

Creating a Lean, Mean Client Integration Machine

Aaron Akzin: Quality Assurance Analyst; Shelley Blouin: Manager of Software Development; Kenny Tran: Integrations Developer; Supriya Joshi: Quality Assurance Analyst; Sudha Sudunagunta: Quality Assurance Analyst; Aaron Medina: Quality Assurance Analyst

Abstract

Reorganizations are all too common in the modern business environment and in particular at technology organizations. This paper chronicles the motivation for undertaking such an effort. We are all aware of the disruption this will interject upon employees, clients and business objectives and the inherent risk in doing so. Despite this, often these actions are necessary. The intent of the authors to you, the reader, is that we detail what was done, how it was done and finally a retrospective about what was successful and what requires additional improvement efforts. WebMD Health Services is a rapidly changing and dynamic organization and we refuse to remain static and dormant which we adamantly oppose in our culture and as a trait in the employees we hire.

Biography

The authors of this Paper are all members of a team within the Integrations department that are under the umbrella of Technology at WebMD Health Services. We all serve the Health Care Market for our customers that are in this vertical.

Shelley Blouin is a Manager of Software Development.

Kenny Tran is an Integrations Developer on the team.

Supriya Joshi, Aaron Akzin, Sudha Sudunagunta and Aaron Medina are Quality Assurance Analysts on the team.

Collectively we have decades of experience in technology in all roles and enjoy our interactions in this new structure and are extremely interested in sharing our experience during the successful transition.

Copyright Aaron Akzin, Shelley Blouin, Kenny Tran Supriya Joshi, Sudha Sudunagunta and Aaron Medina 2011.

CREATING A LEAN, MEAN CLIENT INTEGRATION MACHINE

1. Introduction

In late 2009, WebMD Health Services (WHS) started a process to undergo a radical transformation in its organizational structure. This paper will be a retrospective of not only what went right and what went wrong but also the challenges teams faced during the transition towards a collaborative and cohesive environment.

Before the change, WebMD's client integrations division consisted of segregated development and QA teams. Minimum collaboration and lack of mutual understanding between the two teams resulted in duplication of efforts, finding defects late in the project cycle and delays in delivering enhancements on time to our clients. Realizing a need for increasing client satisfaction and delivering higher quality features within shortened timeframes, WHS decided to take a leaner approach and create a more integrated environment by adopting some of the more common Agile principles and molding them to support our processes and tight client scheduling limitations. While we tried many different techniques, the following are the principles that provided the most benefit to our client-driven development cycle:

- Shortening duration of job assignments
- Reducing the amount of work in the queue
- Creating small, combined and cross-functional teams of development and QA to allow for greater flexibility and better coverage depending upon priorities
- Introducing daily stand-ups and weekly sprints that were timed with deployment schedules
- Prioritizing tasks
- Introducing the role of Market Champions to facilitate Sprint planning and communication with client-facing staff

Throughout this transformation, a real sense of an often cited cliché of “Teamwork” has been enjoyed and evidence is abundant of how this has ultimately resulted in better products and client satisfaction both internal and external. This is often demonstrated on frequent occasions as the definitive delineation between developers and QA has been “blurred” as they pick up each other's tasks as needed in order to meet deadlines and lend expertise as needed.

2. About WHS

WHS is a consumer health solutions company that helps people make better health and benefits decisions, positively change their health behavior, and live healthier lives. We have a customized, integrated set of health management and consumer-guidance resources. Our online health portal is personalized for each individual and branded for their organization. The platform integrates content, tools, and programs from WebMD along with those that are specific to their population to help you drive consumer engagement, health, productivity, and satisfaction (Overview, 2011).

3. About The WHS Client Integrations Team

The client integrations teams collaborate with the client services teams to create online health portals that meet client needs. Clients may select any combination of tools and solutions we offer.

This story is about the journey of WHS from waterfall to an Agile(ish) Software Development Lifecycle (ASDL), responding to change when needed and finding ways to improve client satisfaction every single day (Fowler, 2005). We do not subscribe to any formal ASDL method. So this story is not about applying Scrum or Extreme Programming. It is about taking a leaner approach to software development lifecycle.

