RECORD, Volume 27, No. 2*

Toronto Spring Meeting
June 20–22, 2001

Session 47IF
Data Quality

Track: Computer Science

Moderator: LINDA M. LANKOWSKI
Panelists: GARY B. BRONSTEIN
LINDA M. LANKOWSKI

Summary: The quality of data is paramount to the success of today's data-driven enterprises. The insurance industry is no exception. Insurers use a variety of resources and techniques to validate data. Further, insurers are faced with short development cycles and partial product specifications. Speakers discuss how to develop alternative or improved processing methodologies to reduce errors, correct error tracking tools, and metrics. Participants discuss balancing reasonable testing given time constraints.

MS. LINDA M. LANKOWSKI: Gary Bronstein and I are the two presenters for this session. Gary is currently at Allstate Financial. He's been there for 10 years, and he works in the data quality area in life finance. Prior to that, he was in the valuation area covering all deferred and payout annuities. Before Allstate he worked at Watson Wyatt and TransAmerica. I have been a life generalist. I'm currently at Merrill Lynch. Prior to that, I was with John Hancock and Mass Mutual. And way back when, I was with Buck Consultants doing pension consulting.

Our backgrounds are significantly different. Gary is going to talk about data quality from an end user standpoint. I have been doing quality assurance in the systems end at Merrill Lynch for the past four-and a half years, so we come from completely different viewpoints.
Because I'm with the systems end of things, I want to start with some definitions. Testing for quality—there are four levels of testing that we use at Merrill Lynch.

Types of Testing
- Unit Test
- Systems Test
- Regression Test
- Acceptance Test

The unit test is the programmer test. A systems test starts once it has gone

through the unit test, and you've got a couple of examples that work. You put everything together and see if it still works. Once everything is integrated, the regression test says, "We've got old policies on the same system. Let's make sure that the old policies work once we've pigeonholed the new policies into the system." The acceptance test is generally run once the systems area has signed off. That is when you get all of the clients to actually sit down and work with it and make sure that it's okay. Part of the acceptance test is reviewing output, making sure that timing is not a problem. For instance, if a data entry screen takes 10 seconds to actually apply something after you hit enter, that's too long. They go back and redesign. There are a lot of different levels of testing. If you leave out one of these parts, you've got a problem. I cannot emphasize that enough.

MR. JEFFREY M. ROBINSON: Where would the parallels run? Which one would that be, a regression test?

MS. LANKOWSKI: A parallel run would be a type of regression test.

Roles in Testing

- Client
- Programmer
- Analyst
- Project Manager

For testing roles, you should have a client, a programmer, an analyst, and a project manager. You may have one person performing multiple roles. In fact, if you have a small project that's only going to take a week or a couple of weeks, you are probably are going to have an analyst work as the client, or the programmer will be the analyst.

However, you have to recognize that these are different roles. If you lose one of these roles, you're going to have a problem. If the client says, "I want a new screen," and then he walks away and never looks at it again, the analyst is going to say, "Good, I can design a new screen." The programmer says, "I can interpret this." And then the client comes back later and says, "This isn't at all what I wanted." The same thing happens when you're talking about using the data as an actuary. You say, "Okay, I want this kind of a report. I want premiums, I want in-force business," and you wind up with a report that looks completely different than what you were expecting. It's not usable to you. You have to have the client involved with it right from the beginning. If you are the client, you need to make sure that you are involved in this testing. The analyst is the one who, generally, works in between the client and the programmer,helps the client to design exactly what it is that they want, and explains to the client that something can or cannot be done. Finally, of course, the project manager is somebody who makes sure that everything actually happens on time, on schedule, keeps things on track and moving.

Specifications

- Business Spec
- Technical Spec
- Actuarial Spec

Specifications are extremely important. You don't always have full specs when you start a project, but you should have these things. We have templates that we use at Merrill Lynch. Every company is going to have its own template, but you need to become familiar with what template is being used if you are involved in the data quality end of it. If you don't know what the business spec says, or you don't know what the technical spec is, your actuarial spec may or may not fit in with what is actually happening.

A business spec is, generally, what the end-user wants to get out of a project. In a business spec you might have something that says, "We're going to have a new product coming out. It's going to have screens that are useful to the client services area, to underwriting, claims, anything that your programmers will have access to." Accounting gets involved with those kinds of things. All of this shows up in a business spec.

The technical spec then has very specific things that are involved, so that you can do the testing and make sure that the quality is there. As an actuary I think you should be reading these things. I know that when I worked at Mass Mutual, systems and actuarial didn't talk to each other any more than they absolutely had to. And, frankly, if data was wrong, we all complained and blamed everybody else, but none of us actually sat down and talked with the systems group to say, "This isn't really what I wanted." There are these documents out there and as the end user you should probably make yourself familiar with them.

You should be reading a technical spec to find out: Are they doing the accounting the way I think the accounting ought to be, or am I going to have to take this data and manipulate it once it goes into general ledger? There are a lot of things that are there that you need to be aware of. As for an actuarial spec, there should be a template that you use. We don't happen to have a template for actuarial specs. It depends on what it is that we're doing.

MR. ROBINSON: Where are the results predicted?

MS. LANKOWSKI: Actually, that's not part of a spec. It can be part of your actuarial specs and, usually, that's what you put together for unit test purposes. We don't usually put them into our specs, per se. The specs are the details of what it is that you want to have done. For the business spec, it might include copies of what the screen layout will be in the end, or it will have lists in the technical spec of which accounting fields are going to be used, what general ledger inputs are going to be, and where this is going to be used to feed other systems. That should be in the technical specs.

Who is Responsible?

- Data Entry Clerks
- Data Manipulators (Programmer/Analyst)
- Report Writers
- End Users
- Auditors
- Vendors

Data quality gets pushed around to different people for ideas. Certainly, your data-entry clerks have the responsibility for entering data just as it shows up. If they have an application, they enter the data. If they have an invoice, they enter the data. Unless you're using a very good scanning technology, you've got human intervention. You've probably got quality checks in place, but that's not really where most of the problems come in with data quality.

You also have people who are data manipulators. Your programmers or the analysts, these people take all the data from one spot. There's some program behind the scenes that says combine this for some other field that you're using. This needs to go into premium, but premium is revenue and there are all kinds of things that go on with that. Your programmers often have absolutely no idea what the business end of this is supposed to look like. If you have a programmer that doesn't know that, you've got built-in problems.

