The STAR Algorithm Integration Team (AIT) Research to Operations Process

Presented by

Tom King

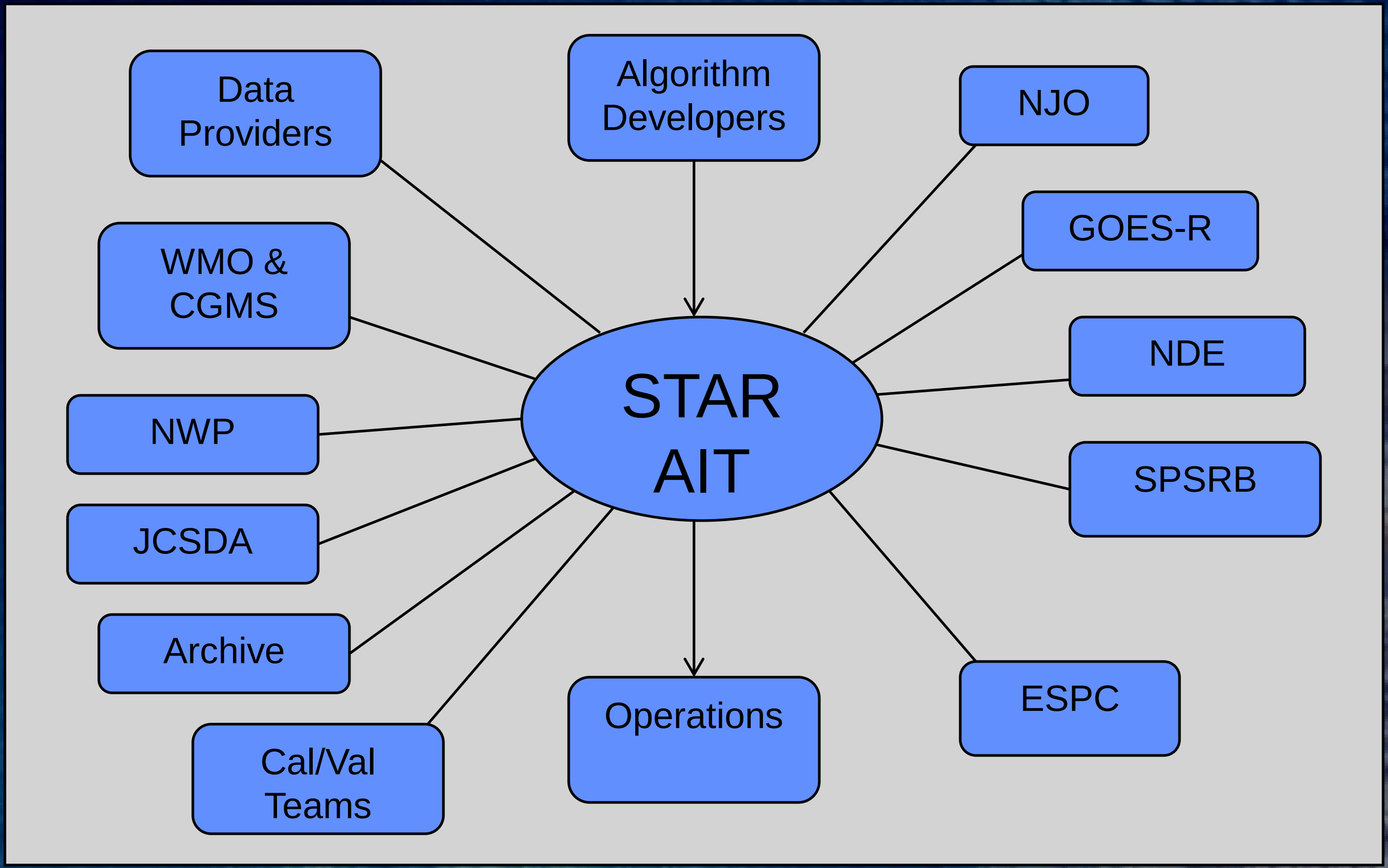
The Problem

- Scientists write code that needs to go to operations, but:
  - Many scientists often prefer interpreted languages like IDL and Matlab or even older languages like Fortran 77. Fortran 90/95 or C/C++ expertise does not always exist.
  - Code written in isolation without considering how it would run within a larger system
  - Code works with only certain compilers
  - Code uses non-standard functions
  - Code doesn't account for operational concerns such as run time, memory usage, disk I/O, error checking
  - Haven't considered what input and ancillary data are actually available in the operational environment and what the latency of those data have
  - Code is often not well documented
  - Code is often written by a mixture of programmers with varying styles and abilities
  - Paths are hardcoded and algorithms assume the data they need will be in arranged in a particular data tree
  - Executable code makes system calls (assuming a certain OS)

- Operations is tasked only to receive, run, and monitor the code

The Problem

- R2O isn't as simple as cleaning up science code and delivering to operations. This work also involves coordinating with many stakeholders.



