Late-Binding in Data Warehouses - Designing for Analytic Agility

Late Binding in Data Warehouses: Designing for Analytic Agility

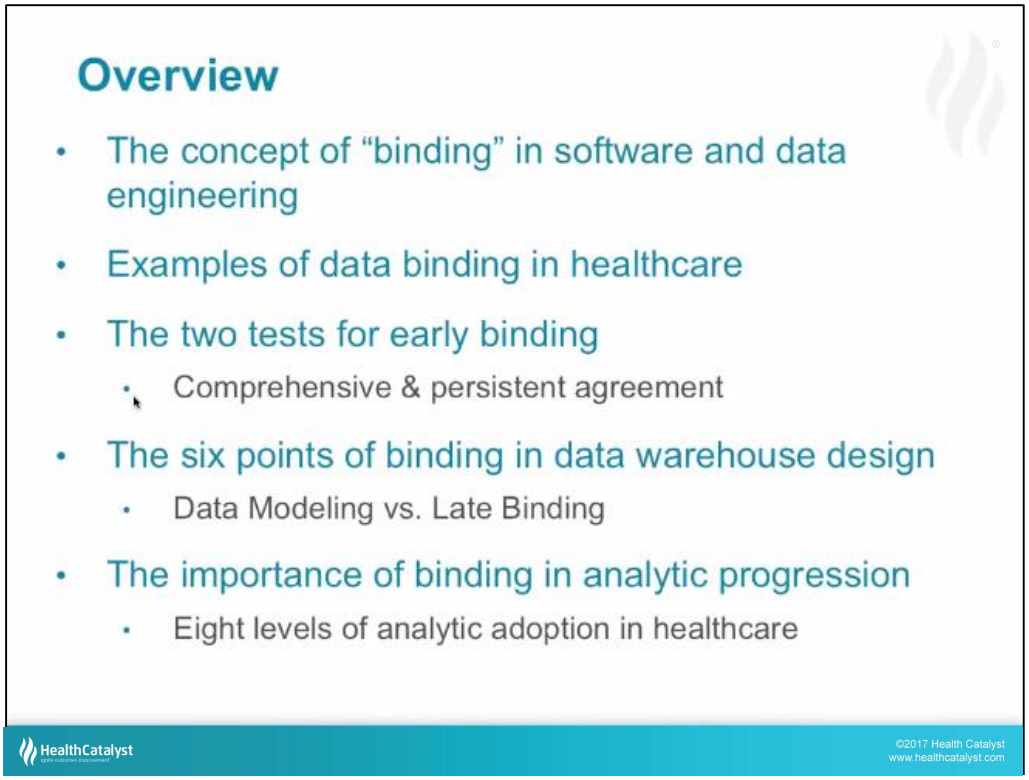
[Dale Sanders]

Thank you, Tyler. Thanks everyone for sharing your time with us today. We hope it's time well spent. We've been using this term Late-Binding ™ in the market commercially now for about two years. I've been personally using it in my career for about 6 to 8 years. And because it's kind of a new concept in data warehousing, we introduced it last year or a couple of years ago. we wanted everybody to have some time to kind of percolate on it, think about it and then this is kind of a repeat refresher on the topic. And so, hopefully everyone or a lot of folks who won't be entirely new and for those who – it is a new topic, we hope, that's interesting as well.

So thanks for joining us everybody in this refresher course.

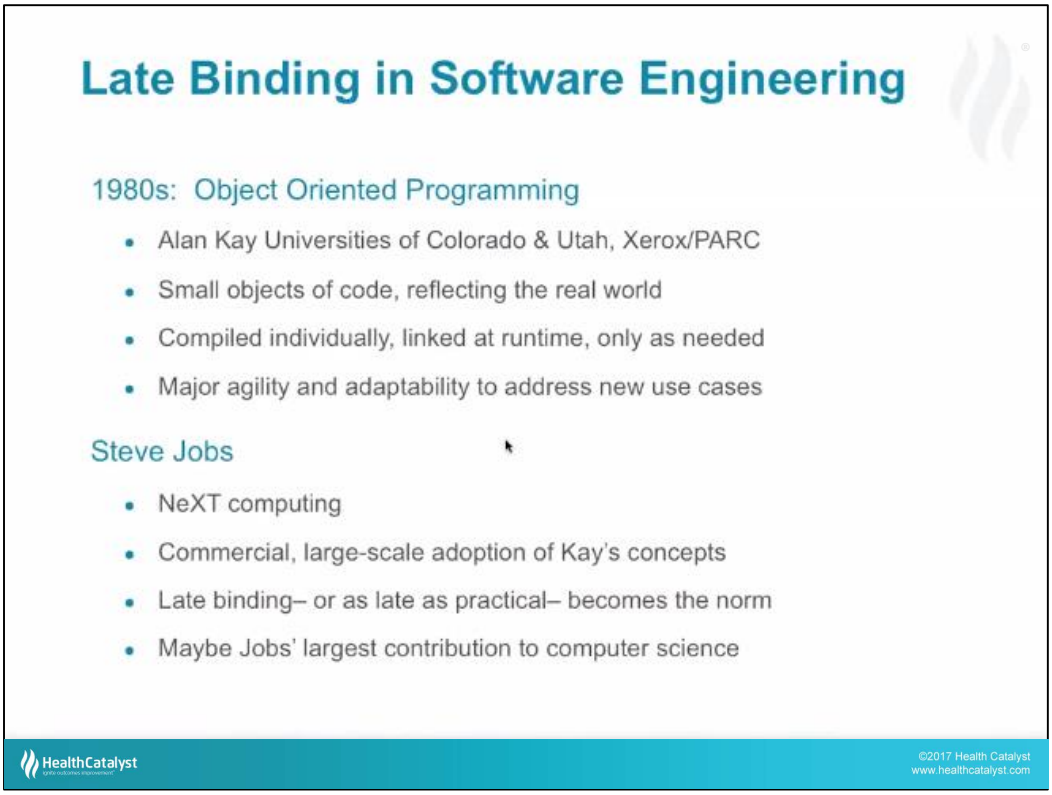
My Philosophy on Webinars

As Tyler mentioned, our philosophy on these webinars and my philosophy in particular is to create an ecosystem of learning that improves all of us. So, we believe that by conducting these vendor neutral webinars to educate and inform our customers in the market in general, we'll put more pressure on all healthcare IT vendors to be better and produce better products and then hopefully of course to spin off from that is better, more affordable healthcare, those software products are doing what they should do. And then that's also, you know, for the vendors that can keep up with this cycle of improvement, it will end up being a benefit to our business as well. And then of course that's all facilitated by these best practices, lectures, and webinars and things like that. So, I encourage all of you to hold us accountable to this standard. We don't want to be preachy. It's not about sales. This is about educating all of us and putting pressure on all of us to do better as a healthcare IT industry.



Overview

So we'll talk about five different things here today in general. We'll talk about this concept of binding in software and in data engineering. I'll give some examples of binding in healthcare that's a little more specific and tangible. We'll talk about these two tests for early binding and its comprehensive and persistent agreement and that's, I call it C&P, and I encourage everybody to kind of stick those terms away. It's a really critical concept to Late-Binding ™ and early binding. We'll talk about the six places where you can bind data in the flow of data in a data warehouse and how that relates back to early and Late-Binding ™. And then I also want to bring this all back to the importance of binding in the progression of analytics and the adoption of analytics in the Analytics Adoption Model that we've developed over the last couple of years. Also, before I go too much further, I want to thank the folks at (02:57) Memorial Hospital. I'm sitting in Keith Jones' office. I'm out here, I had a nice meeting with them and appreciate them letting me use their office for that today. It's good to be back in the middle of a hospital on the front lines. So if you hear a little bit of background noise, it's healthcare being delivered and I'm glad to be here.



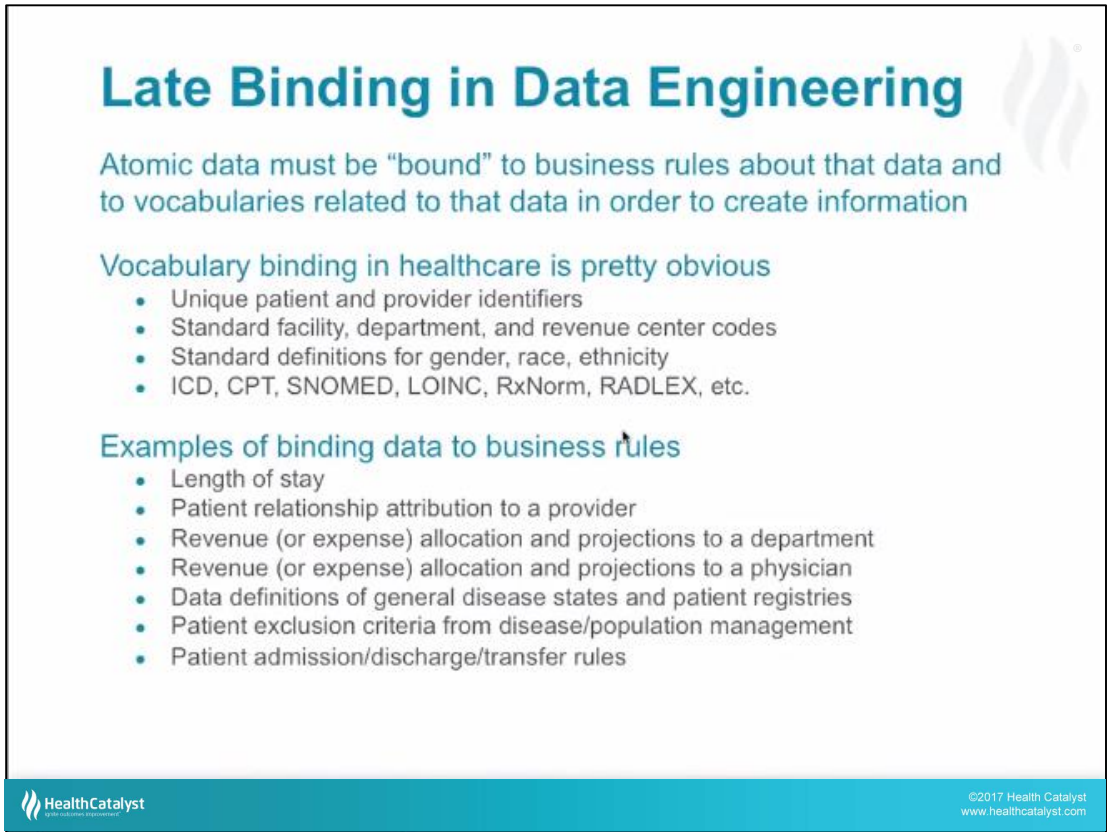
Late Binding in Software Engineering

A little history here, Late-Binding ™ IN software engineering can be traced back to the 1980's. And in the 1980's, I was an Air Force Information Systems officer and a CIO overseeing giant software engineering projects, at that time the world's largest. Now, they fail in comparison to organizations like Google and Facebook and that sort of thing. And we were accustomed to building software. At that time I was involved in software called Jovial and PDP-11 and Fortran and we would write our code in these big giant slots, you know, hundreds of thousands of mines of codes all in one module and we'd link that altogether in compile time and if there was a problem, of course we had to unwind that. There's lots of problems with compiles. Those of you that were around in those days will appreciate all of these.

And then in the 80's, a fellow, named Alan Kay at the University of Colorado and University of Utah and Xerox and PARC, came up with this very simple notion of object oriented programming and that was small objects of code that were written around a reflection of the real world. And instead of compiling those or linking all of those at compile time, you link all of that code at runtime, only when you needed those modules, and it created a major improvement in agility and adaptability software. It really kind of revolutionized software engineering. But I can also remember being in the middle of that change and being completely confounded by this term object oriented programming, object oriented analysis, reading book after book after book on it and going, I really, I'm missing something here because this just seems too simple. I know I'm missing something.

But the reality is it was that simple. It was absolutely as simple and as common sense as it appeared to be. But because my mind was brought up in old school software engineering, it took me a long time to really grasp how simple and elegant it was. And that's really the same message that I want to share with everyone about Late-Binding ™. It's a catchy term. Sometimes I think we hyper-market it just a little bit too much. But when you really look at it, it's incredibly simple. And so, don't overcomplicate it, and that's the message I hope I convey today – is it's probably just as simple as you think it is. Don't over complicate it.

