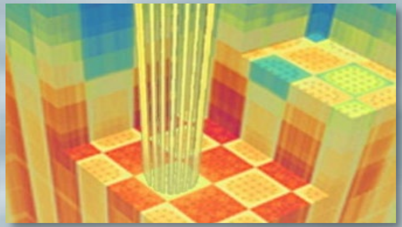
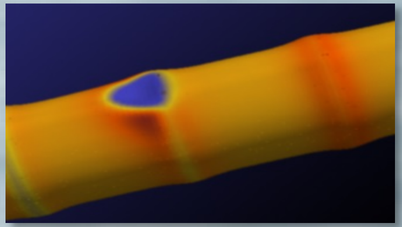
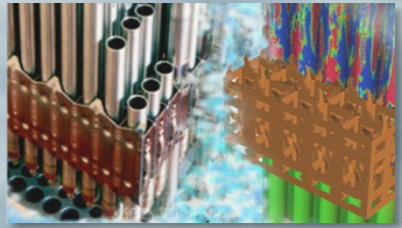
VERA Quality Control and Releases

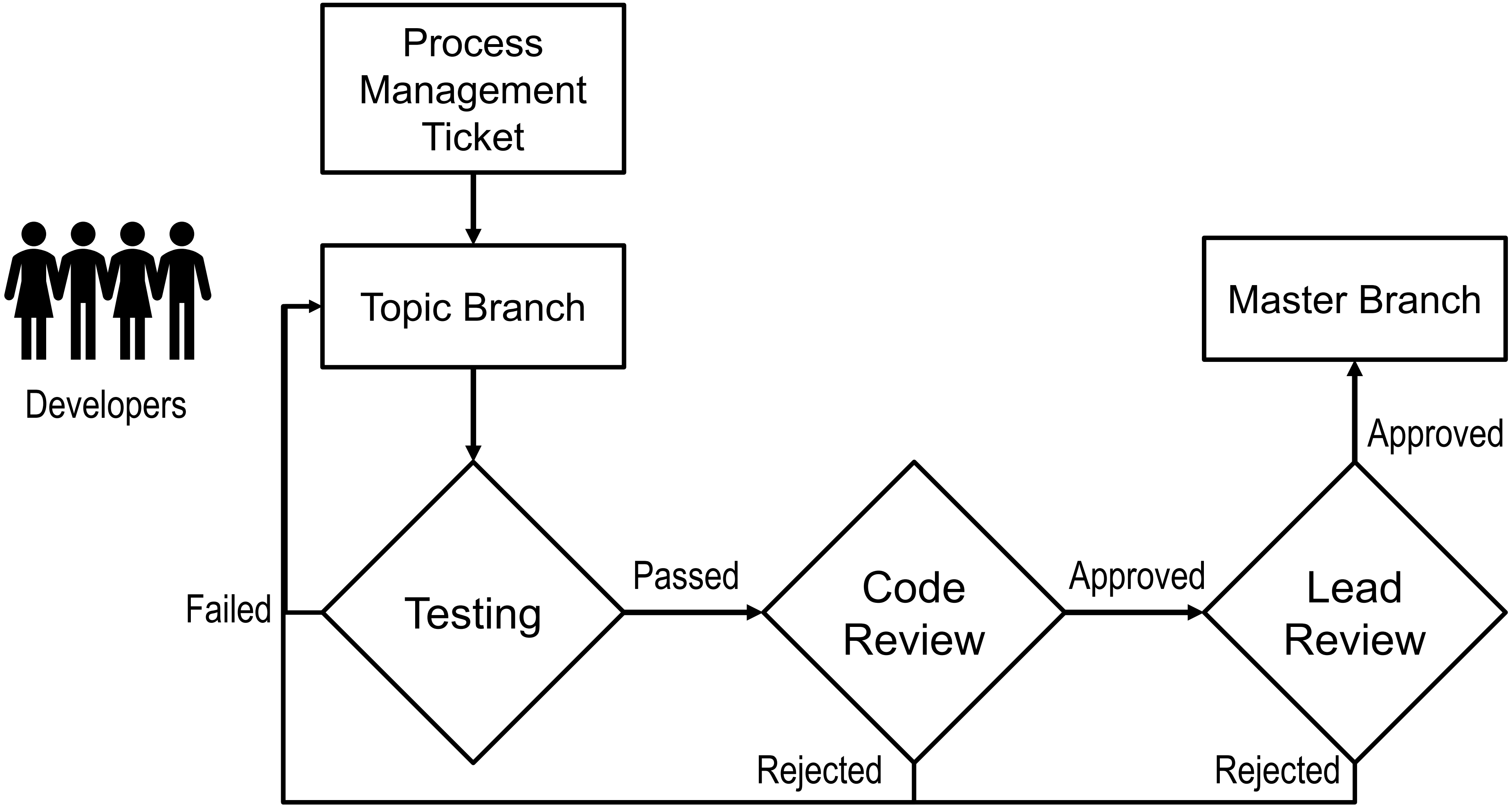
VERA Workshop
February 11, 2019
VERA Users Group Meeting
Oak Ridge National Laboratory

