

Lessons learnt, or: The journey from Systems Administrator to IT Manager

by Christian Unger
of the Translational Research Institute - Australia



Who TRI is:

The Translational Research Institute is made up of:

- Mater Research
- Queensland Government
- Queensland University of Technology
- University of Queensland

What TRI does:

TRI's vision is to:

Improve healthcare outcomes resulting from early innovative translational research.

And “translational research” is:

TRI AUSTRALIA - TRANSLATIONAL RESEARCH PATHWAY



INNOVATION
PUBLISHED &
PATENTED



CLINICAL
STUDIES



CLINICAL
TRIALS



CLINICAL
PRACTICE



INTERNATIONAL
ADOPTION &
ASSESSMENT

Disclaimer

Some hypothetical experiences discussed hypothetically happened to other hypothetical people, who hypothetically were not present at the time, working for hypothetical organisations none of us have ever heard of. ... hypothetically.

No SCSI Standards were harmed during the creation of this presentation.

SysAdmins already know




But managers discuss these issues with others who have a completely different background.

Part 1 - SPoFs

- Every system has them.
- Can be a:
 - Component;
 - Process;
 - Person;
 - Software glitch.
- AWS pride themselves on their high level of redundancy, and by extension lack of SPoF.

The uptime is pretty amazing

 DATA CENTRE SOFTWARE NETWORKS SECURITY INFRASTRUCTURE DEVOPS BUSINESS HARDWARE

Data Centre ► **Cloud**

AWS blames 'latent bug' for prolonging Sydney EC2 outage

First the secondary generators failed, then the glitch slowed some service restoration

9 Jun 2016 at 06:28, [Darren Pauli](#)



14



66

Amazon Web Services has explained the **extended outage** its Sydney services suffered last weekend, attributing downtime to a combination of power problems and a "latent bug in our instance management software".

Sydney recorded over 150mm of rain on last weekend. On Sunday the 5th the city copped 93 mm alone, plus winds gusting to 96 km/h.

Amazon says that bad weather meant that "At 10:25 PM PDT on June 4th [mid-afternoon Sunday in Sydney – Ed] , our utility provider suffered a loss of power at a regional substation as a result of severe weather in the area. This failure resulted in a total loss of utility power to multiple AWS facilities."

AWS has two backup power systems, but for some instances both backups failed on the night in question.

(Good on them for being so open about this incident)

Site Redundancy

Assume two sites, with redundant fibre to stretch the internal network.

- Redundant switches or not?
 - One PSU or Two?
- How does the primary DC differ to the secondary site?
- What failure scenarios are you covered for?
- Are the fibre path really redundant?
- How many power feeds into the DC ?
- Onsite has two power feeds, but what about offsite's CRAC unit(s)?
- Instead of stretching the network, how about two completely separate setups?
- What about an ISP outage? Do you have different ISPs for each site?

System Setup

- NAS with redundant controllers, dual network and dual connected to:
- Two storage trays, capable of holding 60 disks each:
 - One tray expensive storage - 20 disks
 - One tray cheap storage - 60 disks
 - Each tier requires a cache disk and a journal disk
 - “Fortunately” those disks do not need to be in the same tray as their tier's storage.

What happens when the fast storage tray loses power (RCD trips, and the remaining, redundant power supply can't take the load)...?

Eggs and Baskets

- If the Virtual Infrastructure is a basket...
 - Storage system(s);
 - Interconnects;
 - Update / feature / fixes not yet installed;
 - Virtualisation software;
- ...then VMs are the eggs.
 - Not all VMs are equal.
 - What/Who defines “important”?
 - Turns out printing can be “most important”...



Power

An engineer in the DC (on my first day) commented how they just finished installing the UPS. He casually mentioned that the UPS being in the same physical location (room) as the computers is bad, because there are a lot of batteries and that presents a risk because one could catch on fire.

Guess what happened 7 years later...