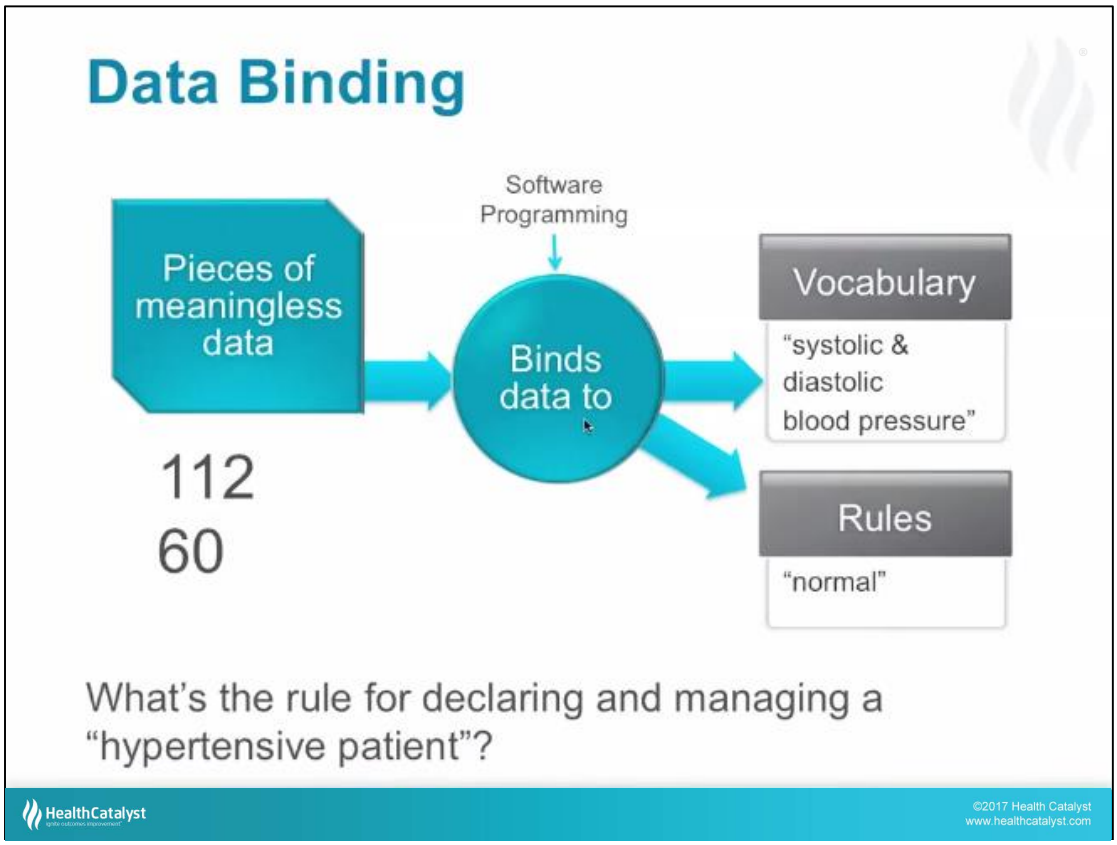
I also want to note Steve Jobs' contribution in this area. He gets a lot of credit for what he did at Apple, you know, with the iPod and the iPhone and I'm using a Mac here today. I love it. But it was actually his contributions at NeXT Computing to the progression of object oriented programming that I think changed the entire industry. So Apple is a company that certainly did well. But what he did at NeXT Computing was he wasn't a programmer, he saw the elegance though of object oriented programming and NeXT was all about object oriented programming. And in those days in Silicon Valley, that was not the case, but he saw it, he started it, he planted the seed, and it took very little time for it to take off in Silicon Valley. So I think it's quite like one of the areas where Steve Jobs probably had more impact on society than even what Apple has had because object oriented programming now is driving virtually everything in software.



Late Binding in Data Engineering

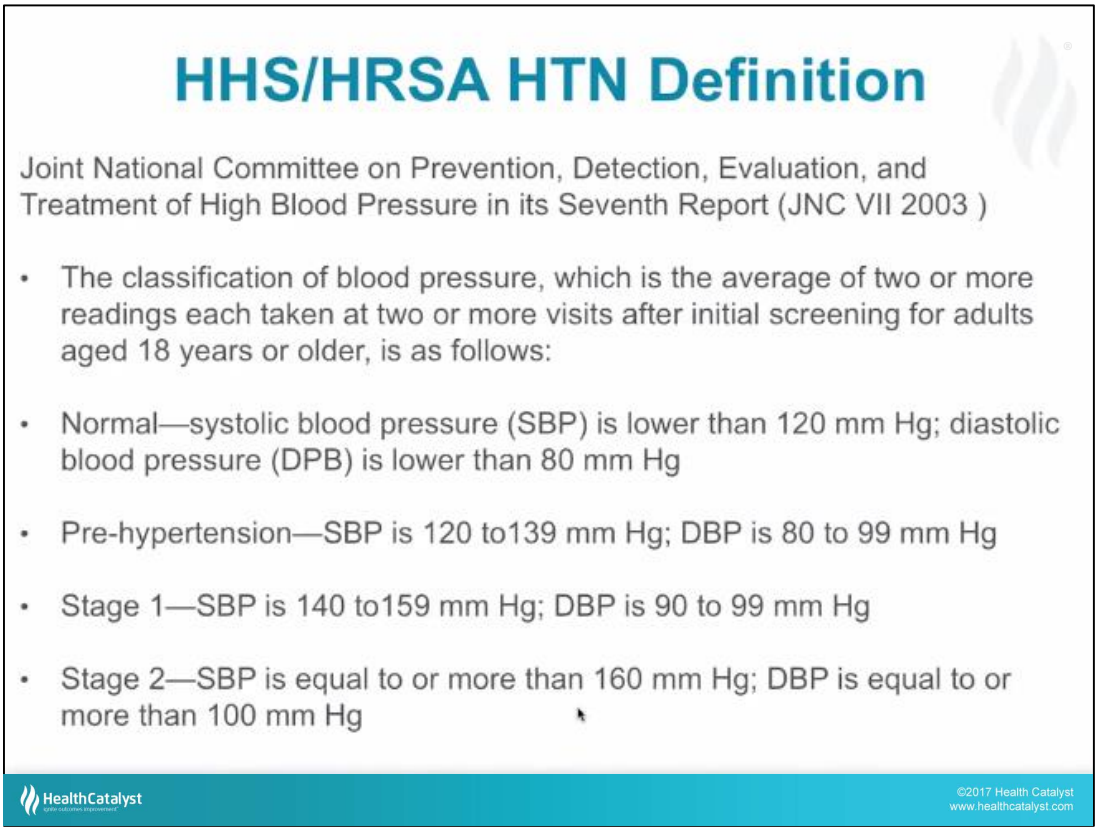
So how does this apply to data engineering? The concept is something along the lines of this, and that is that once you have atomic data, you have to bind that to business rules about that data and to vocabularies that describe it. So some of the examples of vocabulary binding in healthcare are pretty obvious, a unique patient identifier. You have to bind a number to the concept of a unique patient identifier. You have to bind a number to the concept of a provider identifier. Standard codes for facilities, departments, revenue centers, ICD, CPT, SNOMED, LOINC, RxNorm, RADLEX, all of those are just numbers and characters until you bind them to a vocabulary that describes the context.

So, there's another form of binding in data engineering and that's the business rules about the data, the algorithms that do something to that data. So things like length of stay, how do you attribute a patient to a provider, or a patient to an organization in an ACO, how do you allocate revenue or expense to a department, to a physician, the data definitions of general disease states and the inpatient registries, the exclusion criteria from population management or I call it the special management criteria for socio-economic issues and population management. All of those, and patient admission/discharge/transfer rules, all of those are rules that you have to bind data to and define in a standardized way to be successful with analytics in healthcare.



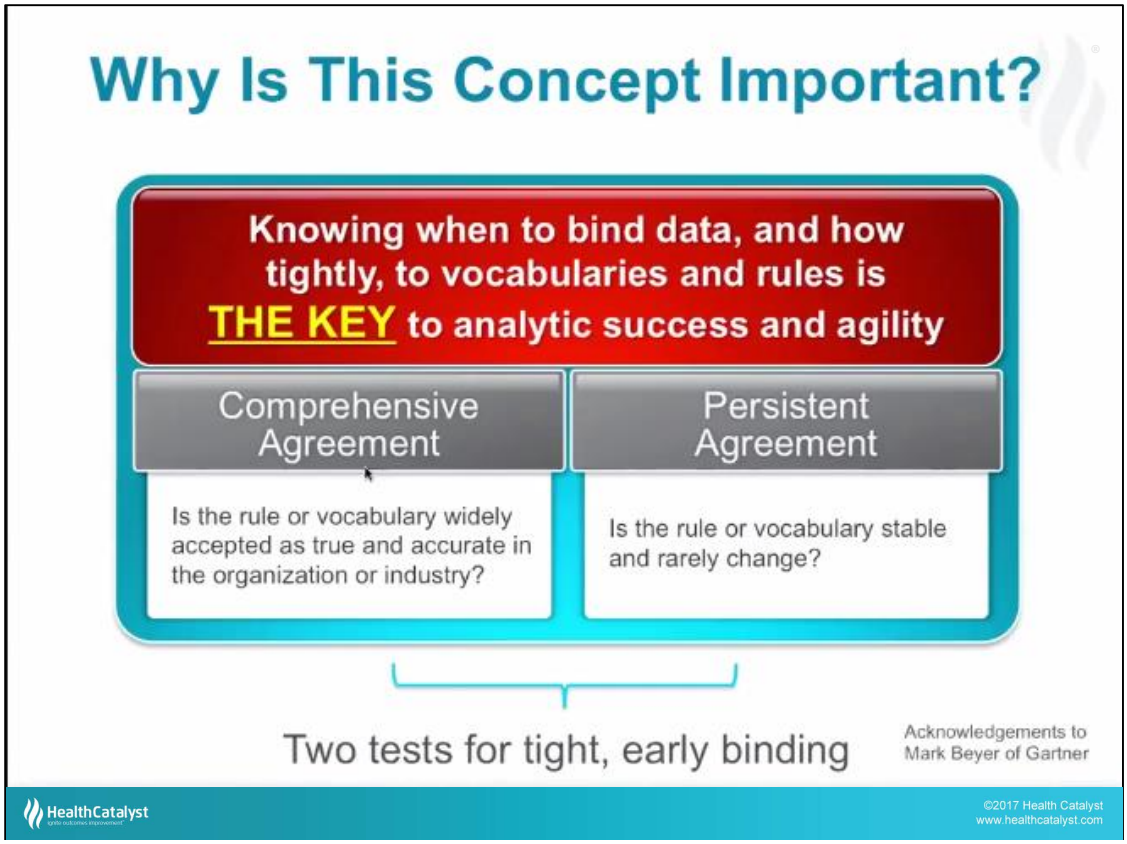
Data Binding

So let's look at this in a diagram and take kind of a simple example. So you've got these meaningless pieces of data here in 112/60 or 112 and 60, when you put a numerator or denominator on it, and through software programming, we bind those numbers to something that has meaning in our context. And in this case it's systolic and diastolic blood pressure. That's one level of binding at the vocabulary level. What ends up being more complicated are the rules about what that data means. And in this case, that's a normal blood pressure reading by most definitions. I think those of you that have been in Healthcare Allina have seen that even though we think we have standard definitions and rules about these bindings, the reality is there's a lot of variability. And if you go from one organization to another, it's quite often to see a different definition for a hypertensive patient. I see that as I travel across the country. I've seen it for years and years. So, the message there is (09:46) field like healthcare where these rules are still evolving and even to some degree, the vocabulary still evolving. The binding that occurs here in this software programming has to be kind of mushy, it has to be malleable, because if you program it too firmly and that's, another term 'tightly coupled software', if you write your software and your data engineering rules too tightly and too early, you don't have the ability to adapt as we continue to evolve and change these definitions of what is a normal blood pressure and what's hypertensive. So that's just as simple as it gets right there. So don't overcomplicate it.



HHS/HRSA HTN Definition

And just for fun, I copied the HRSA definition for hypertension last night and there it is. So you can see, even that definition, and there's some debate about this, and you'll see the World Health Organization, they have yet a different definition for hypertension. It's a fairly complicated rule for what you would think would be a fairly standard and commonly understood thing in healthcare by this time. But if you look at all of this, there's a lot of programming that goes into this. And so, the binding is not only fairly complicated but it can hold you back if you bind too firmly to this rule and over time it evolves as our understanding of hypertension evolves. And by the way, I think this came out firmly as the new standard only in, I believe it's January of 2014. Even though this came out clearly in 2003, I believe it was only endorsed in January 2014 and if there's a contrary thing to that, I'd appreciate someone bringing that up in the question. So anyway, an example of a rule and an example of how you have to be kind of mushy in the way that you bind that data analytically because more than likely it's going to change.

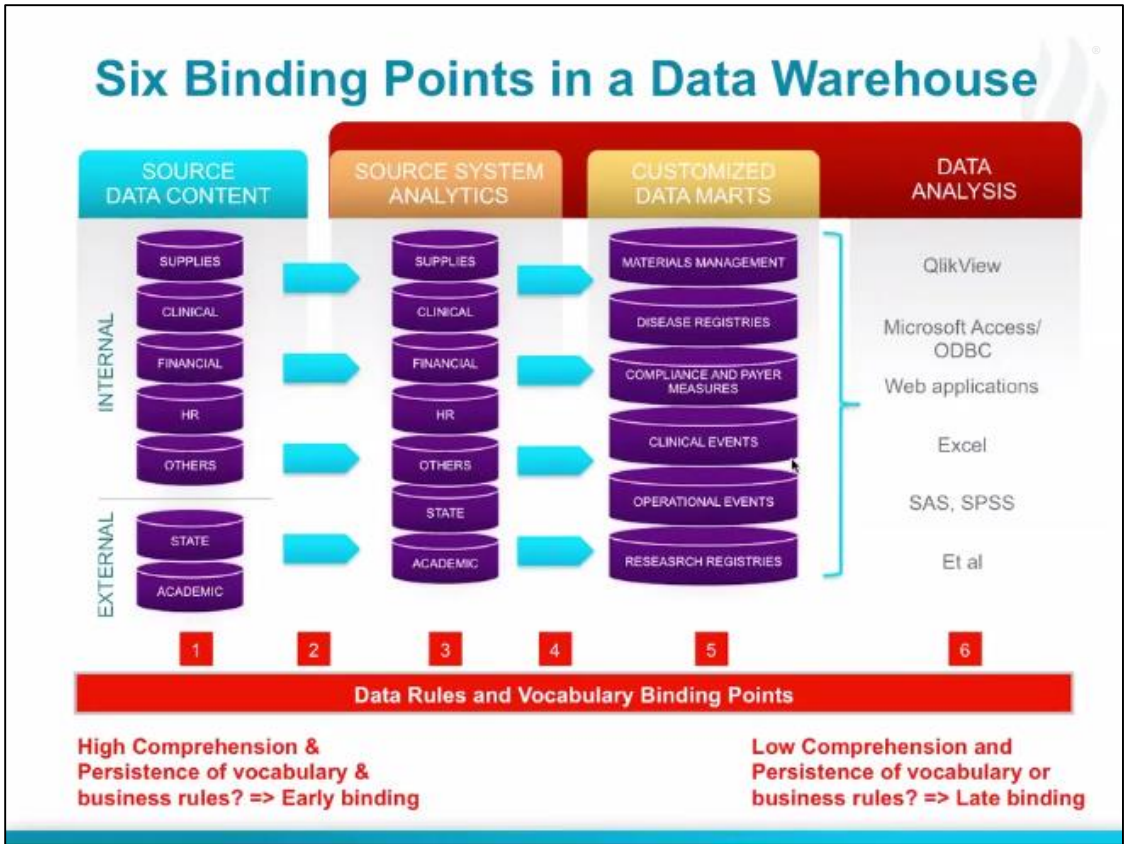


Why is This Concept Important?