The Consortium for Advanced Simulation of LWRs
A DOE Energy Innovation Hub

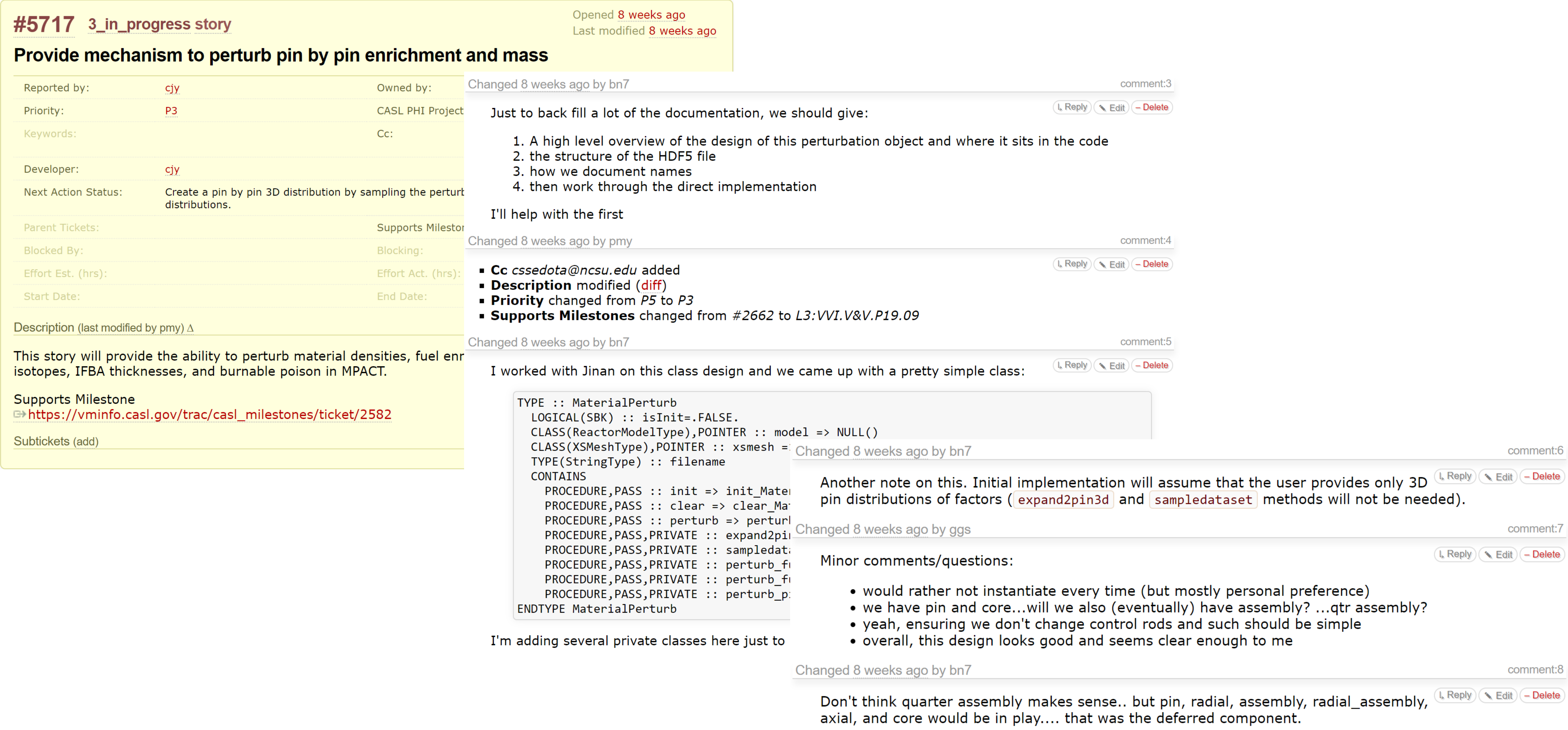
U.S. DEPARTMENT OF ENERGY

VERA-Component Software Development Practices



Each code team has their own workflow

Process Management



#5717 3_in_progress story

Opened 8 weeks ago
Last modified 8 weeks ago

Provide mechanism to perturb pin by pin enrichment and mass

Reported by:	cjy	Owned by:	
Priority:	P3	CASL PHI Project	
Keywords:		Cc:	
Developer:	cjy		
Next Action Status:	Create a pin by pin 3D distribution by sampling the perturb distributions.		
Parent Tickets:		Supports Milestor	
Blocked By:		Blocking:	
Effort Est. (hrs):		Effort Act. (hrs):	
Start Date:		End Date:	

Description (last modified by pmy) Δ

This story will provide the ability to perturb material densities, fuel enr isotopes, IFBA thicknesses, and burnable poison in MPACT.

Supports Milestone
https://vminfo.casl.gov/trac/casl_milestones/ticket/2582

Subtickets (add)

Changed 8 weeks ago by bn7 comment:3

Reply Edit – Delete

Just to back fill a lot of the documentation, we should give:

1. A high level overview of the design of this perturbation object and where it sits in the code
2. the structure of the HDF5 file
3. how we document names
4. then work through the direct implementation

I'll help with the first

Changed 8 weeks ago by pmy comment:4

Reply Edit – Delete

- **Cc** *cssedota@ncsu.edu* added
- **Description** modified (diff)
- **Priority** changed from *P5* to *P3*
- **Supports Milestones** changed from *#2662* to *L3:VVI.V&V.P19.09*

Changed 8 weeks ago by bn7 comment:5

Reply Edit – Delete

I worked with Jinan on this class design and we came up with a pretty simple class:

```
TYPE :: MaterialPerturb
  LOGICAL(SBK) :: isInit=.FALSE.
  CLASS(ReactorModelType),POINTER :: model => NULL()
  CLASS(XSMeshType),POINTER :: xsmesh =
  TYPE(StringType) :: filename
  CONTAINS
    PROCEDURE,PASS :: init => init_Mater
    PROCEDURE,PASS :: clear => clear_Ma
    PROCEDURE,PASS :: perturb => perturl
    PROCEDURE,PASS,PRIVATE :: expand2pi
    PROCEDURE,PASS,PRIVATE :: sampledat
    PROCEDURE,PASS,PRIVATE :: perturb_f
    PROCEDURE,PASS,PRIVATE :: perturb_f
    PROCEDURE,PASS,PRIVATE :: perturb_p
ENDTYPE MaterialPerturb
```

I'm adding several private classes here just to

Changed 8 weeks ago by bn7 comment:6

Reply Edit – Delete

Another note on this. Initial implementation will assume that the user provides only 3D pin distributions of factors (`expand2pin3d` and `sampledataset` methods will not be needed).

Changed 8 weeks ago by ggs comment:7

Reply Edit – Delete

Minor comments/questions:

- would rather not instantiate every time (but mostly personal preference)
- we have pin and core...will we also (eventually) have assembly? ...qtr assembly?
- yeah, ensuring we don't change control rods and such should be simple
- overall, this design looks good and seems clear enough to me

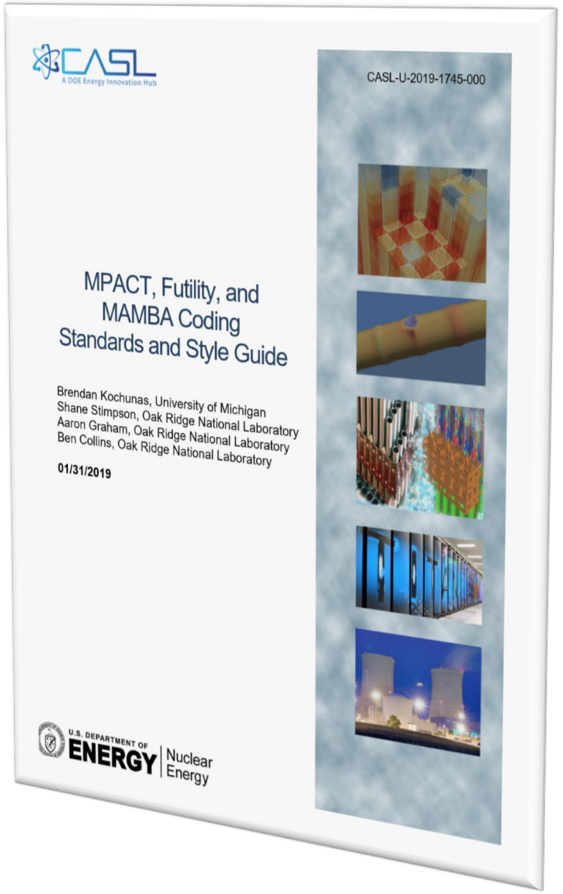
Changed 8 weeks ago by bn7 comment:8

Reply Edit – Delete

Don't think quarter assembly makes sense.. but pin, radial, assembly, radial_assembly, axial, and core would be in play.... that was the deferred component.

