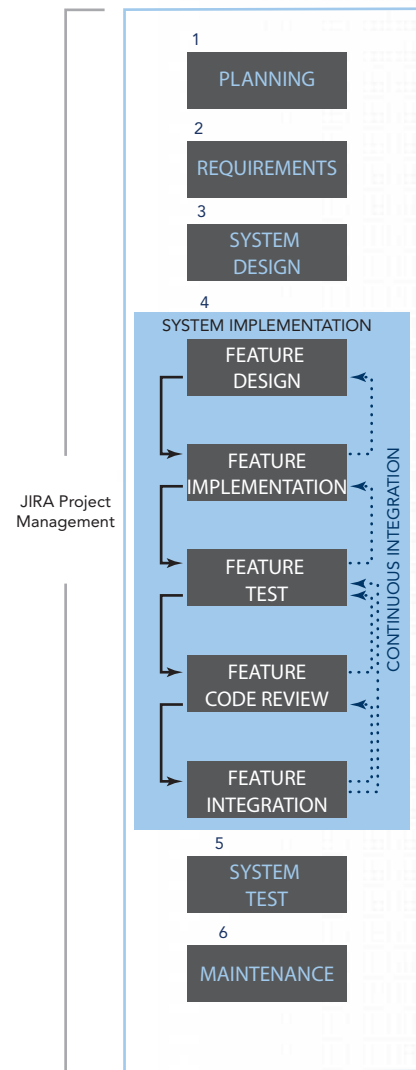


MODULAR SOFTWARE

The Space Dynamics Laboratory's (SDL) Modular Software team specializes in ISO-certified software development and engineering services including entire spaceflight systems, responsive space functionality, individual module development, and consulting. These services also encompass process improvement, requirements generation, and technical coordination, advisement, and development, as well as independent verification and validation, ground support equipment development, and data post-processing.

As a nonprofit University Affiliated Research Center (UARC), SDL has established itself as a trusted Government agent for developing essential software engineering technologies. Due to this partnership, SDL provides unlimited rights to Government customers with no licensing fees for any code that the Modular Software team develops.



Traditional Development Waterfall

The Modular Software team has increased process responsiveness by incorporating software engineering best practices from the general IT industry (shown in the blue box) into the traditional aerospace software development process. As a result, the code is very stable, with minimal regression errors.

PRIMARY BENEFITS

TRANSPARENT DEVELOPMENT PROCESS

- SDL gives customers complete access to the online tool used for software project management
- Customers can see all development activity, status, code commits & the repository in real time

THOROUGH CODE REVIEW

- Efficient, remote & continual peer review; all code is reviewed before being committed to the repository
- Promotes shared knowledge in the project, increasing team flexibility

CONTINUOUS INTEGRATION

SDL's continuous integration (CI) process ensures code quality and stability and dramatically reduces regression errors.

The CI server is responsible for:

- Checking for clean, warning-free code compilation
- Executing unit & system integration suite tests
- Performing memory & static analysis
- Generating software documentation
- Checking code style

MODULAR SOFTWARE

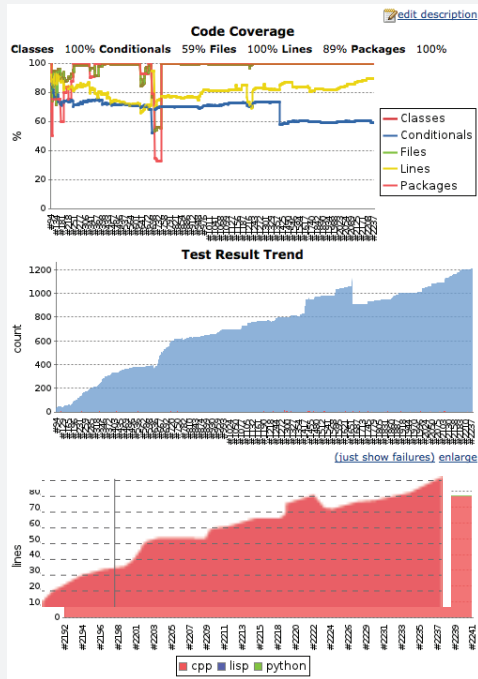
SDL'S MODULAR SOFTWARE TEAM EXPERIENCE

TYPES OF SOFTWARE

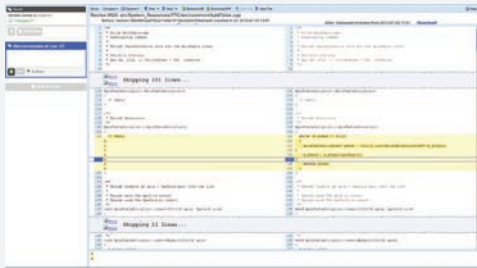
- Spaceflight payload & satellite Command & Data Handling (C&DH)
- Spaceflight Attitude Determination & Control Systems (ADCS)
- Ground Command & Control (C2)
- Ground resource management tools
- Consultative Committee for Space Data Systems (CCSDS)
- AX.25 Link-Layer protocol
- Space Plug-&-Play (SPA) core infrastructure & applications
- Electronic Ground Support Equipment (EGSE)
- Assembly, Integration & Test (AI&T)
- Verification & Validation (V&V)
- Hardware interface, instrumentation & development
- Device drivers
- Database development & management
- Post-acquisition data processing & analysis
- Image processing
- Cloud architecture & networks

TOOLS & TECHNOLOGIES

- VxWorks, Linux, Windows
- JIRA, BitBucket, Confluence
- UML, Enterprise Architect
- SpaceWire
- CCSDS
- XML, XSD
- Code Collab, Jenkins, git, Valgrind, Subversion, xUnit testing including C++
- C, C++, C#, Java
- Eclipse plugin development
- Embedded Software, 8051, AVR, SPARC, LEON-3, PPC
- Amazon Web Services (AWS)



Use of CI technology provides continuous monitoring of development metrics and codebase



Efficient and thorough code review detects and eliminates defects early in the lifecycle

```
[-----] 1 test from SpaNamedSemaphore
[ RUN ] SpaNamedSemaphore.waitAndPost
/home/jacob/ssu/test/common/portability/testSpaNamedSemaphore.cpp:49: Failure
Value of: result
Actual: -1
Expected: 0
[ FAILED ] SpaNamedSemaphore.waitAndPost (2 ms)
1 test from SpaNamedSemaphore (2 ms total)

[-----] 6 tests from SpaTimeVal
[ RUN ] SpaTimeVal.constructStructTimeVal
[ OK ] SpaTimeVal.constructStructTimeVal (0 ms)
[ RUN ] SpaTimeVal.constructSpaTimeSpec
[ OK ] SpaTimeVal.constructSpaTimeSpec (0 ms)
[ RUN ] SpaTimeVal.copyCtor
[ OK ] SpaTimeVal.copyCtor (0 ms)
[ RUN ] SpaTimeVal.assignmentOperator
[ OK ] SpaTimeVal.assignmentOperator (0 ms)
[ RUN ] SpaTimeVal.equalityOperator
[ OK ] SpaTimeVal.equalityOperator (0 ms)
[ RUN ] SpaTimeVal.streamInsertion
[ OK ] SpaTimeVal.streamInsertion (0 ms)
6 tests from SpaTimeVal (0 ms total)
```

Comprehensive unit and integration test suites help eliminate regression errors and ensure code stability