**Application Runbook for Issue Resolution**

**1. Introduction**

This runbook provides standardized procedures for addressing application-level failures, system issues, and job failures. It ensures consistency in resolution, minimizes downtime, and enhances overall system reliability.

**2. Common Failures & Resolutions**

**2.1 Network Configuration Failure**

* **Issue:** Misconfigured firewall rules prevent connectivity.
* **Impact:** High - Users unable to access internal applications.
* **Resolution Steps:**
  1. Verify firewall rules and update access control lists.
  2. Check network logs for dropped packets.
  3. Restart affected network services.
  4. Confirm resolution with a connectivity test.

**2.2 Transaction Dispute Handling Delay**

* **Issue:** Banking front desk system fails to process transaction disputes.
* **Impact:** Medium - Customers experience delays in resolving disputes.
* **Resolution Steps:**
  1. Verify dispute processing logs for errors.
  2. Check if external banking APIs are responding.
  3. Escalate to L2 support for manual intervention if needed.

**2.3 ATM Malfunction**

* **Issue:** ATM machine fails to dispense cash intermittently.
* **Impact:** High - Affects customer transactions.
* **Resolution Steps:**
  1. Check ATM logs for error codes.
  2. Reboot the ATM and reinitialize the software.
  3. Verify cash cassette status and refill if necessary.
  4. If unresolved, schedule a hardware inspection.

**2.4 Patch Deployment Failure**

* **Issue:** IT patch deployment fails due to missing dependencies.
* **Impact:** High - Security vulnerabilities remain unpatched.
* **Resolution Steps:**
  1. Check logs for missing libraries or dependency failures.
  2. Roll back partially deployed patches.
  3. Update dependency list and retry deployment.
  4. Validate patch installation with test cases.

**2.5 Code Deployment Failure**

* **Issue:** New code deployment results in a failed application startup.
* **Impact:** High - Production system unavailable.
* **Resolution Steps:**
  1. Review deployment logs for errors.
  2. Roll back to the previous stable version.
  3. Verify environment variables and dependencies.
  4. Perform a fresh deployment with fixes.

**2.6 Autosys Job Failure**

* **Issue:** Scheduled job execution stuck in pending state.
* **Impact:** Medium - Data processing delayed.
* **Resolution Steps:**
  1. Check Autosys logs for job dependencies.
  2. Restart dependent jobs if they failed.
  3. Clear job queue and retry execution.
  4. Escalate to L3 if issue persists.

**2.7 Database Connection Timeout**

* **Issue:** Application fails to connect to the database due to timeout errors.
* **Impact:** High - Data-driven processes are disrupted.
* **Resolution Steps:**
  1. Check database logs for connection issues.
  2. Restart database services if necessary.
  3. Verify network connectivity between the application and database server.
  4. Increase connection timeout settings in the application configuration.

**2.8 API Gateway Failure**

* **Issue:** API requests fail due to authentication or routing issues.
* **Impact:** Medium - Integration between services is disrupted.
* **Resolution Steps:**
  1. Verify API gateway logs for errors.
  2. Check authentication credentials and token expiry.
  3. Restart API gateway services.
  4. Update routing rules if there are misconfigurations.

**2.9 Storage Capacity Exhaustion**

* **Issue:** Storage disk runs out of space, preventing data writes.
* **Impact:** High - Application crashes due to lack of storage.
* **Resolution Steps:**
  1. Identify large files and clear unnecessary logs.
  2. Expand storage capacity if needed.
  3. Implement auto-archiving for old data.
  4. Configure monitoring alerts for storage thresholds.

**2.10 Load Balancer Failure**

* **Issue:** Load balancer stops distributing traffic effectively.
* **Impact:** High - Some users experience downtime.
* **Resolution Steps:**
  1. Check load balancer logs for errors.
  2. Restart load balancer service.
  3. Verify backend server health checks.
  4. Update routing configurations if needed.

**3. Application-Specific Failures & Resolutions**

**3.1 Data Duplication in Reports**

* **Issue:** Duplicate records appearing in reports.
* **Impact:** Medium - Incorrect business decisions due to data inaccuracy.
* **Resolution Steps:**
  1. Identify duplicate entries using SQL queries.
  2. Check ETL scripts for data duplication logic.
  3. Implement deduplication mechanisms in reporting layers.
  4. Validate data integrity post-fix.