4. The way we were

Before March of 2009, WHS development teams followed a typical industry standard waterfall software development life cycle and we all fit nicely in either the development (Dev) or quality assurance (QA) teams. Some of the challenges we faced following a waterfall ASDL were:

- As is common, developers outnumbered quality assurance (QA) by approximately 3 to 1, yet often the verification tasks took longer to complete than the corresponding development tasks. When a project was late, QA stayed late.
- When a request came in for work, a project manager approved it. It became one of many in an endless personal queue for individual developer and QA contributors. There was an infinite array of “tracker” (ticket) items to complete that developers and QA worked on separately. They set their own priority to complete, as long as the production date was met. This meant that QA might be calling developers at home at 9:00 PM to fix a bug that had to go the next day.
- Each of our clients was assigned one developer and one QA in an effort to ensure that we had experts for each client. However, there were challenges with this approach if you ever wanted to go on vacation. Developer and QA were responsible for vacation coverage as part of their vacation planning.
- Communication happened through email and tracker items. Collaborating on a project meant creating a defect tracker for every issue found or questions that needed to be answered. These items would not only burn cycles for developers and QA staff, but also administrative personnel such as Project Managers and Managers that would need to approve, schedule, and prioritize every item that was submitted.
- Most challenging was the long list of rules of engagement between the two teams. It was dogma that developers did not test and QA would conversely never touch the development environment.
- QA and testing starts only when development is complete.

5. What prompted a change?

At the end of 2009 our technology leadership changed. With this change came a proposal to take a more Agile approach to our software development life cycle. While our product development teams were eager to take on this challenge, those of us responding to daily client requests were terrified. How could we possibly adopt principles of the ASDL which, according to Beck et al. (2001) are:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. (Well, we are already trying to do this...)
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. (Given our tight timeframes and commitments, changing requirements are our worst nightmare!)
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. (Hmmm, we already deliver software daily – so this one might be easy.)
- Business people and developers must work together daily throughout the project. (This one, we really needed to work on.)

- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. (We have motivated individuals!)
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. (Lots of work to do here...)
- Working software is the primary measure of progress. (Yes!)
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. (Post-mortems, again?)

While we already released software daily, shortening our project time frames, ending individual Dev and QA client ownership, and getting rid of our project managers seemed like a daunting task.

Over the first six months of 2010, we created smaller, cross-functional, market-focused teams and took a leaner approach to developing software. By no means would you call us an Agile team of the purest of sorts, we have taken a few of the Agile principles and used them to create an environment that works for our professional services, client-focused environment.

The following changes are what have helped us create a leaner approach to software delivery for our customers.

People

- Creating small, combined and cross-functional teams of development and QA to allow for greater flexibility and better coverage depending upon priorities
- Introducing daily stand-ups and weekly sprints that were timed with deployment schedules
- Inviting client services folks to review our work early, even on test!
- Collaborating with technical business analysts and functional business analysts in order to get requirements right the first time

Work

- Introducing the role of Market Champion (MC) to facilitate sprint planning and communication with client-facing staff.
- Prioritizing tasks
- Shortening duration of job assignments
- Reducing the amount of work in the queue
- Holding retrospectives for larger projects and documenting our ideas for doing things better next time

6. How the Change was Implemented

Create small highly functioning teams

The first step towards a leaner approach was to dissolve the larger development, QA and design groups and create smaller (5-7 team members) cross-functional teams to support our three client markets: Employer, Health Plan and Distributor. We sized these teams based on the previous year's data. The Employer and Health Plan markets each have two teams, while the Distributor market has one. This organizational structure in the Technology Client Integrations mirrors the structure in the Client Services organization.

Par for the course, we picked fun team names (Cerberus, Hydra, Kraken, Phoenix and spent the first few months trying to build strong teams and culture. We participated in team building events such as Tree-to-Tree (a ropes course in the trees!), had off-site parties, went to Taco Tuesday at specific team members houses, designed team shirts, and invested in team notebooks – anything to build a strong team identity.

