Greetings and salutations from the WebMD Health Services Client Integrations department!

“Here’s the story of a lovely lady…” Oh wait! Wrong story! Sorry about that.

More accurately, this is our story of the transformation to a “Lean, Mean Client Integration Machine”. What that means and what it entailed and why this was necessary for our organization.
I am your host, Mr Roarke and welcome to Fantasy Island.....oh darn! Clearly I watched too many TV reruns as a kid. Sorry about that.

Let me properly introduce the team that collaborated on this paper and topic.

My name is Aaron Akzin and I'm a Quality Assurance Analyst at WebMD Health Services within the Integrations group that is part of the overall Technology department. Sudha Sudunagunta, Supriya Joshi and Aaron Medina all hold the same title, position and role.

Shelley Blouin is a Manager of Software Development and manages the Health Plan team.

Finally Kenny Tran is an Integrations Developer in our group.
WebMD Health Services (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 your organization.

Our “Platform” integrates content, tools, and programs from WebMD along with those that are specific to your population to help you drive consumer engagement, health, productivity, and satisfaction.
WHS offers a host of tools and solutions to support our clients’ wellness and engagement goals. Our clients may want only one or all that we have to offer. Basically our clients fall into one of three markets, health plan, distributor and employer. Our client advisors specialize in one of these areas and report to a market director.

We currently have 4 client integrations teams, which are part of the technology organization and consists of technology professionals including dev, QA and design.

We collaborate with the client services teams to implement online health portals and solutions that meet our client requirements. Once these larger projects are in production, our teams also support most of the ongoing maintenance work, which can be anything from batch data extracts to web services and site redesigns.
Before March of 2009, the integrations teams followed a typical industry standard waterfall software development life cycle and we all fit nicely into the development, QA or design team.

Some of the common challenges we faced in the model were:

- As is common, developers outnumbered quality assurance (QA) by approximately 3 to 1 and were on completely separate teams all far, far away.

- When a request came in for work, a project manager approved it and it became one of many in an endless personal queue for an individual contributor.

- Collaborating on a project meant creating an item in our bug tracking database for every issue found or any questions that needed to be answered. Items would linger in a state of ‘Needs more information’ for months!

- There was a long list of rules of engagement between teams, all defects must be logged, all items must be approved and scheduled, all items must be reviewed, etc
No one team could launch anything on their own. Services were supplied inconsistently as side projects for existing teams. Release Program Management was handled by the project management department. It was a game of hot potato!

The last team to touch it is the team who owns any issues. Different managers, different styles, different priorities. All Teams had multiple sources of requests. Developers could do it but they had no QA resources.

Builds and infrastructure support was happening within all teams separately without consistency or collaboration. The way we took on work made it difficult to respond to changing business needs and priorities. We were not able to quickly change directions because we were in the middle of big projects.
We were very siloed.

Job satisfaction was low...people could only see their piece of the puzzle and wanted to be involved in the broader conversation. Little or no cross training was done. There was a lack of appreciation and understanding from function to function of the challenges each function held.

**There was poor accountability for results** – People could say “I did my part” as blame bounced between silos.

**There were bug spikes with no time to fix them** – Everyone recognized our quality problem. Silos created barriers to anyone being able to fix it.

**There was lots of wasted effort** – Starting work on products that never shipped, filing bug reports for problems that were never addressed

**There was decision paralysis** – Release contents could not be pinned down within a responsible timeframe. Teams were frequently overcommitted and therefore setup to fail.

**There were forecasts of doom** – The planning spreadsheet eventually showed it would take almost an entire release to stabilize the last release, taking all the cycles from new product development.
By the end of 2009, WHS was determined to take a better approach. We decided to learn about this thing called Agile.

Our product development teams were eager to take on this challenge. Who wouldn’t 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

It all sounds great, but for those of us who were responding to client requests every day, this all seemed impossible.

We need project managers! How will we ever get through a project without one?

We need requirements and documentation!

Really, you want us to let the customer see what we are doing before it’s fully tested and ready?

You want us to try new things?

We already released software daily to satisfy the customer early and continuously, but shortening our project time frames, ending individual dev and QA client ownership, and getting rid of our project managers (process really!)