Guess what happened less than a year later to the guys that commented “that was stupid” with the identical setup installed at about the same time...

Most likely cause: Someone put a battery down hard or at an angle at some point.

(Batteries had a 10 year life expectancy)

Part 2 - Expectations



PICTUREQUOTES.com

- Explain how every conclusion is reached.
- Document the assumptions predictions are based on.
- It will come up.

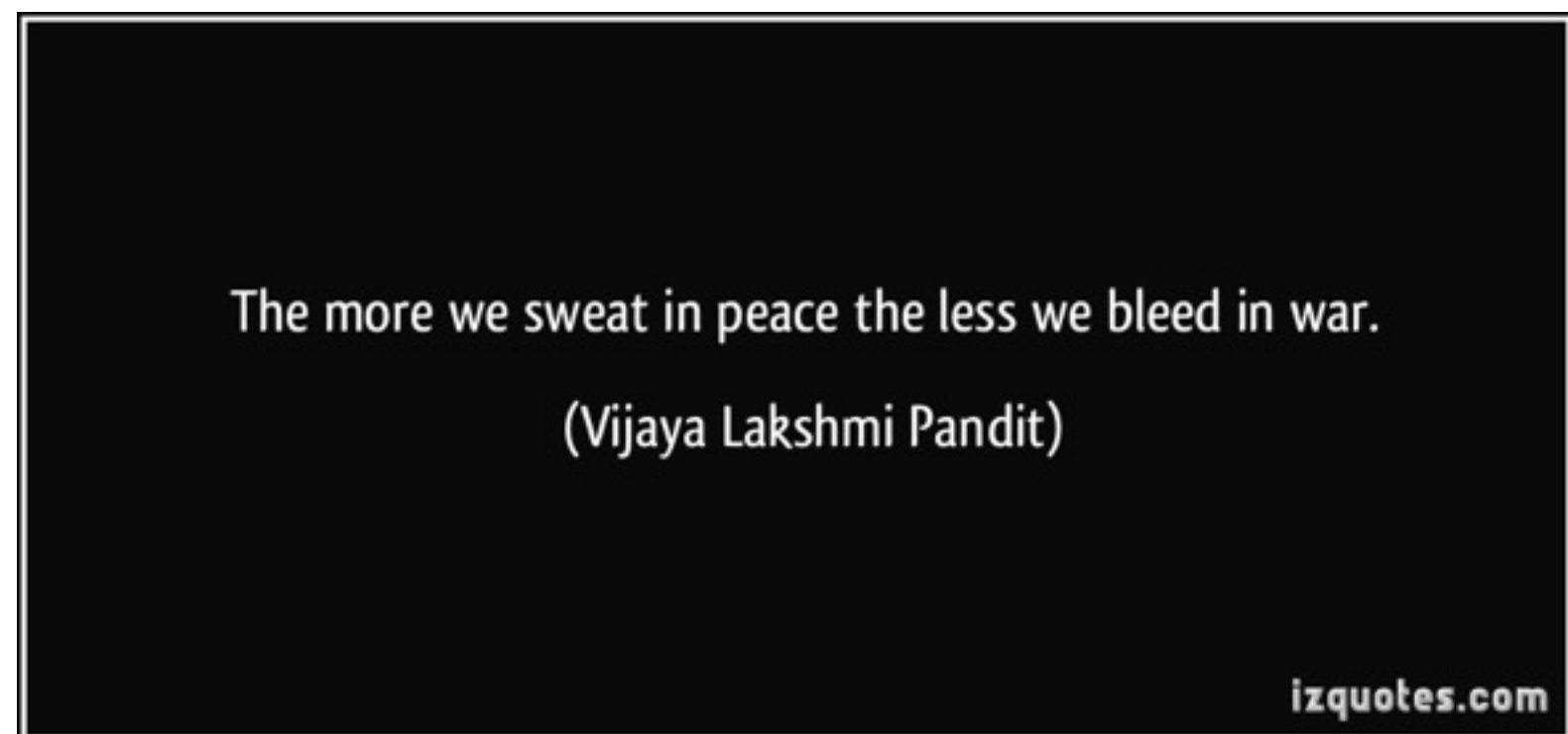
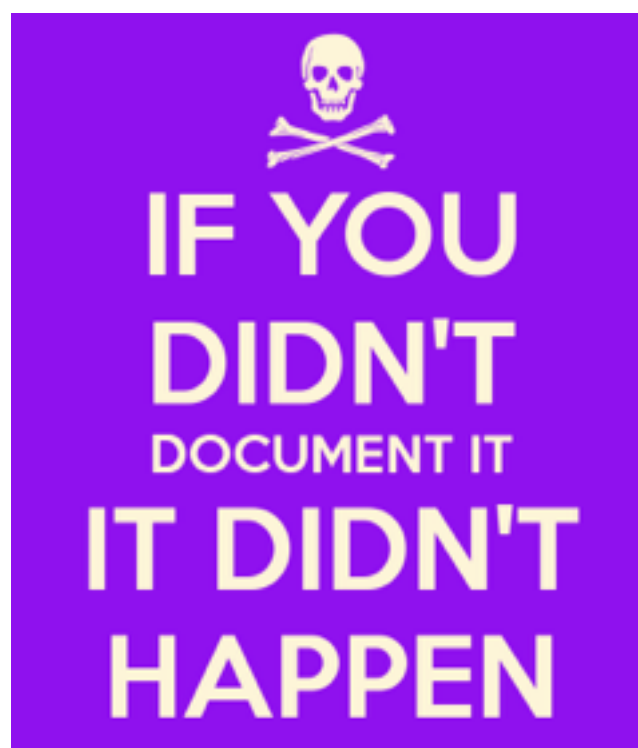
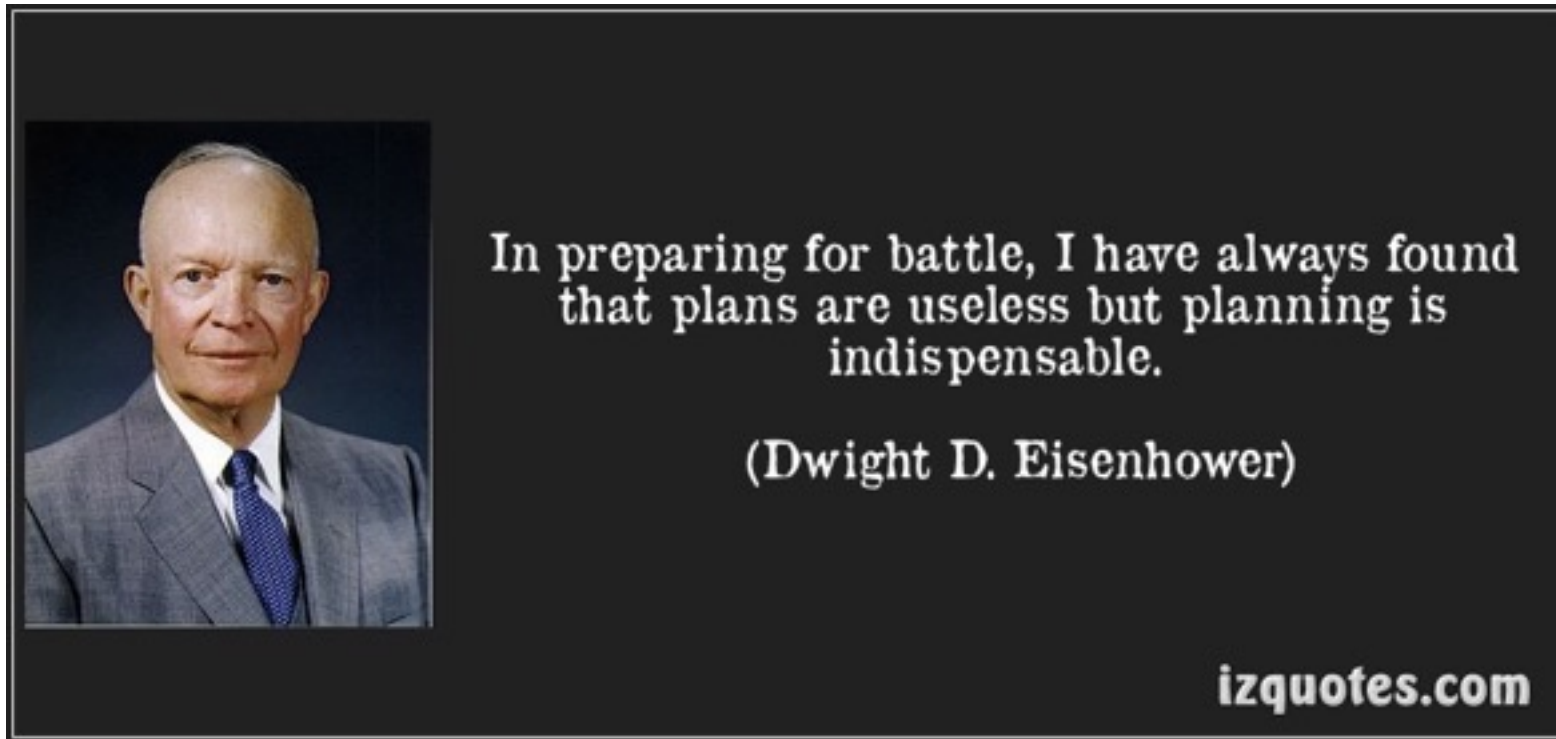
DR vs BC

