Case Study: Implementing Agile Software Development Processes in a Healthcare IT Organization –

Approach and Lessons Learned

By

Huma Sohrwardy, PMP, CSM

Introduction

This paper describes the approach and implementation strategy that the MedAssets’ Technology Project Management Office (PMO) undertook to implement Agile software development methods within the Information Technology (IT) organization. It discusses the current implementation state and lessons learned from the implementation process.

MedAssets partners with healthcare providers to improve their financial strength by implementing revenue cycle, spend and clinical resource management solutions that help capture revenue, control cost, improve margins and cash flow, increase regulatory compliance, and optimize operational efficiency. MedAssets serves more than 180 health systems, 4,000 hospitals and 90,000 non-acute healthcare providers. The company currently manages $45 billion in supply spend and touches over $316 billion in total patient revenue annually through its revenue cycle solutions.

It is important to note that the implementation of Agile methods, customer satisfactions ratings and the lessons learned are related to the IT group responsible for providing technology solutions to support healthcare supply chain and Group Purchasing Office (GPO) solutions.

Until late 2008, the PMO used a Waterfall Software Development Life Cycle (SDLC) methodology on all technology projects. One of the Key Performance Indicators that projects were measured against was the Customer Satisfaction rating. A scale of 1 – 5 (1 being low) was used to measure the level of customer satisfaction after the solution was deployed. The target threshold was set at a rating of 4 and above.

The Waterfall model is a sequential software development process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Feasibility, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.

The Waterfall SDLC was repetitive and consistent but the IT organization was experiencing solution delivery delays which had an adverse impact on perceived ‘value’ to business and low customer satisfaction ratings (Figure 1). Between 2007 and 2008, customer rating was well under the target threshold of 4.
Key reasons for this were:

- IT took a long time to deliver projects and when they were delivered, the outcome did not meet business user expectations.
- Project requirements analysis phase took a long time to complete and by the time solutions were delivered, business needs had changed or key business stakeholders had moved on to other roles leaving a vacuum.
- After completion of requirements analysis, business stakeholders had to wait until the product had been developed to the see the end result. This resulted in ‘expectation gaps’ in what the business stakeholders had specified versus what was delivered as the final solution.

To address these issues a phased approach was used to improve and streamline software development processes within the IT organization. Key improvement areas identified were:

- Provide business stakeholders with the opportunity to visually see what was being developed earlier in the project’s life cycle instead of waiting till the end.
- Quicken the product’s time to market.

In short, focus on and deliver optimized ‘value’ to business and improve customer satisfaction.

![Average Customer Satisfaction Rating](image)

*Figure 1. Avg. Customer Satisfaction Rating, 2007-2008, © MedAssets*

**Approach and Implementation Strategy**

A phased approach was used to implement process improvements in the SDLC methodology. It spanned a period of 14-18 months. In Phase One, which commenced in early 2009, User Centered Design activities were incorporated within the Waterfall methodology. This was followed by Phase Two, in which a formal Agile development methodology was rolled-out within the IT organization.

Agile software development is based on iterative and incremental development in which requirements and solutions evolve through collaboration between self-organizing and cross-functional teams.

**Phase One**

Collaborative design workshops were introduced in the SDLC, prior to requirements gathering, to build ‘mock ups’ and prototypes of the product being developed, supporting User Centered design concepts. These changes were driven by the user experience team.
User Centered Design is a design philosophy and a process in which the needs, wants, and limitations of end users of a product are given extensive attention at each stage of the design process.

The goal was to determine feasibility of the proposed solution up front and provide the business stakeholders with an opportunity to visualize the end result prior to development. Agreement from business triggered finalization of requirements and creation of the ‘product backlog’. In addition to collaborative design workshops, the SDLC process was tweaked to support iterative development cycles. The PMO’s role was to facilitate discussions between the user experience and development teams and socialize the revised SDLC with the IT organization and business stakeholders.

Over a period of 12 months, this resulted in an improvement in customer satisfaction and overall increase in delivery times. The customer satisfaction rating increased and for the first time it hit the target threshold of 4 (Figure 2).

By Q2 2010, the push to move to Agile methods had gained momentum within the development group which started using Scrum activities on some technology projects. Scrum is an Agile framework for completing software development projects.

The most common Scrum activity being used by teams was daily stand-up meetings and the use of Agile artifacts such as information radiators, product backlogs, user stories and burn down charts. The process, terminology and artifacts being used were not consistent and their meaning and format depended on the team’s understanding of Agile methods. However, the inconsistency in process did not have an adverse impact on project outcomes as solutions continued to be delivered incrementally to the business.

This was the precursor to implementing formal Agile software development processes within the IT organization.

