 2012.
16. Martin R, Martin M, Wilson-Welsh P. F. contributors. What are the advantages of FitNesse automated acceptance tests?. (7/24/2017) Available: <http://fitnesse.org/FitNesse.UserGuide.AcceptanceTests>
17. Freeman SP, Pryce N. Growing Object-Oriented Software, Guided by Tests (The Addison-Wesley Signature Series). Addison-Wesley; 2009.

18. Crispin L, Gregory J. Agile Testing: A Practical Guide for Testers and Agile Teams. Addison-Wesley Professional; 2009.
19. Gregory J, Crispin L. More Agile Testing: Learning Journeys for the Whole Team. Addison-Wesley Professional; 2014.
20. Collier KW. Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing. Addison-Wesley Professional; 2011.
21. Hughes R. Agile Data Warehousing Project Management: Business Intelligence Systems Using Scrum. Morgan Kaufmann Publishers Inc; 2012.
22. Adzic G. Test Driven .NET Development with FitNesse. Neuri Limited; 2008.
23. Benilov J. DbFit: Test-driven database development. Write readable, easy-to-maintain unit and integration tests for your database code. 2015. 8/1/2017 Available: <http://dbfit.github.io/dbfit/>

"...we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- **Customer collaboration** over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more." [1]

Figure 1.
Co-development in the Agile Manifesto

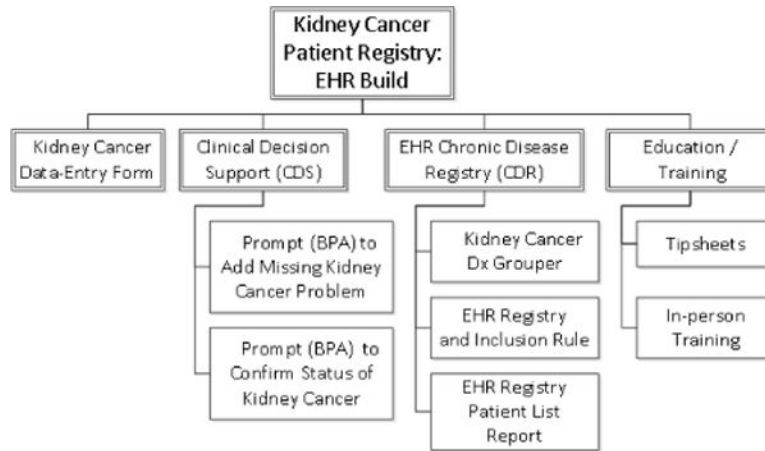


Figure 2. Acceptance Criteria (components) for an entire patient registry

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

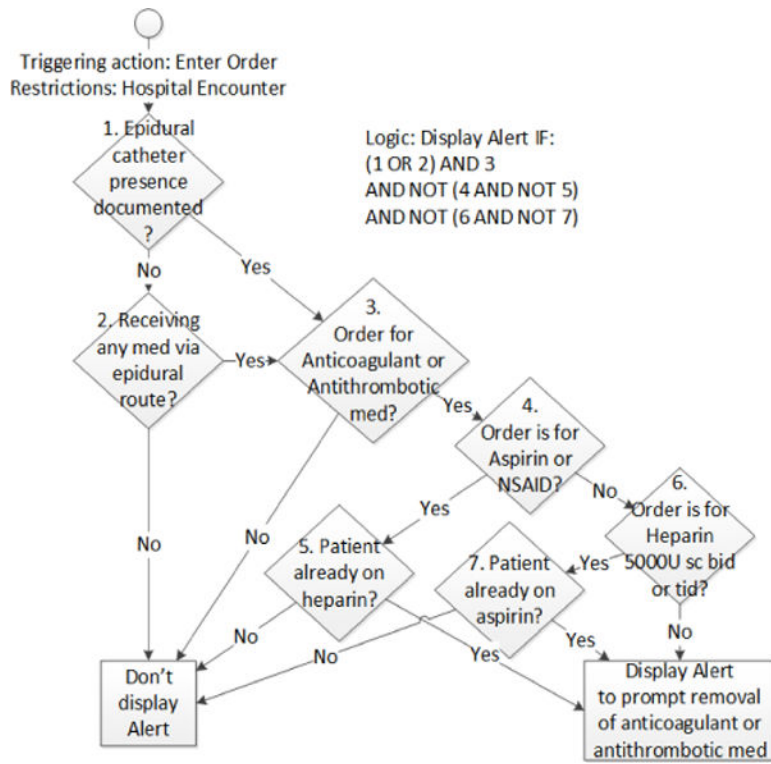


Figure 3. Clinician-elaborated decision tree for an alert to prevent epidural catheter-associated spinal hematoma