You have people who are writing reports and may not be the same programmers. It's somebody who can use the database, manipulate it and say, "Okay, this is what the data is, this is what I want." Sometimes that's somebody with a lot of experience. More often we hire an actuarial student right out of college and say, “Go out to this database and play with it and find the data for me.” And so your reports may or may not have what you really want in them.

Most of us in this room are end users of the data. We take that data, do our own analysis of it, and present a report. Our auditors are, in some respects, responsible for data quality. They go through and do some spot checking. Sometimes auditors do major amounts of checking. They are reviewing it, but it's not really their responsibility to make sure that it gets right.

A lot of us seem to be working these days with vendor products. We have a number of administration systems out there that are created by vendors. Right now I'm in the process of installing a reinsurance administration system. Our vendor gave us a product and said, "Here you go - it's been through quality assurance testing." We looked at it and none of the numbers were right. It was the most amazing thing. You've got to realize that what the vendor says is right you cannot take for granted, and you should be testing this on a regular basis. When they have sent us updates they have not always been correct, or they might break something else. You need to do your own testing; don't rely on the vendor for data quality.

Definitions/Terminology

- Premium
- Benefits/Claims
- Expenses
- Revenue
- Other

One of the questions that comes up quite often is: Just what is a premium? This leads to a lot of data problems, because premium from the sales force is: How much did I sell this week? To a sales agent a sale is, "I got the client to actually sign on the dotted line." Whether we got any cash for it or not, that's a sale. From an actuarial standpoint that's not a sale until it's an in-force policy. You have to be very careful about looking at sales versus incurred and versus accrued. The same thing for benefits and claims. Are we talking about incurred? Are we talking about paid amounts? Expenses. We all know that there are multiple ways of looking at expenses and how we hold this liability. These are data issues that we all should be aware of and paying attention to.

There are other definitions. When I'm dealing with reinsurance, we talk about cessions. "Cessions" is spelled with a C. If you talk to a programmer, "sessions" has three S's in it. They're talking about a Rumba session, a TSO session, and a mainframe session. These are completely different things. When I talk to my programmers, I have to be very careful about what it is that I'm saying. You should all be involved with that, too.

Project Issues

- Built-in Redundancy
- Flexibility
- Scope Creep
- Documentation
- Degree of Accuracy

From a project standpoint, there are a lot of reasons that data quality is a problem. Built-in redundancies may or may not actually help the problem. You might have a screen that won't allow negative numbers to be input. Well, sometimes a negative number really is what you want. And your batch system might do something completely different than your on-lines, and have a different error tracking going on. We try to put flexibility into our systems, but most often instead of building a flexible system, we're trying to shoehorn a new product onto an old system. It was flexible at the time, but 1970 was a long time ago and things have changed. Products have changed significantly and so we do have data-quality issues that come from that. Scope creep happens on any project that you work on and when scope creep happens something has to be cut. Quite often it's the Quality Assurance, (QA), process and so you don't have the testing going on. Documentation is always a problem. Who likes to document?

The degree of accuracy is also a concern. We work with illustrations. Our systems are variable universal life systems. We look for real precision. Vendors think, “Oh, it’s only off by $1.” Well, the first year it’s off by a $1. By the tenth year it’s off by $100. For long-term contracts this is a problem. You have to look ahead in time to figure out how much accuracy is necessary and how much is not.

Other Issues

- Timing
- Transactions between Affiliated/Subsidiary Companies

There are other issues that come up. Timing is, of course, always an issue.. But your data might be off because of transactions between affiliated companies. And this is something if you work for a big company you don’t care about, but maybe you’re moving money from one subsidiary to another. Perhaps you have an offshore company that needs to move money through the U.S. company and back to the offshore company. You’re going to have huge amounts of swings going on from one week to the next that you have to account for somehow. So you need to understand what the impact is on transactions between affiliated companies. Are you double counting your business? Are you not counting enough?

We try to speed up development by starting with incomplete specs. We try to add flexibility and shared code. Each of these has problems. If you have an incomplete spec, your programmer has no idea how to put the system together. It might be okay if the only part that’s missing is something that you think is going to be a fairly easy thing, say a scalar amount, some percentage that is going to always be fixed, or you might want to put it into a table. Put in flexibility for it, and sometimes you’ve got the flexibility, sometimes you don’t. Flexibility takes time, flexibility takes effort, quality assurance effort as well as programming effort, and you may or may not have the luxury of having time for flexibility.

Speed Up Development

- Incomplete Specs
- Flexibility
- Shared Code

And, finally, we will discuss the topic of speeding up development. I know at Merrill Lynch we've tried to use shared code. Unfortunately, this has not worked out the way we had expected it to, because our shared code is extremely small modules. Even though we are using shared code for our illustration and administration system, it's not what I think of as what should be shared code. I think that the shared part of it should be something more like a premium function and instead it's a small function within premium to determine whether it hits tax testing. You've got to be careful about what you think is shared code and what your programmers think is shared code and how much can go back and forth.

MS. SUSAN M. LEE: Earlier you discussed running the applications and the need to test them in-house. We often have an issue with that where a lot of it goes outside

of the actuarial realm. "If we purchase the software, what's the need to test?" How do you have that dialogue with management about those issues? I don't know if there are any vendors here or not, but what about the need for them to test in-house for the way that you use that tool? I've spent a great deal of time in that and management is pushing back saying, "No, don't do as much of that testing, we need your resources over here." How do you have that dialogue with them to explain to them the need to test when they feel they've purchased this application, so isn't it perfect, doesn't it do what we want?

MS. LANKOWSKI: That would be nice. I would love to have a system that was perfect and did what we need. It's a tough thing to tell management, because management says, "You just spent $100,000 on a vendor system, why in the world didn't we get our money's worth?" And that could be a problem.

MR. ROBINSON: As somebody who has sold software, every company does things differently and no system can really reflect all that. If they did, then it would be too heavy. You've got to assess to reflect your particular conditions, and if you don't, you're always in trouble. That's what you have to tell management, that things are different here, but we have specific purposes we need it for, and we have to see whether the conditions we have are met by the system.

MS. LANKOWSKI: Right. One of the things that you could say to your management is that this is different. Our product is different from the others on the market. If you're a member of the Academy, you are bound by the Standard of Practice on Data Quality. There are other standards that reference this Standard. You can always tell your manager that he's got to sign off on this and it's his responsibility for data.

These days, we talk about liability so often and people get sued over the smallest things. The fact that you have a reliance statement doesn't necessarily mean that you're going to be able to do that. If you don't test, you can tell your management that they need to pay the vendor for a reliance statement, and then the vendor will have to provide some statement of reliability. It's going to be very difficult to get a vendor to actually do that, because the vendors all claim that the data that's going in is where the problem is rather than the data that's coming out. It's really a tough situation, but I would suggest that you tell your boss that he has to get a statement from your vendor and that you can't do that.

