

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

Good evening, everybody.

Well I will now hand over to Shikar, who is a very senior member of our human-centred test team.

Shikar has been with Assurity for over eight years now, coming to us from South Africa and has worked in both our Wellington and Auckland Offices.

Shikar is very experienced in this space around ERP and Zero transformations and has worked actually really closely with one of our panellists, Toni in terms of the 2degrees transformation.

So you are in very experienced hands with this lovely panel, but I'll hand over to you, Shikar, as the moderator and leave it to you to introduce our guests.

Awesome. Thank you, Katie.

Good afternoon, everybody. Welcome to our event today.

So, as we all know, ERP transformations are fraught with challenges.

Let's face it: we've all heard those horror stories of ERP transformations, or large software implementations, software projects going wrong, and none of us enjoy being in those sorts of projects.

So that's the reason why we're here today: is to listen to and learn from our two very experienced and esteemed panellists, Toni Warren and Andrew McPherson, who between them, have decades of experience in implementing.

Make us sound old.

Many, many, years... a few years... a decade between us

Some years of implementing large-scale ERP transformations.

So, we want to find out what works, and more importantly, what doesn't work so we can steer away from that.

So, we've prepared a few questions to kick off the panel discussion today.

Some of these questions will centre around some test project management activities, like managing a budget through a test program or ERP implementation and post-implementation for maintenance purposes as well.

Also, let's not forget our favourite topic of the moment right now: Artificial Intelligence. So, let's just jump straight into our discussion today.

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

[0:02:09:23]

**Can you share a specific example from your experience where an ERP transformation program faced unexpected challenges or setbacks?**

**How did you overcome those obstacles, and what lessons did you learn from that experience?**

Sounds a bit like a job interview, doesn't it?

I studied as much.

To give you a bit of background, I was the Test Practice Manager at 2degrees and started in 2020, and the project was already seven months into the making.

We started testing late. I think it was probably the next year.

Either way, we started testing, and during that time, we had two lockdowns, three program managers.

We had The Great Resignation, high staff turnover, and lots of sickness.

So a lot of change, a lot of things happening.

And in amongst it all, we had to keep somewhat to a timeline.

Ultimately, you know, let's do testing and development together.

I think we fell into that normal track where you end up.

We kicked off testing. We made some good progress, but unfortunately, we found some defects, as you would expect, and there was not sufficient capability within the development teams to support any rework whilst they were Dev in flight.

I think that's something that happens commonly, not only not having enough capacity to support rework in flight but also, not just in the ERP, but also in other programs.

It's something that we need to consider when we do our plannings because often it happens, more often than not.

Be my key learning.

[0:04:09:00] So, just to set some context, over the last two and a half years,

I've been leading a large transformation and Stuff.

You probably know Stuff from the [stuff.co.nz](https://stuff.co.nz) website. But in addition to that, we also have fifty newspapers.

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

So, nine of those are daily newspapers, and the rest are weekly newspapers like The Sunday Times and the community newspapers.

So, when I started at Stuff two and a half years ago, the platform was quite old, a lot of different systems and things like that, and we really had to transform everything across that.

And that included how all of the Ad inventory was managed right through all those commercial processes.

And just to add some context, I've just finished up at Stuff over the last week actually, I just finished last Friday.

One of the big challenges I think we had was that the business was in an operational mode and had been in it for many years, and we didn't actually have any BAs that were sitting there, kind of documenting processes or anything like that.

So processes had changed over the years since the systems had initially been implemented, and we were in a state where people were using the systems, but that knowledge about workflows and how systems work, there was no documentation.

So it was really inside everyone's heads, and it was inside those SMEs, and it wasn't a single SME; it was this, people only had the knowledge that applied to their specific role.

So when it came to a point of actually starting to understand what we needed, in terms of a set of requirements or even what we should be going out and looking to purchase, there was no one I could talk to and gather that,

And the challenge, in addition to that, was that all of the people I could potentially talk to were busy with their day jobs.

So, I couldn't sort of stop the press and say, hey, let's all get together in a room for a week and put together a set of requirements or document our workflows.

So I was in a really difficult position there, and, I think a lot of businesses probably find themselves in this role now because businesses have leaned out, and you don't tend to keep those roles in a business, like BAs, you know they're expensive, and they are often seen as extra, why do you need them if you're just operating day to day.

But then, when you come to that point of change, they're absolutely critical.

So what we had to do was, we actually had external BAs come in and actually work with us and our subject matter experts to help draw that knowledge out.

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

It was a combination of drawing the knowledge out of our subject matter experts as well as training some of our subject matter experts on how to be BAs themselves.

So, there was this two-way knowledge transfer that was going on so that would enable us to effectively be able to start by documenting our processes.

And then those people were then able to move through with the project into then helping us work out what the configuration would need to be on the new system, test the new system, train people onto new systems and things like that.

So it was really that people side, but I just want to make one other point as well as a project, particularly when you're under a lot of delivery pressure, and you've got a lot of new people coming in, there is often an over-focus on the happy path.

The happy path is kind of when everything flows fine, and there's no errors or obstacles on the way through.

The challenges are not often on the happy path. It's when something goes wrong.

And often those problems don't come out until much later in the project.

And because there's been an over-focus on the happy path due potentially to this lack of knowledge inside the business, people don't know what all of the obstacles or other kinds of edge cases could be.

You discover those, you know, as you go through.

It's a process of discovery, which means you're constantly having to go back and then maybe make modifications to the happy path to incorporate those things.

