# MAIN PRESENTATION

Good Morning Everyone. My name is Nathan Bligh and I'm a professional tester. When I'm not testing I'm running a company called Informatech, made up of other professional testers. We have a stand outside, come say hi. I promise we're friendly.

INTERNET SLIDE - So, megaprojects. How cool does that sound?! It's not just a Job it's a MEGA job. Don't you think it just oozes cool? Don't worry about Information Technology, that's boring. Megaprojects is where it is at. You know how you're out somewhere and you meet someone, and they ask you what you do for work and you reply "I work in IT" – you can almost watch their eyes glaze over in boredom in real time. Like at that point in time they would rather be doing anything else in the world other than possibly be drawn in to a discussion about IT. I can already hear the excuses, - "oh sorry you'll have to excuse me, I have a PowerPoint presentation I am really looking forward to". Let's just ignore the fact that without IT they wouldn't have the Internet, YouTube, Reddit or FaceBook!   But replace IT with "megaprojects" and you'll get some interest. Even if it is because they've never heard the term megaproject before. I sure have said the word megaproject a lot already hey. Bonus prize for the first person who tells me afterwards how many times I said it during this presentation.

ELEPHANT SLIDE - As you can probably tell I think megaprojects are pretty cool. So, when it was announced last year that the next ANZTB conference would be in Canberra I knew straight away that I wanted to present, and I wanted to talk about megaprojects. I wrote my abstract and submitted it early and was accepted as a speaker. And then things got busy. I mean, I have a pretty busy life in general – I'm always taking on more than I should. Running a business, working as a Test Manager on massive projects, completing uni degrees, renovating our house, building us a new house. Oh, and I have 3 kids under the age of 5 – they distract me every so often.   Have you ever heard the saying "how do you eat an elephant"? Most people will respond with "one bite at a time". But I work off a different philosophy. My approach to eating the elephant is to bite off as much as possible and chew like crazy

OVERWORKED SLIDE - Anyway, days, weeks and months rolled past with this presentation sitting as a line item on my to-do list, and I kept thinking to myself "yeah. I really need to make a start on my ANZTB conference presentation" and then proceeded to do all those other things in my busy life that I just mentioned. Before I knew it, I'm less than a month away from the conference and I literally haven't written a single word of my presentation. And within those 30 days we are buying a house, selling our house, going on a holiday, and I have to prepare deliverables for a major project going for government funding approval. So pretty much par for the course in my life.

And now here we are. That month has passed and, as I figure it, either one of two things happened.

1. I managed to write my presentation and I'm about to wow you with all the knowledge of how to perfectly deliver the testing for a megaproject.
2. OR. I didn't get around to it and I'm about to try and fake a 40-minute-long presentation about testing, to a group of testers. In which case I would expect this presentation to crash and burn epically. Someone should record it just in case - It could go viral on YouTube. Although my wife tells me that no matter how badly it might go, no-one is ever going to watch a 40-minute video about testing

So, I sat down and started to think about what I wanted to talk about. I took a pretty structured approach. I read through my abstract, I googled "what makes a good presentation", I did some reading and so on. I put together a high-level list of topics that I thought might be good. I even wrote my first sentence “What is a megaproject”. And then I got bored with my own presentation and started watching TED talks.  You can tell I’m a tester, right? Basically, what I was doing was applying a testing methodology to my presentation - I read the spec, came up with some test conditions, and started turning those in to test cases.

TED SLIDE - Anyway, the first TED talk I watched was by an English guy called Ken Robinson who talked about a lack of creativity in the schooling system, and he was amazing. Just his approach, his natural confidence with the audience, the passion for his topic, it couldn’t have been better. He engaged the audience, he was funny, he didn’t lecture, and he got his message across. And I thought to myself – I want to be like that! So, I scrubbed out everything I had written and decided that I would tell you my story. Instead of lecturing I am going to tell you about what I did, what I didn’t do, how it worked out, and what I learned from it. By the end of the 40-minute presentation I might even get back to the topic of megaprojects.

