Mission-critical systems with OpenVMS

Oracle Rdb Forum 2017

Everyone has a test environment.
Some are lucky enough to have a separate one for production.

Colin Butcher CEng FBCS CITP
Technical director, XDelta Limited
www.xdelta.co.uk

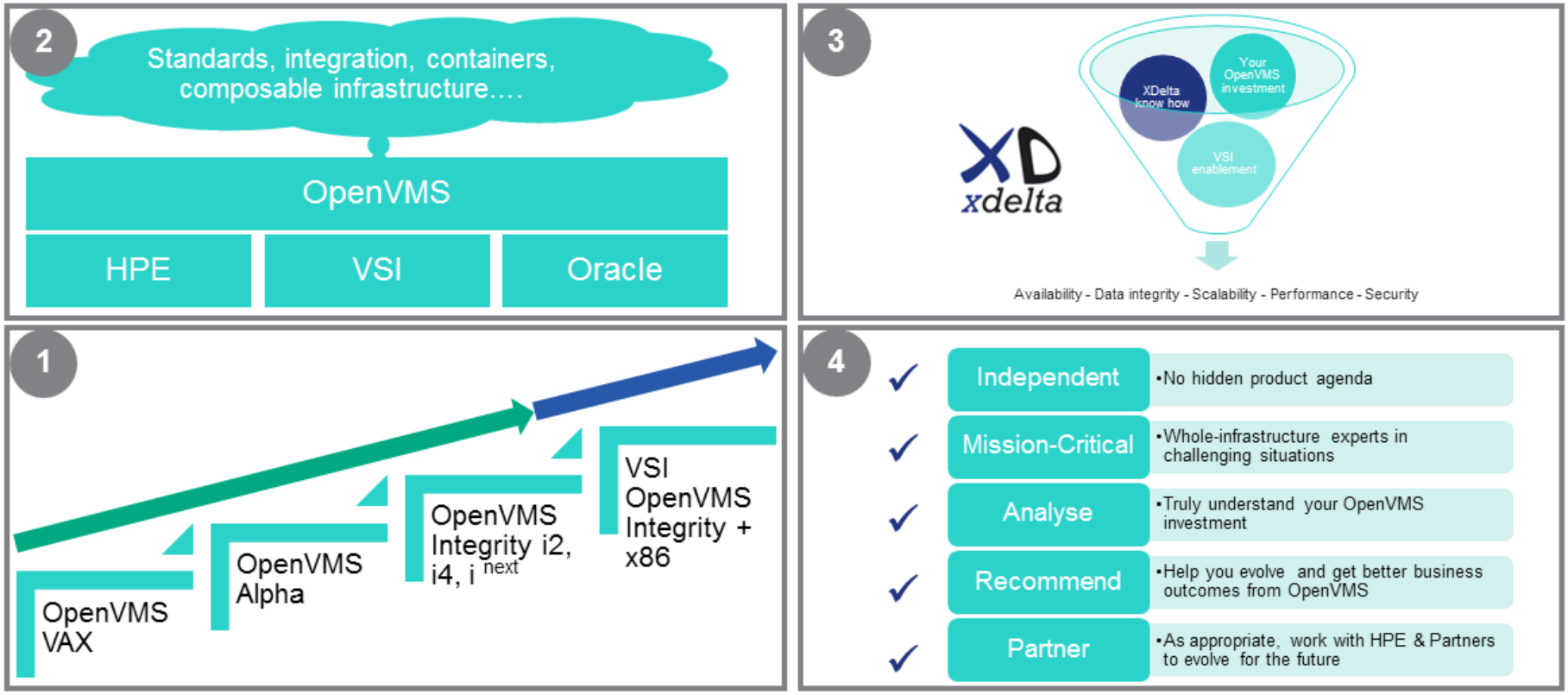
XDelta: Who we are

vms Software

- VSI Professional Services Alliance member
- Independent consulting engineers since 1996:
  - UK based with international reach
  - Delivering OpenVMS based systems for 30+ years
- Technical leadership for business-critical systems
  - Design, planning and implementation
  - Mentoring and skills transfer
  - Systems engineering background
- Gartner (2009):
  - Identified XDelta as one of few companies world-wide capable of OpenVMS platform migration projects

xdelta

XDelta - a trusted advisor to advance your critical OpenVMS application infrastructure