It was interesting to watch how the market teams grew and positioned themselves to prepare for the Open Enrollment busy season. The Employer teams Cerberus and Phoenix grew into strong, distinct teams that did not share work across teams. The Health Plan teams Kraken and Titan eventually decided they would prefer to be known as Thundercats and work together as one larger entity and commit to work weekly as a team.

Manage the Work - Introducing the Market Champion!

Unlike product development, professional services work does not really lend itself to having a product owner because we are doing client work, not product development work. However, in order to function in a somewhat agile manner, we needed someone to prioritize and manage our queue. Before our organizational change in technology, there really was not much prioritization done...all work was approved and all work just magically got done (whether at midnight or six a.m.).

We introduced the role of the Market Champions (MC) in client services. MCs now approve, prioritize and manage our backlog queue of work for us and the team is able to commit to a certain number of points each sprint that we think we can do. We plan each sprint about 10 days in advance. Knowing that we will have expedited client requests, we reserve a number of points each week for expedited and urgent requests, based on historical data.

Our weeks look something like this, based on a weekly sprint with a release day on Wednesday:

Sprint	Monday	Tuesday	Wednesday	Thursday	Friday
Sprint 1	Sprint 1 work	Sprint1 work	Release Sprint 1	Start Sprint 2	Sprint 2 work
Sprint 2	Sprint 2 work	Sprint 2 work	Release Sprint 2	Start Sprint 3	Sprint 3 work
Sprint 3	Sprint 3 work	Sprint 3 work	Release Sprint 3	Start Sprint 4	Sprint 4 work

Managing Planned Work

Thursday is our big planning day. Here is how we make it all happen.

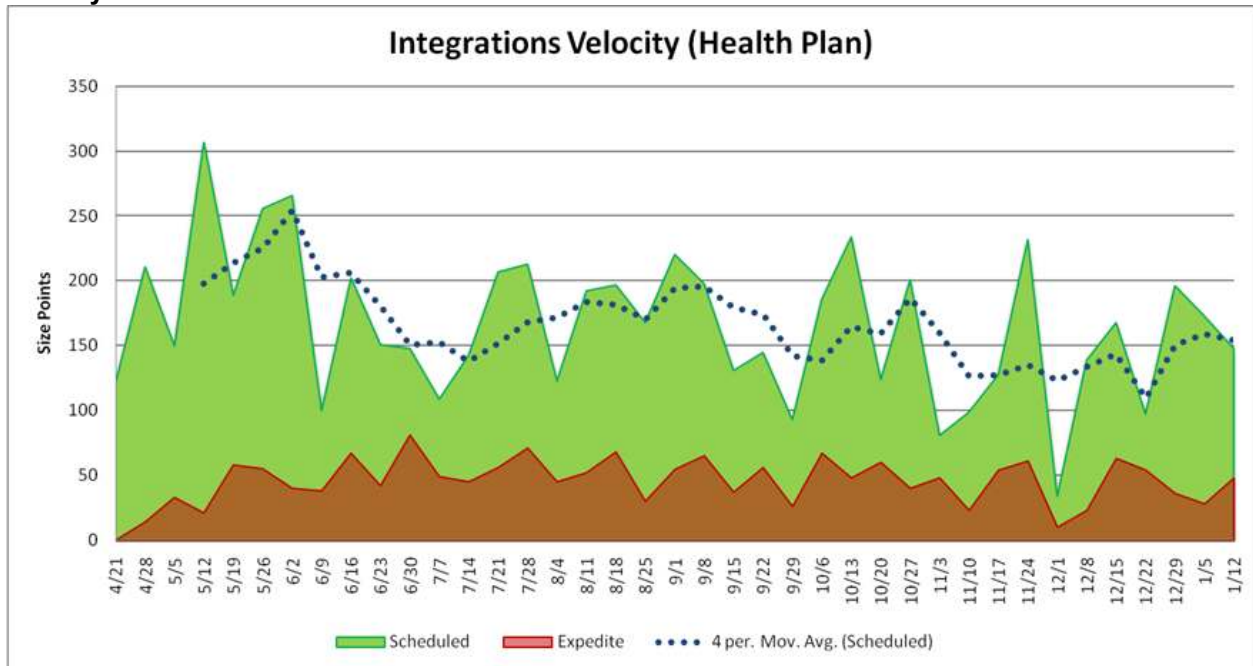
First, the team meets with the MC to review all of the work in the queue and size it as a team. When we started out, we used all of the poker planning card numbers to size our work. After a few weeks of haggling over .5s and 2s, the team opted to use a subset of the numbers for most of our planning:

- 1 = tiny – we can do a few of these in a day
- 3 = small – we can do two of these in a day
- 8 = medium – we can do this in a day
- 20 = large – we can do this in a week
- 100 = huge! – we will have to break this down and work on it over multiple sprints

Occasionally we have the need for a 5 or 13 when and 8 or a 20 isn't quite right, but we mostly stick to the sizing numbers above.

Second, the MC and the team managers determine what our velocity should be for the week. Tracking over several months has shown that we average around 150 points of velocity per sprint, which is about 15 points per person for Thundercats.

Velocity Chart



We've developed an in-house electronic Kanban Board and backlog tool that the MC can use to quickly determine how many items we can take for a sprint once we've determine velocity. The team usually commits to doing 100 points of work per sprint and reserves 50 points for expedited or urgent items that cannot wait until the next sprint to be worked on.

Backlog Tool

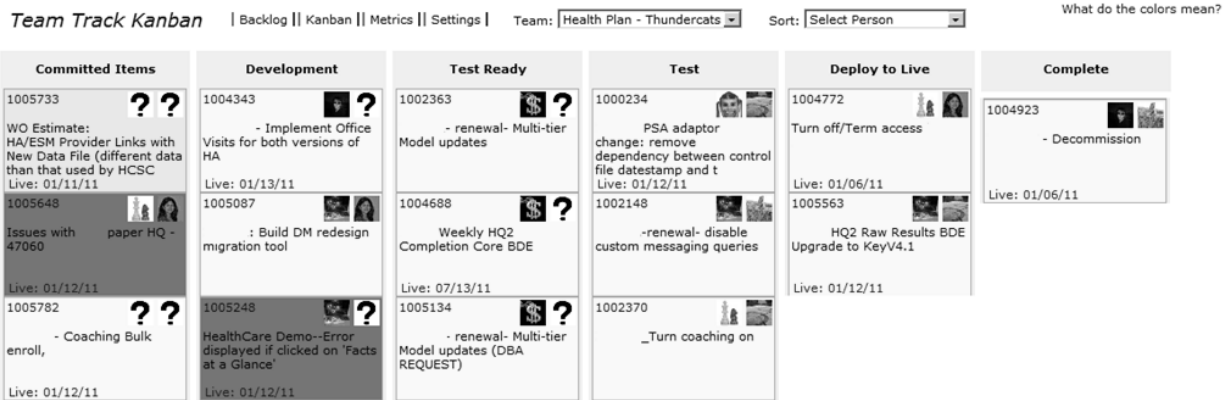
Team Track Kanban Backlog Kanban Metrics Settings Team: What do the colors mean?

See an integrations manager or a market champion to reorder the backlog.

Total Size: 117, Number of Items: 20, Velocity:

	Item ID	Description	Requested Staging	Requested Live	Client Priority	Size	Sprint	Submitted
2	1008833	Rewards Name Change	Requested Staging	7/6/2011	P3 - Standard	3.0	(None)	Submitted 6/8/2011
3	1008880	Health Plan sites Coming Soon page cleanup	Requested Staging		P3 - Standard	1.0	(None)	Submitted 6/10/2011
4	1008859	AuthKey Error: HasDisconnectedAuthenticatedBDEs	Requested Staging		P3 - Standard	8.0	(None)	Submitted 6/9/2011

Kanban Board



Third, the MC and team managers review the list of items we believe we can commit to for the sprint with all of the stake holders in Client Services. This list has already been prioritized by the MC and most often is an accurate representation of what should be done first. However, this weekly meeting is a forum for making trade-offs when we do not have enough velocity to cover the work that needs to be done. It also facilitates conversations about moving work to other integrations development teams if necessary to meet client needs.

