

**Child Welfare IT Managers' Webinar Series: Back to Basics**  
*"Impact of Development Methodologies Software Frameworks and  
Platform Choices on Release Management"*  
August 29, 2013

Presenters: **Joyce Rose**, ICF International  
**Tom Kine**, SSIS Application and Development Supervisor, Minnesota  
**Tom Hammons**, Vendor Manager, Ohio  
**Fred Crawley**, State Development Manager, Ohio  
**Arum Iroabuchi**, Business Applications Manager, Ohio

Coordinator: Welcome and thank you for standing by. At this time all participants are in a listen-only mode. During the question-and-answer session, please press star 1 to ask your question. Today's conference is being recorded. If you have any objections, you may disconnect at this time. Now I will turn the call over to Ms. Joyce Rose. Ma'am, you may begin.

Joyce Rose: Thank you and to begin with I would like to apologize for everyone. We've had a couple of technical difficulties. As you know we are distributed around the country so we are relying on technology and sometimes it doesn't exactly work the way we want it as you all know.

So anyway welcome to Webinar five of the Back to Basics series brought to you on behalf of the Health and Human Services Administration for Children and Families Children's Bureau and is presented by ICF International.

I am Joyce Rose, your host and moderator for today's Webinar and joining me in a few minutes for a discussion on the impact of development methodologies, soft frameworks and platform choices on release management will be a distinguished panel of state parts. Slide. Changes. Next slide, please.

Changes in funding availability and priority mean that opportunities for in-person discussions and networking among professionals working on state child welfare IT systems will be limited this year and likely in future years. Through this webinar series, a division of state systems within the children's bureau is offering a venue for information sharing and discussion.

We're offering six webinars, one per month between April and September. The webinars are intended not just for child welfare IT systems managers but also all the staff involved in getting and keeping child welfare systems up and running.

Although our series theme is Back to Basics, we invite and encourage participation from both experienced and newer managers and staff recognizing that even the most experienced among us have something new to learn or may need a refresher.

All of the Webinars are recorded and are available online as reference and informational resources for you and your staff. A global notification will be distributed once they are posted and accessible. Next.

What I mentioned previously, today's fifth webinar in the Back to Basics series is the impact of development methodologies, software frameworks and platform choices on release management. Please note the topic for the last Webinar Number six in the Back to Basics series has been selected and will focus upon the end user roles in communications. Next.

Attendees are encouraged to participate in our Webinar with questions and comments. All of our participant lines are muted now but we will open them at the end of the presentation for this session. You also can submit questions

through the go-to-webinar chat feature but we will also save those questions until after the presentation.

Should we run out of time, we will respond to your questions via e-mail and or should you have additional questions, you may submit those to me at the e-mail address listed on the slide, [joyce@kassets.com](mailto:joyce@kassets.com) . Next.

Because of our late start, we are going to skip the attendee call so that we can right to the heart of the Webinar topic. Next. The format of today's webinar is an introduction of our state participants followed by approximately 60 minutes of presentation. Then we will invite all attendees to participate in a Q & A session with our presenters ending with a short summary. Next.

Our guest participants are Tom Kine, Fred Crawley, Tom Hammons, Arum Iroabuchi, and myself as moderator. Tom Kine has been employed in the technology industry since 1975. For many of those years, he actually stayed awake so they will submit a few cogent observations based on his many years of experience and wakefulness.

He has been associated with SSIS Minnesota SACWIS efforts since 1993. He is currently the SSIS application and development supervisor and will be assuming Beth Holmgren's duties as Minnesota's SACWIS manager upon her retirement in September and Beth if you are attending, we wish you only the very best.

Fred Crawley has been the functional state development manager on Ohio SACWIS since its inception in 2004 and lived to tell about it. Prior to that he was instrumental in developing Ohio's SACWIS system to ensure Ohio met AFCARS and NCANDS federal reporting requirements.

In his early days, Fred has fond memories of driving throughout the hills and valleys of Ohio's landscape visiting many child welfare staff and listening to impassioned lectures on the importance of technology in supporting child welfare staff and improving services to vulnerable youth.

Tom Hammons is a Director at CGI who is currently the Ohio SACWIS vendor. Tom is a certified PMP and has an MBA and a bachelor's degree in economics. He has worked on multiple Ohio systems within the health department.

