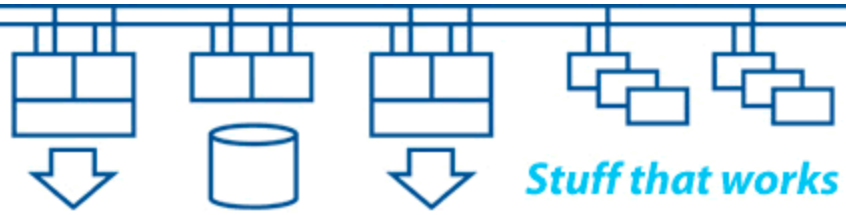
HC2009

Mission-critical healthcare: NHSBT Pulse

Colin Butcher



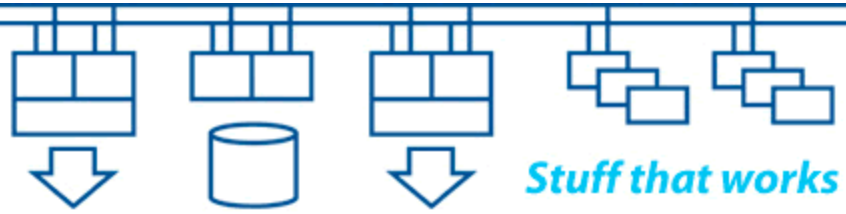
NHS Blood and Transplant

Pulse manages the collection, production and supply of blood products throughout England and North Wales

OCSL

Mimer

savant



Operational context

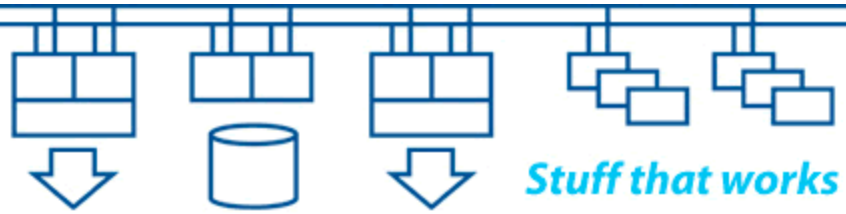
- Move from separate regional systems to single national system with consistent merged national database
- Increased availability and performance demands, hence need for disaster-tolerance
- Hardware refresh

  First had to determine if it was possible to migrate the data within an acceptable time window – so we built a proof of concept system to test it

Implementation specific context

- Pulse application updates (Savant)
- Database major version upgrade (Mimer)
- Operating system major version upgrade (OpenVMS)
- Platform migration from Alpha to Integrity (HP)
- Storage migration from HSG to EVA
- Data network infrastructure segmentation
- Common configuration and setup across all clusters
- Separate test & training environment

- All with minimal loss of service!



1) Design principles for mission-critical systems

An overview of the issues to be considered when designing, implementing and operating disaster-tolerant, mission-critical, multi-site systems.

DT, HA and DR – what do they mean?

- Disaster tolerance:
  - Continue operations while surviving major site outages without loss of data
- High availability:
  - Continue operations while surviving equipment and software failures without loss of data
- Disaster recovery:
  - The process of restarting sufficient operations to run the business after serious disruption

Risk

- What is the probability of a situation occurring?
- What is the impact if that situation occurs?
- What are the consequences?

- Most projects handle medium risk well enough
- Many projects over-specify to cater for what are in fact low probability or low impact issues
- Some projects under-specify and fail to cater for what are in fact high probability or high impact issues

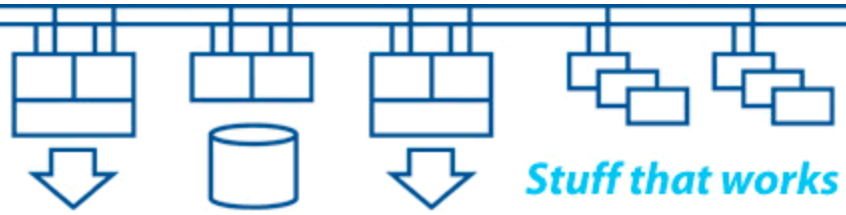
Mission Critical systems

Mission critical systems need to be able to:

- Survive failures (resilience and failover)
- Survive changes (adapt and evolve)
- Survive people (simplify and automate)
- Never corrupt or lose critical data (data integrity)
- Requirements never remain static over an extended period of time, so we need to be able to make changes during the operational lifetime of the system
- Circumstances change, so we often need to be able to extend the operational lifetime and scope of a system
- It's not a theoretical exercise!