Those are some of the challenges I had, but certainly happy to elaborate later on if we need to.

So, the key lessons learned there are business processes documented in order for the program to continue?

I think the learning is that because there's a change in a business over time, you can't expect your documentation from ten or fifteen years ago when the old system was implemented to be up to date; people would have changed how they use the system.

So, I think the learning is... I think you need to...

This is where external partners like Assurity are really important, and getting some impact players in that can bring some structure to, say, your analysis approach into the business and help upskill your people so they can start to document their knowledge in a much more structured way.

I think that's really the insight that I gained.

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

Cool. Awesome. Thank you, Andrew.

[0:08:51:00]

So, let's move over to a hot topic, and this one is around data migration.

**In the context of ERP transformation, data migration is often a significant challenge. So, could you provide examples or an example of how data quality practices have impacted the success of your projects?**

It's always a challenge, you know, for a number of reasons.

I'll take a slightly different tack on it. Data migration, from my point of view, is not a single thing that happens on the evening of go-live that you bring all the data across because now we're in a situation where you're not necessarily working with a single system. You know, if you have your ERP system or your CRM system and all these other systems, there's a lot of interconnected systems together in data sitting across multiple systems.

When you're transitioning off your old environment to a new environment, you actually end up going through a number of transitional stages and transition phases, you may call them, and through those transition phases, you actually need to keep data consistent and effectively have multiple migrations as you go through, and as you move into those transition phases, you may do some parallel running as you try to manage risk between your old state to new state.

It means that data migration is not a one-off event. It's a continuous thing because you're testing and getting ready for that migration, running parallel, then migrating, checking your data back, and validating your new environment against the old environment. Then, you're getting ready for that next transition state as you move through, as you may be introducing a new system into the environment and deprecating an old one.

So, my view is it's good to take a systematic approach where you have some sort of platform or something that helps you migrate data. Our approach was to actually use a cloud-based integration platform, an integration platform as a service. I mean, look, there are lots of different ones out there you could choose, and I don't know whether we chose the best one or not, but we use something called Workato, which is basically an integration platform.

But what it effectively enabled us to do is very quickly integrate between different things, almost like an ad hoc basis, and to enable data to move very quickly between different systems. And when that data was moving through this platform, there was a full audit trail and all those effectively like transactions that were moving, you could go back and have a look at them and at any point in time see if you ended up in a kind of state or transition state, you think maybe some data had got corrupted or damaged or the transformation hadn't worked properly, you were able to go back and look at those and potentially rerun different migrations and things like that.

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

My view is that you have to take a really systematic approach to data immigration and not think of it as a one-off piece of custom code that necessarily moves data. Think of it maybe more as a platform that really supports complex integrations in a way that can enable data migration.

So there's a question there, I think, around data migration, but I think, Toni, maybe add to that, yeah, definitely build on it. I absolutely agree with everything you said, but my key experience in data migration seems to be that we always undercooked the activity.

We leave it to last. So it's always last to come into the environment to test against.

And we don't do enough upfront to have that in a ready state to complete our testing within the timelines that we're aiming for. So I think in addition to that, we need to make sure that we plan out that activity with the right resources and understand that it's quite a complex activity that needs quite a lot of focus on it upfront.

Otherwise, you just keep pushing forward, pushing your time forward and not completing the times that you want.

**[0:12:53:17]**

Yeah, I completely agree with that. I mean, I've been on projects where it gets to that last few weeks with people getting ready for go-live, and it's like, well how are we migrating data? Everyone kind of looks at each other, we hadn't thought about that.

When you actually look at the data migration, that's a six-month project in itself. It's like, well, we're not going live in four weeks. We're going live in four months because we now need to migrate all this data.

You always think of migration as simple, and we just take data from here to there.

But so often, there's some massive transformation that has to happen on the way through, and that's the difficult part. So, it needs to be very structured, very intentional, and done early. So, in terms of done early, how early would you say? Is there a specific project phase or test phase that you would want data migration done by?

These activities will kick off in parallel. It's just about making sure that you understand the complexity. Have the right level of resources and knowledge assigned so that you can progress. I can speak to the example that I was working with at 2degrees. We hit a single point of failure. We had one person who was responsible for the data strategy and data migration strategy. It wasn't enough. It was too little, too late. We finished testing functionally before we even had a level of quality and our data migration strategy. I think you've got to be doing it right at the start and thinking at the same time you're selecting your technology and look at requirements and specifications.

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

You've actually got to be thinking about data migration. How are you going to migrate data?

What are the requirements and data migration? Looking at the current data schemas and structures you've got, versus what would be required in that new system and. effective at it and hopefully be able to get that priced up. And you know, on the timelines right at the start. Well, thank you.

**[0:12:53:17]**

Ok, let's move on to another hot topic, which is around customisation.

**In the world of ERP, customisation is often a double-edged sword. How do you strike the balance between tailoring your ERP system to meet specific business needs and keeping the software manageable and upgradeable in the long run?**

I've said it before to people when we've talked about this, it's about bending your business around the tool, not the tool around the business customisation. The lowest level you can get is the better in terms of cost of ownership ongoing.

2degrees, we had 10%, just under 10% customisation, best practice 5%. I was talking to companies that had more than 20% that were going live and actively reversing at the customisation and trying to align more closely with the total horror story. One that I remember quite clearly, as we had a very manual process for the approval of purchase orders, was about eight layers. Now box functionality for two, three, six, five is for, and we put a change request and to increase the approval layers to eight, and I'm kind of like out-of-box functionality best practice recommendations for why aren't we bending to that rather than putting their change request in and increasing our automation?

So we went live with that, and some time later, it's been reversed back out. So customisation, I think the approach now really is to go more configuration than customisation, really try to minimise the customisation as you said. If you start to zoom out and look at what's happening, you know, across I.T. and products, a lot of these tools are becoming commodity products now, especially with the move to SAS and the cloud and things. If you're taking the approach of trying to minimise the total cost of ownership of a platform, trying to choose a solution that can implement the workflows that you need, and then you're trying to minimise any change because that will minimise the total cost of ownership. The problem is when you customise, then you need to keep those people with the skills that can maintain those customisations across version upgrades and all of that.

It's just an ongoing cost to the business.

**[0:17:25:00]**

I totally agree with what you said; it's really about changing your business to fit in with the workflows that are within that commodity product that you're moving to

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

rather than trying to change that product to fit your business. I think most business owners or your senior leaders would recognise that now and actually support that approach.

So in your instance, Toni, was that reversal because of the cost of ownership and maintenance overhead, that became cumbersome? It actually became quite a cumbersome process once it went live. I think it must have been. I certainly know that as a cost center owner managing that process, it was submit, ring someone, can you approve, ring the next person, can you approve, who's next person to approve it?

I'll ring that person and get them to approve it through an email or Team's message. It was a lot less transparent once it became automated. So you didn't quite know who to ring. So things took longer.

**[0:18:24:15]**

Ok, that's good insight. Let's go on to a question for Andrew. **I know you've been attending a lot of meetups and conferences around Artificial Intelligence recently, so what specific AI tools can organisations use to enable more efficient workflows?**

We used it at Stuff in this recent transformation project. We used AI in a number of ways throughout, certainly in automating steps of workflow, but we go specifically to the project areas where we used it, and it was really effective work and documentation.

We used a product called ScribeHow. It's a tool that enables you to effectively use a software, you know, package, whatever it is, and click through the screens and do what you do, and it automatically creates a user guide off the side capturing the screens, writing what you're doing.

Our learning and development team will use that function to very quickly produce user guides. It's sort of almost like magic. There's other similar software that captures the videos automatically as well. So, something that would have taken weeks of work was done within days and hours.

It was able to be updated very rapidly as we were making changes to the system. So I think in documentation, using AI is really good. You know, we actually used AI, across a number of things at Stuff, right from this new system introduced AI to automatically layout the newspapers. So when you look at a newspaper now, we're where all the articles on the pages is now determined by AI.

The stories are all written, ranked, and rated, and they're also tagged by another AI software that's tagging all the topics and entities that are mentioned within that story. That story gets picked up by an AI engine and, depending on it's priority, is then laid appropriately within the newspaper in a way that fits with all the advertising.



# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

It scales the images to make sure that the story fills the whole page or multiple stories fit together, either in rows or columns and things to fill the page. It's quite incredible what can be done. And before this project that was all done manually, I mean, you think there's nine daily newspapers and there's fifty other papers in total, there's a lot going on there manually.

So you know incredible from a productivity improvement.

It meant the team were focusing more on maybe some of the more artistic things on the front page doing some things like that.

The majority of the paper was laid out with AI. I think now when you look at the tools, you know, if you're tracking, even ChatGPT just this week, it's going multi-modal, and you can use it in a way to do so many things across images and text, but actually one technology that I think is probably the most productivity improving is actually these various AI.

Different products can do this but recording and automatically transcribing meetings. So, I think a few years ago we'd always have a project administrator in the room taking minutes and then summarising that down so you had a record of the what was discussed and what the action items were. That can largely be automated now, with tools that effectively connect into the meeting and actually transcribe it and do bullet points.

We were using that throughout the project as well. you know, again a great productivity enhancer. It's a time saver.

**[0:22:14:23]**

Ok, cool. Let's move on to project budgeting and test budgeting. **So, ensuring alignment of user needs, staying within budget and adhering to timelines are essential goals in ERP transformations. So, could you share any strategies or best practices for achieving this delicate balance, particularly when unexpected changes or user requirements emerge during the project? Plan for the worst and hope for the best.**

That's pretty much the strategy. I think a lot of people from a testing perspective under cook defect contingency, and I know that last time I went into a project, the development manager was trying to tell me it was 5%. And I was like, what? We've got a whole new frontend, a new Web, two new Apps, a Chatbot, and 5% for a new integration layer.

We eventually settled on 30%, which would be fairly typical for a program that size.

So it's the same for an ERP program make sure that you plan for your defect rework through contingency and you know you hold your timelines and you can either do that by extending your timeline or adding more people with regards to the actual

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

overall test budget + or -10% you would have all heard that, make sure you include that, this enables you to support any sudden new requirements or surprises that come on the horizon or that you find out as a consequence of doing the job and yeah.

And generally you should be able to meet your target. I know that within our program for 2degrees. Yeah, we were on target. We did. We meet our financials, and that doesn't happen that often, really does it. So, how often do you find yourself eating into your contingency? Always.

I mean things happen, right, but then I will say, you know, it doesn't really matter how much contingency you plan if the teams wrapping around you, aren't able to support you as well.

I'll give you an example. The average days to fix defects in our last migration project. 54 days. Didn't matter how much contingency we put in, we were never going to have enough. These reasons for that. Whereas on the digital side, 6 days. We had a defect rate of 40% but 6 days turnaround didn't really matter.

**[0:25:01:14]**

Well I think testing is a key technique for managing risk within a project because, the more you test, the more certain you are certain things are going to work when you go live and you've managed a certain amount of risk there. And I think this also links back to what we were talking about before in terms of customise versus configuration, because the more you customise something, the more testing you're going to have to do because all of that customisation is likely to have you introduced bugs and you've done things that, you know, are not right, and you to have to fix them. So again, you know, my approach is now is like, let's try and stick to the base product as well. You know, as much as we can, that's likely being tested by, you know, the company that's written it or other users that are using that functionality. And, you know, you need to do a certain degree of testing to make sure that that works.

But once you've done that, you have a solid foundation and then you can start deciding what are the parts where we have introduced risk through our customers or configurations that we need to spend more, put more attention on. And also you data migration, you're making sure that data is right. And, you know, with my most recent experience of what we've been doing, we you know, we actually had a tester from Assurity helping us and actually a team.

So Rachel here and, you know, we had to make some pretty tough calls kind of on the fly as it got towards the end, for example, as we were going live, our integrations to our billing system was still wasn't fully operational at that point in time, which would normally you know, make anyone quite nervous, but I actually knew that our data was secure. With the testing we'd done and that integration platform we had, we could easily make changes and we could present that data through to our billing

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

system. However, which way we needed to make those changes. Yeah, within the window of time we had before an billing run and it was a calculated risk, calculated decision based on an assessment of the risk we had.

But there was, there was only possible because of the testing we'd done. Had we not done a sufficient testing, and if I was concerned that our base data would be corrupted or potentially or something, we would have had to move out our go live date. So it's really, you know, I think testing is kind of an enabler really in enabling you to to plan and prioritise activities as you go through.

And as such, you know, again, I think it's a continuous activity. You know, I mean, I think we all know shift left and all this sort of thing. It's not something we leave to the end.

So, you know, in terms of budgeting, I view it more as a continuous spend on the way through. And maybe you do ramp it up a bit, you know, at the end as you need, as your data comes through and you need to test your data migration and there's a big push there, but still, you know, if you view it as a, as a constant allocation of resources or something across the project, I think it does become a lot easier to fund or to plan for.

[0:28:01:14]

**Ok. So let's stay in the same ballpark, and we'll go post implementation.**

**So, how can organisations balance the need for continuous innovation and delivery with the need to reduce the Total Cost of Ownership and mitigate project risks? Test automation, test automation.**

I've said it before, I'll say it again. Look, I mean, that was my key learning from visiting the number of people who'd been through this before we started, and that was, you know, get your test automations sorted, there's a number of patches and updates that you have to do on a frequent basis.

I was a cost centre owner. My boss had prickly pockets and wasn't looking to spend any money and I'd done some really estimates that it was looking at around 300k a year just to deal with the updates and that was a bit unacceptable. So after having met with those other organisations,

I went back and looked at our budget of a measly month, and we relitigated that, and we ended up 26 weeks, 12 sprints, and we automated... we had 3600 test cases for the program, which we distilled down into about 430 automated test cases and another 22 integrated test cases of the systems.

We didn't have a lot of automated integration when we went live. We only had 2 integration points automated that would have taken 8 weeks. 4 people took 6 hours, 1 test engineer and up to a week if you included maintenance and any additional

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

automation you needed to add. Now we had that really within the context of our program, and we were able to release testers.

And anyone who's been in testing just before go live, how many of you run around pulling together regression packs, trying to test deploys and prove that they work? And we've all been there, and it's a hive of activity and stress. And I think our guys, they were pushing a button and they were coming back 6 hours later.

Yep, it looks good. So we were releasing people and providing feedback within a day. So then once we went live we had that ability to do that as well. So, it definitely sounds like a wise investment. Yeah, definitely. I know this is helpful when it comes to planning. The cost of our test effort in the program was around 17% of the overall program budget, and of that 17%, 11% was test automation.

So, we started payback before we went live and we expected to get payback within the first few weeks and Andrew, anything to add on that? I think you've done a pretty good job of that. Yeah, I think he just said just to add a little bit is that whole, you know, shift left. Yeah, by finding defects early you actually minimise the cost of everything because you know you've it's likely to be quicker to fix because there's you know maybe you don't live so you're not going to outages and change windows and all that sort of thing.

So I do thoroughly believe in testing what you can as soon as possible and running that in parallel.

And if you're developing things to, you know, test drive driven development when you're writing your test cases, you know, before the developers even start. So they've got something to test against. And obviously that building unit tests and all that into it, into the code as they go, so that when you do get to the end, you can run the automated testing.

And you know where you're at. So again, it's the cost of managing risk, I believe, as I said before. So reducing cost of ownership, automation, automation.

**[0:32:07:13]**

Does anyone want to ask a question?

**I'm really interested in the end user – the customer. I'm just wondering what has worked for you guys in the past, what approaches you've taken to really testing the experience of the end user.**

Learnt this from you in the work you did on our D365 implementation, acceptance testing, user acceptance testing. We took a two-pronged approach, right? This is the functional testing coming in and verifying that the job they do today they can do on the new system. But from a data perspective, you know, we're a bunch of contractors sometimes or where we've got outsource partners supporting us doing

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

our testing, but the businesses are the ones that really know the data and understand it.

And so as a form of acceptance and as a part of the testing, we embed part of the business and our data migration design and make them part of it, they help us, right? So that by the time we end up in production, they're familiar with where it's come from, where it's gone to, what might be transformed with it.

Hopefully, they've got a good understanding of what they're receiving, or expect that they would. Probably a slightly different tack is, and this relates to what I was talking about before in terms of the knowledge that's within the business. And the challenge that I was faced with is that we were trying to change the business process, the workflow, with the new systems bringing in.

**[0:33:38:09]**

So it's very hard to take existing users and say, "Okay, what do you think of this new system?"

Because it's like, the new system doesn't do all of these steps, you know, it's broken. Why does it do that? So that's why we had to take the approach of pulling some open-minded type users, you know, more senior advanced users or subject matter experts out of the business, kind of educating them not in the mind in terms of, hey, these this is the art of the possible, this is what new systems can do these days and start to get to think beyond the current processes, and have them on that journey so that they could then help us to streamline or completely transform the workflow. And then they had the inauspicious job of then going back and selling that into their colleagues who are back in the team. That's the big challenge because, you know, I'm thinking of one user in particular when we showed him this new system that we wanted to go to first, he said, that will never work.

He said that's never going to work. And it's like, no, we've got to make this work. It it was this journey in terms of ok, you know, slowly, sort of converted him to. At the end, he was he was like a raving fan.

This is great and he's sort of been, I guess enlightened in the way of this new system.

And it's interesting that he was very involved and was trusted by the users, so then he was involved in sort of selling it in and training all of the users how to use the new system.

He's then gone on to now be a product manager in the business so that he hasn't gone back to his old job. He's actually really enjoyed... what was interesting was he saw the cultural difference between the technology part of the business versus the of the operational part of the business. And the operational part of the business, they

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

always expected there to be like a concrete answer that this is how we're going to do it all.

This is this is how it's done. Where in the technology part of the business we're like, we don't... we're figuring it out or we're making it up or we're working it out and designing it as we go, and sure we'll get to an answer at the end.

But it's quite a different way of working. And I think that was the thing he struggled with at first when he was saying that will never work. He was coming from a point of view of very black and white. What I see today that will never work.

But he wasn't open to like, that doesn't need to be the end result.

We can configure it and transform it into something that will work and be even better for our team. So I think that's the problem with users is that they don't always, you know, you want involve users, definitely. But I think you have to be 1 you have to make sure you're choosing the right users and you have to be taking them on that journey to sort of open their mind and to the art of the possible, particularly if you wanted to completely transform that the business process and automate things.

Because at the end of the day, we were putting in a system that was going to result in the team that had come from being reduced, you know, by half or two-thirds of size, because suddenly there was all this automation, which is again a really difficult thing to deal with. So there's all those kind of people and cultural issues that you're dealing with as well on the way through, which I think makes it really difficult.

[0:36:58:00]

**What's the balance between 1) manual versus automation testing 2) business versus dedicated QA doing the testing. And who maintains automated testing post-go-live?**

If you're talking in the terms of an ERP program, we wouldn't do it until it's stable code. Once the code was stable, we would look to do our automation. But if you talk in the context of BAU again, you might take an automation-first approach, particularly in the digital domain, which is another big conversation.

So I'll park that for now, but business versus a dedicated QA doing the testing. I don't rightly or wrongly, I kind of think acceptance testing as a marketing activity. It's the first time that the business come in and see the working system. You don't want them to have a bad experience. My recommendation is always to have professional testers or the dedicated QA team going in there and making sure that it works before you let the business at it, otherwise.

To Andrews point, they go out and they tell everyone about how great it isn't.

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

And then it really makes it difficult to transition the change into the BAU. So yeah, when it comes to business versus dedicated QA, Dedicated QA followed by business when we get an acceptable level of quality, right. And it's a carefully managed and again to your point and making sure that you've got the right type of person, inquisitive mind understands that things are changing in that environment and that nothing's like production.

And who maintains the automated testing post-go-live? In my experience, it's been testing, but then I've worked in larger companies that have got full testing practices. That might not be the situation that you're in, and in which case you would be choosing a tool that would work for the people that are using it, you know, coded versus codeless.

So again, that's context-based, in my opinion.

**[0:39:06:10]**

**Testing in parallel to development could be easy to implement in a new project, but in an ongoing project where testing is a catching-up task, how can we correct things?**

So when you say testing is a catching up task... just to put some context around that, that's sort of implying that there's kind of defects being reported and they need to be validated. Is that how you interpret that question? No testing is a catchup task. It's like, do you develop and then test rather than test and develop in parallel? Right.

So we're talking waterfall versus sprint or Agile, right?

I do see ongoing testing, it is a capability and it's also a mindset in the business that quality is built in. You can't test in quality when you're testing at the end, all you're doing is finding defects.

But if you if you have quality practices as you go through, then you have a better chance of building it in. I think that the catch up is I think is more like we're just saying about, you know, people reporting defects and that you have to validate those and then you have to fix them.

So you end up with this backlog of defects. And again, it's just a matter of prioritisation. It's just the way we always have is P1 P2 P3 or whatever and working through them just as best you can.

**[0:40:38:22]**

Do you see ongoing testing as a capability that needs to be built as part of business user roles – especially as organisations move to more Agile, continuous delivery?

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

Again, it's context based, right. I worked in telcos. You couldn't get the business to come in. There's quite a lot of complexity, a massive amount of change, whether it's on a network perspective or just in the IT area. And then you've got to integration between them both.

You've got businesses that are very siloed. They understand their bit, and the other person understands their bit. So I think in that situation you do need a capability to support testing and change, right? But if you're a smaller organisation, then I think there's a blending that can happen. Yeah, or possibly a level of capability that needs to be embedded within subject matter experts so that they can contribute effectively to the test effort.

[0:41:39:06]

**How do you balance focus between the happy path and edge cases when edge cases may represent less than 1% of instances and similarly small revenue impact?**

I'll pick that one up because that was the point I raised earlier. I find that each case is or is not on the happy path can be really disruptive because often what can happen is, if you go off the happy path and you end up in a manual process, it's something that then needs to be handled manually.

Even though it might only happen, say 1% of the time, the amount of effort that goes into managing those those edge cases can be like 50% of the whole effort of the team or whatever. It's just really disruptive, really time-consuming, especially with those edge cases, you know, because those cases can lead to corrupt data, you know, manual processes, people not being billed properly.

And so you end up in a situation where you may have to go back and unwind some whole of the transactions manually and then replay them through. And all of this net messiness can just take so much time. And that's, again, what I've personally experienced with projects I've been on when we haven't managed those cases properly. And I think the question here is around that focus and I would say that you inherently end up having to put more focus on the edge cases because the happy path is pretty simple aside, this happens, this happens, this happens and yeah great, we're through.

That's kind of what you do in the first week. The other kind of 51 weeks of the year is that, hey, when this odd thing happens here, what do we do? You know what? A good example is and I'm stealing something from from my sister here. She just started working for an organisation that is in health care and they that she lives in the U.S. So they provide health care systems in the U.S.

And the problem in the U.S. is that you have people that don't have birth certificates and, you know, systems are built in a primary key. It's like, yeah, everyone has a birth certificate, everyone has an ID and all that. You get people coming in to



# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

hospitals that you know either are undocumented migrants or they grow up on a reservation.

And so when they were born, it was, you know, they were born on the reservation and there's no birth certificate. What do you do? And so, you know, there was a scenario there where hospitals couldn't provide health care to people that suddenly turned up with some really bad medical condition because the IT system wouldn't allow it.

You know, in these and it's kind of you know, this is probably a bit of an extreme case, but is actually real life in so that you start to realise. From my point of view I think each case is often where the where the real values delivered because, you know, if you can handle them properly, you save a lot of time and then you can get much better performance out of your automation.

**[0:44:43:18]**

**In terms of data migration, how important is it to ensure your existing data is cleaned, and how much effort, resources and time is needed?**

Context based. Look, I've worked on some really difficult data migrations. You try doing a network, migration. So we had the physical and logical aspects of and inventory being migrated from a basically it was garbage in, garbage out. You could write whatever you liked. And I don't know if you know much about networks, telco networks, you actually have to build the network from the inside out, and it didn't matter how much cleansing we did on that project.

In the end, we had to build it manually, and we stopped that migration. Yeah, just as needed. Whatever you need. I'm certainly not for cleansing data more than you need to, but, yeah, I mean, it will be needed. Yeah. I mean, the project I'm working in at the moment, we've done a bit of a fair bit of data cleansing before we migrate because we are migrating in tranches.

We're doing some whilst we migrate and go back and do a sweep of that data that we didn't migrate. Yeah, it's what you need. So you gave the number on the test automation being 10%. Do you have a time effort, resource. You mean 11% of our budget? Yeah. So you have something similar for data migration?

No. It's different depending on the situation.

**[0:46:21:14]**

**In Andrew's view, having being a CIO is his advice that business bends to platform rather than vice versa? Other CXOs might be very reluctant, especially CX/UX related.**

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

Yeah, I mean that's the million dollar question really, which is how much is the business prepared to change?

And again, it depends. Are you prepared for the cost of customisation and what that entails? And I think if you're a large business, maybe like a telco or something like that and you can see you know, realistically you can afford the cost of having those teams there able to support the customisations.

I think you've just got to have the conversation upfront when you're making, you know, go into it with your eyes open, consciously know what you're doing because if the business expectation is that are putting in a new system that's going to do everything it's going to be really cost effective to operate. But you get to the end, and you've done so much customisation that you need to maintain all these development teams and testers and everything.

You really haven't achieved the business outcomes of that whole thing, you know, that whole transformation that you've done. And you've got to really ask, well, why do we do it? So, I would just say go in, have those discussions with your other execs at the start and be hard. Often what I find is when it comes to talking about dollars and you're really pragmatic, you know people are certainly prepared to compromise.

It's like, well, you know, we're going to need a team of five people to maintain it. This is going to cost us, you know, plus support is going to cost us \$1,000,000 a year or something.

People suddenly start to view these things quite differently. So, you know, I think there's one thing we haven't probably covered enough here is the financial and budgetary side of things.

And in the case it comes back to test automation. It's an investment upfront, but you get a long term return from it where if you're doing testing manually, you always having to do it. It's this might be cheaper upfront, but it's this constant grind and yeah, on the on the budget. So again, it's about making these decisions constantly rather than getting to the end of the project and thinking, damn, whoops.

**[0:48:37:03]**

**In hindsight, what are the questions you wish you had asked before choosing your system and/or vendor?**

Well, I think you've got to look, and I'm stealing an answer from you off the spreadsheet. So you've got to look at who else is using the software and even go and see them, spend time with them.