Tom is gifted in bringing diverse people together for a common cause. And lastly, Arum Iroabuchi, he has worked on Ohio SACWIS since the planning phase in 2004. He is the Business Application and Technical Manager of the Office of Families and Children.

Arum has three different Masters degrees and speaks several languages. Originally from Nigeria, he is skilled in project management, software development and testing. We are so very pleased to have these experienced and qualified individuals as our guest participants and myself formerly the project director for the Wisconsin SACWIS, retired in 2004.

Since that time, I have been involved with several ACF Children's Bureau sponsored training events and now I hand the podium to Tom Kine. Tom?

Tom Kine: Thanks Joyce. Could we go to the next slide? Thanks. So, I want to start out by saying I believe in truth in advertising so here's a little bit of that and I actually think this is an interesting and exciting topic which may give you some idea of what you're in for here with me.

In Minnesota we've put a lot of time and effort into developing our release process and schedule and we've had a lot of internal discussions about it. We worked through a lot of different kinds of options so I was pretty happy when Joyce asked me to present on this.

Could I have the next slide, Renee? So, this is the presentation format that Joyce sent out as an outline and I'll be following it closely. I'll give you some background on the State of Minnesota, talk about our development methodology and in our case it's actually methodologies.

The software framework that we use, our system platform, how all of that is informed and impacted our release management process, lessons learned and I'll talk about what we would do if we had it all to do over again.

And I would say as well I'm standing at a fairly high level here but if you want more technical details, you're free to contact me through Joyce if you're interested. Next slide.

So the Minnesota SACWIS is a case management system for county social workers. We have functions for child protection, foster care, adoptions, children's mental health, other child welfare programs, adult male treatment reporting, claiming and other adult services.

There's a great deal of functionality in the Minnesota SACWIS system and throughout its development we've kept the system as open as possible so that it can be adapted for basically any social services program so it is although it's obviously a child welfare system, it is used for adult services as well.

And it's a highly flexible system with a great deal of county and user customization possible. Could I have the next slide, Renee? Minnesota is state

supervised and county administered. I believe Ohio is as well. That brings its own challenges as I'm sure you are aware and it has had an impact on our platform and our process throughout the history of the project.

The Minnesota SACWIS which is now the social service information system or SSIS is used by 80 individual counties, seven additional counties which who are organized into two regions and two Indian tribes and the regional organization is setup so that they can share resources and make better use of their staff.

And regionalization is a trend in Minnesota. We're continuing to add counties to regions and we're seeing counties exploring regionalization so that's a particular challenge to us the way we're architected in Minnesota.

And the system serves over 6,000 agency social workers, case aides and DHS staff. We use SSIS to track cases involving about 270,000 individuals annually. Could I have the next slide? SSIS started planning in 1995. At that time there was no statewide social service system.

It took years of negotiation with the counties to get SSIS established and the counties wanted to keep their data separate from one another. There wasn't a lot of trust between the counties or between the counties and the state.

And actually we started some work in 1993 before the SACWIS legislation was even passed to start the planning for a standardized statewide system. In originally working with the counties, there was a lot of mistrust with the state.

It's a much different atmosphere now and we have a good partnership with the counties but because of the atmosphere as we started this system, it led to

some fundamental architectural decisions that I'll talk about a little later in the presentation. Next slide.

In Minnesota we use basically three methodologies. We use Waterfall, RAD and Agile. We're pretty pragmatic here and we can only - in the development area - we can only control some aspects of our world I would say so we're open to use whatever works in any given set of circumstances.

And I think these methodologies the way we use them and I'll talk about them in a little more detail are appropriate for the tasks that we use them for so I think they have a place in how we do software development at SSIS. Next slide.

So with Waterfall we use that for fiscal-related development and as you probably know Waterfall is good with well-defined and understood requirements. The fiscal-related development involves payments to vendors for services provided and or reimbursements to counties so Waterfall gives us a high degree of control for these kinds of financial systems.

We do a lot of up-front work design-wise to make sure that we have everything correct and it gives us the ability to create detailed testing plans based on the design documentation and that can be done concurrent with development. Once the design is complete, development and testing can go on concurrently.

The con, of course, to Waterfall is that it's rigid and harder to adjust for changes in requirements but again for financial type applications. I think it's appropriate and even in a Waterfall, developers get involved early in order to have a good understanding of the whole picture and to help with the technical design issues.

