DMT Implementation Plan

- Data distribution infrastructure mostly working
  - Shared memory buffer manager & utilities are working.
  - NDS interface / Data Pusher now working with frames!
  - Prototype application data interface.
- Need well defined sand box framework.
- Use FrameCPP, Algorithm Library (TBD).
- Defer trigger interface except operator messages.
- Have a basic system for Detector Studies and Monitor Development ASAP (April)

Sand Box Interface

A "Sand Box" interface will provide the functions needed to gather, manipulate and analyze LIGO data.

Data Access:

- Synchronous access of multiple channels
- Arbitrary stride length.
- Produce meaningful objects (time series).

Meaningful Data Types:

- Time series
- Frequency series
- Time stamps and time intervals.

Algorithms:

- Operations on data (Filters, FFTs, etc.)
- Data consistency checks
- Error calculation/Propagation

Utilities:

- Error reporting
- Data Conversion
- etc.

```
//-------------------------  Process one frame (FrameCPP).
void
BitTest::ProcessFrame(FrameCPP::Frame *frame)
{
    //---------------------  Check that the Frame has data
    if (!frame->containsRawData()) {
        cerr << "No raw data in frame." << endl;
        return;
    }

    //---------------------  Loop over channels
    const FrameCPP::AdcData* adc;
    for (channel_iter c=mChannel.begin() ; c!=mChannel.end() ; c++) {

        //-----------------  Find the data in the frame!
        const char* ChName = c->getChannel();

        adc = &(frame->getRawData().findAdc(ChName));
        if (!adc)  continue;

        const FrameCPP::Vect* vect = &(adc->refData().front());
        if (!vect) continue;

        uint_t nw = vect->getNBytes()/2;
        if (nw <= 0) return;

        const short* data = (short*) vect->getData();
        if (!data) continue;

        //-----------------  Scan for stuck bits
        c->Scan(nw, data);
    }
}
```

Sand Box Interface

A "Sand Box" interface will provide the functions needed to gather, manipulate and analyze LIGO data.

Data Access:

- Synchronous access of multiple channels
- Arbitrary stride length.
- Produce meaningful objects (time series).

Meaningful Data Types:

- Time series
- Frequency series
- Time stamps and time intervals.

Algorithms:

- Operations on data (Filters, FFTs, etc.)
- Data consistency checks
- Error calculation/Propagation

Utilities:

- Error reporting
- Data Conversion
- etc.

DMT Implementation Plan

- Data distribution infrastructure mostly working
  - Shared memory buffer manager & utilities are working.
  - NDS interface / Data Pusher now working with frames!
  - Prototype application data interface.
- Need well defined sand box framework.
- Use FrameCPP, Algorithm Library (TBD).
- Defer trigger interface except operator messages.
- Have a basic system for Detector Studies and Monitor Development ASAP (April)