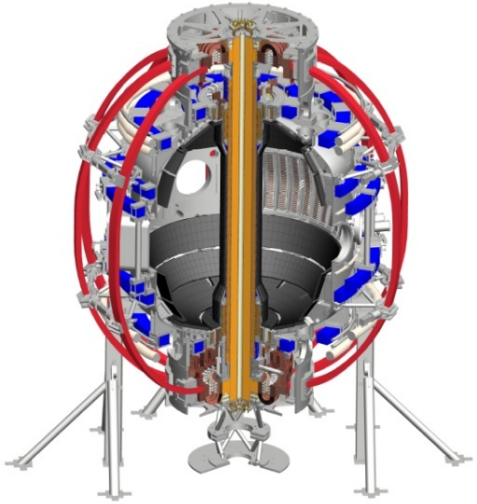
Results from the Data Resources Survey and a Software Framework for Efficient Data Usage

Coll of Wm & Mary
Columbia U
CompX
General Atomics
FIU
INL
Johns Hopkins U
LANL
LLNL
Lodestar
MIT
Lehigh U
Nova Photonics
ORNL
PPPL
Princeton U
Purdue U
SNL
Think Tank, Inc.
UC Davis
UC Irvine
UCLA
UCSD
U Colorado
U Illinois
U Maryland
U Rochester
U Tennessee
U Tulsa
U Washington
U Wisconsin
X Science LLC

D. Smith, K. Tritz, H. Yuh

6/29/2015



Culham Sci Ctr
York U
Chubu U
Fukui U
Hiroshima U
Hyogo U
Kyoto U
Kyushu U
Kyushu Tokai U
NIFS
Niigata U
U Tokyo
JAEA
Inst for Nucl Res, Kiev
Ioffe Inst
TRINITI
Chonbuk Natl U
NFRI
KAIST
POSTECH
Seoul Natl U
ASIPP
CIEMAT
FOM Inst DIFFER
ENEA, Frascati
CEA, Cadarache
IPP, Jülich
IPP, Garching
ASCR, Czech Rep

Survey motivation

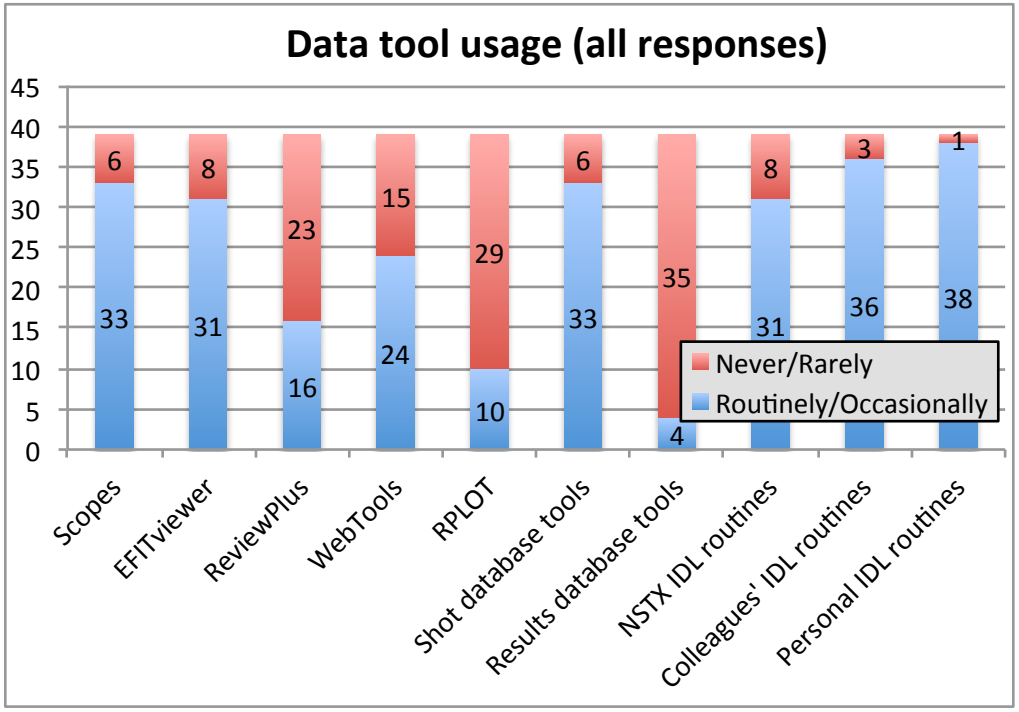
- Goals
  - Increase scientific productivity of NSTX/NSTX-U data
  - Reduce barriers to entry for new team members
  - Reduce burden for data publication requirements