And at the back end of Waterfall development effort, we do a formal code review that's a little different than a standard code review and then we compare the code to the specifications and it gives us an opportunity to ensure that the code is actually meeting the spec and vice versa, the specification event according to the code.

And I would say Waterfall works well for what we use it for. Next slide, please. We use RAD or rapid application development for social services related development. These would be the UI portion of our system and I think the approach for RAD is good for development that focuses on user interaction.

You can do fast iterations focused on incremental improvements. Testing can be a challenge with RAD. Similarly to a Waterfall, development gets involved early in these kinds of efforts or we try to.

Generally, we focus first on getting a database structure in place and getting it fairly close to what we think it's going to look like before we start work on the UI so we do try to get a decent database foundation in place early in and do some planning up-front but then we're going to an iterative process to develop supply over it.

We've been successful with this methodology as well and it can be a challenge for the designers to keep up with the documentation because it moves pretty quickly and again because of that the testing can be a challenge. Next slide.

And finally, we use Agile for managing bug fixes and minor enhancements for bugs. The pros to Agile are that it's flexible particularly as compared to

Waterfall. We get quick turnaround to the iterations. We do QA throughout the process so we wind up with fewer defects.

With this kind of methodology, it can be hard to get or maintain documentation and for us it works really well for managing a large number of small tasks so we use this basically for our product backlog and we use Agile as the core for how we drive our entire release process so this is really the driving methodology for how we do our releases. Next slide.

This would be a gross oversimplification which I'm known for but SSIS is your classic end tier application. On a desktop, we have a 132 client that's written in Delphi. We are introducing some C-sharp through a component that helps facilitate 132 interaction with managed or dot-net code.

Then of course we have a middle tier that we connect to. It has some business rules and it provides a data connectivity. We can also hang a server off of this and get some interfaces and the back end is an Oracle database. Next slide, please.

The way that deploys though because of our county administration because of our history is we actually deploy a copy of SSIS to each of the sponsoring agencies so the 80 counties each have a server with SSIS on it. The two regions, each have a server with SSIS and the two Minnesota tribes have a server or servers with SSIS on them.

So we look a lot like a piece of packaged software or a COTS product that's installed in each of the counties. There are a number of centralized connections that we use for instance to help un-duplicate clients statewide and to consolidate data for state and federal reporting.

But the system actually can run without connectivity so it can run as a standalone system and can pick-up the connections when needed if there are any outages and that kind of gives us the unique advantage I think on some of the disaster recovery planning type functions. Next slide, please.

This is an eye chart I know and I'm not going to try to go into it in any detail but what this shows is on the right-hand side our county connections down through the middle are all our state servers that we use for un-duplicating clients and so on in state reporting and on the left-hand side are all our interfaces.

So it's a complex architecture. And, keep in mind that we have about 6000 client machines out there in the field all of which have a SSIS application installed on them. Next slide, please.

So as you can imagine, distributing software to 6000-plus desktops is a really big deal. Historically, until we introduced Agile, we just bought three releases every two years. It was complicated and of those three releases, one or maybe two of them were major releases and then there was a bug fix or several bug fixes in-between.

We use a custom-written application launcher. When a user takes the option on their desktop to launch SSIS, the launcher runs. It checks the server for updates. If an update is found, the launcher patches or replaces the EXE and launches the SSIS worker application.

The patch technology minimizes the impact of pulling down the entire EXE over the county networks, that introducing patching was a big improvement for us. It really helped with software distributions. Next slide, please.

So as I said, Agile project management has had a huge impact on release management in general. When we moved into Agile, we converted to a quarterly release cycle for bug fixes and minor enhancements and we divorced new developments from the release scheduling process.

So in the past the reason releases took so long was it was - the release schedules - were dependent on when new development was ready which is relatively unpredictable. With the Agile and the focus on quarterly releases, we release every quarter and we introduce the new development at the appropriate place in the release cycle.

So if new development isn't ready when a quarterly release is ready to go, we go with the quarterly release and delay the new development and again in the past. With the more-unpredictable release cycle or the release cycles that were dependent on new development, the later they were, the more behind we got.

