# Complete rewrites
# When, why, and how?

James W. Pflugrath[1]
*Rigaku/MSC, Inc., The Woodlands, Texas, USA*



## Outline

- Previous software packages
- Current software packages
- When & Why
  - Whenever you get a benefit and the money to do it
  - Hardware requirements
  - Software requirements
- How
  - Modern software engineering practices in scientific programming
- Example
- Future software packages
- Bottom line
  - Whenever you get a benefit
  - Code maintenance



## Previous software packages

- FRODO, Alwyn Jones, 1970's
  - Evans & Sutherland PS300 version 1983



## Previous software packages

- MADNES, w/ A. Messerschmidt, 1984
  - FORTRAN77, structured (no GOTOs)
  - VAX/VMS, IRIX, Sun4, Linux, OSF1
  - Device-independent
    - Enraf-Nonius FAST, ADSC multiwire, Xentronics
  - EEC-workshops, 1980's
    - Vectorial description & algorithms, David Thomas
  - Department of Energy, Beamline X8C, E. Westbrook early 1990s



## Current software packages

- JWP moves to Molecular Structure Corporation, 1994
- d*TREK: device-independent diffraction image processing
  - DOE subcontract, 1994, E. Westbrook
    - Simple re-write or adaptation not possible
- C++
  - Object-oriented programming language
  - No standard template library



## When & why?

- 1994
- Anytime! (2005)
- Whenever a benefit or advantage arises from the re-write
  - This is always the case, you would not make a worse piece of software would you?

## When and why?

- New programming tools
  - New languages and libraries
  - OpenGL, X Windows, OSF/Motif, Tcl/Tk
  - C++, Python
- New features
- New hardware
- New people
  - What skills do they have?
- Maintenance issues
- User issues
- Legal issues
- Who pays the bills?



## How?

- Write a grant
- Start a company or go to work for a company
- Start a consortium
- Make your users pay
- In other words … sell it and get money



## How?

- Build infrastructure
- Get computers
- Get software tools
- Get people
- Read books
- Get help



## Software engineering practices

- Nuts & bolts
- Design beforehand
- User requirements
- Hardware requirements
- Data structures
- Algorithms
- Code management, version management
  - make, SourceSafe, cvs, bugzilla, backups
- Book: *Code Complete*
  - In the trenches: How-to
  - Variable naming, Hungarian notation



## Example: d*TREK

- Design submitted to DOE in late 1994
  - Data objects
    - Devices
    - Source, Shutter, Goniometer, Detector, Crystal
    - Images, Reflns, Headers
    - Interprocess communication
  - Methods
    - Single objs: Goniometer move, Image write, etc.
    - Multiple objs: Find, Index, Predict, Refine, Integrate, Scale/Average
    - Reflnlist: merging, editing, sorting



## Devices and Objects

1912

2005

## Example: d*TREK

- User interface
  - Simple: command line arguments
  - Scripts
  - Graphical user interface helps build command lines
    - 1994: X-Windows/Motif



## Scripting

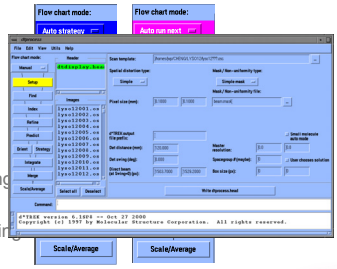
- User defaults
- High throughput
- No need for GUI
  - no button processing
- Customization
  - Beamline
  - Detector
  - Crystal

```
#!/bin/csh -f

set IMAGE_NAME  = ../lyso12001.osc
set FIRST_IMAGE = 1
set LAST_IMAGE  = 99

dtextractheader $IMAGE_NAME 1.head
dtfind 1.head -seq $FIRST_IMAGE $FIRST_IMAGE -out
dtindex dtfind.head dtfind.ref
dtrefine dtindex.head dtfind.ref +All -go -go
dtrefine dtrefine.head -seq $FIRST_IMAGE +All -go
dtrefine dtrefine.head -seq $FIRST_IMAGE +All -go
dtintegrate dtrefine.head -seq $FIRST_IMAGE $LAST_
            -profit -window 0 0 -batch 1 4
dtscaleaverage dtintegrate.head dtprofit.ref -sigm
        -errormodel -reject .0075 \
        -batchscale \
        -reqab spherical 4 4 \
        dtscale.ref
```