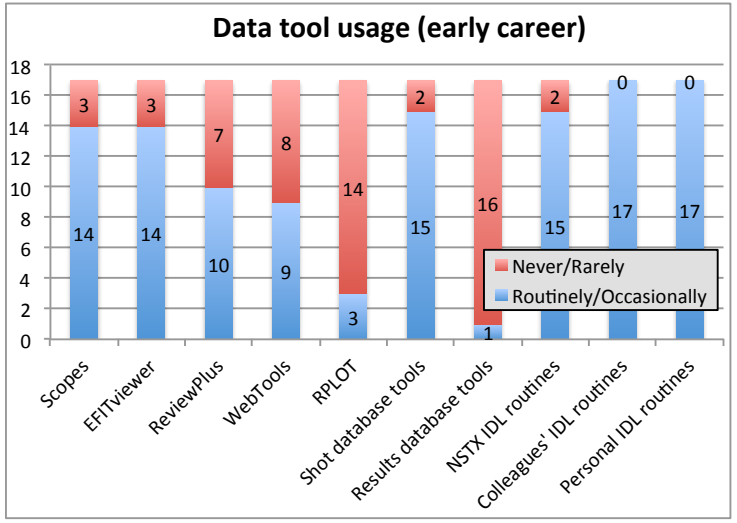
- Objectives
  - Identify inefficiencies in our data ecosystem
  - Develop strategies for improvement
    - Software framework for data access and management

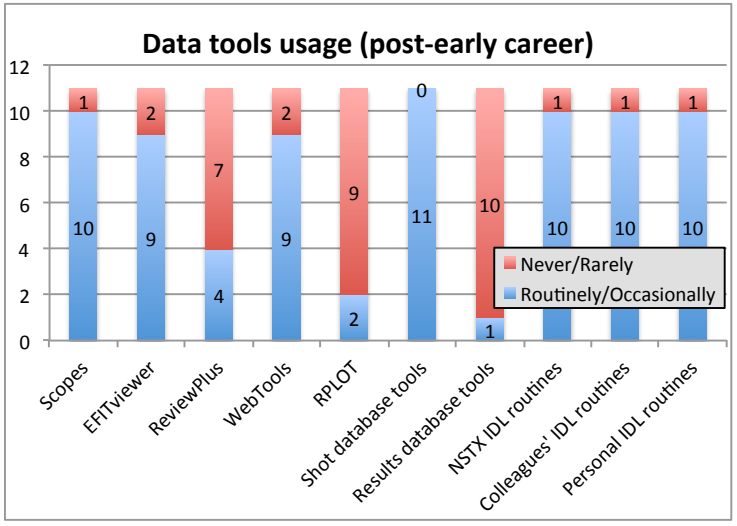
39 responses total – thanks!

- 17 early career
- 11 post-early career
- 11 unknown (9 anonymous)