BC and DR - different things to different people for different systems.

- Business Continuity:
 - If most systems are up by Monday morning all is well; vs
 - Everything “just” keeps running.
- Disaster Recovery:
 - All data is safe and can rebuild over the next fortnight; vs
 - If a failure is detected at the primary site, fail to the hot standby.

Is the recovery worse than the failure?

Platitudes...



Disasters will happen, keep calm, know what will happen

Say a rack loses power, so:

- What if that rack contains the link to your BC site?
 - No worries local site is still up.
- What if a service in that rack is replicated to BC site?
 - Wait ... how does BC take over?
- What if both happen at once?



...and by extension, what won't happen.

time = \$\$\$

“The <service> must be up 24/7”

Two providers, specialising in <service> offer:

- 24/7/365 support desk
- 1 hour guaranteed response time
- No limits on support hours
- 99.95% uptime guarantee (hosted in AWS Sydney)
- \$30,000 per year
- Support during 09:00 - 17:00
- No response time guarantee
- 5 hours support hours per month
- No uptime guarantee (hosted in AWS Sydney)
- ~\$6,000 per year

The initial statement may stand, or be revised...

Requirements

How Projects Really Work (version 1.0)

Create your own cartoon at www.projectcartoon.com



How the customer explained it



How the project leader understood it



How the analyst designed it



How the programmer wrote it



How the business consultant described it



How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

\$\$\$ = time

- If you need something urgently, consider getting it set up. But:
 - Some “providers” direct what to do (driving the project along) and implement only some very specific components.
 - Others build the entire service.
- Be clear what is being paid for and what expectations are.
- Who will maintain the system & using what documentation?
- Maybe get someone to do the BAU stuff, while your in-house team builds the new system?

BUT: Will it actually save time?

A tale of three vendors

Say you outsource your <service> to:

- An IaaS vendor for the VMs with the OS pre-installed;
- Another to:
 - Install additional software;
 - Backup “the system”; and
 - Maintain a database; and
- Another vendor to configure the <service>.



