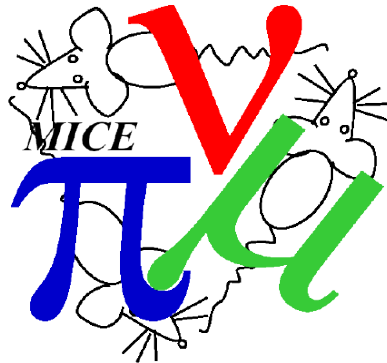




# MAUS Status and Plans

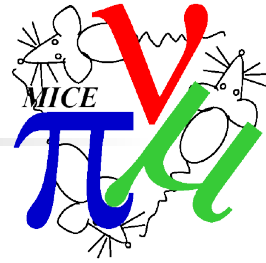
---



Chris Rogers,  
ASTeC,  
Rutherford Appleton Laboratory

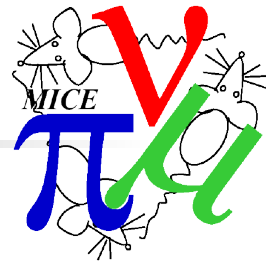


# MAUS



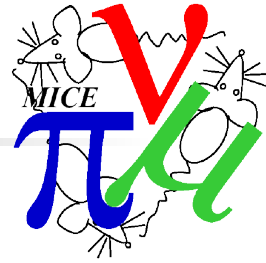
- G4MICE -> MAUS
- Manpower turnover
- Infrastructure status
- Code QA
- Beamline Paper
- Accelerator side code
- 122 closed issues since MICE CM 29
- 86 currently open issues

# Manpower turnover



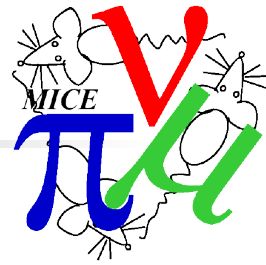
- Manpower turnover in progress
  - Malcolm Ellis long gone
  - Mariyan on paternity leave?
  - Vassil, David (Adey), Mark, writing up
  - Morteza, Edward, Ruslan, Gene, Stefania coming on board
  - Chris (Tunnell) well integrated in the team
  - Yordan, Chris (Rogers) carry code over
  - Supported by Summer and Rob
- Welcome new members!

# G4MICE -> MAUS



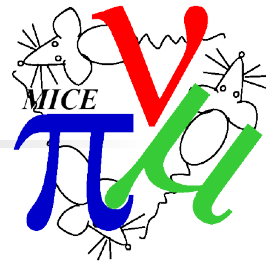
- Reimplemented software framework
  - Changed data structure
  - Developed common code to drive executables
    - Multithreading/multiprocessing is possible
  - Nightly build/test cycle
  - C++/Python both available
  - **More robust** and easier (**cheaper**) to develop code
  - **Fast release cycle** makes debugging in control room possible
- All developers are now using MAUS
  - Except legacy tracker code required for cosmics
  - Except Step I paper code
- Not all functionality has been moved
  - Old data structure was evil and deeply embedded
  - Expect to recover TOF code by October
  - Impact on operations

# Infrastructure



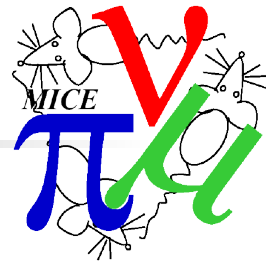
- Schema Validator
  - Commonly agreed data structure
  - Soft coded
  - Validator checks that we are using the agreed data structure
- Error handler
  - Common error handling code
  - Reduces code duplication
  - Makes fancy stuff possible
    - Dynamically change error handling at runtime
- Automated checks for code style
  - Dangerous C++ and Python language features are banned
- Defined interface for calling interfacing core C++/Python functions
- Configuration Database interface (Antony Wilson)
  - Call configuration database from MAUS using python

# CAD Import



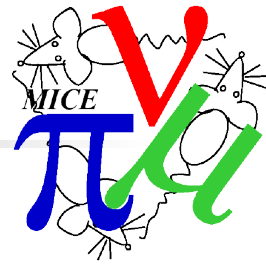
- CAD Import partially implemented (Matt Littlefield)
  - Scriptable interface to engineering CAD model
    - CAD model is authority for MICE geometry
    - Scales well with flexible experimental schedule
    - Less error prone
    - Probably slower to execute - not quantified
  - Conversion from CAD -> GDML available
  - Conversion to MiceModules imminent
  - Need to add field map, detector tags
    - e.g. flags to tell MAUS that detector volume should make hits
  - Need some cross-checks (Stefania Ricciardi)

# Infrastructure (ToDo)



- Aims for next collaboration meeting
  - Online data quality + reconstruction (Chris Rogers)
    - Ping data structure and make histograms
  - MAUSProcess, MAUSRun? (Chris Rogers)
    - Handle initialisation and destruction of “global” objects
    - Field maps, geometries, error handlers, logging ...
  - Application testing framework (Chris Rogers)
    - Partial implementation in G4MICE
    - Needs port to MAUS
  - Documentation streamlining (Chris Rogers)
    - Too many formats, too many documents
  - CAD Import closed out (Matt Littlefield)