Lastly, on Monday, the team manager sends out a list of committed items that will either be worked on or completed for the following sprint that starts on Thursday of that week.

Committed Work

Item Id	Title	Staging Date	Production Date
1007969	Claims DI schedule request		4/28/2011
1007957	alter text on WebMD id card		5/4/2011
1007912	site manager issue		5/4/2011
1007953	Site Move from staging to production		5/4/2011
1007995	Turn on Challenges slide	5/2/2011	5/16/2011
1007946	PSA results page getting cut off on the right when emailing from add list		4/19/2011
1007939	Redesign proposal: Spec #2 (research)		5/4/2011
1007948	Privacy Policy	4/27/2011	5/4/2011
1007949	Privacy Policy	4/27/2011	5/4/2011
1007962	WO Estimate: create custom hydration tracker		4/29/2011
1007907	BDE notifications		5/4/2011
1007996	Add 3 new nodes to support coaching implementation	5/11/2011	5/30/2011

Managing Unplanned Work

Teams have daily stand-ups. We each do a quick review of what was completed the day before, what we are working on today, what we need help with and whether or not we can help someone else. Once we make the rounds with the team, the MC reviews any expedited requests that have come in since the previous day. We review and size them as a team and the MC puts them into our backlog as committed work.

The MC and managers keep a close watch on how many expedited points are being used per sprint. As we approach capacity, MCs and managers will work with the client services teams to either make trade-offs or find help from other teams to get the work done.

7. What has gone well

Improved client satisfaction

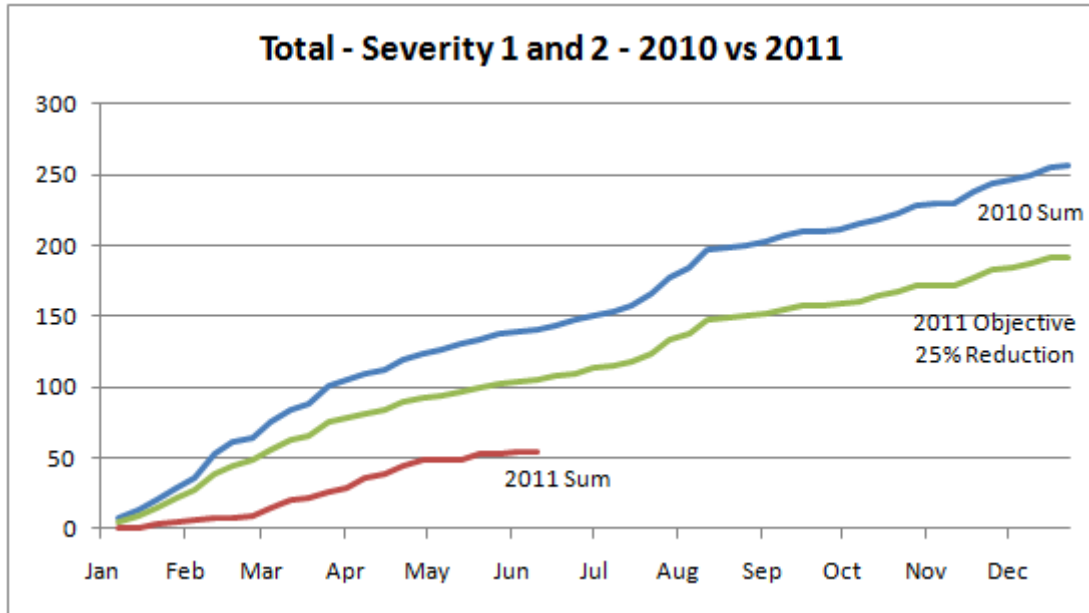
Our Health Plan clients have a whole team to support them instead of two individuals. We all know our clients well and anyone can take on any request. We also have developed close relationships with our client services team since it's a limited number of people. Before we aligned by market, each Dev or QA could potentially have daily interactions with every client advisor in the organization!

