

The Changing Database

Victoria E Holt

One of the most important features of the database is that it continually changes as changes in the real world are reflected within it. The changes are temporary and the database is continually evolving and growing. Aristotle believed there were four types of changes

- Substance the coming into being and going out of existence or generation and destruction as the database is created and deleted.
- Quality called alteration. A plant alters when it grows green in the sunlight or pale in the dark just as a database alters when data is added or deleted.
- Quantity is growth and diminution. Natural objects typically begin growing and end by diminishing just as a database grows naturally with new data and design etc.
- In respect of place is motion 'nature is a principle of motion and change'. The database is nature itself evolutionary and affected by its environment.

Aristotle argues that every change involves 3 things, there is the state from which the change proceeds, the state to which the change proceeds and the object which persists through the change. "an example, the log, the hot the cold. When a log becomes hot in the grate, it changes from a state of coldness, it changes to a state of hotness, the log itself persists through the change" see figure 1

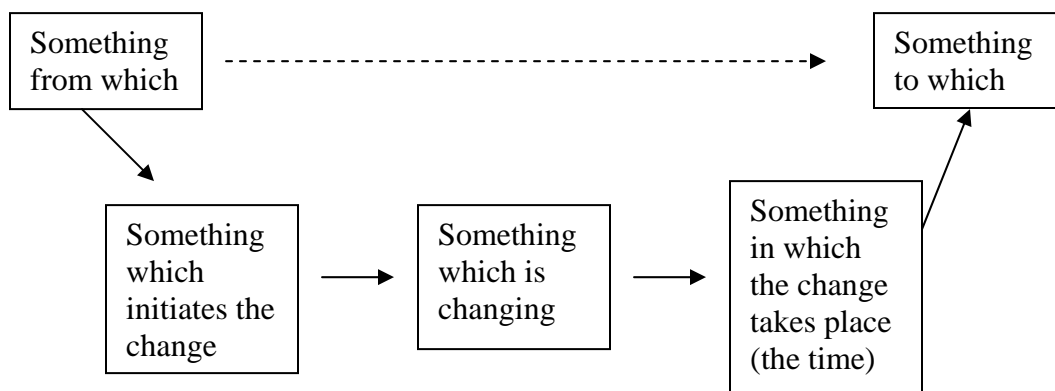


Figure 1 change states based on Aristotle's ideas

Returning to Aristotle he defined "change is the actuality of the potential qua (as in the capacity of) such". This is the potentiality to be changing. This neatly describes the state of the database as "something is in the process of changing whenever it possesses a capacity to change and is exercising that capacity" within a certain time of place and empty space which is infinite.

The database purpose can be affected by the change initiator and the environment and we learn through our actions applied to the system. After a database has been created and deployed to the production environment this is just the beginning of the cycle. Environmental factors are always changing and it is a fact of life that change must take place for businesses to survive and grow. Databases are becoming more and more complex with greater demands on the people's roles and responsibilities needed to manage the changes successfully. Changes that occur can be for new hardware or software and it is necessary to avoid and protect the databases from failure and downtime. Unplanned downtime and planned downtime are the two opposing sides of the coin which need to be balanced with proactive working taking place to try to ensure unplanned downtime happens rarely, see figure 2.