So, why is this concept important? Knowing when to bind data is critical and how tightly coupled you bind to that rule is also really critical. It's the key to analytic success and agility.

So, the two rules, and it's a mandate between these two, you have to have comprehensive and persistent agreement about the rule or the vocabulary before you tightly couple it and bind it early in the flow of data in the data warehouse. You can't be one or the other. It has to be comprehensive, sort of widespread throughout your organization at least, or maybe the industry. And it has to be pretty persistent. It has to stick around for a while. So it can't be that you agree on a widespread basis today but tomorrow it changes because if you bind too tightly in your software, you're not going to be able to keep up with those changes. So you have to look for both of these. So this is where being a data engineer and a software engineer requires more than just technical skills. You have to have an understanding of the volatility of the environment that you're working in, the business and clinical environment that you're working in, and you have to be sensitive to these cues.

So it's not unusual to sit in a room full of physicians and nurses and others where there's a debate about a particular rule for defining a patient state or a disease state. And you have to be able to sit there as the data and the software engineer in those rooms, in those meetings and hopefully you're participating in those discussions, and you have to ask yourself, am I seeing the signs of comprehensive and persistent agreement. And until you see that debate settle down, you better make sure that the software and the data that you engineer is loosely coupled and follows Late-Binding ™ concepts because otherwise you're going to be constantly going back and reworking that. So, parameterizing object oriented data engineering, software engineering, really important until you see this equilibrium take place in the settlement of the discussion.



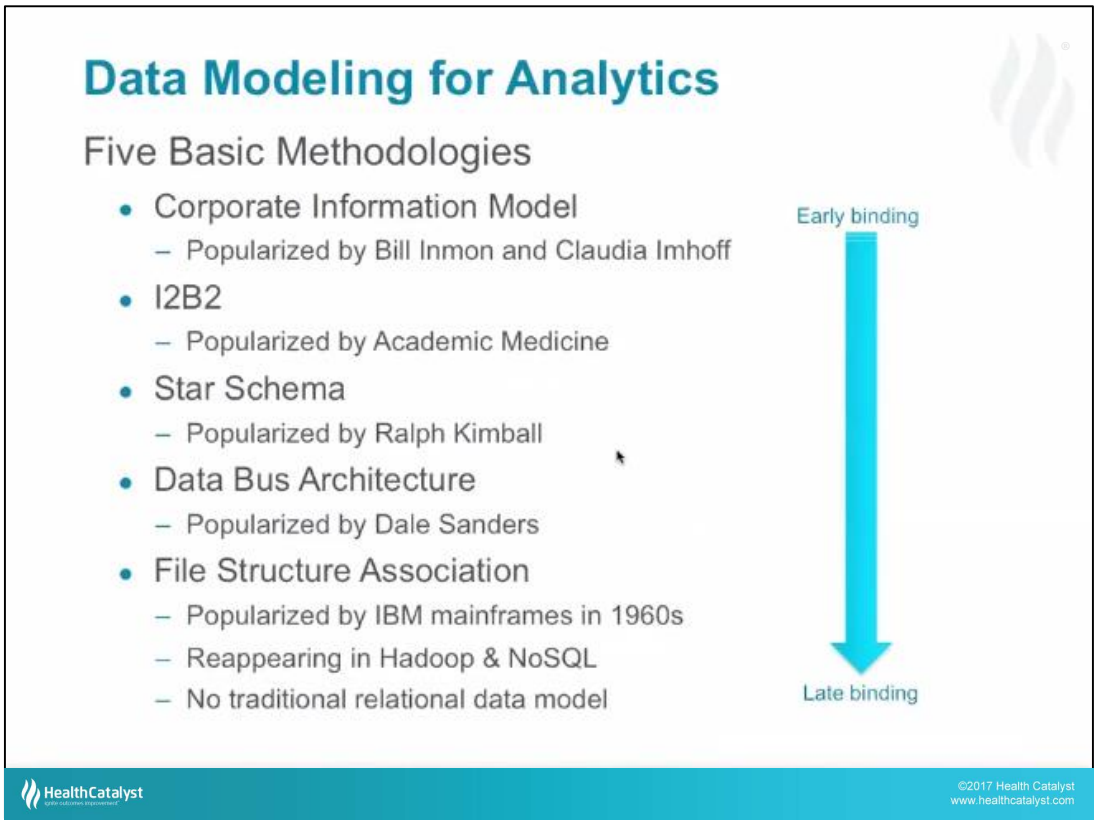
Six Bonding Points in a Data Warehouse

This is kind of a typical diagram. I'm sure a lot of the IT folks in the audience have seen this. This is a flow of data from left to right in a data warehouse environment, in an analytic environment, and it turns out that there are six places in my observation over the years where you can bind to a rule or a vocabulary and you have to be very consciously aware of where you bind in these six places because everything downstream of that is going to be constrained by that binding. And what that translates into is less agility to support new analytic use cases if you bind clearly over here at points 1 and 2 before you reach comprehensive and persistent agreement. That means all the binding that you impart upon the data, over here points 1, 2 and even 3, all the analytic use cases downstream of that are going to be held accountable to and tightly coupled to, whatever you've decided to bind to here in levels 1, 2 or 3. Now, it's completely fine if you want to and if you see comprehensive and persistent agreement about the data vocabulary and about the rules, it's totally fine to go ahead and bind early here in 1, 2 and 3 because these can kind of take it off your plate. You don't have to worry about it anymore. But I would caution, in healthcare, it's such a volatile analytic environment right now. There's not a whole lot that you can safely bind to in those early levels without suffering some kind of negative consequence downstream in your ability to support new use cases.

So, applying that question, if there's high comprehension and persistence of vocabulary and business rules, go ahead and bind early in these areas. If there's low comprehension and

comprehensive agreement, then you need to let it be a little more mushy and let the binding take place in levels 5 and 6. In fact, one of the things that advocate is that if there's a lot of debate, a lot of uncertainty about a particular vocabulary or be more likely a rule about a particular analytic use case, give your end users, give those physicians and those nurses and those administrators, give them a tool that allows them to explore those bindings in this visualization layer. That's where you're doing the 'what if'. What if we change the definition of a hypertensive patient, what if we change the definition of a diabetic patient, what if we change the algorithm for length of stay, what if we change the definition for readmission. And let folks go through that exploration in that visualization layer. And then when you see that settlement of discussion and you start to see that comprehensive and persistent agreement about what it means to be a diabetic or a hypertensive patient or what it means to have a readmission, then you can move the binding from point 6 up into point 5 or even maybe 4 or 3 in your programming. In fact, you want to do that because that starts to ensure consistency of the truth and consistency of the analytics that you're providing.

So, let me pause there and kind of let everyone absorb that diagram for just a minute and we'll move on to the next.



Data Modeling for Analytics

Okay. So there's basically five methodologies for data modeling in healthcare and I've listed them in kind of early to Late-Binding ™. So the corporate information model that's been advocated and popularized by Bill Inmon and Claudia Imhoff is a very early binding model. And in some industries, in particular retail, where a retail transaction has been around for many many years. I mean the basics around a retail transaction haven't changed in 5,000 years literally. And so, if you can look at that environment, you can say, you know what, there's a lot of comprehensive and persistent agreement about what constitutes a retail transaction. So using an early binding data model, one of these corporate information models, in those settings is not bad. Points processing in healthcare and the reality is points processing really hasn't changed much in many years. So, the data warehouses that support the insurance companies have a pretty standardized data model. It's okay to bind early to that data model. The rules haven't changed much since Medicare came along. They do change but not dramatically. Nothing like what we're undergoing on a care delivery side and accountable care. So it's okay to use those models in those industries and in those scenarios where you've got a lot of stability in the understanding of the environment, that comprehensive and persistent agreement.

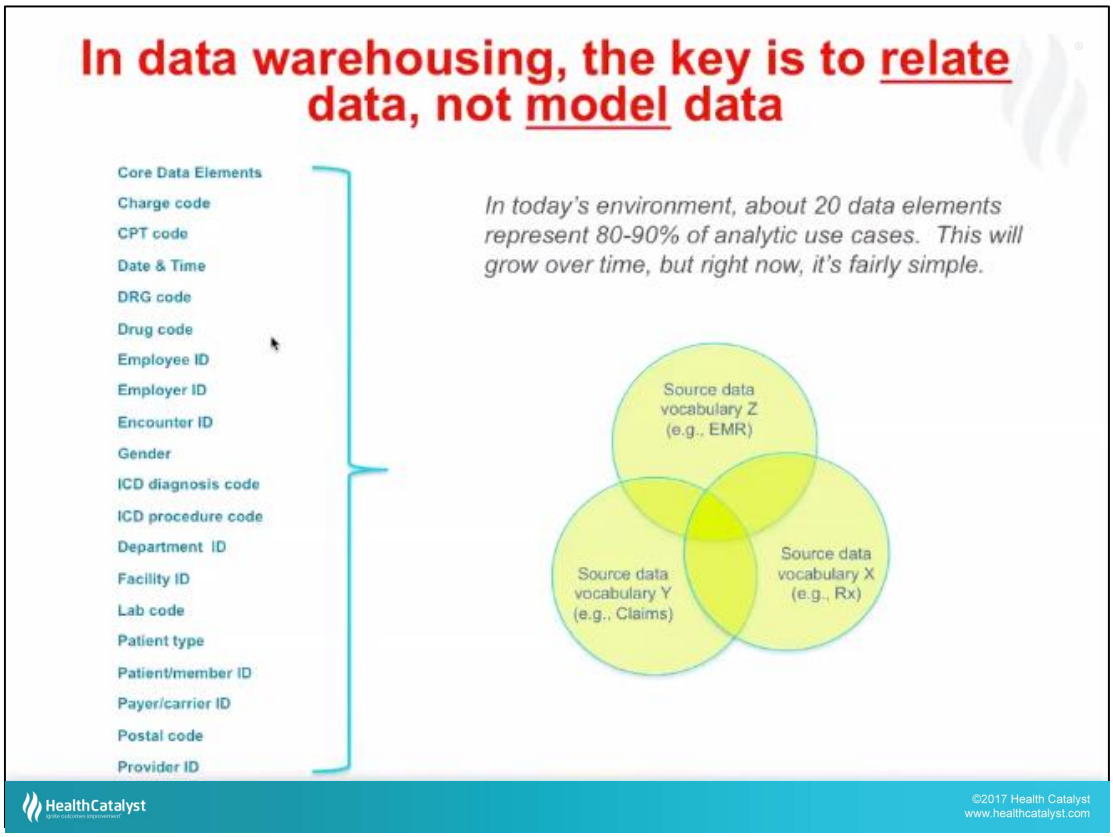
I2B2 was popularized by Academic Medicine. It was intended to create an ability for academic medical centers to share cohorts of patients and explore cohorts of patients in kind of a virtual sense and it has its role. The truth is it's not very useful in the kind of detailed analytics but we all need to support accountable care, but it's not bad to meet the needs of academic research centers and the ability to find quick cohorts of patients. I would argue that the methodologies for data modeling that I advocate will still allow you to find and manipulate very quickly cohorts of patients to support research. But at the same time, that same data model and that same approach to Late-Binding ™ will also support a whole bunch of other analytic use cases. So, I'm a big fan of the intent and the aspirations of I2B2. I'm less a fan of how effective it is as a truly effective tool for analytics in healthcare.

Star Schemas were popularized in the mid-90's by Ralph Kimball. Again, kind of a retail-driven environment. Star Schemas are really good for kind of quick counts of things and in environment. They're particularly effective in materials management. They can be effective in analysis of billing data, even claims data. And they're very fast in kind of easy-to-use with tools that recognize the Star Schema. So they have their benefits. But they also have a glass ceiling that a lot of organizations are hitting right now in healthcare that assumed that these Star Schemas could expand in scope to meet the more complicated analytics required for accountable care and what I call complex numerator analysis of patient subsets. So, I'm a fan of Star Schemas in a limited sense but certainly not as the centerpiece of a data warehouse in healthcare.

This Data Bus Architecture, I lectured about this a long time ago. A friend of mine who was in the electrical engineer actually gave me that term. Ralph Kimball popularized it in a couple of books but I think I actually beat Ralph Kimball to the punch when I started using the term. Again, it he was an electrical engineer that said why – a buddy of mine watching us struggle to create this big information model to support nuclear ICBM data analysis, it had I don't even know how many tables. 5,000 tables in it probably. I'm trying to analyze data from 500

different source systems. An electrical engineer, a buddy of mine, came in and said, "Why don't you guys just follow a Bus Architecture here and plug all these different data sources into a common vocabulary in something that they can link across? But why in the world are you remodeling all that data?" And so, that's when I started this journey of breaking away from the centralized data models as a centerpiece of a data warehouse. That was back in the mid-90's.