FROM THE FLOOR: Also, the definition varies.

MS. LANKOWSKI: Absolutely.

MR. ROBINSON: And that's very true and then some of the differences are probably due to the terminology.

MS. LANKOWSKI: When I went to Merrill, I never realized that there was a difference between issuing a policy and actually putting it in force. And so we issue with amendment, but that does not mean that a policy is actually in force. We've

sent out a new amendment to the policy, and until that gets signed and sent back to us, we don't put the policy in force. So it's just a difference in definitions of what is going on and those definitions are bound with problems.

MS. LEE: What's the normal process in the instance that somebody is personally responsible and it's their responsibility to test the data? What if it isn't the data going in? What if it's the calculations themselves within the vendor's application? What recourse do you have?

MS. LANKOWSKI: If you are testing a vendor product and it's a new vendor product or they've sent you an update, the only recourse you have is to go back to the vendor and say it doesn't work. Now if the bill comes to you and it hasn't worked yet, then you can always tell the vendor, "Look, I've got this $25,000 bill, I'm not going to pay you." It doesn't always work. We are in the process where a vendor has created a system for us. We've paid the initial fee. We have not paid any of the additional consulting costs for getting this thing up and running. It doesn't matter to them. They're going fairly slowly to get us up and running. It's a tough thing to do and there's not a lot that you can do other than your own testing.

FROM THE FLOOR: Check with your vendor.

MS. LANKOWSKI: Exactly, right.

FROM THE FLOOR: Microsoft.

MS. LANKOWSKI: Now there's somebody that I can argue with.

MR. ROBINSON: You should hold back a significant amount of money until you finish acceptance testing.

MS. LANKOWSKI: Yes. You should not pay the bill until you have tested. With most vendors that's the only thing that you can hold over their head. The problem happens with illustration systems; it happens on the annuity side; it happens on the life side; it happens on the health side. If you have a software system that ties in directly to your doctors, if you're a managed-care company, this will happen, too. And your doctors will enter things wrong. One of the things you do with systems testing is you push all the wrong buttons, because somebody is bound to get tired and hit F10 14 times. And, suddenly, you've got 14 different claims coming through, or premiums, or changes to date of birth, or something.

MR. GARY BRONSTEIN: As Linda had mentioned, I'm looking at the administration system from the viewpoint of being the end user. Obviously, valuation is what I've been doing for quite a number of years before I went to data quality. Annuity is my specialty. Even though I may be talking more about the annuity side, I think everything fits on the life side as well. Actually, as I went through some of the causes of the breakdown of data quality, I talked to some of my colleagues on the life side, and I asked if these were the same issues encountered on the life side, and they said, "without a doubt." Whether you're an

annuity person, a life person, or a combination of both, I think you'll find that this impacts both.
Here are some of the causes of breakdown in data quality. One is a proliferation of new products. I know from Allstate it seems like we put out 10 new products a month, which is ridiculous. Obviously, systems cannot keep up with all the product launches, so what ends up happening is you don't get all of the features of the product on the system, and we have a lot of what we call post-launch items which sometimes, as you'll see further in my presentation, usually are forgotten. Then when something happens when it's being exercised, you find out the calculation is being done incorrectly.

Programmers lack the product knowledge. A lot of programmers, are told, "This is what we need to program." They don't really know the product, the product features, or the impact it has on other people. What they're being told to program is what they do, and you find that that causes a breakdown.

The administration system's lack of checks and balances—we're putting together a marketing data warehouse, a huge data warehouse of all our products we sell on the life side, and included in that was some information. The person who we hired to do this made a presentation. He says it was interesting that on the annuity side there were a large number, hundreds of people, who were age 99. He was questioning that, and it turned out that there was a default in the system. The programmer said, "Anyone who doesn't have a date of birth may get 1/1/1901," which, obviously, makes no sense. I mean, from a deferred annuity side is mortality/age important? Well, yes, it is. It's become that way, especially with minimum guaranteed death benefits (MGDB) and so on. It seems like he asked, "Isn't that part of data collection, getting the date of birth?" He talked to administration and, well, sometimes we don't get it, and then we put it on after the fact. Well, sometimes after the fact never happens.

Training of administrators—I could tell you about one vendor's administration system that, in a matter of six months as a valuation actuary, I knew more about the administration system than the administrators did, and I could explain what went wrong. They were calling me about how to input something in the administration system. There's also high turnover in the administration areas, and that makes it even more difficult. You train somebody and they get really good, and they're either too talented and move up and do something else, or they're not talented and don't want to do what they're doing and leave, so that causes more breakdowns.

Shortcuts taken—a lot of administrators are saying, "I know this is the way you do things," but they end up doing something differently. It's like, "I can get around this by doing that." They have no idea what the impact is on the data when it comes off the system.

Lack of ownership of the administration system—it's interesting, because on the life side we have the vendor, and the programmers. No one wants to take ownership. They want this small piece of responsibility. There's no one owner. I mean there

shouldn't be one owner, but still it's difficult and you think that this is something that should be the responsibility of, say, the programmers. They're always saying, "No, it has nothing to do with me; let's go back to the vendor for that."

Multiple administration systems for the same product cycle—someone was asking me yesterday how many administration systems we have. I was counting them up, and we have five life administration systems, of which three of them are from the same vendor, but different versions. We're now down to two annuity administration systems and that's going to go to one. Our intent is to go to one administration system on the annuity side and one administration system on the life side, because we found some of the features of some older products do not fit well on the new systems. Also, because we have subsidiary companies, we can be selling the same product, but it's going in different administration systems. So the output off the data is completely different than you would see from those things.

MS. LANKOWSKI: At Merrill Lynch we have 90,000 in-force policies that we administer on six different life administration systems. It's amazing how these things proliferate. One system does one way of testing for definition of life insurance. Another one tests a different way. You wouldn't want to have those on the same system, would you? It's kind of amazing to see how it proliferates, because nobody understands what the implications are.

MR. BRONSTEIN: One of the things we noticed, too, was that on 1035 exchanges on life insurance products, we couldn’t identify one of the systems. At least the output couldn't identify that it was a 1035 exchange, so we were paying premium tax twice.

