**Approach Document for Control-M Job Failure Dashboard with Integrated Log and Incident Visibility**

**Background & Problem Statement**

Currently, in the event of a Control-M job failure:

* An **incident is auto-generated**, but **without the actual reason or error log**.
* The RTB (Run-The-Bank) team must manually log into the Control-M console, search for the job, and analyze logs to identify the failure reason.
* This results in:
  + Delayed incident resolution
  + Increased manual effort for RTB
  + Frustration for CTB (Change-The-Bank) teams who lack visibility for proactive fixes

**Proposed Solution**

We propose to develop a **Control-M Failure Monitoring Dashboard** with the following features:

* ✅ **Live Dashboard** displaying failed Control-M jobs with:
  + Job Name (Tracker)
  + Current and Previous State
  + Time, Order ID, Run Number, Elapsed Time, CPU Time
  + **Error Snippet** (first failure line extracted from sysout)
  + **Downloadable Full Log**
  + **Linked ServiceNow Incident Number** with direct URL
* ✅ **Automatic Extraction of Sysout Logs**  
  The sysout logs will be copied, uniquely tagged with the business date and job run details, and made available for download.
* ✅ **ServiceNow Incident Integration**  
  The dashboard will pull incident numbers via ServiceNow APIs and display alongside job details.

**Benefits & Value Proposition**

| **Current Scenario** | **Post Implementation** |
| --- | --- |
| Manual log checking by RTB | Logs readily available on dashboard |
| RTB unaware of immediate failure reason | RTB and CTB both can analyze failures instantly |
| Higher incident handling time | Reduced turnaround time for incident resolution |
| Repetitive incidents with no corrective action | CTB can proactively identify automation opportunities |
| Poor incident-to-problem linking | Better incident tagging and root cause identification |

**Estimated Effort & Plan**

| **Activity** | **Duration** | **Notes** |
| --- | --- | --- |
| Development & Unit Testing | **1 Week** | Flask-based dashboard, log parser, ServiceNow integration |
| Deployment to Production | Same Week | As per release schedule |
| Monitoring & Feedback Collection | **1 Month Post-Deployment** | Continuous monitoring of usage and impact |
| UAT / Feedback-based Enhancements | Post 1 Month | Optional, based on RTB/CTB feedback |

**Expected Impact After Deployment**

* 🎯 **Reduction in incident analysis time by RTB**
* 🎯 **Improved collaboration between RTB and CTB for root cause analysis**
* 🎯 **Higher quality incidents with actionable information**
* 🎯 **Better alignment with Problem Management by accurate tagging**

**Conclusion & Recommendation**

We believe this solution will directly address the recurring pain points of both RTB and CTB teams, enhancing operational efficiency and reducing incident resolution time.  
We recommend **approving this initiative** for a quick development cycle, followed by a monitored production rollout.

**Next Steps**

* ✅ Stakeholder approval
* ✅ Finalize system access for sysout and ServiceNow APIs
* ✅ Development kickoff post-approval