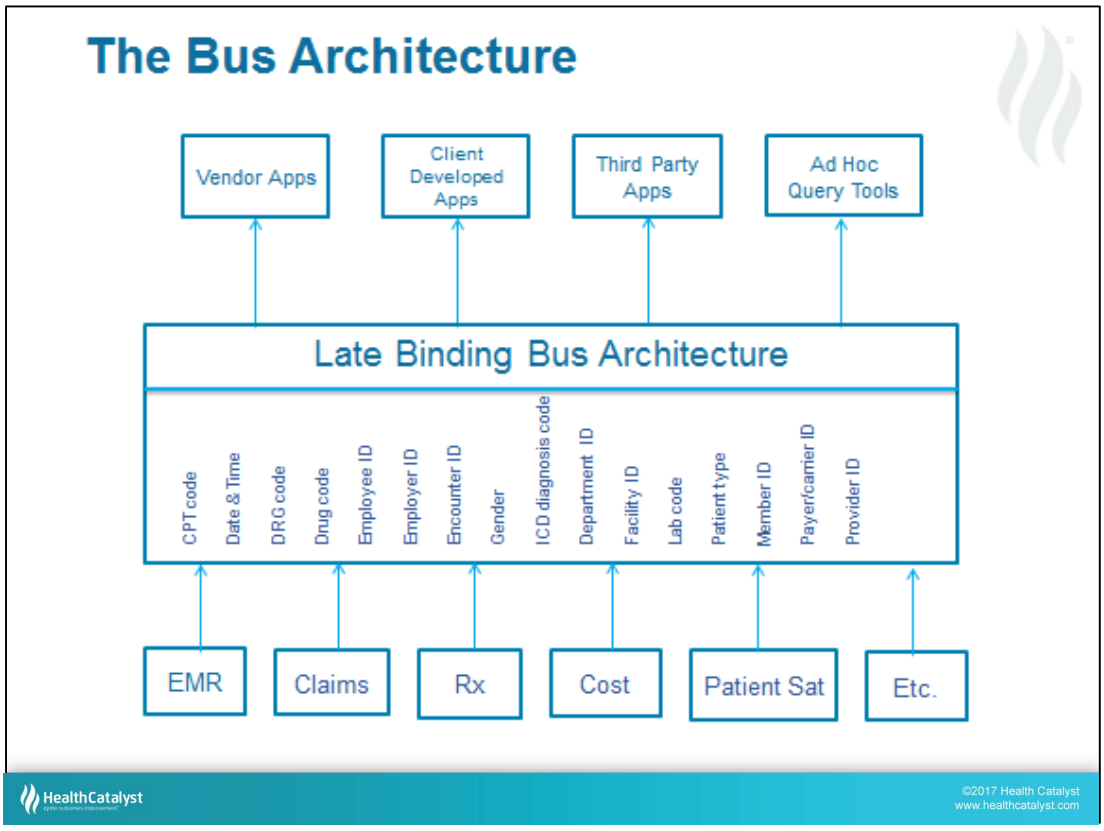
Finally, there is what I call File Structure Association, and it was really the only method of conducting data warehousing centered around IBM mainframes going clear back to the 1960's. But it's interesting for me to see if the concept is reappearing in Hadoop and NoSQL. I mean the tools are pretty dramatically different but the concepts and the design are not much different than what we were doing with IBM mainframes for many many years where you have these kind of big aisle or these big lakes, I guess that was the term, of data where you can just pump data into this big and kind of ASCII style file formats and then you wrote programs over the top of that to create the relationships. There was no relational database engine at that time doing that for us. We had to do that ourselves with programs. And we didn't impart any kind of strict data model on those ASCII files. So it's kind of interesting to see the role coming back to that in Hadoop and NoSQL and it's very cool what Google and Amazon and others have done with Hadoop and that whole environment. It's going to have a big impact on healthcare, although not as big as everyone thinks it is right now. That's a separate topic and I'd be happy to entertain questions and thoughts about what I think the appropriate use of Hadoop is in healthcare right now and how to migrate towards that.



In data warehousing, the key is to relate data, not model data

So, what I always emphasize is that the key to data warehousing in healthcare is not data modeling. In fact, I believe there's an inverse relationship between the data modeling that goes on and the success of a data warehouse. If I see an organization spending enormous amounts of time on data modeling and data mapping, I can almost guarantee their failure because you get caught up in that cycle and you never get out of data modeling, you never get out of data mapping. They talked about this, that, and the other forever and you never release the data and start exploring it.

So the key is to model less and relate more. And in today's environment, there are about 20 data elements that represent 80% to 90% use cases that we're dealing with and they're listed on the left-hand side there. So this is going to grow over time and it's growing as we speak right now. But to be honest, most of the vocabulary issues right now in healthcare are fairly straightforward and easy to define and you need to focus on them as they overlap with one another, not try to create a data model that encompasses all three of these source systems that I've shown here in this Venn diagram but just focusing on those overlapping areas that allow you to relate data between those three areas. I don't need to remodel these three source systems. All I need to do is create a common overlapping vocabulary at a data type level at a column level that allows me to relate data between those systems. I don't need to remodel it to relate it.

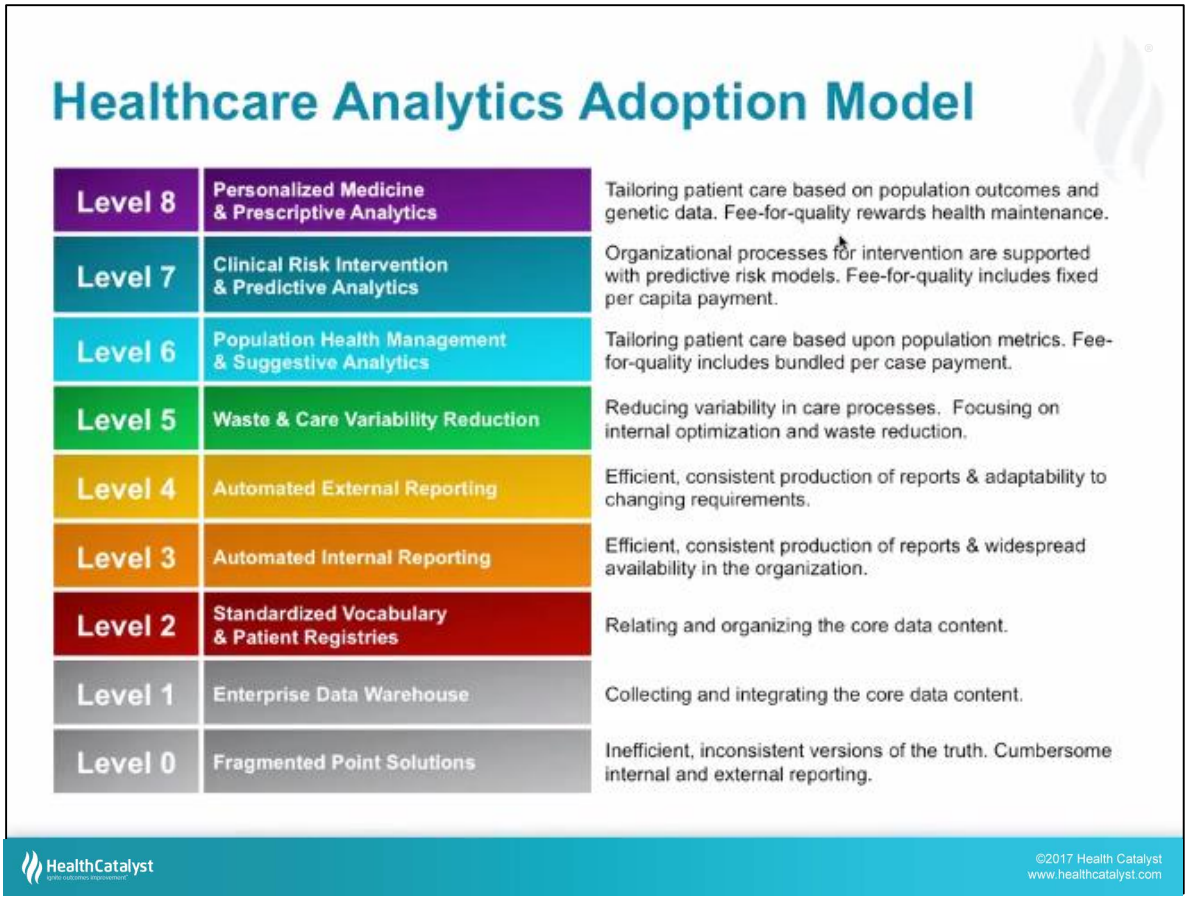


The Bus Architecture

So, here's just a quick diagram of how this Bus Architecture functions. You've got all these different source systems, the data in the healthcare environment, and in Accountable Care Organization, it's increasing exponentially. So you've got literally dozens that will be hundreds and eventually thousands of data sources that you have to feed into a data warehouse. Well, there's no way that you can keep up with that if you're trying to map that into a common data model all the time. You just know why it's not going to happen. And I've tried it, so I'm speaking from painful experience.

So, the thing to do is to land that data in a data warehouse and bind it together with these very commonly understood, persistently, comprehensively understood dimensions of analysis that we all know so well, and some of those are listed here. So now I haven't remodeled anything. I I've just added what amounts to foreign keys across these source systems and I can cross-lock my queries across these source systems right now and I can make these look like an Enterprise Data Model without actually having to go through the pain of it. Now, that's what a Bus Architecture is all about. And by the way, it's a conceptual thing, it's a logical thing, it's not a hardware device. It's literally just as simple as adding common foreign keys to the data structures that you land in a data warehouse from these source systems.

Finally, on top of that, you can supply data to all sorts of different applications once you have your hands on that data. So, you can supply data from this bus to products like Crimson or Midas. You can develop your own applications against this data warehouse architecture. You can supply data to CMS from this architecture. You can expose it to ad hoc query tools. I've even used this to support call center operations. So it doesn't have to be an analytic use case. Once you have your hands on the data, you can do anything you want to with it. So the call center application that we developed around this model supported our patient customer service. So as soon as a patient called in, their telephone ID kicked off an integration to the data warehouse and they'll pop to summary, still protecting the patient's confidentiality and privacy but that popped the summary of their recent encounters to the healthcare organization, just like you would expect to see when you call in to American Express or a myriad or anyone else that uses telephony integrated with the call center. So, you can use this architecture for analytics but you can use it for what I call secondary use of transaction data as well.



Healthcare Analytics Adoption Model

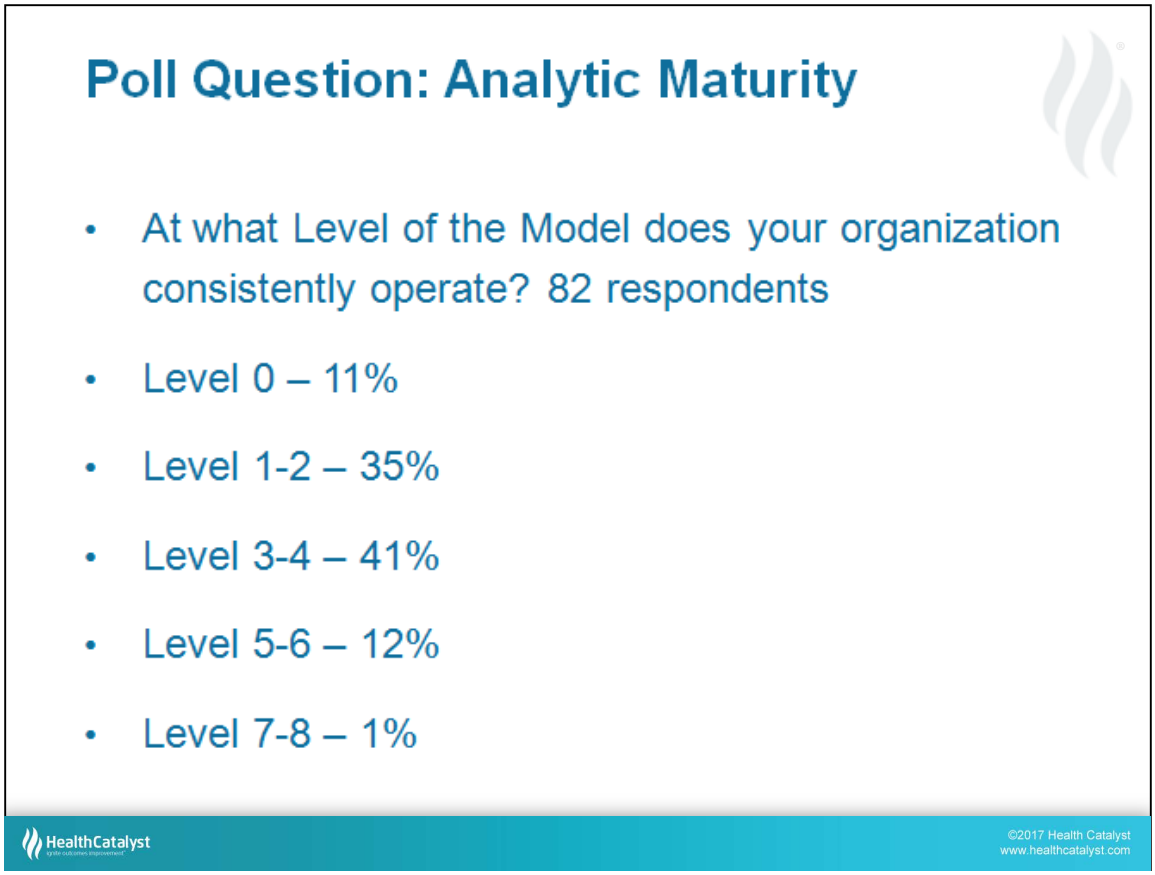
So, let's talk about this Healthcare Analytics Adoption Model for a bit. So a group of us have been working on this for 15 years. Over time, that's how long it took to evolve. It only really took off the last few years with the points spent on interest in analytics in the industry. But what I tried to do and what we tried to do here was show a progressive model of adoption of analytics in the industry. And the reason it's progressive, and it's important I think to progress through this somewhat linearly, is because you learn as you go. And so, it's not unusual for folks to aspire to and take on projects up here at level 7 or 8, personalized medicine, prescriptive analytics, predictive analytics. Those are all really attractive bright and shiny terms. But it's like trying to understand or take a quantum mechanics course in college when you haven't passed fundamental college algebra. So you can go up there and you can spin your wheels and you can spend a lot of time and you can get the executives all interested in these bright and shiny topics, but while underneath, you're still struggling to produce very basic reports. And the reality is because you haven't gone through this progressive learning experience, you're probably not doing a very good job at level 7 anyway. You think you might be but in reality you're not. So the degree that you can move through this in somewhat linear fashion like a course curriculum is my recommendation.

Now, let me talk about this for just a second, the progression of this. So first is putting in the infrastructure of the data warehouse, getting the basic patient registries and vocabulary put together, as I depicted in that Bus. So there should be query across all the data content. Then starting to bind your data to the rules associated with internal and external reporting and the next logical steps. And in those two levels, you can use a fairly early binding model and principle because those are, for the most part the internal reports in healthcare are for the most part, they've been around for a long time. We understand those. They're pretty stable. They pass that comprehensive and persistent agreement rule pretty well, although I would also argue that we don't even have a common definition for length of stay truthfully across the industry. But we have probably 80% stabilization of our understanding at that level. And the same thing started to happen at level 4, where the federal government in particular is starting to impose the definitions for reports upon us, whether we think they're meaningful or not. We know what the rules are.

