VAT Flowchart Reference Guide
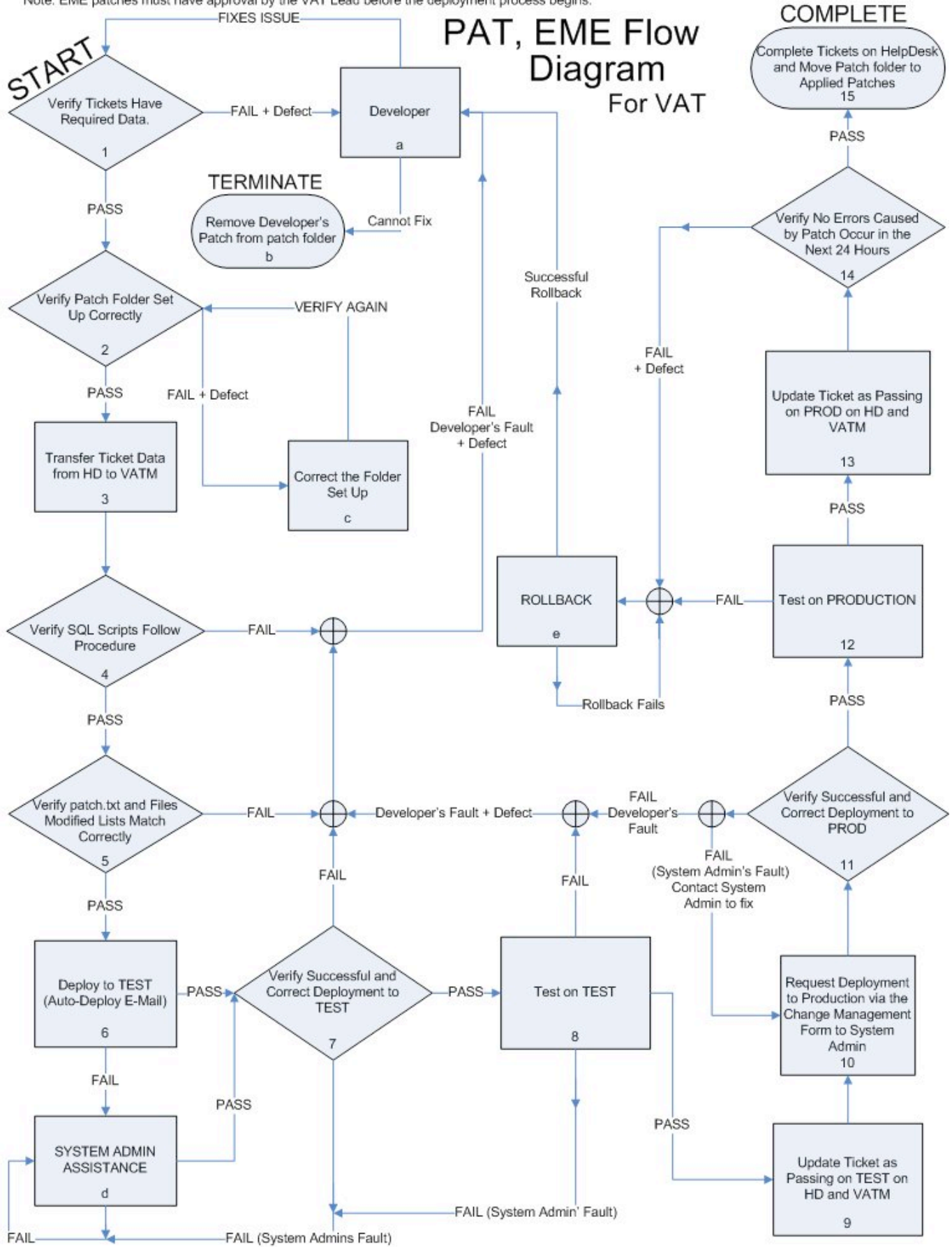
**Patch Flow
(If EME patch, verify that VAT Lead has approved of the EME patch)**

Version 1.1 – 01/09/2006

Note: EME patches must have approval by the VAT Lead before the deployment process begins.

