SOFTWARE DEVELOPMENT PLAN FOR *x*Flo

September 2003

1.0 SCOPE

The scope of the software development effort is described in detail in the software requirements description for *x*Flo. The xFlo software will model non-isothermal multiphase flow and multicomponent reactive transport in porous/fractured rock.

2.0 BASELINE ITEMS

The xFlo software will be released in stages. Version 1.0 will include multiphase thermal-hydrological flow modeling capabilities. Reactive transport capabilities will be included in Version 2.0. The specific products to be delivered from this software development project include: (1) source code for *x*Flo V1.0, (2) source code for *x*Flo V1.0, (4) *make* files for each release; and (5) revised User Manual for each release.

3.0 PROJECT MANAGEMENT

3.1 Work Breakdown Structure

3.1.1 Version 1.0

Task 1. Code design (Painter, 4 weeks).

Task 2. Implement generic code framework (Painter, 2 weeks).

Task 3. Implement modules for linear solvers (Painter, 1 week).

Task 4. Implement thermal-hydrology module (Painter, 1 week).

Task 5. Implement equations-of-state and thermophysical properties module (Scherer, 3 weeks).

Task 6. Preliminary testing and debugging (Painter, 1 week).

Task 7. Implement keyword-driven input format (Scherer and Sun, 4 weeks).

Task 8 Final testing and debugging (Sun and Painter, 4 weeks).

Task 9 Write user's manual for Version 1.0 (Painter and Sun, 2 weeks).

Task 10. Code validation test plan (Painter and Sun, 1 week).

Task 11. Code validation report (Painter and Sun, 2 weeks).

3.1.2 Version 2.0

Task 1. Code design (Painter, 3 weeks).

Task 2. Implement non-reactive transport capabilities (Sun, 2 weeks).

Task 3. Implement modules for initialization of chemical system (Sun and Painter, 2 weeks).

Task 4. Implement reactive transport capabilities (Sun and Painter, 6 weeks).

Task 5. Preliminary testing and debugging (Sun, 2 weeks).

Task 6. Implement keyword-driven input format (Scherer and Sun, 4 weeks).

Task 7 Final testing and debugging (Scherer and Sun, 4 weeks).

Task 8 Write user's manual for Version 1.0 (Sun and Painter, 3 weeks).

Task 9. Code validation test plan (Sun, 1 week).

Task 10. Code validation report (Sun, 2 weeks).

3.2 Schedules

The following schedule is preliminary and applies only to Version 1.0. As of this writing, there are no formal delivery dates imposed by the client. Final schedule will depend on level of funding and priorities of client. Tasks 1-6 will be completed by December 31, 2003. Tasks 7-9 will be completed by September 30, 2004, and tasks 10-11 will be completed by December 31, 2004.

3.3 Staff

Work will be directed by S. Painter, and will be performed by S. Painter, A. Sun and C. Scherer.

4.0 DEVELOPMENT PROCEDURES

4.1 Hardware and Software Resources

All code development will be done on SUN workstations or servers running the SOLARIS operating system, or on Microsoft Windows compatible PCs. The SUN FORTRAN 95 compiler (Version 6.0, or later version) will be used on the UNIX platform. Lahey Fortran 95 Version 5.0 (or later version) will be used on the PC platform.

4.2 Coding

Coding for MULTIFLO will be done in FORTRAN 95. Coding style will be consistent with the style and conventions of Akin, 2003.

4.3 Acceptance Testing and Analysis

The results of testing will be recorded in scientific notebook. Acceptance tests will be similar to those used in MULTIFLO development. Results of the standard acceptance tests and variants will be included with the source code for all releases.

5.0 CONFIGURATION MANAGEMENT PLAN

The working version of the code will be maintained by S Painter on the SUN server named SPOCK. New or modified modules will be tested and reviewed by S. Painter before the changes are incorporated with the working code. A description of the changes and locations of the working directories will be recorded in scientific notebook E. Baselined versions are kept in the QA records vault.

5.1 Tools

Standard development tools included with the Solaris compiler suite version 6.0 will be used. Unix utilities *diff*, *filemerge* and *make* will be used to perform the code manipulations required to maintain the official version of the working code.

5.2 Configuration Identification

The configuration identification scheme will be as follows: the first phase release will be Version 1.0. Subsequent minor bug fixes will be released as Version 1.0.1, 1.0.2, etc. Similarly, the second phase release will be initially be denoted Version 2.0, and will supersede the Version 1.0 series. Subsequent revisions will be numbered Version 2.*i*.*j*, where *i* is incremented for a major revision and *j* is incremented for a minor bug fix.

5.3 Configuration Procedures

The standard SCR change request form will be used for all significant changes to the controlled source code.

6.0 REFERENCES

E. Akin, “Object-oriented programming via Fortran 90/95”, Cambridge University Press, 2003.

APPROVED:

E. C.   6/7/2007

Signature of Element Manager   Date

The original of this document was prepared in Sept. 2003. That original has been misplaced and this duplicate prepared in its place. The approval is as required at the time the original document was prepared.