STAR R2O Solution

- The solution is to have the STAR Algorithm Integration Team (AIT) act as a "middle man" to:
  - » Assist the science teams in providing Quality Assurance (QA) for the entire R2O process and do so in a way that isn't a burden for them
  - » Work with stakeholders to refine requirements and enhance user readiness

- Product QA is concerned with assuring that the work products (software & documentation) created during the project's lifecycle meet their requirements.

- Process QA is concerned with assuring that the process standards (reviews & stakeholder interaction) are met throughout the project lifecycle.

The STAR AIT Team

- The STAR AIT team is lead by Walter Wolf and consists of 30+ contractors

- The STAR AIT R2O process has been successfully applied to a number of past and current projects:
  - IASI
  - NUCAPS
  - GCOM
  - BUFR/GRIB2 Toolkit
  - Blended Cloud Products
  - JPSS Risk Reduction
  - GOES-R AIT
  - JPSS AIT
  - OSPO Product Monitoring
  - VIIRS Polar Winds
  - GOES Winds
  - Advanced Composition Explorer

R2O Process Methodology

- The STAR AIT R2O process evolved from a CMMI level 3 process that was tailored and blended with the existing SPSRB process.

- The process consists of working with science teams and the stakeholders to do the following:
  - » Conduct a standard set of project reviews
  - » Generate a standard set of documentation
  - » Stakeholder interaction
    - – Requirements development/refinement
  - » Risk tracking and mitigation
  - » Code cleanup for:
    - – Coding/Security
    - – Configuration Management
    - – Software Testing & Product Validation
    - – Common data formats and metadata (CF & ISO)
    - – Standard languages, tools, and libraries
  - » Delivered Algorithm Package (DAP) delivery

Reviews

- The review process is described on the SPSRB website at (http://projects.osd.noaa.gov/SPSRB/design_review_guidance.htm)
  - Preliminary Design Review (PDR)
    - Present preliminary requirements
    - Identify the problem, provide background, and discuss competing solutions
    - Identify an initial design
    - Presents risks
  - Critical Design Review (CDR)
    - Finalize requirements
    - Verify that the chosen design is able to meet those requirements
    - Present algorithm theoretical basis
    - Software architecture & Concept of operations
    - Product QA (Validation plans)
    - Presents risks

Reviews

- Unit Test Readiness Review (UTTR)
  - Present test plan, procedures, and results
  - Test must demonstrate that software is meeting its functional requirements
  - Presents risks

- Software Review (SR)
  - Check that code meets all SPSRB coding and ESPC security standards

- Algorithm Readiness Review (ARR)
  - Demonstrate that all data products are meeting requirements
  - Identify Delivered Algorithm Package (DAP) components and demonstrate that they meet requirements
  - Presents risks

Documentation

- STAR project documentation:
  - » Requirements Allocation Document (RAD)
    - Identify basic and derived requirements
    - Tie these requirements to user requests
    - Allocate requirements to components of the system design
  - » Review Item Disposition (RID) – Risk Tracking
    - Track, rate, mitigate, and assign individuals to address risks for the lifecycle of the project
  - » Presentation slide packages
    - Preliminary Design Review
    - Critical Design Review
    - Unit Test Readiness Review
    - Algorithm Readiness Review

Documentation

- SPSRB required documentation (Templates are available here http://projects.osd.noaa.gov/SPSRB/standards_data_mtg.htm)
  - System Maintenance Manual (SMM)
    - Describes the system design, interfaces, files (input, intermediate, and output)
    - Identifies the hardware, system requirements
    - Identifies the installation and operational procedures (shutdown/restart) required to run the system
    - Describes monitoring (error message, quality monitoring), maintenance, and troubleshooting
  - External Users Manual (EUM)
    - Describes the detailed format of the output data files for end users
  - Algorithm Theoretical Basis Document (ATBD)
    - Provides the theoretical background and description of the algorithm
    - Performance estimates, practical considerations
    - Validation procedures
    - Assumptions and limitations