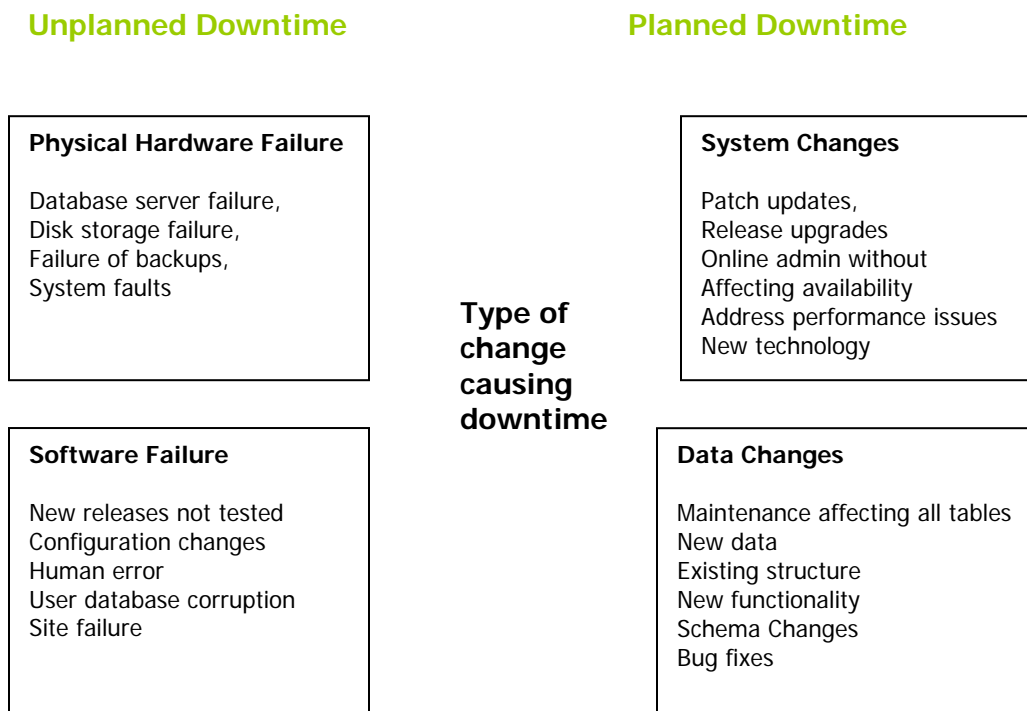


Figure 2 downtime

The hardware needs to be changed and upgraded to meet the continuing demands on resource and to ensure the maximum availability of the architecture. Software changes are inevitable as new pieces of functionality are required to be added or bugs in the current software are found.

If the changes are not monitored and managed problems occur when good code has been over written by changes, broken code is applied during an upgrade, loss of data , confusion over what actual changes have been made etc makes the whole system chaotic. This identifies the types of changes that can cause downtime. The different types of events that affect change of the database system are shown in figure 3.