Code Development

- Detailed coding standards
- Inline documentation

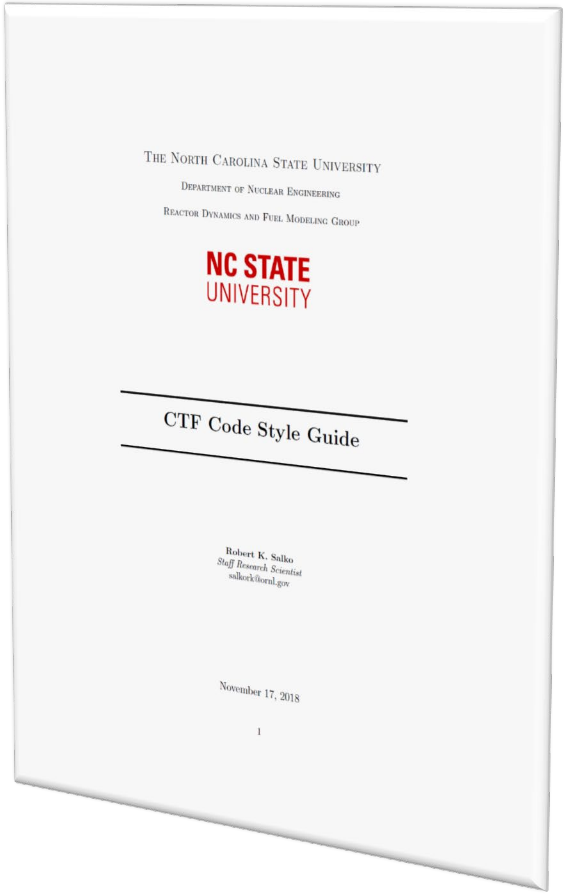


CASL-U-2019-1745-000

MPACT, Futility, and MAMBA Coding Standards and Style Guide

Brendan Kochunas, University of Michigan
Shane Stimpson, Oak Ridge National Laboratory
Aaron Graham, Oak Ridge National Laboratory
Ben Collins, Oak Ridge National Laboratory