Improved Cycle Time

Projects that used to take at least two weeks to complete, are now completed in one sprint (week). Only items that are larger than a 20 should take more than one sprint to accomplish. New client implementations used to require 3 weeks of Dev time, 3 weeks of QA time, and then 3 weeks of staging time. Working together as a team and with the client services group, we've been able to reduce the development and QA time to less than 3 weeks because both Dev and QA can work on any task needed to get the job done. Anyone can test; anyone can code. We even created a new client site, including a web service and SSO in just one week!

Fewer defects on production

As part of the transition to a leaner approach WHS set corporate objectives reduce the number of defects found on production by 25%. Our Severity Committee has determined that we met this goal for 2010, which we believe to be a direct result of our changes.



Increased Job Satisfaction

Team members know that they can go on vacation. They know it's ok to ask for help to complete a task. Rarely does anyone on the team have to stay late because the whole team is focused on completing the work. QA folks are learning development tasks and leading testing efforts across the team, including developers.

8. What we still need to work on

Despite best efforts, our teams were not right-sized for our markets. Our employer market sees the most project work and requires the most resources. Because the employer teams were engaged in large projects, it was difficult for them to manage any maintenance or expedited work that came in. Last year, this maintenance work spilled over into different teams, which was difficult for a few reasons:

- Non-employer market teams didn't have the background knowledge to work on defects or research items that came in.
- Non-employer market teams tended to get the most unappealing, yet necessary tasks, such as work order estimates, to keep things moving along.

9. CONCLUSION

In summary, WHS determined it had to take bold and timely actions in order to increase throughput, productivity, client and employee satisfaction. We did this by reorganizing our Technology Client Integrations department in an Agile manner and staffing according to defined markets instead of client assignments. After our transformation it is a radically different culture which has produced immense benefits in all the areas previously identified.

Over the course of our transition we tried several approaches some of which worked where as some did not. Overall, the Transition was a success. Dev and QA teams are working collaboratively to deliver features in a timely manner. The rate of production defects has gone down and client satisfaction has increased. Employee satisfaction also increased as a result of improved work-life balance.

There are still some things we need to work on like sizing the teams based on workload. We brought members of other teams into the Employer team and split this into two groups. So far this has been going well and as expected. We are continuing the journey because we know we are on the right path. We would like to close with some comments from our Client Advisors and their Director about how this transition proved a success for them....

“Years ago when client services and integrations were at odds with each other because there was no process and a huge gap, it made impossible situation for everyone due to tension and conflict.” But now that there is a process that we follow, we have conversations involving both the sides and have market champion to make sure no one drops the ball makes it easy to understand the expectations. Communication helps! Now we are not afraid of speaking up because we know that our voice is being heard.

“Direct visibility into velocity and sprints lets us better manage client expectations and understand capacity.”

“Always people first - has drastically changed relationships.”

“Workload / client satisfaction side is far more aligned than what it used to be.”

“Overall my clients have noticed that we complete work in a more timely manner and are able to maintain schedules. This has been a much appreciated change in our service level.”

References

Beck, K., et al. (2001). Principles behind the Agile Manifesto. *Manifesto for Agile Software Development*. Retrieved from <http://agilemanifesto.org/principles.html>

Fowler, M. (2005). *The new methodology*. Retrieved from <http://martinfowler.com/articles/newMethodology.html>

Overview (2011). WebMD Health Services. Retrieved from <http://www.webmdhealthservices.com/about-us/>

Glossary

Kanban Board: defined by Miller at <http://geekswithblogs.net> as “essentially a signal, and can be duplicated by using a whiteboard, index cards on a bulletin board, sticky notes, or more elaborate means”.

Market Champions (MC): approve, prioritize and manage our backlog queue of work.

Sprint: a time period in which the team does whatever it can to implement features/requirements that the Team has committed to.

Velocity: a measure of productivity sometimes used in Agile software development. The velocity is calculated by counting the number of units of work completed in a certain interval, determined at the start of the project.