![Average Customer Satisfaction Rating](Figure 2. Avg. Customer Satisfaction Rating 2007-2009, © MedAssets)

**Phase Two**

To formally implement an Agile SDLC methodology within the IT organization, the PMO facilitated the setup of a key stakeholder group that jointly defined and formalized the Agile software development process. The group comprised of software development, architecture, business analysis, quality assurance, user experience, and project management representatives. The group also utilized the services of experienced Agile consultants who provided input on Scrum methods.

The group’s goals were to define a lean software development methodology and at the same time, have necessary controls that met IT Governance needs. The role of the PMO was to:
a) Facilitate the development of a SDLC methodology using Agile methods, incorporating key internal requirements. These were:

IT Governance
- Group ‘themes’ by project and track scope, budget and schedule for a project;
- Implement a leaner project change management process;
- Ensure sufficient control and monitoring measures; and
- Account for project activities related to software release management, production support handover, architecture reviews, etc.

Lean SDLC methodology
- Reduce SDLC checkpoints and artifacts;
- Incorporate a leaner process without compromising key activities such as collaborative design, project startup, and regression, performance and load testing; and
- Incorporate a flexible project team role and responsibilities structure.

b) Manage communication and training;

c) Socialize new methodology with senior IT management and get their sign off/support; and
d) Monitor how the process was being implemented and incorporate improvements and efficiencies.

The key stakeholder group met over a two month period to develop the Agile SDLC process (Figure 3). Refer to Appendix A for details.

The group agreed to incorporate Scrum principles in the Agile development process and tailored it to meet internal requirements. The overall atmosphere was positive and teams were supportive of the change. There were concerns, however, about lack of internal experience and training in using Agile methods which also needed to be addressed.

The roles and responsibilities matrix was kept simple representing the three Agile roles i.e. product owner, team and Scrum master. Project Managers transitioned into the Scrum master role.

At the end, the PMO team documented the process, published the process and project artifacts and got final sign off from stakeholder group and senior IT management. Initial training was provided to 80 plus IT staff. This was followed up by refresher training for teams as they transitioned to Agile SDLC at a later date.

Non IT based training was deferred to a later date. The rationale was to provide internal training to IT staff first and later focus on training product owners (business stakeholders). The focus was to standardize internal IT process first.

For the initial roll-out, a group of 2-3 pilot technology projects was selected. These were projects that had not started and lent themselves to Agile development, i.e. where requirements were not clearly defined, solutions had substantial User Interface (UI) components and did not rely heavily on business process reengineering.

The overall scope of the pilot group projects was to provide healthcare spend analytics, group purchasing office functionality, and workforce platform integration solutions.
Experienced Scrum masters (external consultants) were assigned on key pilot projects to mentor the teams and help them through the Agile SDLC. This is driven by resource availability.

The PMO, in collaboration with the Enterprise Architecture team, monitored and interviewed pilot teams on the process. In addition, sprint lessons learned were captured and evaluated for implementing process improvements.

Key IT team members formally attained Scrum master certification after pilots were commenced. The rationale for delaying external training was so that the team members could first get Agile SDLC prior experience prior to getting certified. Project managers, development and business analysis team leads were certified. The aim was to internally build a base of Scrum masters.

By the end of 2010, the Agile SDLC methodology had been formalized, process related training provided to IT staff, and SDLC process rolled out on pilot projects and monitored. In addition, the IT organization had an internal base of certified Scrum masters.
The 2010 customer satisfaction results continued to grow positively and based on business stakeholder interviews, key factors that led to improved ratings were reduction in time to market and delivery of IT solutions were more closely aligned with strategic business goals (Figure 4). Since 2007, there has been an increase of approximately 80% in overall customer satisfaction ratings.

While there is no conclusive evidence that the implementation of a formal Agile SDLC methodology is the key factor in improved customer satisfaction ratings per se, there is a correlation between the ratings and the use of incremental development cycles.

**Current Status**

As a result of organizational restructure, other IT groups have merged with the Spend & Clinical Resource Management Segment. Today over 80% of technology projects underway within the segment are using Agile methods. Rally² is being used by some IT groups to manage the Application Lifecycle Management (ALM). Plans are underway to get all the Spend & Clinical Resource Management segments’ IT groups on the Rally platform by Q3 2011.

The PMO introduced internal product owner training in Q1 2011 to socialize the Agile SDLC methodology and the level of commitment needed.

The offices in Plano, Texas have three dedicated Scrum rooms which have helped increase team collaboration. Each room is large enough to be shared by two Scrum teams, each comprised of 8-10 members. Video conferencing equipment allows face to face daily Scrum meetings with geographically disparate team members. Experienced Scrum masters continue to mentor and coach team members.