So, what is a megaproject. There are loads of definitions out there around time, cost, complexity, and various other factors, but no one agrees on exact definitions and they are industry specific anyway. So my definition is if it’s in I.T., it’s over a billion dollars, and the timeline is over 5 years, I reckon that’s pretty big. But the best way I could describe it is like this. FAMILY BEFORE SLIDE – This is my family when I started working on the project. And this is my family now – FAMILY AFTER SLIDE – the project is still running by the way.. So, I mentioned earlier that I had kids. 3 kids. Literally all while working on the same project. I can tell you that they are pretty much 5, 3 and 1, just don’t ask me their birthdays because I’m pretty terrible at remembering them. But as it turns out, it is super convenient for remembering the 3 phases of the project, because one was born on each phase. So, I don’t have to remember their birthdays, I can just remember them as phase 1, phase 2 and phase 3.

So that’s my definition of a megaproject. But how do you survive and succeed in megaprojects. That’s what we are here to talk about. REDACTED SLIDE - What I can do is to share some of my experiences with you. What I’m not going to do is go in to detail about the project itself because it’s not appropriate and it is secondary to the point of the presentation.

One of my earliest strong memories of working on a megaproject came a few months after I officially started working on it. I came across this certain issue that took some creativity to resolve. I remember in those first few months everyone always talked about how the project was so big, how it was worth so much money, how it had such a large team, how it was of strategic importance, how it was a once in a lifetime opportunity and all of that. So, naturally, I thought my issues wouldn’t be too hard to resolve in the grand scheme of things. I needed a test lab – some space, an assortment of equipment, some software, and configuration control applied to it all, so we could have confidence in the accuracy of the results. Not a huge ask right? My thoughts exactly. So, I was thinking that I would do the typical thing – raise the issue with my Project Manager and they would sort it. So, I spoke to the Project Manager - “Sorry, there is no money”. So, I went to the facilities team – ‘sorry, there is no space’. And then I spoke to the logistics team ‘sorry, there is no spare equipment’. ITS NOT FAIR SLIDE - So by this point I was pretty frustrated. I remember thinking to myself “how can we be on this massive project of such importance, with this huge budget and all these resources, and I can’t get a test lab to test things before we deploy to prod!”. I’m pretty sure that came with a lot more cursing but ANZTB is a family friendly organisation so none of that here.

I don’t know whether it came to me instantly or over time, but I ended up realising that all I had actually done was gone to people and given them a problem, expecting them to solve it for me. The facilities manager was already overwhelmed, trying to squeeze 200 people in to 150 desks and had a project manager hounding them because milestones weren't being achieved because people weren't getting enough work done as they were having to share desks. Effectively what I had done was gone up and said, "me me me, I need space too!". To which the only answer I was ever going to get was "go and play outside". And the same thing for all the other people I approached. Looking back in hindsight, those people had their own challenges with working on a megaproject and were still trying to find their way, just like I was.

So, I decided to try and solve it myself. MESSY STORAGE ROOM SLIDE - I found 2 storage rooms that were full of equipment, but the space was poorly utilised. I went back to the facilities manager and said, "if I clean up these two rooms, can I use whatever space is leftover". See the difference there – a solution, not a problem. Of course, the answer was yes. He was more than happy to have someone volunteer to clean it up. In fact, they even gave us a small amount of non-project money to purchase storage racks, so we could store things safely. So, we cleaned it up and we ended up having a whole room to spare. An empty room but a room nonetheless. We walked around and found some tables that weren't being used. Again, the facilities manager was happy to get them off his plate. Now we had a room with tables. Then we started asking around – did anyone have some spare workstations. Inevitably a few turned up, but they weren't what we were after. Another person didn't have any but used to work with someone in logistics who could get us some. We got in contact with them and sure enough, jackpot. Anything we requested they were able to loan to us – workstations, monitors, keyboards, mice, printers, scanners, all sorts of stuff. CLEAN ROOM SLIDE - Now we had a room with tables and equipment. Baby steps.