01/31/2019



The North Carolina State University

Department of Nuclear Engineering

Reactor Dynamics and Fuel Modeling Group

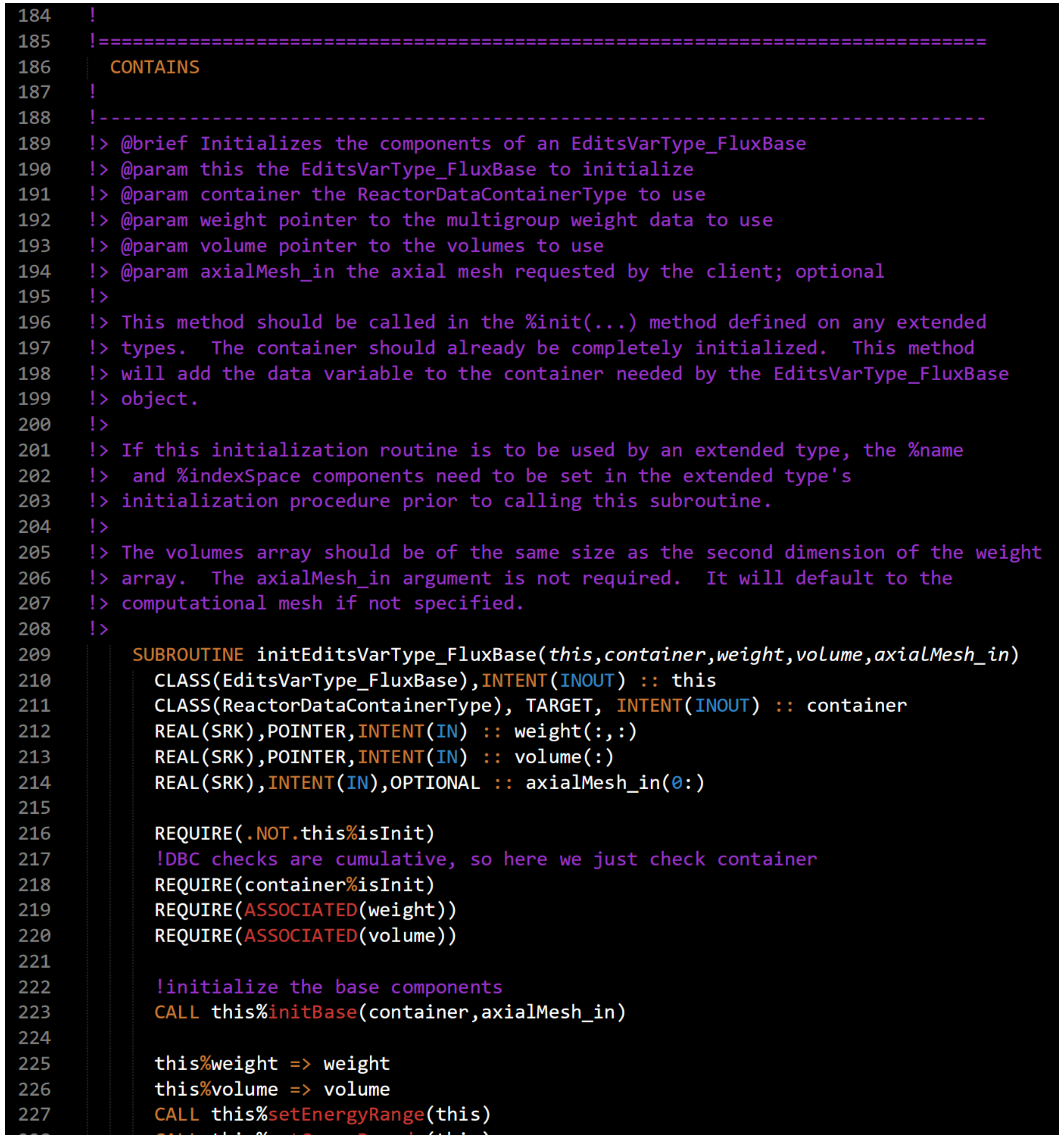
NC STATE UNIVERSITY

CTF Code Style Guide

Robert K. Salko
Staff Research Scientist
salkork@ornl.gov

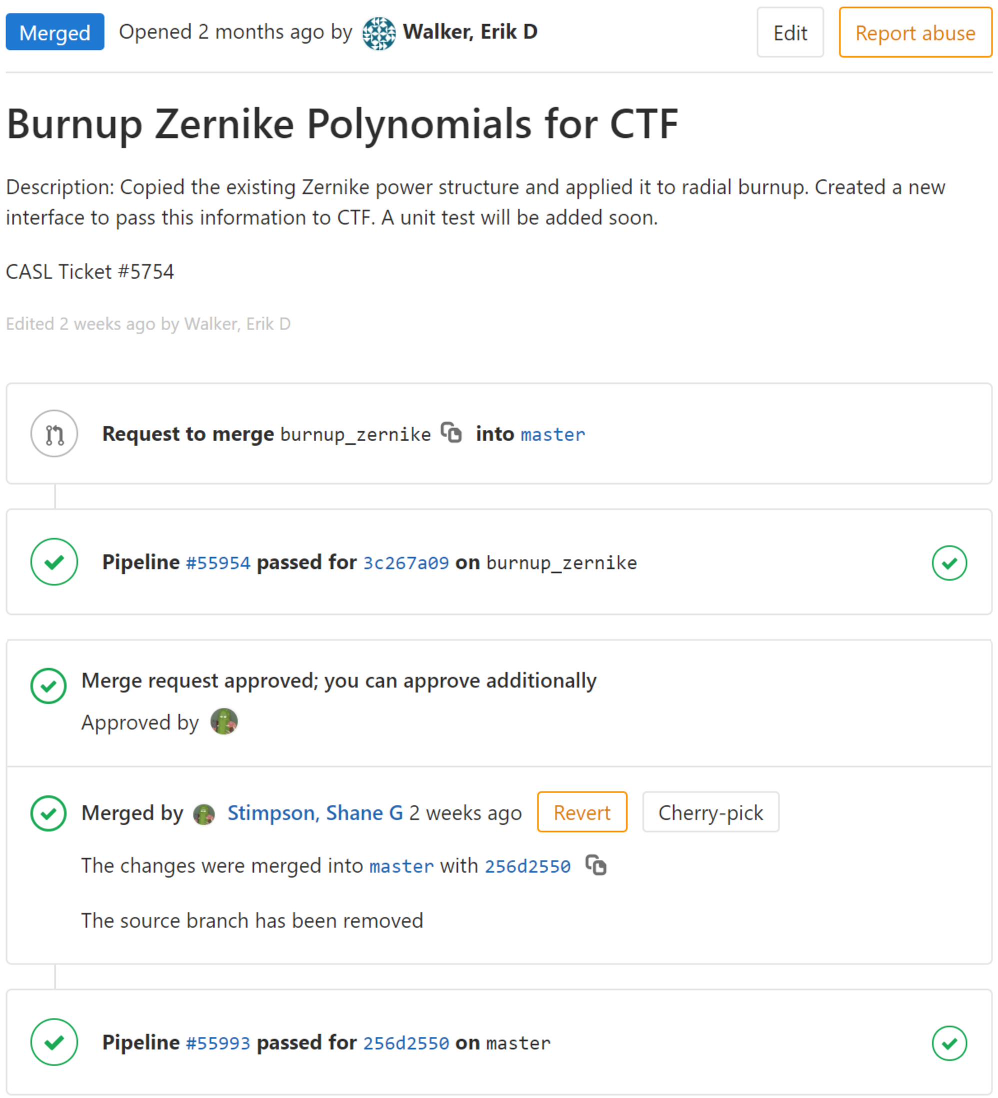
November 17, 2018

1



```fortran
!
!===============================================================================
  CONTAINS
!
!-------------------------------------------------------------------------------
!> @brief Initializes the components of an EditsVarType_FluxBase
!> @param this the EditsVarType_FluxBase to initialize
!> @param container the ReactorDataContainerType to use
!> @param weight pointer to the multigroup weight data to use
!> @param volume pointer to the volumes to use
!> @param axialMesh_in the axial mesh requested by the client; optional
!>
!> This method should be called in the %init(...) method defined on any extended
!> types.  The container should already be completely initialized.  This method
!> will add the data variable to the container needed by the EditsVarType_FluxBase
!> object.
!>
!> If this initialization routine is to be used by an extended type, the %name
!>  and %indexSpace components need to be set in the extended type's
!> initialization procedure prior to calling this subroutine.
!>
!> The volumes array should be of the same size as the second dimension of the weight
!> array.  The axialMesh_in argument is not required.  It will default to the
!> computational mesh if not specified.
!>
    SUBROUTINE initEditsVarType_FluxBase(this,container,weight,volume,axialMesh_in)
      CLASS(EditsVarType_FluxBase),INTENT(INOUT) :: this
      CLASS(ReactorDataContainerType), TARGET, INTENT(INOUT) :: container
      REAL(SRK),POINTER,INTENT(IN) :: weight(:,:)
      REAL(SRK),POINTER,INTENT(IN) :: volume(:)
      REAL(SRK),INTENT(IN),OPTIONAL :: axialMesh_in(0:)

      REQUIRE(.NOT.this%isInit)
      !DBC checks are cumulative, so here we just check container
      REQUIRE(container%isInit)
      REQUIRE(ASSOCIATED(weight))
      REQUIRE(ASSOCIATED(volume))

      !initialize the base components
      CALL this%initBase(container,axialMesh_in)

      this%weight => weight
      this%volume => volume
      CALL this%setEnergyRange(this)
```

VERA Component Code Reviews



Merged Opened 2 months ago by Walker, Erik D Edit Report abuse

Burnup Zernike Polynomials for CTF

Description: Copied the existing Zernike power structure and applied it to radial burnup. Created a new interface to pass this information to CTF. A unit test will be added soon.