MR. ROBINSON: Gary, one thing I've always found and I think it's our fault, is that the actuaries do not explain to the people who input data what its purpose is. There's one vendor that sells a GAAP system that's related to each policy's accounting and we're often naïve in thinking that people do this stuff right. You made a point on this. This undo, redo. A lot of people just get to the end. They don't really care what happens in the middle. But this system requires you to go through the accounting properly or it just blows up. But it's our responsibility to tell people what happens to the data. If you're a valuation actuary, you see all the mistakes. But if you don't tell them, they don't understand the impact of their errors on what you're doing.

MR. BRONSTEIN: I agree with you, but some of the times you're restricted by management by saying you're a valuation actuary, you shouldn't be responsible for data. You think, then, “Who is responsible for that?”

MR. ROBINSON: Right. And when you see it is wrong, you want to tell them.

MR. BRONSTEIN: Right, exactly. What kind of impact would you get by having these breakdowns? Inaccurate financial statements. That's a big issue. We had fluctuation in income for month-to-month that we couldn't explain, because we couldn't figure out what was causing the problem. That's a problem not only for

senior management, but also for Wall Street. If you get numbers and you can't explain your numbers fluctuating from month-to-month, that's a big problem. Luckily for us we're a part of the property and casualty division, and they have a much bigger income, so we get hidden. We can hide the $7-8 million fluctuations in income.

Inability to evaluate product profitability accurately—basically, that came down to a project we ended up working on, our first project for Six Sigma, where we couldn't tell on a product basis what we were making in our spreads. We had no idea, because credited interest from month-to-month was fluctuating. It would be negative some months, and then it would jump up to two million, and then 10 million, and down to negative three million. We had no way of figuring out, as our target spreads what do we really want to get at? Are we getting there? And we had no idea.

Having to restate financial results—if your data is really that bad and you have to restate, you don't want to know what happens. Your stock price will take a hit. You may be downgraded by the different rating agencies. It's not the kind of thing you really want to get into.

State auditor concerns—we just had an audit a few years ago. They're somewhat nit picky. Sometimes they miss a lot of things, but you're sort of glad they missed some of those things. There are some things they pick up, and you feel guilty and think, “Yes, we know about that, but we haven't allocated the resources or the time to work on that.”

Slow to market new product launches—sometimes I know we've held up an introduction of a new variable universal life because we couldn't get the correct data coming off the administration system to build the valuation extracts. That could lead to an inability to make tiny business decisions. That goes back to crediting rates. On our deferred annuity products, if you don't know what your spreads are, you don't know if you're making money. You can't really make a decision about what you should be crediting.

Why is this still an issue? The impression is that it takes too much time and effort to fix the problem. “It costs too much money. We don't have the financing.” This thing about the product features is not exercised at a later date, as I mentioned before. To give you an example, with our MGDBs the feature was a maximum anniversary value or five percent roll up. They figured the way the market was humming along, we know we don't have time to implement this, but we'll be fine. Well, the market tanked over the last year-and-a-half. People were dying and they were getting incorrect death benefits.

All of a sudden, we needed to program this. This is when I started getting involved. And it's like, "You never programmed this thing? You've had this thing for four years?" Well, there wasn't really a need to do it, and we didn’t have time. We didn't have the resources. These are the kind of things that will cause huge problems, especially if you're underpaying somebody; there are class action lawsuits. If you're

overpaying somebody, they're not going to tell you. Our problem was that it was a little bit of both. I think it was more than just the MGDB.
The impression is that the impact is small. People have an idea it's a small impact. I don't need to put any resources or any money into it. What you find out when you do actually measure it, is that the impact is a lot greater than you thought.

MS. LEE: How do you measure those programming deficiencies or other deficiencies in terms that allow you to effectively communicate the cost? How do you put that in dollars? For example, a feature that wasn't programmed now, you have to go back and reprogram it later. What are the tools used to capture that?

MR. BRONSTEIN: I don't think you know the financial impact until it starts happening. Most people don't think the impact is large. The impression is that the impact is small. Until you start finding these things, it's very difficult to assess what the impact is. I've gotten a lot of negative feedback from saying that you should not be launching products without all the features programmed in the system, because there're going to be things that happen that are going to fall through the cracks. You're going to miss it. When something is exercised two and three years later and you haven't programmed it, all of a sudden, you're going to have issues.

MS. LANKOWSKI: For some of it I'm going to suggest that you talk to a lawyer at your firm and get him or her on your side. Because the lawyers can say it's a compliance issue. Anything that winds up being a compliance issue suddenly gets funded. If it's just an actuarial issue, they don't fund that. Make friends with your legal department. I know that's hard sometimes.

MR. ROBINSON: There is another way to communicate though, and that is by sensitivity testing. If you get an error, you use extremes. When I check a report, I go to the beginning and the end. Most errors happen either at the beginning or the end of the report. To address it, you've got to exaggerate the results. How do you do that? By testing the limits. Sensitivity testing is often a way to communicate what could happen if the data is lousy, which happens often.

MR. BRONSTEIN: Right. We have a new process for launching new products. I have asked for for years, and people on my team have asked this for years, "Can we get a test environment, which we do have, to test the product features, run it through three years later and see what we come up with, to see if we feel that the data that's being populated is the correct data?" They're still saying, "Well, if you use the test data." And someone would say, "What you have in the test data, well, you've built policies for every product. So when you bring Mickey Mouse and Donald Duck to your administrator system in a test environment, we can't give you the extract just for those two people. We've got to give you everything. Then if one bombs, the whole thing bombs." You need to get around that. That's not my problem. I want you to give me a test environment. You give me five policies. I can tell you exactly what I want in these policies, plus some of the features, and you're going to tell me if it works.

We're not a very innovative company, and most of the products we launch are

because of distribution channels, through banks and broker/dealers. Our features are not that different from product to product. Anything that does come up as a product feature is something that someone in the industry has already done. But the testing of that new feature always needs to be tested to make sure.

MR. FRANK REYNOLDS: I'm afraid I come from a different, older environment of the computer systems of what you would call yesteryear. We solved some of the problems you're suggesting very simply. We had a universal set of test data that did two things. It made sure that the changes being made didn't foul up what was already there and then we added cases for each new product.

Now the way we solved your problem of identifying things was to do a comparison before and after of the test data, so that we isolated only those elements that were changing. Accordingly, if we got something in which there were massive changes that were unexpected, the programmers never saw them. The line department would never see them. My staff got them.