Code Updating

- Getting code to meet SPSRB coding standards
  - » Removing hardcoded paths
  - » Adding comments and standard headers
  - » Using meaningful variable names
  - » Standard indentation of blocks
  - » Avoiding non-standard functions
- Porting code to target operating systems, compilers, and platforms
- Adding error checking and logging
- Profiling and debugging
- Rewriting code into ESPC approved languages
- Testing to verify offline research and operational codes produces the same results
- Providing updates or tools for handling operational interfaces
- Integration into a test system (e.g. ADL or the GOES-R Framework)

Development Standards

- Coding of software:
  - » SPSRB Coding standards available on the SPSRB website (http://projects.osd.noaa.gov/SPSRB/standards_software_coding.htm)
  - » OSPO Technical Reference Model (TRM) is the source of IT standards and specifications
  - » Software review conducted with OSPO PAL and ESPC IT security

- Using the STAR CM Tool (IBM Rational ClearCase, Version 7.0) to track and baseline development

- Implementing use of standard data formats such as netCDF, HDF, BUFR, and GRIB
  - » netCDF and HDF are preferred formats for many users and the archives
  - » BUFR and GRIB are standard NWP formats
  - » Metadata follows Climate and Forecast (CF) and ISO 19115 standards

Development Standards

- Use of standard tools
  - Common use of home-grown functions for time calculations, error checking, wrappers to netCDF and HDF API functions
  - Code generators for I/O handling (read, write, allocate, and deallocation) for Fortran 90 and C/C++
  - Use of Valgrind for profiling (resource usage and memory leaks)
  - Common set of home-grown coding checking/cleaning

- Implementation of standard test procedures
  - Code unit and system testing
  - Presentation of the results to stakeholders at the UTRR and ARR
  - Development of test plans
    - Identify test environment
    - Identify test data sets (input, intermediate, output)
    - Identify test code
    - Show test steps
    - Show test results and compare to requirements

Stakeholder Interaction

- Requirements development/refinement
  - » Working with end users to identify and agree upon on data formats and content
  - » Defining archive and metadata requirements
  - » Identifying or defining interfaces between algorithms and system into which they run
  - » Identifying documentation needs
  - » Identifying production rules for downstream integrators
  - » Identifying file name conventions
- Coordinating additional paperwork and documentation
  - » Data Access Request (DAR) forms
  - » Coordination with DMWG
  - » CLASS Submission Agreements (SA)
- Providing sample data products and software to end users prior to operational implementation for
  - » Product validation
  - » End-user readiness
- Reprocessing data for science teams to assist Cal/Val activities
- Attending Integration Product Team (IPT) meetings
  - » Keep track of upstream changes to algorithms and input formats
  - » Coordinate development with updates to the system in which the science algorithms will run
  - » Coordinate common standards for output and algorithm interfaces

Risk Tracking

- Identify risks and impacts
- Developing and managing schedules
- Assigning risks a rating as a function of likelihood and impact
- Developing mitigation plans
- Assigning actions to individuals for mitigation efforts
- Opening and closing risks as needed
- Risks and actions are presented and discussed at each review step

Delivered Algorithm Package (DAP)

- DAP contents
  - » Test plans and test data
  - » SPSRB documentation (ATBD, SMM, EUM)
  - » Source Code
  - » All scripts, static data files, and configuration files
  - » Production rules
  - » Description of interfaces
  - » Delivery memo and README

R2O Example: NUCAPS

- NOAA Unique CrIS/ATMS Product System (NUCAPS)
  - Project Lead: Walter Wolf
  - STAR algorithm science lead: Mark Liu (previously was Chris Barnet)
  - OSPO PAL: Awdhesh Sharma

- It is an SPSRB-funded project whose goal is to produce a software package that runs in NDE to provide
  - Produce CrIS thinned radiances in BUFR for NWP
  - Produce retrieved profiles of temperature, water and trace gasses
  - Produce validation products for STAR Cal/Val and OSPO monitoring
  - VIIRS cloud products collocated to CrIS
  - CrIS OLR

- NUCAPS was designed to be delivered in several phases so the process was tailored to the project schedule, scale and funding