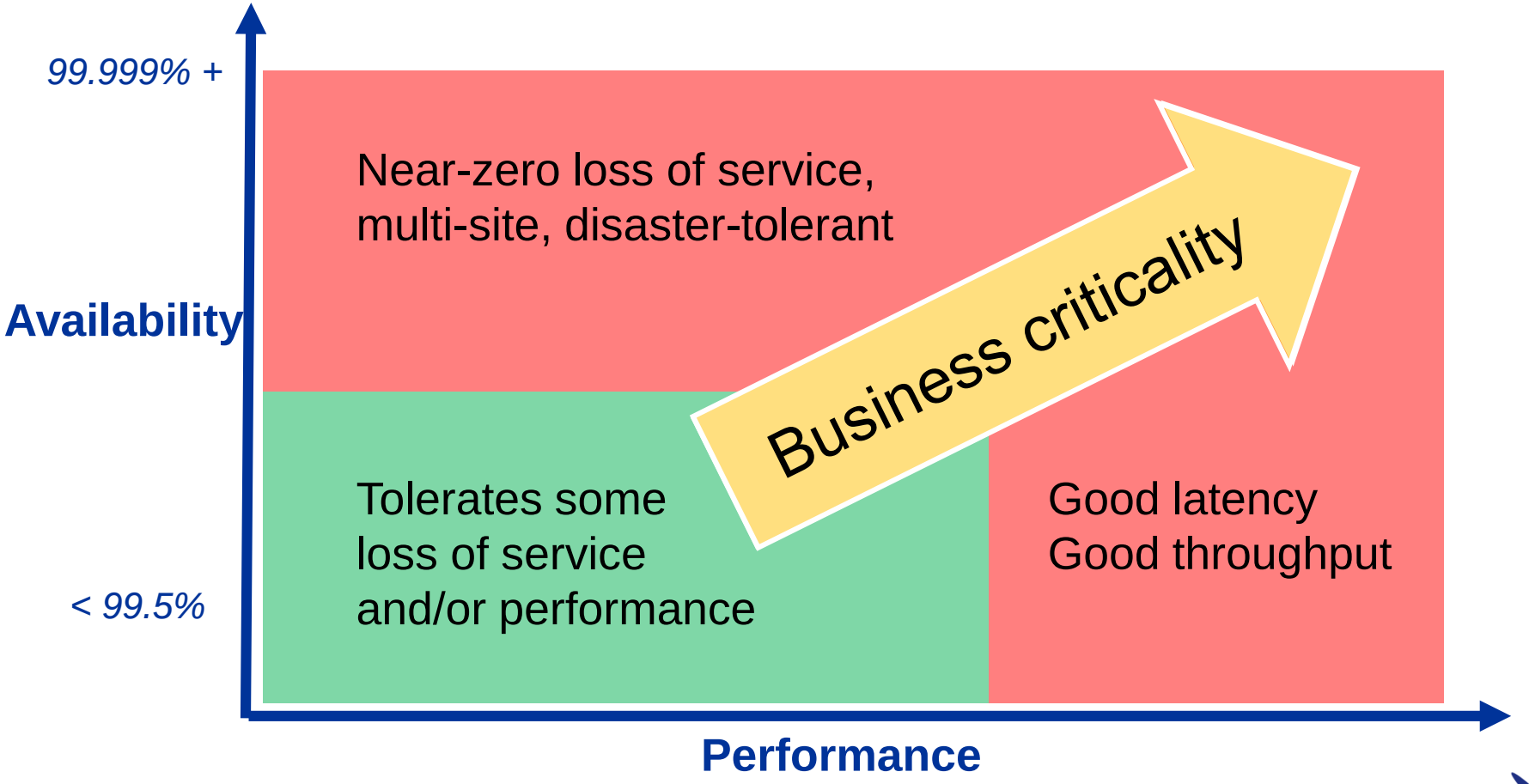
Hewlett Packard Enterprise

xdelta

Agenda

- What is mission-critical?
- Design for survival and change
- Why OpenVMS for mission-critical?
- OpenVMS guidelines for availability
- Project leadership