As for getting the changes, the computer area often has a problem with the product launch area. I can remember somebody launching a new product. My staff walked in and said, "That's just wonderful. It's now December 1 and the quickest we can do this new product is 90 days." After I got that communicated to the chief actuary, I found one morning on my desk a pile, and I'm not kidding you, it was two feet high, of all the new products they were going to bring out in the next two years and where they were. Unless you can turn around and do some of these things, I'm afraid it's a matter of being mean and nasty a few times. You get a bad reputation until you stop being a thorn in their side all the time, and then you regain your reputation.

MR. BRONSTEIN: I agree, I think you have to. I've learned from an old manager that you have to put your foot down, and you can't let these people walk over you. You've got to say this is an issue, and you've got to do something about this. Also, who is responsible? Is it the valuation actuary? Is it the pricing actuary? Are they doing the testing of the features within the system? No one wants to own up.

Data quality is something completely new within the company. We decided to have a separate department within finance just for data quality. We created a data quality team within Allstate Financial. It ended up being a group of valuation actuaries, some accountants, and some financial analysts, and we were responsible for tackling data.

MS. LANKOWSKI: Gary, how many people are on your team?

MR. BRONSTEIN: There are five, headed up by an accountant, two valuation actuaries and two financial analysts.

MR. ROBINSON: That's a high-powered team.

MR. BRONSTEIN: Yes. We learned a lot of things. One of the people on our team

is all life, and he ended not being part of this project that we worked on, this Six Sigma project. But it's amazing how many things you can find if you really dedicate the resources and effort to something. If you really tackle something you'll find a lot of things.

We started focusing on processes that weren't working. Where are the data problems? Where are they happening? What can we do to fix them? We attacked blocks of business on administration systems. You've got to take a small block at a time.

For our first project we ended up tackling a smaller block of business. It was, actually, a pretty big block of business on one administration system that didn't have any quirky features. Most of our focus was on the GAAP side, because that's what we were looking at. The problem is that we can't analyze results. It seems like, at this time, we're not so concerned about stat and tax, we're more concerned about GAAP. What's the bottom line impact?

We introduced Six Sigma methodology in December 2000, right after our team was created. It was an organization-wide effort, so it wasn't just data quality that was using Six Sigma. I'm not going to go into a real big detail about what Six Sigma is, but, basically, it's a process model you follow in order to look at your processes. You measure it through tools and then you've got to control it. You can't use a patch. You can't just fix it once and assume everything's going to be okay afterwards. You've got to keep monitoring it month-to-month.

We completed our first data quality Six Sigma project. We actually finished it, but we haven't passed it onto the process owner. We're waiting to do that when we get someone who will be willing to take it over for us. We've identified the area. They just haven't allocated a resource to us yet. We've worked on our second one. The first one is annuity. The second one is life.

G.E. Capital is probably the biggest company which uses Six Sigma tools. I believe, and I may be quoting wrong, but $400 million or 20% of their income in either 1999 or 2000 was due to Six Sigma. So $400 million of their income was due to saving through implementing Six Sigma tools. I know St. Paul is another company that started using this. Allstate among other companies is doing the same thing.

The projects are usually four to six months in duration. Usually, the first set is a little bit longer, but you try to keep it to four to six months. That's how you can scope your project down. You may ask, “What can you get done in four to six months?” Well, if you tackled every product on every administration system you'd never get to it. One of the tools you use is the DMAIC Model.

The DMAIC Model basically includes going through a series of steps and trying to get at the root cause of the problem and how you're going to control it. Define stage is when you create what we call a project charter. What's the problem? If you fix the problem, what's the potential financial impact? Measure—we measure, using a statistical tool. We use Minitab. You come up with the baseline

measurements. What is the problem What's the extent of the problem? How bad is it, and what are you going to do with the data? You get this data and you analyze it.

Six Sigma Project Model (DMAIC)

- Define
- Measure
- Analyze
- Improve
- Control

Now you've identified. Usually, in the Analyze stage you've identified the root causes of the problem. You'd be surprised at how the things you thought were the problem are usually not. The things you bring up may be small parts of it. There are other things that are going on that are causing it. In the Improve Stage.we have identified what the root causes are. How are we going to improve it? How are we going to fix it?

The last stage is, now that we fixed it, what kind of control environment do we monitoring it from month to month? It depends upon what the process is. For us it's closing the books monthly. Depending upon the other areas, I know a few of the other projects were no surprise. It was the calculation of death benefits. That was one of their Six Sigma projects. One of things was reversing and reprocessing data on the administration system. Something like 15% of all of their new applications were reversed and reprocessed. They had to say we're spending way too much time and effort doing this. How can we do it? We ended up with pretty good savings from all of our cost benefit analyses. You've got to do a cost benefit analysis, because if it's going to cost you more than the benefits, then it's not really worth doing. For almost all the parts that we picked, the benefit was far larger than the cost.

MR. ROBINSON: There's a theorem or axiom on getting errors that says, the longer you wait to fix them, the more it costs you in the ultimate, and the bigger the impact on the situation. If you can get the error fixed closest to when it starts, you're much better off. If you wait, you have to explain to analysts why your income is off by so much. If you don't catch it early it costs you a hell of a lot more money.

MR. BRONSTEIN: Right, I agree. The monthly credited interest was fluctuating month-to-month for our deferred annuities that were on this specific administration system.

FROM THE FLOOR: How much of a fluctuation are you talking about here?

MR. BRONSTEIN: I think the range we had was a decrease of about 40% and an increase of 60%. That's a pretty big fluctuation from month to month on credited interest. Obviously, number of days has an impact in the month, but this was beyond anything like that. The credited interest on this administration system is not

stored. It's calculated, but it's not stored. Thus, it is a plug in the accounting system. It will say, “I've got the account value at the beginning of the period, account value at the end of the period, your premiums, and all the cash transactions, and I'll plug what my credited interest is.” My boss's boss was always saying, “I want credited interested to be credited interest, not a plug. I want that plug out of there so I can analyze my business better.”

We compared premiums and benefits that we got from accounting to our valuation extracts. Now the valuation extracts usually didn't have all the cash transactions. They had it. It was year-to-date. The stuff that was in it was never verified, never checked, and it obviously was not right. You could compare last month to this month. It was year-to-date and nothing ever tied.

We worked with the systems programmer, who created an account value roll forward table that had all of the different cash transactions and said, "This is what I'm using to build my account value roll forward." He said then you could tie it back to what accounting gets and you tell me what the difference is. That's what we did. Valuation is a snapshot at the end of the month—what , you've got as of the 31st of the month, whereas the accounting system is being fed daily. Should that be an impact? No. What were some of the things we thought were the root cause of the problem? Manual journal vouchers were being put in, creating timing issues. They were a part of the problem, but not a big part of the problem. There were other things that were going on.