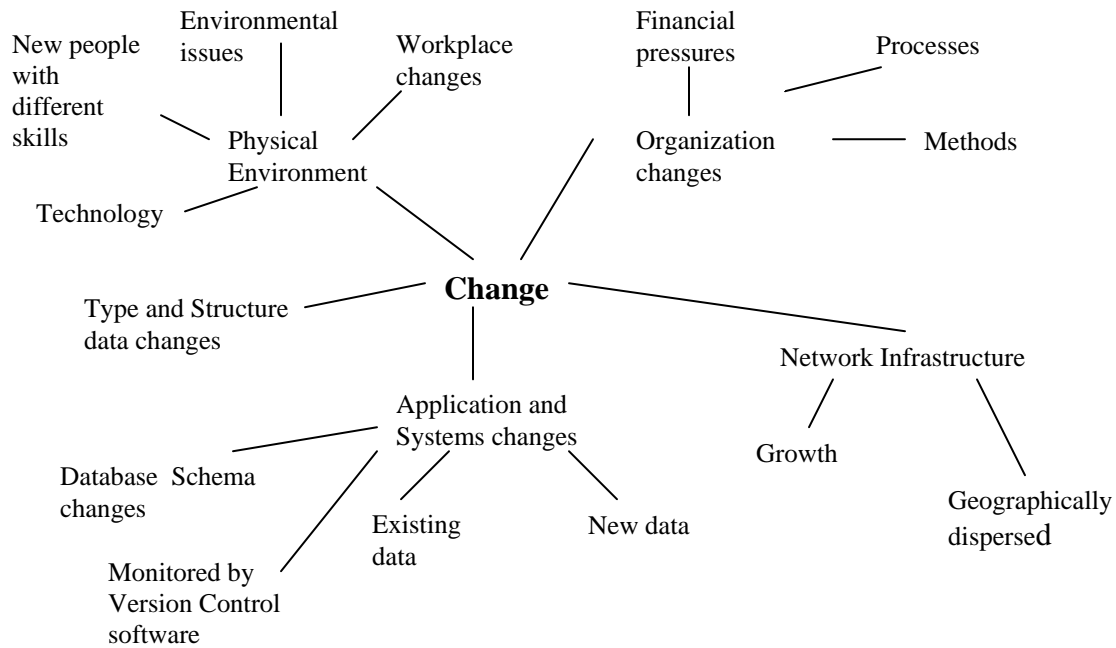


Figure 3 A spray diagram showing the different type of changes

To ensure that the database is continually available it is necessary that any change within the physical database is monitored. Version control software is a useful tool to manage the objects within database. The information held in a database is dynamic and with these changes regular backups are required to ensure the data is recoverable in the case of a disaster or if a change affects the software adversely. There is a need for infallible techniques to make sure the data is reliable and available. To manage changes effectively there are various key factors which all aid in a higher level of accuracy.

- Assessment of the risk and impact to the company of doing the change within the global situation.
- It is necessary to have a detailed plan of changes to reduce the risk and impact of making the change.
- Beware that a simple change may cause a change to be made in another area.
- Design a rollback plan as a contingency in case the change does not go according to plan.
- Attempt to automate the change via a script as this can reduce the impact of human error.
- Have a documented approach to deliver changes.

- It is necessary to plan for the smallest disruption to service as possible to maintain the highest quality of service.
- Look at making proactive changes to eliminate future issues.

The Database Administrator (DBA) perspective

The DBA has a clear perspective and challenge which is to keep the database functional, accurate and secure. To achieve this aim the standard practice for the set up and deployment of databases is to have 3 environments, one which is for development, one for quality testing and the production environment. In some cases there are also servers for disaster recovery. The set up is shown in figure 4. To ensure the changes are correctly tested and deployed the code is moved between the environments.