Business criticality



Terminology

- Availability:
  - Probability of system being available for use when needed
  - Function of MTBF (reliability) and MTTR (repair time)
- Disaster tolerance:
  - Surviving major site outages without loss of service
- High availability:
  - Surviving failures at a site without loss of service
- Disaster recovery:
  - Restarting systems after loss of service (however brief), typically from another location (DR site)

Design goals

- Understand the purpose and your “duty of care”
- Design for change, not steady-state
- Think long-term

- Operational safety - minimise risk of errors and disruption
- Systems management - logging and information gathering
- Adapt to changing workloads - performance, scalability

- How to test it ?
- How to transition into service ?
- How to transition to follow-on system ?

Survivability matrix

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

Design decisions

- All design decisions are compromises and require you to exercise judgement
  - Big decisions which have long-term implications and constraints
  - Small decisions which seem big at the time
  - There will be requirements and constraints you don't yet understand or know about

- Establish meaningful naming conventions
- Modularity and simplicity
- Document your decisions and reasoning

Availability and performance

- A system that fails to perform is a system that is unavailable.
- Performance related failures are often transient and exceedingly difficult to fully understand and resolve
- The systems have to cope with the workload under normal, failure and recovery conditions

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 ?

- Protect the data, even if everything else fails

- Risk is a combination of probability of occurrence and worst-case effects for a given failure scenario.
- Allow for failure – success is only one of many possible outcomes.

Why OpenVMS for mission-critical ?

- The best platform available
- Now has a clear future

Top-ten reasons for OpenVMS

1. Well architected, well documented
2. Stable, reliable, modular, consistent interfaces
3. High availability, performance and security by design
4. Oracle Rdb cluster-wide relational database
5. Scales well from small to large implementations
6. Shared-everything architecture
7. Shadowing with direct path to storage
8. Clustering with direct path between nodes
9. Straightforward user interface
10. Culture of continuous service, not “reboot to fix”

Guidelines for mission-critical OpenVMS

- Naming conventions
- Network connectivity
- Storage connectivity
- Multi-site clusters
- Housekeeping