We went back to the PM and said "hey, we've got a room, power, and equipment, but we need a bunch of fibre cables with specific connectors". The PM didn't have any project money to spare, but they did have access to the corporate account at a commercial electronics supplier. Now we had network connectivity as well. I mean, there was more to it than that, nothing is ever that simple in complex environments and on major projects, but it gives you the idea. And don't get me wrong, that room was less than ideal. TOO HOT/COLD SLIDE – It had no opening windows and with so much equipment running in there it got really hot. But it did have an industrial sized air conditioner designed for a server room in there. If anyone has been in to a data centre before you'll know just how cold and loud they can get. That's what it was like in this room. We'd turn it on for 10 minutes and we'd be shivering. And have to wear ear plugs while it was running because it was so loud. Then we'd turn it off and an hour later it would be hot again and we'd start the cycle over. Just thinking about it makes me laugh – you can picture a bunch of testers hunched over their workstations, shivering, wearing ear plugs and shouting so that the person next to them could hear them. No, not the greatest room ever, but we solved it ourselves and that room was the centre of testing for many, many months.

SOLVE YOUR OWN PROBLEMS SLIDE - So that's my first piece of advice – work out how to solve your own problems. Now I'm not saying that you don't raise the issues – you absolutely need to. Any Test Manager that doesn't raise the issues as they are identified is in for a pretty short career. But what I am saying is that you shouldn't expect that someone else will solve all your issues for you. They have access to the same resources and information that you do. So wherever possible, try to solve your own problems.

WORKED BUTT OFF SLIDE - That room holds so many memories for me. In the lead up to the first release we had so much to do and so little time – everyone knows what I'm talking about right! Testing is always squeezed at the end! And we are suckers for punishment. We always want to show off how good we are and that no matter how late the rest of the project is running, we are squared away. Anyway, we were working SO much. Literally 12, 14, 16 hours a day. Every day. Weekdays. Weekends. There were weeks where I didn't see my family. They were asleep when I left for work and asleep when I got home from work. I remember this funny story that my wife told me – she went to see some friends of ours and took our oldest son (William – don't worry, you can remember him as Phase 1. See, I told you it was convenient ). When she got there William was so excited because he got to see daddy. No, I wasn't there, I was at work. He just got confused and thought that our friend David was his dad. And then when David left, William cried and was so sad because "daddy left". I mean, I think it's hilarious. But I'm sure there are some of you sitting out in the crowd thinking "oh how sad that he didn't know who his dad was". Don't worry, he was only young, he'll never remember. And if he does I still have 2 other kids I can use as backups There's another important I.T. lesson there for everyone – always have a backup.

DUCK/RABBIT SLIDE – That's an interesting concept though, isn't it? The idea that different people can perceive the exact same situation differently and come to different conclusions. Most of us see it on a daily basis – we all have access to the same specification, and yet the BA meant one thing, the dev interpreted it one way, and we tested it based on how we understood it. We all had access to the same information yet we all drew different conclusions. The same thing happens with projects – we get given this abstract concept that we have to achieve, but we all have different ideas and understanding of what it is meant to look like and how to implement it. Timeframes are a classic example – everyone always has a different opinion on how long it will take to deliver.

Megaprojects are no different – except that everything is inflated by orders of magnitude. In particular, there are extremely high levels of uncertainty. To give you an example, my last megaproject started in 2010, and it is still running. 2010. Do you know what was happening in 2010? The leadership spill from Kevin Rudd to Julia Gillard was hot news. The first iPad was just being released and we were still 7 iterations away from Samsung producing exploding smartphones . Think about the technological change that has happened since then. But somehow, we have to plan and implement a massive project with huge amounts of complexity and uncertainty, with future technology advances in mind. So how do you do it? The way I did it was to embrace uncertainty. EMBRACE UNCERTAINTY SLIDE - Awesome catch phrase huh?! Sounds like a Nike ad. Just Do It - Embrace Uncertainty. It sounds so simple, but it's actually really hard to do! It requires you to admit that you don't know everything, which flies in the face of almost everything we are taught – always be prepared, answer every question, don't look silly in front of other people. And it also requires other people to accept that you don't know everything. That's a big ask.

