This is a transcript for the VIDEO

Interview with Michael Oldham

<https://web.microsoftstream.com/video/1fa60edd-5f46-48b3-97e0-b326ccb9e430>

PARTICIPANTS

**Michael Oldham**, consultant/developer, Telstra Purple

**Adam Maiorana**, RMIT student

**Angelo Lapuz**, RMIT student

**Grace Field**, RMIT student

**Taylor Thurley**, RMIT student

**William Currie**, RMIT student

|  |  |
| --- | --- |
|  |  |
|  |  |

**[Beginning of recorded material]**

**Adam:** Hi Michael could you please tell us about your IT work and what exactly it is you do?

**Michael:**

OK, so right now I work for Telstra.

OK, a company called an arm of

Telstra called Telstra Purple.

I don't know you've heard of them.

Uhm there probably.

They've got the most consultants,

actually as a consulting arm in Australia.

Believe it or not,

but not too many people know about them,

you know, so so I I basically

work as a consultant for them.

And what I do is I come.

I designed contact Centers for that,

you know and deployed them.

So basically you this is

for external customers,

not internal customers to Telstra.

So for external customers.

And so I sort of like.

Go through the requirements,

uh, that they need for this

contact center alright?

And then I put in the from the requirements?

Uh, we go into the the design, build

design other thing and then we build it.

Then we test it.

And then the support side.

I don't do.

We hand over to a support group

within Telstra, you know.

So is is she typical waterfall model?

Is what we do.

In terms of the project.

Uh, so personally, uh,

I'm involved in all phases.

OK, and uh, so in in some respects

of a consultant slash developer.

So I also write the code. OK, So what?

What does a contact center do?

You might not.

Understand what a contact center do.

The simplest thing is you know

when you call RACV 1/3 RCV number

and you're broken down and you've

got to go through menu system and

then you get to an operator that

says what's your member number.

Things like that.

That that system I actually put in for RACV.

So when you call Irish TV that

is a Telstra system.

We just went live about a

month ago with that one.

So that's one angle.

The other angle is dumb people chat,

so if you're on a web page and

you want to chat to somebody

and you're chatting and then

you will connect to an agent,

so that's another piece.

That's another.

What they call channel that you can get in.

To the contact sensor,

so they're the most common one chat web.

Phone call and the third

one is outbound dialer,

so you know you're sitting down for

dinner at 6:00 o'clock and you get hit

with somebody trying to sell you something.

So there is a systems I put in to know,

uh?

So let me think.

So this job right now is just Telstra.

I mean it's Telstra Swiss Australian wide.

So it's not sad it's not outside

of Telstra if you know what I mean.

OK, so.

I think the other thing to know is.

And with this particular job,

there's not a lot of travel.

You don't do a lot of

travel because you're doing

everything just like we're doing right now,

right with a teams call,

we do all three teams.

Oftimes and the system that we put

it in is a cloud based system.

So since the system is in the

cloud and we do things teams,

everything's done remotely.

So that's kind of like how things are

being done in itnow is all remotely and.

Is absolutely, really no need to go into

an office anymore 'cause it brings him out.

You know, I think the biggest thing is.

Requirements A requirements is typically

you need to be in front of the customer.

Right? Uhm,

but with kovid that's made that impossible.

We have to do that remote too.

Now you know, and that that causes

problems 'cause getting requirements.

A lot of times you have to draw figures.

You have to explain things

to your clients right?

'cause they don't understand even

though there in my case in the vertical.

I mean she's contact centers.

I'm talking to supervisors talking to agents.

They're not very technical.

In fact, 1/2 of them don't even

know where the keyboard is.

No, I won't go to that level,

but the you cannot assume

they know anything you know,

and then you you range up to the developer,

the business, you know you.

You're integrating with the for reports.

For example, we have to put stuff

into the business intelligence platform.

So now I'm talking to somebody who's it.

By expert, you know,

data analytics experts.

So the range of clients is is from

pretty well technically ground 02.

Way up there,

so you've gotta kind of go up and down.

In terms of your.

When you're talking to people,

what level do you need to talk

to them about that make sense?

Yeah, so that that's quite difficult,

you know.

Ah,

just want a soft skill perspective,

and it's even harder remotely

because a lot of this stuff you get

when you talk to people you get the

body language and you can see, oh,

OK, you're talking this person,

but they don't really understand because

you can tell from the body language,

right?

When you do it remotely,

you don't get the body language, yeah?

So right now I'm in a conference

with you guys.

I only see you Adam.

I don't see anyone else,

so I can't see the face of anyone else, so

I can't see whether or not what I'm saying.

Is being comprehensive.

Well, you know, yeah.