So we can bind our data to those rules with relative comfort in that early binding and relatively tight coupling because it's probably not going to change very much and it certainly has become comprehensive. If it's pushed out by the federal government, you can say that's affecting the entire country. Where we try to get people, and this is kind of a Health Catalyst® principle of operations, we try to get folks pass those levels just as quickly as possible. So, everything below level 4 should be pretty much commodity technology by now, and we try to get that deployed just as fast as possible because where you want to get to as an organization is at levels 5, 6, 7 and 8 just as fast as you can because that's where you're starting to save lives and save money and differentiate yourself as an organization. So the faster you can get to level 5, the better.

Now, what happens a lot of times is vendors will come in and they will show you level 3 and 4 because that's kind of the easy step to produce right now. The hard thing for an organization to do is levels 5 and above and that's where the flexibility of a Late-Binding ™ Architecture comes into play. So, if you have no other choice, certainly achieving level 3 and 4 reporting is not necessarily a bad thing. That's okay. But if you have aspirations in the near future of getting above level 4 in your analytic maturity, you better have kind of a Late-Binding ™ infrastructure to support that because these early binding models that support only level 3 and 4 like Accountable Care Reporting for example at level 4, they're not going to grow with you as you increase your analytic maturity and complexity of analytic use cases up this model.

So, I think we have a poll coming up. Yes.



Poll Question: Analytic Maturity
AT what level of Model does your organization consistently operate? 82 respondents

So kind of let's go back to this because we're going to ask everyone, at what level of the model does your organization consistently operate? What's the highest level of that model that your organization is consistently operating right now? Let me go back to that for just a second and I'll pause and take a look here, where do you think your organization is operating consistently in this model?

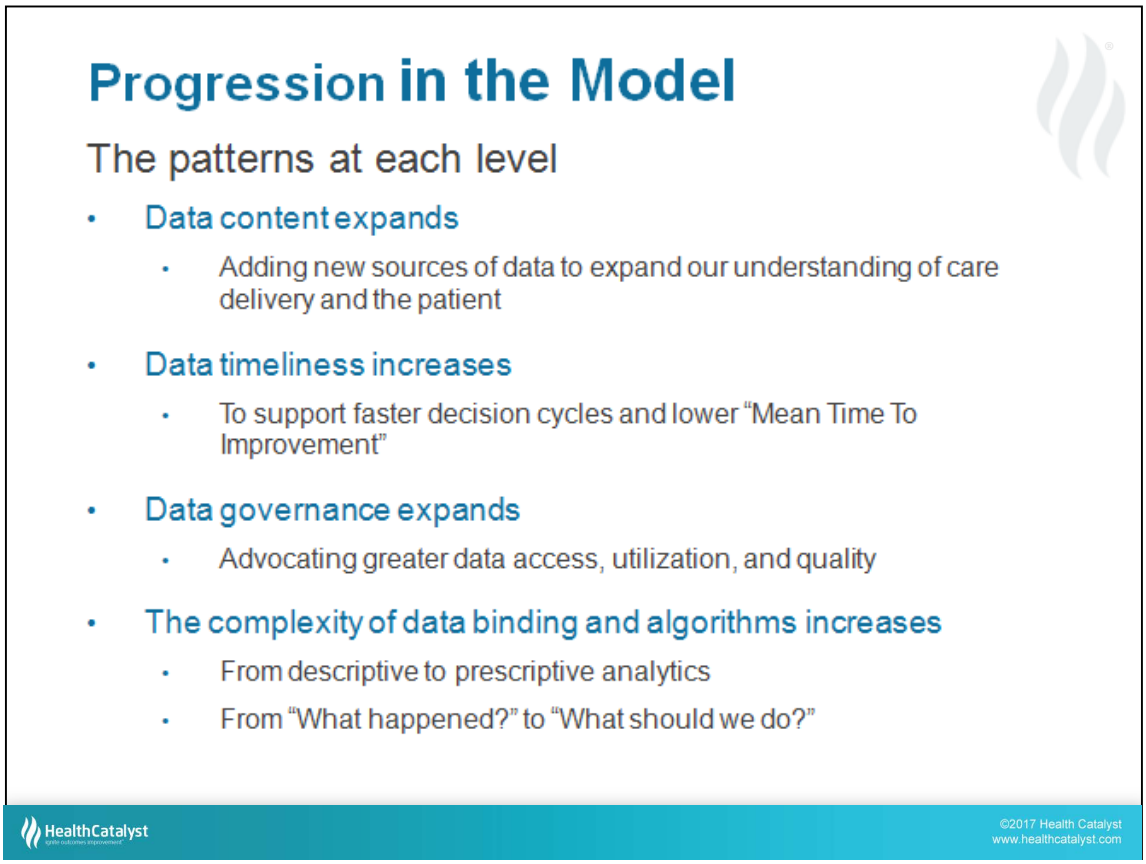
[Tyler Morgan]
While everybody is filling out the poll, I'd like to remind everyone that you can submit your questions by typing in your question in the questions pane in your control panel. Alright. I'm going to go ahead and close that poll and let's share the results. I'll read those up to you, Dale, if you can't see those results. We've got 11% at level 0, 35% at levels 1 and 2, 41% at levels 3 and 4, 12% at levels 5 and 6, and 1% at levels 7 and 8.

[Dale Sanders]
Okay. Pretty similar to the numbers that we've seen in the past with similar folds, wouldn't you say?

[Tyler Morgan]

Fairly similar. I think that the largest ship is just from levels 1 to 2 to levels 3 and 4 from when we did this previously.

[Dale Sanders]
I see. Okay. Thanks, Tyler.



Progression in the Model

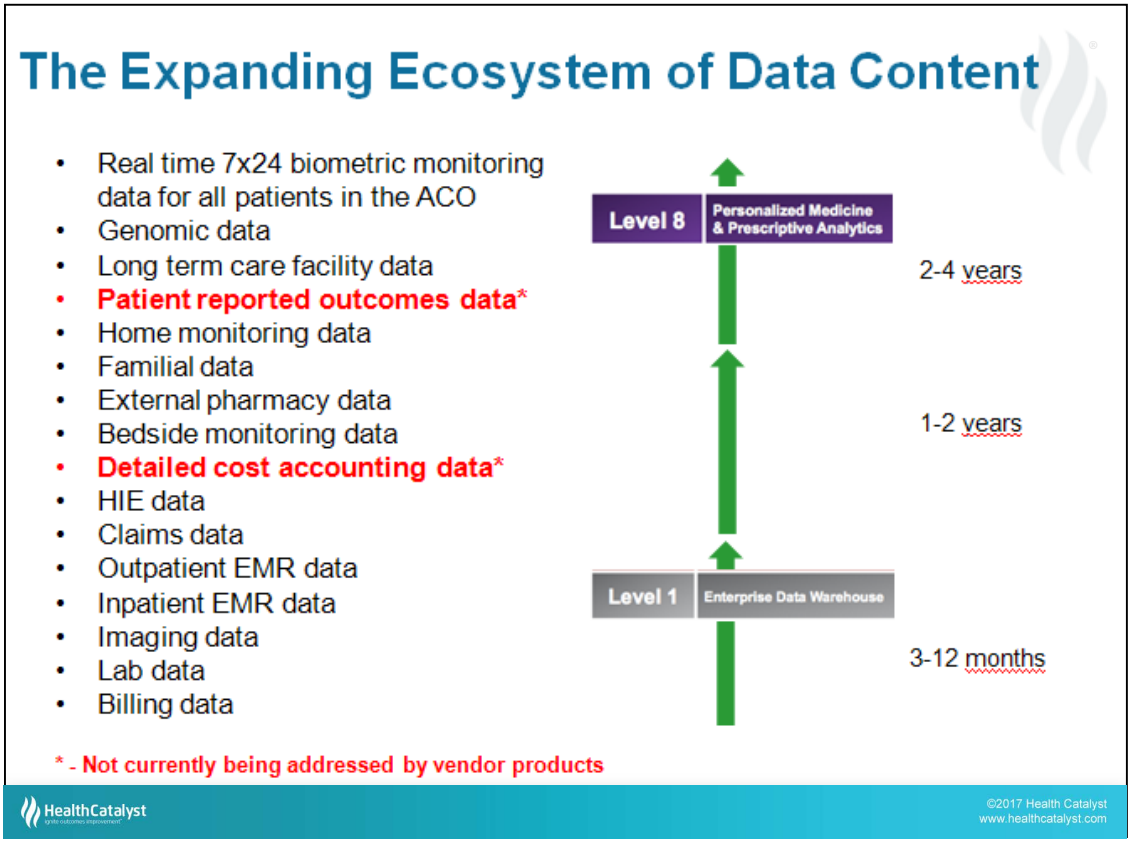
Okay. So let's go back and look at some of the patterns that are in this model. They're kind of at the middle layer for just a second. As you progress up that model in those levels, data content expands. So you're adding new sources of data to each of those levels. Virtually everytime you're at a level, you're expanding the content of the data in your analytic environment because you need to broadly understand. So you're adding – and I'll go over this a little bit more later but adding content is really part, an important part of that progression.

Timeliness of the data increases because you're starting to speed up your decision cycles and you're lowering what I call your "mean time to improvement." So your organization will get better and better as you progress in turning that data around and actually driving quality up and cost down and that will drive then your mean time to improvement. So what that means is you've got to be able to incorporate data content faster and faster all the time as you go up. So

you can't sit there and spend months mapping new data sources in your data model. You can't do that. They're just not time in the market. And as your organization progresses and its maturity and its hunger for data, your data warehouse team won't be able to keep up with that.

Data governance will expand. So you'll move from really basic stuff at the lower levels about, you know, what do we want to support and kind of what are the key metrics and you'll turn that data governance body into a strategic data acquisition body, where you're plotting out the acquisition of data to round out your understanding of healthcare.

And finally the complexity of data binding and algorithms increases as you go up in each of those levels, from descriptive to prescriptive analytics. So you're moving from what happened to the data suggesting what you should do in reaction to a situation.



The Expanding Ecosystem of Data Content

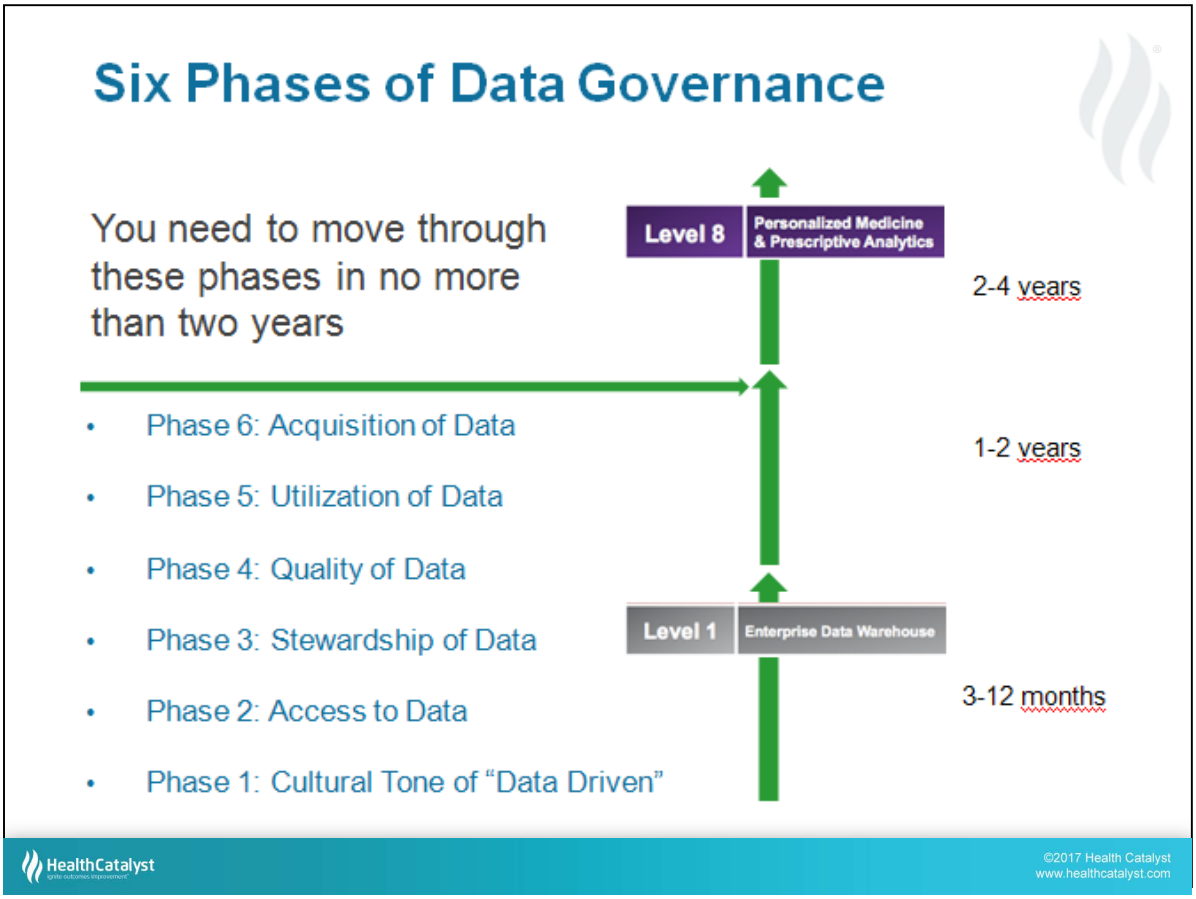
So, talking about that expanding ecosystem of data content, really critical in Late-Binding ™ because Late-Binding ™ allows you to keep up with this progression of data in the environment and it's inevitable. And it's not just inevitable, it's critical to our future as an industry. And so, the data governance committee needs to think about itself as its strategic data acquisition committee. So for instance, right now we know that billing, data lab, data imaging, inpatient,

all of this data is kind of within our grasp right now and we should be working on the integration of all these right now. It's in our environment. And you're going to need it to get up to those levels 3 and 4 in the model. Right now, there's a big missing piece of data in healthcare. I mean it's been there for many many years but it's becoming more and more critical and that's cost accounting data. So at a time when we are trying to take on greater risk as healthcare providers but yet we really don't know what it costs to treat the patient or to achieve an outcome, this is a big big missing piece of data in the ecosystem. So, your data governance committee needs to be working with and putting pressure on vendors to provide products to fill that gap. We're probably 10 years behind where we need to be but we only have maybe one or two years to close that gap.