CASL Ticket #5754

Edited 2 weeks ago by Walker, Erik D

Request to merge `burnup_zernike` **into** `master`

Pipeline #55954 **passed for** `3c267a09` **on** `burnup_zernike`

Merge request approved; you can approve additionally

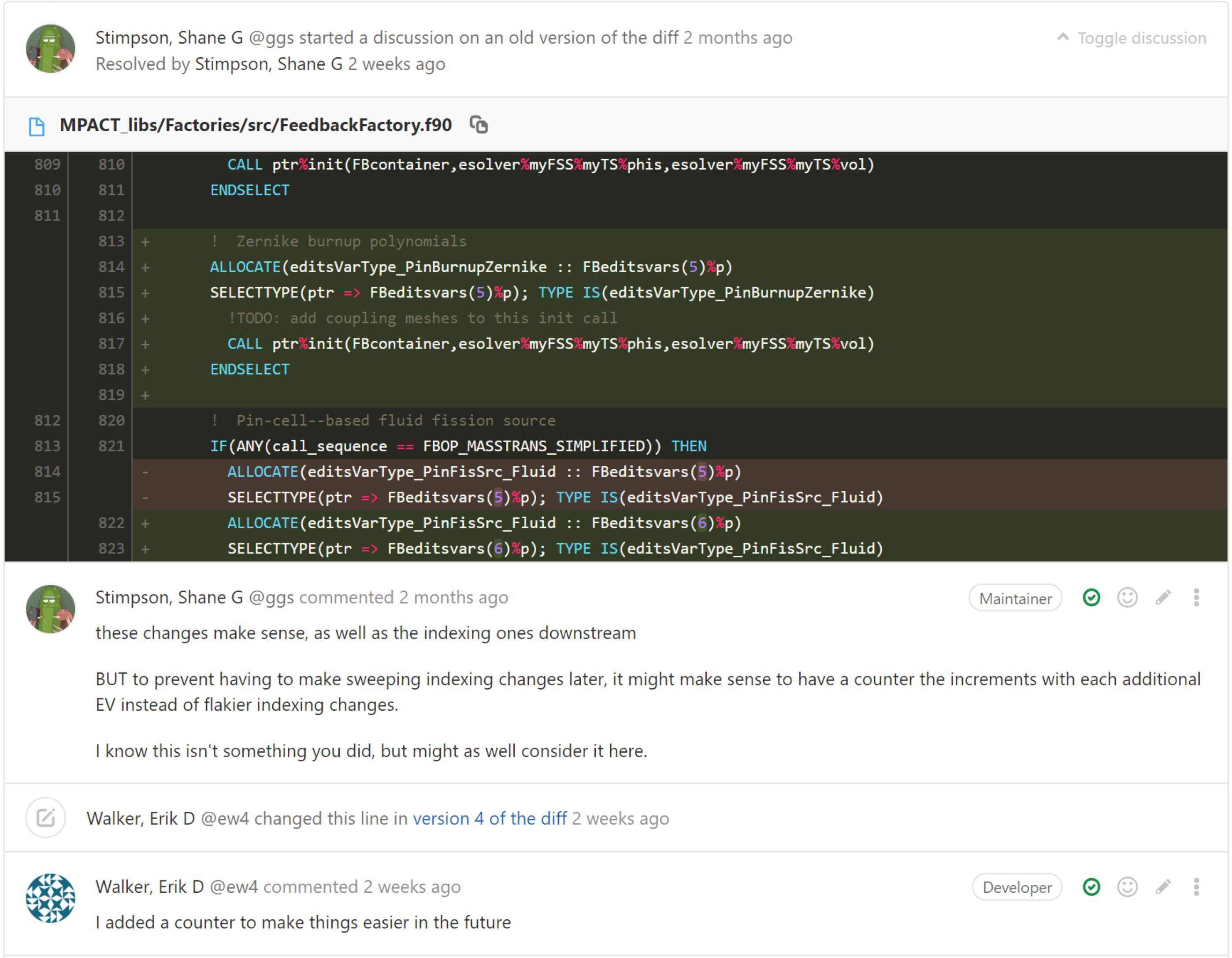
Approved by

Merged by Stimpson, Shane G 2 weeks ago Revert Cherry-pick

The changes were merged into `master` with `256d2550`

The source branch has been removed

Pipeline #55993 **passed for** `256d2550` **on** `master`



Stimpson, Shane G @ggs started a discussion on an old version of the diff 2 months ago
Resolved by Stimpson, Shane G 2 weeks ago

Toggle discussion

MPACT_libs/Factories/src/FeedbackFactory.f90

```
       CALL ptr%init(FBcontainer,esolver%myFSS%myTS%phis,esolver%myFSS%myTS%vol)
     ENDSELECT

+    !  Zernike burnup polynomials
+    ALLOCATE(editsVarType_PinBurnupZernike :: FBeditsvars(5)%p)
+    SELECTTYPE(ptr => FBeditsvars(5)%p); TYPE IS(editsVarType_PinBurnupZernike)
+      !TODO: add coupling meshes to this init call
+      CALL ptr%init(FBcontainer,esolver%myFSS%myTS%phis,esolver%myFSS%myTS%vol)
+    ENDSELECT
+
     !  Pin-cell--based fluid fission source
     IF(ANY(call_sequence == FBOP_MASSTRANS_SIMPLIFIED)) THEN
-      ALLOCATE(editsVarType_PinFisSrc_Fluid :: FBeditsvars(5)%p)
-      SELECTTYPE(ptr => FBeditsvars(5)%p); TYPE IS(editsVarType_PinFisSrc_Fluid)
+      ALLOCATE(editsVarType_PinFisSrc_Fluid :: FBeditsvars(6)%p)
+      SELECTTYPE(ptr => FBeditsvars(6)%p); TYPE IS(editsVarType_PinFisSrc_Fluid)
```

Stimpson, Shane G @ggs commented 2 months ago Maintainer

these changes make sense, as well as the indexing ones downstream

BUT to prevent having to make sweeping indexing changes later, it might make sense to have a counter the increments with each additional EV instead of flakier indexing changes.

I know this isn't something you did, but might as well consider it here.

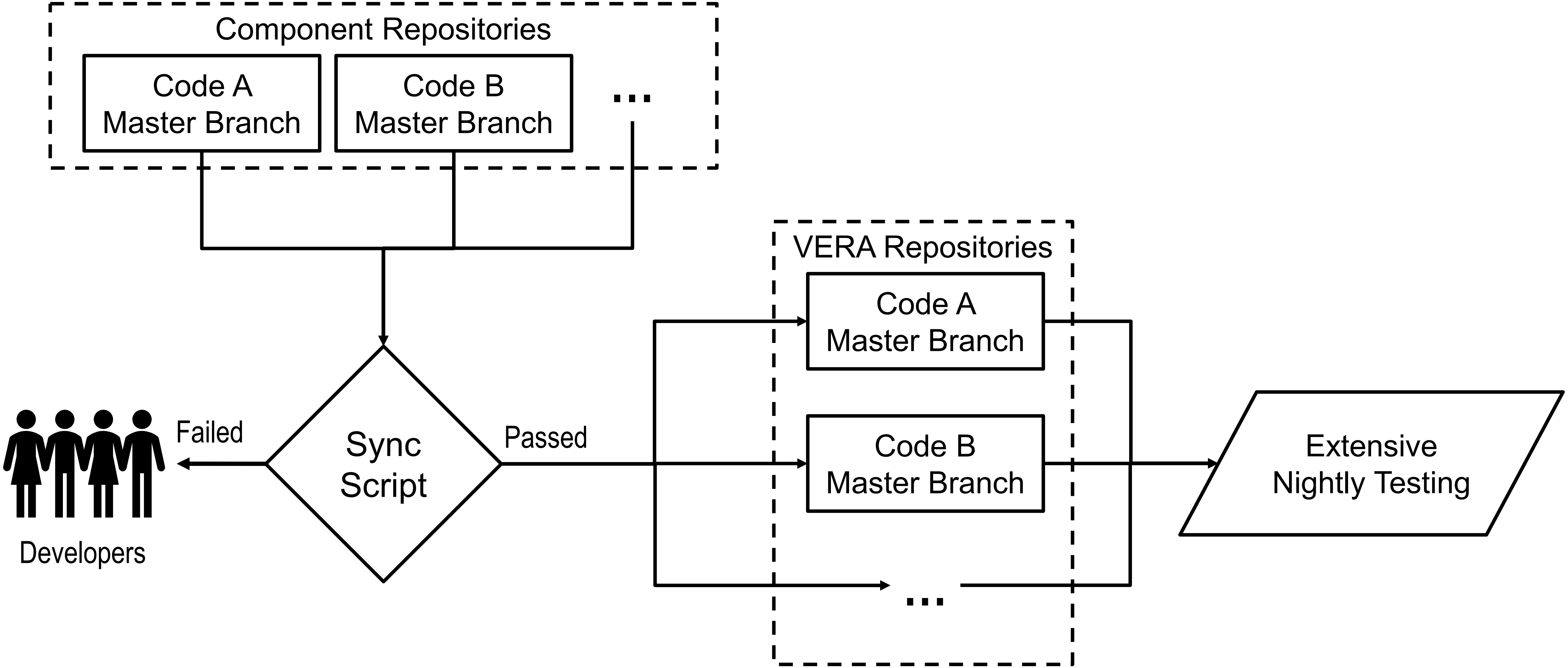
Walker, Erik D @ew4 changed this line in version 4 of the diff 2 weeks ago