But fixes continued to be pushed into the release as it was delayed for new development and the more bug fixes we pushed in, the later the release got so it was really a vicious cycle and when you get to the tail-end of these releases, there was more and more work being pushed into them and at the back end we'd have to have negotiation about what would be in the final release.

It was a really difficult process to control and extremely unpredictable. Next slide, please. So with Agile project management to give you some background of how we approached and introduced it into SSIS.

And in 2009, Tom Jones and I who's the GUI development supervisor attended an IT symposium here in Minnesota and they had an Agile development track so Tom and I followed that track through the symposium

and started learning about Agile. Tom joined a self-study group for scrum master.

We started educating project staff on what Agile was and we developed a plan to implement Agile for bug reports and for us we were kind of cherry-picking here when we chose to do it for bug reports and minor enhancement requests because as in the development group, we had more control over the scheduling of those than we do over new development.

So we're able to introduce this here because this was a part of the project that we could more easily control. It took quite awhile to start using Agile, a lot of planning so we attended a seminar in 2009 and in 2011 we actually started using Agile.

And now we're to the point where we've had three or four releases, maybe five in a row delivered on time so we've been successful with it. It's been quite a process though. Next slide, please.

Again I know this is an eye chart but this is the kind of material that we prepared and presented to the project staff when we were introducing Agile. We spent a lot of time explaining what we were doing, why the mechanics of how it would work so what you're looking at here is a sample iteration and what tasks need to be completed throughout it by who needs to do it.

So down the left-hand side we have tasks for managers, functional analysts, systems analysts, developers, QA and so on showing what these groups of people would be doing and when.

Down towards the bottom middle is the actual release cycle so you can see Sprint 1, Sprint 2, Sprint 3 in there so what we do is several weeks before

we're going to branch to a new quarterly release, we spend time planning up-front so we turn this around where in the past we've planned, you know, we've negotiated the work at the end of the release.

Now we're negotiating up-front for how much work we can do. We only plan the first couple of sprints. Neither Tom nor I felt that we knew enough or were smart enough to be able to anticipate what would happen by the time we got to the third sprint.

So we do two four-week sprints and keep the third one open for any remediation that we have to do as a result of the first couple of sprints and it gives us a place that we can push more work in as we go so assuming things are going okay, we fill-up Sprint 3 pretty quickly as well and that's a five-week sprint.

After the three sprints, then we move into pilot candidate builds. We're focused at this point on getting a good build for the county. If we're going to pilot our application typically in Minnesota that would be two or three counties and then we move into a four-week pilot before the statewide release. Next slide, please.

And again I apologize for the eye chart here but this is showing how new development fits into that cycle so at the top and bottom and the greenish color, you can see our quarter release with the three sprints.

Through the middle we're trying to show how new development works and again we use this internal communication with the staff but basically new development follows its own branch and we have some technology that helps facilitate that.

We use our team for source control to support the flowing branch which really helps with this process but new development follows its own path, its own QA, etcetera. It's getting code from the mainline throughout to help make it easy to merge code at the end and when it's finally ready, we schedule it up for the next quarterly release that it fits into Sprint 1.

And we want it in the first sprint so that we have time to do a full QA on it or regression tests throughout the development life cycle. It takes a lot of QA databases to support this.

We have 12 to 18 QA databases I would say and because of our scheduling. Could I have the next slide please? We'll have three releases that we're working on concurrently which is what this slide shows. We have the current statewide release that we may be taking bug reports on so we'll maintain that in QA.

We have whatever release is in pilot and then we'll already by the time we're going into pilot, we've already branched into our next release so we have to juggle quite a bit at the end of a release cycle.

It's complicated and it's taken us some time to ramp-up so we've had to move into this fairly slowly. It works well at this point but it has been a process. Next slide, please. This is planned versus actual release dates and again we're doing well as far as hitting our release dates and I say that with some surprise and I think everyone's been surprised by that.

There wasn't initially an expectation that we would actually be able to hit release dates but we've stuck to it and we've stuck to the policy of delaying functionality if needed so the data's going to take precedence over introducing functionality and that's been a shock to staff including developers.

We're trying to push changes in late in the development cycle and we're just not having it. We are sticking to our guns as far as well, scheduling when and when we're actually going to release.

We're still in educating our customers, pretty nearly a state policy people on the scheduling applications this has because we do need lead times for new development for (anything about bugs). Next slide, please.