Think about it – you're sitting in a meeting with your boss and 20 other people. Your boss asks you what your plan is for testing the next release. You have no idea, so instead of making it up, you have to be brave enough to say, "I don't know". I learnt during my time in the Navy it is ok to say, "I don't know" as long as the next 5 words are "but I will find out". And then after you say "I don't know (but I will find out) your boss has to trust you enough to accept that answer, and trust that you will do it at the right time in the project. TRUST SLIDE - All of this is in front of a room full of people who have been taught the values of having answers, not looking silly, and always being right. It's no small thing. But think how many times you have been asked to write a test plan that you have no information for. Or to provide estimates when you don't even know what is being delivered. Imagine if you could document what you did know, and then just accept that you didn't know the rest, but that you would work towards finding it out. Like that schedule section in the document that you have no clue about because its too early to know. Anything you put in there is more likely to be wrong than right. So why do we do it? its crazy. So, you know what you do? You write "to be advised" and you move on. It's a megaproject after all – there are a thousand other tasks that need your attention

CAN/CANT CONTROL SLIDE - So that's my second piece of advice – embrace uncertainty. Control the things that you can and accept the things that you can't. Don't be afraid to make decisions based on the information that you have at the time. Just make sure you continually review those decisions as new information becomes available to see if they still hold true.

Just to be clear, embracing uncertainty doesn't mean you get a free pass! I know some of you were thinking you might get away with it!   It doesn't mean that everything is high level with pictures of clouds and abstract concepts! We're testers not architects! That's actually probably one of the biggest misconceptions associated with megaprojects - that they are so big that the small details aren't important. Particularly earlier on in the project when defining things can be more difficult, people tend to operate at a macro level and underestimate the importance of the smaller details. It's all about finding a balance, finding that sweet spot between being able to keep your eye on the big picture while not getting caught out by the finer details. And I've got a really good example that shows how important finding that balance is.

PATRIOT MISSILE SLIDE - This is a MIM 104 Patriot Missile System. Originally designed as an anti-aircraft surface to air missile, deployed in the mid-80s, it was upgraded and repurposed to perform the role of ballistic missile defence in the early 90s due to the first gulf war. The theory behind it is relatively simple – a radar tracks an incoming target missile and launches an outgoing defence missile to intercept it. When the defence missile is in close proximity to the target missile it detonates and creates a field of shrapnel which destroys the target. In practice, things are a little more complicated for a bunch of reasons, one of the main ones being the relative velocity of the two missiles, as both can be travelling at twice the speed of sound. Fractions of a second make a huge difference. The smallest discrepancies in time synchronisation between devices can cause the complete failure of the system, which is exactly what happened in 1991 with an incoming Iraqi Scud missile headed for an American Army barracks. On that day the Patriot missile defence system was testing in the field for the first time, and that test failed. The missile battery that the patriot missile was fired from had been online for over 100 hours at the time the missile was launched. They were focusing on the big picture, making sure the system was on and tracking targets. However, over those 4 days of operation, the clocks on the missile system and the radar system had drifted by three tenths of a second. 0.3 seconds. Literally no more than the blink of an eye. But with the missiles travelling so fast, the patriot missile launched to intercept the scud missile was off target by 600 metres. The patriot missile failed to intercept the scud missile and 28 US soldiers were killed. They got caught out by the smallest of details.

Think about that for a minute – at a high level the system operated correctly. The radar system tracked a target, the launch system fired a missile to intercept the target, and the guidance system directed the missile to the target. All of the systems performed their individual role. Except for a tiny little defect where timestamps were calculated differently. Which resulted in the complete failure of the system and 28 people were killed as a result. Finding that balance between the big picture and the finer details is crucial. Obviously, that's an extreme example but it is a good one that shows how important finding the balance is.