Walker, Erik D @ew4 commented 2 weeks ago Developer

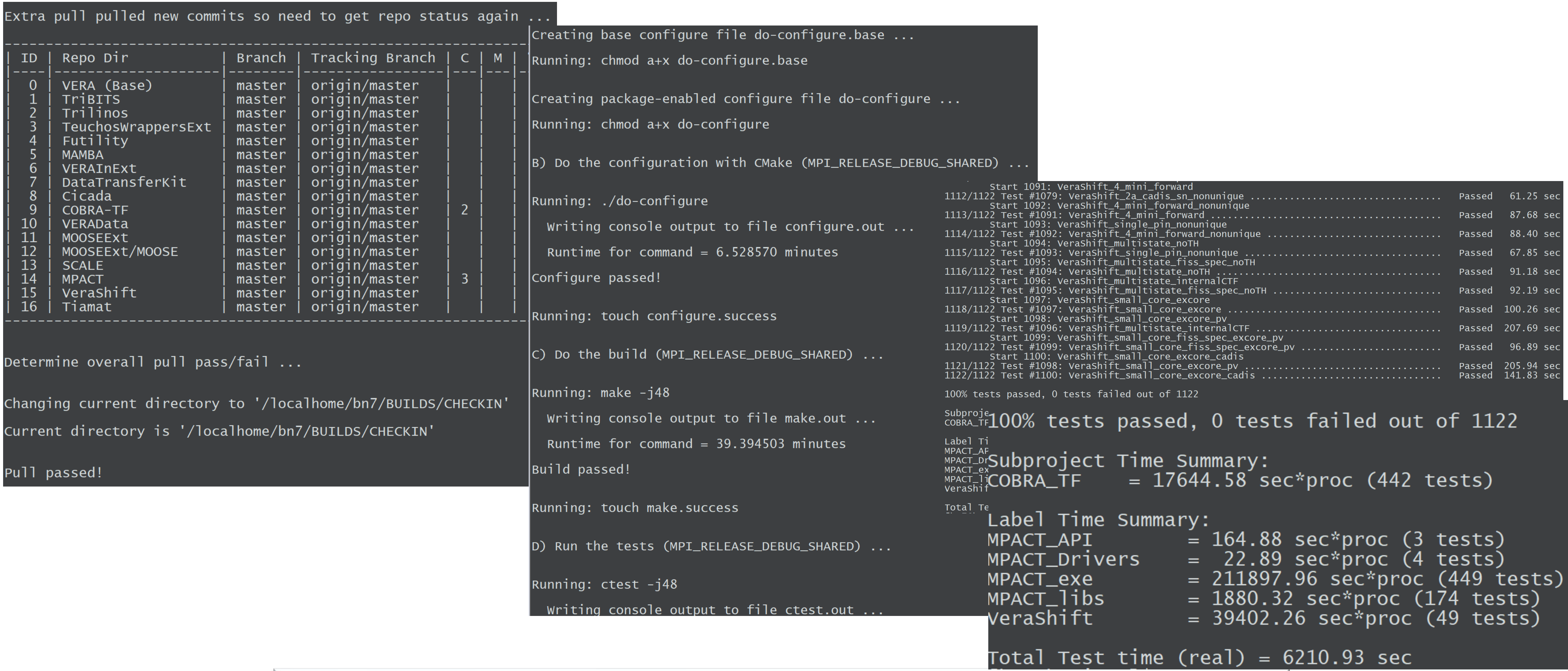
I added a counter to make things easier in the future

Line-by-line review of changes

VERA Integration



Testing prior to integration with VERA



```
Extra pull pulled new commits so need to get repo status again ...

-------------------------------------------------------------------------------
| ID | Repo Dir              | Branch | Tracking Branch | C | M | ? |
|----|-----------------------|--------|-----------------|---|---|---|
|  0 | VERA (Base)           | master | origin/master   |   |   |   |
|  1 | TriBITS               | master | origin/master   |   |   |   |
|  2 | Trilinos              | master | origin/master   |   |   |   |
|  3 | TeuchosWrappersExt    | master | origin/master   |   |   |   |
|  4 | Futility              | master | origin/master   |   |   |   |
|  5 | MAMBA                 | master | origin/master   |   |   |   |
|  6 | VERAInExt             | master | origin/master   |   |   |   |
|  7 | DataTransferKit       | master | origin/master   |   |   |   |
|  8 | Cicada                | master | origin/master   |   |   |   |
|  9 | COBRA-TF              | master | origin/master   | 2 |   |   |
| 10 | VERAData              | master | origin/master   |   |   |   |
| 11 | MOOSEExt              | master | origin/master   |   |   |   |
| 12 | MOOSEExt/MOOSE        | master | origin/master   |   |   |   |
| 13 | SCALE                 | master | origin/master   |   |   |   |
| 14 | MPACT                 | master | origin/master   | 3 |   |   |
| 15 | VeraShift             | master | origin/master   |   |   |   |
| 16 | Tiamat                | master | origin/master   |   |   |   |
-------------------------------------------------------------------------------


Determine overall pull pass/fail ...

Changing current directory to '/localhome/bn7/BUILDS/CHECKIN'

Current directory is '/localhome/bn7/BUILDS/CHECKIN'

Pull passed!
```

```
Creating base configure file do-configure.base ...

Running: chmod a+x do-configure.base

Creating package-enabled configure file do-configure ...

Running: chmod a+x do-configure

B) Do the configuration with CMake (MPI_RELEASE_DEBUG_SHARED) ...

Running: ./do-configure

  Writing console output to file configure.out ...

  Runtime for command = 6.528570 minutes

Configure passed!

Running: touch configure.success

C) Do the build (MPI_RELEASE_DEBUG_SHARED) ...

Running: make -j48

  Writing console output to file make.out ...

  Runtime for command = 39.394503 minutes

Build passed!

Running: touch make.success

D) Run the tests (MPI_RELEASE_DEBUG_SHARED) ...

Running: ctest -j48

  Writing console output to file ctest.out ...
```