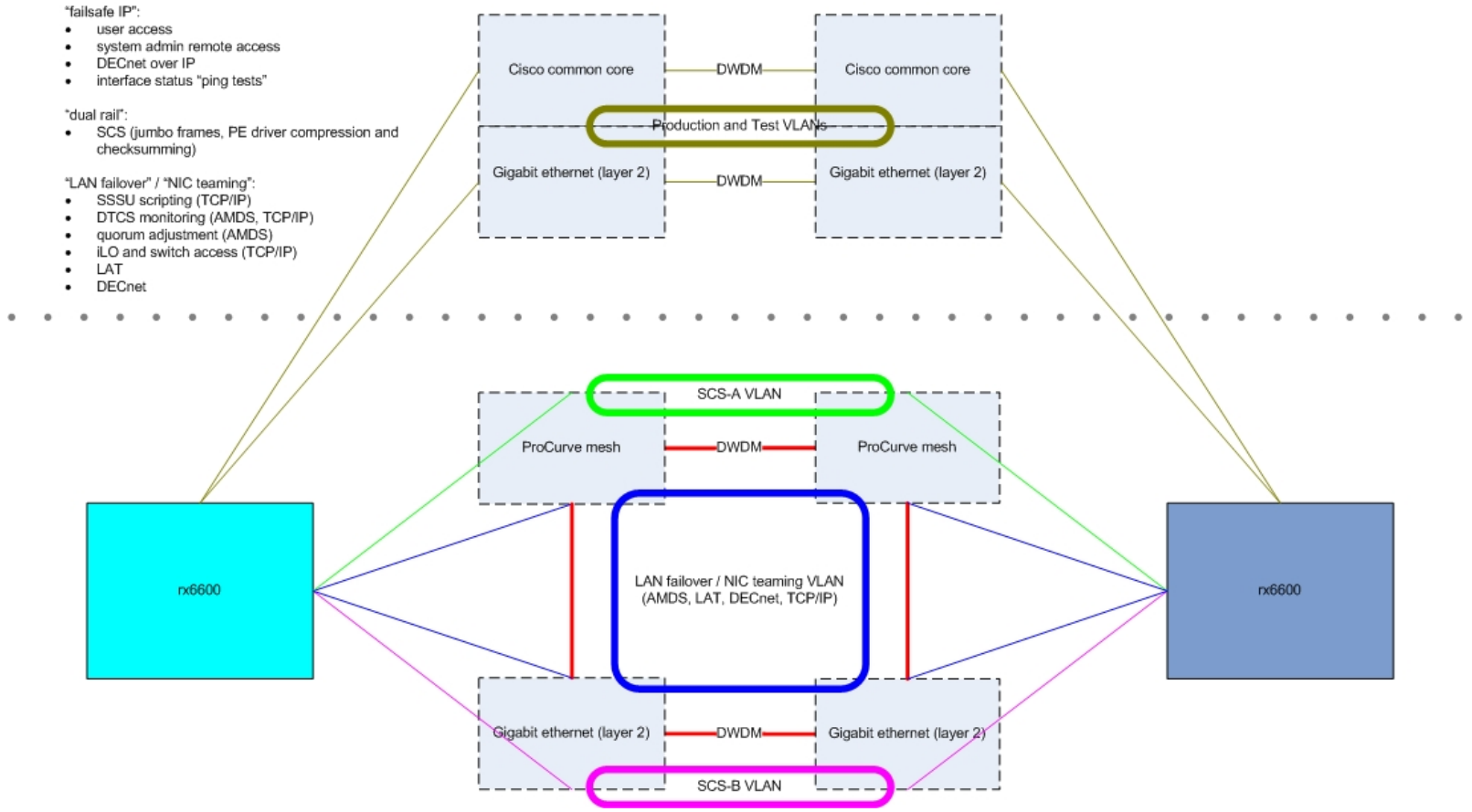
Naming conventions

- Choose your naming conventions very carefully – they are the hardest thing to change later
- Don’t tie nodenames to physical locations
- Choose storage device IDs that help you identify them easily (e.g.: environment, site, array and purpose)
- Choose network addresses and hostnames that make sense in your context.

Network connectivity

- Multiple protocols: SCS, TCPIP, DECnet, AMDS, etc.
- Use LAN failover with multiple NICs for hardware resilience
- Use VLAN tagging and multiple NICs to separate flows
- VL / LL devices map to physical NICs, avoid configuring protocols on physical NICs
- Understand how protocols behave with multiple paths
- Use “service addresses” to separate data flows
- Use SCACP to control which port(s) SCS runs on
- Use LATCP to control which port(s) LAT runs on
- Disable unused protocols on NICs (eg: DECdns, DTSS)