PAT, EME Flow Diagram
For VAT

START

COMPLETE

FIXES ISSUE

Verify Tickets Have Required Data.
1

FAIL + Defect →

Developer
a

Cannot Fix →

TERMINATE

Remove Developer's Patch from patch folder
b

PASS ↓

Verify Patch Folder Set Up Correctly
2

VERIFY AGAIN

Successful Rollback

FAIL + Defect

PASS ↓

FAIL + Defect

Correct the Folder Set Up
c

Transfer Ticket Data from HD to VATM
3

FAIL Developer's Fault + Defect

Complete Tickets on HelpDesk and Move Patch folder to Applied Patches
15

PASS ↑

Verify No Errors Caused by Patch Occur in the Next 24 Hours
14

Update Ticket as Passing on PROD on HD and VATM
13

PASS ↑

Verify SQL Scripts Follow Procedure
4

FAIL → ⊕

ROLLBACK
e

← FAIL —

Test on PRODUCTION
12

Rollback Fails

PASS ↓

PASS ↑

Verify patch.txt and Files Modified Lists Match Correctly
5

FAIL → ⊕ Developer's Fault + Defect → ⊕ FAIL Developer's Fault → ⊕

Verify Successful and Correct Deployment to PROD
11

FAIL

PASS ↓

FAIL (System Admin's Fault) Contact System Admin to fix

Deploy to TEST (Auto-Deploy E-Mail)
6

PASS →

Verify Successful and Correct Deployment to TEST
7

PASS →

Test on TEST
8

Request Deployment to Production via the Change Management Form to System Admin
10

FAIL ↓

PASS ↑

PASS ↓

SYSTEM ADMIN ASSISTANCE
d

PASS →

FAIL ↓

PASS →

Update Ticket as Passing on TEST on HD and VATM
9

FAIL

FAIL (System Admins Fault)

FAIL (System Admin' Fault)

Patch Flow

(If EME patch, verify that VAT Lead has approved of the EME patch)

1. Verify Tickets Have Correct Data – Each ticket is required to have a list of the files modified for the patch and a testing method for the VAT.

   > 1.1. EX: **By Developer X on Sep 12, 2005 at 05:45 PM**
   >
   > PATCH ready for deployment with PAT20050919.
   >
   > Files Modified:
   > webroot\wwwadmin\apps\webmanager2\add_edit\dsp_add-editPage_Title.cfm
   >
   > Testing Method:
   > 1. Edit PID 1421 and confirm that you can update the page properties, but can only hit the update button once.
   > 2. Also, make sure you can only hit the save button once when you create a new page.

   1.2. If for some reason the ticket does not have the above information, VAT should contact the developer (see step "a").

2. Verify Patch Folder is Setup Correctly

   2.1. Named Correctly – the Patch Folder located in 1416_a$ must be named in the following format, TYPYYYYMMDD. Where TYP is the type of patch being either PAT, EME, or REL, and YYYY being the Year, MM being the Month, and DD being the day.

   2.2. Ticket.txt exists in the TYPYYYYMMDD folder. The tickets must be comma delimited without any spaces. EX: 277930,276470,278110

   2.3. No files in the TYPYYYYMMDD folder except for ticket.txt, readme.txt, patch.txt, .config folder, webroot folder (exceptions may include other folders such as seminary, lcabulldogs, thelight).

   2.4. All folders are name lowercase (except for the root folder TYPYYYYMMDD).

   2.5. If any of the above procedures are not followed exactly, go to step "c."

3. Transfer Ticket Data from HD (HelpDesk) to VATM (Verification and Testing Manager) – The VAT Tester will open a ticket from the Patch deployment user on the helpdesk and copy the ticket number, problem/issue, files modified, and the test cases into the VAT Manager program.

4. Verify SQL script follows procedure
   4.1. No tabs.
   4.2. No GO's are to be used in the script, use ;'s instead.
   4.3. Creates and alters need to have their own separate scripts.
   4.4. The SQL files must be named so that the creates come before the alters. EX: EZ1Create.sql, EZ2Create.sql, EZ3Alter.sql. Auto-deployment will run them in alphabetical order.
   4.5. Must have a use statement on top of every file designating the Database they want to use.
   4.6. If using notepad, the .sql file's encoding must be saved as ANSI.

