- Software Configuration Management – SCM Release Management
- James Brabyn

TABLE OF CONTENTS

Mission and Scope

Mission

The focus of Release Management is the protection of the live environment and its services through the use of formal procedures and checks.. Release Management takes a holistic view of a change, or group of changes to a service to ensure that all aspects of a release, both technical and non-technical are considered. Release Management should be performed with source controlled change and incident management policies and procedures

Scope

Release Management undertakes the planning, design, build, configuration and testing of hardware and software to create a set of release components for a live environment. Activities also cover the planning, preparation and scheduling of a release to many customers and locations. Release Management is heavily dependant of Change Management and Configuration Management to ensure that the shared CMDB is kept up-to-date following changes implemented by new releases, and that the contents of those releases are stored in the DSL.

SCM / Release Management Overview

- Companies who seek substantial productivity gains use SCM tools policies and procedures to reduce development and maintenance costs. They deliver quality products on time and have laid a foundation for on-going business success ensuring their first success can be replicated enterprise-wide.

- Software Configuration Management (SCM) is a collection of tools and policies to develop and maintain a software system.

- .A SCM system is a means to achieve higher quality systems while minimizing development and maintenance efforts. This is made possible by way of following a well-defined set of automated policies and standards designed to:
- Minimize risk of errors to the various components of the software system
- Eliminate confusion among the different members of the project team as to what each individual is expected to do and how the task is expected to be performed.

Release Management Overview

- A repeatable and auditable process to move applications into production
- Adoption of Software Configuration Management (SCM) Practices (source, change and release management)
- Collision avoidance among applications and between applications
- Part of the Version Management and Change Management process and a result of the Release Planning Process
- Will involve scheduled major releases, scheduled minor releases, and emergency releases.
- Release policy and planning
- Release design and build
- Release identification
- Definitive Software and Hardware Store
- Build Management
- Release health checks (pre-release / post-release checks)
- Release implementation
- Back-Out plans
- Additional DR level

Benefits of Release Management Adoption

- Greater success rate in the Release of hardware and software
- Improved quality of delivered service (fewer back outs / emergencies to fix turnovers)
- Integrated repeatability and traceability of releases (sources / changes)
- Regulatory Compliance
- Consistency in the Release processes of the software environments
- Minimization of the disruption of the service to the business through synchronization of Releases within packages involving software components from different platforms and environments
- Assurance that the software in live use is of good quality, because the Releases are built from components that have been subject to quality control and effective testing
- Stable test and live environments, because Changes are normally combined into Releases and so there should be fewer individual implementations
- Better use of user resources because of combined efforts when testing new Releases - this also means that it will be easier to justify the cost of system-wide end to end and regression testing
- Redundancy of tools VSS, RCS, Perforce

Benefits of Release Management Adoption (Contd…)

- Better expectation setting within the organization on publication of a Release schedule in advance (ability to book release space)
- An ability to absorb high rates of Change to the live systems, effectively and without adversely affecting IT
- A complete record (or audit trail) of Changes to the live environment is maintained, both of software distributions and of hardware Changes
- Error reduction through the controlled Release of software to the live environment
- Savings in support costs through the ability to maintain consistent software over a large number of locations
- Longer term there will be fewer Releases (fewer risk windows)
- Ability to check / test a collection of changes together in a Production like environment
- Financial Cost reduction due to Bulk License cost of single SCM tool (through redundancy of other tools)
- ITIL Change / Incident and Release compliance
- UAT testing will be scheduled and planned
- Development / Test tool integration

Release Management Adoption - Cons

- Financial Costs (SCM Tool licence seats, training costs, SCM server, Consultancy, tool support by HD)
- Initial reduced development speed due to extra tool / procedures introduction
- Developer Reluctance to change
- Increased stages in the Development process (longer release cycle)
- Business acceptance of Release Schedules
- Upkeep and Planning of Release Schedules
- SCM tool upgrade / maintenance time
- Dedicated SCM Resource or team
- Complexity of release planning with multiple environments
- Costs of POC and tool evaluation and Selection
- Version / Change Migration to selected tool
- Parallel running of SCM systems
- SCM DB replication / proxy for distributed development sites around world

Release Management Activities

- The Release discipline is tightly linked to the management of the software test, user acceptance and pre-production acceptance processes.

- Release Management activities include:
  - Release policy and planning
  - Release design, build and configuration
  - Rollout / implementation planning and testing
  - Sign-off of the Release for implementation
  - Communication, preparation and training
  - Release, distribution and the deployment of software.

Release Schedule

The Release Schedule will consist of dates for Major Releases and Minor (support) Releases. Emergency Releases will by definition be impossible to schedule. However, the criteria by which a change is classified as an Emergency should be clearly specified;

Major Releases will be scheduled as evenly as possible throughout the year. There will be a minimum cycle of 8 weeks, allowing for about 6 Major Releases each year.

Minor Releases will be scheduled regularly, with a maximum of 2 weeks between releases.

Emergency releases will be handled on a case-by-case basis, and only if the modification of software cannot be postponed by identifying a work-around until the change can be fitted into a scheduled Minor Release.

