APPLICATION SOFTWARE DEVELOPMENT RESPONSIBILITIES

PROGRAM DEVELOPMENT

1. ANALYZE PROGRAM REQUIREMENTS PER INPUT, PROCESSING AND OUTPUT. UNDERSTAND ALL OPERATOR CONTROL CAPABILITY AND DISPLAY OUTPUT TO SATISFY THE SYSTEM REQUIREMENTS.

2. DEVELOP SOFTWARE IMPLEMENTATION TECHNIQUES AND HIERARCHICAL STRUCTURE TO SATISFY SUBSYSTEM REQUIREMENTS AND NASA SOFTWARE STANDARDS.

3. REVIEW PROGRAM REQUIREMENTS FOR APPLICABLE SUBSYSTEM FD'S TO UNDERSTAND FD DATA BANK DEFINITIONS (EXAMPLE: SUBTYPE, APPROPRIATE DATA LINKS, DATA FORMATS).

4. DECOMPOSE REQUIREMENTS INTO LOWER LEVEL PROCESSING IN THE FORM OF FLOWCHARTS, DETAILED SPECIFICATION, PDL OR ANY OTHER AS REQUIRED PER PROGRAM NASA STANDARDS. NOTE: GENERIC SPECIFICATION WRITEUP IS ALWAYS REQUIRED.

5. DEVELOP GENERIC OPERATOR MESSAGES AND PROGRAM CONTROL TO SATISFY ITEM 1.

6. DEVELOP ENGINEERING SUPPORT REQUESTS (ESR'S) TO APPROVE IMPLEMENTATION OF THE SOFTWARE MODULES ONLY IF AN E.S.R. WAS NOT SUBMITTED DURING ORIGINAL SUBSYSTEMS REQUIREMENTS DEFINITION PHASE.

7. DEVELOP SOFTWARE IMPLEMENTATION PLAN (SIP) THAT IDENTIFIES THE SPECIFIC DEFINITION OF A PROGRAM MODULES IMPACTS.

   IDENTIFY THE FOLLOWING ITEMS ON THE SIP:

   1. SOFTWARE PROGRAM NUMBERS (RUMS)
   2. TIME REQUIRED TO:
      - A. GENERATE CODE
      - B. DEBUG SOFTWARE
      - C. VERIFY SOFTWARE
   3. STS CONSTRAINT DATE
   4. RESPONSIBLE ENGINEER

8. DEVELOP DOCUMENTATION IMPLEMENTATION PLAN (DIP) THAT IDENTIFIES SPECIFIC DEFINITION OF DOCUMENTATION IMPACTS.

   IDENTIFY THE FOLLOWING ITEMS ON THE DIP:

   1. DOCUMENTATION IMPACTS
      - A. REQUIREMENTS WHEN REQUIRED
      - B. SPECIFICATIONS
      - C. USERS GUIDE
      - D. SVP
      - E. CONSAC
   2. RESPONSIBLE ENGINEER
   3. STS CONSTRAINT DATE

9. SUBMIT ESR, SIPS AND DIPS FOR ANY REQUIRED SOFTWARE CHANGES TO THE CONFIGURATION GROUP FOR TRP APPROVAL.

10. SUBMIT DATA BANK UPDATE FORMS TO ADD APPROVED SOFTWARE PSEUDO FD'S THAT ARE REQUIRED TO SATISFY PROGRAM IMPLEMENTATION REQUIREMENTS.

11. DEVELOP GOAL CODE PROGRAM INTERNAL STRUCTURE ELEMENTS.

    1. INTERNAL NAMES
    2. OPERATOR INTERRUPTS
    3. DATA VALIDITY
    4. GOAL MEASUREMENT INTERRUPTS
    5. LED'S
    6. PFP KEYS & PF KEYS
    7. CURSOR INTERRUPTS
    8. DISPLAY SKELETON AS NECESSARY
    9. CLASS ERROR PROCESSING
    10. GOAL TIMING REQUIREMENTS
    11. PROGRAM TERMINATION OPERATOR
    12. HOUSEKEEPING REQUIREMENTS
    13. COMMUNICATION BETWEEN SOFTWARE MODULES

12. CONDUCT GOAL CODE WALKTHROUGHS TO DISCUSS IMPLEMENTATION OF REQUIREMENTS PER SOFTWARE DESIGN WITH COWORKERS/LEAD ENGINEER WHEN NECESSARY.