5. Verify that patch.txt and the files modified lists given by the developers match correctly.
   5.1. Take the list from patch.txt and mark off each file as you find it in the developer's comment on his/her tickets.
   5.2. If there are any files missing from patch.txt that are on the files modified list, contact developer (step a).
   5.3. If there are extra files on patch.txt that are not documented in the developer's comments, contact developer (step a).

6. Deploy to TEST
   6.1. There is an automated deployment system for VAT to deploy to TEST

6.2. Send an E-Mail to deploy@liberty.edu with the following in the subject line: TEST TYPYYYYMMDD [type of deployment]. Where TYPYYYYYMMDD is the patch folder, such as PAT20050807, and [type of deployment] specifies what you want deployed from the patch folder, below are the options for [type of deployment]

    6.2.1. FILES_ONLY – Deploys only files and no SQL.

    6.2.2. SQL_ONLY – Only runs SQL, does not deploy files.

    6.2.3. FILES_FIRST – Deploys the files first, then runs the SQL.

    6.2.4. SQL_FIRST – Runs the SQL first, then deploys the files.

6.3. If any problems persist, go to step "d."

7. Verify a successful deployment to TEST - A confirmation e-mail will be sent back to you telling you the results of the deployment. Check the SQL executions to verify that they were ran as expected. Also, make sure the e-mail reports a successful deployment of the files.

7.1. If a problem occurs and it appears to be a deployment issue or systems admin issue, go to step d, if it looks like a developer's issue, go to step a.

7.2. If a problem occurs and it appears to be a developer's coding issue, go to step d.

8. Test on TEST

8.1. Testing using developer's test plan – Follow the steps that the developer provided in the test plan, but try to expand as much as possible. If they tell you to try a specific value, try a lot of different values that the developer may have not have thought of.

8.2. Regression Testing – Any time a developer has a patch that affects an application or the website, test that application in full to ensure that the developer's patch did not break anything on that application.

8.3. If something breaks, determine if it is the developer's code (see step a), or system admin's configurations (see step d)

9. Update Ticket as passing on TEST on HD (HelpDesk) and VATM (Verification and Testing Manager).

   9.1. HelpDesk – Comment the following message, "Test Completed on TEST, ready for PRODUCTION.

   9.2. VATM – Mark the Ticket as Pass on Test, and select your name in the drop down box.

10. Deploy to PROD (by System Admin) – E-mail a request to System Admin in the following format requesting a deployment to PRODUCTION.

   10.1.   Subject Line: Deploy PAT20050627to PRODUCTION

   10.2.   Body:

Change Management Plan

Date Prepared: [Today's Date]

Prepared By: [Your Name Here]

Revision: [??? Not yet indicated by System Admin what this is for]

Change Priority Level: [Refer to Change Management Procedures.doc in \1414_a$\VerificationAndTesting: Level 1 is Emergency, Level 2 is Urgent, Level 3 is Normal]

Purpose of Change: [What is being fixed?]

Description of Problem that Necessitates the Change: [Describe what the problem was that is being fixed]

Proposed Resolution: [Describe how it is being fixed]

Deployment Schedule: [When should this be deployed?]

Post Deployment Monitoring Schedule: [Who will monitor the changes, for how long?]

Potential Issues and Issue Mitigation: [List any issues you one might expect]

Roll-Back Plan: [List the files that will need to be rolled back]

Systems Affected: [List the applications, areas of the website that are affected]

Services Affected: [List services within the systems that are affected]

Notifications: [Who to notify in case of problems]

Research Documentation Location: [Any documents related to deployment]

Software Location: [Indicate where the files are coming from: EX: 1416_a$\PAT20051010]

Step-by-Step Change\Deployment Process: [Give a detailed step-by-step, including readme.txt's, SQL, and file deployments.]