"Survivability test"

Cause of Outage:	Planned (Maintenance)	Unplanned (Failure)
Hardware	?	?
Operating System	?	?
Network Layer	?	?
Layered Products	?	?
Application Software	?	?
Application Data	?	?
Environment	?	?
People	?	?



RPO and RTO

How long have you got?

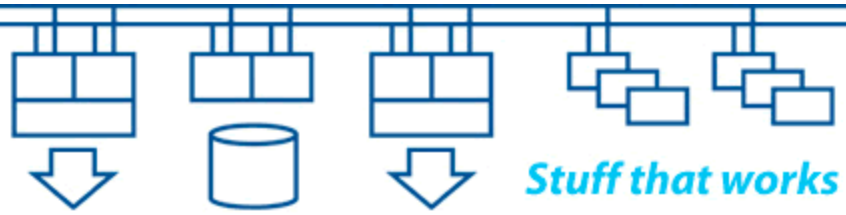
How much data can you afford to lose?

RTO = Recovery Time Objective

- What level of service outage can we tolerate?
- How quickly do we need to recover?
- How quickly do we need to be ready to deal with a subsequent failure?

RPO = Recovery Point Objective

- How much data can we tolerate losing?
- How quickly do we need to react to a failure?



Availability and performance

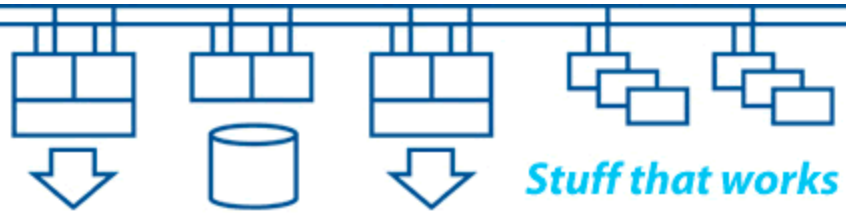
Availability:

- Probability of system being available for use at a given instant in time within the 'operational window'
- Function of both MTBF (reliability) and MTTR (repair time)

Performance:

- Performance issues are often the cause of transient system failures and disruption
- The systems have to have sufficient capacity and performance to deal with the workload in an acceptable period of time under normal, failure and recovery conditions

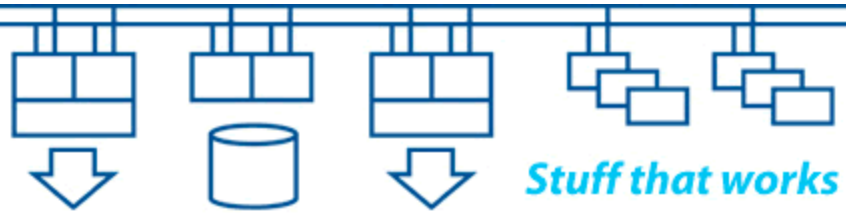
Availability is more important than performance



Designing for availability

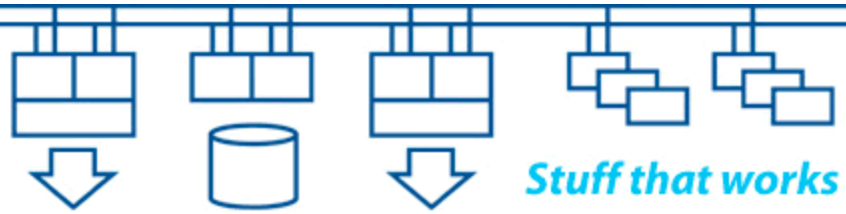
- Which parts of the system are mission-critical?
- Which parts of the system are safety-critical?
- What kind of failure do we prefer?
- What state transitions occur during failure and recovery?
- How can we recover from a failure without data loss or data corruption?
- How will you test your failure and recovery scenarios?