13. ENTER THE SOURCE CODE AND CORRECT ANY LOGIC ERRORS.

14. COMPILE THE SOURCE CODE AND CORRECT ALL COMPILE ERRORS AND PROGRAM WARNING STATEMENTS.

15. CONFIGURE GOAL SOURCE FILE TO APPLICABLE CONSOLE AND CORRECT ANY TCID CONGIGURATION ERRORS.

16. DEBUG ALL PROGRAM LOGIC PATHS PER DESIGN REQUIREMENTS FOR SUCCESSFUL AND UNSUCCESSFUL LOGIC PATHS.

17. DEVELOP ANY MATH MODEL CONTROL PROCEDURES TO AID IN THE DEBUG AND SVP PROGRAM VERIFICATION.

18. DEVELOP PROGRAM SOFTWARE VERIFICATION PROCEDURES TO VERIFY THE GOAL CODE SATISFIES THE ORIGINAL SYSTEM REQUIREMENTS IDENTIFIED IN ITEM 1. ONLY DO SVP'S WHEN REQUIRED.

19. DEVELOP DETAILED USER GUIDE DOCUMENT TO REFLECT OPERATOR CONTROL OF SOFTWARE MODULES. VERIFY THAT THE USER GUIDE DOCUMENT ADEQUATELY REFLECTS/SATISFIES PROGRAM REQUIREMENTS.

20. GENERATE PROBLEM REPORT IF ANY ERROR IS DETECTED TO THE OPERATION OF THE SOFTWARE IN VERIFICATION.

21. CONDUCT CODE WALKTHROUGH PRIOR TO PROGRAM VERIFICATION WITH COWORKERS AND/OR LEAD ENGINEER.

22. MAINTAIN ALL PROGRAM UPDATES PER CHANGES IN SUBSYSTEM REQUIREMENTS.

23. DEVELOP/FINALIZE PROGRAM SPECIFICATION DOCUMENT UPDATE BEFORE COMPLETION OF THE SOFTWARE VERIFICATION, IF REQUIRED.

SYSTEM RESPONSIBILITY

1. UNDERSTAND SOFTWARE SET, ONE IS ASSIGNED TO, CONCEPTUALLY AND OPERATIONALLY.

2. ATTEND DOCUMENTATION DESIGN REVIEWS WHEN APPLICABLE.

3. ATTEND ASWT MEETINGS.

4. IMPLEMENT STANDARDS RELATED TO SOFTWARE APPLICATIONS.

5. TURN INTO ALL DOCUMENTATION DEPOSITORIES ANY APPLICABLE SOFTWARE DOCUMENTATION AND RECORDS.

6. INSURE ALL SOFTWARE, FILES, MODEL CONTROL PROCEDURES, DEBUG AIDS, ETC., ARE CONFIGURED, LOADED, TO THE PROPER CCMS SUPPORT BUILD FOR A PARTICULAR FLOW.

7. GET REPRODUCTION MADE FOR DOCUMENTATION AND DISTRIBUTE AS REQUIRED.

8. KEEP COWORKERS INFORMED OF PROGRESS RELATED TO ASSOCIATED TASKS.

9. KEEP VLS COUNTERPARTS INFORMED AS REQUIRED.

10. MAINTAIN ACCURATE FILE/HARDCOPY OF APPLICABLE PROGRAMS AND RELATED DATA.

11. VERIFY INITIAL SPECIFICATION OR USER GUIDE DRAFT INPUT/REDLINES ARE INCORPORATED WITHIN THE FINAL DOCUMENTATION VERSION.

12. TRACK ALL APPROPRIATE SUBSYSTEM MIEWF ITEMS PER PROCESSING FLOW AND VERIFY COMPLETION OF THOSE ITEMS.

FIRING ROOM ACTIVITIES

1. UNDERSTAND AND IMPLEMENT PROTOCOLS RELATED TO FR ACTIVITIES.

2. SCHEDULE FIRING ROOM TIME FOR TESTING AND VERIFICATION PURPOSES.

3. REPORT FR SUPPORT IMPACTS.

4. MAINTAIN FR SCHEDULE AND ACCOMPLISHMENTS LOG FOR ON-STATION ACTIVITIES.

5. TROUBLESHOOT PROBLEMS DISCOVERED DURING HARDWARE OPERATIONS (PR'S, IPR'S).

6. KNOW AND UNDERSTAND HOW TO OPERATE SPECIAL UTILITIES AND RETRIEVALS TO SUPPORT PROGRAM DEBUG AND VERIFICATION.

7. REVIEW SPA PRINTOUTS WHEN REQUIRED TO SUPPORT PROGRAM DEBUG AND VERIFICATION OPERATIONS.

8. REQUEST FEP'S REQUIRED TO SUPPORT PROGRAM DEBUG AND VERIFICATION OPERATIONS.

CHANGE PAPER

1. KNOW THE CURRENT PAPER SYSTEM.

2. DEVELOP ESR'S, SIP'S AND DIP'S FOR APPROVAL OF SOFTWARE IMPLEMENTATION CHANGES AND ANY OTHER PAPERWORK REQUIRED.

3. ATTEND AND BE PREPARED TO DISCUSS APPLICABLE CHANGE PAPER AT CHANGE SCREENING BOARD, SHUTTLE DATA SYSTEM CONFIGURATION CONTROL BOARD, (SDSCCB), AND TECHNICAL REVIEW PAPEL (TRP).