If we keep metrics, we measure this every release so we know story points completed by release and we feed this back into our work planning process if I could have the next slide, please, and what we do is we take these story points and we estimate how much of our time is spent on working on a problem reports in these quarterly releases versus new development.

And we allocate the available time to the product owners and again we're pragmatic here so we're considering our product owners to be the development supervisors and a more classic kind of Agile, you know, your product owner would actually be a business customer but because of how we're doing this, we've changed around a little bit.

So we take the story points that we know we can do on average. We allocate them out to three different areas of the application and we let the supervisors in those areas determine what they want to schedule into the release and there's give and take among those people to negotiate how much work is done per release in the different areas of the application.

And again this has helped us plan up-front for our releases. It's helped us get out ahead of things quite a bit more so that when we're finally towards the end of a release cycle and in that last sprint, there really aren't any surprises. We

do weekly updates with everyone to keep track of where - to inform everyone - where we're at in the sprints, how well we're doing.

Tom Jones does an excellent job of that. He runs - we only do - two stand-ups a week but he manages those and keeps the developers on track and we've really developed a pretty good process and a pretty good rhythm in here. Next slide, please. Could I have the next slide, Renee?

Okay, as far as lessons learned go, I would say establishing fixed release dates and hitting them really builds credibility and it takes a lot of pressure off. If you know - if your customers know - that you're going to release every three months, it's not as difficult to delay the things if you need more time to work on them than if you have unpredictable or ad hoc release dates which is what we had.

The fixed release dates really facilitate work planning. Everyone knows when we're going to release and it's helped a lot and you wouldn't necessarily think about this but it's helped and as we evaluate bug reports, we can schedule them into specific releases with a high degree of confidence when they're actually going to go out to the counties.

Whereas in the past when we tried to schedule bug fixes or minor enhancements, we'd push them into arbitrary releases in the future without really knowing what date they were going to be released so now we've named our releases by quarter so in 2014 we'll have a release 14-1, 14-2, 14-3, 14-4. In '15 we'll have 15-1 and so on.

They're very predictable. They're easy to understand and we can drop work into the appropriate buckets right up front. As I said with credibility, the users

are willing to negotiate scheduling the project team really functions a lot more efficiently when you get a rhythm like this established.

There's a distinct ebb and flow of those things. People know what to expect and I think it's really helped clean-up a lot of processes and our software quality really has gone up as well with the predictable release dates.

It's taken pressure off of everyone. I think there's been a noticeable improvement in quality and we are reducing our backload and I have metrics to support some of the quality improvements as well. Next slide, please.

If we had it to do all over again, I think we would establish a fixed release cycle as soon as possible and we would focus on hitting the published release dates every time. Don't push last-minute fixes into the release. That really has whenever we've done that we've regretted it. It, you know, it can make as many problems as it fixes.

Negotiate scope at the front of each scheduled release so at the front of that planning process; be clear about your work capacity, what you can do. Be realistic about it and people understand that if more work gets pushed into any given quarterly release, something has to be moved out so it's a prioritization process.

There are not unlimited resources from the development side, the QA side, the design side and we've worked that out internally within the project staff and now we're communicating that out to our customers more and giving them a view of our development process.

And then finally make it a priority to develop metrics measuring the quantity and quality of the work and this helps sell these kinds of ideas. I think if you

have the metrics and can show that you are improving quality and are whittling-down your backlog of work to be done, it really again helps with the credibility.

It raises the comfort level of everyone with an interest in the product and our counties are big supporters of this as well I would say because for them now they can plan on when they're going to get a release so this release cycle not only impacted the project staff and the policy people at the state but the county workers as well.

We planned this out carefully to not interfere with the county financial cycles. We don't do it at a quarter end, you know, at the year end and so on and we publish our release dates a year in advance and we have some idea of what's going into each release so we communicate as much as possible, as often as possible and I think that's been much appreciated.

So that is the view from Minnesota I guess on release cycles and how our methodology and architecture has informed it, the view from 80,000 feet I would say and Joyce I'm going to hand it back to you.

Joyce Rose: Thank you, Tom and as expected a very informative presentation and I suspect our attendees will have some questions for you during the Q&A portion of the Webinar so now let's move on to the Ohio segment and leading off will be Arum Iroabuchi. Go ahead.