Then we needed to set up a controlled environment to monitor this because when you're tackling, you're not supposed to tackle every little problem. What you try and do is look at some of the bigger items. You could probably tackle 80% of the difference between accounting and valuation. We found three or four problems. If you can tackle those, the other 20%, do you ignore them? You can't tackle everything. Again, you can't tackle world hunger. You've got to do the things that will have the largest impact.

MS. LANKOWSKI: Yes. And it's that cost benefit analysis that makes a huge difference.

MR. BRONSTEIN: Are the small things going to ever creep up? Sure, they may. You can have a series of small policies that are generating $50 in differences and, say, it totals $500, which is 10 of them, but you ignore it. It's not that big. But if you have one big policy that generates a variance of a $100,000, what's going on here? Then you find out that something's not working correctly.

This is basically what we did. We have an account value roll forward. We have a beginning account value and an ending account value that comes from valuation. We have premiums and benefits that come from the accounting system and we have this plug. The question is, all the data starts in the administration system and why would it be different? Another question is, how could it possibly be different? It's coming from the administration system. People say, well, one's being fed today, one's being fed at the end of the month. Is it just that or is it because of the

timing? You find out that there were a lot of things going on that other areas were aware of, but had no idea of the impact. They thought, “I know that's not working correctly, but I didn't think it was that big of a deal.” It ends up it was probably the biggest reason for other things.

Basically, it was a translator error that we have. It was a series of products that came from an old administration system to the new administration system, and they couldn't separate the cash transactions between general account and separate account, so everything was being thrown into separate account. What happens when those premiums and benefits go into the separate account? Your plug, which is your credited interest, makes up for that difference. That's why you have this huge fluctuation. That was the root cause of that fluctuation. I asked how long has this been going on and he said, "We converted the block of business in '97." I said, "It's been four years and you knew about it and you didn't do anything about it?" He said, "It wasn't. We were waiting for all the products to be converted. The last product was converted in May, last month.” They waited four years to do this, and they're wondering why they couldn't analyze their results, because it was a fluctuation in credited interest.

Again, we tied the premiums and benefits from the cash ledger. We tied them to this account value roll forward that was created by systems for valuation to assume that if the premiums and benefits tie, then the credited interest is going to tie. There were other things that were being thrown into this plug that we weren't aware of, deferred bonus interest, Guaranteed Minimum Income Benefit (GMBI), and the change in reserve for GMBI was being thrown into this. All these little things go into the benefits line of the income statement, but we wanted it separated so we knew what true credited interest really was.

What did we learn? You definitely need cross-functional teams. You can't do it all yourself. We were trying to get people from different areas, and I don't want to say they didn't cooperate, but they couldn't allocate the time, resources, and effort to help us do the project. We ended up using people, the valuation actuary, the financial analyst, and the accountant to do all the grunt work. That leads to using the subject matter experts. You've got to have those. You can't just use a warm body, which we ended up using, people who are talented, but have to learn the entire process as opposed to having somebody who says, "I know exactly what's going on, I can help you, but I can't allocate it."

We picked a product that had a lot of accounting impact. We started the project in January. We had no resources from accounting until year-end was done. (I've been through that process.) We didn't have anybody from accounting until the end of March, so that made it even more difficult.

You need commitment from senior management. It's very important. If they don't buy into it, it will never happen. That's one thing with Six Sigma. We had coaches who also helped us through.

You've also got to find the process owner. One of the things we learned as well is

that we didn't have a process owner till we were done. No one wanted to take it over when they found out how much time it takes and, granted, it does take a lot of time, because you're investigating all the differences. If everything were working right, you wouldn't have that many differences. I think it took us four days to do all this. It was somewhat manual. I'm sure we could have done it in a more efficient manner and that's what we said. Can we give it to a group of people who can use technology better than we can and use it better so we can be more efficient? Because when accounting said four days, there's no way we could take the time. But they agreed to it. It's just that they hadn't allocated the resource.

Scoping the project was a big deal. First we thought we'd tackle all annuities on all administration systems. It ended up being deferred annuities only, only on this administration system, excluding the fixed piece of variable products, but not variable products and not indexed products, because there were just too many other things that were going on.

The controlled environment is extremely important. Most people fix something, then walk away, and that's not going to cut it. You can fix all the things now, but I can bet you if you completely ignored it for a year, a year later you'd have the same problems and different issues with the same problems.

Our team included operations, IT, valuation, and accounting. I think that was it. But the problem with that was is that we had people in different locations. We're not all in Northbrook, Illinois. We have a separate building for IT people, which is about eight miles away. We have operations people in a completely different building. They're about 15 miles away. We have one of our subsidiary companies that is located in Lincoln, Nebraska. Sometimes it's hard. Geography is difficult, but it's the kind of thing you need to tackle.

Our next project is actually tackling an administration system that's specifically geared toward the products that are being sold by our subsidiary company in Lincoln, Nebraska. We've got to spend a lot of time out in Lincoln. We try to avoid it, because we don't want to be flying to Lincoln all the time. But our operating committee member said this is what you need to do because this is where the problem is, and you need to tackle this thing.

MR. RALPH GORTER: Why didn't your team include anybody from the administration or the systems side?

MR. BRONSTEIN: We used operations. Operations is really administration. There are people in financial control within the administration areas and those are the people we used. It wasn't as though we excluded administration at all; we used operations to find out. They're the ones who told us about some of the things that weren't working right. We explained to them, “Here are some things that we don't think are looking right. You know, here it's working okay. It's coming through on the valuation extracts, but it's not coming through on the accounting system, what's causing it?” And they'd say, “I think it's a problem with this, this, and this.” And then you'd create some sort of systems request to fix it. The interesting thing

is most of the time they were aware of it, and they had already submitted a request to get it fixed. There are so many different systems requests that are out there we said, that Six Sigma is a big priority for Allstate. We've got some backing behind us. We're going to tell you what your priorities are. Now that's pretty unprecedented being able to tell systems what their priorities are. And we said these are your priorities, these are the things you need to work on first, and we got the cooperation. That could be an issue for a lot of companies. I'm not saying it's going to be easy.

MS. LANKOWSKI: Sometimes you can get the priorities changed, because if it's a manual process and you can actually get systems to program something you might say, look, this is a manual process. It's taking one resource out of my team of 30 to process all of these things. And even though there are five people doing it, if you add up all the time, it's one full-time employee who does this kind of thing as opposed to programming it once and not having to worry about it anymore (the calculation of GMDB and so on.)

