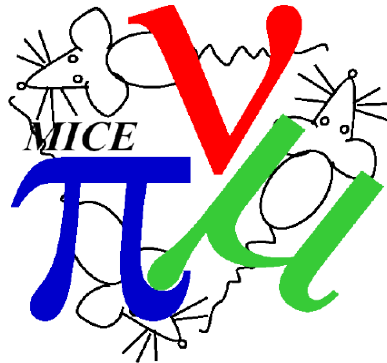




# G4MICE - Status and Plans

---

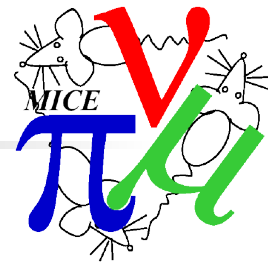


Chris Rogers,  
ASTeC,  
Rutherford Appleton Laboratory

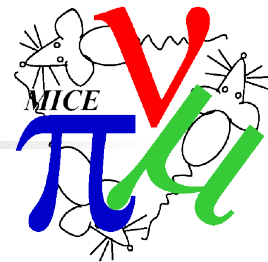




# Progress in G4MICE



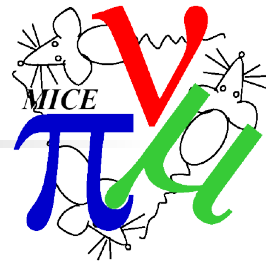
- New releases
  - mice-2-4-0
  - mice-2-4-1 imminent
    - All tests pass, documentation is ready
    - Want a quiet time to just check things over
  - “Roughly” 2 month release cycle
  - 27 open bugs
- Review of build procedure, dataflow, infrastructure
  - MAUS - C. Tunnell talk
- Code and Responsible persons
- Review of code QA and documentation
- New code



- User Documentation
- QA Process
- MC Simulation
  - Geometry model
- Digitisation
- Reconstruction
  - Segmentation fault
- Physics analysis tools
- Optics simulation of MICE (transport matrices)
- Event display
- Data quality check
- Calibration application(s)

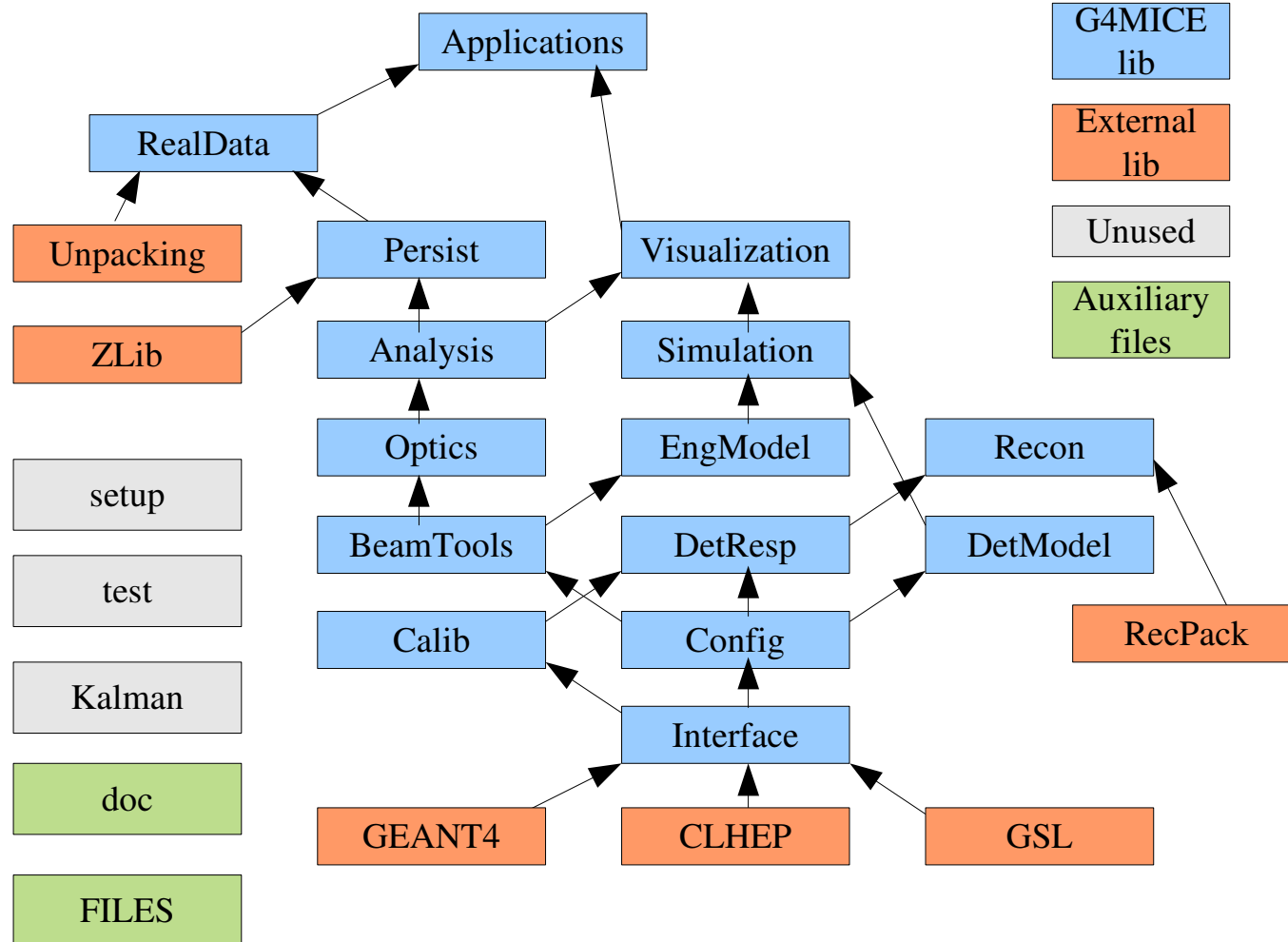
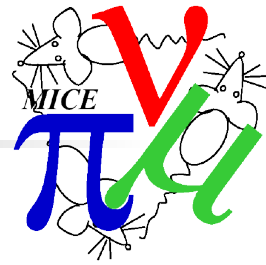