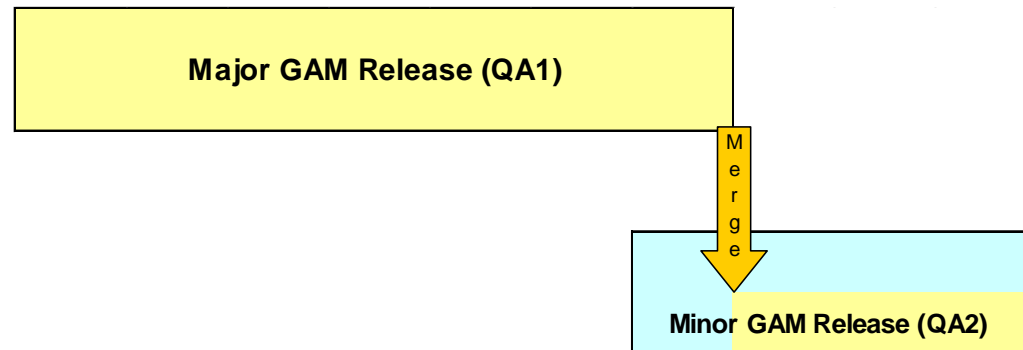|	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8	Month 9	Month 10	Month 11	Month 12
Major			■			■			■			■
Minor												
Emergency												

Major Releases

- The simultaneous deployment of numerous interrelated applications
- Release Management Planning is required for Scheduled Major Releases
- Annual Planning and the Change Management processes contribute to Release Management Planning
- Annual Planning will drive out the dates for major releases
- Change Management will closely control changes to applications, operating environments, user interaction, and support during a major release.

Minor Releases

- The simultaneous deployment of numerous non-interrelated applications
- Release Management Planning is required for Scheduled Minor Releases
- Incident Management and the Change Management processes contribute to Release Management Planning
- A schedule of dates for minor releases will be published.
- Change Management will closely control changes to applications, operating environments, user interaction, and support during a minor release.

Major GAM Release (QA1)
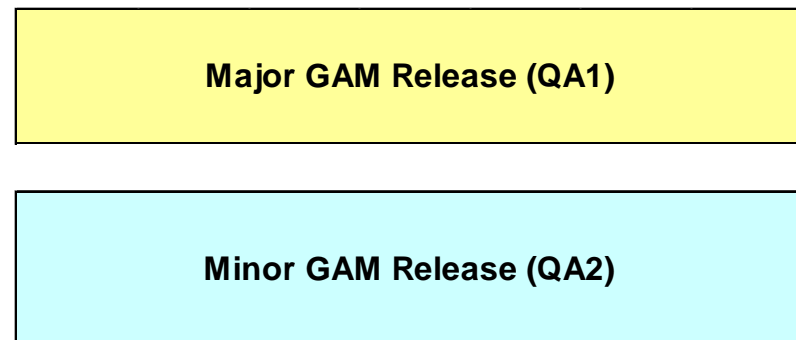
Merge

Minor GAM Release (QA2)

Release Scenarios

Release 1 is being tested and then Release 2 gets introduced, using the second QA environment (QA2) we can revise Release 2 on Release 1 and have both release continue testing. This assumes that there is shared code

Advantages – Good use of test resources, Allows Rapid Deployment of multiple releases

Disadvantages – R2 will have to be revised if test defects are found in R1

Major GAM Release (QA1)

Minor GAM Release (QA2)

Release 1 and Release 2 are both being tested in parallel, whatever Release completes testing first is permitted to go live

Advantages – Good use of test resources, Allows Rapid Deployment of multiple releases

Disadvantages – the Release that has to go live second will need to be revised (re-baselined) on the first release if there is shared code, causing extra testing effort

Process Maturity

- **Process Maturity – Roadmap for 2-3 years**

  - Complete Inception to Deployment (end to end) Software Configuration Management
  - Formal Change Control with Integrated Version Control
  - Dynamic reporting for project / ticket (job) progress
  - Metrics collated (estimated vs. actual)
  - Visibility of proposed projects via a schedule

SCM Implementation Activities

- Business Requirements
- Existing process evaluation
- Technical Scope
- SCM Tool demonstrations
- Proof of Concept Evaluations
- SCM tool trial
- Acceptance
- Tool Development
- Process Development and mapping
- Tool / Procedure training
- Phased Migration of Technologies (version / change requests / workflows)
- Parallel Development / Release strategy
- SCM tool control
- Redundancy of tools

Software Configuration Management Tool integration

- **SCM Tool Integration**

    Helpdesk tools and Test tracking tools are linked via custom APIs to allow SCM tools and process to provide true end to end SCM.

Tools

Tools

The current IT SCM tool, Perforce, could be used to version control Data Modules and GAM digital assets, version management should then be followed by change and release management from within the SCM suite of tools this should be a phased rollout

Version Control within Perforce for DM and GAM should be started within 5-6 months.

PWM – SCM / Release Management

Questions ?