The project change management process is leaner within the Agile methodology and is used only to document and get approval for major changes in project scope (or “themes”). In the Waterfall SDLC methodology, the project change management process was used extensively to document changes to project spend, schedule and scope and get formal approval prior to project re-baselining.

From a financial reporting perspective, Scrum masters are required to track capital / operational estimates, spend and forecasts. Per software capitalization rules, capital reporting is broken by Release to allow the Finance team to amortize capital spend as soon as working software is deployed in production within the project’s life cycle.
Lessons learned

Some of the key lessons learned are:

a) Strong business product owners who clearly understand their role and responsibilities and fully commit to process are an important factor in the successful and timely outcome of Agile projects. This group was overlooked in the initial rollout in 2010.

As a result, projects experienced temporary delays in getting user stories confirmed, reviewed and signed off and delays in backlog grooming. IT business analysts took on the product owner role to fill the gap which did not always address the core issue. The implementation approach should have included product owner training and communication for more effective results.

b) A dedicated experienced resource is necessary to coach teams in using Agile methods. This resource was not consistently assigned to all projects initially and depended on availability. In some cases this has resulted in:

- Teams members not fully understanding the meaning of ‘accountability’, ‘self management’, ‘commitment’, etc.;
- The principles of ‘continuous integration’ not being applied consistently across all projects;
- The lack of tight coupling between the development and quality assurance teams as they continue to work in silos resulting in a ‘Waterfall’ like approach to testing within sprints; and
- Scrum masters continuing to use a ‘command and control’ approach instead of taking on a ‘facilitator’ role.
- Product owners (business) not fully understanding the meaning of commitment and inadvertently becoming road blocks within the process.

Conclusion

While the process to implement and effectively use Agile SDLC has brought about positive change within the IT organization and has met minimal resistance, it has faced challenges and continues to do so today.

It is a slow process to bring about behavioral and organizational change within teams to embrace self management and empowerment. The Scrum masters still need to frequently put on ‘project manager’ hats and drive outcomes. The PMO recognizes that the change from a ‘command and control’ model to that of self empowerment does not happen overnight and the transition has to be managed through continuous mentoring and coaching of team members.

The Product owner role is most challenging to fill but at the same time key in delivering successful outcomes. Even though the focus on training non IT product owners started in 2011, its absence has not been a showstopper in implementing Agile methods. Project teams that experienced product owner availability issues, addressed this by using IT business analysts to liaise with business stakeholders and firm up product backlogs. To move up the Agile maturity ladder, this role needs to developed within the organization.

The uptake of Agile methods continues to grow, from a software development approach it is gaining foothold as a formal project management framework. The Project Management Institute (PMI) recently
announced the PMI-ACP³ (Agile Certified Practitioner) certification pilot program. The objectives are to demonstrate an individual’s level of professionalism in Agile practices of project management.

The technology PMO will continue to support both Agile and Waterfall software development methodologies within the IT organization. While the application of a consistent SDLC methodology is important, process should not drive outcomes.
Appendix A: Agile SW Development Process

This section provides a high level description of key components of the SDLC. Strategic Planning kicks of the project after approval is given to initiate the project.

1. Strategic Planning

   Strategic Planning consists of Vision and Roadmap Planning activities:

   a. Vision Planning
      Objective is to refine ‘Themes’ and low fidelity estimates, conduct high level risk assessment, determine high level technical architecture requirements and operational and functional / non-functional critical product constraints (performance, ease of use, volumes, etc.)

   b. Roadmap Planning
      Objective is to define the project roadmap, including development of release plan, refinement of architecture plan, prioritization of ‘Themes’ and determination of user experience (UX) strategy and timeframe.

2. Collaborative Design

   This is an optional activity. Following Strategic Planning the UX team can plan and conduct collaborative design workshops with business to develop prototypes.

3. Release Planning

   Objective is to refine Strategic Planning data and define a more detailed plan that will satisfy the product/project roadmap for the targeted release. The entire team will be involved in this exercise.

4. Sprint Planning

   Objective is to establish target sprint velocity, refine user story details, assign story points to user stories, break user stories into tasks (design, code, unit test, etc.) and provide team estimates in hours, make team commitment and broadcast on Information Radiator (IR).

5. Sprint Execution

   Objective is create working software, maintain working agreements and quality standards, hold daily Scrum meetings, deliver user stories, post sprint burn down charts and produce weekly status reports for projects stakeholders. In addition, groom the product backlog for future sprints.

6. Sprint Review

   Objective is to demonstrate each completed user story to the product owner and product owner accepts or rejects each user story.

7. Sprint Retrospective

   Objective is to facilitate a continuous improvement process for the team and determine if team met our commitments, what well or not well and what needs to change in next sprint.
References