- Leveraged the algorithms of AIRS and IASI

- Users consist of NWP, archive users, science teams

R2O Example: NUCAPS Stakeholder Interaction

- Stake holder interaction and requirement derivation efforts
  - » Acquire documents defining the project requirements (JPSS L1RD Supplement, SPSRB Project Plan, OSPO TRM, SPSRB coding standards and document templates, NDE DAP delivery standards). From this develop the RAD.
  - » Define and negotiate interfaces to the NDE system
  - » Identify required data formats, naming conventions, DAP delivery standards, documentation, system requirements (target platform, OS, compilers)
  - » Articulate algorithm needs to NDE (input and ancillary data, production rules, resource requirements)
  - » Worked with JCSDA, EMC, EUMETSAT, and WMO to define contents of and approval for the CrIS BUFR table descriptors
  - » Worked with NGDC and NCDC/CLASS to define metadata and archive requirements and methods
  - » Worked with OSPO PAL, NUCAPS science lead, and the Product Quality Monitoring team lead to define a monitoring methodology
  - » Worked with the STAR NDE, OSPO, and DMWG to acquire output data to support Cal/Val efforts at STAR

R2O Example: NUCAPS Project Reviews and Documentation

- Assembled and led the following reviews
  - » Preliminary Design Review
  - » Critical Design Review for Day 1 and 2 Products
  - » Unit Test Readiness Review for Day 1 and 2 Products
  - » Algorithm Readiness Review for Day 1 Products
  - » Algorithm Readiness Review for Day 2 Products
  - » Software Review
  - » Critical Design Review for Day 3 Products

- Developed and delivered project documentation for each phase
  - » SMM
  - » EUM
  - » ATBD
  - » RAD
  - » RID
  - » Review Slide Packages

R2O Example: NUCAPS Software Development

- Software development and update efforts
  - » Acquired CrIS and ATMS sample data provided by IPO
  - » Developed code following SPSRB coding standards
  - » Developed a near realtime simulation data generating system outputting IDPS-like HDF5 CrIS and ATMS (using GFS as input and a forward model)
  - » Developed a near real time processing system to ingest the simulated data, mimic the NDE interfaces, ran the algorithm code, and distributed data to a STAR ftp server (all on a 24/7 basis)
  - » Developed readers for input data and writers for output
  - » Developed the pre and post-processing software for the NUCAPS retrieval algorithm
  - » Developed the software to spatially and spectrally thin the CrIS radiances
  - » Developed the netCDF4 to BUFR conversion software
  - » Developed the software to generate the validation products (daily gridded, binary, and matchup data sets)
  - » Developed software for product monitoring of SDRs and EDRs
  - » Cleaned up retrieval code and developed scripts to create and "operationalized" version of the code (remove diagnostic print statements)
  - » Ported retrieval code to the target platform (IBM AIX)
  - » Tracked updates in ClearCase revision control

R2O Example: NUCAPS Validation Efforts

- Validation efforts
  - » Delivered preliminary DAPs to prepare NDE for integration
  - » Delivered test data products to NCEP, EUMETSAT, AWIPS, JCSDA
  - » Reprocessed NUCAPS focus days for product validation
  - » Reprocessed of the retrievals at the locations of AEROSE dedicated radiosondes for product validation
  - » Providing data to NPROVS for product monitoring
  - » Delivered monitoring product software to OSPO
  - » Made CrIS BUFR and NUCAPS retrieval products available to end users
  - » Coordinated with NDE, ESPC, and the STAR DMWG to gain access to the optional product output files here at STAR in support of validation and monitoring efforts
  - » Validated the DAP contents

R2O Example: NUCAPS Additional Efforts

- Additional efforts
  - » Tracked and mitigated risks throughout the lifecycle of the project
  - » Delivered Day 1 and Day 2 NDE-compliant DAPs to NDE
  - » Assisted with NDE integration, troubleshooting, and validation after delivery
  - » Handled project logistics and provided guidance to the NUCAPS science team to
    - – Get links to documentation templates
    - – Update schedules
    - – Review process (advising on content, reviewing ATBD slides)
    - – Assisting with access to tools and data sets, paperwork to access development hardware

Summary

- The STAR AIT role consists of working with science teams and the stakeholders to do the following:
  - » Conducting a standard set of project reviews
  - » Generating a standard set of documentation
  - » Stakeholder interaction
  - » Risk tracking and mitigation
  - » Code cleanup
- The STAR AIT R2O process is to each project depending on the scale, scope, and schedule
- The intended outcome of all this effort is meant to improve the lives of algorithm developers, operations, and end users so
  - » They can do their jobs
  - » Projects can enhance user readiness
  - » Reduce transition costs
  - » Improve maintainability of code in the long term