- Bandwidth – determines throughput
  - It's not just "speed", it's throughput in terms of "units of stuff per second"
- Latency – determines response time
  - Determines how much "stuff" is in transit through the system at any given instant
  - "Stuff in transit" is the data at risk if there is a failure
- Jitter ("div latency" or variation of latency with time) – determines predictability of response
  - Understanding jitter is important for establishing timeout values
  - Latency fluctuations can cause system failures under peak load



Multi-site issues

- Naming conventions
- Quorum and voting schemes
- Data replication schemes
- Effects of distance on network and storage protocols
- Symmetric or asymmetric operation – how good is your "crystal ball"?
- Centralised (and replicated) monitoring and alerting
- Remote access for management and operation
- Avoid automation of decision making



Testing

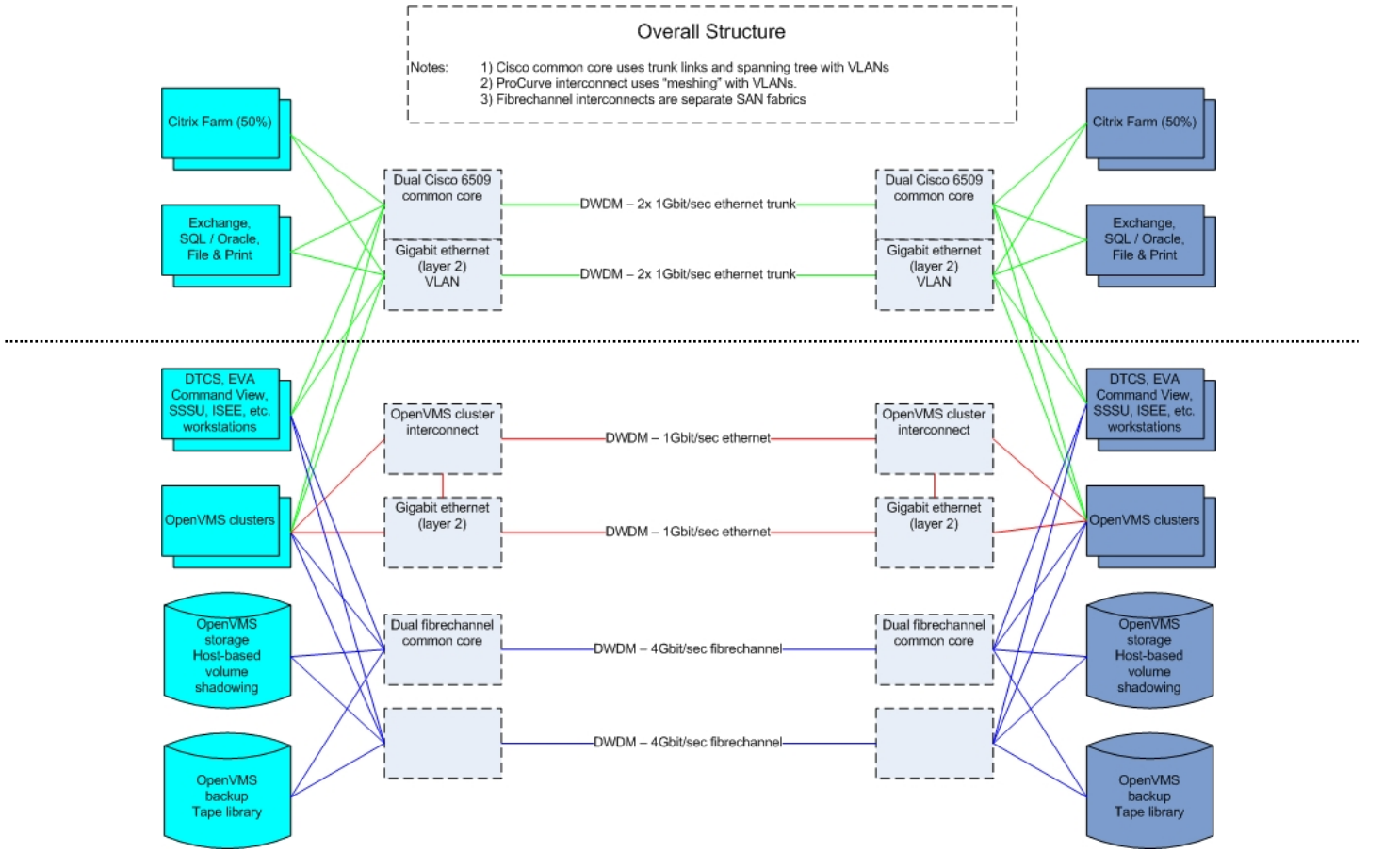
- We need to prove that service will continue with minimal disruption during failure and recovery
- We need to test for scale as well as functionality
- We need to test every aspect of the system and surrounding infrastructure under normal, failure and recovery conditions
- How will we generate a realistic load for testing?
- We need to regularly rehearse and test our procedures and plans to ensure that we stay current