You might get actuarial to buy in and then you get your operations area and your client services area to buy in, because it's a savings for them as well. And everybody that's accountable for budgets then is tied into this thing. The systems department does have a budget, a resource allocation model, but they don't really care which project they're working on as long as they don't have to hire anybody to do it. If you get different areas of the company to tie into, and buy into, what your project is, it actually works much better. If you're trying to do it as an actuarial department, you're probably not going to have the same impact as if you actually get the rest of your company to tie in as well.

MR. BRONSTEIN: I'd be interested in hearing some feedback from the audience. Obviously, you came to this session because you probably had some of the same issues within your organization. The question is, are some of these issues different? I think people want to hear what other issues people have. Also, what have you done to fix it or haven't you done anything? I think people would like to hear that if you guys have anything your organization has done. I'd like to listen myself, because that will help me in my future.

MR. ROBINSON: Actuaries love rollovers or roll forwards. And the results never come out right, but I've never known anybody to solve the problem. However, a very important thing is predicted results. And a rollover really is a predicted result and it gives you something to measure against and that's the value of predicted results. If you don't know what you're going to get, you don't know whether it's right or wrong. If you predict something, even though you might predict it wrong, it gives you a comparison base. And the way you attacked the thing was very good, but you had that base to start with. And the other thing is in whatever testing or work you do you've got to write the answers down, because you've got to seek patterns in whatever you do. And if you don't write it down you never see the pattern.

MR. BRONSTEIN: I think what you're going to see is that you're going to have

one administration system that's one problem and you're going to have similar problems with the other administration systems.

MS. LANKOWSKI: You have the same programmers working on them. And so if you didn't tell them right from the beginning this was a problem, they think they did it just fine. And they are getting the same results they got on another system, so, obviously, that's the way it should work.

MR. BRONSTEIN: And a lot of times you can't fault them, because this is what they're told to do. They're told to, "Program this specifically" and that's what they basically do.

MS. LANKOWSKI: Quite often they don't have an example so that they can unit test or they have one example that's a plain vanilla case. You have a male, age 35, and it goes for 10 years. Instead of saying, this is a male, age 35, and he's got four transactions on the same day. Who really is going to do four transactions on the same day? You and I look at it and say that's crazy, but somebody's going to do it. You're going to have clients that are going to call up and say, "Yes, I want this to happen. "And they'll want a reallocation, and they want to pay a premium, and maybe they want to take out a loan at the same time, because, finally they're doing some financial planning. There are a lot of stresses that can happen and cause problems all the way through at the same time. And if you don't have testing going on in some of the stress situations, you're going to have built-in problems from that, too.

MR. PAUL SCOTT PAINCHAUD: Linda, you had piqued my interest earlier when you mentioned you were involved in a reinsurance administration system implementation, because we're in the process of at least looking at systems. And then you lost my interest when you told me what happened.

I did want to mention just two suggestions I had that are just merely extensions of topics you had discussed earlier. One is, as you mentioned to build up rapport with the legal department and the systems department. I'd probably also recommend the same with state auditors and that was brought up a little bit earlier. But that way I've found that we can keep them informed during the whole project implementation. And they tend to like the fact that there are no surprises at the end.

Also, you mentioned all kinds of things with problems you may have with vendor systems, and what they're responsible for, and what their products should produce. I just recently purchased a valuation system, so I recommend that we don't take vendor contract negotiations lightly. I was actually surprised to see all that they're willing to change. You can change the verbiage, and you can alter things, and you can add things, and you can make them change insurance limits for medical and disability.

MS. LANKOWSKI: Oh, right. Vendors are great at customizing things, but they don't always necessarily test what they've customized. As the purchaser you have

to be very careful about what it is that you're expecting out of them. You have to spell out very carefully what is necessary for customizations, and what things are different about your products. Having a vendor come in is often much more cost effective than doing it yourself, but you should not be letting the testing and quality assurance part of it stay with the vendor. You still need to do that in-house.

We are currently going to outsource our maintenance for our administration systems. We are outsourcing them to India, so the company is going to work at times that we're not awake in the U.S. We're going to be providing specifications for them. We're going to be giving them detailed technical specs, business specs, all of that. We'll ship it out to India. They'll work on it. The next morning we'll come in and we will have to do the testing. Our agreement with them is that they will do a certain amount of testing, but their agreement basically says that it's still our responsibility to make sure that this stuff works. And when we made the decision to outsource our systems, we actually took into account the fact that we would have to hire additional business analysts and QA testing people in order to make the outsourcing work.

Merrill Lynch has a very high reputation and all we need is one client to be upset. All of our brokers start hearing about it and, poof, we've lost our reputation. We've lost our large clients. Word gets around very quickly among clients that have the kind of money that Merrill Lynch clients do. We're very careful, very particular about what we do and it can be a problem at times.

I had mentioned that we wanted to talk about the test plan a little bit. I think that having a test plan is crucial to any systems project. And I don't care if it's just changing a screen or whether it's implementing a whole new product on your administration system. There ought to be a test plan, and there might be multiple test plans. Frankly, even if you're not involved in testing, if you are involved at all in the implementation of a product, if you're on the product design team, you should check out what the test plan says. Because as an actuary, you're better trained than a lot of these people to understand all the implications that are going on.

And if you read a test plan that says “not to be tested” and “to be tested” you could say, well, wait a minute, why aren't we testing this? I think it's important. You need to make your voice known. More often than not, you read a test plan and you say, wait a minute, where's accounting in all of this? We really aren't going to be testing the accounting system; it's just to feed the general ledger. If you don't test it, then you've probably got errors that you're putting into place right from the beginning. And so I think it's important that you read the test plan.

The test plan is, generally, not the test matrix, not all the things that are going to be tested. But make sure that you read something like this to find out what is going to be tested, what is not going to be tested, and who's going to be involved with the testing. I can't emphasize that enough. Every test plan should have a statement that says what's not to be tested. And it's, certainly, quite agreeable to not test certain things. If you don't want to test the general ledger feed because it's just a screen change, be my guest, that's fine with me. But you need to be aware

of what things are or are not tested and what the implications are.

FROM THE FLOOR: What is the position about not testing, and different ways to then build safeguards into the system so that the application or the tool can't be used in the fashion that you haven't tested it for? We find that we're asked to do that frequently, where we've developed a test matrix. The test matrix would take four months to go through. We don't have four months. Instead, we back off. We identify the key areas in which we want to use the tool in a particular fashion. But then as a safety net for us as the developer, we build in safeguards so you can't use it in a fashion we haven't tested it for, either by graying out something or saying, “to be added at a future time.” Have you seen that in the work that you do, or maybe some others here have seen it as well, or does that not-tested element fall through the cracks?