**3.2 Missing File During Processing**

* **Issue:** Required files are not found during batch processing.
* **Impact:** High - Incomplete or failed data processing.
* **Resolution Steps:**
  1. Verify file existence in the expected directory.
  2. Check file transfer logs for errors.
  3. Implement retry mechanisms for file retrieval.
  4. Notify stakeholders if file remains unavailable.

**3.3 Performance Degradation in Web Application**

* **Issue:** Slow response times and increased page load durations.
* **Impact:** High - Poor user experience and potential revenue loss.
* **Resolution Steps:**
  1. Analyze server performance metrics.
  2. Optimize database queries and indexing.
  3. Implement caching strategies.
  4. Scale infrastructure resources if necessary.

**3.4 Partial Data Load in ETL Process**

* **Issue:** ETL jobs load incomplete data into the warehouse.
* **Impact:** Medium - Business reporting discrepancies.
* **Resolution Steps:**
  1. Identify and resolve failed records in staging tables.
  2. Check ETL job logs for error messages.
  3. Restart failed ETL processes.
  4. Implement error-handling mechanisms in ETL scripts.

**3.5 Data Mismatch Between Systems**

* **Issue:** Data discrepancies between different applications.
* **Resolution Steps:**
  1. Compare data snapshots from both systems.
  2. Identify and correct sync failures.
  3. Implement reconciliation scripts.

**3.6 API Response Latency Increase**

* **Issue:** API responses taking longer than expected.
* **Resolution Steps:**
  1. Analyze API performance metrics.
  2. Optimize queries and caching.
  3. Scale backend resources if needed.

**3.7 Unexpected Application Crash**

* **Issue:** Application terminates unexpectedly.
* **Resolution Steps:**
  1. Check logs for crash reports.
  2. Debug error stack traces.
  3. Apply fixes and deploy patched versions.

**3.8 Incorrect Data Formatting in Reports**

* **Issue:** Reports displaying incorrect formatting.
* **Resolution Steps:**
  1. Validate data source formatting.
  2. Update report generation scripts.
  3. Test report output against expected results.

**3.9 Failed Background Job Execution**

* **Issue:** Scheduled background jobs failing.
* **Resolution Steps:**
  1. Check job logs for failure reasons.
  2. Restart failed jobs and monitor execution.
  3. Implement retry mechanisms.

**3.10 Authentication Failure in User Login**

* **Issue:** Users unable to log in due to authentication errors.
* **Resolution Steps:**
  1. Verify credentials and authentication service logs.
  2. Reset user access tokens.
  3. Restart authentication services.

**4. Monitoring & Preventive Measures**

* **Real-time application and system monitoring tools are in place.**
  + **Prometheus and Grafana are used for real-time metric visualization.**
  + **ELK Stack (Elasticsearch, Logstash, Kibana) provides centralized logging.**
  + **Datadog, New Relic, and AppDynamics monitor application performance.**
* **Automated alerts are configured for critical failures.**
  + **Nagios and Zabbix generate infrastructure alerts.**
  + **PagerDuty and Opsgenie handle automated incident escalation.**
  + **Amazon SNS and Slack/MS Teams notify teams of urgent issues.**
* **Regular maintenance processes are in place.**
  + **Ansible and Chef automate security patching.**
  + **Splunk and Dynatrace are used for periodic system health checks.**
  + **Oracle AWR and MySQL Performance Schema optimize database performance.**
* **System performance benchmarking and optimization is ongoing.**
  + **Apache JMeter and Gatling conduct load testing.**
  + **Locust and k6 perform stress testing.**
  + **PT-Query-Digest and SQL Server Profiler optimize database queries.**

**5. Escalation & Support Process**

**L1 Support**

* **Role: Initial troubleshooting and issue triaging.**
* **Contact: support-l1@example.com | Phone: +1-800-555-1111**
* **Availability: 24/7 Service Desk**

**L2 Support**

* **Role: Advanced issue resolution, in-depth system analysis.**
* **Contact: support-l2@example.com | Phone: +1-800-555-2222**
* **Availability: Business Hours & On-Call Support**

**L3 Support**

* **Role: Code-level debugging, infrastructure fixes.**
* **Contact: support-l3@example.com | Phone: +1-800-555-3333**
* **Availability: On-Demand with Vendor Escalation**
* **Escalation to vendor support for unresolved critical issues.**
* **Additional Support Channels:**
  + **Slack Channel: #tech-support**
  + **On-Call Engineer: Available 24/7 for emergency escalations.**