Leveraged Agile principles

- Adapted principles to work in a client-focused Professional Services environment

Encountered some reluctance from those responding to daily client requests

- Shortening project time frames
- Ending individual development and QA client ownership
- Eliminating project managers
Our first order of business was to create smaller, cross-functional, market-focused teams. We went from teams of 15 QA and 30 dev to smaller teams – with around 7 +/- 2 contributors and a manager. In our Health Plan team we currently have 6 Developers and 3 QA + a designer that we share with another team.

We created these smaller teams to support a market, which mirrors the structure on the client services side of the house, each client advisor supports several clients within a specific market. Because of this alignment, we have been able to build effective relationships with a smaller number of client advisors in client services instead of having to build relationships with every single one! We also know our market now and can anticipate questions or issues that our clients might have.

While changes in process were imminent, we really focused on building our team, which meant lots of team building, including Tree-2-Tree, a little rock band, Ground Kontrol and parties offsite. After all, only well functioning teams create well functioning processes and software!

We also physically moved.

We tore down some walls (even though we weren't supposed to!). The managers sit with or very close to the team. We started working together – even with Client Advisors! - on projects at the same time. We reviewed all work coming in. We talked about it as a team. We asked for help and offered it. We
The second order of business was to find ourselves a product owner to prioritize our work. The title was problematic for us since we don’t really work on products – we work on projects for our market – Aha! We needed a market champion! Someone to review all of those requests coming in – make sure they make sense and let us know when to do them! It was no longer a free for all for project managers. Someone was going to be herding the cats.

Dev and QA are no longer expected to prioritize their work and quite frankly, just get it all done. In our previous world, all requests were created equal AND expected to be completed on time and with quality, regardless of the circumstances.

Our market champion now has those tough conversations ahead of time, before we even start work, instead of the QA engineer who just couldn’t get it done. We know how much work we can do now. Once we are at capacity we make tradeoffs when important items come up.

We now have an advocate who plans ahead for us and knows how much we can do in a given sprint because we have continual conversations in our daily stand-up and a planning meeting each week for the team to review our commitments.
In order to get our market champion what she needs in order to facilitate prioritization meetings and trade-offs with client advisors, we had to start tracking how much time we were spending on each task. Before our move to smaller cross-functional teams, we really did not have any visibility into how much work we were doing or could get done in a week. We went through fits and spurts of putting hours on all of our items, but those were always short-lived projects.

So basically it was free for all – the integrations team was expected to do endless amounts of work and we really didn’t have any data to prove otherwise.

After we formed smaller teams we started sizing all items, in fact, you cannot even start working on an item until it has been sized by the team. It took us at least a good six weeks before we were able to determine our average velocity for planning, so we got it wrong a few times before we settled into a routine. Our team is currently at about 15 points per person per sprint. So during our planning meetings we know that we can do about 135 points of work per sprint.

We created the internal tool you see here to track our productivity over a number of sprints by team – managers use this information for planning our sprints as well as determining how many points we use for unplanned requests.
As part of our sprint planning process, the integrations team meets with the Market Champion once a week (and even daily as expedited items come in) to size any new requests that come in. We’ve played planning poker, we’ve assigned a wide range of sizes to items, but ultimately the team has figured out that we really only need about six different sizes 1, 3, 8, 20, 40, 100. We’ve figured out that anything over a size 20 needs to be broken into multiple work items in order to work the project over multiple sprints.

Once we have sized all of the items, the MC and manager meet with the key stakeholders on the client services side to determine how to spend their 135 points. What are the most important things to work on during the next sprint? What projects need to be started? Once priority is set, the MC and integrations manager work together to determine whether or not we truly have 135 points this week since velocity is determined based on historical data for the team and can fluctuate weekly due to time off, holidays, release parties, or other team commitments.

At the end of the planning process the integrations manager sends a ‘commit’ email like the one on this slide, to our key stakeholders on the client services side of the house and by this time there should be no surprises for anyone! We have all agreed – between technology and client services, what we are going to do.
Part of our planning process is to recognize that our business can be quite cyclical.

As you see in this slide, certain markets have different peaks at some points in the year versus others.

For example, the Health Plan market peaks in Q1 while the Employer market doesn’t see a similar peak until Q4. Since we track these annual demands, we leverage our lighter times to create tools to improve efficiency, that will also reduce the potential for error when executing integrations work.
If there are a lot more submitted items each week relative to the number we close, we know we need to take notice. We need to stay on top of that so we don’t find ourselves in a deficit hole that becomes difficult to climb ourselves out of.
As with many companies that offer a diverse range of services, we trend each facet of our work and quantify the work load associated with each area. As with any organization, some areas require much more energy and efforts than others.