Now to my knowledge I have never missed a defect that has cost people their lives, but we've all experienced mistakes that have made our hearts race and our palms sweaty. For example, lots of people say they are more scared of public speaking than anything else in the world. Thankfully that's not me or this could be going very differently!   But there was a time where I got caught out by the details on my last megaproject. TOO BUSY SLIDE - There was a really busy time where we were working across two releases – the first was running behind schedule but we needed to start on the second as well, so that didn't get delayed. Which meant the team was stretched pretty thin. Regardless, we were all working together really well, and things were running smoothly, albeit under a bit of pressure. We were even starting to catch up. I remember commenting to one of my teammates that for what seemed like the first time in at least a year, we were actually starting to get on top of things. And man did we think we were cool. Here we were, managing the testing for this megaproject, governing a bunch of test teams from global companies, delivering to compressed timeframes, testing many capabilities in parallel, running System Integration Testing concurrently with User Acceptance Testing (disclaimer - not best practice, nor recommended by the ISTQB ). You name it, we were doing it, and we actually had it under control. I was like "cool, we've got this".

The very next day, the project director was walking past and casually asked me "when will the Disaster Recovery Test Summary Report for capability x be complete?". I remember being lost for words (and that does not happen often). I had no answer. STATUE SLIDE - We had completely missed Disaster Recovery testing for an entire capability. All of us. Days before the planned release to production and we had missed a test phase. And in our project, no DR test means no change approval. Release to production delayed because testing missed something. Not because of a critical defect or anything like that. Just we made a mistake. Some of you must be thinking "how does that even happen"? The answer is simple – people make mistakes, that's part of why we as testers have jobs. We had a lot going on, there was a lot of complexity, there were some debates around scope, everyone was overworked, and as a result it got missed. Life happens. We didn't find that balance between the big details and the small.

FALL TEXT SLIDE - So, I know there are a number of my team out in the audience, and they're probably sitting there going "Nath, what are you talking about mate, it was a minor capability, we got it done, it wasn't an issue". And yes, they are right. We scrambled, my team worked their magic and organised and ran a DR test on the weekend before release, and we didn't end up causing a delay. But like I was saying before, different people perceive things differently. For me, that was one of the lowest points on the project. Firstly, because it smashed my ego – just as I was proclaiming that we were on top of things we missed something. Secondly it was the first mistake we had made that was visible outside the team.

BALANCE SLIDE - So that's my third piece of advice – Find a balance between the big picture and the small details. Delve in to things and find out what is going on but check back to whether you on track for the big picture. And always check your work – don't assume that you are perfect because none of us are.

That's a really important thing for us to remember as testers too! We are not perfect. We're all humans, we all make mistakes. In fact, in megaprojects its not all that uncommon that you will have "mistakes" that come about from things that were well planned and running smoothly. Something changes, what you were doing before now wont yield the desired outcome, so you have to change what you were doing. Lots of people seem to struggle to adapt though. MAN WITH SPEARS SLIDE - But if none of us adapted then we'd all still be running around naked chasing food with spears! Maybe even without the spears…And yet when it comes to IT we always default to what we have always done previously. Often to the point of failure. You hear it all the time don't you – people justifying their approach by saying "we've always done it this way" or "this I show I did it on my last project" or some other similar variation. That doesn't mean it was good, it just means it was familiar. Maybe it wasn't even right then either!

I used to have a manager that loved the saying "if you always do what you always did then you'll always get what you always got". Terrible grammar! But basically, if the inputs never change then the outputs wont either. I reckon he used to say it at least once a day, and it has stuck with me. So nowadays I'm not afraid to change things if I don't think they are working. Or if we get new information and I think we can get a better outcome by doing it a different way. Or if I just realise that my initial assessment wasn't right. THE RIGHT TOOL SLIDE - The reasons aren't the important part, the important part is that you adapt to your circumstances. Don't get stuck using tools and techniques that you think will work, just because that's what you have and that's what you have used before. Do some critical analysis and evaluate whether they are likely to work for you in your environment, with the limitations that have been placed on you. Use the right tool for the right job!