And that makes it very, very hard.

Very hard 'cause you get clues

from that as to.

Uh, are you going down the right

path and and and?

And if you know you jump to a white

board and then you explain on a whiteboard,

And people who are not technical

whiteboards are perfect.

To do that with and people who.

English is not their first language, right?

Uh, whiteboards are great for that,

too, you know.

Uh,

for example,

I want to design a system

in South Korea and

I did through whiteboards 'cause they

couldn't speak much English. Yeah, yeah.

So you can imagine doing it remotely now.

It's extremely difficult without white board.

Yeah, yeah. So I'm looking for something.

Well, mimic a whiteboard,

but it's it's kind of.

Yeah, there's not a lot out there to do.

**Adam:** What about a digital whiteboard? What are they like?

**Michael:** It's too jerky. Yeah it's not as

it's not as fluid as a whiteboard.

It's not quite there yet, but anyhow, uhm.

So I've kind of answered

your second question there.

The industry are working right, yeah?

**Adam:** what aspects of your work

do you spend the most time on?

**Michael:** Yeah yeah, OK I I would say.

Uhm, when you're when you're

writing software right,

which you guys I believe are doing right?

Yeah, I think most of I don't think all

of us are doing the programming thing,

but if you have a staff,

yeah, OK, OK.

So one of the things you spend

most of your time on and it kinda.

At first you wonder why,

but if you if you're trying

to put a program together that

does a certain activity, right?

Uhm, for example,

I'm doing 1 right now with payment system.

So you call into the.

You call into this system and you can

make a payment through your phone,

And you could say you put your

credit card number and you put

your CVC number in and you go.

Yeah go pay that bill right?

So doing that is just doing

that flow is is fairly trivial.

You can get it done like that right?

Where all your time spent is

in the exceptions, right?

You gotta think of all the

exceptions and how are you going

to recover from that exception?

You know, like you call it a try,

catch right.

Yeah, so it's it's similar to that.

So if I'm trying to pay a bill

and for example it times out,

the system is not responding.

I've gotta I gotta figure out how

I'm gonna capture that in my program

such that it comes back to the user

in a way that is is pleasant rather

than just saying system error.

You know?

Yeah,

that's a lot of the programming.

You you could spend,

I would say at least 60% in that area.

Just looking at exceptions.

And if you don't do that what will

happen is if you don't invest the

time then what will happen is when

they go live then it's worse right?

So when you're developing your

own the system yourself and

you're in control of the system,

no one else is on it so you can control

it quite well when it goes live.

Guess what everybody is

using is chaos out there.

And people will ring in saying I

saw this thing and it was not.

It wasn't working right.

It's it's said this or said that now

you've got chaos going on and you're

trying to troubleshoot it so it pays

dividends to do all your exception

testing at the very beginning.

And I get it right.

But a lot of the time you're

under time pressure.

**Adam:** Yeah, I was gonna say oh sorry to interrupt.

I was going to say do you

spend a lot of time?

How much time is generally spent?

Figuring out those exceptions and like

testing and all that sort of stuff.

But within the within the deadline, how?

How much time do you

generally give to testing?

**Michael:** Well, yeah, that's a good question.

UM,

when you're when you're doing these projects,

have a project manager right?

And a project managers Darryl level

of understanding it's not that high,

but they think they do know at

all and they turn into architects

off time but not architect right?

They just they just naturally turning

architects. I don't know why.

And yeah, uhm,

so they set the time frames and they go OK,

it's gonna take two weeks,

three weeks to build it and a week

to test it and they get it all wrong

and you fighting with them constantly.

Because they've got dates to

me and you've gotta goo.

You've gotta do good quality right?

So good quality and dates don't mix.

Yeah, you know. So what it is is a triangle.

I don't have teachers triangle like time

quality and I figured they were whatever.

When you change your gonna

affect the other one, yeah.

So if you decrease the time.

You gonna sorry your increase

quality increase the time.

Obviously if you decrease

the time that you've got,

the quality is going to go down right?

So it's it's a tradeoff.

That come in the in the in the

real world you've gotta like.

When you can't wait till it's perfect right?

But then you can't wait.

You've got to have a certain feel for where.

OK, I think we're right to go, yeah?

'cause like my case,

if I if he didn't workout and you

called our CB now you wouldn't get

somebody to come and get your car fixed.

'cause Michael decided I didn't need

to test it that bad. Yeah yeah.

So this year, your first string of it,

I guess that kind of that leads

into the next question, uh,

do you find that aspect of your work

the most challenging their timeline?

Yeah, always always.

So in this business you you

worked like a dog.

I'm sometimes I think working for McDonald's

would be better turn around are away, yeah?