# Code as Engineering



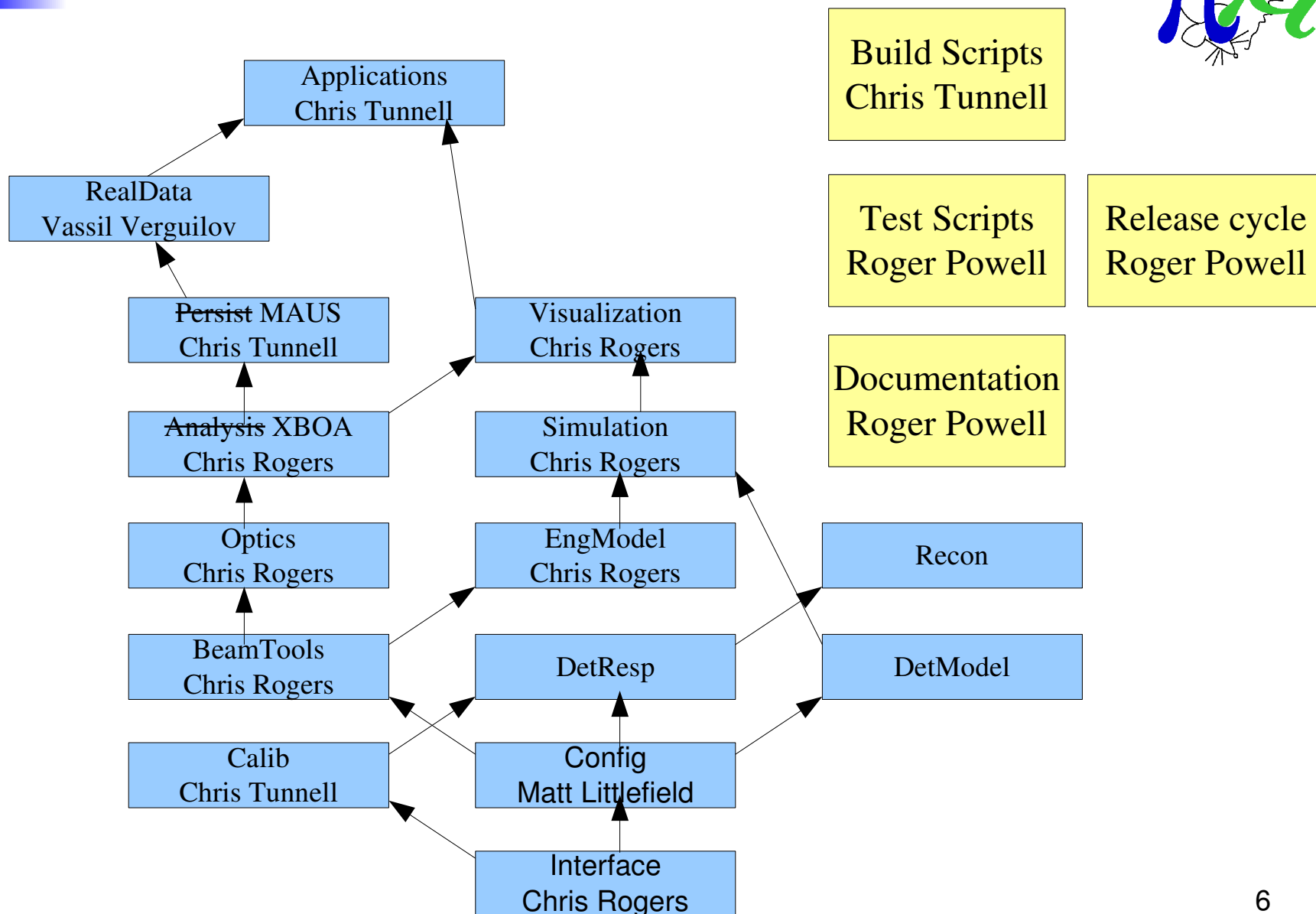
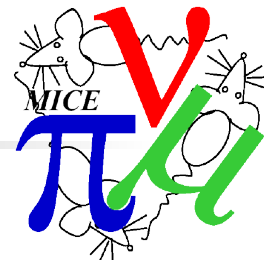
- Coding is a complex engineering problem
  - As G4MICE manager, it's my job to manage this complexity
  - We have brilliant coders making brilliant fancy algorithms
  - But somehow the bureaucracy of complex engineering gets us
    - Developers move on to bigger things
    - Code goes mouldy
    - People forget
- Keys to managing complexity
  - Clear interfaces
  - Rigorous and extensive testing
  - Rigorous and extensive documentation
- Not much fun!
  - We're physicists
  - We like doing fancy stuff
  - But the boring stuff will get us in the end
- My job is to manage the complexity
  - Use best practice/industry standard processes

# G4MICE Domains Diagram



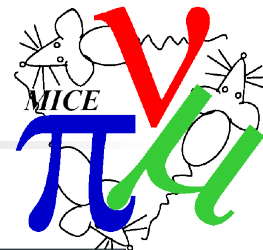
- Each domain contains ~ 10 objects each with ~ 20 functions
  - ~ few thousand custom components all of which need to work together

# G4MICE Code Ownership





# Recon Area



## **DetResp - Chris Tunnell**

Global - Chris Tunnell  
SciFi - David Adey  
EMR - Vassil Verguilov  
TOF - Yordan Karadzhov  
KL - Mariyan Bogomilov  
Ckov - Peter Sonnek  
Virtual - Chris Rogers