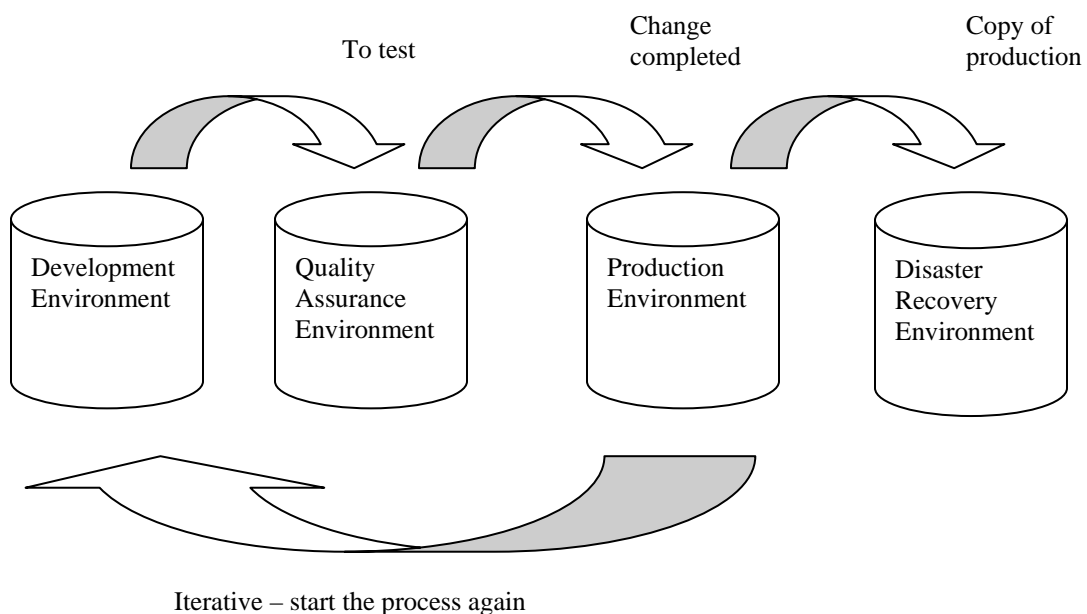


Figure 4 The database environment

The DBA is the custodian of database changes but not always the requestor. Databases are all configured differently so it is difficult to support efficient changes at the lowest level equally amongst databases. The complexity of current databases is on the rise and the need to deliver functionality in a timely and accurate manner allowing for information which can have an audit trail is more important.

Examples of the changes at the database level can be seen in figure 5

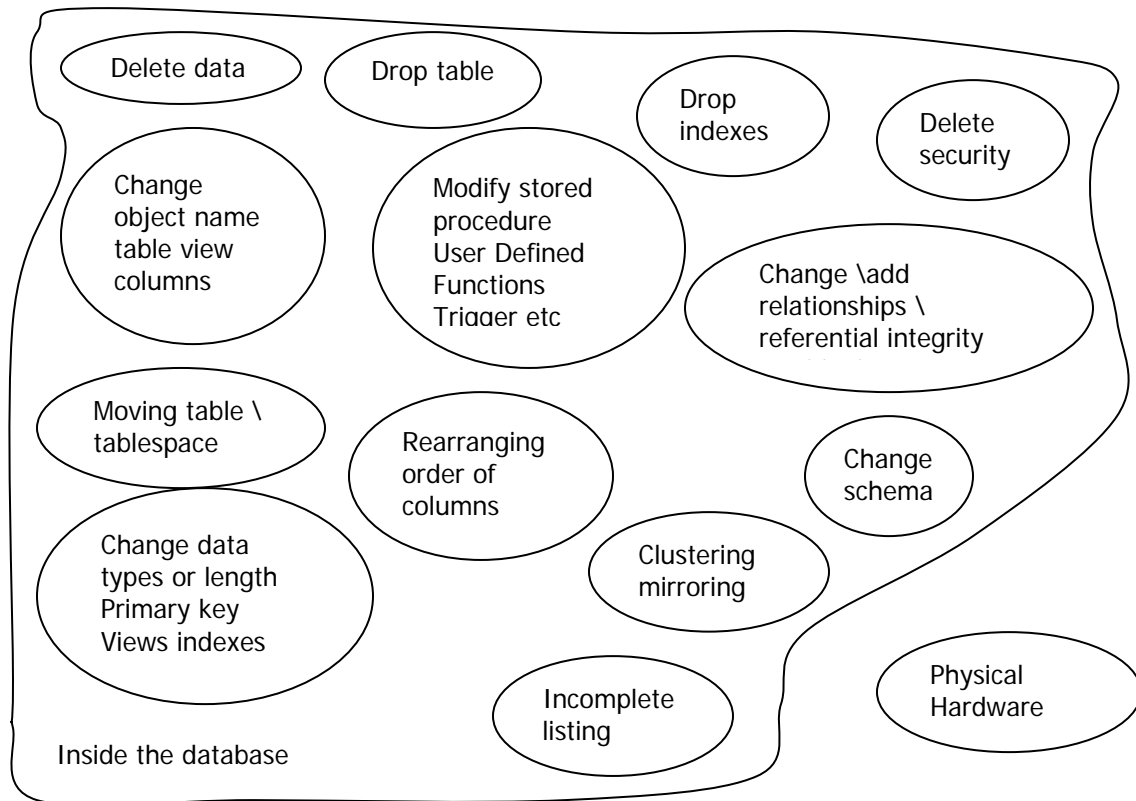


Figure 5 A systems map of the types of database changes

Change perspective

A change manager for database more often than not, now follows the ITIL framework. The IT infrastructure Library (ITIL) defines the goal of the change management process this way:

to ensure that standardized methods and procedures are used for efficient and prompt handling of all changes, in order to minimize the impact of change-related incidents upon service quality, and consequently improve the day-to-day operations of the organization.

