Child Welfare IT Managers' Webinar Series: Child Welfare Information Technology Systems Managers and Staff

From Waterfall to Agile: Understanding the Differences and Managing the Cultural Impacts

December 17, 2014

Presenters: **Patricia Mellen**, ICF International
Kevin Burt, Assistant Director of the Eligibility Services Division, Utah Department of Workforce Services
Wade Owen, Technical Director for the Utah SACWIS System (SAFE)
Tom Kine, Application Development Supervisor for the Social Services Information System; Manager, Minnesota SACWIS

Coordinator: Welcome and thank you for standing by. At this time all participants are in a listen-only mode until the question and answer session at the end of today's conference. If you would like to ask a question at that time, you may press star 1. Be sure to record your first and last name clearly when prompted.

Today's conference is being recorded. If you have any objections you may disconnect at this time. Now I would like to turn the conference over to your host, Joyce Rose. Thank you and you may begin.

Joyce Rose: Thank you and welcome to the Child Welfare Information Technology Systems Managers and Staff Webinar series brought to you on behalf of the

Health and Human Services Administration for Children and Families Children's Bureau and presented by ICF International.

I am Joyce Rose your host and moderator for today's Webinar, and today's Webinar is entitled From Waterfall to Agile: Understanding the Differences and Managing the Cultural Impacts. Due to changes in funding availability and priority, the opportunities for in-person discussions and networking among professionals working on agency child welfare IT systems are limited.

As an alternative, the Division of State Systems within the Children's Bureau continues to provide a series of Webinars supporting information sharing and discussion. The content of the Webinars is structured so as to appeal to state and tribal welfare staff participating in an agency's Child Welfare initiative. Next slide Elizabeth, please.

We will be doing monthly presentations from December through September 2015. Our target audience is state and tribal Child Welfare IT Systems managers and staff, and state and tribal program and policy staff. All Webinars are record and will be published to the link showing on this slide. Next slide please.

Today's Webinar is entitled From Waterfall to Agile: Understanding the Differences and Managing the Cultural Impacts, which I think you will find unique and interesting as our guest presenters, Kevin Burt, Wade Owen, and Tom Kine will share their state's Agile journeys.

The next Webinar topics are in January - Bridging the Partner Gap: Best Practices in Working with the Courts. In February - Bridging the Partner Gap: Best Practices in Working with Education; and then we move to The Project

Management Office: The Effects of Organizational Improvement Frameworks on Large System Development.

In April we will talk about the CMMI Model - Demystifying the Capability Maturity Model Integration. And then the last Webinar that is planned right now is going to be focused on Managing Today's Electronic Workforce. So who is all attending today? Attendees are encouraged to participate in our webinar with questions and comments. All of the participant's lines are muted now, but we will open them for the Q&A session at the end of the presentation.

However, please be aware that you can submit questions at any time using the GoTo Webinar Chat feature, and those also will be addressed during the Q&A session. Should we run out of time we will respond to your questions via email, and/or should you have additional questions, you may submit those to me at the email address listed on the slide, joyce@kassets.com.

Now, who is attending today's Webinar? Well we are very interested in knowing who is attending as it is our intent throughout all the Webinars to make the content applicable and attractive for everyone participating in the agency's Child Welfare Information effort. We ask that you self-select one of the five categories listed. Elizabeth, will you please conduct the poll.

Elizabeth: I have. I've gone ahead and opened up the poll. So if you could please choose from the appropriate category. We do have a relatively large audience today, so we're really interested, as we always are, in knowing who is joining us. So I'll give you just a minute here to go ahead and cast your vote.

If you are participating in a room full of people, I would say pick a category that represents the majority of people in the room. And we're slowing down a

little bit. We have 85% of you. If we could just get the last couple of people to go ahead and tell us who you are. Last 11%, if you could click on the right category - correct category - for you.

Okay, go ahead and close it. And it looks like for today we have 34% state child welfare information systems project managers, 44%, our largest group, is state child welfare information system program policy or technical staff.

We have 6% tribal representatives, 2% that are tribal information system project managers, and 4% that are tribal child welfare program policy or technical staff. And special welcome to the 16% of our participants who are ACF Children's Bureau personnel or ACF contractors.

Joyce Rose: That is an excellent representation, as we all know that Agile Development processes touch across all disciplines within an agency. So let's move to our Webinar format. We will be doing a brief introduction, talk about some objectives. The participant presentations will last about 60 minutes. We'll do a Q&A session and then we'll do a short wrap-up.

So, let's introduce our participants. I am extremely pleased to welcome Kevin Burt who is the Assistant Director of the Eligibility Services Division at the Utah Department of Workforce Services; Mr. Wade Owen who is the Technical Director for the Utah SACWIS System named SAFE, and Tom Kine who is the Application Development Supervisor for the Social Services Information System, and also manager of the Minnesota SACWIS.

To our Webinar objectives, we're going to showcase three state initiatives, the Utah SAFE Modernization Project, the Utah Affordable Care Act Compliance Project, and the Minnesota Northstar Care for Children Project. And in each case what cultural impacts occur by using the Agile Development processes.

We will talk about the business case. We'll do some overview and some goals. We'll look at Agile concepts, outcomes, lessons learned, key insights, and future plans for each of the three projects. So let's start with why you are all here, and let's listen to our state presentations. We're going to start with Wade Owen who will talk about the Utah SAFE Modernization Project. Wade?

Wade Owen: Thank you Joyce. Let's go ahead and advance to the next slide. I thought I'd start with just a brief overview of what our system is like and some of the conditions here in Utah.

So we had a big legacy system written in PowerBuilder and running against a SQL server back-end. And our future direction - well actually current and future direction is to move towards Dot Net Architecture Web based and mobile.

And we decided to use an Agile methodology to switch to that for both our new development and to manage our maintenance and enhancements. We're doing that with a group of mixed contract and state staff. And unlike maybe some of your projects, we have staff augmentation contracts which have been pretty flexible and allow us to work with Agile maybe a little better than maybe a fixed price contract.

We have a very limited modernization budget. We're trying to, like a lot of states, do a lot with a little. So, next slide please. We had several goals, and I kind of distilled these down. We wanted to be more responsive in terms of change delivery. Historically with PowerBuilder and just a regular waterfall kind of methodology, we had delivered three to four releases a year maximum.

We also wanted to - so we wanted to get to where we were doing that much more often, which we've accomplished at this point. We wanted to simplify, as we move forward into the Web world and the mobile world, is to simplify and provide a better user interface for the workers.

We definitely - while we're focusing on Web based development, at this point we are looking forward towards mobility. So we're trying to design with that in mind. And then as we talk about strategic path, we wanted to get away from PowerBuilder because we did not feel it was a real sustainable way long-term to move forward.

So we joined a fairly large camp at the state that's in the Dot Net world. And as we moved into Agile we thought that that would be really more nimble and help us accomplish some of the goals and the responsiveness that we wanted to get to. Agile has seen a lot of usage throughout the software development world. And you might not know this, but it also spills even into manufacturing processes and, you know, physical design processes and not just software processes.

So it's kind of a big movement; getting bigger all the time. And if you don't know a whole lot about Agile development, this isn't a training on Agile per se. There's lots of resources out there on the Internet. If you're just starting, I would refer you to the Agile Manifesto. You can just search that on Google and it would talk about some of the goals and the emphasis of Agile.

The focus for Agile is on working software. So we're developing things that work versus lots of artifacts necessarily. Okay, next slide please. So I wanted to talk a little bit about Sprints. You might be familiar with that term. The

term itself kind of implies this quick movement in a short duration, and that's exactly what it is.

So instead of a long, several month process of getting the release out, we're defining ours as a four week cycle. And so, a defined, repeatable piece of work. You focus on what you can actually get out to the users in that short period. Sometimes Sprints are as short as short as two weeks, depending on the group that's doing it. Sometimes as long as a month, but probably not a lot more than that.

And the Sprint itself, a lot of people when they think of Agile, they're thinking of "oh joy we don't have very much process." And my actual experiences with Sprint is actually very highly defined and very highly planned. And what I mean by that is when we start into a Sprint we have a backlog that's sized and estimated at a task level. And we assign those resources to the team that works on them. And we manage that throughout the month. And so it's highly managed.

We like the idea of a phased approach. And I - you know another term for that iterative, it plays into how you design things. And for example we have just done a case list window. And rather than a big massive case list window, our first iteration is a very minor little list. It was easy to build and create in a month, and we will add layers of functionality to that in coming Sprints.

So you get to deliver things frequently. Sometimes you don't always deliver that to the field, but it's a working piece of software that gets done in a month. And that's - it's really kind of a nice thing to see.

So the users get to see and use the product sooner. And the small size of that unit makes everything more manageable, which I would, when I say

everything, I'm talking about analysis, coding, testing, integration, training; all those activities are not sized down to that small piece. And it really works out pretty well. Next slide.

So in Sprint planning it's considerably different than our classical Waterfall planning. In classical Waterfall planning we've got this big Microsoft Project file and a huge list of tasks and projects, and those are all worked out into detail, and it lays out this comprehensive plan for however long it is, up to a number of years.

With Agile, your long -- I'll even start at the bottom here -- your long-term planning is actually quite fluid. You certainly know at a high level, the kinds of things you want to accomplish. For example, with SAFE modernization, we know the modules that we need to create, and we know roughly, an order that we might want to create those in. But that's very fluid, depending on what happens in coming months.

So we do some pretty specific planning in terms of three to six months. And then when you get down to the Sprint level and you've sized and you've groomed your backlog, meaning that you've prioritized and gotten the things into the list of things that you need to do. When you get to that point and you're ready to lay that into a Sprint and apply resources to it, it's very detailed at that point.

So for your month -- and I've mentioned that before -- we're very detailed in our planning and our execution. But we don't always know the end product from the beginning, I guess would be one point that I'd make. As we talk about phasing, it might be we have a general idea, but that can change and evolve as we go through. So that's a little bit different mindset.

I would kind of point out too that when we made those big Waterfall plans that spanned a number of years, that in many respects those were all fiction in a way. We kind of just created them without always having a lot of knowledge about each task and project. And then we were sometimes held to those plans, so they really weren't based on a lot of facts, necessarily. So, next slide please.

Now just an overview on roles: It's a little different with Agile than maybe some of our traditional roles. And I've got some points here to talk specifically about a product owner. But if you look at a Sprint team, you typically have what they call a Scrum master, and I'll get to the Scrum concept in a minute.

That's roughly analogous to a project lead who's managing, you know, the tasks and the completion and making sure that everything happens in that Sprint in that month it needs to happen.

You have a product owner -- and I'll give some detail there -- and then you have team members. And the point that I want to make here is that when you get in and team on a Sprint, you may or may not define yourself by maybe some of the traditional skills that you have always defined yourself by.

So you might be a DBA by trade and on this team, and you may have to do some other tasks; some test plans or some other kind of things. So it's - you're seen as a resource and your skills can help work on that end product. But there's some self-organization that takes place based on those skills. But you may find yourself doing some different things than you have in the past.

The point I wanted to talk about with the product owner is that this is really a business resources that comes into the team. And it's a really key thing that

we've found out the hard way, if you don't have that product owner in your team and defined and available, that it can actually cause you some real problems in delivering your software.

So the product owner has an overall responsibility for a project's success. They've got to be empowered. And the point that I want to make there is that sometimes in our historical way of doing things, and you may have had different experiences in your state, but what we would see was that we'd have a business analyst and they'd be over a project, and they would consult with program specialists and workers and various stakeholders.

And there would be this month and month cycle that just goes on and one of either email chains or trying to wait until next month's meeting to talk about some more requirements before you can build that business spec and move forward.

That's a real problem for Agile. You really need to bring that in and compress it. So the project owner brings a lot of that into the team. Certainly you still have to consult with stakeholders. But they bring that into the team and they're empowered to make decisions during that month. And that's what helps get you through those design processes and get to where you can code and produce that output.

So they convey vision, they outline work, there's a prioritization process that takes place, and they may very well work with another group that helps with prioritization. But when they - when the product owner comes into the team, they know what those priorities are and can make decisions during the month based on those priorities.

And then availability is a big issue too. So a product owner isn't somebody that's available 30 minutes a week to answer questions. It's a person that's really engaged to the team and is available any time to answer questions and give the direction that's needed. Next slide please.

So - and I'll probably say this again later but Agile, what I found, is a journey not a destination necessarily. So I would say that we are part way down that road here on SAFE. We have a long ways to go and we're trying to make some of those changes as we move forward.

But even with the limited amount -- well limited is not the right word -- we've done a significant amount. But I'll give you some of the things that we still to do in some next slides, but even with what we've done, we have this idea of a daily Scrum meeting. And that might be a new term for a lot of you.

Scrum is a rugby term. And if you've ever watched rugby, they have this - it's like a huddle sort of like in football - but it's this big chaotic meeting that this group gets together and kind of talks about what they're going to do in the game.

That's what happens really on this Agile team. You get together every day. It's a short meeting. You stand - typically stand up to keep the meeting short, and it creates a lot of transparency because every day, every person on that scrum team talks about what they're doing. And they talk about maybe the problems that they might be having, and then you identify resources.

You do it very quickly. You don't solve the problems in the meeting, but it's used to daily keep things in focus. And we have found that, even with our imperfections in Agile, to be absolutely enabling for us. And it's - I talk about

- I've got a bullet here for our PowerBuilder backlog. We had a large number of calls that had kind of languished for a while.

And as we got into this Scrum mode and the monthly release mode, we knocked through those very, very quickly just with those two changes. We also saw improvements in our testing because our communication was taking place every day. And they could say oh, make sure you test this, or have you thought about this.

So even though we have a long way to go, there's not a feeling that we would want to move back because we've seen so many benefits so far. Okay, next slide please. In this slide I want to talk about some of the things that we still need to do. We have a great group of people who are very good at coordinating and communicating in a lot of ways, and cooperating.

But with Agile you're really talking about a team mentality, and that's a culture shift that we're still working on. And what I mean by that is we can end up having a room with 20 people working on 20 individual projects and not feeling responsibility to each other to complete the work for the Sprint. They kind of look at their own work and just get it done.

So what we want is a team with common goals and responsibilities. And when somebody fails it's the whole team failing. So we're implementing some things to try to improve that in the coming months. It's important to have individual commitment to Agile, and I think we've got some places on our team that maybe we don't have true believers in everybody yet. We're trying to counter some of that with some training that's available that we're sending people to.

And what we're - the approach we're doing is we have some people on the team. And I think it's common to a lot of teams, is you have a person who's a de facto leader. He may not be in the org chart as a leader, but he's that specialist that people look to and admire and listen to. And so we're sending some of those people to our Agile training to try to help improve our commitment to Agile as a group.

One of the things that we have struggled with, and I've kind of eluded to this, is that getting the timely business involvement. And initially because we couldn't get the quick turnaround that we wanted with the product owner like I just talked about, we kind of actually would separate that analysis portion so that it wasn't really part of our Sprint.

And so we would wait until that analysis was done and then start a coding Sprint, which is a legitimate way to do it. But the problem was is the analysis was taking so long, and our team was doing so well in terms of delivery that we ended up with a real bottleneck in terms of providing work for them to code to. So that's why we're trying to pull that into a closer model where we're doing some real-time analysis, and getting some specs to them much more quickly so that we can code and deliver more quickly.

And another problem we've had, and I think this happens a lot, especially on states with limited budgets and big operational support issues, is that we tried to mix our operational support and our development work onto the same teams. And the problem with that is that the operational tends to trump the developmental work. Every crisis comes in and suddenly you're not delivering as much development as you'd like.

So we're working to find ways -- creative ways -- to split that out and make sure that those development teams can be heads down. And you've got

resources that you can plan and know that you're going to be able to utilize during your Sprint.

Another reality I think for a lot of groups - and I think this is probably true, even out in industry and not just government, but we have a group of existing people. And we'd love to have the top in everybody in terms of skill and capabilities, but there's a reality in terms of having to bring people up. Or maybe just some core capabilities that people don't have. And so we're having to be creative using the people we have to try to still move forward. Next slide please.

So these are just some of our lessons learned. And I've mentioned this before. And the reasons I mention it more than once is because I think a lot of people, when they think of Agile they're thinking oh, this is a technical process. This is what the Development group is going to do to get the work done.

And in reality, Agile really can't work without committed business resources. So from the top down, from the executive stakeholders down to the subject matter experts, they need to be committed into Agile and integrated into the process. It's not just a technical process. And our historical practice as I said - insulated people from - some of our program people from the Development teams. It was kind of an us and them thing.

So we are trying to gauge and get them involved. We have a business culture and this is probably fairly common as well, where they want to get - they'll identify at a business level - not even at a technical coding level, but at a Child and Family Services level, there's a list of 100 projects that they want to work on. And they assign a whole bunch of things out and just make incremental progress on 100 projects.

And it really works best if the business culture can say no, these are our top three things and we're going to really focus our resources on them for the next short-term so that we can knock these three things out and then move on to other things.

In reality, while it feels like you're getting stuff done working on 50 or 100 projects, you really in total, get more done working on a few specific things and focusing your energies. And then the last bit is that we - the resource issue, and I mentioned that in the last slide. So, go ahead and progress to the next slide.

This is another little insight. I think in a lot of organizations, the physical location of offices and cubicles tend to revolve around political and power hierarchies rather than what really makes sense to be the most efficient. That can be hard to overcome.

But I think it's really important to co-locate your teams. That means mixing your business and technical staff. Some places even have, you know, the technical staff in a different wing of the building from the - or maybe in another building, from the business staff.

So it's good to co-locate. And even at a more granular level if it's your Web team, and they're small teams, that you locate each of those together, even in a bullpen kind of situation, it's a great idea if you can do that. So they're there interacting every day.

The other thing that's a good thing to do is to split your Operations and your Development teams and physically separate them so that the Operations - you don't get over the wall interruptions that impact the Development team -

because that can certainly happen, especially because you've got a lot of your specialists that are on the development teams. Next slide please.

And I've talked a little bit about this too, the design process. So what we're trying to do is make our design processes more close to real-time. So rather than this big serial process of emails and correspondence and monthly meetings, that we're getting the right people into the right room and typing it up as we go and coming out with something that's useful.

One of the principles in Agile is that artifacts, meaning design documents or test plans or any one of a number of artifacts, are tools. They're not end products. They're not our deliverables. They are something that's used to help deliver software.

So, we want to be flexible in the tools we use and how we use them. We certainly need documentation. It doesn't mean that Agile is something that you just sit down and start coding because you're Agile. No, you really still need some tools, but you have some flexibility as a team to decide what's going to help you get to that endpoint. It might be that a note on a napkin is a low end document that can help you move forward, and that's just fine.

Or it might that you need a really detailed test plan. Those can just depend on what kind of work you're doing. You have to have teamwork. I've talked about that a little bit. There's a concept of good enough rather than perfect as well. So you might just have to make some compromises in order to make this small iteration with a useable piece of code.

It might just have to be good enough for that iteration rather than perfect, because that perfect tends to push you out beyond your timeframes. As we design things, one of the things we've had in the past as well is that you try to

approach a design and come to a consensus agreement with a large group of people that something's right or a rule is correct. And that kind of approach just is too slow. So you work in a group and if you can't get consensus you have decision makers that can make that call. Next slide please.

So - and this is a business slide thing as well. I've talked about prioritized projects. One of the things that executive leadership can do is give you that prioritized list if they're not doing that already. And again, focusing on a small number of projects.

And engagement - just engagement in your overall work is important. And sometimes it's easy for your upper executives to be focused towards the legislature or outside projects or regions, but it's really important that they also focus inward on the development projects in your shop. Now I've also included on this slide a little discussion about contract management. And one of the things that I've found -- and I mentioned this at the start -- is that it's hard to get a contract model that fits Agile exactly.

Fixed price is by its nature more closely tied to Waterfall because you're really defining all the things as something you need for a system. And then they're going out and bidding that overall system versus gee, we don't know exactly what we need. We know what we want to do in the next three to six months and we kind of have an idea.

So I guess what I'm looking for and asking for is how - I mean we have this time and materials model that we're using, staff augmentation contracts. But I wonder if there's a better model that could be created around Agile. Next slide please. And this is another challenge as you move to Agile, and that is that the Oversight groups, whether it's your federal partners or your legislature or even senior management, they're used to seeing Waterfall type plans.

They're used to long-term projections and, you know, that's part of your funding request. So you can find yourself actually having to create a static list of projects kind of like you did before for these purposes that you're not really managing to in the Agile world.

And so finding a middle ground between those historical kinds of expectations, helping them understand what you can and can't do, and finding a legitimate way to ask for funding – and it's legitimate to ask for them to want to know what you're going to be doing with the money that they're giving you – so, finding a middle ground where you could all be working together and getting what you need would really be helpful. And so next slide please.

This is just a slide to talk a little bit about our future plans. We've been involved clear through December; late November and through December in refining our process. And we're going to roll that out with our team in January/February, and maybe beyond.

We decided to be Agile in those process improvements, partly because we don't have them all defined yet perfectly, and we want to engage our team in helping us to even refine those further. But partly just because the cultural change is so big, we don't want to inject too much change all at once.

So, I mentioned that we were bringing our training in. And we're not just training technical people. But we're training our program people and our business people with that as well. And that actually took some creative financing because those training funds are actually this time, coming out of our technical budget. And so we had some support from our technical side that they recognized it was important to get everybody involved.

And so we're bringing a trainer in-house which keeps the cost fairly reasonable. And I would assume most states have local resources to get some training. That's not something you necessarily have to send out of state for. We're planning on embedding our program people as our entire team as the product owners to really help us shorten those development cycles. We're splitting our Operational and Development teams as I mentioned.

And one of the things we're looking at; and this is one of the hard things to define, is we've kind of had a backwards support model that involves our database administrators in frontline support more than we would like. And we're looking at a tiered support model that would shift some of that to more like technician people who would handle the first tier in the triage of problems.

I talked about empowering the product owner, and I talked about our design. So we're really hoping that some of these things will help us. And we're excited to move forward with Agile. Again, it's a journey and not a destination. We're hoping to get more teamwork and more buy-in from individuals and then build our skills and increase our – what Agile terms is velocity which is your - the amount of work that you get done in a given period of time.

So that's my presentation - and thank you.

Joyce Rose: Wade, thank you very much. And I am sure, when we get to the Q&A session that there will be several questions for you. So let's move now to Mr. Kevin Burt who's going to talk to us and tell us about the Mandatory Affordable Care Act Compliance Project. Kevin?

Kevin Burt: Good afternoon. If you want to go ahead and go to the next slide, I just wanted to do a quick project overview. I work for the Department of Workforce Services. We determine eligibility for Medicaid, CHIP, SNAP, Childcare, and TANF.

Currently we do have an integrated rules engine as well as an integrated business model, meaning that we're a state based model, not a region or county based model. And the project I'm going to talk about is the mandatory piece of the Affordable Care Act which had a significant impact in the way the rules worked for Medicaid and CHIP. It went to what's called a MAGI methodology.

And so as a state we decided the rule changes were so significant that at that time we would actually rewrite the rules for all the programs determined by the Department of Workforce Services.

So we submitted an advance planning document which is to request some additional federal funding in February 2013. And you can see the date was fairly late because we had to be compliant by October 2013. And a lot of that had to do with the political climate related to the Affordable Care Act. So it took us a while to be able to get approval to request that funding.

And the AH7 exception is just to be able to also include the Human Services programs in the rules rewrite and get them funding as well. You can go to the next slide. So the business rules in general is built in-house, meaning we didn't contract it out. We did use the MAGI methodology, and we rewrote all program rules. And just as a reminder, it wasn't a screen rewrite; it was simply a rules rewrite.

So what we were able to accomplish in that six months was a rewriting of the rules, whereas to the user the screen looked the same. But when they went to apply the evidence to get a generated decision, the rules were completely different.

Incorporating - some of the programs didn't change at all. Of course with the Affordable Care Act, but incorporating the mandatory changes with the Affordable Care Act.

And so I wanted to go into some of the key success as we delivered this. So if you could go to the next slide, the first one was - and Wade mentioned this, is a decision maker has to be assigned full-time to the project.

And for a project this big, to me it made - it had to be an executive level type decision maker. And so what we have, just to understand the makeup here, the Department of Workforce Services as I mentioned, determines eligibility for all the programs in as the state agency, for all programs except for Medicaid and CHIP.

The Department of Health in Utah does not determine eligibility for the program but continues to house the policy and is the single state agency. And we contract, DWS with the Department of Technology Services which is our IT department.

So that's the reason why the coordination with Agile really worked, is you have three main players or main department that needed to coordinate together so well to be able to deliver something so large, so quick. And so to me the key point was that we didn't want a decision token - a token decision maker. It needed to be someone that could actually make a final executive type decision for this type of process.

And it says, watch out for snipers. That's an Agile term. A sniper is just someone who's not involved in the Agile process. So from a distance they're shooting down project ideas and the progress and the - all the different things that are going on in Agile.

Oftentimes you have a lot of people that want to be involved but aren't committed to being involved, and so that's where the snipers come in. And you've just got to be careful of that because they're so used to a different design when you're first doing the Agile process. And in this case actually the late start helped minimize snipers because everyone knew that there was real tight timelines and we had to deliver it real quickly to meet the Affordable Care Act mandatory timelines. So that actually caused in a lot of ways, the group to band together because we knew the ask was so great.

And so I think that that - sometimes we hesitate to jump into things late because we're afraid we're not going to be able to deliver it. But sometimes short timeframes or time boxes really help expedite because the decisions are made faster because they have to be. If you'll go to the next slide.

And this is what Wade talked about, is we really believe the teams needed to be empowered to be able to make the decision. And they should not be going outside of the team for answers. If the team seems to be going outside for answers, we didn't look at a way to create this rapid answer model, we actually just reorganized the team because we felt the team wasn't sufficient to be able to get - be empowered and be able to make the decisions.

And you see the structure of the team. Once we had a scrum master which was primarily IT and we had a product owner which was mostly represented by business or policy. We had the developers, the business analysts. We had a

tester that was the end user. We brought in actual operational staff to be able to do the component testing. And then we had policy specialists that were involved and program specialists with the trainers.

And the key - and this is the Scrum - the team met every morning in a stand up or Scrum, to define what tasks were completed yesterday, what needs to be done, and they drove each process.

And in this case we had to change some of our state statutes; some of our rules. We had to rewrite policy. We had to develop new training because it was so different. We had to change notices.

And in our Scrums or in our Sprints, not only was the IT being built, but all the pieces that needed to be completed were being built, meaning there were tasks to write policies. There were tasks to develop training; tasks to do notices.

And so every piece from start to finish was built and we didn’t move on from developing that program until that was actually completed. If you’ll go to the next slide. And so we had constant organized communication. We had a daily standup with the business owner, the policy owner, the project manager and the Scrum Master. So each department head that was there dedicated, dull-time co-located, met with the Scrum Master every morning on top.

And then the Scrum Master would go and have the Scrum team stand up that same day. We did also have weekly conference calls with executive leadership because there was a lot of people that can’t dedicate the time to be there but need the status, risks, and concerns.

And the key here is decisions made. We didn't use the weekly conference call to discuss what decisions were needed, but more just define what decisions were made so the executive leadership knew what was happening. So it was again, much more communication of what's done; not a permission asking conference call. And then we would also do bimonthly all-hands where we pull in everybody in the project, kind of discuss how things are going and just address anything that was slowing the whole group down, and also deliver some key messages and what's going to happen next.

Communication is always in person, and it was very less formal write-ups. And this is probably of everything, the most difficult thing of Agile. Waterfall, you have everything clearly defined; clearly written, and that's a really hard transition for a lot of people to have it agile, and not have as many things in clear writing and being developed as it goes, and being written as it goes. And again, that's the reason why it's important to have the key decision makers involved.

And if you could go on to the next one and then the next one, just to kind of go over the project design, what we had is we had six Scrum teams. One of the teams was a common rules team meaning they developed the individual program rules that were used across all programs. We didn't want each team for each program writing how residency worked or writing how a relationship works or Social Security or citizenship. So we had one team that did all of the common rules.

And if you could go to the next one, we had a three-program team. And this is just a list of some of the programs that are determined in our eligibility rules engine. And what this did is - so the child Medicaid, ages zero to five, had very specific rules on top of the common rules. So the common rules were

built and these teams would deliver the program specific rules. And you can see there two additional teams there.

And if you could go to the next slide, there was also an ancillary team. And what we found was that there were a lot of people that as they were writing the rules or they were making the changes in the other Scrum teams, they found that, you know there were new notices that needed to be created.

There were maybe some screen changes that were needed. There was electronic verifications in coordinating with the Federal Data Hub that is healthcare.gov. And so what we did is we didn't want to slow down the teams and development of the program rules, and so we kicked a lot of miscellaneous stuff like that to the ancillary team as tasks, and they would complete all of those.

If a notice needed to be changed, the Scrum team that was the program team would write the exact language that needed to be on the notice. And then the ancillary team would only follow and create that notice using that exact same language. And of course we had an infrastructure team with test environments and builds.

And my last slide, if you go to that is just some of the lessons learned in Agile. And IT -- and this is the benefit of having everybody together to be able to challenge each other. What I found is IT does not resolve flawed procedures or operational designs. Oftentimes IT is just asked to work around bad business designs. And so what this did is by having a business owner there, you were able to challenge the process and redefine the process where it made more sense, based on the changes that were being implemented, in this case with the Affordable Care Act.

Number two is IT can be built to account for silo-produced policy, but it is error prone, it is expensive; it's confusing. And so having everybody in the room you can challenge the policy. Waterfall, you don't get a chance to challenge that because it's already written and predefined. You just have to build around existing business requirements.

And then finally, the third leg in the stool is IT shouldn't drive the process. Oftentimes IT will push for the easiest way rather than the right way. And so we challenge the IT as well. And just real quick, the dedication of important resources, especially for a large project, you have to have a real decision maker to be able to make this thing go at the speed that we had to go in this case.

And again, short deadlines can really help expedite it. And again, it wasn't just an IT design. Scrum is not just an IT product, its actually delivering the entire product. By the time the project was done; the policy was drafted, the procedures were drafted, the training was developed, you had experts because you had business there and frontline users that were able to transfer that knowledge as you actually have to implement that you've designed.

And that's it for me.

Joyce Rose: Kevin thank you very much. I find - I found your lessons learned to be quite fascinating. And I suspect that Wade is very grateful of having someone with prior Agile experience so that I'm sure he probably leans on you a little bit for advice given that they are just starting the Agile in their SACWIS system. So, thank you again. And now let's move to Tom Kine from Minnesota to tell us about the Northstar Care for Children project. Tom?

Tom Kine: So in a lot of ways I could just say everything Wade and Kevin said and be done right, because they've really done a nice job of describing Agile well and its use. So I think for me I'll take a little bit different approach to this. Could I have the next slide please?

So Minnesota, on the SACWIS projects, started working to implement Agile in 2011, and we did take a technical approach to this rather than have the business heavily involved. So our focus was on minor enhancements and bug fixes, and really on developers and QA. So there wasn't any business involvement and limited analysis involvement.

And really that was a result of the fact that as Development Manager, I didn't have much control over new development processes, but recognized that Agile could bring benefits to the project, even in a limited way - in a limited use way. Up until we implemented Agile we didn't have a defined release schedule here in Minnesota for the SACWIS System. And there was a lot of skepticism about our ability to set and hit a release schedule.

So in Minnesota we historically had two to three releases every couple of years. And we set out to develop a quarterly release schedule, and there are some reasons that that is a pretty agile schedule for Minnesota because we are county administered. And so there are some architectural reasons that that is plenty often enough.

When we introduced Agile, we spent about a year really socializing the concept with the technical project team. And I mean talking about it and selling it and talking about the changes that it would engender, you know, across the technical staff.

In Minnesota it sounds similar to what Wade was describing in Utah in that our developers, our state staff, there's no significant use of contractors. And so there was a pretty fair amount of training and preparation work required to get started with Agile. If I could have the next slide?

So I'm going to show you three slides that are really busy graphically. But these are - these were the basis of conversations among state staff here, again from the IT side. So these - from IT I'm including the Development team, Analysis and Design, Quality Assurance, Helpdesk, and Infrastructure; not Business.

And the first slide that we'll go to will show the rules and responsibilities within an Agile release cycle. The second slide I'll talk about how new development fit into that which is similar actually to what Wade was describing, how they started. And the third slide, I want to talk about the supporting infrastructure. So, if we could go to the next slide?

We used this to - in front of the entire division to talk about what Agile was. So we talked about our quarterly release cycle, and if you look at the bottom - towards the bottom of this slide, a couple of weeks to start planning a Sprint. Three Sprints within the release cycle, a couple of weeks to develop a pilot candidate, six weeks to pilot and then we're out the door and on to the next Sprint.

And we also, through this then, discuss what the roles and responsibilities of the players would be. So managers, functional analysts, system analysts, developers, QA and so on. Down through this, what the expectations would be for those areas. And we talked about this with the division as a whole and then went to individual unit meetings for more questions and answers about how this would work.

And again, this was all about education and setting the expectations for staff. And you know it is a cultural shift. If you've been working Waterfall, the move to Agile and it takes time to get people prepared to make the changes. Could I have the next slide?

And then this is an illustration of what Wade was describing with the new development. Basically at the top you have one of our quarterly release cycles. At the bottom you have an overlapping quarterly release. And in the middle we have new development following a Waterfall kind of approach.

And you know the problem with new development always is it's unpredictable, as far as when anything will be ready, particularly in a Waterfall type process.

So what we said was that we'll focus our Agile efforts initially on bug fixes and minor enhancements. We'll take that as the product backlog. It was simpler and it was a good base to develop Agile processes and procedures within IT, and relatively low risk. But we continue to run new development in a Waterfall manner, merging in new development only during the first Sprint of one of these release cycles.

And what that did was give us time for full regression testing once we merged the code in. And new development didn't put our schedule at risk. And the emphasis initially was simply on hitting the schedule and building credibility, both within the IT arena and with the business users that we could in fact, establish a schedule and stick to it.

And so hitting the dates was key to really changing attitudes and getting - building credibility and getting buy-in to an Agile process. Could I have the

next slide? What I'm after with this slide was shorter release cycles put a lot of pressure on the IT infrastructure, and I don't know to what extent non-technical people realized that. But for instance with a quarterly release cycle we're supporting three releases at any given time.

We've got the current statewide release that's in the field. We also - you know, we generally have a pilot in process somewhere. And often enough we have the pilot plus one release. And really we need a place to run all of the - all three of those releases in order to handle issues, troubleshoot problems in the current release, work with a pilot, and be moving forward.

And so our infrastructure team had to be prepared for this as well and provide us with the needed development and testing servers and environments in order to support this. And then new development also required a test bed so that it didn't interfere with whatever releases we were working on.

Beyond that, all of this takes a pretty sophisticated source control and branching strategy, if automated builds and deployment are needed, because to - you know, for the technical team to build all this for testing every night and deploy it into these internal serves would be a huge task if it weren't automated.

And automated regression testing is a huge - offers huge benefit in here as well. And so we in Minnesota employ all of those tools in order to help facilitate the Agile process. But it takes time for the IT staff to develop the sophistication to be able to manage all that. Could I have the next slide?

And then we really did start hitting our dates, and that's what all of this is about. Since 2012 we really haven't had any significant delay in a quarterly release. And you know in a way of talking about cultural shift, what that has

done and what Agile has done here is it's shifted the conversation about what should be included in a release from the end of the cycle to the front.

So it used to be, at the end of the release we would engage in a lot of heated conversations about what additional functionality should be put into the release. Because as the release becomes later, there's more pressure to put more and more fixes in it because the users and the business analysts know the next release is unpredictable or it's dated.

And that's putting more fixes in and making the release bigger also makes it later and it's a bad cycle to be in. So once we started hitting these release dates consistently, there was a lot more willingness to defer work; to keep things to a manageable size knowing that we are hitting our dates and we're only three months away from the next release. So could I have the next slide?

And so the quarterly releases really were more successful than anyone expected. And as Wade said, I would second what he said which is in addition to hitting these dates, our backlog of problem reports really started coming down. We started getting results with this - within a few releases.

And we actually got to the point where we were making significant inroads on reducing our backlog of problem reports and small enhancements. But from this phase of our Agile, I would say lessons learned were it is a long process to introduce Agile or it can be to the Technical team, particularly in light of technical sophistication needed to make this successful.

At least in our case, everyone needed time to adjust. But in the end, now that we are on this quarterly cycle and we work in these defined Sprints within that, the project does run a lot more smoothly. Everyone knows what to expect and there's a rhythm to it.

And so that was Minnesota's initial foray into Agile. And then we got this project called Northstar. So if I could have the next slide? So this was - this is a significant change in how Minnesota handles benefits for children in foster care, children in relative custody assistance, and children from foster care who have been adopted.

So what it does is it sets a uniform subsidy for children in all those placement settings in order to improve the permanency of children, and it impacts all our children. But what also makes this complex is, the current programs and the current methods of paying subsidies for children are also being maintained.

So we need to develop these shifts in how we pay for funding and how we evaluate the funding levels or the subsidy levels for children in these different kinds of placement settings while maintaining everything that's already out there.

And I think both the business and IT were somewhat surprised when this legislation was passed. It had been proposed, you know, a number of times to the Legislature and had been turned down. And there wasn't much public discussion about this. So the legislation passed and we were left with how to implement all of this. Could I have the next slide?

And so there are a lot of moving parts. Guardianship assistance replaces relative custody assistance. There's a single uniform assessment process for all these different placement setting called MAPCY which is the Minnesota Assessment for Parenting Children and Youth. There's the Uniform Benefits set.

And it's a fiscal partnership of state, counties, and tribes. And what that means is it's supposed to be cost neutral for the counties. And again we're a county administered state so the counties are responsible for funding quite a bit of child welfare in Minnesota. And prior to this, foster care and relative custody assistance was paid by the counties while adoptive subsidies were paid by the Central Office.

And subsequent to Northstar, foster care will continue to be paid from the counties, but Northstar kinship assistance and Northstar adoption assistance will be paid for centrally. So we're changing how the payments work, and again, while maintaining everything that's legacy. Could I have the next slide?

This had and has an aggressive timeline, so the legislation was passed in 2013 with the changes to take effect in December of '14. And as you can imagine -- excuse me -- not everything was well defined from the business side. It's a complicated piece of legislation.

It's hard for the - it's hard even for the policy people to understand everything that's in here so there's been a lot of interpretation and definition needed. And it's not well understood or it wasn't well understood from the system side either.

So you know basically, we have a big piece - a significant piece of change for the SACWIS System, not well understood by the business or IT. And as you can imagine, it was just a recipe for trouble which is what we quickly got in to. Could I have the next slide?

And so as I said, Agile had limited use in new development to this point. This MAPCY is this Minnesota Assessment for Parenting Children and Youth, was on a particularly aggressive timeline. So the SSIS project was engaged in July

of 2013, and they wanted MAPCY implemented in January of ’14 in order to start piloting it. And we’re actually talking about the potential of bringing in a contractor to do it.

And we peeled that off and used an Agile-like approach in developing a MAPCY -- the Minnesota Assessment for Parenting Children and Youth -- and it was very successful. So it was a complex assessment of the needs of the child and what the caregiver can provide in order to meet those needs, with a complex set of calculations behind it to develop a subsidy around it. But we did deliver it on time and have added additional features over several releases.

So we’ve got a base product out there which was what Agile typically does, and that started enhancing it. So with the remainder of the project we were in real trouble by about April of ’14 trying to follow a Waterfall process. There was just too much undefined. And really the development processes were unclear to the business. So they didn’t have full appreciation of our need for timely decisions. And we had to have timely decisions due to the length and complexity of the - you know, the analysis, the design, and development tasks.

And so as both Kevin and Wade emphasized, one of the benefits of Agile is to get the business to a point where they can make timely decisions and they do make timely decisions. And another piece of this that was causing trouble was we were working basically through large committees sequentially. And so we were - everyone was more or less working on everything.

So in April we - I stopped the project because we simply weren’t going to make - we weren’t going to make any of our deadlines with where we were at. And we engaged with an outside project manager. So he was still within the state with IT, but he hadn’t been engaged on this project.

But we kind of got a stranger to come in and take us offsite as a group -- Business and IT -- and we re-scoped the project. We developed at a high level, work down - a high level work breakdown structure, and developed a new schedule. And you know this is consistent with what Wade said is, you know, you can plan some of the larger parts without going into all the detail.

And further complicating all this for good measure, is by this time IT within the state had been centralized into its own agency and separated from the business. So there were a lot of challenges around the fact that IT is reporting into a different organization which is new for Minnesota.

And I won't go into all the impacts of that except to say it was another complicating factor in all this. Could I have the next slide? And all I'm showing on the next couple of slides is emphasizing the scope of this. We took the Northstar project and broke it into a large number of subprojects. And even after having done this at the work breakdown structure level, we still struggled with resource contention.

We were still having trouble delivering anything because for the most part we were still in large committees; we were still single threaded. So finally what we did is we took a very agile approach to this. I would say agile approach and agile-like because to a purist, I don't know what this would look like. But basically we appointed small teams to work on these subprojects. We had a business analyst, a policy person who was empowered to make decisions, a developer, and a Scrum Master. And the Scrum Master held everyone accountable.

And then we established an overall steering team of supervisors and managers from both IT and the business to keep all this on track. We met weekly, and we empowered these teams to make the decisions and to move things forward.

Could I have the next slide? In the planning again, is huge. This is going out into 2015. The previous one was most of 2014, and then the next slide, are also Northstar related tasks.

So it's just a massive amount of work and it would be absolutely impossible using conventional Waterfall with existing staffing levels. There's no way we could work through all this and at the pace that Waterfall requires. Or if not, at the pace that Waterfall requires, to staff it in such a way that we could get turn around things in this kind of timeline.

So on this slide the SSIS lead is really an IT business analyst. And they are the general organizer of the team. They're responsible for the documentation that needs to be done for the analysis. So you know again, going back to Kevin and Wade, there's still a need for documentation, but it tends to happen during the course of the Agile Scrum as opposed to up front, and our BA's are handling that.

We have a policy lead who's from the business, engaged, empowered to make decisions. Essentially that's our product owner. And we have a developer on the team. And the developers really keep pressure on to get decisions made. They help with design and own all the technical responsibility for the subprojects, but the developers are in the mindset of, you know, I have a deadline that I have to meet, and they really prod this along.

And then the Scrum Master is there to ensure that it is moving, to get rid of barriers. And the Scrum Master participates in the steering team to keep that group informed about where things are at overall. Some of the cultural impacts are here. So initially the business struggled to meet any timelines for decisions. But through this process the business really has a much better

appreciation of how long it does take to develop the complexity that's in there, and why those timelines exist.

And by the same token, development has a better understanding of business and business considerations. We improved working relationship really at all levels of the staff. Initially not everyone even knew everyone else from Business and Technical, and now people are on a first name basis. It's a much different atmosphere on this project than when it started.

And, you know I would say project success is really based on good working relationships, mutual respect, and being willing to engage in give and take when making hard decisions. And Agile encourages all of that because you're meeting regularly. You're working together and you have common goals and purposes. Could I have the next slide?

So we're not really - we're not following a pure Agile process. We have - Scrum - you know we have subproject teams. But we are holding policy to the fire in the same way that we view development, and this is consistent I think with what Kevin said as well which was that Policy has tasks that they have to deliver in here, as well as Development. Could I have the next slide?

And what I'm showing here, and I know it's a little hard to see, but this is a Scrum board. It's in a very public area of the project so it's easy for anyone to see what the statuses of the various subprojects. But you're looking policy related swim lanes or milestones that have to be met in here.

So we do use software to help with Agile. We use a product called OnTime which has a computer based Scrum board and burn down charts and everything you could want to manage an Agile project. But the slow fidelity and very public board is still useful because everyone can understand this and

you can quickly get an idea of how we're doing on the various tasks. Could I have the next slide?

And this is just more conventional Agile. These would be the development tasks, and you might notice that everything is sitting in Policy and not Development. This was early into the release cycle so things hadn't moved to development yet. The next slide.

So to summarize some of this, the outcomes of using Agile for Northstar for Children are that we are delivering significant functionality over several releases, and it is an incremental delivery.

We're not delivering everything all at once. We're delivering base functionality and enhancing it over a number of releases. And part of the reason we're able to do that is that we do have the credibility that we are going to have a quarterly release in three months.

We have exceptional buy-in from the business community. They've really stepped up to the plate with this and are fully participating in this. The business has really an increased appreciation for the complexity of software development and the level of detail that's needed in order to be successful; better understanding of all the timelines that are needed. And overall it's just fostered excellent working relationship. Could I have the next slide?

So what I would say as lessons learned is that Agile is needed by the project, take a look at what Agile is. You know we did that from the technical side. Some of my people did go to formal Agile training and learned quite a bit about it. But in the end we just adapted it to meet our needs.

We’re transparent with the business community. They understand our planning process. They understand a lot more about what IT can and can’t do; what’s reasonable for us to deliver and what’s not. And we have those conversations with the business and let them participate in everything - in the prioritization, in the decision making, and so on and so forth. So there are definitely partners at the table in planning the projects which is a healthy change. And finally, Agile tends to hold everyone accountable, including the business, and that’s important. Next slide.

As far as insights go, the business analysts are really the key players in here. They’re - for us they’re the organizers, they’re the documenters. They do understand business and IT, and they’re a lynchpin in this.

The developers do push the process along. They put demands on business and analysis for decisions. The developers are timeline driven and get pretty excited when they think they’re going to miss a deadline which leads to business has to make timely decisions.

And then finally, plans for the future - we are - we will continue to incorporate the business and Agile projects as we go forward. And we’ll continue to work to improve the process internally. There’s plenty of improvement possible, a lot to clean up, but we are evangelizing the benefits throughout the organization because Agile has really made what looked like an impossible project, possible. So, that’s what I have.

Joyce Rose: Tom thank you so very much. And I suggest that we just listened to three absolutely excellent presentations with a tremendous amount of information having been shared.

So I want to extend a huge thank you to Tom, Wade, and Kevin. And now let's move to our attendee Q&A session. May we open the phone lines and/or Chat to our attendees. And Elizabeth, I'm going to let you manage this portion.

Elizabeth: Sure. If I could ask our operator to remind the audience how to queue up for questions please.

Coordinator: Thanks. For audio participants, if you would like to ask a question, please press star 1. Be sure to record your name clearly when prompted.

Elizabeth: While we're waiting for people to line up on the phone, we did have some questions come in during the presentations.

A couple of different questions about user testing; how did you have to adapt testing strategies to support Agile development? How did testers react to not having an entire finished product to test, but rather to be testing pieces or testing in phases?

Tom Kine: So this is Tom Kine from Minnesota. We tend to handle that by having the testers involved in it throughout the process. So even though the formal documentation may be less than what has been there in the past, they've participated in a process and have a good baseline understanding of what needs to be tested and what the expected results are as a result of that participation.

Wade Owen: This is Wade from State of Utah. One of the things that we stepped up was our expectation for our testing for our developers. But that wasn't - they kind of had a mindset that oh, I'll just do some quick testing and throw it over the wall. And most cycles that that generates is - was just way too much to get

anything done in a Sprint. You could code for a week and then toss it over the wall for three weeks. So we stepped up our expectation of what developers test, and that was part of why we saw the overall delivery quality go up.

Elizabeth: Excellent.

Kevin Burt: This is Kevin again from Utah - or Kevin from Utah. I agree. We had our testers involved in the process. And I agree with Wade that in Agile sometimes the lines blur a little bit in what position you do, because everyone is involved.

And when you finish a Sprint, everyone agrees that it is complete, complete, so everyone is involved in the testing of it. In our case was program start to finish eligibility.

Elizabeth: And then second question was about DBAs and their move from Ops to the Development team. And that was from - that was a question from our first presentation.

Wade Owen: This is Wade. I'm not sure what the question is. It's been somewhat of a challenge. I guess I'll just address what we've done. We underwent a database migration -- and I didn't talk about that today -- but we went from a Sybase database that we managed ourselves operationally, to a SQL server database that was hosted centrally at our Department of Technology Services.

And so that was the first slice of the change in roles for our DBAs because they were no longer worried about or doing the operational kinds of tasks of keeping the database up and backed up and replicated to our warm site. So yes, we are actually still struggling a little bit, and we even talked about that

today, the challenge to split out just our operational in terms of our application, you know, the fixes and bugs in the development teams.

Part of the issue too is we've had a historical culture of our DBAs doing all of the stored procedures, both coding and testing. And we're fairly stored procedure heavy. And we're actually looking that we will shift some of those responsibilities, really stored procedure development, just software development.

And so there's no reason that, at least for the simple ones, but our developers can't do some of that as well. So yes, we have some work to go there. Again that might play into the blurring of roles. The DBA may get involved in some testing or may develop some different skills as well. At the same time we've had some attrition on our team which has helped kind of balance out what - because we've lost those operational tasks.

Elizabeth: Another question that's come in; if you say you don't have as many things written down, how do you make sure you have quality documentation at the end of the effort? If another state asks for your documentation do you have anything that you could share?

Tom Kine: From Minnesota yes, we generally do. That's a key part for us is a business analyst who can document as we go. And it's difficult - I think it's a difficult skill, but it is important that we do wind up with a documented process and we do at the end of the day.

Elizabeth: Okay.

Kevin Burt: In ours, the answer would be they're just different. They're developed in design sessions with the teams and are probably just a little bit harder to read

for some individuals because they're brainstorming of how to be able to deliver what it is that's being required to be delivered in the Sprint.

So there is a lot of documentation, it's just not in the normal format of predefined business requirements. And so I think that that's probably the tricky part and with the verbal communication. We did find that some people it was very difficult to just have verbal communications, so we would draft some things but we would try to keep that outside of the teams and have the product owner and the business owner draft some of the formal communications of what it was we were doing.

But the teams would define what tasks needed to be completed. And so the formal business requirements were developed in the form of tasks that needed to be assigned to people on the team to be able to complete.

So there's a whole history of our system to be able to monitor those tasks. It's just probably a little bit difficult to follow and just share across someone that wasn't involved in it. So, it does limit that possibility from sharing and being readable to another state.

Elizabeth: Do we have any questions on the phone?

Coordinator: We do. Our first question comes from Ann Hunt. Your line is open.

Ann Hunt: Hi. I'm Ann Hunt from the State of Washington and we are just starting to move into an Agile shop, and we are planning using Team Foundation Server as a supporting tool. I was wondering if you guys are using supporting tools at all, or if anybody is using that one specifically?

Wade Owen: This is Wade from the State of Utah. Yes, we are using Team Foundation Server (TFS). It aligns well if you're heading down the Microsoft road.

We have had a few challenges, and I'm probably not the very best expert to talk about that, but we're trying to do some customization to some columns that would allow us to do some reporting and pulling of information so that we can manage things better through TFS. But yes, it's been a great tool for us. We're doing all our version control and, you know, managing our releases out of that as well.

Tom Kine: And this is Tom...

Ann Hunt: Cool, you may hear from us later.

Wade Owen: Okay, yes.

Ann Hunt: Thank you.

Wade Owen: Glad to help.

Tom Kine: Tom from Minnesota here. We use OnTime; it's the name of a product. We also use TFS. I'm not as familiar with TFS. OnTime is highly configurable and tailorable. It's a little more expensive and it doesn't have the source control that TFS offers.

Ann Hunt: Thank you.

Tom Kine: Yes.

Elizabeth: Okay, do we have other questions on the phone?

Coordinator: Currently there are no other audio questions. Again if you'd like to ask a question, please press star 1.

Elizabeth: Okay, we have quite a few that came in over Chat. Why did you choose to develop in-house rather than using COTS software?

Tom Kine: At the time the Minnesota SACWIS started, the project at this point is more than 20 years old. There simply was no COTS software available. For the first release of what became the SACWIS System was in late '99. At this stage, within the Minnesota Department of Human Services, COTS software is being evaluated as a potential replacement for many legacy systems including SACWIS.

Wade Owen: This is Wade. I mean I think historically SACWIS is one of those things that's so highly customized, we just have stayed away from that. But part of that has also just been a funding issue and cost issue for a small state like ours.

But our modernization effort this time around is actually continuing to use existing database structure that we have. So we'll have new modules that we'll refer to table in our regular database. And so it's kind of a phased rollout that way. We'll disable modules in the PowerBuilder site as we create modules in the new site. Now there's some reengineering that's going to take place, but that's part of the reason. And again, just resources.

Elizabeth: Great, thank you. When you're using a combination of both Waterfall and Agile approaches, how do you address budgeting for these two different approaches?

Tom Kine: I would echo what Wade said about the difficulty of contracting. So we do some staff augmentation contracting; not a lot of it. And I've had trouble getting contracts through that describe an Agile process. And I've had contract review and legal staff, in order to satisfy them, have to go back in and really do a more Waterfall like approach to the contract. You know and as far as budgeting, you know as Wade said, we know in general what the overall target is, so I don't know if that's a lot different. Wade?

Wade Owen: Yes, we've had these time and materials contracts in place for a number of years. We've never had to fix the price. And so it kind of was just a natural evolution for us to continue to use those.

And as far as budgeting, it's been a matter of we don't even have building blocks. So we haven't done it that way. We've had some grant money and some carryover money made available to us. And so we've just worked with the amount we have there, if that helps.

Elizabeth: How would you also suggest selling, for want of a better word, Agile to executive staff?

Tom Kine: Well what I did was address a specific problem with it and was able to prove that it worked in the small. So by developing a reliable release schedule we had a lot of benefits. And by the time we needed it for the business we had a track record that showed that we could deliver using Agile. So if I were trying to sell it I would look at specific issues that you have and pick out what looks like some low hanging fruit to try to address with an Agile process, and see if you can prove out that you can do it.

Wade Owen: Yes, this is Wade. I think that that's - I'd echo that same approach. And we certainly pointed out that the ability that we thought we'd have to be able to

deliver more frequently, and we've been able to do that. So, we've had their support.

Elizabeth: Okay.

Joyce Rose: Elizabeth, this is Joyce. And I would just like to add - make a comment here. I think in order to market to executive management, if you appeal to the historical track record of plan driven or Waterfall driven projects, you certainly have some statistics that you can use.

Elizabeth: Okay, thanks Joyce. I want to go back to the phones and just see if we have any questions on the phone.

Coordinator: There are currently no audio questions. Again if you'd like to ask a question, please press star 1.

Elizabeth: Okay, I have two more on line. First up, what type of automated regression testing software did you use?

Tom Kine: We are using Test Complete, and that - for Minnesota that was a good choice because we're still a Windows based application instead of Web based, and that particular product works well with a conventional Windows based application.

Wade Owen: And this is Wade. We've used Rational Robot for our PowerBuilder site for a number of years, and we're transitioning. It did not work as well with dot Net, so we've got a new product called Ranorex and we are just getting that rolling.

Elizabeth: Okay, and then last question I have, it's a nice easy one. What do requirements look like under Agile? I was kidding about it being an easy question.

Tom Kine: Well, you know, requirements are kind of what we're listing at the front of the process. So when I put policy up there on the board with everyone else, the tasks, the policy takes off for this are to deliver the requirements and be able to stand by them, or the end of Sprint or iteration as a part of it.

Wade Owen: And this is Wade. I think I'd add that Agile isn't something - you know there's just a book for it that you just plug in. I think a requirement looks like what you need it to look like for your project.

Elizabeth: Okay.

Kevin Burt: And what it needs to look like for the people that are involved and how well they embrace Agile or not.

Elizabeth: Okay, perfect. I'm showing that we're right at 3:30, so Joyce I'm going to turn it back around to you to wrap things up for us.

Joyce Rose: Okay, and thank you. Excellent questions and excellent answers. So we hope that the information shared with you today was both informative and valuable. And we also understand that Agile Development processes may be new to some of you who have attended.

Thus we have attached a reference guide which you may find useful, and Elizabeth if you'd click ahead a couple of slides please. So we've included some definitions - next slide. We've included some contacts, comparing Waterfall versus Agile, and we've also included some disciplines and the

differences between plan-driven versus Agile. So go back to our conclusion slide please. So again, and as a reminder, please remember to register for the January Webinar once that announcement is released.

Additionally, if you have any questions regarding today's topic, and I'm sure there might be some, so please, or if you would like more information about any of our scheduled Webinars, or would like to volunteer your state as a topic presenter, please do not hesitate to contact me again at the email listed, joyce@kassets.com.

Again this Webinar has been recorded and will be made available on line. When it is complete and posted we will send a message via the SACWIS ListServ.

And lastly, again I want to thank our three presenters. You did a fantastic job. And I'm very pleased that we have had a large number of attendees because I think this topic is obviously of great interest. So Happy Holidays to everyone and thank you for attending. Good bye.

Coordinator: Thank you. And this does conclude today's conference. All parties may disconnect.

END