4. ASCERTAIN ALL REQUIRED SIGNATURES RELATING TO SOFTWARE PAPER PROCESSING.

5. UNDERSTAND ALL CHANGE PAPER THAT YOU ARE RESPONSIBLE FOR AND VERIFY TECHNICAL ACCURACY BEFORE GRANTING SIGNATURES.

6. SUBMIT DATA BANK UPDATE FORMS WITH ESR'S, SIP'S AND DIP'S WHEN DATA BANK UPDATES ARE TO BE SUPPORTED FOR YOUR SYSTEM.

7. GENERATE INTERIM PROBLEM REPORTS (IPR'S), PROBLEM REPORTS (PR'S) TO CORRECT ANY SOFTWARE PROCEDURAL ERRORS OR OPERATIONAL DEFICIENCIES.

GENERAL DUTIES

1. MAINTAIN FILE OF ENGINEERING DESK INSTRUCTIONS.

2. ATTEND GROUP MEETINGS.

3. TURN IN WEEKLY ACTIVITY REPORTS.

4. KEEP TIME CARDS ACCURATE ON A DAILY BASIS.

5. MAINTAIN CLEAN AND SANITARY WORK AREA.

6. ATTEND ALL SCHEDULED TRAINING CLASSES.

7. UNDERSTAND LPS, CCMS, CDS, LSDN AND ANY OTHER RELATED COMPUTER OPERATIONS SUPPORT STRUCTURE.

8. MAINTAIN PROFESSIONAL RELATIONSHIP WITH ALL COUNTERPARTS (NASA HARDWARE/SOFTWARE, CONTRACTOR HARDWARE/SOFTWARE).

9. NOTIFY SUPERVISOR/MANAGER OF ABSENCE.

10. UNDERSTAND ALL APPLICABLE KSC FACILITY AND ORBITER SUBSYSTEM ACRONYMS.

LEAD FUNCTIONS

IN ADDITION TO RESPONSIBILITIES ALREADY LISTED, LEADS ARE ALSO RESPONSIBLE FOR THE FOLLOWING FUNCTIONS:

1. UNDERSTAND OVERALL FUNCTION OF ASSIGNED SUBSYSTEM SOFTWARE APPLICATION SETS.

2. REVIEW SUBSYSTEM SOFTWARE CHANGE PACKAGES FOR POSSIBLE IMPACTS TO ANY APPLICATION SOFTWARE SET AS REQUIRED.

3. ESTABLISH RESPONSIBLE GROUP PRIORITIES OF WORK TO BE DONE AND PRIORITIZE WORK TO BE ACCOMPLISHED WITHIN SCHEDULED TIME PERIOD.

4. PROJECT WORK TO BE DONE VIA SCHEDULES IN SIX (6) MONTH PERIODS AND WEEKLY AS REQUIRED.

5. DELEGATE WORK TO COWORKERS.

6. INSURE THAT EMPLOYEES ACCOMPLISH WORK ASIGNMENTS PER SCHEDULE AND PER DIRECTION ACCORDING TO SCHEDULE.

7. PROVIDE TECHNICAL ASSISTANCE WHEN REQUIRED TO ASSOCIATED PROGRAMMERS AND VERIFY THAT THEY UNDERSTAND THE ASSOCIATED TOPIC.

8. DISSEMINATE TECHNICAL AND NON-TECHNICAL INFORMATION TO EMPLOYEES ASSIGNED TO THEIR APPLICATION SETS.

9. PROVIDE THE COORDINATION BETWEEN THE SUBSYSTEM ENGINEERING GROUPS AND THE SUBSYSTEM APPLICATION PROGRAMMERS IN RESPECT TO THE PROPER DEFINITION OF SUBSYSTEM REQUIREMENTS.

10. ESTABLISH AN EFFECTIVE CHANGE PACKAGE FILE SYSTEM WITH A FILE LOG BOOK THAT IDENTIFIES THE WAD AND DESCRIPTION OF CHANGE FOR HISTORICAL REFERENCES.

11. PERFORM SUPERVISOR FUNCTIONS UPON REQUEST IN THEIR ABSENCE.

12. ATTEND MANAGER/SUPERVISOR/LEAD FORUMS.