Arum Iroabuchi: Thank you so much, Joyce. First off, I would like to commend you for attempting to say my full name and you didn't do badly actually you did very well. Most people actually call me by my last name. It is much easier to pronounce.

First of all, I would like to very quickly thank SEF for giving Ohio I know the opportunity to participate in these Webinar series and it kind of sticks to the importance of the work that we all do.

In line with what Tom of Minnesota, we got to try the best we can, my colleagues and I, Tom, Fred to stick with the schedule as prescribed, so looking at the timeline, I'll try to give you essentially a background of how you operate when it comes to child welfare.

And then we'll actually step through development methodology and we go through the software's framework and work about a system platform as well and then we'll talk about the impacts of release management and then wrap it up with lessons learned and a response to the question as to what we would do differently if we had it over again.

Could you advance the slide, please? Ohio as most people know is like Minnesota. It kind of is a supervised state. We have 88 counties in public child welfare agencies that we work with but in addition to that, we also work with some travel agencies.

I think when we realized on last count we had about 94 of them, a little less than 100 private agencies that we work with as well. The Ohio software system is a Web-based system. It has in excess of over six million lines of code I was told and so it's a very large advanced system.

We've been actually going at it now since 2004 and we rolled-out our last public agency in 2008. Currently we're working an effort to be thoroughly compliant SACWIS wide. We had our federal partners here earlier this year and as a consequence of that visit, we have a long laundry list of findings we

are actively working to complete before they return early next year for another round of reviews.

The way the Ohio project is setup, we have a mix of staff members that we work with so we have an integrated project team that is made up of state development staff members as well as state policy members and also state software business staff members and in addition to that, we obviously are very fortunate to work with some vendors as well. I represent the CGI organization.

We have over 7000 users of our system and since we rolled-out in last county which was in 2008 like I said, we have a typical systems like that being able to implement a number of important improvements or enhancements of the system that was supposed to help our users get through their business processes and I would like to mention a couple of them.

We were able to successfully rework the financial module of the system and was able to from a deployment perspective upgrade our Cognos system from 8.1 to 10.1.

Also, some tests, the core process and module, we've actually realized that two of our mandated interfaces we have interface with our system up and running at this point and we also have an interface with America system up and running as well.

We also went through a very critical upgrade of our infrastructure of the environment that we use to do development work where we went from Oracle 9Is to Oracle 11G over the last year as well, last two years I think.

Looking forward into the next year maybe two years, you have a list of important project initiatives that we are actively working to accomplish and

one of those that are private initiatives is trying to complete Phase 2 of our private agency roll-out.

Now like I did reference we have 94 private agencies that are currently using our system but they're using our system very limitedly to perform functions that relate to training, training stipends and allowing these reimbursements.

There's two of the private agency initiatives again and that will private agencies to use our system to maintain or manage for better records in the system, complete home studies and the like.

We'll also have an initiative that will load this back in a form that would bring on the four systems that we have in Ohio to use SACWIS to manage their kids' rules. Currently, we have about 44 county four E-Courts because I will be benefiting from that effort.

Additionally, we're working on a mobile application that we call field assistance that was going out in Nevel county agency workers in the field to complete some basic functions while on home visits with clients since like activity logs and access some intake as well from the field.

That application is actually being tested now so if all goes well, our hope is to have that out very soon so that workers are using the app well out in the field. Additionally, we're also working on an Optimal J refactoring that would enable us to upgrade because Optimal J currently is Indian supported.

So obviously that brings about quite a bit of exposure and some risk so if all goes well, that was our process initiative that we looking to accomplish over the next fiscal year and I'm going to hand the discussion over to Tom so to provide more detailed information about Optimal J as well. Thank you.

Tom Hammons: Great. You can advance the slide, please. Now this is very similar to Tom Kine's slide. What we've tried to illustrate here is our development schedule and we run each one of these things in parallel.

So at the same time that we're doing our JAD or our joint application development in developing our, you know, screens whether it's going to be the enhancement, we have folks that are developing the code or they're working in the construction phase, executing test scripts and also doing business analysis and architectural review.

Now each one of those phases feeds into the next and that allows us to keep a four-week release schedule so every four weeks we're delivering a mixture of large, medium and small enhancements to the system. You can advance it to the next slide and what this chart tries to show you then is how those four weeks fit together.