Example data network connectivity



SAN fabrics, HBAs and storage devices

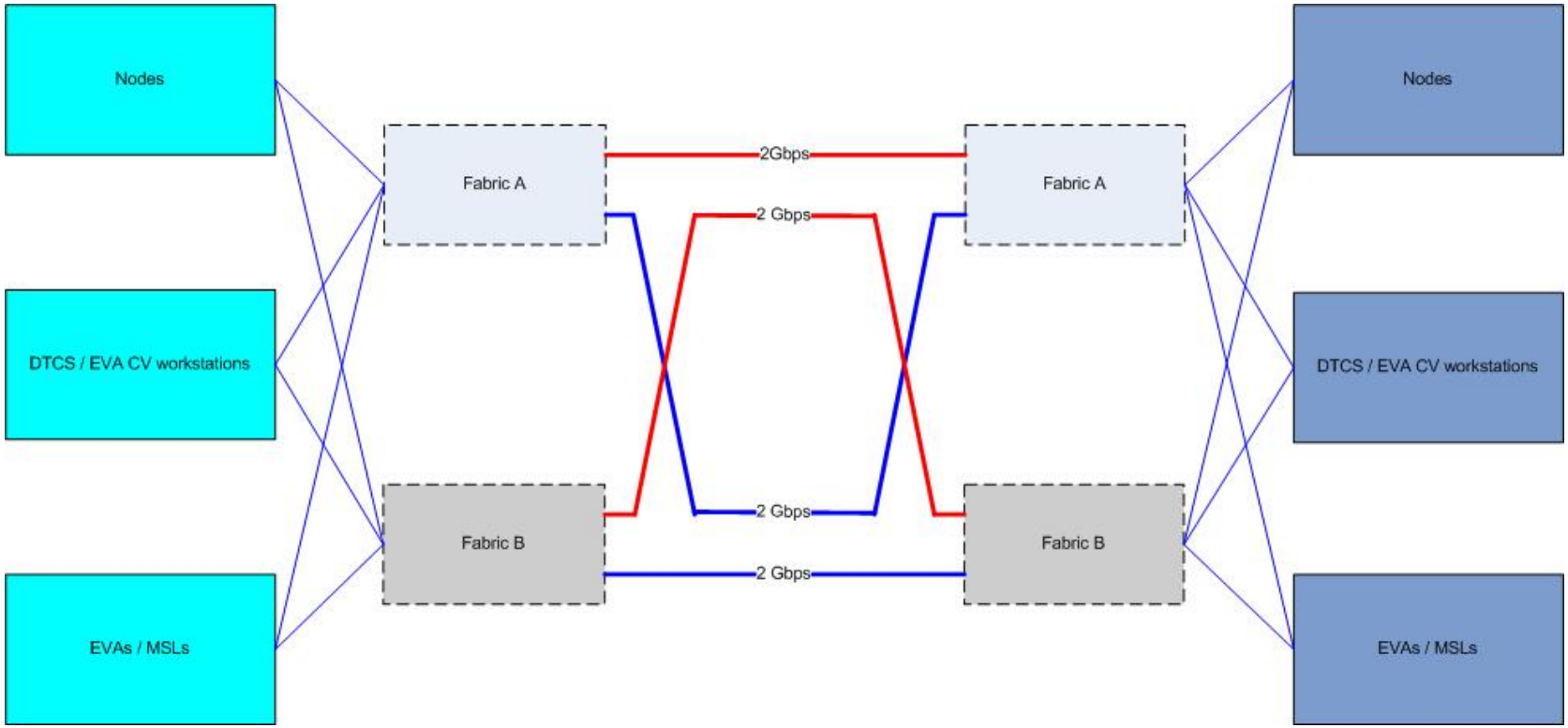
- Zoning (in SAN fabrics):
  - Soft zoning by WWPN, not WWNN
  - Single initiator, multiple target
  - Consider locking port type and speed

- Visibility of storage volumes on arrays by multiple nodes:
  - Presentation (EVA) / Export (3PAR) to HBAs in nodes
  - Identical UDID to all (determines DG device name)
  - Identical LUN ID to all

Multi-site storage

- Use direct path fibrechannel with SAN extension
- Avoid path switching on ISL failure by dual-path connection per fabric with one connection over each ISL per fabric
- Enable MSCP as an alternate path mechanism
- Use mini-copy and mini-merge
- Avoid cross-site booting
- Only mount site-specific discs at their site, even if shadowed to all sites (eg: per-site shadowed system discs)

Example SAN connectivity



Shadowing

- Many shadow sets for performance with multi-path discs
- Small shadow sets to minimise copy/merge time
- Enough arrays per site to always have local source
- Set “site” values to bias reads from site-local storage
- Use minicopy and minimerge for performance
- Set shadow copy buffer size to multiple of 16 (112)
- Multiples of 16 are good for extent size, cluster factor etc.
- Enable DVE on volumes

Array configuration (3PAR, EVA, etc.)