... and the gaps that connect them

- Who is patching the OS?
- Is the OS configuration being backed up? what is “the system”?
- Is anyone patching the “additional software”?
- Who is in charge of security?
- What are the risks of (a fourth party [you]) doing anything to the <system>... like patching?
- What if any of the vendors go out of business?
- Think of the <service> like any in-house <service> to find gaps. When could we patch this? How do we update firewall access?

It's not me, it's you

A <critical service>, connected by iSCSI to a storage appliance, which is presenting storage as multiple 1TB LUNs.

Under load individual LUNs disappear.

The maker of the appliance's OS say "Yeah ... that's a known issue, it's fixed in <this version>...".

The vendor maintaining the appliance says: "No, it's <your service>. We have not certified <this version>."

You see the array fall over, you predict the next failure and you trigger the fault in other systems. It **is** that bug, but no joy.

Replace array with something stable, expensive and devoid of any SPoF (*cough*).

The appliance maintainer calls: "The appliance is fine; it's the service"...

\$\$\$\$ = \$\$\$\$

Incidentally, what if you bought that appliance in the first place?
This is why a lawyer checking your contracts is a good thing.



Evaluating Solutions

Risk analysis is

AWESOME

- Two solutions, functionally the same, but made up of different components (or using the same components differently):
- Cost is just one metric, so consider:
 - Organisation's ability to obtain value;
 - Complexity of the solution, both to implement and maintain;
 - Provider alignment with your organisation's industry / size / etc.;
 - Familiarity with the components being considered;
 - Availability of skills to perform the work; (there will be more).

What if evaluation gives the wrong result?

Don't cheat - at the end of the day the aim is to choose a solution that can actually be delivered.

- Manage the risk instead.
- What if you stage deployment?
Reduce risk by spreading the implementation of components to gradually achieve a “complete” solution?



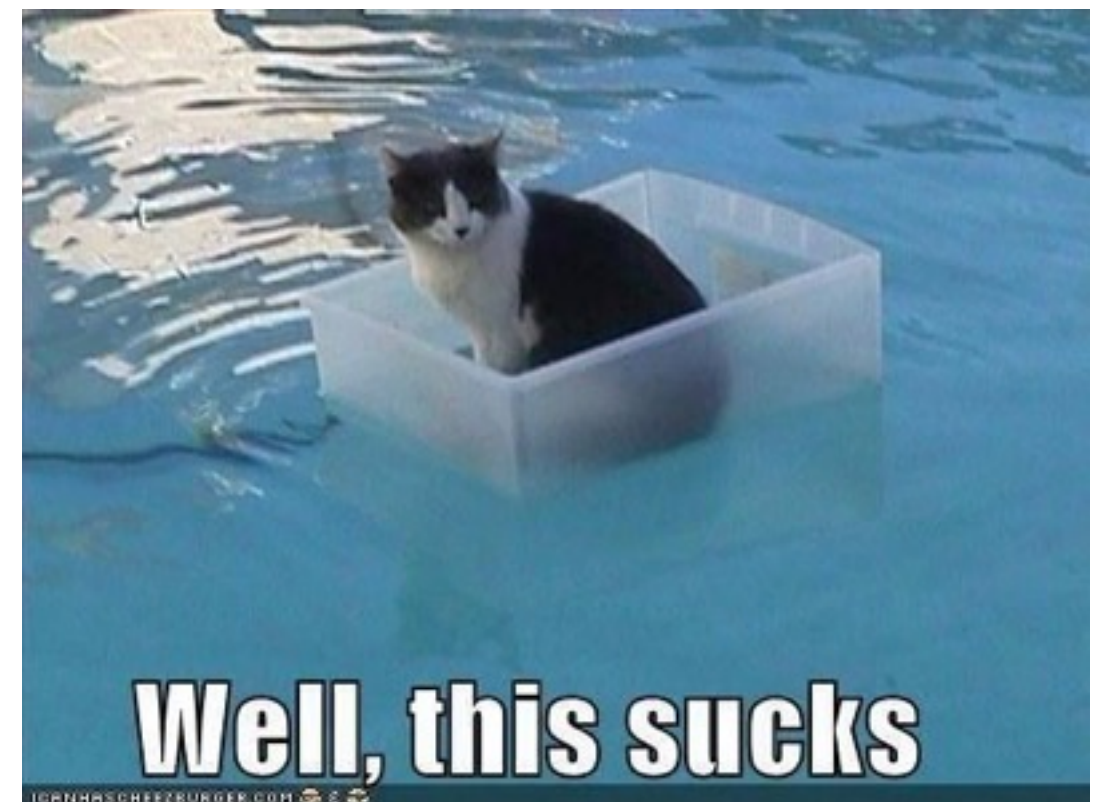
One more platitude...