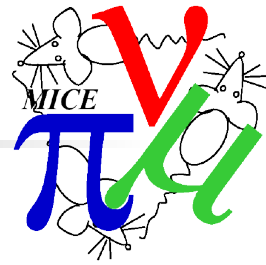
# Code QA



- Defined QA process
  - Every time we write a function, we make a comment saying
    - What the function does
    - What parameters it takes (if any)
    - What the function returns (unless it's trivial)
  - For each function we write, we also make an equivalent test
    - Check that the code does what we thought
    - Check for typos, logic errors
      - In software, small typo can make big problems
      - Many small typos are difficult/expensive to fix
    - Regressive - the next person to edit the code can re-use old tests
  - For each module we write, we also make a set of tests
    - Check that code units talk to each other properly
    - Physics checks
    - Effects from running many tracks/spills/runs
    - etc
  - Check code for banned C++/Python language features, etc
  - Code review - Go/No Go for release of code to trunk

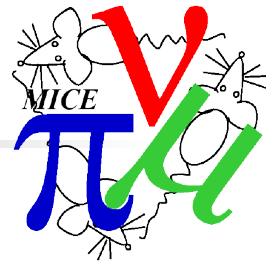


# Consequences



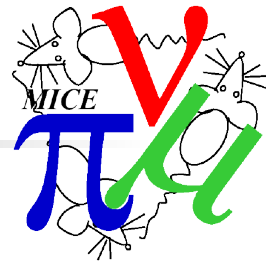
- Poor code quality
  - Staff turnover **expensive** and incurs **schedule risk**
  - High **training overhead**
    - How do I do this? How can I make the code install?
  - High **operations overhead**
    - Code crashes and running time is lost
  - Ultimately, **high development cost**
    - I changed this old piece of code and now it's broken!
    - I wanted this old feature but it doesn't work any more - what happened?

## Code QA (2)



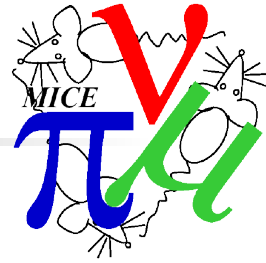
- Code which does not obey these requirements is not eligible for publication
- Code which does not obey these requirements is not eligible for control room running
- 
- If I cannot control what goes in publication, I cannot do my job
- E.g. you wouldn't ask detector managers to implement detector without having control over the hardware involved

# Step I Emittance Paper



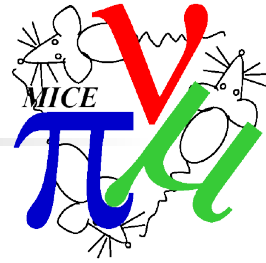
- Code used for emittance paper does not pass these requirements on a number of counts
  - Code developed prior to implementation
  - Not a criticism of the code - but we need some cross-checks
- Proposed way forward
  - Freeze code when first draft paper is finished (end July?)
  - Code review (Rogers?, Tunnell?)
  - Implement few cross-checks (Rogers? Rayner?)
  - Remake all plots
  - Expect not to find bugs
    - But we need to look

# Release Cycle



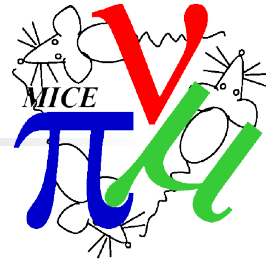
- New framework enables rapid release cycle
  - Release planned every two weeks
  - Patch releases are possible if required (e.g. bug in control room)
  - Missed a release due to CM
  - Missed a release due to webserver crash
- Web server failure issue
  - micewww web server failed due to hard drive problem ~ mid June
  - Took 2 weeks to recover redmine
  - Some things still not recovered
  - 24 hours downtime is acceptable - 2 weeks downtime is not and delays work
  - Believe the robustness issue has been resolved

# Accelerator Code



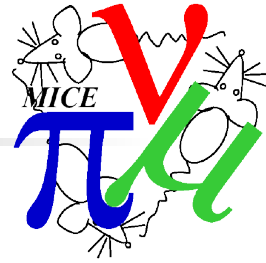
- Clean up of simulation module
  - Reimplement with improved testing
  - Fix/improve structure to reflect more closely MICE running
  - Reimplement beam generation with view to modelling spill structure (Tim Carlisle)
  - Reimplement RF automatic phasing routines with new algorithm
  - Big piece of work to do this
- Still missing
  - Interface to Geant4 visualisation routines (Chris Rogers)
  - Energy loss and dE/dx tests (Chris Rogers)
  - Some routines still need to be cleaned up (Chris Rogers)

# Aims for Next CM



- Interface MAUS datastructure to physics analysis routines (Chris Rogers)
- Physics tests (Chris Rogers)
  - Fields
  - dE/dx tests
- Partial clean up of beam envelope pusher (Victoria Blackmore/Tim Carlisle/Chris Rogers)
- Field mapping? (Victoria Blackmore)

# Accelerator ToDo



- Interface MAUS datastructure to physics analysis routines (Chris Rogers)
- Physics tests (Chris Rogers)
  - Fields
  - dE/dx tests
- Partial clean up of beam envelope pusher (Victoria Blackmore/Tim Carlisle/Chris Rogers)
- Field mapping? (Victoria Blackmore)