So here is an example for you. Because I've been on the project for what feels like an eternity, I was involved in drafting the contracts for the various vendors that were brought in to deliver certain capabilities for phase 3. Part of what that entailed was specifying the documentation they were required to produce. I based it on IEEE29119 – the International Standard for Software Testing, and one of the documents I included was a test design specification. Logically it makes perfect sense, tests have to be designed before they are written, we should review the tests that are being designed before the vendor starts producing test cases. Its good practice, its structured, it mitigates risk, and so on. For a number of reasons, it didn't have the intended result – there were time pressures, the tool didn't support the auto export in the manner we had planned, there were commercial implications around reviews and payment of milestones, and some other project specific factors that created some challenges. So, we gathered up the key testing stakeholders, got in a room, discussed the pros and cons, and concluded that, in this specific circumstance, the test design specification wasn't adding enough value to justify its cost (both in terms of time and money). So, we got rid of it. SUCCESS SLIDE - I proposed a contract amendment to remove the test design specification, put it up to the appropriate authority, got it approved and had the documents removed. And all the heartache disappeared. In this case it was the right decision. It removed a bunch of commercial complexity and allowed people to focus on what was important – writing tests to validate the solution without adding unnecessary overhead.

Of course, there were times when I got it wrong though. WRONG ANSWER SLIDE - The project is complex – our systems development methodology was effectively iterative wrapped up inside incremental wrapped up inside v model. Yep… You should have seen the faces we pulled when we were trying to work out how to draw the diagram for that one!  What that meant in reality is that we had to assess each release for each capability as an atomic unit, rather than the solution as a whole. This happened many times over a lengthy period. We started out with a definitive list of test exit criteria which people had to meet. As the project evolved and we adapted, we modified the exit criteria. That meant that different capabilities achieved test exit against different criteria. Conceptually this was ok, because the expectation was set during test entry, so people knew what exit criteria they had to meet. But in practice it created issues when the assessing capabilities iteratively. The same capability, assessed at different times, was measured against different criteria. So did they have to use the initial criteria or the modified one? It created confusion, and the end result was that I could (and did) give a different answer to the same question asked at two different points in the project. You can see the problem that created.

I'm pretty sure there were times during that phase where my own team thought about running me over in the parking lot and claiming it was an accident. Probably in an electric car so I couldn't hear them coming – they're smart like that and love to embrace the latest technology  Trying to be adaptable to changing circumstances but also providing consistency and a common target for everyone to work towards is tricky. Adapting too often feels disjointed and leaves no common goal.

Not adapting enough leaves you using outdated methods or doing things that add no value just because some process says so…

BEAR AND PENGUINS SLIDE - And that's my fourth piece of advice – Adapt to your circumstances. Tailor your approach to the specific limitations of your project. Don't blindly follow process just for the sake of it – if it isn't adding value then get rid of it. Don't be afraid to change tact if you see that something isn't working, or you get some new information. But be careful to strike a balance between adapting too often and not enough.

So, here's the summary version of how to survive and succeed in megaprojects for the people that were too focused on counting how many times I said the word megaproject :

1. NUTSHELL SLIDE - Work out how to solve your own problems. Rarely will others solve them for you.
2. Embrace uncertainty. Its ok to not have the answer to everything, so long as you have a plan for how to find it out .
3. Find a balance between the big picture and the finer details. Make sure you pay just as much attention to the micro level as you do to the macro level.
4. Adapt to your circumstances. Learn new ways of doing things. Don't be afraid to change your approach based on new information.

But the most important thing I can tell you is to be confident in yourself. Have confidence in your ability and strive to achieve the best outcome you can. There will be plenty of times where you doubt yourself – don't let it get you down. People are always quick to tell you why something can't be done, but rarely will they tell you why it can be done.

And it can be done.

It just might take a while. ELEPHANT SLIDE - So keep chewing on that elephant.

Thank you.