<table>
<thead>
<tr>
<th>Component</th>
<th>Grand Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Data Export</td>
<td>4559</td>
</tr>
<tr>
<td>Other</td>
<td>2595</td>
</tr>
<tr>
<td>Claims/Lab Import</td>
<td>1965</td>
</tr>
<tr>
<td>Provider Selection Advisor</td>
<td>1573</td>
</tr>
<tr>
<td>Creative Design</td>
<td>1424</td>
</tr>
<tr>
<td>Rewards</td>
<td>1390</td>
</tr>
<tr>
<td>Coaching</td>
<td>958</td>
</tr>
<tr>
<td>Inbound SSO</td>
<td>802</td>
</tr>
<tr>
<td>HPRS</td>
<td>731</td>
</tr>
<tr>
<td>Custom Reports</td>
<td>711</td>
</tr>
<tr>
<td>Hospital Advisor</td>
<td>659</td>
</tr>
<tr>
<td>Health Quotient</td>
<td>640</td>
</tr>
<tr>
<td>Eligibility File</td>
<td>630</td>
</tr>
<tr>
<td>Custom Content</td>
<td>552</td>
</tr>
<tr>
<td>Outbound SSO</td>
<td>529</td>
</tr>
</tbody>
</table>
About a quarter of our work comes in as unplanned work – we call these expedited requests, which means an item came in that is expected to be released before the next sprint planning session occurs. We review expedited requests in our daily stand-up to size them, determine priority, and either put them into the current sprint or schedule them out.

During the sprint planning process The Market Champion and Manager allocate at least a quarter of our points (or more during our busy season!) for each sprint so that the team can accommodate these requests without putting planned work at risk.

Ultimately our breakdown per week looks something like this:

- Total velocity – 135 points
- Project work – 65 points (planned)
- Maintenance items – 40 points (planned)
- Expedited items – 35 point (unplanned)

We track all expedited requests by adding an ‘e’ to the end of the sprint number so that we can continue to monitor and adjust the point allocation accordingly. As we near our 135, the integrations manager and Market Champion monitor closely and work with the client advisors to make tradeoffs when we hit our maximum or work with the team to accommodate the request. Even if the team
Each sprint week starts on Thursday and ends the following Wednesday when items from this sprint are released to production or to our staging environment for client UAT. For projects lasting longer than a week, we try to complete some set of work during a sprint. There are other times when our work cannot be completed within on sprint, but we allow items to span multiple sprints as needed.
Once we start our sprint, we need to track our work...we implemented our own electronic Kanban to easily track the status/progress of sprint work items. We quickly found that our physical Kanban didn’t really help us since we have some geographically dispersed team members. Taking a photo each day and putting it on the wiki didn’t last long!

Along with tracking the status of the work, our tool also indicates the type of each work item. For example, if it’s a Work Order estimate then the item is displayed in Green, if it’s a bug then it’s displayed in Red etc...We use this tool in our stand ups and talk about the items that might be at risk. For example, if the item is still in the Committed Items list and supposed to go Live in two days then we flag that item and talk about it. Using this tool in our daily stand ups provides us an ability to reshuffle the items if needed. If one Dev or QA is overloaded with the items or if there are no Dev or QA available to work on the committed items then we initiate a talk to act upon this situation.

Keeping track of the sprint items on a daily basis helps us to be on track on deploying the items on time.
And when the big ones come in...we start asking for help early!
So what has all of this change gotten us? Quality is up! We get to see our families…

Before re-org, every client was assigned a developer and a QA. And every Dev and QA had many clients! The client knowledge was not shared across the team or multiple people. Don’t even think about taking vacation! When you did go on vacation, it was likely that all of that client work was still in your queue…which just meant it was best to not go on vacation.

Now the whole team shares the client and market knowledge and so anyone can work on a client efficiently. Since we all know our clients, anyone is an effective peer reviewer. Client services has multiple people to go to and talk about their client needs instead of one DEV & QA. This improved our relationship with the client services team and now it’s more open and collaborative. We communicate with them to understand the requirements on the given client work and also involve them into early testing efforts to review the work being done for their clients. This gives them a visibility into the work they are getting for their clients.
We have also significantly improved our cycle time – which translates to client satisfaction.

In the old days, new client implementations would look like this:

3 weeks of dev time, 3 weeks of QA time (lots of emails back and forth with dev trying to catch up), 3 weeks of staging time (oh my, is this what I asked for?)

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 we all work on the project as a team. We have project leads, but everyone on the team knows what is going on for the project. And everyone on the team contributes to the project in any way they can.

We even created a new client site, including a web service and SSO in just one week!
Again, Quality is up and can be contributed to some of the follow agile-ish approaches we have adopted:

• The whole team works on the same thing – means more eyes, more peer review, more good ideas, more testing
• Smaller, co-located teams communicate in person, not over email or through TeamTrack items.
• Prioritizing work means that the most important things are addressed first, and not at the last minute, when there is not enough time to do the best job possible.
• Planning sprints means that we commit to what we know we can do – we truly commit and do what it takes to get it done.
• Striving for technical excellence by reserving the time to review all of our severity 1, severity 2 and urgent issues, to determine root cause, and to add test cases or process improvements to ensure we don’t make the same mistake twice and
Even though most of what we changed was working, we quickly found that we hadn’t quite gotten our team sizes right to support their markets.

Our employer market sees the most project work and requires the most resources, yet all of our teams were rather equally staffed. Even with 2 employer teams, we still were unable to keep up with the demand, particularly the expedited requests.

In order to get the work done, we tried several different experiments that really didn’t work at all.
We created a two person team to help out the employer market each week. This team would rotate weekly – two from another integrations team. We quickly found that two people are not a team and these two people had no bounded context. They really didn’t know how best to address an issue that came in for a client they new nothing about. This team got the most unappealing, yet necessary tasks, such as work order estimates, to keep things moving along. This lasted all of a week with pushback from everyone.

We had a work party – an after hours event for one evening which included snacks and dinner. While this was a great PR stunt for our client services teams to show we were really trying everything to get all of the work done, this also failed. Again, bounded context was a huge issue. Items from the work party lingered on for months waiting for someone to test them or those that were completed required significant assistance from the employer teams, who could have actually completed the work themselves.

We added resources from our near-shore Argentina team to an employer market team, but adding new resources onsite or even off, that needed to be trained was not a good idea. This overwhelmed and overburdened the employer team. Instead of providing training, the employer team would simply fix the mistakes made by the offsite resources.
This year we have been more actively engaged with our market directors in the client services department to forecast our work. All client requests are entered into our system with the following info:

Project name, project launch date, how likely the project is, whether or not the date is flexible, size (estimated by the manager and sometimes simply a guess since we don’t have any specifications from the client to work from).

This helps us create charts like this one so that we are able to move entire projects around if a team is overburdened. The bounded context issues decrease significantly if a team takes a project from another team because we are on the project from start to finish.

The blue line represents our capacity of 135 for the health plan team. You can see that between the weeks of 8/28 and 9/18, we had capacity to help another team. You can also see from the green forecasted work that we will likely need help from another team starting at the beginning of November.

This exercise has helped us be far more proactive in having conversations about resourcing for projects and trade offs that need to happen.
We have fully integrated our near shore resources onto our teams. They are not an after thought. They join us for stand-up every day and we Skype with them most of the day to get our work done. We work with them on projects. If they make mistakes, they fix them. We even had them come work onsite in our Portland office for two weeks so that we could do some intensive training and get to know them. Even though they are not onsite with us, they are an integral part of our team and have helped shape and build our processes.

We also re-orged our onsite teams to better support each market’s needs.

As part of our overall efforts, we couldn’t resist the opportunity to name our teams and create cool logos. Hey, if work isn’t fun, what’s it for?
As with any process change, your ultimate objective is to make your clients happier with the results. As part of our efforts we surveyed our clients, the Client Services teams and were very happy to get the wonderful feedback you see on this slide which are direct quotes from members of that staff.

What Our Client Services Teams Think About Us:

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

“We have conversations – we now operate with both the client needs and technology needs in mind”

“It’s always people first, which has drastically changed relationships”

“We are always refining but this is great – there is no more tension and conflict”
In summary, the Technology Client Integrations were reorganized in an Agile manner and now staff according to defined markets instead of client assignments.

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.
Thank you for your time and we hope you enjoyed this story of our transformation.