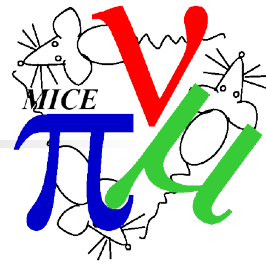
## **Recon - Chris Tunnell**

Global - Chris Tunnell  
+ Simon Fayer  
SciFi - David Adey  
EMR - Vassil Verguilov  
TOF - Yordan Karadzhov  
KL - Mariyan Bogomilov  
Ckov - Peter Sonnek  
Virtual - Chris Rogers

## **DetModel - Chris Tunnell**

Global - Chris Tunnell  
SciFi - David Adey  
EMR - Vassil Verguilov  
TOF - Yordan Karadzhov  
KL - Mariyan Bogomilov  
Ckov - Peter Sonnek  
Virtual - Chris Rogers

# Code QA and Documentation



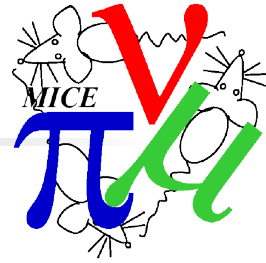
- Code QA
  - G4MICE has issues with stability and usability
    - Reconstruction code returns segmentation fault and has done for years
      - Knowledgeable, competent developers take months to get it to reconstruct
    - Bugs all over the place
    - Crashes in control room
  - Need QA process to fix that
    - Unit tests allow to run every line of code at least once (and check it does what we think)
    - Application tests allow to run a few configurations and check certain conditions
- Documentation
  - We are turning over almost the entire G4MICE crew this year
  - MICE will need to operate for at least another 3 years
  - We need to ease the turnover process
  - Documentation!
    - Also leads to better quality code





# Code QA and Documentation

---



- Unit testing framework implemented and operational
  - A few developers have contributed
  - Release 2-4-1 does not include changes that have not been tested
  - Rule will apply to future releases
  - Writing unit tests will take many months

# Code QA and Documentation



cr67 on linux-rp26: /home/cr67/G4MICE/MICETESTS-dev/gUnit/mgr - Shell No. 2 - Konsole

Session Edit View Bookmarks Settings Help

```
[ RUN      ] BTMultipoleTest.BTMultipoleConstructorTest
[ OK       ] BTMultipoleTest.BTMultipoleConstructorTest (0 ms)
[ RUN      ] BTMultipoleTest.BTMultipoleTransformToRotated
[ OK       ] BTMultipoleTest.BTMultipoleTransformToRotated (37 ms)
[ RUN      ] BTMultipoleTest.GetFieldValueTest_HardEdged
[ OK       ] BTMultipoleTest.GetFieldValueTest_HardEdged (0 ms)
[ RUN      ] BTMultipoleTest.BTMultipoleGetConstTest
[ OK       ] BTMultipoleTest.BTMultipoleGetConstTest (0 ms)
[ RUN      ] BTMultipoleTest.GetFieldValueTest_TanhDipole
[ OK       ] BTMultipoleTest.GetFieldValueTest_TanhDipole (4 ms)
[ RUN      ] BTMultipoleTest.GetFieldValueTest_TanhQuad
[ OK       ] BTMultipoleTest.GetFieldValueTest_TanhQuad (4 ms)
[ RUN      ] BTMultipoleTest.GetFieldValueTest_Enge
[ OK       ] BTMultipoleTest.GetFieldValueTest_Enge (1 ms)
[ RUN      ] BTMultipoleTest.PrintTest
[ OK       ] BTMultipoleTest.PrintTest (0 ms)
[-----] 9 tests from BTMultipoleTest (46 ms total)

[-----] 3 tests from BTFieldConstructorTest
[ RUN      ] BTFieldConstructorTest.EndFieldTest
[ OK       ] BTFieldConstructorTest.EndFieldTest (5 ms)
[ RUN      ] BTFieldConstructorTest.GetMultipoleTest
[ OK       ] BTFieldConstructorTest.GetMultipoleTest (4 ms)
[ RUN      ] BTFieldConstructorTest.GetCombinedFunctionTest
[ OK       ] BTFieldConstructorTest.GetCombinedFunctionTest (5 ms)
[-----] 3 tests from BTFieldConstructorTest (14 ms total)

[-----] 10 tests from MiceModToG4SolidTest
[ RUN      ] MiceModToG4SolidTest.checkDimTest
[ OK       ] MiceModToG4SolidTest.checkDimTest (0 ms)
[ RUN      ] MiceModToG4SolidTest.buildWedgeTest
[ OK       ] MiceModToG4SolidTest.buildWedgeTest (2 ms)
[ RUN      ] MiceModToG4SolidTest.buildBoxTest
[ OK       ] MiceModToG4SolidTest.buildBoxTest (2 ms)
[ RUN      ] MiceModToG4SolidTest.buildCylinderTest
[ OK       ] MiceModToG4SolidTest.buildCylinderTest (1 ms)
[ RUN      ] MiceModToG4SolidTest.buildTubeTest
[ OK       ] MiceModToG4SolidTest.buildTubeTest (2 ms)
[ RUN      ] MiceModToG4SolidTest.buildSphereTest
[ OK       ] MiceModToG4SolidTest.buildSphereTest (5 ms)
[ RUN      ] MiceModToG4SolidTest.buildPolyconeTest
[ OK       ] MiceModToG4SolidTest.buildPolyconeTest (0 ms)
[ RUN      ] MiceModToG4SolidTest.buildMultipoleTest
[ OK       ] MiceModToG4SolidTest.buildMultipoleTest (1 ms)
[ RUN      ] MiceModToG4SolidTest.buildTorusTest
[ OK       ] MiceModToG4SolidTest.buildTorusTest (4 ms)
[ RUN      ] MiceModToG4SolidTest.buildEllipticalConeTest
[ OK       ] MiceModToG4SolidTest.buildEllipticalConeTest (2 ms)
[-----] 10 tests from MiceModToG4SolidTest (19 ms total)

[-----] Global test environment tear-down
[=====] 61 tests from 13 test cases ran. (876 ms total)
[ PASSED ] 60 tests.
[ FAILED ] 1 test, listed below:
[ FAILED ] MiceModuleTest.printTree
```