You can see on a day-by-day basis which days that someone's conducting a JAD in development, doing the testing and at the end of this presentation what we've done is we've provided an illustration of what goes into each one of these steps, the inputs, the outputs, what's needed and where the handoffs are so that you might be able to use this information on your systems. I'm just going to check and see if (Fred Crawley)'s with us.

Fred Crawley: Yes, I'm here.

Tom Hammons: Okay, and Fred feel free to jump in here. You're a very integral part into this.

Fred Crawley: No problem. Next slide. All right, that's you Tom.

Tom Hammons: Okay, to talk to our framework platform and what we try to illustrate here rather than show you all the different servers is to show how the different tiers work together, the presentation tier, the business tier and the database tier.

As Iroabuchi had described, we use a component called Optimal J to generate some of our code and we're going to be refactoring that into using some of the newer and latest Web technology.

That's going to enable the Ohio SACWIS system to expand on things such as the mobile field assistant and allow the application to use more of the new technologies and meet the needs of our child welfare workers and (Fred), I'm going to pass it off to you with the case management on the next slide.

Fred Crawley: So next slide, please. One thing we wanted to say, we think a great deal of our success in our process and our productivity is that we've been in an integrated project team since approximately 1995 so a lot of the things that we're doing and we have been doing for a very long time that are now being known as Agile methods, we've been doing a lot of this stuff for a long time.

One of the big things in Agile is that you have to involve your customers very tightly in your development work and we actually our business analysts and our folks in policy on the business side, we don't even really consider them to be customers as we consider them to be our partners.

So our team is broken down - our integrated project team - is broken down into five distinct mini-integrated project teams that are consistent with the major modules of the SACWIS system and those groups act as kind of autonomous little automatons that basically look at the work that we have to do and do a lot of the planning and development.

For example, all new work that's coming in, enhance it requests, defects, all of that stuff that's coming in on a weekly basis is looked at and put into the team queue so each team has its own queue and that queue is organized by the business analysts.

That is to say they put new work onto the queue and they prioritize work in the queue. The developers involved would then work on looking at the impacts of this new piece of work, determining when it can be done and which release so they kind of together identify in which release each new piece of work is going to occur.

So those queues as it were respond to or correspond to the scrums backlog so that's how we look at it so we've been doing a lot of things that are consistent with the Agile techniques well before Agile became prominent so we do a lot of up-front planning.

We introduce new functionality to our county workers on a regular basis and that's part of our four-week time box schedule. We have specific times in the schedule when we bring county workers to let them look at the results of our JAD before we actually solidify things and start to really kind of hard-code them so the counties are built into that process.

At the back end we actually have counties come in to help us with testing so that preliminary review of JAD or designs and that back-end testing involving the counties, we do that in every release. Once again the release schedule mouse comes with tech on a regular basis.

We have our BAs and our developers moving every day reviewing work, even when development is going on, the business analysts are logging-in to developer machines or visiting the developers at their workstation and making

sure that the work is right and not after it's done but while it's being done.

Next slide.

Well, once again we think collaboration is key and we can't really stress this above anything else. I don't know exactly how other counties are working but to put your business analysts, your policy folks, and your developers in a collocated situation is key to delivering quality software on a regular basis. We have a saying as we want to deliver often and frequently.

I think I talked about the self-managed teams and how they meet every day on the work that's in process but also things that are coming downstream and one of the things that we're very proud of is up until we start developing, we have a lot of flexibility to change things in the queue, change the priority of work. Our testing is a combination once again of manual and automated testing.

We're not doing as much automated testing as we would like and we're improving that but we have our business analysts, we have our developers and we have our system testing team all working jointly in the testing process. Next slide. I think you guys want to hand this off to Iroabuchi and Tom on this one?

Arum Iroabuchi: Thank you, Fred. Reflecting our impression as to what we would do differently if we had it over again, one of the things that we talked about was trying to streamline the front end to ease use of the system by case workers in the field.

We did a lot of that relative to the intake area when we had to go kind of set through the functionalities that were implemented initially. The streamlined them in a way that allowed them to make logical sense and to allow the flow to be very smooth for case workers in the field.

Regarding the plan for incremental roll-out modules, because of the way we rolled-out Ohio staff with system when we first rolled-out, we rolled-out the entire application all at once.

Highest IBM 2020s, we had it to do all over again, we probably would have effected some kind of change to that methodology that would have allowed us to kind of segment the roll-out of the application as opposed to a rolling-out the entire application all at once.

The other thing we would talk about too was to find a way to maybe to have formalized the dependencies between the different areas.

We find out that when we went through our first initial round of JAD and requirement definitions process, because of the way it was setup in teams there's a sense in which the work was kind of siloed in a way that compromised our ability to communicate across those different teams.

So if we had it over again, we probably would have emphasized maybe spend a little bit more time in trying to establish structures that would have allowed those kinds of crossed in communication to happen. Tom?

Tom Hammons: Great, and next slide, please, so these are the slides that I mentioned that coincide with the other slide that depicted everything on a day-by-day basis so it identifies the activity, the goal, the participant, past, inputs and outputs and we share those with you in hopes that they help and if you ever have a better idea, please share it with us.

We certainly have learned over the years that everyone's input is important and valued and so if you can please advanced the presentation then to

approximately Slide 46 and as they advance, you'll see each one of the steps here and so one more, excellent.

And we also provide you then some supporting information that may help you if you were a system to URLs and Agile, classic reports, findings and other information that might find useful.

One other thing I'd like to add is that we also do yearly planning so we will forecast or a given year when we plan to do our releases and then we break that down into, you know, the four-week structures so everyone can have a very good picture about what to expect about what time of year.

And even though that we work on the four-week structure, we also try to be sympathetic and flexible in working with our end users whether it be a holidays or other types of gears that people are faced with differing kinds of ups and downs whether that be their workloads or vacation schedules and things like that.

So we try to maintain the flexibility even though we have a very frequent release schedule and with that, we'll go to the next slide and hand it back to Joyce.

Joyce Rose: Thank you to the Ohio gentlemen and Arum I spent hours practicing your first name.

Arum Iroabuchi: I'm sorry.

Joyce Rose: That's all right so I found both the Minnesota and the Ohio presentations absolutely fascinating and I think we all have to agree that yes, this is rocket science so let us now - next slide, please - it's time for any discussion if we

have any audience questions to our presenters. Elizabeth or Renee, may we please open the phone lines or chat to our webinar attendees?

Coordinator: Thank you. On the phone line if you'd like to ask a question, please press star 1.

Joyce Rose: Do we have any chat questions?

Elizabeth Mertinko: I'm reviewing them right now. I don't think so but give me one second.

Joyce Rose: Okay.

Elizabeth Mertinko: No, we do not.

Joyce Rose: Okay, and nothing coming in via the phone?

Coordinator: At this time there are no questions.

Joyce Rose: Okay. I...

Man: Not sure how to take that.

Coordinator: One moment, we do have a question coming in on the phone line.

Joyce Rose: All right, here we go.

Coordinator: Mr. Kumar, you may ask your question. Mr. Kumar, you may ask your question.

Mr. Kumar: This is regarding the field agent. What are the technologies being used for this mobile application?

Fred Crawley: Yes, the field assistant is basically it's not a mobile app in the sense that we are deploying something to the device or that we are storing anything on a device. It's actually a mobile Website.

The front end of that basically is (J core E mobile). We're using obviously HTML 5 and CSF 3 but the presentation layer is J query J query mobile. The rest of it is basically Java using sprint.

Mr. Kumar: Thank you.

Fred Crawley: Sure.

Coordinator: At this time there are no further questions.

Joyce Rose: All right. Because we started late and we are going to end late, let us go to our wrap-up, next slide, please, so today we have shared with you the insights of two states regarding their development methodologies, frameworks and platforms and how those choices have actually impacted their release management.

So what's next? We will provide any follow-up regarding any questions so if because I cut the questions off early, if you have any please don't hesitate to send those to me at [joyce@kassetts.com](mailto:joyce@kassetts.com) and then of course the last of the Back to Basics series Webinar is next month and then we will be starting the next set of six Webinars in October.

While the overall theme has not yet been determined, the focus of the next set will be requirements in procurement. So again, this webinar has been recorded and will be made available online. When it is complete and posted, we will send the message via the Child Welfare Managers ListServ with the link.

So I want to thank you, our attendees for attending and of course I want to thank our state presenters and I guess the next thing is I will see you in September. Thank you. Goodbye.

Coordinator: Thank you. This concludes today's conference. You may disconnect at this time.