2) NHSBT Pulse

An overview of how the new NHSBT Pulse systems fit into the NHSBT infrastructure.

We have Production, Archive and Test environments with a shared common infrastructure, all of which is separated from the rest of the existing infrastructure.

Where these systems fit in the new infrastructure

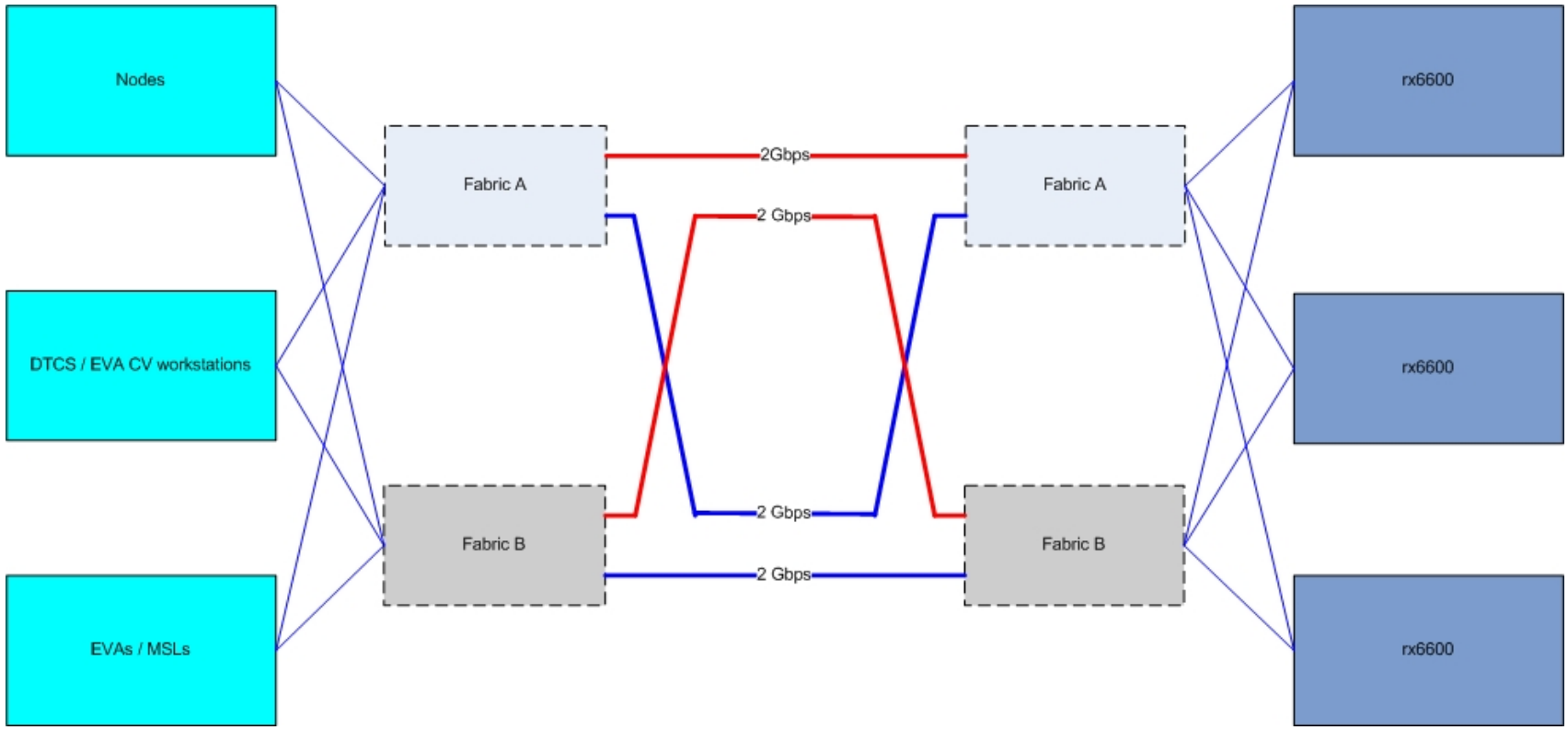


Functional layout

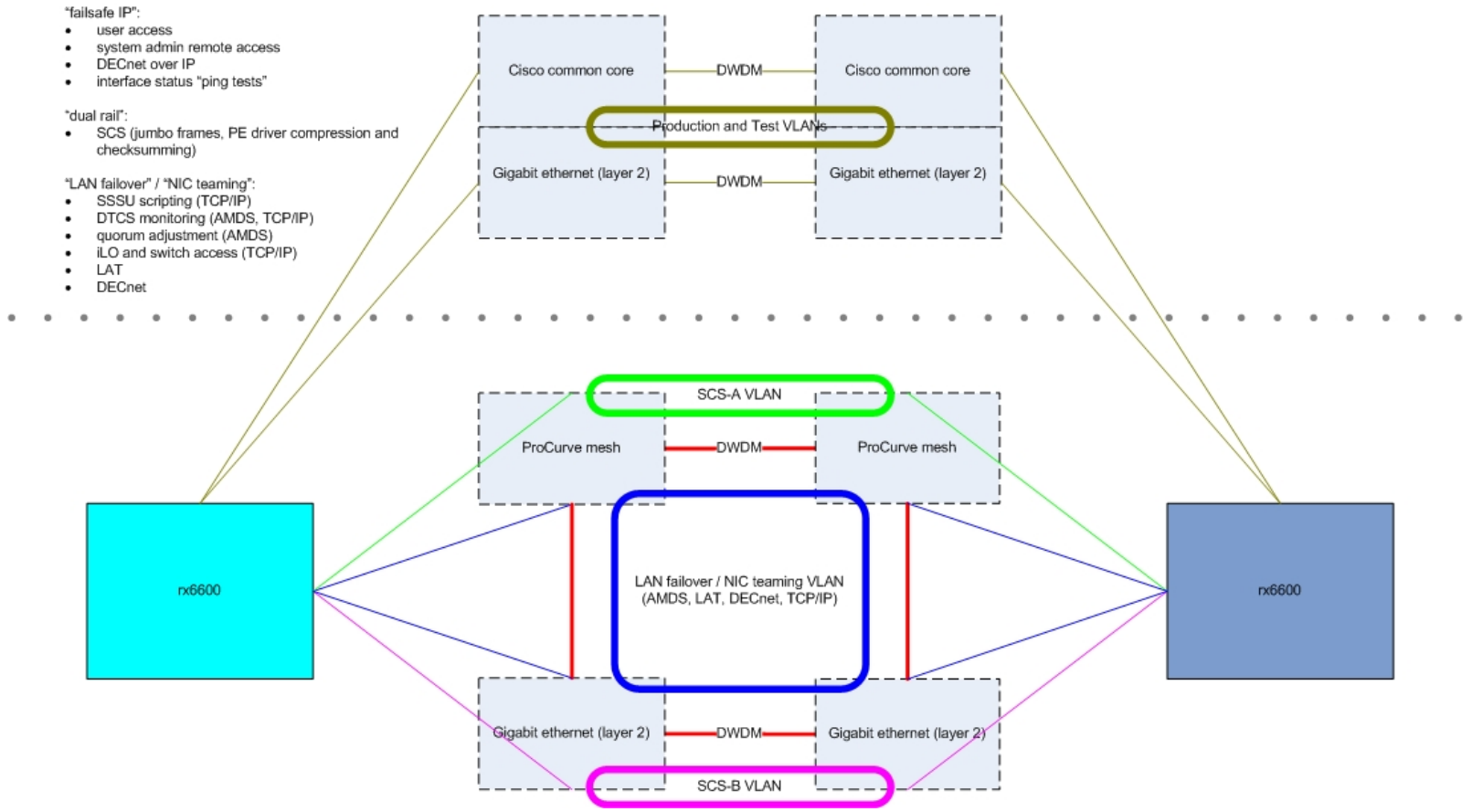
The systems are split up into:

- Common infrastructure (SAN fabrics, private network interconnects etc.)
- Production environment (a split-site cluster with host-based volume shadowed storage)
- Test environment (a split-site cluster on a smaller scale)
- Archive environment (a single node at Site A)
- Duplicated monitoring and reporting facilities
- External connectivity for users

Common SAN fabric infrastructure

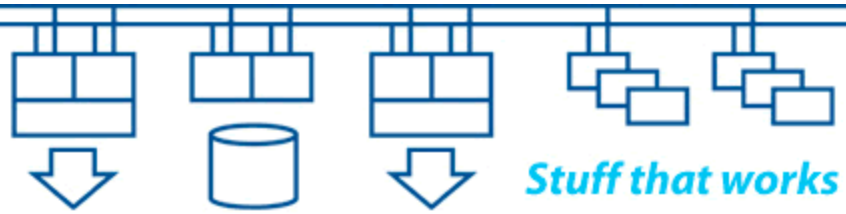


Common data network infrastructure



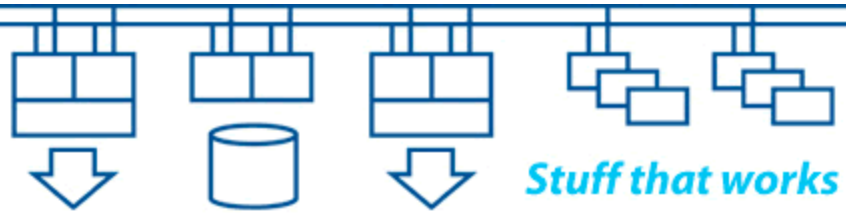
Multi-site clustering

- Split-site OpenVMS clusters give us "shared everything" access to data with protection from loss or corruption, even in the event of site failure
- Host-based volume shadowing (HBVS) ensures that data is consistent across all members of the shadow sets.
- The quorum scheme lets Site A continue if Site B fails and protects us from data corruption due to a partitioned cluster
- The DTCS software monitors the systems for us and (most important of all) controls the formation of storage shadow sets when the systems boot and when nodes rejoin the cluster



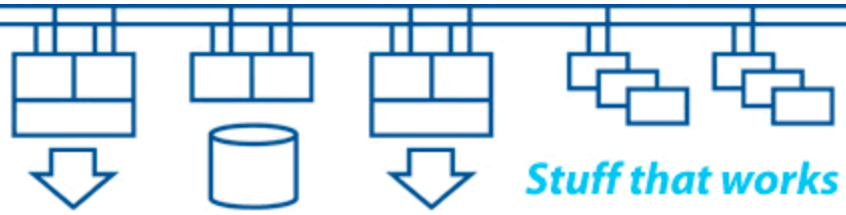
DTCS – monitoring and control

- DTCS is a set of HP and 3rd party products with installation, configuration and support services
- Remote console access, management and console output logging
- Integrated monitoring and quorum adjustment
- Rule based monitoring of individual systems / nodes
- Rule based SNMP polling of equipment
- Rule based TCP/IP "ping reachability" polling
- GUI and e-mail based alerting
- Scripting of failover and recovery actions across all systems / nodes and storage subsystems



3) Project delivery

An overview of why we believe the project was delivered successfully.



Success factors

- Small team of committed people
- Clear objectives
- Built ‘proof of concept’ data migration system first
- Built system ‘on paper’, discussed it extensively and resolved potential technical problems prior to purchasing equipment and building system platform
- Project management and planning
- Leadership and collaborative working
- Trust between team members
- Sufficient flexibility to cope with issues as they arose

Thank you for your participation

Colin Butcher

For further detail, please see: http://www.availabilitydigest.com/public_articles/0310/uknbs.pdf