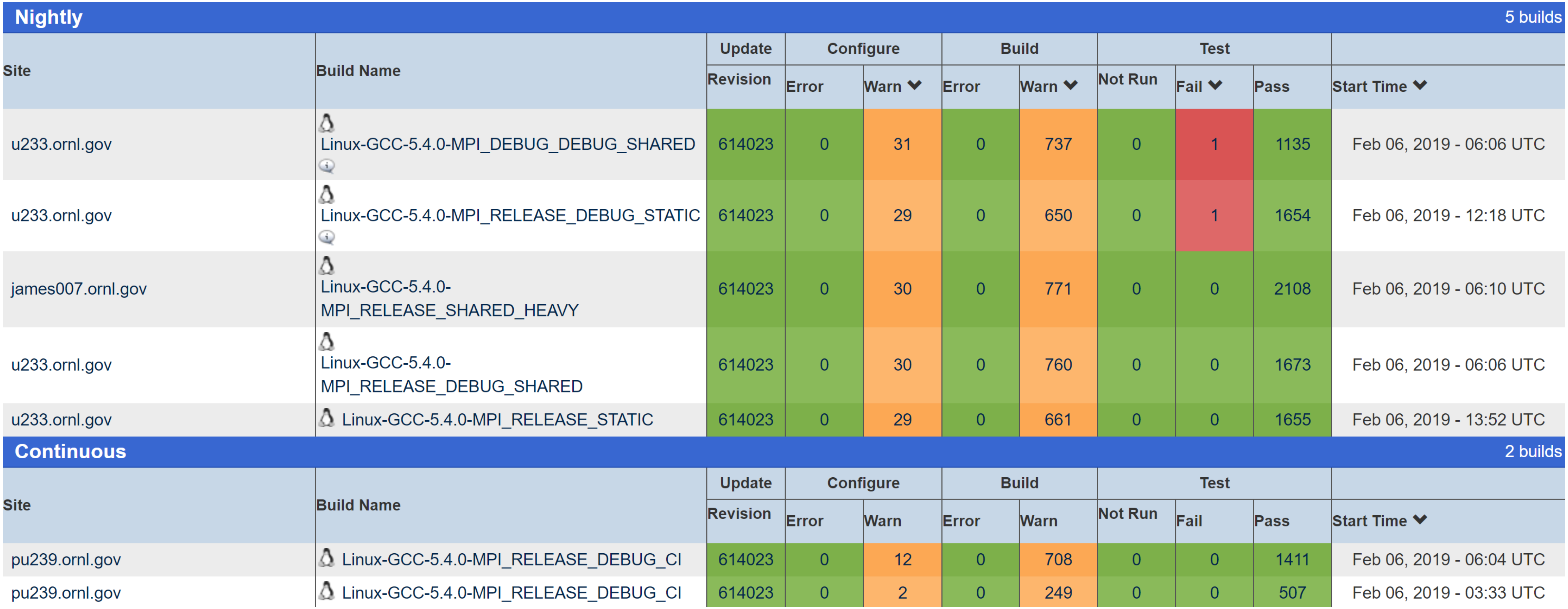
```
        Start 1091: VeraShift_4_mini_forward
1112/1122 Test #1079: VeraShift_2a_cadis_sn_nonunique ...............................   Passed   61.25 sec
        Start 1092: VeraShift_4_mini_forward_nonunique
1113/1122 Test #1091: VeraShift_4_mini_forward ......................................   Passed   87.68 sec
        Start 1093: VeraShift_single_pin_nonunique
1114/1122 Test #1092: VeraShift_4_mini_forward_nonunique ............................   Passed   88.40 sec
        Start 1094: VeraShift_multistate_noTH
1115/1122 Test #1093: VeraShift_single_pin_nonunique ................................   Passed   67.85 sec
        Start 1095: VeraShift_multistate_fiss_spec_noTH
1116/1122 Test #1094: VeraShift_multistate_noTH .....................................   Passed   91.18 sec
        Start 1096: VeraShift_multistate_internalCTF
1117/1122 Test #1095: VeraShift_multistate_fiss_spec_noTH ...........................   Passed   92.19 sec
        Start 1097: VeraShift_small_core_excore
1118/1122 Test #1097: VeraShift_small_core_excore ...................................   Passed  100.26 sec
        Start 1098: VeraShift_small_core_excore_pv
1119/1122 Test #1096: VeraShift_multistate_internalCTF ..............................   Passed  207.69 sec
        Start 1099: VeraShift_small_core_fiss_spec_excore_pv
1120/1122 Test #1099: VeraShift_small_core_fiss_spec_excore_pv ......................   Passed   96.89 sec
        Start 1100: VeraShift_small_core_excore_cadis
1121/1122 Test #1098: VeraShift_small_core_excore_pv ................................   Passed  205.94 sec
1122/1122 Test #1100: VeraShift_small_core_excore_cadis .............................   Passed  141.83 sec

100% tests passed, 0 tests failed out of 1122
```

```
100% tests passed, 0 tests failed out of 1122

Subproject Time Summary:
COBRA_TF    = 17644.58 sec*proc (442 tests)

Label Time Summary:
MPACT_API        = 164.88 sec*proc (3 tests)
MPACT_Drivers    =  22.89 sec*proc (4 tests)
MPACT_exe        = 211897.96 sec*proc (449 tests)
MPACT_libs       = 1880.32 sec*proc (174 tests)
VeraShift        = 39402.26 sec*proc (49 tests)

Total Test time (real) = 6210.93 sec
```

Full nightly test suite run before integration into VERA

The Consortium for Advanced Simulation of LWRs
A DOE Energy Innovation Hub

Continuous and Nightly VERA testing



Nightly 5 builds

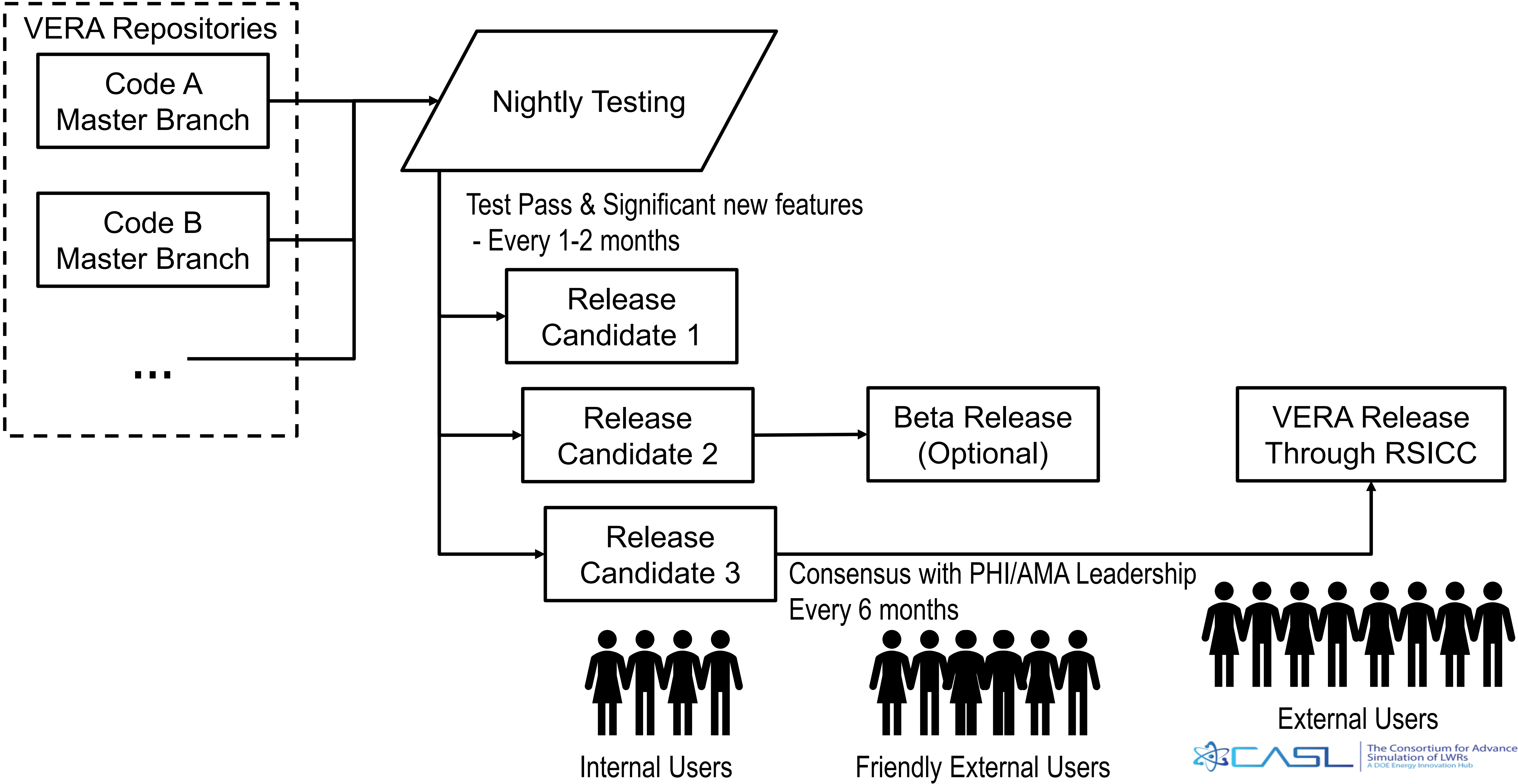
Site	Build Name	Update	Configure		Build		Test			
		Revision	Error	Warn	Error	Warn	Not Run	Fail	Pass	Start Time
u233.ornl.gov	Linux-GCC-5.4.0-MPI_DEBUG_DEBUG_SHARED	614023	0	31	0	737	0	1	1135	Feb 06, 2019 - 06:06 UTC
u233.ornl.gov	Linux-GCC-5.4.0-MPI_RELEASE_DEBUG_STATIC	614023	0	29	0	650	0	1	1654	Feb 06, 2019 - 12:18 UTC
james007.ornl.gov	Linux-GCC-5.4.0-MPI_RELEASE_SHARED_HEAVY	614023	0	30	0	771	0	0	2108	Feb 06, 2019 - 06:10 UTC
u233.ornl.gov	Linux-GCC-5.4.0-MPI_RELEASE_DEBUG_SHARED	614023	0	30	0	760	0	0	1673	Feb 06, 2019 - 06:06 UTC
u233.ornl.gov	Linux-GCC-5.4.0-MPI_RELEASE_STATIC	614023	0	29	0	661	0	0	1655	Feb 06, 2019 - 13:52 UTC