On your list of data to acquire and bind in your data warehouse is bedside monitoring, external pharmacy data, familial data, eventually getting up to home monitoring data. Patient reported outcomes is still something that we don't have access to right now, and by the way, Press Ganey patient satisfaction data does not count. I'm talking about activities of daily living kinds of metrics, SF12, SF36 data, that kind of thing. Long-term care facility data, genomics, and then finally real-time 7/24 biometric data. It's really going to help us understand what's happening in the ambulatory space.

So, there's two messages I want to leave behind here. One is your data governance committee better start putting this on a roadmap. Imagine tilting this thing 90 degrees to the right and laying a timeline over the top of it, just like I've done here, and putting in place a strategic plan to have real-time 7/24 biometric data in your data warehouse within the next 4 years. What are you going to do to do that.

The other message is here is if you're going to keep up with this in healthcare, you can't follow an early binding tightly coupled data model strategy. Your data warehousing team, your vendor simply can't keep up with this pace of change. You'll be spending all your time mapping data and none of your time analyzing data. So you have to follow a Late-Binding ™ approach. And by the way, Late-Binding ™ is not something that Health Catalyst might have trademarked the name but we want everyone, we want all of the competitors to Health Catalyst to start using this. It's not like we have a lock on it. We want the entire industry to make this change because it's so much better for everyone to do that. We'll compete on different grounds. This is not something we should be competing on. There should be commodity kinds of design principles.



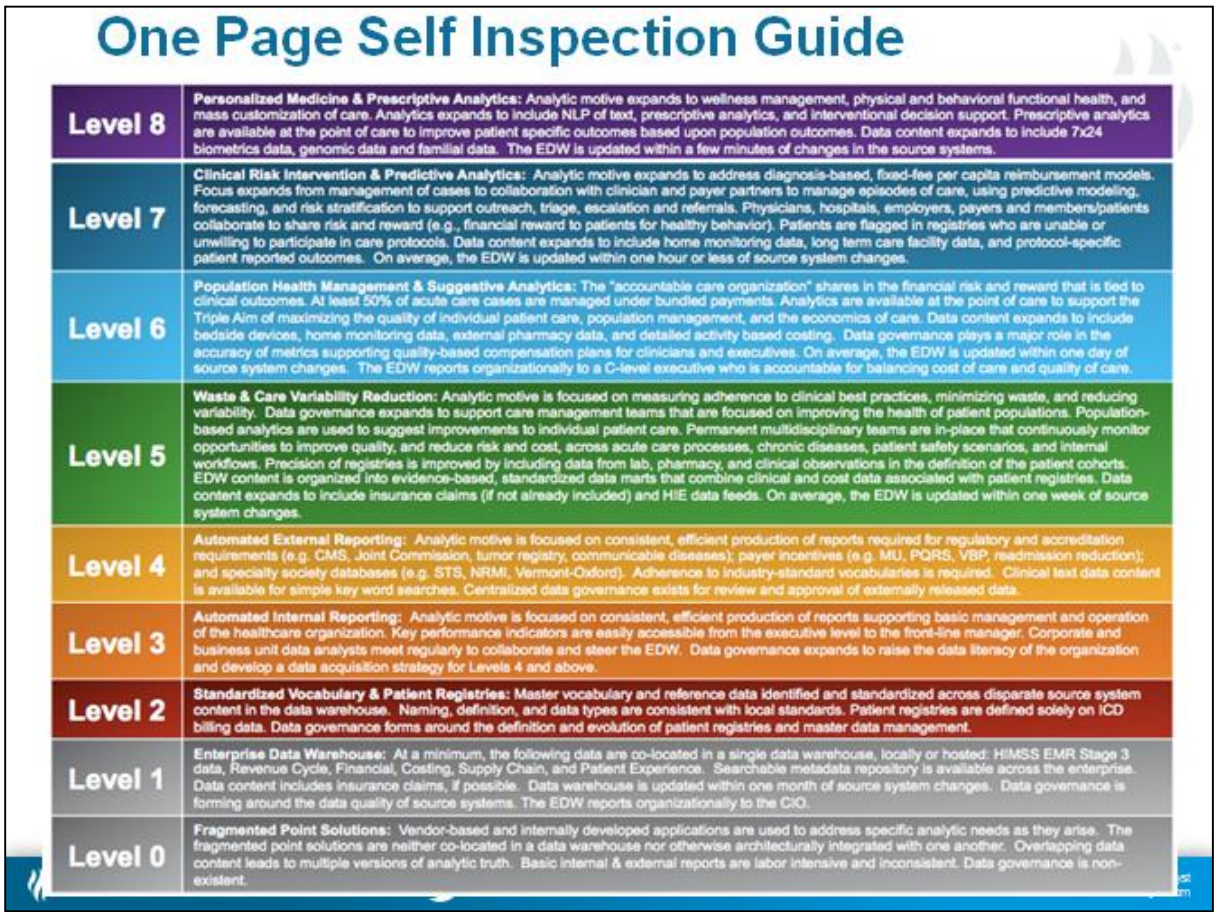
Six Phases of Data Governance

I'll just mention too that in addition to the data governance concepts I've talked about, you go through six other phases of data maturity as a governance body and you have to move through these pretty quickly. So, setting a tone for being cultural and data-driven culture is the first step. The next step for a data governance body is to advocate access to data and provide access to data. Knock down those barriers where people are afraid to share data or they're protective in the organization, this is my data, not yours. The data governance committee has to knock those walls down. You want to assign data stewards, those are the folks out there on the frontlines of data who know their data best and they're going to have a new and important role in this new era of analytics in healthcare and I'm glad for it. They've been underappreciated for many years.

Then once you provided access to data, you provided stewards to help oversee the data, then you want to start focusing on the inevitable and that is the data quality problems that you're going to see as a result of providing this access to data, and there will always be this trough of (41:29) where you start seeing data quality problems you didn't expect, and that's just the nature of the journey, but at least by then you got data stewards in place that can help you work through those problems.

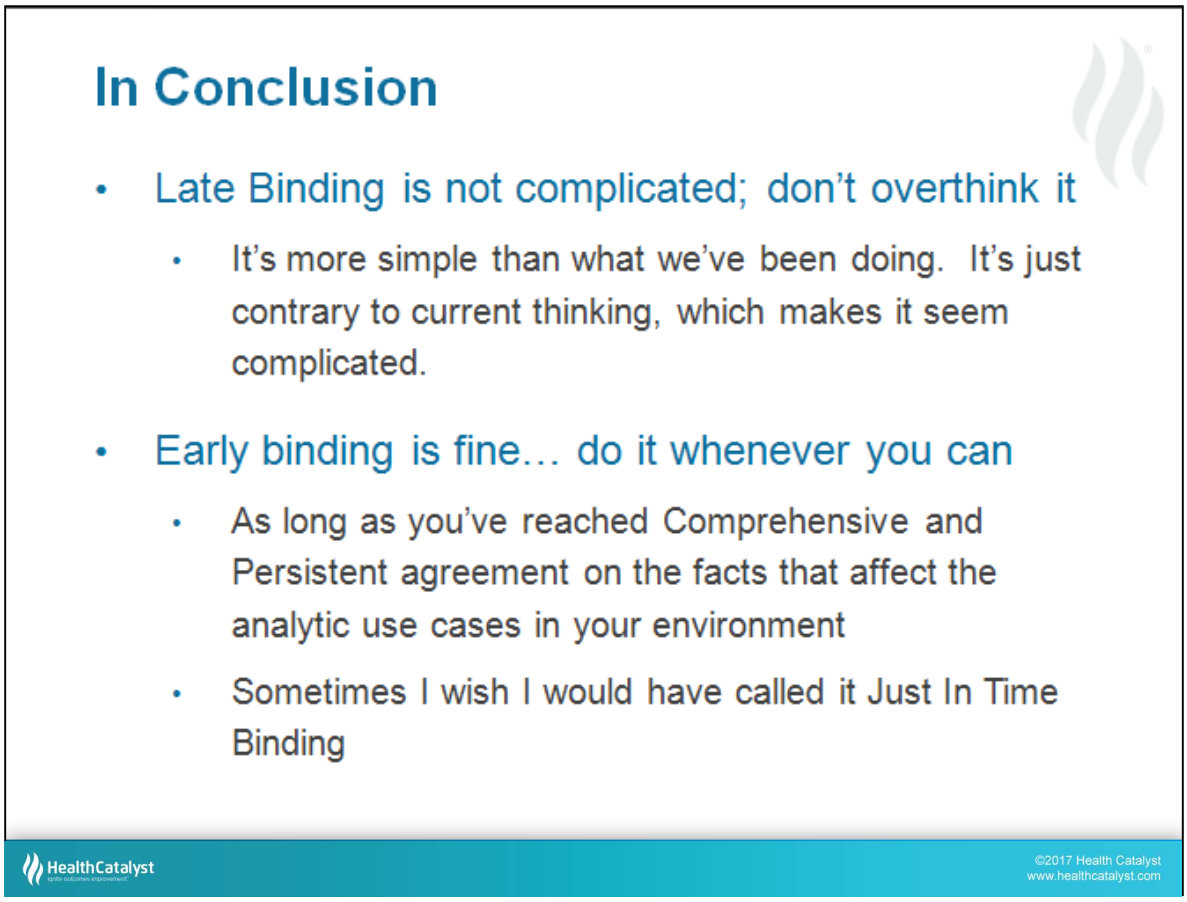
Then you'll go through a phase where you want to maximize the utilization of data. So this is where you really start becoming a data-driven culture when you're driving change back to the point of care, back to your processes to care improvement teams and continuous quality improvement initiatives and that sort of thing.

And then finally, you've got to move your mind towards the acquisition of data and back to that chart on the previous slide that shows all of those different data sources that we need to start acquiring and start planning for right now. In fact, we're too late. We need to get moving on those yesterday.



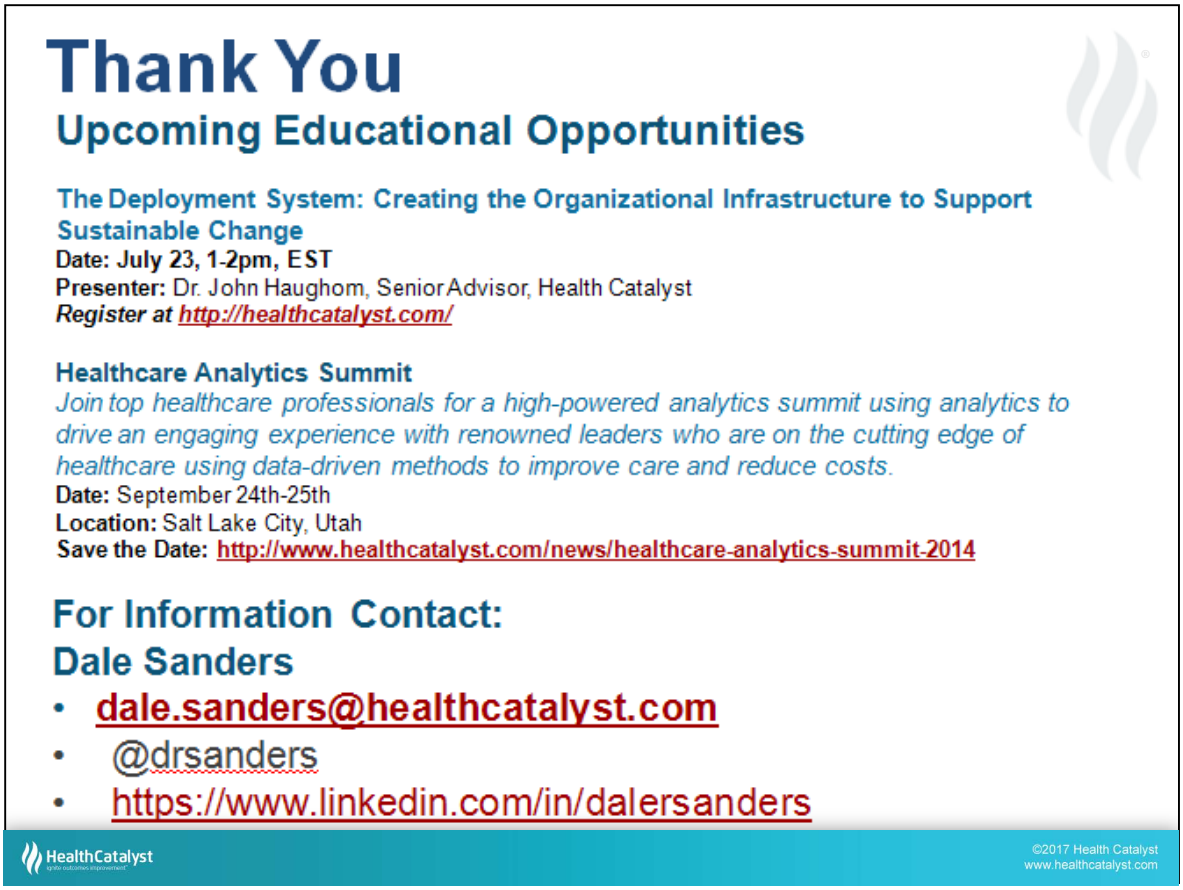
One Page Self Inspection Guide

Okay. So I won't go into this i-chart but the intent here was to provide a one page self-inspection guide so you can take this and go through at the checklist and you can assess yourselves what level are we operating at and at what level do we really have in place and functioning from a data perspective, and there are some cultural aspects in the details as well. So that would be in the slide deck and Tyler will make this available, I'm sure.



In Conclusion

So, in conclusion, I want to emphasize that Late-Binding ™ is not complicated. So don't overthink it. Like I said, I spent my wheels in the Air Force and then with TRW, trying to understand what object oriented programming was and it was really just as simple as I thought it was but I couldn't believe it. Late-Binding ™ is that simple. So don't overthink it. The only thing is it's just contrary to current thinking which makes it seem complicated. I want to emphasize that early binding is fine. So, do it whenever you can. As long as you can reach that comprehensive and persistent agreement on the facts that affect the use cases in your environment, by all means, bind that early and get it off your plate, get it out of the way, achieve some standardized version of the truth in your analytics. There are times when I wish I would call this whole concept Just In Time Binding because you can take Late-Binding ™ a little too seriously sometimes. There is nothing wrong with early binding as long as you achieve comprehensive and persistent agreement on the facts. But the way it worked out, this Late-Binding ™ and Just In Time Binding is really what it's all about.