The only right you know,

no yeah.

So I T is not exactly a

nine to five job. It depends

which vertical urine, right?

If you're doing a developing is

is is definitely not nine to five

if you wanna get in that area.

I know it's fun though that's a nice thing,

but uh, and I think. Uh,

the other challenging piece of the work is,

uh, the clients to write,

so one of the things that.

You have to be. Very stringent on

is when you get the requirements OK,

it may sound this may sound simple right

on the outside you get a requirement.

You get the requirements signed off

and then you start designing right?

That sounds pretty simple, doesn't it?

Right then you start designing.

You get the design signed off and

then you start building and then

you get the build signed off and

then you start testing and then

you get the testing signed off.

Then you go live.

Pretty simple process, right?

You'd be amazed how many times

does she go through this?

They've forgotten something.

So when we come to do the testing,

they thought oh we didn't do this test.

And then you find out that

that was not a requirement.

But your requirement is signed off.

Right,

so now you've got this conflict resolution.

You've got to go through.

You go OK if you want me to put that

feature in which you forgot at the sign off,

it's going to cost you this many dollars.

Alright, now you've got friction.

So now the project at the

start is honeymoon period.

Everyone's happy,

everyone loves each other,

and that nearly everyone really

hates each other.

Yeah, it happens all the time

because they rushed the front bit.

Don't get the requirements right,

and at the end it it.

It gets the problem snowballs if you follow.

If you don't get in the requirements

and you catch it at the testing

you've got that huge amount of time

you gotta go back and do it again.

so you've always got to invest

as much as possible on the at the

front end to try and stop it.

Going down the bottom, for example,

I've just got one where.

We got the requirements we got all

signed off. We did the testing.

And we found out there was those cases

in there. They went live as well.

They went testing.

They went live, but they fit.

Missed a few cases. A few scenarios and dumb.

It means we have to redesign it again,

yeah, but it costs money. To do that.

You know it's not all matter.

OK, I'll go off and do it.

Cost a lot of money.

So now you've got friction to handle,

so that's probably a challenging part.

Of the job, you know when it comes to money,

it's always challenging.

**Adam:** can you share an example of the

work you do that best captures

the essence of the IP industry?

**Michael:** So you'll find that. Uh.

Small projects you can work

on your own right? And.

Uh, working on your own?

In some cases it's easy.

It's easier, right?

'cause you don't forget to

tell yourself things right.

It's you booking in a team and

team dynamics is difficult.

It's very difficult, uh,

you got different people.

I'll give an example.

Uh, I will work in the center I'm

putting in right now where there's

a team of about eight of us.

Right, and we all take different parts of it.

And then we work on that.

But we have to be.

We have to sort of like collaborate,

such that if if for example,

my peace interfaces with their peace,

they need to know what

what the parameters are.

They gonna send them.

Conversely, I need to know where

they're gonna send me back, right?

So one of the couple of things

problems you got in teams is.

Some people right and you may

find this in your team right?

Some people are very good at communicating,

It's a it's their primary language, right?

Some people you know,

it's not their primary language, right?

It's it's the secondary language.

we got a guy from China in our team,

He is his conversation skills

are out of 10 about 2.

Is technical skills out of a tan or by 10?

Yeah, so you got that with when

you're working with a team,

everyone is different,

Some people are not very good technically.

Some people are very good at

doing requirements.

So what you what you tend to do is

you try ask Team figure out where

your strengths and weaknesses are

and then leverage that right.

So if someone in your team is really

good at collecting requirements

and you say you've got a project

as a team to build something,

get that person to collect requirements.

If you've got somebody who's

good at coding get that person,

do the coding that's.

Oh, I think we've lost Jeff.

Yeah, no, I'm not you back you back yeah?

So what you need to do is find

out like if you guys are doing

projects together as a team.

Find out what your your strengths and

your weaknesses are and you've got

probably deadlines to to to meet, right?

So so leverage that learn from each other.

Leverage that and and then you'll be able

to get the thing in in time quicker, right?

And then I put on the converse side of that.

If you're trying to learn it.

OK, I'll do that.

'cause I know nothing about

it and I wanna learn it.

Then you times gonna go blow right out right?

It's like you gotta you gotta

change it dynamically we wanna do

so but you're right I think UM.

These days, uh teams is a big one, yeah?

And knowing how to work in a team.

Yeah, yeah, definitely. Alright.

That was really good. Thanks, Michael.

You're welcome, hopefully it makes sense.

Yeah, I think we covered just about

everything, so that was great. Awesome.

Alright. OK, then thank you very much.

Thank you very much.

You're welcome thanks.

**[Ending of recorded material]**