- Use RAID 0+1 (EVA vRAID1) for best performance
- Snaps are only a short-term point in time temporary entity – they can hurt array controller performance
- Clones have better performance, but require more space
- Consider explicit path specification and explicit controller preference for preferred path configuration
- Use SSD for best performance
- Avoid thin-provisioning in IO intensive applications

Quorum and voting

- Is application “cluster aware” or rapid failover ?
- What do you want to happen when a site fails ?

- Votes and expected_votes
- Avoid quorum disc if possible
- Biased voting in symmetric two-site clusters

- Availability manager / DTCS quorum adjustment

- <Ctrl-P> quorum adjustment on Integrity

Log and audit file management

- Fragmentation is a problem worth avoiding
- Use LD containers: write log files to the LD device, then simply move containers to archive.
- Block net$server.log (and others) by creating an empty ;32767 version
- Avoid too many files in a directory

Backup & restore, archive

- Backup – be able to restore quickly if needed
- Archive – remove stale data and retain for reference

- FC tape libraries, drive based encryption
- Off-site copies, VTL (virtual tape library)

- Backup strategy & process:
  - Must be consistent in time
  - Applications must be quiescent
  - Drop shadow set member (or stall IO to member)
  - Use array based clones and snaps

Hardware maintenance and replacement

- Keep firmware up to date – plan sequence to avoid disruption
- FC devices with same UUID but different WWPNs will show up as the SAME device but with extra paths
- Keep systems modular with minimal configuration per node
- Alternate system discs for upgrades / replacements
- Record configurations

Testing, transition and operation

- A mission-critical system hardly ever fails, so we need the people responsible for its operation to have a good understanding and ‘feel’ for the way it works.
- Most failures result from a combination of several things, so are complex and difficult to understand and resolve.
- We need to regularly rehearse and test our procedures and plans to ensure that we stay current
- We must have a representative offline test environment

Project leadership

- Planning for change and transition
- Continuous service
- Managing the people

Planning the changes

- Permissible downtime usually controls the planning
- Risk of disruption usually controls the sequencing
- Never do too many things at once
- Only do one thing at once for critical actions
- Identify any one-way steps as early as possible
- Stay current enough to be supported

Getting things done safely

- Shift workload around to deliver continuous service
- Write everything down in painful detail as a checklist, with anticipated timings
- Rehearse as much as you can
- Do as much as you can in advance
- Make sure everything is tested and ready to go
- Tell everyone what's going on and monitor everything
- Never proceed until you know things are safe
- Don't be afraid to call it off and back out
- Never work alone

Management

- Concentrate on quality of information and decision making
- Be thoughtful, not reactive - do not rush to respond
- Regular briefings, in person, phones off, no e-mail
- Need people to be committed
- Never have one person working on their own
- Find good people, guide them and trust them
- Good administrative support lets people focus on their work
- Good management enables people to get things done

Key success factors

- Strong team of good people
- Collaboration and willingness to share information
- Build a “proof of concept” early on and learn from it
- Minimise complexity
- Structured design
- Clearly defined interfaces
- Document your decisions (what, how, why)
- Make life as easy as you can for those who come after you

Mission-critical systems with OpenVMS

Oracle Rdb Forum 2017

Thank you for your participation

Colin Butcher CEng FBCS CITP
Technical director, XDelta Limited
www.xdelta.co.uk