Continuous 2 builds

Site	Build Name	Update	Configure		Build		Test			
		Revision	Error	Warn	Error	Warn	Not Run	Fail	Pass	Start Time
pu239.ornl.gov	Linux-GCC-5.4.0-MPI_RELEASE_DEBUG_CI	614023	0	12	0	708	0	0	1411	Feb 06, 2019 - 06:04 UTC
pu239.ornl.gov	Linux-GCC-5.4.0-MPI_RELEASE_DEBUG_CI	614023	0	2	0	249	0	0	507	Feb 06, 2019 - 03:33 UTC

>10,000 tests run daily

VERA Release Process



VERA Repositories

Code A
Master Branch

Code B
Master Branch

...

Nightly Testing

Test Pass & Significant new features
- Every 1-2 months

Release
Candidate 1

Release
Candidate 2

Beta Release
(Optional)

Release
Candidate 3

Consensus with PHI/AMA Leadership
Every 6 months

VERA Release
Through RSICC

Internal Users

Friendly External Users

External Users

VERA Deployment

- VERARun provides a mechanism to use “release” versions of the code on deployed machines

```
[~]$ verarun -l
     Available VERA versions:
       VERA_4.0RC3
       VERA_4.0RC2
       VERA_4.0RC1
       VERA_3.9
       VERA_3.8

[~]$ verarun –v VERA_4.0RC1 p7
```

Software Quality Objectives: The Path Toward NQA-1

- The goal of this work is to produce an NQA-1 certified version of the VERA suite.
- US Utilities are required to use NQA-1 software for safety calculations. Having an NQA-1 version of VERA will add value to the product by making adoption by utilities easier.

- Primary focus on year one is the core simulator and BISON

Organization of VERA for QA Purposes

- For QA purposes, VERA is organized as a collection of software “Products”.
- A product is one or more computer codes (executables) that can be used to perform analysis and meet software requirements. Products may also include utility programs that directly support the product.
- Each Product will have a separate Software Management Plan (SMP) and corresponding QA documentation.
- A Product Software Manager (PSM) will be assigned to each Product.
- Products may contain source code from external sources (e.g. PetSc, MPI, Origen, Futility).
  The QA of any external source code will be included in the QA of the product. (no CGD)
- Products may also have dependencies on other Products (e.g. MPACT depends on VERAIn, MPACT depends on CTF). The “higher-level code” must include the QA for the coupling.

Working Definitions

- **Product** – VERA component that is delivered to customers and is designed, developed and managed under a Software Management Plan (SMP).
- **External Library** – External source that is included in VERA Products. External libraries do not have a separate QA plan and the QA for these is included in the Product QA. (If an External library does have a QA plan, it should be classified as a Product).
- **Third Party Library (TPL)** – common designation in CASL to represent a library from an external source (MPI, GCC, CMake, etc.). TPL's will usually be considered as "External Libraries", but they may also be considered as Products if necessary.
- **Utility** – A separate executable or script that is used to support a product. The QA for the utility is included in the Product QA.
- **Dependency** – A Product may depend on another product. This relationship is a "dependency". The interface for the dependency must be defined and tested in one of the products.
- **Product Software Manager (PSM)** - Each Product has one individual responsible for the management of the QA for that Product

VERA Products

- It is important to split QA responsibilities into smaller pieces to manage more effectively across different development groups
- Product list may be slightly modified or added to in the future
- The following details for each product are not complete. Official details will be included in the product software inventories.
- First year focuses on first 3 components

1. VERA-IO
2. CTF
3. MPACT
4. Bison
5. MAMBA
6. Tiamat
7. VERAShift

1. VERA-IO

- Product Software Manager – Kevin Clarno
- Includes:
  - VERAIn – Common input processor used to generate XML files
  - VERAOut – Text based output utilitiy
  - VERARun – Run script to run many executables in parallel environment
  - VERAView – graphical output visualizer
- Includes libraries: none
- Depends on: none

2. CTF

- Product Software Manager – Bob Salko
- Includes:
  - CTF – Standalone subchannel T/H code
  - xml2ctf – input preprocessor
  - driver – standalone CTF driver
- Includes libraries: MPI, PetSc, Futility
- Depends on: VERA-IO and Mamba

3. MPACT

- Product Software Manager – Brendan Kochunas
- Includes:
  - Standalone neutronics executable
  - Executable coupled with CTF
  - Executable coupled with CTF+MAMBA
  - ORIGEN (subset)
  - Post-processing utilities
- Includes libraries: MPI, PetSc, Futility, etc.
- Depends on: VERA-IO, CTF, and MAMBA

4. Bison

- Product Software Manager – Rich Williamson
- Includes:
  - Bison – Standalone Fuel Performance Code
  - xml2bison – input preprocessor (?)
- Includes libraries: Moose, MPI, PetSc, libmesh, etc.
- Depends on: none

Bison QA will be done under INL QA management program

Other Components

- MAMBA
  - Includes:
    - MAMBA - standalone CRUD code
    - Driver utilities
  - Includes libraries: MPI, PetSc, Futility, etc.
  - Depends on: none
- Tiamat
  - Includes:
    - TIAMAT - Standalone automation of fuel performance code runs using HDF5 MPACT output file
    - May include drivers for other fuel performance codes such as FAST, FRAPCON and ENIGMA
  - Includes libraries: Moose, MPI, PetSc, Futility, etc.
  - Depends on: Bison
- VERAShift
  - Includes:
    - Subset of Shift (which also includes Omnibus, Denovo, … treated as library)
  - Includes libraries: MPI, PetSc, etc.
  - Depends on: MPACT

VERA QA Program Plan (QAPP)

- QAPP is the overarching quality plan for VERA
- QAPP defines:
  - Quality Requirements (high-level)
  - Procedures
  - Documentation
  - Training requirements
  - Roles and Responsibilities
- QAPP will define lower Product QA documentation
- QAPP is being independently developed with SLT and ORNL Performance Assurance and Quality organization

Work in Progress

Product QA Requirements

- Each Product will be required to generate the following documentation:
  a) Software Management Plan (SMP)
  b) Theory Manual(s)
  c) User Manuals(s)
  d) Software Test Plan (STP)
  e) Software Test Report (STR)

- This is a preliminary list and details will change based upon the development of the QAPP

Preliminary Requirements!

Questions?

CASL

A DOE Energy Innovation Hub

www.casl.gov