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

I think it's very painful when you're the first to implement a system in region or in country, an ERP system. And that will drive a lot of costs. If you're the first one to be do it using a system in a particular way or the first in region. And there's very little support around you.

It sounds luxurious like you're getting on a plane and flying to somewhere else where someone's using the system. But, you know, it's well worthwhile. And, you know, I, I always, you know, often when I'm later in a project, always wish I'd spent more time in due diligence and found out where those real fishhooks were before I got in neck deep in a project. But having issues trying to go live.

[0:49:44:21]

**Can Agile principles be applied to test earlier in ERP transformations and de-risk end phases that are closer to critical milestones like go live.**

Well, I think we did. I think we applied. We certainly tested aspects of the delivery sooner. We didn't wait till all of development was complete before we started our test activity, as I mentioned. But we planned to do that in some respects. I think we had customisations and configurations.

We did one phase and then the next. And there's no reason why nobody else can't do that or anybody else can't do that. So yes, I would agree. Agile principles can be applied. Yeah, I think thinly slicing things and the Agile mindset. Yeah. Just think about Agile, it's all around the Deming Cycle, which is, you do something, you PDCA, whatever it is,

Plan, Do, Check, Act, you take some action and then you look at how it went and you, you look at the result and then you improve and you know, you're able to improve over time.

So this, this idea of constant improvement. But I remember some big projects I worked in Telco back in the 90's. Showing my age here again, we had these massive big projects that, you know, was called telecom at the time and they went on for years before anything was planned to be delivered. And they were so off track before we even got anywhere near the delivery.

There's a number isn't there, the business changes something like 18% a year. So, by the time you deliver a project 2 years later, you've already lost. So I think the discipline of having a smaller increments and being able to play back what you've done to your key business stakeholders is so important to keep you on track, particularly if there's a business change.

And look, even if you're implementing a big monolith ERP system, you're like,

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

well, can't we just deliver that and that one big chunk. But these work going through the whole way through and you just have to view work differently in terms of it's not just delivering that big chunk, it's all of those activities that combine to turn into that end result.

[0:51:56:12]

**Can Toni clarify if she means 30% of the ERP implementation budget should be for testing, or does the contingency need to be 30% of the test budget?**

Oh, 30% would be horrendous, wouldn't it? Industry standard used to be about 24 or 25% didn't it. Look, our budget was 17% of our budget and that included 30% contingency during testing execution plus some additional contingency for sudden and extras requirements or needs. So no.

[0:52:34:13]

**Andrew, who should be on the hook to write the test cases if this is done ahead or in parallel with development?**

Yeah, good question. I think there's a certain mindset and discipline for the testing and to be a good tester. I always find it really interesting when I'm interviewing a tester or a test manager and they start asking me pedantic questions about, you know, the contract and when they can take leave or what hours they, you know, because that's what you want from a tester.

You want them to be, you know, going through every line of the contract and in asking you the tricky questions. And that's a mindset, and you know, not everyone has that mindset. And often, you know, developers, it's classic developers a bit more, you know, gung-ho about wanting to build things. They're not necessarily wanting to check things.

And, you know, managers always think a big picture or something like that. I don't know. But it's there's a specific mindset that with a set of disciplines and practices and structured approach that goes around testing. And I think it's not necessarily the person you need. You need that mindset within the team and then those disciplines and those approach. And I think you need it right at the start, because you know, if you don't have that mindset, it's difficult to write a consistent set of a well constructed set of test cases that would then test the system and then you know, you may not even think to put together a defect register and prioritise it and everything that goes with that.

So think of testing, you know, as a practice, a discipline and a mindset, not just a single act of I, you know, I bring up a screen, I push it back and I expect X to happen. That's going to right at the end of a whole lot of other things. But if you don't have the mindset and the approach, then that pushing the button and something happening on the screen, I mean, that's almost irrelevant, isn't it?

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

So I would say, absolutely.

[0:51:56:12]

**Can Toni define what she means by defect rate and what is a good benchmark?**

So defect rate, usually we talk about the number of defects per 100 test cases, so 30 defects per 100 test cases would be 30%. Depends on the project. If it's out of the box, no integration, you'd hope for 5-10. Large amounts of integration, certainly 30% is good. I've worked on projects where the defect rate's been 50%, and even higher. For an ERP transformation similar to the one that you undertook at 2degrees.

What was your benchmark? I won't talk about Benchmark I'll tell you what it was, in system test it was 20%. In SIT it was 40%, bit higher than you'd want, and performance test, security test and all those peripheral type testing, it was 21%. So it wasn't too bad. I think 3600 test cases, just over a 1000 defects.

[0:55:47:09]

**When in the journey, do you involve business users? Some of the examples where issues have been identified, sound like it was too late? Was it waterfall?**

So are we talking the test journey or the program journey. Because the program should be upfront, right? Yeah. What I was describing before is we took people, SME's out of the business right at the start and involved them as really business users that we eventually turned into product managers and various other roles and they were then that conduit back into other business users and all that.

So again, I think, right up front, you've got to be gathering requirements right up front and then going through vendor selection and all that. You're going to need the business right from those point. And it's good to then be building people who have that knowledge and skill that you can then, you know, they can be on board right through the project as you, as you come right the other end.

So it's from day one. And from a testing perspective, the same. You want to know what their acceptance criteria is so that you can check that when you're doing your testing that there's no gaps between what they want to check and what you're already testing. Otherwise, this could be some unpleasant surprises. I think you also alluded to earlier when you spoke about the acceptance testing being the marketing.

Exactly yeah.

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

[0:57:10:16]

**Thin slices can only be achieved if the foundation is there. How do you balance getting foundation in and convincing stakeholders about vertical slicing with the business?**

This is the challenge isn't it. And depending on how you can slice things up.

There is a lot of context there. But your point is really that if you don't have the right foundation in place, then it's hard to then start to deliver. And, and often, we talk about micro-services and all of that. But often if you're working with a big monolith system, and maybe the integration with it is really limited, you can't. and you know, I agree, you just sort of have to work with what you've got, really. And I think, again, if you've got that mindset of wanting to move towards a more Agile approach, then it is a journey that you just have to take the steps that you can as you go to get more Agile as you progress.

But maybe you can't start by slicing things, and that's kind of why I talk about transition stages because where I've been is that we had to keep that big old kind of system running right until we turned it into we migrated the very last thing off it, but we were starting to replace parts of the functionality in the old system with new systems.

As we as we turned on but then having data being sort of migrated across in real time through this integration platform. And in the end we would end up in a much better Agile position, but there was always this pain of being some parts, being able to be agile and new platforms, some parts still being on old platforms and just navigating through those transitional states.

Yeah, obviously much better if you can go with a big bang.

[0:58:52:20]

**Ok, next one. Penultimate question for now. Do you have any examples of where there was a challenging stakeholder that was not engaged in the ERP transition and how that was overcome?**

I think the thing with stakeholders is that often, stakeholders kind of just want it to work. They just want to get on with their business because if they have systems that are not performing properly or performant, it's going to make it hard for them to hit the targets.

The challenge is, when you're causing problems for those other stakeholders because the systems not working right or whatever and they're not going to hit their targets which is going to cause problems, it's a really difficult conversation. I think you've got to just try and work with them, be as open and transparent as you can

# ASSURITY

## TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

and having that discussion because, I mean, unfortunately it's not a perfect world that, you know, with technology, you are going to have issues on the way through.

I guess it's also about understanding what the problems are and trying to overcome them right. Yeah, open the lines of communication. The usual. One of the challenges is often when you're going live on a big project like this is that there might be a defect that you have to put a manual work around in place to go live, and that is using resources or people in the team to run that manual workaround while you're busily trying to fix it.

And that's kind of like this friction or the strain on the team. And that can be, again, a really hard discussion and often be being a discussion, which is, yeah, how long am I going to have to run this manual workaround before you can deliver me the system that actually removes that and enables us to run more efficiently and negotiate, you know, the business wanting.

The technology to pay for those people that are running that manual work around. But often is the case with technology, you don't have the people budgets to to resource another part of the business and provide resources for that around it. So it's kind of a lose-lose. There's no other way to do it that business unit needs to be on board and do it themselves.

Again, you know, a difficult conversation to have.

[1:01:01:04]

**Last Slido question. How are ERP transformations being done today, Agile vs Waterfall.**

My experience mainly waterfall. What I find is often you end up in a situation where from a team point of view that working in some sort of agile methodology within a team, but then from a more of a program management around the outside of portfolio management, you have a much more of a waterfall approach where you're thinking about the type of scope of the program and the budgets and the timelines of it.

But you've got this agile delivery mechanisms which are fine grained inside. An interesting insight on that is that it's very hard even to go out and recruit young Gen Y, Gen Z, whatever it is that they want to work in a waterfall model, they all just want to work in Agile. That's all they know.

Often, that's all they've worked in. And if you kind of go to them, well, yeah, we don't do that agile stuff round here, you know, they're not even going to sign up.

They're going to be like, nah, you guys are so like 1985, whatever, we don't want to work here.

# ASSURITY

TRANSCRIPT FOR FUTURE-PROOF YOUR ERP TRANSFORMATION PROGRAMS

---

And so I think as leaders, we have to think about that if we want an engaged workforce that are youthful with lots of energy who can work on our projects. We have to think about how we can empower them and engage them within that sort of Agile methodology, because that's what they want, and they want to work remote, and they want to have their scrums and their stand ups, all that fun stuff. Their ceremonies.

We can talk about it like old cynics, yeah. No, but seriously, I know we make fun of it, but it is true from an organisation point of view, I think, look, I grew up, you know, I sort of worked on waterfall projects way back when, but that's not what your workforce expects in any anymore, I think.

So you've got to be pragmatic in terms of how do you get the best people? How do you empower them? And I think a lot of that is through those those scrum teams, but you put some guard rails around the outside to make sure you actually get done what you're supposed to get done. I think that question took us back to the beginning of your decades of experience.

**[1:03:24:18]**

**Last question, just to wrap up with a quick-fire question. if you have one piece of advice to give people embarking on digital transformation, what would that be?**

Always ask questions. Make sure you ask questions.

You're not the only one who's ever going to do what you want to do.

You can talk to other people who have done it, get their learnings, snag bag and tag, and call them your own and make it work.

I absolutely agree. Going as I said, whether it's, you know, the local industry here in New Zealand getting around and talking to people who might have done it before and who have experience as good as that.

It's that context of their perspective that you really need to go to knowing where that where the issues may lie in how to approach it. I think these are the more you can learn on the front, the more the more that you can then set expectations and manage those expectations with your stakeholders, because it's that stakeholder relationship, which is the most important thing as you go through.

And the more informed you are, the easier that will be. So yeah.

Awesome. Thank you Andrew, thank you Toni. Thank you all. They were great questions that came through.