This framework

- ensures standards methods, processes and procedures are used for all changes.
- facilitates efficient and prompt handling of all changes
- maintains the proper balance between the need for changes and the potential detrimental impact of changes

so that change become a monitored process. With requirements for new functionality on the increase, services timely delivered and 24 / 7 availability becoming much more of a

reality the demand for integrity, audit changes and corporate governance become much more to the fore. This brings with it the need to adopt procedures and best practice to ensure change affects the smooth running of services as little as possible.

There are various other tools out there which help with the change process. A defined philosophy or strategy for processes is key. Figure 6 demonstrate the ACT PLAN CHECK DO tool

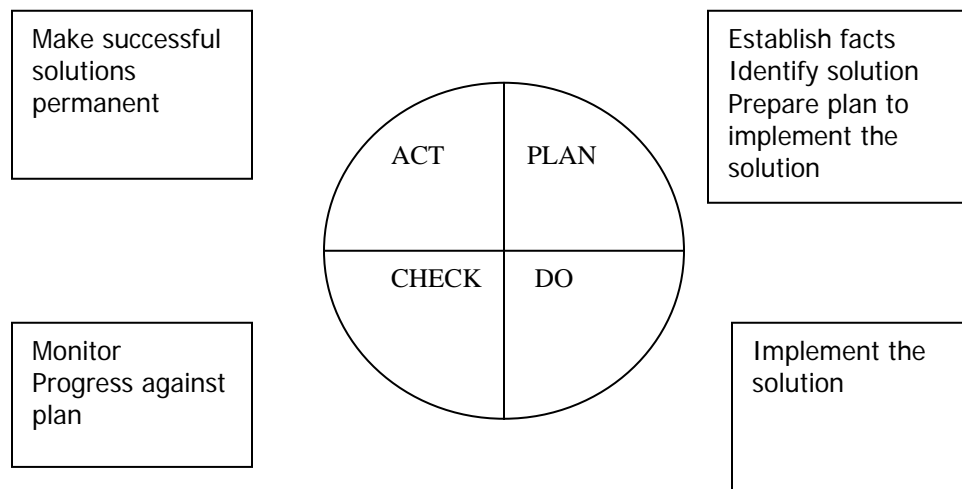


Figure 6 Change cycle taken from Create that Change

This makes you consider the change system. Thinking holistically about the change and, the organisational system so that what affects the system helps manage the change. Complete database management control is not a reality for the DBA. So having a change framework which allows control and provides a tool for rollback in case the change goes wrong is a key to providing a highly available database. Kolb's model highlights this experience with reflection in figure 7.

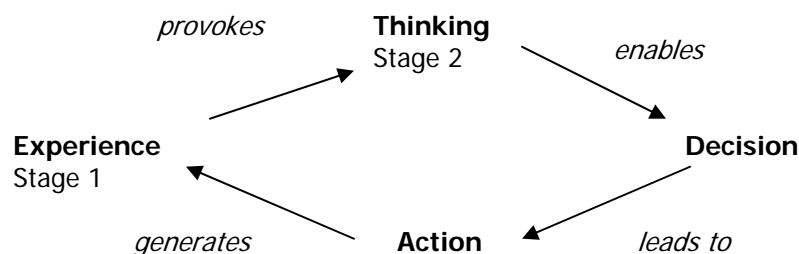


Figure 7 Kolb's model

The change control process should be planned and well documented and should disclose that:-

- All proposed changes are fully documented
- All changes are tested to ensure that the functionality is not affected
- The likely impact of the change on the production system identifying whether the level of impact on the production system is low, medium or high risk.
- Management has approved the change
- The change is scheduled so changes don't conflict and to enable ease of identification if problems occur
- There is a rollback plan

Obviously there are times when emergency changes are needed for example when a previous deployment had failed but wasn't noticed immediately or if an environmental factor like a security attack has happened. Emergency changes do not follow all of the above but should attempt to cover the issues.