If your background is Linux, it's hard to consider Windows.
Do it anyway.

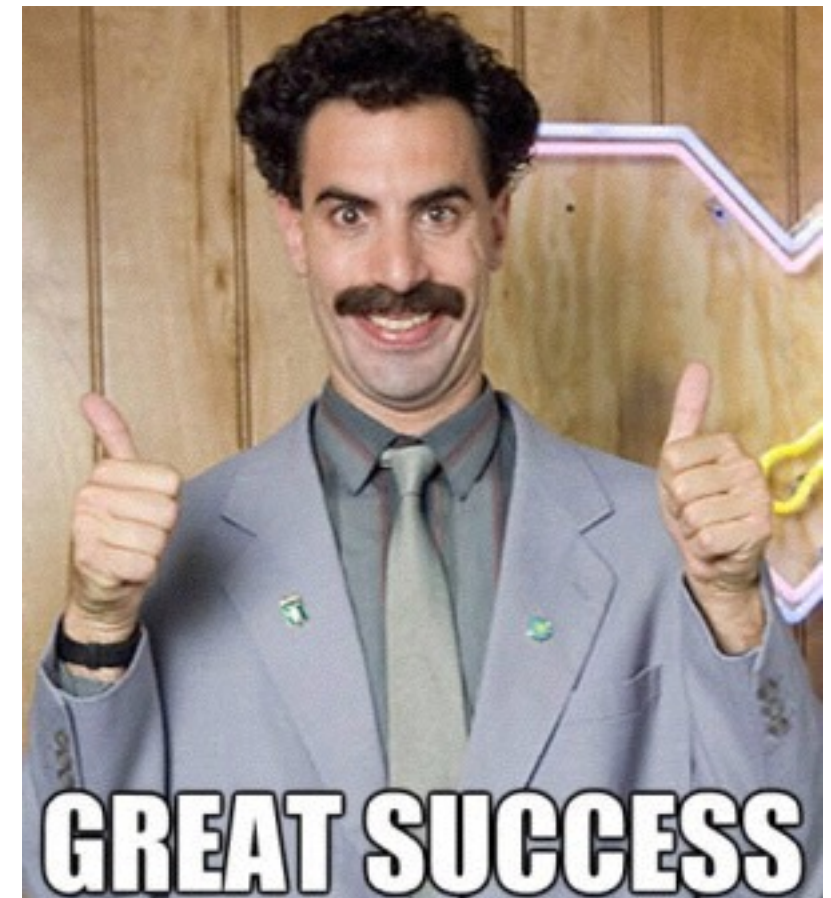
Conclusions - SPoF

- There is always a SPoF, you just have not found it.
- Solutions without SPoF, mean the someone assumed a set of specific failure scenarios.
- Your disaster will vary.
- If it doesn't it's not a "disaster".



Conclusion - Expectations

- Manage expectations (up and down).
- Look for gaps in your SLAs / outsourcing setups.
- If it's not in the SLA / contract it is not getting done.
- Document everything.



Questions? Comments!



christian.unger@tri.edu.au