Assertions: 0 right, 6 wrong, 0 ignored, 0 exceptions (0.849 seconds)

Specify linked criteria used to evaluate this alert:

Line	Linked Criteria Name?	Linked ID
1 <i>missing</i>	UTSW ANES HAS ACTIVE EPIDURAL LDA CRITERIA	
2 <i>missing</i>	UTSW ANES RECEIVING MED VIA EPIDURAL CRITERIA	
3 <i>missing</i>	MED ENTERED IS ANTICOAGULANT OR ANTIPLATELET	
4 <i>missing</i>	UTSW ANES EPIDURAL ANTICOAGULANT - ASA BUT NO SC HEPARIN EXCEPTION CRITERIA	
5 <i>missing</i>	UTSW ANES EPIDURAL ANTICOAGULANT - SC HEPARIN BUT NO ASA EXCEPTION CRITERIA	

Specify Boolean logic to combine above linked criteria:

Boolean Logic
(1 OR 2) AND 3 AND NOT (4 OR 5) <i>missing</i>

Figure 4.
Automated acceptance test for the epidural anticoagulant alert – failing test prior to build

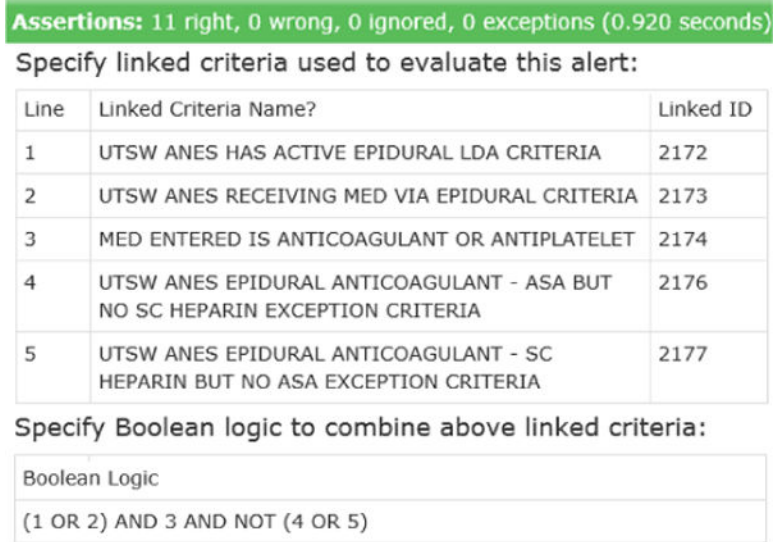


Figure 5.
Automated acceptance test – passing following successful build

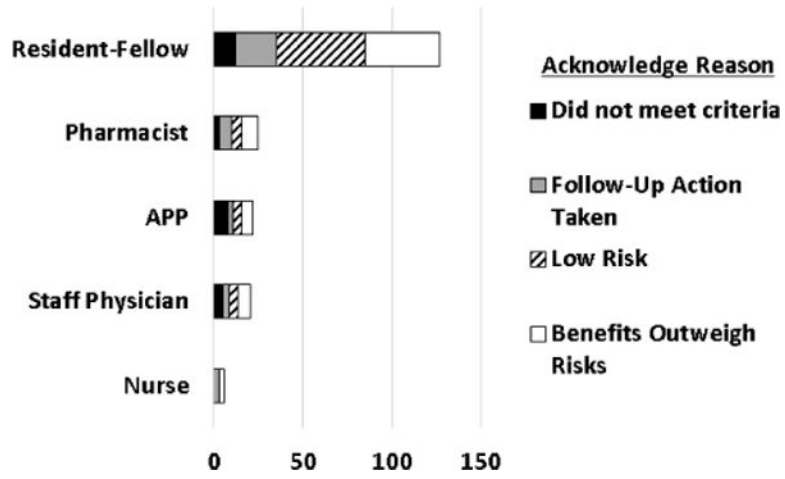


Figure 6. Monitoring clinician response to epidural anticoagulant alert – analysis of alert overrides by provider category and reason

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript