NSW SMALL BUSINESS MONTH 2026

What Happens If Your Website Person Disappears?

A practical continuity plan for small businesses and growing organisations

Wil Brown · Zero Point Development
Recorded 29 October 2026

EDITED TRANSCRIPT

Lightly edited for clarity and readability. StreamYard captions have been joined into paragraphs; obvious name and transcription errors, some verbal fillers and minor repetitions have been tidied. Examples, advice and the order of discussion follow the session as delivered. This is a transcript, not a separate technical audit or fact-check.

Section timestamps refer to the supplied StreamYard VTT captions. Select a topic to jump to that section.

00:01:02 Welcome and acknowledgement

00:02:36 Opening poll: could someone else take over?

00:04:58 What would stop during a two-week absence?

00:07:43 01 | What does your website person actually do?

00:12:07 02 | Where are the single-person dependencies?

00:17:47 03 | Who steps in?

00:24:14 04 | Write a useful handover

00:30:28 Give your handover records a home

00:33:41 05 | Test whether the plan works

00:34:21 The takeover exercise: a customer cannot access a course

00:38:05 Planned departure versus sudden absence

00:40:17 Choose one responsibility this week

00:42:07 The Website Handover & Cover Plan resource

00:43:16 Questions and closing remarks

Welcome and acknowledgement [00:01:02]

Good morning, everyone. Thanks for joining me for this NSW Small Business Month session. Today we are looking at a very specific continuity question. What happens if the person who normally looks after your website is suddenly not available? That person could be an employee, a freelancer, an agency contact, maybe a volunteer, a committee member, or even yourself. This is not about assuming that the person is unreliable, but people leave, they get sick, they change jobs, retire, or simply become unavailable at inconvenient times. This shows whether your business can keep moving with all the knowledge sitting in one person.

So if you joined my previous session, is your website quietly becoming a business risk? This kind of builds directly on top of that webinar. That session helped identify gaps in processes. And today we're going to turn that into a practical handover and a cover plan. Before we begin, I'd like to acknowledge the traditional custodians of the land in which we meet and pay my respect to their elders, past, present and future. I extend that respect to the Aboriginal peoples joining us today. I'm proud to be hosting this event as part of the NSW Small Business Month programme in partnership with the NSW Government.

Okay, I've got the comments on. So if you want to ask any questions, then please drop them in the comments while I'm talking and I'll try my best to get to them and maybe there'll be some time for Q&A at the end.

Opening poll: could someone else take over? [00:02:36]

But I want to open with a poll. So if your website person disappeared tomorrow, could someone else take over?

So you want to drop a one, two, three or four in the comments. so one is you're very confident if that person leaves you've got good documentation you've got good cover you know what's going to happen dropping number two if you're fairly confident you can probably work something out if your website person leaves number three if you're not very confident at all and you know that there's a lot that depends on one person it's in one person's head and number four is honestly we'd be in trouble now This is public, so if you don't want to drop a number in the comment, that's fine.

But generally, when I've run these webinars in the past, people usually go in the middle, two and three. Yep, that's what I'm seeing, two, three, two, three-ish. So some people fairly confident, some people... you know, maybe not very common. That's probably the reason why you're on this webinar in the first place. So thanks, David, Elizabeth, and Gina for letting me know. That's roughly, as I say, in the middle is usually the response that we get. So keep your number in mind.

By the end of the session, I want you to know what could have changed to move, what you can change to move one step

About Wil Brown [00:03:57]

closer to getting that number one very confidently. Hi, I'm Wil Brown. I am a WordPress consultant and educator. I run Zero Point Development. I help businesses and organizations solve problems practically when their website has grown into something that they depend on for their inquiries, their bookings, their payments, or delivering services. Often the challenge is understanding how everything fits together, who's responsible and what happens when something changes or stops working. I also organize the WordPress Sydney and Elementor meetups. So making technical topics practical and understandable is a big part of what I do.

The important thing for today is that I'm going to keep this non-technical. You don't need to know how to develop a website, for instance. You just need to know enough to make sure your organization is not depending on knowledge that only one person possesses.

What would stop during a two-week absence? [00:04:58]

Let's imagine this very ordinary scenario. Your usual website person is unavailable for two weeks. So nothing dramatic has happened. Maybe they're sick or on unexpected leave or dealing with a family issue. You cannot call them and ask a question. What would stop or suddenly become difficult for your website? Would someone in your business still know how to deal with an urgent update? What happens if an inquiry form, a contact form stops sending? What if someone cannot access something they paid for, like a membership or a digital product? Who deals with hosting or an expired domain, for example?

Now, something topical has just come up in the news last week. There was vulnerabilities found in a programming language called PHP, and about eighty percent of web applications run on this programming language. And there was a big massive movement to try and get this upgraded to a version that had all the patches. And a lot of people were stuck out with that. A lot of people didn't know what to do because they heard it on the news and then they didn't understand where the business website was hosted or who to contact or how to fix it.

So this is a kind of real life scenario that happened just a couple of weeks ago. But for this one, My webinar last week looked at things like ownership, access, backups, and customer journeys. Those checks still matter, but today we're assuming that those systems exist, and we're asking a different question. Could another person actually use what you

The five parts of a workable handover [00:06:42]

have and keep moving on? So we're going to work through five parts of a workable handover. First, what does the website person actually do? second are where are the single person dependencies so what are the things only they can do third is who can step in if this person is not available fourth is what does the backup person actually need to know and fifth is how do you test whether actually this plan works or not So notice I've not started with a list of passwords, domains, or plugins.

Those things are part of the picture, but continuity is about responsibilities and knowledge as much as access to systems. So by the end of this talk, you should be able to choose one important website responsibility and create a simple cover path for it.

01 | What does your website person actually do? [00:07:43]

So let's start with the role itself. When someone says, Jane looks after the website or our developer handles that, it sounds like just one job, but usually that means a bundle of very different responsibilities. Before you can create cover, you need to know what that person actually does, especially the things that nobody really notices because they just normally happen in the background without a fuss. So looking after a website is really just editing pages. There's routine work, there's updates, there's publishing, there's usually adding users or registration, there's checking forms, renewing services, and dealing with small requests. Then there's problem solving.

So something behaves strangely and integration fails. Maybe contacts are not being sent to your CRM or your email system. Maybe a supplier needs to be contacted or a customer reports an unusual behavior when they're trying to log into a dashboard or portal. And then there are decisions and relationships. Somebody knows which changes are safe and which things should not be touched without testing and which supplier or specialist to call if that particular thing goes wrong. So those last two categories are where continuity problems can often hide.

The visible task may be documented, but the judgment around whether or not to apply it or do it or reach out to an expert is not. As you listen, think about the person who knows all those little quirks of your own website. Here's a quick way to expose that hidden knowledge. Can you answer the following questions? What does your website person do regularly?

The hidden knowledge test [00:09:42]

What's their day-to-day job? What do they know how to do that nobody else does? And who do they contact when something goes wrong? And what decisions are they authorized to make? What can they actually do without by themselves? And who do they need to contact in order to make certain changes? So these answers don't have to be technical. So for example, just say every Monday, they check the new member application has reached your CRM. Or if the payment system has reported any errors, then they contact the developer before changing anything. That's just operational knowledge. It's not technical knowledge.

A useful handover starts by making that knowledge visible. You can't provide cover for work that nobody has recognised as a responsibility. So this is the central idea for today. Having the passwords is not the same as being able to take over. A lot of people put a lot of emphasis on passwords. They've got a password sheet, they've got a password manager. If you know the password, the admin password, then you have this false sense of security that you'll be able to deal with problems.

Having the passwords is not the same as taking over [00:10:57]

But that's really the case. You can give me a login to a system I have never seen before, and I still may not know what the business expects me to do, what other systems it affects, or what normal looks like, or even what I'm allowed to change. The better question is, could somebody else understand the job and not just log in? Now, your first practical check is to list the recurring tasks and the usual expectations that currently belong to that one person. This person looks after your website.

Start with the things that would have business consequences if they stopped working for a week or two. So think about a contact form that suddenly doesn't submit or nobody receives or doesn't A renewal

system that isn't working and you're now getting complaints from people that they can't renew and get access to their portal. Things like that. What are the business critical things? A booking system, for example. If it went down, it'd have a direct impact on your job. And then think about how that website person looks after that particular thing. What do they do?

02 | Where are the single-person dependencies? [00:12:07]

So once you know what the work is, what they're actually doing and maintaining and updating, look for the points where everything depends on that one person. So sometimes that dependency is obvious. Maybe only one person has an account, but more often it's less visible. Several people may technically have access, but maybe only one person knows what to do with it. And that's quite common in a small business. you might find that several people have an administrator login, but only Jane in marketing knows how to log in and fix X, Y, Z. So the risk is often not the actual system itself.

It's the knowledge that lives up in people's brains. It's the judgment about how to apply, where to apply, what to apply, or it's the relationship wrapped around that particular system.

Knowledge, access and relationships [00:13:02]

I think these dependencies are grouped into kind of three broad places. First is knowledge and decision. So one person understands why something was built in a particular way. Maybe they were the person who built it in the first place. They know what they can safely change, or they know how an exception is handled. So if something doesn't follow the rules, oh yeah, I know why that is the case. I will deal with it. Second is access. So one person can log in, recover an account, approve a change, or receive or generate a security code on their phone, for example.

This one happens to me quite a lot. So for example, a client recently-ish gave me a login URL. It was a username and password to the hosting account. Everything looks good when I try and log in until after I put in my username and password, it asks me for a security code, which has just been text to Bob, the business owner. And of course, Bob's on holiday or out on a job. So I've got the correct login details, but I still can't get in. That's not a password problem. That's a business continuity problem.

And these things only really expose when urgent things need fixing. Third is relationships. So one person knows which hosting account or contact is useful, which developer built that particular integration, or who at the software company can resolve a billing issue, for example. Now, a mature continuity plan does not necessarily duplicate every single skill. It just makes sure that there's a path to move forward when that usual person is absent. So it's not all relying just on this one person.

We've got other contacts and other people who reasonably know what to do or who to contact to then progress that issue to the next step. These are four warning signs that I would pay attention to. Number one, first is a task simply stops when one person is away. So that one person, Bob or Jane, has taken a holiday, they're on long-term leave, they've had sick, they've had to deal with a family emergency, and something on the website stops working. Okay,

Four warning signs [00:15:24]

that's a big warning sign that that knowledge, that operational risk is just with that one person. Number two is nobody else knows what normal looks like. So they can't tell whether something is working correctly or not. And this one actually happens to be an awful lot as well. So for example, a business owner calls me up, they explain something that's happening on the website and then ask Will, is that normal? Now I'm thinking, How do I know? How could I know? Sometimes it's obvious because I understand WordPress and websites and how they work.

But often, very often, we're talking about business rules or processes that are very specific to that organization. So I can tell you what the technology is doing, but unless someone has explained what it's supposed to be doing, what's normal, I can't always tell you whether it's right or wrong. So that's a good warning sign. Nobody knows what normal actually means. It's not been defined. Third, a workaround exists only in someone's head. Again, that's, you know, Jane in marketing, it's Bob, the software developer, that sort of stuff. And number four, again, the default answer to any question is, oh, just go and ask Bob.

He fixed that last time. Or you still have to send an email to Jane. Jane knows how to deal with that stuff, okay? So those are kind of four warning signs that when these things happen in your business, more often than not, you've got a continuity problem because just this one person has that knowledge. Now, none of these automatically mean that you have a disaster waiting to happen. They're just warning signs that your organization has a single person dependency. And this can happen with excellent staff and excellent suppliers.

In fact, competent people often accumulate knowledge precisely because everyone trusts them to deal with these things. So the goal isn't to lose expertise. It's not to blame anyone. It's not to blame anyone who has that particular knowledge. It's just to make sure that the business itself can keep moving when that particular person isn't available.

03 | Who steps in? [00:17:47]

So now we get practical. If the usual person is not there, who steps in? So for every website responsibility that really matters, you want to have some sort of backup path. Now, you can start simple. Start with the main thing that your business website deals with. So is it bookings? Is it a contact form that people will contact you they can get back on? Is it a membership portal? Those sort of things. What's the critical path? for your business website, do you have a backup path for if that website person is not available?

So it doesn't mean that two people must be equally experts. It just means that the business has decided who takes that next action rather than discovering an answer during an incident that happens. So here's a simple model. It's primary, backup, and escalation. The primary person does the normal work. That might be the communications manager, the business owner, a freelancer, your developer, or an agency. The backup person keeps things moving when the primary person is unavailable.

Primary, backup and escalation [00:19:03]

And the escalation contact is the person or supplier that that backup person goes to when the issue is outside their ability or their authority. So that last part really does matter. A backup person who doesn't know the answer shouldn't be forced to improvise on a production website. You shouldn't be forced to update something or delete something or remove a user or add a user. A good cover plan gives them permission to say, okay, this is beyond my role. I need to call this person to get them on the issue.

And that could be a developer or an agency, a supplier, a web hosting company, for example. But it's having that next step of who to contact and where to escalate that issue. That's the good continuity part of this. So here's what a simple cover map can look like. These are examples, they're not rules. The right person depends, of course, on your organisation. So for website content, the backup might be someone in operations or communications.

For a customer form, a contact form issue, an admin person might be able to confirm the problem, check the normal process, and then maybe escalate to the developer, or maybe if it's a problem with your CRM, maybe contact them,

A simple cover map [00:20:29]

a support person within your CRM. For payments or hosting, For things like perhaps a card has expired, for example, finance or admin may own the account, even though a technical supplier kind of handles the fix. So the important part is that responsibility has been decided before someone is absent and it's written down and stored somewhere that if something happens, then you can look it up and say, this is the next person that I need to get in contact with. So again, you're not trying to duplicate the whole org chart.

I'm not saying you need to have a second person for everything that everyone does regarding the website. But start with the responsibilities that have real customer revenue or operational consequences if they were to stop tomorrow. So a backup person doesn't need to be the expert. So if you have a website person, a technical person that you go to, an agency or whatever, that's good. They are the experts and that's great. You don't need to have, or the backup person doesn't need to have that entire expertise. They just need to know enough context to keep that organization safe and moving on.

So that means knowing where to look, what they can safely do, what they should not change, and who to call when they reach that kind of edge of their responsibility. I don't know what else to do. Okay, I know that I need to contact Jane in Accounts or Bob at the web hosting company, that sort of thing. So this is actually a better continuity model than just expecting someone to improvise. So a good instruction might say, check these things out first. If the result is X, then follow this process. Do not change Y without testing.

What a backup person needs [00:22:18]

If this continues, then contact this next person and give them these details. That's much, much more useful than a folder full of passwords with no context. So there are four basic things I'd want a backup person to have. One is they know where the instructions live. Number two is they have access, the role actually requires. And number three is they understand the boundaries, including what not to change, what they can change, what they can't change. So we're talking about production, a live website here. And number four, they know when and who to escalate. So notice that access comes second and not first.

Many business owners fixate on this username and passwords, but knowing where to go and where the instructions on handling issues lives is much, much more important as a first step, because a lot of businesses I deal with, they don't have that first step. They don't have a knowledge base. somewhere where it's recorded for the business, and that when people move around, this new person or backup person can go to to see what the process is and understand the context.

So if someone has a full admin access, but no idea what the system does, that can actually create more risk than less because they might be tempted to go in and do things like I work with WordPress so they might disable some plugins that could have a knock-on effect for other things and that could end up with real risk for the website not working for everyone. So the goal here is appropriate access plus enough context to use it responsibly. So step four,

04 | Write a useful handover [00:24:14]

now we come to the actual handover itself. So could another competent person understand how the work operates? Good handovers explain the job, not just the login. You give them enough context for someone to recognize the task, follow the normal process, normal process that's documented, and know when the situation is no longer normal. So for an important recurring task, you can document it in four different parts. I love my four steps. Step one is where and when. So what is the task? It could be filling out a contact form. The contact form then goes to an email inbox and then somebody responds to them.

So what is the task?

Four parts of a task record [00:25:02]

What are the triggers? And how often does that happen? So the what and when. Number two is where and how. So which systems are involved and what is that normal process? It needs to be documented so when something goes wrong or something looks different, then you can look at documentation and say it should do one, two, three, four. And if it does one, two, and five, or one, two, and we don't know, then that is not normal. Number three is why an expectation, exceptions, sorry. So why does the process work this way? And what unusual cases should somebody watch for?

Now, in lots of websites that I get involved with, there's a lot of processes that work automatically, but there are usually a few things that have been custom built for that particular website. So there might be a booking system, but if somebody books on a Friday afternoon, then there's, For example, the system may say there's no availability. You need to book for a Monday or a Tuesday. So there's usual exceptions in business logic when I come to look at a website.

So definitely how things work, why they work that way, but also what are those little exceptions, that business logic that makes it a bit quirky for someone outside or backup person to understand how this actually works. And then number four is escalation. So when should the person stop and get help and who is that next person to contact. So you don't need a forty page manual for this simple website task. A short, clear record with these four paths is often enough for a competent person to pick up a thread if they are not that primary web developer, web support person.

The most valuable documentation is usually stuff another competent person could not guess. So we're talking about things like document triggers and dependencies.

Document triggers, manual steps and exceptions [00:27:06]

So what happens before this task and then what happens after it? I love to give an example of a contact form, a simple website contact form. People just assume there's a contact form, it gets filled in and someone gets notified and there you've got a potential job coming in. But actually, contact forms are a little bit more complicated than that. So first of all, when someone visits the website, they need to be able to find the contact page with the form on it.

Second, they need to be able to fill it in and submit it, and it needs to submit properly. then in the background that system then needs to send those details to a crm or an inbox and then lastly someone needs to be able to pick that item up from a crm or inbox so there's lots of processes it's not just submitting the contact form there's lots of triggers and dependencies so example if the contact form would not submit.

If there's a problem, a technical problem with the actual form itself, then that whole process, you need to know what happens before and what happens afterwards. So document triggers and dependencies. Next thing is document integrations and manual steps. So perhaps the form sends data to the CRM, as we say, then somebody manually approves the customer and only then does an account get created. So there might be manual step ins you see.

Another example, just say you're running WooCommerce or a shopping cart or Shopify, an order, maybe order data is exported every Monday manually by this web person at nine AM and then forwarded to your sales department. So they can then include it in reports. That might be a manual process. So they need to be documented as well. and document exceptions and workarounds. So maybe one particular plugin or extension cannot be updated until another system has been tested. This is very common around booking systems. It's very common around payment systems as well.

You have these dependencies, so you might have your shopping cart system, and then you might have your payment system. And you want to make sure that both of those are always talking to each other. So when you get an update for one that comes out, you want to make sure that one cannot be updated and it's not going to affect the other one. So things like that. Maybe a particular supplier needs a specific PIN number before they will help. And this is very common as well. So you might know the hosting companies who host your website.

You might be able to find and give them a call, but they might ask for your support PIN number, which maybe your website person only knows where to get or where it's documented. So that's the kind of documentation I'm talking about that's kept as business logic with the right person to get access to the right documentation. Now, a new developer, a new web person doesn't need a page explaining what WordPress is, but they do need to know why this organization's unusual workflow behaves the way it

does so they can get up to speed on that. So we've talked about what to document,

Give your handover records a home [00:30:28]

but we should talk about where does all this information actually live? Because I do get that question a lot. I get lots of people say, well, you tell me to document all this stuff, but where do I put it? The good news is you probably don't need to buy another system. If your business already uses, say, Google Workspace or Microsoft, you've got a perfectly reasonable place to start. Create a shared folder for your website documentation. Keep your handover instructions, your supplier contacts, important decisions, and all the current work should live in there, in a folder there.

But make sure that the folder belongs to the business and isn't just sitting in Bob or Jane's personal Google Drive. Things like passwords. Again, I said don't fixate on passwords, but some people need to know where passwords live. For passwords and recovery codes or multi-factor, 2FA kind of codes, use a proper business password manager software, such as Bitwarden or 1Password. Don't put all your passwords in the handover document in plain text. We don't want to do that. We want to make sure things are secure. So I use Bitwarden for all my personal and business stuff. I'm not affiliated with them or anything.

I just find that that's the useful one for me. I do have a paid account. There's free and paid. The paid account allows me to share my passwords for systems with somebody else, another email address, and they can use that to log in, but they don't know what the password is. So anyway, have a look at different options for those. Bitwarden and 1Password are usually the two big ones there. And for the work that's currently happening, use whatever task management system that your organization already uses.

That might be Trello, Asana, ClickUp, Notion, Microsoft Planner, whatever it is that your company already uses, just build that into your systems. We're not trying to buy or use a big system just for all this. We're trying to use whatever you've got at hand, whatever your business already uses. So you don't need all three systems to be complicated or heavily integrated.

You just need a clear place, a clear one place for each type of information, whether that's the documentation and processes, whether it's a task, whether it's the passwords and pin numbers, and the right way for people to access it, business people to access it. And here's the important part. Make sure your backup person knows where these records are and has the permission to open them. There's nothing worse than having a document somewhere that tells you all this stuff, but this backup person doesn't have access to it. So make sure that they have the proper access.

And equally, make sure that everyone in the business doesn't automatically have access to this one folder. You don't want everyone being able to have a look at pin numbers and passwords or processes for things like that. Just make sure the appropriate people have access to that important folder. Otherwise, all that documentation, all that stuff that you've created is just another single person dependency. It's just you that has access to it.

05 | Test whether the plan works [00:33:41]

Which brings us to our final section. How do we test whether someone else can actually take over this critical website function? So documentation can look excellent and still fall flat on its face when someone actually needs it. So the fifth part is testing the plan. So please don't wait for a real absence to discover that the backup person can't find that document or doesn't have access to the folder or can't tell which instruction applies for a particular process. A short takeover access is one of the simplest ways to test this.

The takeover exercise: a customer cannot access a course [00:34:21]

So let's work through a simple scenario. Your usual website person is unavailable for whatever reason. It's nine, fifteen in the morning and a customer sends an email saying, I paid for the course yesterday and I've tried accessing it several times and again this morning, I still can't access it. So nothing here is necessarily a technical emergency. So the website's still up and running, you still got your critical paths, but it's a real customer problem and somebody needs to own that. So imagine this happening in your own organization. Do not jump straight to the fix.

First, ask yourself whether the replacement person knows how to move that issue through the business. Do they know what the process is or where to get the process? How to read through the process? Do they understand the process? Do they know which next step to take in resolving this issue? So again, they don't need to resolve it themselves. It's just to move the step a bit more forward towards a fix. So start with these questions. Who owns the problem? Is it support? Is it admin? Is it finance? Is it the website person? Or is it nobody in particular? That's the problem.

Secondly, where do they look first? So is there an instruction? Is there a task history? Is there an order record? Is there a known troubleshooting problem? Thirdly is, what are they authorized to do? So this person, can they resend access, for example? Can they change an account password? Can they issue a refund?

Or can they only gather information and then know which next step to take? and then of course the last one is when did it escalate and to whom so i'm sure this has happened in your organization at some point you've probably been in an organization before where something's happened you know and there's been a bit of a fuddle and they don't know what to do The point is not to solve this fictional access problem. The point is to see whether the organization has a path from customer report issue to responsible action. So takeover exercise tends to expose these gaps very quickly.

Sometimes there's no clear owner, so the issue simply waits for the usual person to return. Sometimes the backup person technically

What the exercise exposes [00:36:56]

has access, but they can't find instructions or understand what the process even means. And sometimes everyone gets the same point and then discovers nobody knows who to call or what information the supplier will actually need if you do get through to them. So again, we talked previously about pin numbers. A lot of web hosts won't do anything. They won't even raise a support issue if you can't authorize yourself with them by giving them a pin number. So this is why I prefer testing over simply just asking, do we have documentation?

If a continuity plan is useful only if another person can use it, that's why we need continuity. So there's an important difference between planned departure and a sudden absence so when you have notice then use it so if a person gives you notice that they're going to do something go on holiday or leave or extended leave that sort of stuff transfer responsibilities access current

Planned departure versus sudden absence [00:38:05]

work supplier relationships and the reason behind important decisions then let the replacement person try that process while the original person is still available. You know, that's kind of why we have notice periods in place, right? So that another person that comes in can get up to speed while the regular person is still there. So they can shadow them and get used to what means. So if you do have a notice period, then use it and put your plan into place. With no notice periods, the priority is, of course, different. Stabilize what's stopped. So if a person does unexpectedly leave, then stabilize what's stopped.

Find what records already exist for this person, what they do. Identify the known suppliers and avoid unnecessary changes while you establish what's going on. So if a person does suddenly leave, you don't want people to be poking around the system too much. You want to stop. Okay, what does this person do? And what's critical to the business? Where's the information that they do? Okay, right. Who's responsible for this? And who do I need to contact? That sort of stuff. But in both situations, you need to prioritize customer facing and revenue critical work first and record what you learn as you go along.

So a sudden absence can actually reveal what your next handover documentation needs to contain. It's happened to every business. Somebody leaves and nobody knows what to do. And that's the red flag. And that's when you need to step and say, right, we need to have this continuity process in place. So I don't want to leave you here thinking you need to document your entire website this afternoon. That's not what I'm talking about. But I would like you, I was going to say need, I would like you to choose one important responsibility this week. So we're talking about small businesses here.

It could be something as small as your contact form. Or it could be as large as a WooCommerce or product catalog purchasing. It could be your Shopify checkout process.

Choose one responsibility this week [00:40:17]

But choose one important responsibility for this week. Step back and ask yourself, who normally handles this? It could be yourself or it could be a website person or Jane and Accounts, something like that. So who normally handles it? And then think about if they step away or something happens, who covers for them? Who steps in? Now, I know there'll be a lot of single freelancers and solopreneurs and solo business owners here. And I'm talking about all this stuff and you're thinking, well, that's me, that's me, that's me. And that's understandable. But what does happen if you're not you, if you're not there?

Something happens, there's a family crisis. who can cover for you. So that could be a PA, for example, a virtual staff. So that's the next person. Find where the process is documented and make sure that the backup person has access to whatever they need. then ask whether that person could actually follow that plan today and who they would need to escalate it to if your process failed. And that's quite common if you have VAs, virtual people, virtual assistants that are working for you. All is good and well while you're there, but what if you're not there and something goes wrong?

You might be able to access the documentation, but can they then talk to the next person to escalate that to get it fixed. That could be the web hosting, it could be your bank, for example, or your payment system, that sort of thing. So do that for one responsibility and improve on what you find, then move to the next one. That gives you a practical continuity plan without turning this into a giant documentation project that never gets finished.

The Website Handover & Cover Plan resource [00:42:07]

So I've put these ideas into a website handover and cover plan that you can use with your team or your website provider. I will copy that link and put it in the comments. Yeah, so you can download that now. I'll also send an email out to everyone on the webinar and I'll have a link to this document as well. And the QR code there on the screen goes to the documents. You can get the document either way. There's no sign up. You just download it.

It's designed to help you map responsibilities, name the backup person, record the escalation contact, document the critical task, and run the takeover test just as we walk through. So you can use that QR code or the small business page that's on my website to get to this. The aim is not to be perfect documentation. The aim is just to make it possible for another competent person to step into the business and move it forward when that usual person is not available.

If the exercise uncovers bigger ownership, technical or strategy problems, that's also the sort of work that I help organizations untangle through my zero point development.

Questions and closing remarks [00:43:16]

So that's the formal part of the session over. I'm happy to take some questions. A good place to start is to think about your own website and ask where would another person get stuck first. If you do have any questions for me, then drop them in the chat just now. If you don't want to do that publicly, that's fine. You can reach out to me afterwards. I'll send an email out and you can reply to that. If you've got any follow-up questions, I'm happy to help you or point you in the right direction. When answering though, keep...

Keep the description back to those three basic ideas. Responsibility, usable information, and clear escalation path. That's kind of the things I've covered for this webinar today. So again, any questions, please drop them down. You can get in contact with me here at my Zero Point Development website. I've got a special page for this Small Business Month. It's zeropointdevelopment.com.sbm and you get all my replays there. What I'll do with this webinar, I'll put it up later on today and you'll be able to rewatch that and there'll be links to the resources and things as well.

I'll give another... few seconds if anyone's got any questions put them in the comments otherwise it's coming up to the top of the hour oh quarter to that's not too bad that's great gina thank you david thank you so much okay if there's no questions then i'll just wrap this for today thank you so much for joining

me i hopefully you found something useful with this hopefully you can have a look at your website have a look at that critical path and just think about continuity what happens if that person Can't bear, it's absent, it's on leave.

And what do we need to do to get that up and running, keep functioning? Thank you, EC. Okay, everyone, have a great day. Thank you so much and see you next time. Bye.