11. Verify a successful deployment to PRODUCTION – System Admin will send an e-mail to the VAT notifying them that the deployment is complete. At that time, the VAT will confirm that it has been deployed to PRODUCTION by performing a quick test to verify that the new files are actually on the PRODUCTION server. Also, the VAT should also check the HelpDesk Department Tickets box to verify that no immediate problems are pouring in through the ticket system. If there has not been a successful production, then determine if it was developer's fault (go to step a) or system admin's fault (most probable). If system admin's fault, contact system admin and find a solution.

12. Test on PRODUCTION

 12.1. Testing using developer's test plan – Follow the steps that the developer provided in the test plan, but try to expand as much as possible. If they tell you to try a specific value, try a lot of different values that the developer may have not have thought of.

 12.2. Regression Testing – Any time a developer has a patch that affects an application or the website, test that application in full to ensure that the developer's patch did not break anything on that application.

 12.3. If something does not pass on PRODUCTION, you must immediately contact system admin for a rollback (see step e)

13. Update Ticket as Passing on PRODUCTION

 13.1. HelpDesk – Comment the following message, "Test Completed on PRODUCTION, Ticket Complete.

 13.2. VATM – Mark the Ticket as Pass on Production, and select your name in the Production Tester drop down box.

14. Verify No Errors Caused by Patch Occur in the Next 24 Hours – After the ticket has been commented as completed, leave the ticket in the Patch Deployment User's box as well as the developer's box for at least 24 hours.   If errors happen, go to step e to rollback.

15. If no complaints or tickets are generated directly related to the patch, then the VAT should mark the ticket as completed, thereby closing the ticket. After all the tickets for a particular patch folder are completed, the patch folder should be moved to the applied folder in 1416_a$.

Explanations of What to Do when Something Fails

a   Developer – If at any point, the developer cannot fix the below issues, or they are not fixing the issue in reasonable time, follow the flow to step b to Terminate his/her patch.

1) Step 1.1 Failure (+1 Defect for Developer) – E-mail and talk in person to the developer informing them that his/her ticket does not have all the required information in the ticket.

2) Step 4 Failure (+1 Defect for Developer) - E-mail and talk to the developer and tell them to fix his/her SQL script. Make sure they know all the policies for writing a sql script.

3) Step 5 Failure (+1 Defect) – E-mail and talk to the developer, make sure they have the correct files in the patch and the correct files listed on the ticket.

4) Step 7 or 11 Failure (+1 Defect) – Contact the developer and make sure they research and find the solution to the issue.

5) Step 8 Failure (+1 Defect) – E-mail the error you received, along with the steps taken to get the error. Also, talk to the developer and make sure he/she is aware of the problem.

b   Remove Developer's Patch from patch folder – Simply tell the developer to remove all his/her files related to that patch and take his/her ticket number out of ticket.txt. Remove Patch Development User from the Ticket on HelpDesk, and disassociate the ticket from the patch date on VATM.

c   Correct the Folder Set Up

1) Step 2.1 Failure (+1 Defect) – Name the folder correctly, find out who created the folder, give them a defect, and make sure they know how to create the folder correctly.

2) Step 2.2 Failure (+1 Defect) – Create ticket.txt if it does not exist and find make sure the developer's know that ticket.txt must exist in the root of the patch folder. If there are spaces in ticket.txt, find out who made the mistake and make sure they know how to set it up correctly.

3) Step 3.3 Failure (No Defect) – If any files exist in the patch folder that do not belong, talk to the person that put them there and get the issue resolved.

4) Step 3.4 Failure (+1 Defect) – Change the folder to all lowercase, find the developer responsible and make sure they know what they did.

d   SYSTEM ADMIN ASSISTANCE

1) Step 6 Failure (No Defect) – Contact System Admin to either deploy the patch to TEST manually, or to fix the auto-deploy system

2) Step 7 or 8 Failures (No Defect) – Contact System Admin, report the problem, wait for a solution.

e   Rollback – System Admin is contacted to rollback the patch that breaks on production. This might be the entire patch if the specific problem is unknown, or it may be a single file or several files. After system admin rolls back, you must retest and confirm that the rollback was successful. If not, then repeat step e, if successful find the solution and contact developer responsible.