1 FAILED TEST  
gUnit/mgr> █

Shell Shell No. 2 Shell No. 3

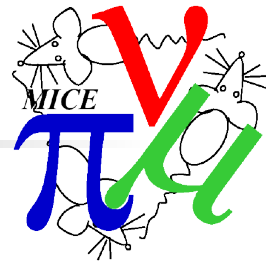
cr67 on linux-rp26 MICE Collaboration

MultipoleConfigurat

08:24

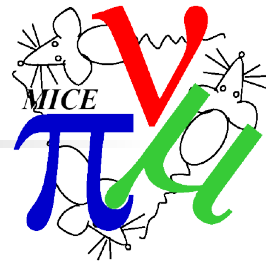
London

# Code QA and Documentation

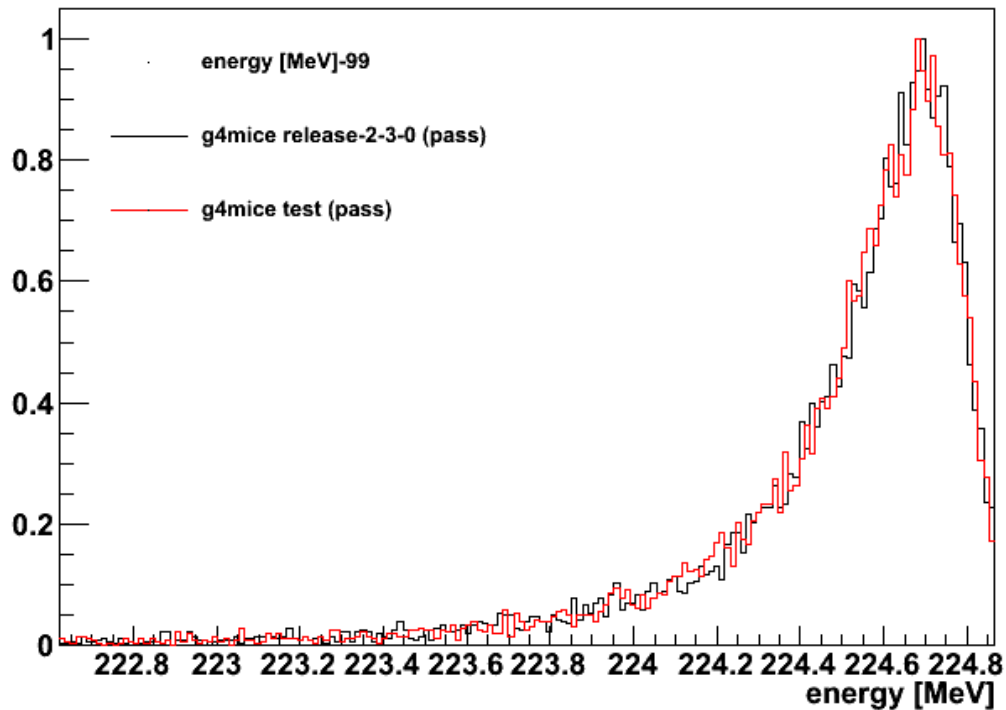


- Unit testing framework implemented and operational
  - A few developers have contributed
  - Release 2-4-1 does not include changes that have not been tested
  - Rule will apply to future releases
  - Writing unit tests will take many months
- Application level framework implemented
  - Focus on simulation but should be extensible
  - So far implementation has regression tests for physics list, quadrupole model

# Code QA and Documentation



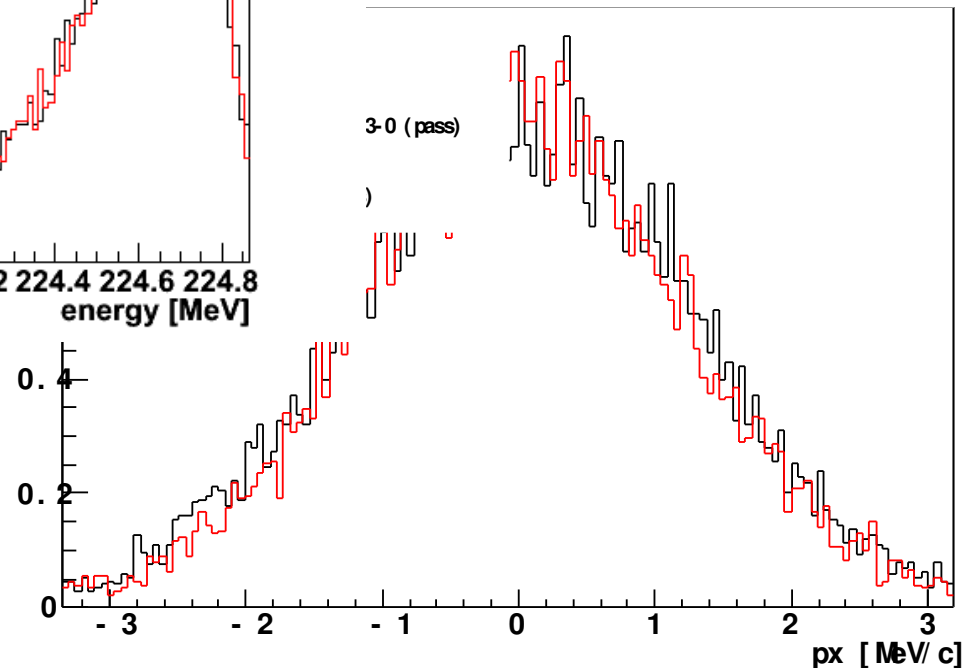
10.0 mm LITHIUM\_HYDRIDE with 10000 200.0 MeV/c mu+ 1.0 mm steps



d operational

s that have not been tested

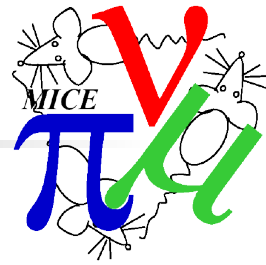
100.0 MeV/c mu+ 10.0 mm steps





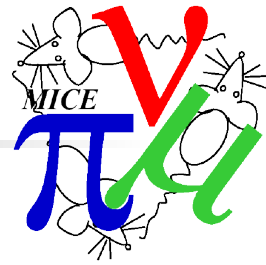
# Code QA and Documentation

---



- Code documentation being implemented
  - Doxygen style comments inline

# Code QA and Documentation



- Code documentation being implemented

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://www.mice.iit.edu/software/doxygen/html/`. The page title is "G4MICE - Mozilla Firefox". The browser's menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The address bar has a search icon and a "Google" search button. The browser's status bar at the bottom shows the system clock as "08:22" and the location as "London".

The main content area of the browser displays the "PolynomialVector Class Reference" page. The page has a navigation bar with tabs for "Main Page", "Related Pages", "Namespaces", "Classes", and "Files". Below this, there are tabs for "Alphabetical List", "Class List", "Class Hierarchy", and "Class Members". The "Class Hierarchy" tab is selected, showing an inheritance diagram for the "PolynomialVector" class. The diagram shows "PolynomialVector" inheriting from "VectorMap".

The "PolynomialVector" class is described as "an arbitrary order polynomial vector class." The page includes a "List of all members" link and a "Public Member Functions" section. The functions listed are:

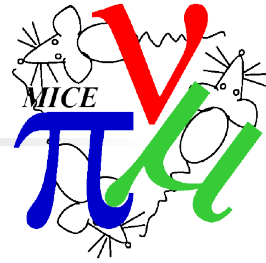
- `PolynomialVector` (int pointDimension, `MMatrix< double >` polynomialCoefficients)  
Construct polynomial vector passing polynomial coefficients as a matrix.
- `PolynomialVector` (std::vector< PolynomialCoefficient > coefficients)  
Construct polynomial vector passing polynomial coefficients as a list of PolynomialCoefficient objects.
- `~PolynomialVector` ()  
Destructor - no memory allocated so doesn't do anything.
- `void SetCoefficients` (int pointDim, `MMatrix< double >` coeff)  
Reinitialise the polynomial vector with new point (x) dimension and coefficients.
- `void SetCoefficients` (std::vector< PolynomialCoefficient > coeff)  
Reinitialise the polynomial vector with new point (x) dimension and coefficients.
- `void SetCoefficients` (`MMatrix< double >` coeff)  
Set the coefficients.
- `MMatrix< double >` `GetCoefficientsAsMatrix` () const  
Return the coefficients as a matrix of doubles.

The left sidebar of the browser window shows a list of files and folders, including "PhaseSpaceVector", "PhaseVector", "PidFits", "PMTCheck", "PmtKey", "pmtParam", "Point", "Polycone", "polycone", "polyhedra", "PolynomialInterpolator", "PolynomialVector", "PositionSizeParameter", "positionType", "PrimaryGenHit", "PrimaryGenHitTextFile", "property", "Q35", "quadrangular", "Quadrupole", "QuantityType", "QuantityVectorType", "Reader", "Recon", "ReferenceListType", "ReferenceType", "reflectedSolid", "Region", "ReplicaPlacementType", "ReplicationAlgorithmTy", and "RFBBackgroundParam".



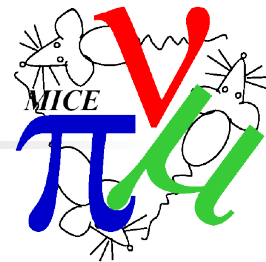
# Code QA and Documentation

---



- Code documentation being implemented
  - Doxygen style comments inline
  - Wiki + pdf for overview documents
    - Developer documentation
    - User documentation
  - Support email list
    - Use it!

# Code QA and Documentation



- Code documentation being implemented
  - Doxygen style comments inline

W **G4MICE - DeveloperOverview - MICEmine - Mozilla Firefox**

File Edit View History Bookmarks Tools Help

http://micewww.pp.rl.ac.uk/projects/g4mice/wiki/DeveloperOverview

Most Visited Latest Headlines Twitter / chris\_rutg...

Overview Activity Issues Documents Wiki Files Repository

Back to G4MICE Developer documentation

## Overview of software structure

Dependency tree of G4MICE is shown below:

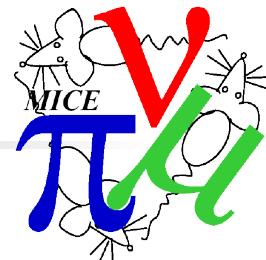
```
graph TD
    Applications --> RealData
    Applications --> Visualization
    RealData --> Unpacking
    RealData --> Persist
    Unpacking --> ZLib
    Persist --> Analysis
    Persist --> Simulation
    Analysis --> Optics
    Analysis --> BeamTools
    Analysis --> Calib
    Simulation --> EngModel
    Simulation --> DetResp
    EngModel --> Config
    DetResp --> Config
    DetResp --> Interface
    Interface --> GEANT4
    Interface --> CLHEP
    Interface --> GSL
    Visualization --> Recon
    Visualization --> DetModel
    Recon --> RecPack
    DetModel --> RecPack
```

Summary of deliverables

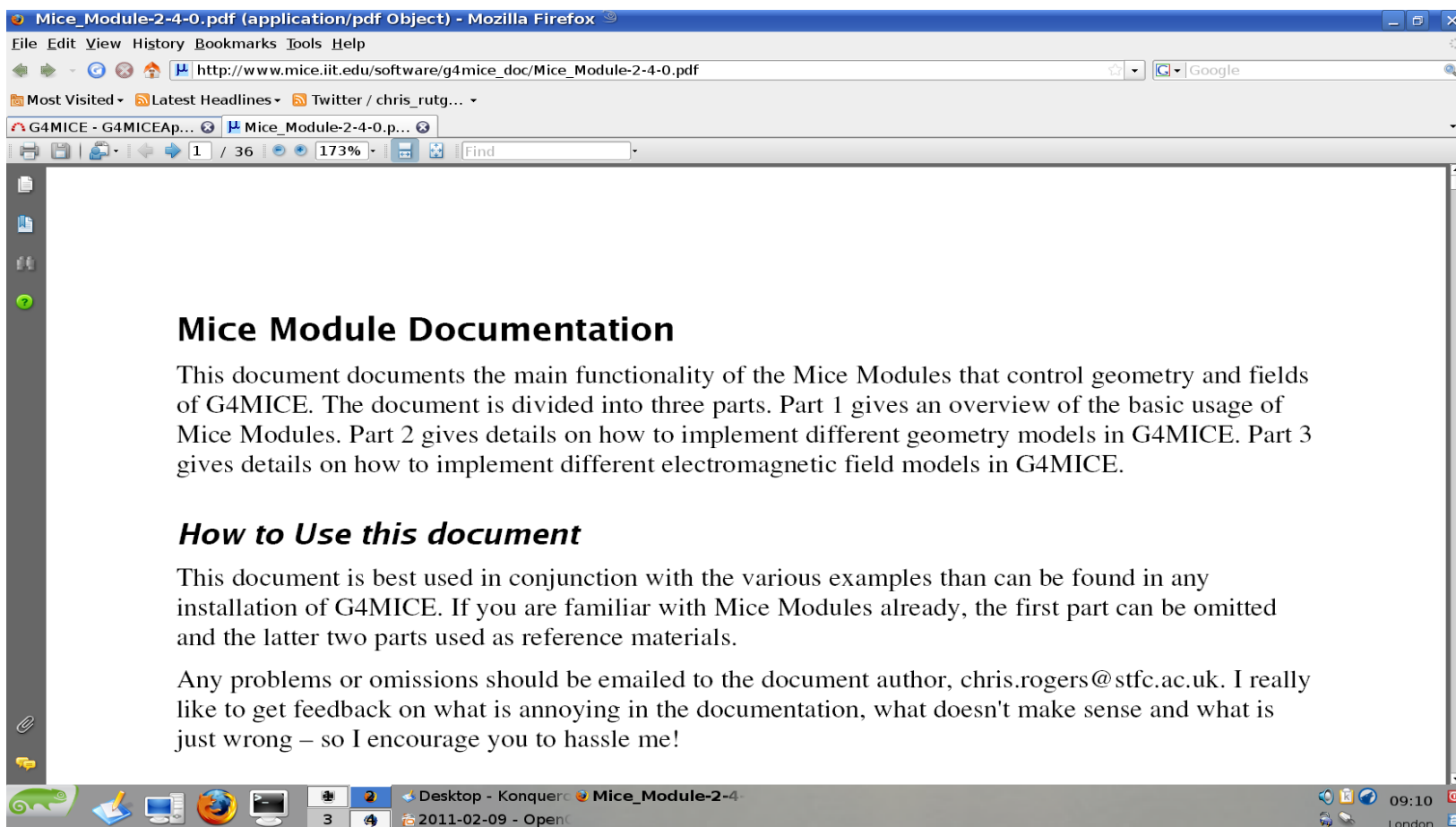
- User Documentation
- QA Process
- Tracking simulation of MICE
  - At arbitrary level of detail
  - Model fields and detectors
  - NOT implementation of geometry
- Electronics model of MICE detectors
- Space point reconstruction
- Track reconstruction
  - Track fitting
  - Particle identification
  - Capable of working with any subset of detectors
  - Online and offline
- Physics analysis tools
  - Extraction of accelerator physics parameters of interest from beam data (emittance, Twiss parameters, probably some more stuff)
  - Statistical manipulations (weighting, cuts, particle selection)
  - Transport mapping



# Code QA and Documentation



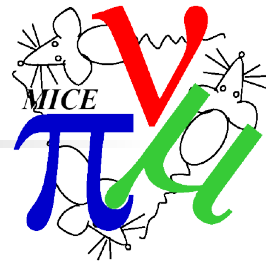
- Code documentation being implemented
  - Doxygen style comments inline





# Continuous Integration

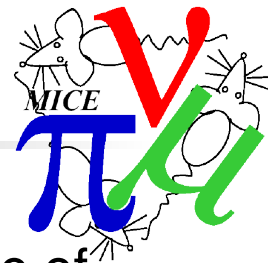
---



- Continuous Integration
  - When we check in the code we do some rudimentary tests on coding standards, testing
  - We have regular automated scripts to run nightly build and test jobs
  - Aspire to daily (or so) email to developers if the test job is broken
- If the code breaks, we know what broke and where



# Geometry + ConfigDB Interface



- It is not G4MICE crew's job to figure out what MICE is made of
  - We can't end up with the job of writing TRD - too much
  - DECISION: we will implement engineering CAD model
- We will implement the MICE CAD model
- Some subtleties
  - Manage issues like diffuser, proton absorber
  - Add tags for detector guys (e.g. channel + station number)
  - Add field map tags (e.g. field type)
- Workflow: CAD -> GDML <-> MiceModule -> GEANT4
  - Allows to add coil configurations
- Workflow: CAD -> GDML <-> ConfigDB
- Status - preliminary implementation complete
  - Further work requires license for CAD -> GDML converter code

# Geometry + ConfigDB Interface

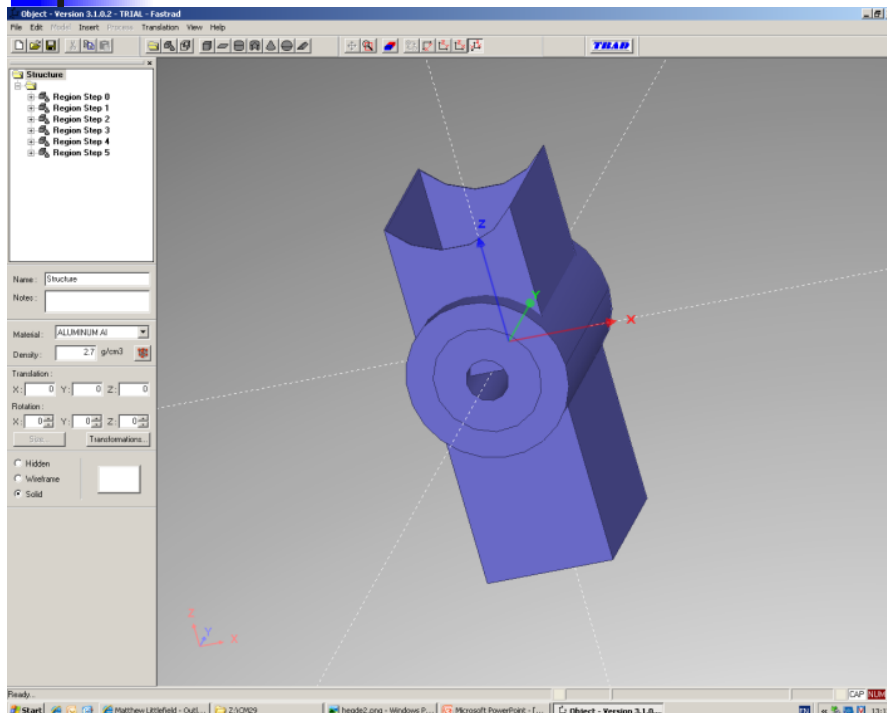
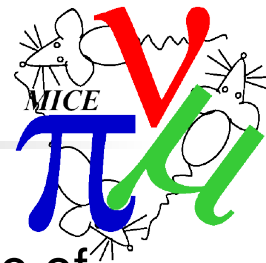
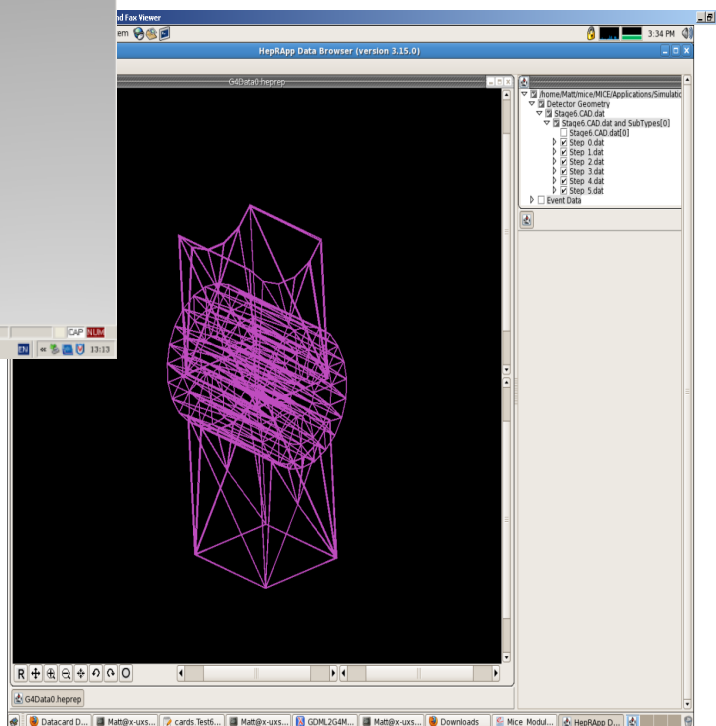
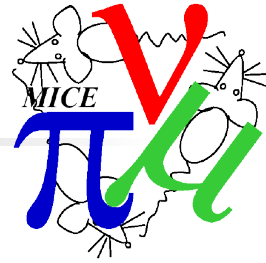


figure out what MICE is made of  
writing TRD - too much  
engineering CAD model  
AD model

- Allows to add coil config
- Workflow: CAD -> GDML
- Status - preliminary imple
  - Further work requires lic



ode

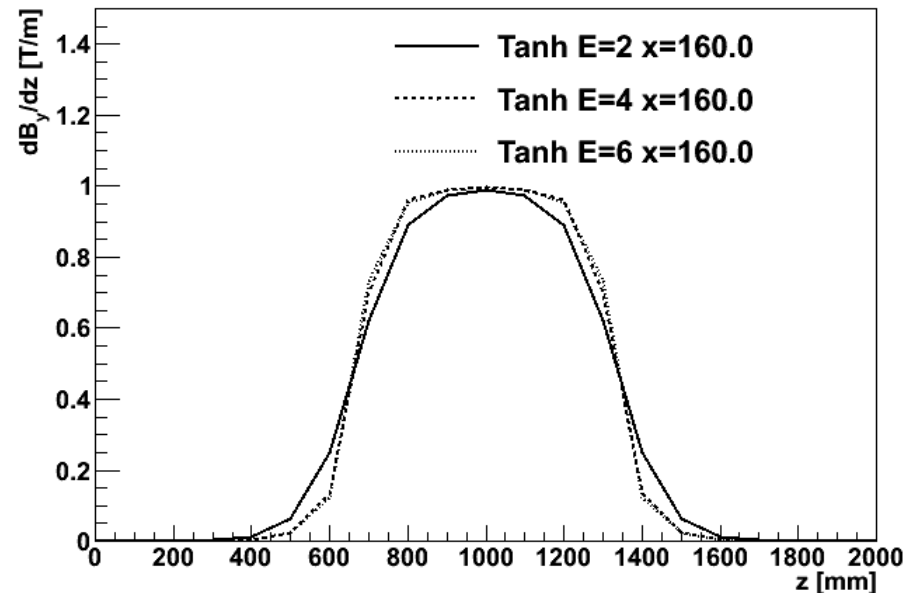
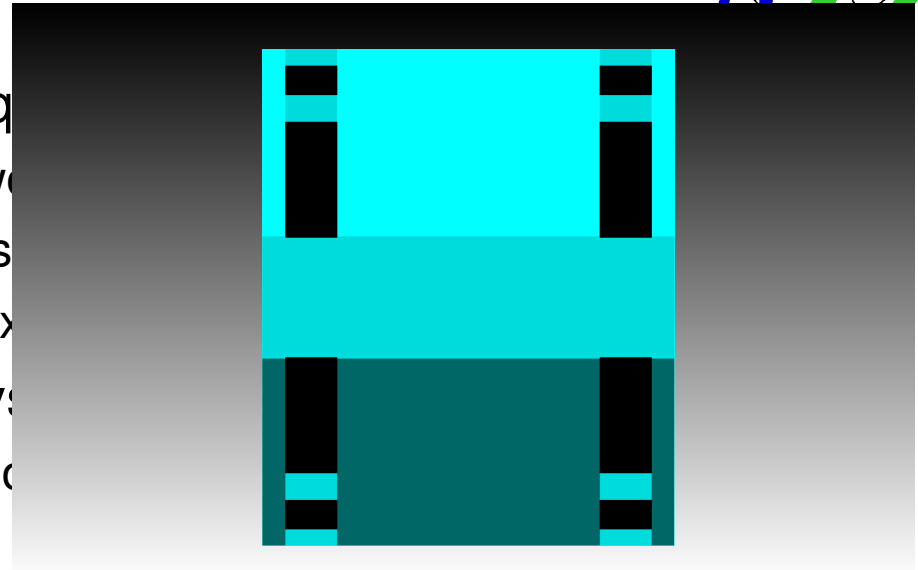


- Bug fix to multipole model for quadrupoles
  - Required some significant reworking of the code
  - Proper treatment for end fields
  - All new code is unit tested, doxygenated, application tested
- Added regression tests for physics model
  - Check that physics process model does not change from one version to the next
  - Test for most relevant materials, particle species, particle momenta

# Simulation



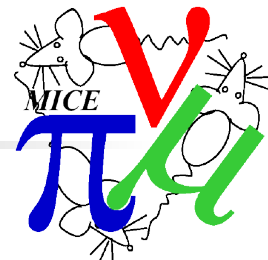
- Bug fix to multipole model for q
  - Required some significant rework
  - Proper treatment for end fields
  - All new code is unit tested, dox
- Added regression tests for physics
  - Check that physics process model version to the next
  - Test for most relevant materials





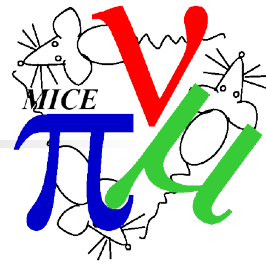
# Detectors

---



- EMR work is progressing
  - Simulation is advanced
  - Some work on cluster finding
  - Not committed to CVS
  - No manpower for testing
- TOF progress in documentation
  - Code is being commented up
  - Testing is planned
- Tracker is work in progress
  - Focus is on unpacking (out of scope)
  - But getting reconstruction code to work seems non-trivial
- Need to integrate with MAUS data structures
- Need to start work on global interfaces

# Physics analysis tools



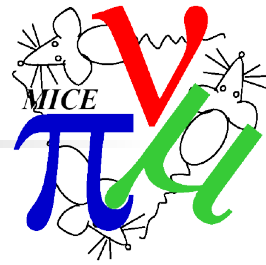
- MICE analysis software requirements
  - Physics analysis routines for MICE control room
    - Beta, alpha, emittance, plotting
    - Need fast turnaround of plots
- Current physics analysis routines are not fit for purpose
  - Written by some rubbish graduate student as a first coding project
- Instead use XBOA physics analysis library
  - ~ few thousand lines of code
  - Good test coverage
  - Well documented (function-by-function + worked examples)
  - Local expertise
  - Written by some rubbish RAL staff guy
- Allows for quick and dirty physics analysis online
  - People need to get acquainted off line to use it (quickly) online
  - Avoid long turnaround to get slight changes to plots, etc





# Operations

---

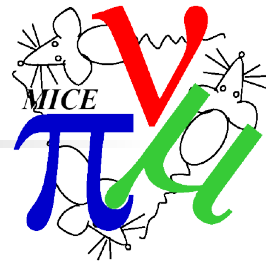


- We will make an on-call rota during operational periods
- Some discussion about how we manage robust approach to bug fixing in control room
  - We find a bug and fix it...
  - Did we really fix it?
  - Did we make a new bug?
  - Are we keeping the control room code and offline code in sync?
- On the other hand, we don't want to compromise running with procedural crud

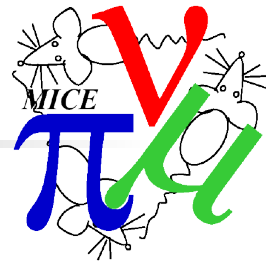


# Manpower

---



- Almost total turnover of manpower in this year
- Welcome to new folks joining the team!
- We have our work cut out for us
  - Documentation is poor
  - Testing is poor
  - How can I change the code if I don't know what it's supposed to do?
    - Was that a bug?
    - Was that supposed to happen?
- We should anticipate **now** another turnover cycle in 3 years time
  - Documentation
  - Testing



- Hope for
  - Config DB Interface - March 1
  - Simulated beam timing within spill structure - March 18
  - CAD model first implementation - April 8
  - Existing functionality replicated in MAUS framework - April 15
  - 3D meshing routines (but no interpolation) - May 1
  - Optics refactor - August 1
  - EMR
  - Ckov
  - Tracker refactor