Thank You
Upcoming Educational Opportunities

Okay. Thanks. I just want to put a plug in here and I will turn this really over to Tyler. Tyler, do you want to talk about these upcoming webinars here, friend?

[Tyler Morgan]
Absolutely. We have, coming up on July 23rd will be the next webinar in Dr. Haughom's Healthcare Transformation series connected to his healthcare ebook that he has recently completed. That will be the deployment system. He spoke about the analytic system previously and he'll be talking about the deployment system on July 23rd. Also, we have our Healthcare Analytic Summit that's coming up here on the 24th and 25th. And so, before we jump in to questions, we do have two passes to give away.

Transforming Healthcare Through Analytics

Before we cover on the passes, Dale has got his contact information up and we do have one more slide. Dale, I believe if advance the slide, just a little bit more information. A few teasers on some of the keynote speakers that we do have for our summit, Billy Beane, Ray Kurzweil, Mike Leavitt, Jim Merlino from Cleveland Clinic and Dr. Steele from Geisinger Health System. We're really excited about this summit. And so, we have these two passes to give away. The first is a pass for a single registration and the second is a pass for a team of three. Now, before doing this, I would like to mention that because of the high demand of limited space, these registration passes that we're giving away today, they must be redeemed through registering for the summit by August 1st when the coupon codes we provide will expire. The approach is really simple. I'm going to put a poll question up here and this is for the single registration pass, if you would like to be entered into the giveaway and are confident that you can attend the Summit on those dates. Now, if you can't see the poll question, please send me a note in the chat if you are in the chat feature, if you would like to be entered into this drawing. We have had a couple of individuals say they haven't been able to see the poll. So, we'll leave this open just for a few more seconds and then we'll go to the poll for the team registration pass.

Alright. We're going to go ahead and close this poll. And now, let's start our poll for the team. Now, this is a registration pass for a team of three. We'll leave this up for a little while and give

a chance to respond to that. I would like to remind everyone as well we've had several questions that – and we will be providing the slides as well as the information around the – excuse me – our poll question, we'll give you the responses to the poll questions as well in our followup email. We'll go ahead and close this poll now. And Dale, let's go right into the questions. Can you see the questions now, Dale?

I hope this has been helpful…

[Dale Sanders]
I can't. Yup. Thank you, Tyler. I appreciate it. So let's go through these. And by the way, I can stay past the top of the hour and I'm happy to do so to get through these questions. Assuming anyone else wants to stay on the line, I will be here.

QUESTIONS	ANSWERS
Does Late-Binding ™ cost data ballooning or performance hit since essentially all the data have to pass through from source to the higher level applications?	Well there might be two answers to this actually. It might look like all the data is passing through to the higher level applications but actually only subsets of the data. It's kind of object oriented in that sense. So, you're only taking thin slices of all of that massive data and passing it through to the higher level applications because you're kind of, you're binding and you're bundling data together to support fairly specific analytic use cases. So, not everything passes through as the diagram might suggest. You actually get better performance from a query perspective from this bundling because you're kind of pre-caching data. It's another form of data caching, really it's at the database level, not at the memory level. So performance actually improves. And actually it's kind of hard to imagine this but it's true that those big monolithic data models have a harder time handling really broad swat queries that aren't supported by those data marts, those subject area data marts, where we've taken that thin slices of data. If you want to run a really broad query across all of those data sources, you can actually do it faster through the Bus than you can through those big monolithic data models. So, this is actually one area where, you know, an object oriented programming and Late-Binding ™ in software engineering, you can get performance hit sometimes in just in Late-Binding ™ in a pure software engineering environment. But so far we don't see that kind of problem in Late-Binding ™ data engineering.

Can Health Catalyst perform both early and Late-Binding ™?	Yes, we can and that's what we advocate, even though we sound like all we do is Late-Binding ™. Like I indicated, early binding is totally appropriate and should be practiced when you reach that comprehensive and persistent agreement state.
What about graph database, NoSQL technologies?	It's a relational heavy approach. I don't know enough about graph databases yet to really answer that question and I see another question of that kind coming up later on. Interestingly, the folks at Intel are asking the same question, the Intel would like to collaborate on this area with us but I don't know enough about graph databases to answer that, so I'm going to have to punt that but I will go out and study and make sure that next time I can answer it.
How is this different from master data management?	Well, master data management is really important because a lot of that Bus in Late-Binding ™ is based upon master data in your data warehouse environment. So, master data, in kind of its simplest form, is really that Bus Architecture layer that I showed in that one diagram. So good question on your part. Master data management is really important. But I would also say that one of the things that I advocate that's very different, a lot of times organizations will try to impose master data management on the inflow of data to a data warehouse and that's what causes that huge delay in all of that time spent and data mapping and compromises and things like that. And what I suggest is you have a master data management area, master reference area, in your data warehouse, and you bind to it as necessary. But you don't bind to it early on because it just takes too long to do that and you may be binding it in ways that aren't necessary or that will change later on. So bind in your master data as necessary.
Can you comment on the use of somatic databases for supporting healthcare analytic architectures? It seems to offer flexibility, global internet readiness, plus much health data seems available in name/ value formats.	Well, let's see… I don't know how to answer this really. I mean kind of I come and go on my opinions about somatic databases. I will say that having been around name value paired data models and environments, the beauty is they're almost infinitely flexible. The ugliness is they're almost infinitely flexible. So, in a practical sense, I have never seen somatic databases and name/ value paired analytic environments work effectively. They're just too flexible. So it might be we're changing that one diagram where I show early to Late-Binding ™ data models. I might put an EAV-type of description in there. They're really Late-Binding ™ and they're almost no binding in some ways. So, I mean I think

	they're cool conceptually and especially when I think around with these in the military, in NSA, we just spent a lot of time with them and in those environments where the data is really complicated and disparate, I think they have some promise but I personally think in healthcare that they're too open-ended. So feel free to refute that and open my mind if you think otherwise. Send me an email or something.
Am I understanding correctly? Bus Architecture means you literally copy the source data from the transactional source to data repository and (54:16) foreign keys?	Yup, that's pretty much it. That's pretty much it. You've captured the essence of it. It's a little different than that but essentially yes. Copy the source data. Don't do much in the way of transforming it. What we really want is just the data. There are some vendors who believe that by copying the source system data, that we're kind of stepping on their intellectual property, that somehow we're leveraging their data model and that there's some value to that and we're stepping on their IP. Well, that's not the truth at all. All we want is the data and their data model just happens to be the container in which it arrives but we don't really care about the data model. What we want is just the data in that. And so, that's why we land at pretty much as it's represented in the transaction systems.
Can you offer us a definition of an Enterprise Data Warehouse as you defined it in this context? What does the EDW look like?	Imagine having the library of congress in every community in the US. That's what an Enterprise Data Warehouse is for you. It's all of the data that you wish you could have at your fingertips readily available, easy to search, easy to find, and you've got informed staff that work around that data warehouse, that manage the uptake and the intake of new books and new periodicals and refreshing it and organizing it and they can help you find the books and the periodicals. Now, imagine that in a data warehouse. Everything that you want to understand about the healthcare delivery that you're responsible for providing is in your backyard and it's staffed by people that can help you take advantage of it. So I hope that helps and again, reach out to me if I can be more clear on that.
I'm enrolled in predictive analytics master's program. I have not seen or heard of this concept within my program as of yet.	Well, now you can go back and share it. That would be great.
Have CRO's or pharma biotech companies used the product for patient recruitment? Seems to be a natural fit to identify populations and subpopulations.	Well, Health Catalyst doesn't have any CRO's or pharma biotech companies but it certainly can be used for that. And when I was at Northwestern in operational role, there is CIO and analytics champion there on the campus. We definitely used it for patient recruitment in the clinical trials. To my knowledge, none of our Health Catalyst clients are using the

	products for that yet but it certainly can and it should be. I would like to work with the CRO's and pharma techs more to help them with this in this area. So eventually I hope that we can do that.
What about the large volume impact on Late-Binding ™?	The large volume impact, I'm not sure I understand completely what you're asking here but Late-Binding ™ is actually well suited for large volume data sets. That's where it really starts to take off as the most efficient way to build the data warehouse. So, the example I always use is imagine if Google tried to build an Enterprise Data Model for the worldwide web and match all of that data into that data model and try to keep up with it. You couldn't do it. There's no way. Imagine if the National Security Agency had an Enterprise Data Model to support every intelligence analytics use case that they might encounter and if they had to map all of the disparate data that all that is dropping equipment takes advantage of, they couldn't possibly build the data model to do that. Now, we're not as complicated in healthcare as Google or the worldwide web or NSA's environment, but we're pretty darn complicated. And so, this is – it's complicated enough to justify Late-Binding ™ and that's Late-Binding ™ really takes off in large volume disparate data environments. That's where it really excels.

QUESTIONS	ANSWERS
Is there a risk of too many metrics and a lack of standardization or single source of truth in Late-Binding ™?	That's a very common question and the answer is there is that risk but that's no reason to use an early binding model. So, what you have here is a very flexible model on a very flexible platform and it's up to you as a culture and a data governance organization to make sure that you use this flexible architecture in a standardized and not overwhelming manner. And that's a problem really with any design truthfully. I think Late-Binding ™ might enable that even more so, but that's really a function of data governance. So you wouldn't want to constrain yourself technically because you don't think you could take care of things culturally. But great question though.
Describe the difference between what we are doing in that early binding and Late-Binding ™ in simple terms.	Well let's see here. Let me go back to the library metaphor for a second. Well, imagine a library that had nothing but encyclopedias as the resource. So someone has gone out and they bought the world's best encyclopedia and that encyclopedia now represents their distillation and their binding of information on all sorts of topics across the world

	from A to Z and they summarize that and they bound that together in an encyclopedia and that's the only thing you have access to. That's better than nothing but I think most, and if you're just learning to do research and you're just learning to read – you know, in the 6th grade, man, I love encyclopedia. But by the time I got to college, I wanted a full up library. And now, at this stage of my life, I want the worldwide web and that's the difference between early and Late-Binding ™ is that we're giving you in a Late-Binding ™ environment all of the granular data in one of the richest libraries of the world, your backyard, and we're letting you decide how you want to bind it together. We're not giving you the distillation of that in an encyclopedia. So I hope that helps kind of describe it in a metaphor that's relatively simple or adjacent.
Does your data warehouse repository (62:25) contain IDs for various dimensions or it contains some metrics, facts, also as they were captured in the transaction systems? Can you show a view of the contents in the repository?	Well, it has dimensions and facts definitely. So we have everything. It's got dimensions and facts galore, as they were captured in the transaction systems. I can see where that diagram might suggest that we're not bringing in the facts but we are very much so bringing it that granular data.
Can you show a view of the contents in the repository?	I can't today but we can and would be happy to do that if you'd like to reach out to us.
	Steve (63:13), dear friend for many years. He's going to ask me a question here on graph data model that I am not going to even try to answer as this much I do know and I agree with Steve, it's schema-less and it's all about relating data. But I don't understand enough about graph data models to offer an opinion on this yet. I'm just now getting my arms around and I think I feel like I'm back in object oriented program where I suspect I'm trying to make a graph data model more complicated than it is because it still kind of eludes me. So, I'll do some studying on that one. Thank you.
What tools are used for the binding itself?	Well, in this environment, in Health Catalyst environment, we have developed a couple of extensions and tools to SQL server. That's kind of the core technology platform that we have and used, and we can bind to any source system with those tools first, that's the first level of binding, and then we have tools within the suite that allow us to bind data to vocabulary and rules. So after we bind to a source system and pull in everything, the granular facts and dimensions there, then we can also use these tools that we've developed to quickly and repeatedly bind additionally within the data warehouse itself. The tools are called source mart designer and subject area

	mart designer are the tools and they're pretty powerful tools, very handy, and that's a big proprietary advantage for Health Catalyst actually.
How does Late-Binding ™ differ from semantic web?	Not much actually. I mean it kind of gets back to the early question about EAB data models a little bit but you know, in a perfect world, we wouldn't have to aggregate all of this data in a Late-Binding ™ physical data warehouse in healthcare. We'll be able to utilize the concepts of the semantic web and leave all the data resident in the source systems. That's what Google does for the most part. The only thing that Google consolidates are indices. They actually don't consolidate all of the data in the worldwide web, that would be impossible. What they consolidate are indices and the semantics associated with the content in the web and then you can reach out and grab that data in its source system. Now, I'm cautious to warn everyone that those concepts are a long way away in healthcare. Just the nature of the way we've organized data in a very non-standardized way and the nature of the analytics that we perform are dramatically different than indexing and mapping data to semantics, which is what Google does. So when we bind data together, for example, to define what a diabetic patient does. That's an algorithm and that's very different than a semantic concept. I mean you might be able to implement the definition of a diabetic patient someday as a semantic concept but right now that's not possible and it's certainly not possible to do that in a virtual sense. To some degree, that's what I I2B2 tried to do. So, it's an interesting concept but I don't know in my lifetime that we'll ever reach the semantic web concept across healthcare. Now, we might be able to do it within the boundaries of a data warehouse once we pull the data in and we put some kind of constraints around it and binding around it. So maybe that's within our grasp.
We have a Meditech data repository warehouse, will this work with current system or do we move the data over to a new platform?	The last time I saw the Meditech data repository was when I was a CIO in the Cayman and I was recomputing the Cerner contract and I looked at Meditech's tools around analytics and it was a fairly early binding, a pretty traditional data model that would support reasonable use cases up to level 3 and 4 in that Analytic Adoption Model but I didn't feel like at that time that it would support the aspirations that we had as an organization above that. So, I think my answer to that is that the current design that I understand that the Meditech data repository is early binding kind of a traditional data warehouse design. You might be able to continue utilizing that to achieve