MS. LANKOWSKI: Actually, it does often fall through the cracks. But if you've got a good business analyst on your side, that business analyst will help you to identify what things could happen and will talk to the client area who says, oh, but what does this button do? ”There's a lot of that stuff that occurs, that as an actuary you say, “Who would do that?” But it happens a lot.

I think that one of the best things that you can do is to have on your team a good business analyst who can actually look at that and say, “We're going to gray this out. We're not going to allow this key to work. You know, it works on another administration system. They're used to hitting that F6 key to commit something. We aren't going to let it happen.” Or it's really easy to program in an edit that says if this field is not filled in with a numeric, then you cannot commit. And sometimes it's just that the numeric could be zero and that's okay to put in there, but they can't commit because of that. You need to work with your analysts to make sure that these things happen. And I can't say enough for having a good business analyst. Pay him or her whatever it takes.

MR. ROBINSON: There was an item that I wanted to bring up before that just mentioned hard or soft edits. Sometimes people want to put in hard edits which won't allow you to put in a code, and then further on you realize that there is a reason why you want to do that. And sometimes there's a soft edit that says you could do it, but you're going to get an error message. What's your feeling on those?

MS. LANKOWSKI: As long as there is training, you could set up these edits. Today, it's a lot easier to set different levels of security, so you have one person who's allowed to bypass an edit and somebody else who is not. And so you might set up a situation like that, where you have one team that can do something, but the other team cannot. There are ways that you can do it. You couldn't do that 20 years ago, but today it's much easier. You can do that and I, certainly, recommend doing that.

MR. ROBINSON: You sometimes put in defaults and you wind up with a lot of people age 99 or strange issue dates.

MR. BRONSTEIN: It's a problem with checks and balances. I've seen systems rarely have checks and balances within the data, where you have to populate every field. Sometimes they will have a year field and say the field can't be any larger than the current date, but then everything else is fair game. I think that's a problem, especially as we're looking on the payout side, where we had a different administration system, and they could put in any date and you'd have issue dates. The policy was issued in 2055. I don't think we issued the policy in 2055. But the administration people, when they type it in, there's no checks and balances. They type it in, and it's done. I'm looking at it and I'm thinking, “This isn't going to work.” And he says, "I don't understand." And I say, "Look at the issue date." I need to change that. No checks and balances.

MS. LANKOWSKI: Sometimes it's something as simple as typing too fast. You know, you transpose numbers very easily. Instead of it being 1998, you get 1989, so there's a lot of going on. Unless you have some kind of an error that comes up on the screen, that's not going to happen. We do a lot of testing of that sort. Our client area does an awful lot of testing to make sure that they can't do strange things like that. Error boxes come up on the screen. And quite often, yes, our client services people say, “It's the error that I get all the time.” and they just ignore it, but at least there would have been an error at some point. And then maybe the new person who comes in is not going to ignore it the same way the old person does.

MR. BRONSTEIN: On the payout side, we ensured that when the whole “year of issue” was complete, we'd evaluate the data and we'd say, okay, take the present value of future benefits for this person at the issue date, add in any loads or expenses, commissions, and does it match the premium that we've got? That's what I was doing for a number of years, and you'd be amazed at how many did not tie. We're talking differences in reserves of tens of thousands—millions of dollars. If they're not caught up front, you're going to be holding reserves that just don't make any sense.

MR. ROBINSON: After the initial stage you figure a report is right. But as a valuation actuary and with monthly statements, you've got to look at a lot of reports quickly and you've got to look at them right before you use them to make sure they're right. One trick that I found that always works is I write down the number of pages and I keep track from month to month, or from report to report, how many pages are involved. If the number of pages goes up or down there's always something wrong. What I said before is if there's an error and the program can't understand like bad data not considered by the program, it blows up and you've got too many pages. Now, generally, you have some idea. If it's a new business you know what the activity is. But number of pages of reports never should vary dramatically month to month. If it does, that report is wrong for some reason. It always works.

MS. LANKOWSKI: Looking at things is a very simple way of knowing what's going

on right from the beginning.

FROM THE FLOOR: What testing tools do you use?

MS. LANKOWSKI: The systems area uses SQA, which recently was bought by a company named Rational. It's mostly used from a systems standpoint for resource planning.

One of the problems is that you do all this tracking and you take all these ideas and get the metrics put together about how many bugs you found, and how many bugs you fixed, and how long it took you to fix some of these things. At the end of a project you go through this big debrief and you figure out what went right, what went wrong, and whether this had an impact, but the problem is that we never use them again afterwards. The next project comes up and instead of reviewing what the last project was, we start from scratch and start measuring these things all over again. Hopefully, your business analyst is hoping that he or she can find fewer errors this time, less bugs that they track. But we don't start out looking at the errors from last time. I think that that's one of the things that we should probably start doing—looking more often at what those errors are before we start a project and paying attention to, the problems. We need to keep it in the back of our mind that we need to fix it from the beginning.

There are a couple of tools out there to help you track the bugs so that you understand them. For some of them, you might think the bugs are too small and that it's not going to have much of an impact. At that point, when the project is done you should be distributing those bugs out to your users to say these are the things that we know were wrong, you're going to have to a work around. They're going to scream and shout and say, “No, this project is not done” or they'll say, “Okay, fine, I can work around that.” Tools are pretty tough to find out there. There aren't that many things that are tied to actuarial things or to insurance products or anything that's going to help us really do testing.

FROM THE FLOOR: We have our quality assurance people telling us we can't regression test a lot of situations in our client server.

MS. LANKOWSKI: Actually, regression testing is the hardest thing that you can do. You can't often roll something back. And your client server is not something that they generally want to put into a model office environment. It's just too big. It's not portable enough. An administration system can be ported into a model office situation and often is, but to put your client server into that, it's just too big. And having two copies of the same thing is extremely unwieldy. You have to extract a certain amount of data and put it out into a model office platform and then move that forward. It's not an ideal scenario. I don't think anybody out there yet has designed a way to fix that.

MR. ROBINSON: Well, if it worked, then take that into a testing environment or a model office situation that takes real data, which is good, and then you can use it in nonproduction areas by utilizing what happened last night, That’s a way of

regression testing.

MS. LANKOWSKI: Still, with a client server you're talking about a massive amount of data. And to have two environments at that size, there is a problem occurring with it.