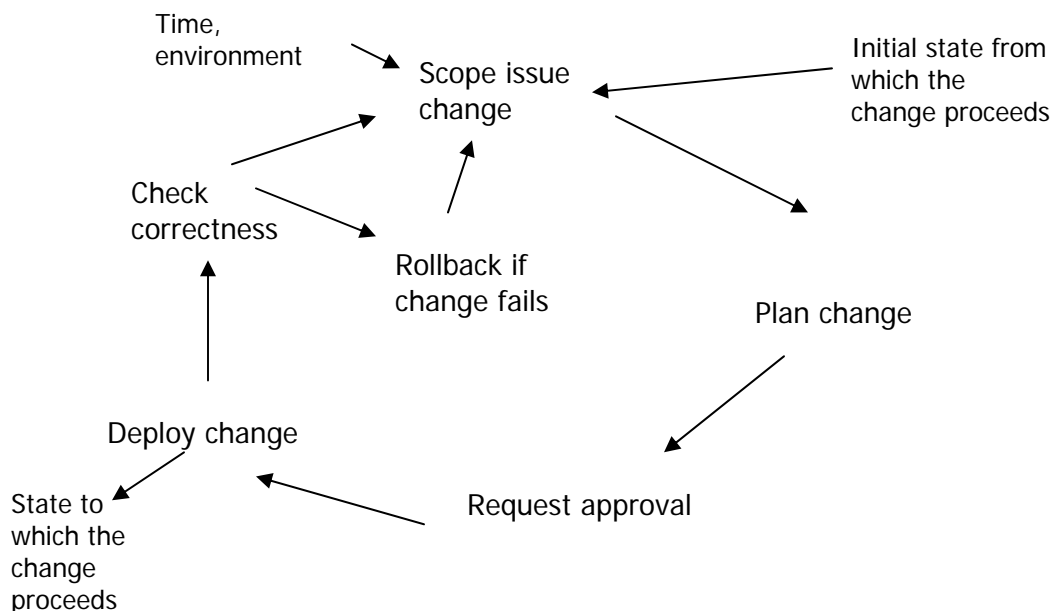


Figure 8 Change loop for the database

The actual procedure is shown in the change loop in figure 8.

Key factors that need to be addressed which could cause the failure of this system are

- Unit testing, regression testing and new functionality testing needs to be carried out to ensure there are no errors or broken code.

- Scripting changes ensures the change is auditable and repeatable and version controlled.
- Auditable database changes are very important to ensure that mistakes can be quickly identified if there is an incident.
- Change management is responsible for managing the change process for the evolving database.
- Ensure all procedures associated with the running, support and maintenance of live systems are documented.

References

7 Steps to database change control Michelle Poollet March 2005 SQL Server Magazine

Change Management (ITSM) From Wikipedia, the free encyclopedia
Retrieved from "http://en.wikipedia.org/wiki/Change_Management_%28ITSM%29"

Managing Database Change

By Craig S. Mullins from Database Trends October 2000.

http://www.craigsmullins.com/dbt_1000.htm

Data Change Management Best Practices – Achieving an Automated Approach and Version Control Innovartis white paper

ITSMF IT Service Management A Companion to the IT infrastructure library

Create that change ready made tools for change management edited by Steve Smith
quest toolbox series

Managing Database Change Chpt 8 SQL Server 2000 Resource Kit

Always available by Alan Joch Oracle Magazine Jan/Feb 2007

Jonathan Barnes Aristotle a very short introduction