	your level 3 and level 4 analytics but I think you're going to get up above level 4 or you're probably going to have to go to a Late-Binding ™ Architecture. And so, if I were your CIO, that's the direction I would be going.
What is Health Catalyst philosophy process tool sets for the management of what I'll call disparate business rule rationalization on the journey toward comprehensive and permanent persistent agreement and the reconciliations that might need to be automated between two or more reports where consensus can't be achieved in a short term but the reports must continue to inform.	Wow, that's a good question. I can't really say that we have some supernatural great tool for the management of the business rules. They emerge or they're manifested in the development of the data marts, those subject areas that are kind of specific. So, the spin-off, those object oriented analytic data marts that are kind of subject specific encapsulate the rules associated with that use case, and we expose that in our metadata repository. So for example, if you have a need for two definitions of what constitutes a readmission, a Late-Binding ™ Health Catalyst Architecture supports that. We recognize that. It's an important need. You have to allow for that in healthcare because, as you indicate, consensus can't always be achieved and there may be reasons for many years to have multiple definitions of the truth, depending on the context. So, you can have multiple definitions in this architecture. Now, have we done a great job developing some kind of rules or management tool? I can't really say that. But we exposed it in our metadata repository. So you can browse that and you can see it and you can expose it, but I can't think of anything that we've done that would make it easy to manage those as separate configurations of the same concept. So, interesting thing to think about and good for you for thinking of it because that's probably something that we need to think about more in the future. And again, software engineering has addressed this with the tools to support, those logic rules that are manageable across the enterprise.
You mentioned some guidance on Hadoop and I understand you're looking at Health Catalyst Hadoop. Could you comment?	Yeah. So the quick comment on that is that we're playing with Hadoop. There aren't any compelling reasons or use cases for the use of Hadoop right now in healthcare is my argument. I mean it is compelling in that it's essentially free. Nothing is free but it's this close to free from a licensing perspective. And so, if I'm a budget-pinching CIO, as I usually am, I definitely would be looking at Hadoop for that reason. But the reality is there are hidden costs associated with Hadoop right now that everyone needs to be aware of, and that is, for one thing, there just aren't very many skills that can take advantage of Hadoop, and

	especially the tools that you have to lay on top of Hadoop to analyze data. So we're looking at it, we're prototyping it. We don't see any compelling need or use case for it yet in healthcare. We're very pragmatic about what we do. All of us started our careers in, you know, operationally in your shoes. And certainly if there was high value to it, we would be the first ones to embrace it. But it's just not compelling right now. I think it will become more compelling as our skills catch up to it, and as the real time nature and the volume of data in healthcare starts to really take off. I think relational database engines will be less and less capable. But you know, the SQL server platform, we love Microsoft, we have a great strategic partnership with them. They're working very closely with us on the Hadoop platform and how to migrate our clients to it if and when it's needed, and we will be the first ones to get there when there's a compelling reason to do so. But they're just not right now.
Has the development of IBM's Watson and its beginning applications in other industries such as healthcare altered existing ways of thinking? Also, is Watson a totally different type of system or something that runs on top of the layers previously mentioned?	Watson is almost a completely different kind of system. We could certainly feed Watson with the data. So if you wanted to lay Watson on top of that Bus Architecture, you could definitely do that. I had the opportunity to become fairly familiar with Watson over the last couple of years through some relationships. And frankly, it's one of the most exciting things in my career as an information technology professional going back to 1983 where I kept my teeth in the Air Force and NSA and I mean I was really at the cutting edge of the most expensive and forward-thinking computer science and Watson is one of the coolest things that I've ever seen. The problem with Watson is there is no learning set equivalent to what they learned and trained Watson on in jeopardy. And so, unfortunately, the executives that know just a little bit about technology became enamored with the bright shiny things that Watson could do on jeopardy, and immediately as soon as you could apply that to healthcare. That's not the case because there is no equivalent learning set in healthcare. In particular, we don't have any outcomes data. We have patient satisfaction data but we don't have any outcomes data. And if I had a diagram, I would show you this that I used to describe the importance of a closed loop data system. When you're trying to train predictive algorithms and systems like Watson, you have to have a closed loop environment to train those. And we don't. That's why I highlight this giant need that we have to collect outcomes data

	because we don't know what we're training Watson to do. And the other side of it is Watson consumed virtually every text source of data from Wikipedia on down to learn about certain current culture in the world that supported the kinds of questions that they were trying to answer on jeopardy. There's no equivalent text-based data source of that volume in healthcare. So if you consume, for example, even all the digitized journals that are out there, it's a tiny fraction of what was consumed to support jeopardy. I mean tinny tiny. If you consumed every healthcare journal that was ever written, it is still a tinny tiny fraction of what was consumed and indexed and analyzed for jeopardy. And oh by the way, most everything that's produced in those journals doesn't really apply to real healthcare in an organization like (76:34), where so much of what is driven towards outcomes is a consequence of the environment and the lifestyle in this community. So depending on which study you believe, only about 15% of clinical trials ever actually apply to real healthcare. So if you're feeding Watson strictly from clinical trials-type data and journals and things like that, you're probably training it on data that doesn't really apply anyway. Now, it all sounds like I'm sort of critical of Watson, which I'm not. I think it's one of the coolest things ever. It's the fault of healthcare, not Watson, that we can't train it better. So we need to catch up our data ecosystem to take advantage of the technology actually. So I hope Watson doesn't dump things out.
Are there particular reporting tools that you prefer for levels 3 and 4 and for past higher levels such as Tableau, PowerPivot itself?.	Yeah. So, the tool that we picked at Health Catalyst is QlikView and we've done very thorough analysis of the tools that we think are best for the industry and for healthcare in the mission that we're serving. That said, QlikView is pretty expensive but it's very very powerful. It is we think the best overall value of the tools that are out there. But if I didn't have the budget, if I were your CIO and I didn't have the budget to afford a QlikView implementation, I would probably take advantage of the tools that come bundled with SQL server, like PowerPivot. So, Microsoft, you know, God Bless, and they don't always do great right out of the shoes with products but they do pretty good over time. And the analytic tools that they're starting to bundle for free with SQL server are getting better and better all the time and if I didn't have any other budget, but I still had an analytic need, I would take maximum advantage of those tools. Tableau is a very

	good product. We have seen some and we've tested it thoroughly in different environments. It hasn't worked out for us but it is a good tool. Spotfire is another that we think very highly of. But again, QlikView is our choice. We think it's the best out there. But one of the nice things about this Late-Binding ™ design is that you can feed data from that platform of granular data content into almost any tool that's out there. And I'm quick to remind everyone that the overwhelming, most popular analytics tool in the world remains Microsoft Excel. So if you don't have anything else, you can go a long way using nothing but Microsoft Excel and I encourage you to do that. Don't overcomplicate things. I've seen a lot of value fold out at Microsoft Excel and being able to feed data to people who have been starved for data in healthcare for a long time and using nothing else but Microsoft Excel.
With Late-Binding ™ in a Data Warehouse, does that mean the data warehouse contains a copy of the all source systems? Would that make the data stale? Would a data federation tool make sense?	Well yeah, you bring up a good point and that is you don't want the data content in your data warehouse to become stale. So you have to update it on a very regular basis. Now, there are some people who will talk about a real time data warehouse and all that and there are some data sources where collecting data in real time does make sense. And we can do that with the Late-Binding ™ Architecture. That's one of the nice things about it. You can pump data into a Late-Binding ™ Architecture in real time because you're not remodeling it. You're just dumping it in pretty much. And that's the good news. But, you know, what I see a lot of times is people falling in love with that notion of a real time data warehouse but they don't even have the ability to consume it. And they certainly don't have the ability to drive change in the organization at that rate. In fact, if you drive real time data into an organization that can't consume it, it ends up just confusing the situation and making their process improvement initiatives worst. So, this whole timeliness of data has to be delivered in the same context and in the same rhythm as the organization's ability to absorb it and actually act upon that data. Otherwise, it's just a waste of time and resources to do that. I wish that there were data federation tools that would work in healthcare. There aren't. I don't expect that to happen for a while. Again, I2B2 is an attempt of that and it's better than nothing but it's not good enough for what we need to do. We still have to unfortunately bring data together physically.

How does this work with multiple disparate systems containing a lot of the same demographic information. Is there a tool layer into the product for data cleansing?	Well kind of two answers there. You definitely have to reconcile and bind to a common patient identifier in this architecture. So that's one of the things you need to do in an early binding fashion right away. There are two core dimensions of analysis that constitute healthcare. One is the identification of the patient. The other is the identification of the healthcare provider or coach or physician, nurse, whatever. That's what constitutes a healthcare encounter right now are those two pieces of information. And so, you have to bind to and reconcile that patient identity and that provider identity right upfront very early on. And so, there is a tool and in that Bus layer that I depicted in that one diagram, there is an MPI, you know, patient identity, provider identity matching layer that goes on in there. Data cleansing is not something we advocate unless that's what you call it, data cleansing. We don't do a lot of – I don't advocate data cleansing in the data warehouse. I advocate letting all the ugliness and the dirtiness of the data show up in the data warehouse and then feeding that information back to the source systems and letting them clean this up, because you'll spend yourself right into the ground if you try to keep up with cleaning the data that is really the source system's problem.
I love your library of congress example, very helpful. So we need that in place prior to moving forward with the Late-Binding ™ approach?	Kind of. Yeah, you need the kind of the basic building block is that granular data from all of the source systems in your environment for delivering care to your patients and running your organization. So you do need that sort of your version of the library of congress in your context in your data warehouse in order to take advantage of these concepts.
How do I translate all these disparate data? What makes it interoperable? API? Something else?	Well fundamentally what makes it interoperable s that layer in the Bus where you're first imposing that those common dimensions, those common foreign keys that allow you to relate across those disparate source systems. So it's not an API really. You could describe it as a data engineering API but it's not an API in a traditional sense. So that's basically – again, keep it. Don't overcomplicate this. That Bus level, what it really comes down to are foreign keys that are common data types and common naming conventions. It's really simple.
Healthcare vendor products tend to generate a massive amount of audit level of detail of all changes	Great question and very good insight. Yeah, and in fact, I've used log data, not a data, from an EMR to

and corrections. With your definition of EDW being a consolidated source system data, is there any analytics value that could bring in this data?	produce what was probably the first meaningful use dashboard in the industry. So we were trying to understand how our physicians and nurses were using Cerner and Epic. And so, we went into the audit log files and pulled out all those patterns to look at what was being used by who and it revealed pretty quickly that our EMRs were not being used to their fullest extent, and we push that dashboard back to our physician leadership and we said, you know, you've invested tens and millions of hundreds of millions of dollars in these EMRs and they're being utilized for a little more than a transcription system. So it's up to you. If you want these tools to be more useful for healthcare, you're going to have to change the way they're being used because the audit logs and everything else are showing significant underutilization of the function. And it literally changed the entire culture of how those EMRs were being used. So yeah, great insight.
As I understand of the team, these journal reports and any respectable healthcare sources into Watson is limited to what has been published. It might not be too current.	Yeah. Yeah. Yup. I think that's pretty much my understanding too. And again, it's been a little while but I had a pretty in-depth insight to Watson there for a while and I'd like to catch up on it again actually and see if they have made any other things.
What if the health system is finding to bring on their ADT, as well as other EMRs that would have different identity? How can that happen?	Well in that case you want to bring the data in from the different ADT and different EMRs and you want to land it in that lower level of the Bus down in that source mart layer and I could even go back to that if it would be helpful. Listen right here. Let's pull that up. So, you want to land that in this environment, land all of those disparate EMRs and different systems down here pretty much as they reside in the source system. So where you see EMR, just add an S to that for plural, multiple EMRs. You're going to land that in there. Then, if a patient is showing up in multiple EMRs and there's not a common identifier, then you have to resolve that with an MPI identity matching tool in this layer. So that's where the MPI comes into play, in that layer, and that's where, again, all you're doing is adding a common foreign key that now allows you to query for Dale Sanders wherever I might show up in these source systems. So it ends up looking just like a virtual enterprise data model but you haven't had to go through the pain of that.
Regarding interoperability and simple foreign keys using a common language at the Bus level, you must surely know that there is no such common language used across data collection systems in healthcare.	Well, I kind of do but I have definitely been in the trenches of healthcare data for a long time. And as I mentioned earlier, 80% to 90% of our analytic use cases of high value right now actually revolve around a

	pretty standardized vocabulary and it's all those things that I listed in that Venn diagram. So you're right in that ICD10 being a classic example of that, like how much larger the vocabulary of healthcare is now under ICD10 than it was under ICD9. SNOMED is another good example. But the important thing is on the source system side, they need to start using those bindings to those vocabularies. And so, for the most part, ICD, CPT, those kinds of things, they're fairly standardized across all source systems. What's not standardized are things like LOINC, RxNorm, SNOMED, and eventually ICD10 will help us. But, again, that's where you use tools to do that kind of binding to those vocabularies and resolve that in this layer here of the Bus, but you do it as needed. You don't try to do it all here at this level because you'll never get out of the mapping process because of what you just said. So you bind the vocabulary as you need it. You don't try to do it at this level. They'll never get out of the mapping process and I've seen some of the smartest academic medical centers in the US get lost in doing that, binding the vocabularies at this level, and they never actually get to the level of using the data and taking advantage of any of those data.

[Dale Sanders]
Okay everyone. Wow, I'm done. I'm tired and I appreciate everyone staying around. It's been a lot of fun and I always learn from these and I'm going to go out and read up on graph databases right away, that's for sure. Feel free to reach out and offer suggestions about how to make these webinars more informative. Thanks everyone.

Tyler, back to you.

[Tyler Morgan]
Alright. Thank you, Dale. So before we close the webinar, we do have one last poll question. How interested are you in the demonstration of Health Catalyst solutions? We'll just leave this open for a few moments to give you a chance to answer this. And while you're doing that, I would like to remind you that shortly after this webinar, you will receive an email with links to the recording of this webinar, the presentation slides, the poll question results, and also the names of the winners of the summit registration pass giveaways.

We'll go ahead and close the poll now. And on behalf of Dale Sanders, as well as the folks at Health Catalyst, thank you so much for joining us today. This webinar is now concluded.

[END OF TRANSCRIPT]