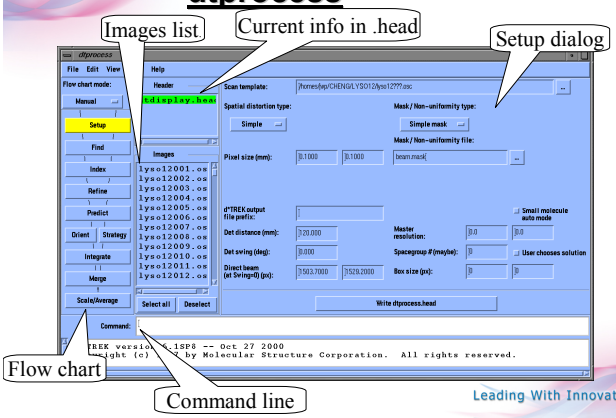
## dtprocess

- GUI to control subprocesses
- Master scripter
- Flow chart
  - Manual
  - Auto strategy
    - one button screening
  - Auto processing
    - one button processing



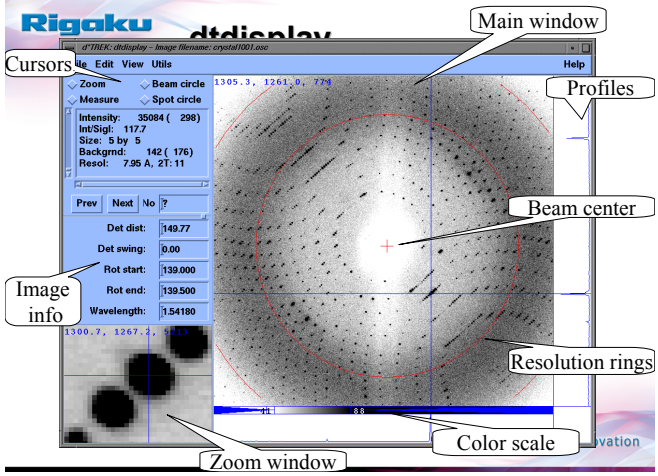


## dtprocess






## dtdisplay






## Classes and objects of the d*TREK toolkit

- Class Cimage
  - Data and methods for a 2D diffraction image
  - Several constructors
  - nRead(), nWrite(), fGetPixel(), nSetPixel(), nGetRect(), etc.
- Class Cdetector
  - Cspatial, Cnonunf, Cgoniometer
  - Cspatial::nPxToMM(), ::nMMToPX()
- Class CrefIn
  - nGetHKL(), nGetH(), nGetK(), nGetL(), …
- …

## Classes and objects of the d*TREK toolkit

- Class CreflnlistConstructor CreflnlistConstructor Creflnlist()
  - Constructor Creflnlist() *(like Ralf's __init__)*
  - ::nRead()
  - ::nReduce() (needs Ccrystal object)
  - ::nSort()
  - ::nInsert(), nDelete(), nSelect()
  - ::nWrite()



## Hungarian notation

- Used at Microsoft
- Invented by a Hungarian employee of MSFT
- Examples as used in d*TREK:

```
int      nH;
Int    *pnH;
double dH;
int      anHKL[3];
Int      a3x3dMatrix[3][3];
Ccrystal *poCrystal;
Cspacegroup *m_poSpacegroup;
```



## Hungarian notation

```
poRefln     = poGetRefln(nRefNext++);
nCentPhase = oSpacegroup.nReduceHKL(poRefln,
                                    a3nReducedHKL,
                                    &nFplusminus);
nPackedHKL = poRefln->nPackHKL(a3nReducedHKL);
```

This is controversial. But don't forget what Ralf showed us:

```
template
```

can be used with float, double, int, unsigned short int, etc, so use a different



## Future software

- Current problems
  - Code maintenance in multi-platform environment
  - Lots of Windows users
    - CrystalClear – MFC-based (native Windows GUI)
    - Team of programmers know MFC
  - Lots of Linux users (X Windows is native to Linux)
    - One person knows OSF/Motif
  - Lots of Mac/OSX users
  - Installation problems
  - Users know less than before
- Solution
  - Java?
  - Python?
  - wxWidgets?



## Bottom line: When, why, and how?

- There is no such thing as free software
  - At a minimum no one in this room works for free
  - "You get what you pay for." – *Harry Powell August 2005*
- Whenever there is a clear benefit
  - New hardware, operating systems
  - New users
  - New programmers
  - New methods
- …



## Bottom line: When, why, and how?

- Whenever you can get money to do it
  - Consortiums
  - Beamlines
  - Charge for-profit companies
- How:
  - Get serious about software engineering practices
  - Read books
  - Take classes
  - Hire staff – give them a stake in it

## One last thing …

"Remember, software is just like paper: It's the result of research."

--- *Wladek Minor, May 29, 2005*

"Software is just like toilet paper: Users want to use the softest available, then throw it away."

--- *Jim Pflugrath, June 1, 2005*



## Acknowledgements


- Rigaku/MSC
  - Thad Niemeyer
  - Robert Bolotovsky
  - Cheng Yang
  - Kris Tesh
  - Tom Hendrixson
  - Joe Ferrara
- Ed Westbrook
- R. Jacobson
- US Dept of Energy
  - Contract 943072401
- Gerard Bricogne
  - EEC Workshops
- Clemens Vonrhein


Leading With Innovation