Transcript

The Xledger Podcast: Scoping and Implementing a Finance System Part 1

Podcast commences

00:00:02 Sian Hewitt

Welcome back to Part 2 of the Xledger podcast. In this session, the Xledger experts will be discussing how to avoid the pitfalls when implementing your new finance system.

We hope you enjoy.

Introductory segment

00:00:18 Sian Hewitt

Hi, everyone. Welcome back to Part 2, where we're going to be talking about implementing the finance system, so talking about the practicalities of what is the best way for a smooth transition when you are changing finance software? I am still joined by my lovely colleagues Zheng, Tim, and Ana. We are all the team here at Xledger.

We are going to be talking about the key challenges we see organisations face when it comes to implementing finance software, how you can deal with things like integrations, and sort of top tips of what to look out for.

So, first things first, what are the key challenges you see for organisations when it comes to actually implementing their new finance software? Ana, do you want to kick off?

00:01:03 Ana Rodrigues

Yeah, sure. So as a project manager, when clients actually start implementation, what I notice is there, there are a lack of clear expectations. They are just so excited because it's a new software, it's going to do a lot of things for them. It's going to be amazing. It's going to make their life so easy, and they forget about all the work. It comes with it, and how much time they will have to dedicate into actually making the software come to life.

So yeah, they come to me, and they have no idea how much time they are going to have to dedicate to the to the implementation of Xledger, how much resource it's going to be put into it. So, I think it's really, really important that from the start we set the clear expectation that it's going to be hard work.

00:01:50 Zheng Zeng

We are seeing more and more customers that when they are thinking about business transformation or system transformation, they are doing multiple things at the same time in the organisation. We've seen a group of customers, especially in certain vertical markets, that are upgrading their final software as well as some other software they are going after at the same time, which is a very difficult challenge for any organisation to manage in terms of resource and prioritisation.

And that is one thing that we can see that is a key challenge we are seeing a lot recently, and Anna has been involved in a couple of these projects. I think I'm sure Anna would have some comments on what those difficult situations could be when it comes to these.

00:02:43 Ana Rodrigues

Yeah. So when clients try to implement multiple systems at the same time, it numbers of things can go, from integration not working, and being very, very complex integration and that makes that they need more resourcing on their side, so that can delay the process of implementing Xledger as well which will impact of course the timeline we have set at the beginning and it's going to impact the budget as well, so we don't want that, but sometimes it happens. It's when they try to do a lot of things at the same time, sometimes it can be a very smooth transition.

Others, unfortunately, they are more often it, it can be a disaster because it's a lot to do at the same time and it's depending on the same resourcing, and I think that's the problem. It's the key people working on implementing Xledger are going to be the key people working on implementing, for instance, the housing management system.

So, it's something to think about. It's better to maybe do one at a time.

00:03:51 Zheng Zeng

I think prioritisation is very important in these scenarios where we are competing for resource and there should be a better management of expectation in that what should be done at a point of time and what can be done later on as an enhancement of that feature.

Then, if that is managed well, that could control the resource and set the clear expectation and manage the timeline much better.

00:04:22 Ana Rodrigues

I agree with Zheng. And this is where having a project manager, a dedicated project manager on both sides would help, because then the conversations are there, the expectations are clear, and the conversations can be happening. So hopefully the train wreck is avoided.

00:04:40 Tim Wright

Slightly more complex implementations, again, we see a big difference between those projects where there is a project manager on the client side and when there isn't someone to take ownership of controlling the resource and the allocation of tasks and the time management and so on. It can really help, particularly as you say, when you're implementing multiple systems, that's when things start to get trickier. It's not always necessary that the vendor side project management is often more than enough to get a team through more simple or standard type of implementation. But as soon as you add complexity that that extra project management can.

00:05:16 Sian Hewitt

Would you say that having a project manager if you were implementing multiple systems, would you say having a project manager for each implementation of those systems is key. Or is it one that could be overall for everyone? Or is it more beneficial?

00:05:32 Ana Rodrigues

Think having one per system would be overkill. Think we would need someone to oversee it from their side and to be able to kind of coordinate everything and make sure that with all the systems they are implementing, everything is working up to a timeline up to the objectives and to the milestones. So I would say 1 is more than enough.

00:05:52 Tim Wright

And one person brings consistency. If you have a good overview of what's going on with the different systems, you identify pinch points and make sure that you're smoothing out the path.

00:06:03 Zheng Zeng

As it is possible that for very large organisation you could have multiple project managers each focusing on individual projects that they're managing. But they need to coordinate it to a single person on the top that is coordinating all the project managers to make sure they are allocating tasks and taking the work correctly as well. That's possible as well.

00:06:23 Sian Hewitt

So, it's as much as about the client-side resource and how they dedicate their project managers or time as much as the way that we do where we provider consultant who has accountancy, trained background and also a PM, like Ana, who can take control so it's about both.

Again, it's that partnership thing we were talking about in Part 1, wasn't it, that both sides need to have the right resource to be able to transition as smoothly as possible I guess, right?

00:06:50 Tim Wright

Absolutely. And when we get asked about timelines – how quickly can we implement Xledger if people want to go live quickly – the answer is almost never, you couldn't go live quicker. It's almost never from an Xledger side or one of our partners' sides. The answer is not; we can't get you live quicker. It's how much resource do you have on the client side to commit to it and as much as we want to help, there are certain things that are that are our clients' decisions, and we can't make decisions on their behalf or certain tasks that we can't carry out.

So, it does come down to resourcing on the client side almost always.

00:07:31 Sian Hewitt

So, you mentioned some of those pain points, Ana, about some of the challenges that can come from the implementation side.

Are there any tips that you would say to how to overcome them other than obviously allocating the correct resource both sides? Is there anything else that you'd add to that?

00:07:48 Ana Rodrigues

Communication is key, so start to have those conversations early on in the implementation. So, communication and also flexibility because things happen and we know that, and we know that they are managing an implementation and their business-to-business as usual, so they have multiple tasks that need to be completed.

So, it's important for us to be flexible and to understand that, to change things whenever we need to and well, yeah, that falls back to a Plan B. So. if this doesn't work by this date, what is the utmost final date that we need this data by for instance. So yeah, it's always working on flexibility as well.

Understanding the need for integration

00:08:26 Sian Hewitt

Yeah, absolutely. So earlier we were talking about multiple systems and integrations and stuff. I mean, I think at Xledger we know what a hot topic integration is and how important integrations are when you're implementing a finance system and when you're trying to get that insight from lots of other technology as well so you can sort of have that one single source of truth.

We know talking to customers and businesses that integrations can be quite daunting. How do you ensure that the integrations are planned and executed effectively and like what are the best practises can organisations adopt to make this as seamless as possible?

00:09:05 Tim Wright

Yeah, I think to start with, I would always say prioritise early. So you may have three, four, five, or six different business systems which could be integrated. And the first piece of advice I think is always identify which ones are most important to integrate and understand why you're trying to integrate as well. Is it time saving? Is it added control? Is it elimination of risk? Which of those brings the highest benefit or which one brings the most risk if you don't integrate them and then prioritise those because it's easy to get lost in that, and start with [the idea that] I've got 7 systems, and I need to integrate all of them all at the same time, and that can become overwhelming.

Once you've got your list of integrations, I always think the next step is to go and speak to the vendors of those of those systems and ask them. There's a chance that they've already built those integrations. There's always a chance that integration already exists. It's already available as a product which is ideal. In that scenario, if the integration is already built and it matches your use case, then that's perfect. You can really just save time, save costs, save energy by using what's available out-of-the-box.

If build is needed. If the integration does need to be built, it's not already pre-built, neither the vendor has built that integration before then, I would always say the next step is to identify your technical resource early.

From our experience, that is the number one reason why an integration doesn't get finished. It's that there is, perhaps, not a technical resource allocated to the project who understands in detail the technical requirements and is able to build that integration themselves. There's lots of resources out there to help you identify those resources, if you don't have them internally. You know we understand that not all organisations have developers on hand to start building things, so there's always people you can turn to, but I think that is that's key. And if the technical knowledge isn't there then it's always going to be difficult to achieve the outcomes that you're trying to achieve.

00:11:09 Ana Rodrigues

Yeah, I agree with Tim. Choosing the right integration method is very, very, very important. It's being aware of the resourcing you have in house and being able to outsource it if you need to. A lot of the integrations we see that are not completed, it's because it's a complex integration and because the companies don't have the in-house capability of building that integration and then, yeah, things don't work out very well.

I also would say test integration early on. It's really important to make sure that everything is working. It's just not building the integration. Sometimes it can be a very simple one, but if it's a complex one especially then test it very, very early and make sure you're happy with the process.

00:11:52 Zheng Zeng

And just to add an extra on my points from integration; integration is a big world. From transforming data and migrating data from A to B, I consider is an integration. Whether that's automated or not, they are very different, but they are still integration.

So, when we talk about integration and when we're considering integration, even it's a manual process of getting the data in, it can still be considered as successful integration. So it's important to distinguish what are the expectation for the integration so it isn't considered a failure when it's actually migrating the data successfully.

The second thing, I think it's worth mentioning is that the documentation for integration is very important to make sure that integration process can go smoothly. It comes from the point of finance teams, as well as technical consultants who need to know what the integration is supposed to be doing, how frequent the integration needs to happen and what are the expected outcomes.

Without clear specification and documentation, because integration has multiple party involved, it can get lost in translation and without documentation everybody can get very confused. So, for me, documentation for integration is extremely important.

As well as to make the process more successful, we need to have a mindset that integration building is a very iterative process in that you can never be 100% when you first start the integration. There are always other bits that when we come to do the spec we might not have considered as we haven't got large volume data, or they consider all the scenarios and integration always needs to be tweaked along the way.

So this will help manage the expectation that customer and software vendor side needs to be resourced for a long period of time for the integration to allow the appropriate amount of testing and real-life data coming through.

00:14:19 Tim Wright

I agree with that and actually when we helped us spec an integration at the beginning, we have seen before getting held up trying to cover with the integration every possible scenario and, sometimes, it is helpful to work on that sort of 80%-20% rule and identify the 80% of the scenarios and the 80% of the data. Because almost all of these processes have something that that is unique and different. And, in this scenario we need to do it differently and trying to build an integration to handle all the different potential scenarios can sometimes add a lot of complication that doesn't really save time, doesn't really add the efficiency, and doesn't really bring the value.

So, even when you figure out which systems to prioritise, it is also worth making sure you prioritise the scenarios that you're trying to integrate as well so that you don't lose time and focus trying to solve problems that don't necessarily need solving.

00:15:13 Sian Hewitt

Absolutely. I think one of the key points that I've taken from consultants as well, I've done a few of these podcasts now and I've heard that testing is key.

Just test, test, test, and don't be scared to break the system is what I've heard. You test it as much as you need to try and break it, and then that is going to be your best way of succeeding to make sure that you get that seamless real-time data as well across your integrations.

Practical tips for aligning new finance software with existing business platforms

00:15:39 Sian Hewitt

Thanks for those points guys. What are some of the practical tips for aligning the new finance system with the existing tools for example like a CRM or an operations platform?

00:15:49 Tim Wright

Yeah. I would say to start with, just start by designing what you want to happen and draw it out. So what data are you looking at? Where should it go? When should that happen? What should the outcomes be? Putting in that work to begin with just really helps, particularly when you're talking to the vendors or whoever's delivering the integration. It just helps to give everyone that absolute clarity. We know exactly what we want to happen and how we want it to happen in a perfect world, that's always the best starting point.

And then speak to both vendors. I think I mentioned earlier that we know that, historically, some software vendors haven't focused on their integration capabilities as a high priority, and that means there are still those that are trying to catch up and starting to build OpenAPIs that you can connect into but don't have a fully featured set available yet. And so, it is worth at that early stage speaking to both vendors then and saying look what are your capabilities? What can we do? What's feasible have you done this use case before? Are there any limitations that we should be aware of heading into this project?

I think the earlier you can identify that and have that reassurance from both sides that yes, we understand this is feasible and we agree that this is feasible, the better the bigger chance of success. I think you know we have seen it a lot of times before when

clients have come to us and said we would like an integration with System X, and System X's website says we can integrate with systems and so it all seems very positive and then we speak to them and it's clear that actually it's a very, very limited capability to build this this new custom integration and that prevents it from going as far as the client would like it to go.

We'd probably still get a solution but is it the automated solution that was ideal. So the earlier you can have those conversations, the better to make sure you have that clarity.

After that, I think data cleansing gets overlooked almost always. So cleaning up your data can take forever. It's often a big task if your data hasn't been kept clean and well maintained in the past. It's usually a manual job. It's always a boring and unappealing job. It's never something that you want to invest a lot of time in.

00:17:51 Sian Hewitt

It's the last thing everyone wants to do!

00:17:53 Tim Wright

Absolutely, yeah. No one wants to do it, but there's no amount of integration wizardry that can overcome bad data. You're not going to get the outcomes you want unless you get that data clean and continue to keep it clean. So that's always my tip for making sure you allow enough time to do that.

00:18:13 Zheng Zeng

Definitely agree with the data aspect of the risk, as well as clean data is important. It's also very important to point out that we should have a single source of data which goes in hand with [the fact that] there shouldn't be any duplicated task of inputting data into multiple places when the data gets duplicated and copied. There's no way of maintaining that cleanly. As well as, build and integration that can cater multiple sources of the same data.

Sometimes the integration is built perfectly, but because the source of data is coming from multiple places that they are not in sync, it causes the integration to fail. But that's data related, not necessarily integration related. But it is caused by data not being clean or not having a single source of data.

How manage the fear of losing historical data

00:19:07 Sian Hewitt

So, on data, I've seen in the market out there in content at the moment that there is a slight fear or resistance against changing your finance system because of the fear of

loss of historical data. Is that something you're seeing? Is that something that people are nervous about? Or how easy is it to make sure that you maintain your historical data that you've had in the past.

00:19:30 Zheng Zeng

They historical data from previous finance system, we would always say that Xledger is capable of getting all the data in. The challenge is that the more historical data customer tries to bring in from previous system, the more work is needed to be done.

There is a balance of considering how costly it is bringing it over, comparing that to the benefits of what it is brings. It is possible that some of the data can be exported from external previous finance system that just kept offline, that doesn't come into Xledger but still achieves the same result.

There is a risk that historical data could be lost because of previous finance system has limited capability of exporting data. We have seen that. If that does occur, it is a judgement call for the customer on what are the core data that must exist for Xledger to function and how to getting that data into Xledger.

But it's a consideration for customer to have with vendors to talk about what is appropriate and what's not. That conversation should never be shied away from and that is one of the conversations that should always have when we considering bringing historical data into Xledger.

00:20:53 Tim Wright

I agree with Zheng. I think a good system will be able to handle the data. A good vendor will be able to help you migrate the data it is. It is a judgement about how much data you bring in and whether it's worth the effort.

I think what we typically see or hear during an implementation is that people want to bring across seven years' worth of transactional data and, the main reason for that is normally because of the statutory requirement to keep seven years of historical records in case we get audited for tax purposes. And what we typically find is that actually there's not a need to bring in seven years of historical transactional data. It's possible to store that offline on a separate drive on a separate database or a separate set of spreadsheets, depending on how complex or how much data there is to store.

Because the reality is that you don't actually need to interrogate six years ago or seven years ago of historical data frequently enough to justify the effort of bringing it into the new system. You need it available in case you get audited.

So, we're normally talking about what's your reporting need, how many years of comparative reporting do you need for management purposes, normally for

management reporting? And that normally drives the decision about how much transactional historical data to actually bring in.

00:22:10 Ana Rodrigues

What I often see is that they want to bring seven years, but then when they actually realise how much work would be involved in mapping the old data into the new system, they just give up the idea.

00:22:23 Sian Hewitt

It's not worth it.

00:22:23 Ana Rodrigues

No, it’s not worth it.

00:22:26 Sian Hewitt

When you said, Zheng, about that it can sometimes be relative to the system they already have is that, would you see that it's more common in legacy systems when they have to bring historical data over? It's more difficult or is it all cloud-based products or on-premises? Is there a theme?

00:22:46 Zheng Zeng

It is definitely that we've seen the older type of software that potentially are going out of service or stopped development, are more likely to have a limited capability of exporting the data because they are not geared up to do that as well as continue to provide support for customers to do that. Though, that's one of the reasons why finance teams should always stay up to date with an up-to-date finance system that is continuously developed for the future.

What is the role of leadership during a finance implementation project?

00:23:22 Sian Hewitt

Amazing. Thank you. So next question I have on my list is about leadership. So, Tim, you mentioned in Part 1 that you had an awkward counter with a CFO who stood up and said they don't need a new finance system when you when you were in the middle of the process. But yeah, so what is the role of leadership in making sure that the project actually delivers?

00:23:44 Zheng Zeng

I think one of the most important points for me in this area is the guidance and the strategic direction senior leadership can provide during the whole process is very

important there. There will usually be a lot of decision points and there could be compromise. There could be things that have competing resources. With a senior leadership that gives us direct and clear guidance on all of those points, this would reduce confusion, would help progress towards the timeline as well as manage the expectation for individual parties, people who are involved in the process. So that's the most important aspect of me.

00:24:31 Ana Rogridues

I think like Zheng was saying, having clear roles and responsibilities, it's really, really important that, from the beginning, you start defining those roles and responsibilities, so everyone knows what they need to do, and they are accountable for it as well.

It also it's really important open communication because if everyone knows what's happening, if the objectives are clear, if we all know what we are working towards, if we are all in the same page, it's going to make everything much easier and it's going to make people engage with the process

00:25:06 Tim Wright

On the roles and responsibilities, I think we see this quite often, especially with small teams. It's hard to have that separation between roles. But I would highlight the importance of having a sponsor level role. We would always assign a sponsor from the Xledger side, and we would always ask our clients to assign a sponsor and making sure that role is clearly defined.

It’s natural, especially among small teams, for the leaders who are driving the change and ultimately accountable for it, to try and get involved in the detail as well as stay close to it and of course that that feels like the right thing to do, but it's helpful to have that separation. When you have a lot of decisions, a lot of changes, and a lot of conversations going on throughout a project because there is a lot of communication, it's easy to lose perspective. It's easy to lose sight of the big picture.

I've absolutely been there myself as a consultant. When you're the consultant and you think that you've had to say no to some requirements a few times throughout the project for whatever reason and you start to think this project isn't going well. This this isn't going to be a good outcome. Why is this not going to be a good project?

And then I've certainly had it where the project sponsor or the project manager has pulled me back out to look at the objectives, and said, let's go back to the project objectives and look at what we're trying to achieve here. Are you happy that we're actually going to meet all of these objectives for this client? And you reset and think, OK, yes, the stepping back again, I can see that we are still having a positive implementation.

But on both sides, it's easy to get sort of sucked into the day-to-day and lose perspective of the big picture. So, keeping some roles who are not involved in the day-to-day aspects of the implementation can make a big difference because on both sides, both teams need someone they can turn to, to provide them that perspective and that reassurance, or to tell them if there's a problem, a genuine problem that needs to be resolved as well. I think that that separation is important.

00:27:01 Zheng Zeng

From cross-departmental alignment perspective, it's useful to have department leads that build an overall environment that work together. That energy and culture would delegate down to individual, department and team. It's not just a case of showing that at the leadership level departments are working together. It also helps manage the conflict of interest, sometimes across departments. We do that well at Xledger where we collaborate across departments and foster the culture of working together to understand each other's challenge, to understand how we can help each other, and the compromise and to make decisions on what's best for each department at point in time.

That aura and that energy is very important to have when we are considering successful cross-departmental alignments when we are implementing a project.

Key advice for future-proofing your business systems

00:28:07 Sian Hewitt

Great. Thank you, guys. So, the last question, I want to ask was, basically your advice.

What key piece of advice would you give to organisations to address their implementation challenges or even with integration as well as they start this process and what should businesses prioritise to future-proof their systems to try and avoid issues down the road?

00:28:30 Ana Rodrigues

So, I think for to have a successful implementation resource is key. So, it's really, really important that companies start thinking about the amount of resource they can actually give to this project and define roles and responsibilities early on.

Also have clear set of objectives: Why are we implementing Xledger? Why do we want it? What is it going to make for our company? What can we achieve by it? Some tangible objectives are really, really, really important so that we can measure against it, whilst implementation is going and after that, of course.

00:29:07 Zheng Zeng

Just to add on Anna's points, which we already discussed earlier as well, for me the other two important aspects are communication very important throughout the organisation throughout the project team, as well as guidance and direction from senior leadership throughout their whole journey.

00:29:26 Tim Wright

It also goes all the way back to system selection, I think. You know, we talked in Part 1 about how do you scope what you need, and future-proofing at that stage is important. And that could be because you anticipate changes in your organisation that you need to make sure you're prepared for.

It could be that you anticipate changes in your business that you need to be prepared for. Or it could be that you don't know what's going to change, like a lot of businesses don't, right, there's a lot of unexpected change within organisations. And going right back to which system to work with, which vendor to select it's important that you're factoring that in how? How flexible is that system? How easy it is it to make changes in the future that weren't planned for? And often it's useful to look for feature sets that maybe you don't need today but are available in the future. And again, as I said earlier, a Clear History and future of development of that product.

If you're working with a product that's continually developing, then the chances are they're going to release feature sets that you didn't know you needed when you bought the but become helpful for you later down the road. All of those things can help you to future-proof, and particularly with true-cloud technology now; it's so much easier to be future-proofed. It's so much easier to adapt and change and pick up new features and everything else compared to legacy on-premise systems.

00:30:48 Zheng Zeng

Thank you, Tim for adding something at the start. Now that's triggered me to think of something at end as well. We normally consider finishing the implementation as a success and the point of project close, but actually it's not.

There are a lot of benefits that only get realised three to five years, potentially down the line where all the whole organisation is integrated with using this system to do reporting, to the budgeting, to do forecasting. It's important that we always have a mindset and have a plan for realising the benefits that finance software is bringing into future. It has to be continually assessed.

And if we do look at and review how successful an implementation was, we don't look at the point of project close. We extend that further. We go back to our customer that has started the Xledger journey three years ago and see how they are doing and that is a very good point that the customer gave us real valuable feedback on how Xledger has transformed their whole day-to-day life.

00:31:57 Sian Hewitt

Very nicely round up there Zheng. Thank you.

Well, yeah, that's the end of my question. So, thank you so much everyone for all your insight and your input and I think, it was really an insightful session. Look forward to seeing you soon.

Conclusion

00:32:13 Sian Hewitt

Thank you for listening to the extruder podcast, if you would like to get in touch with someone from our team, you can get in touch with us on mail@xledger.co.uk, or you can head to our website at www.xledger.co.uk. Thank you.