Popularity of data tools

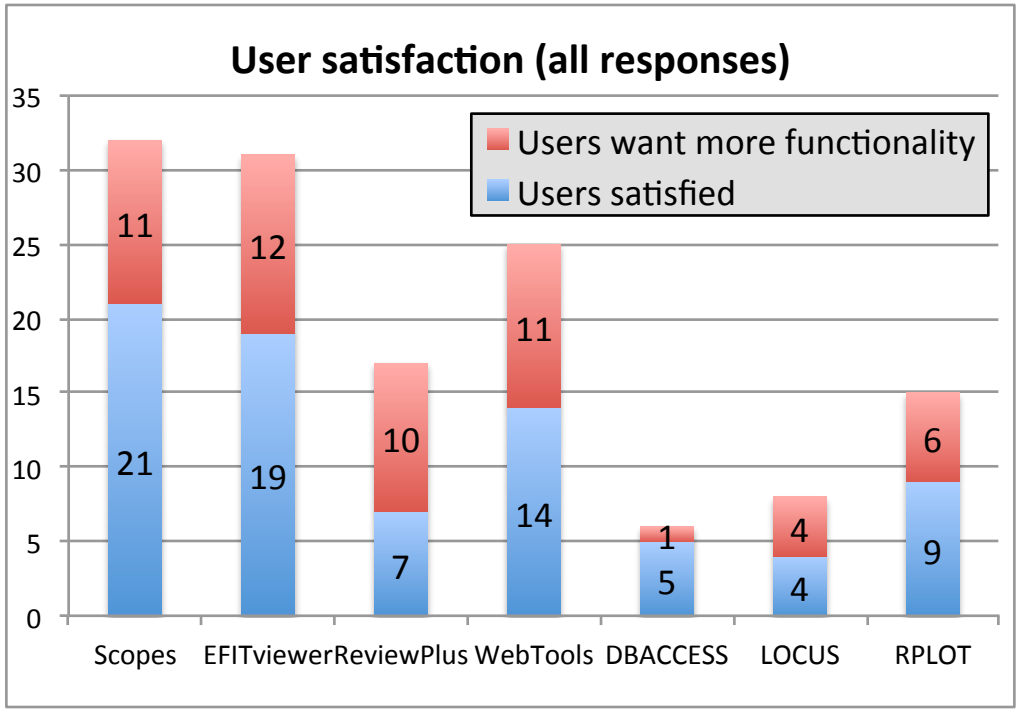


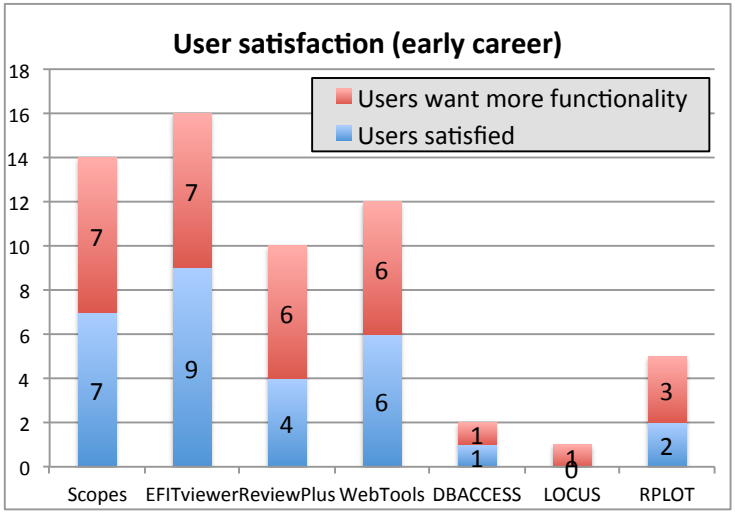


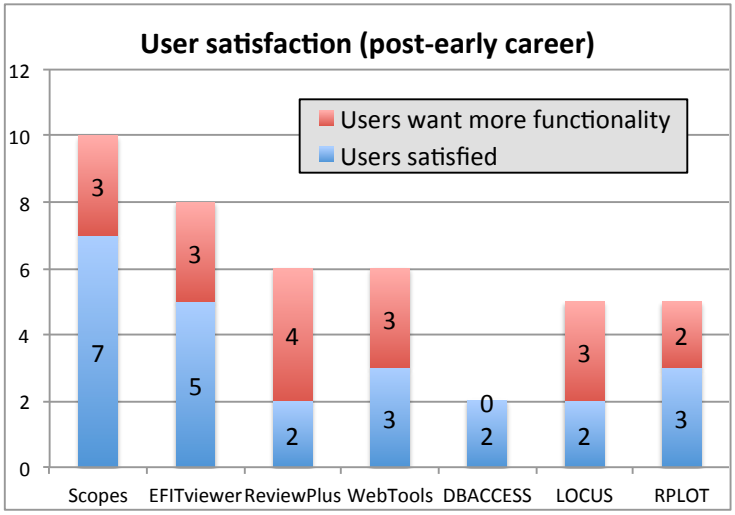


- Most popular: scopes, EFITviewer, IDL, shot db
- Next: WebTools and ReviewPlus
- Least: RPLOT, results db tools
- Early career responses
  - more ReviewPlus, less WebTools

User satisfaction with functionality in data tools

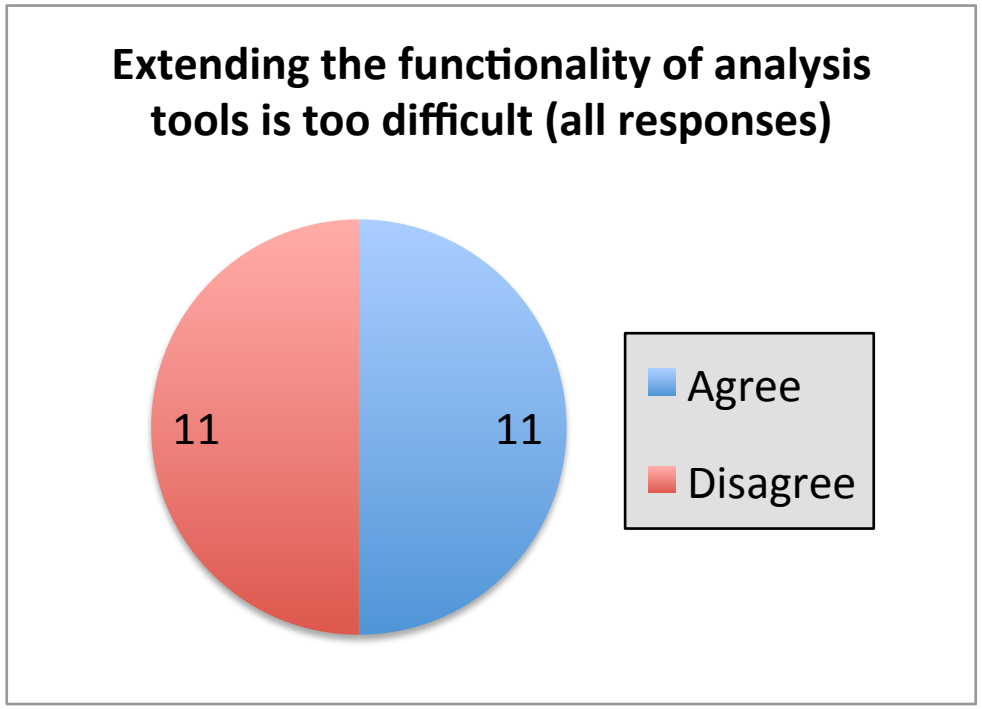


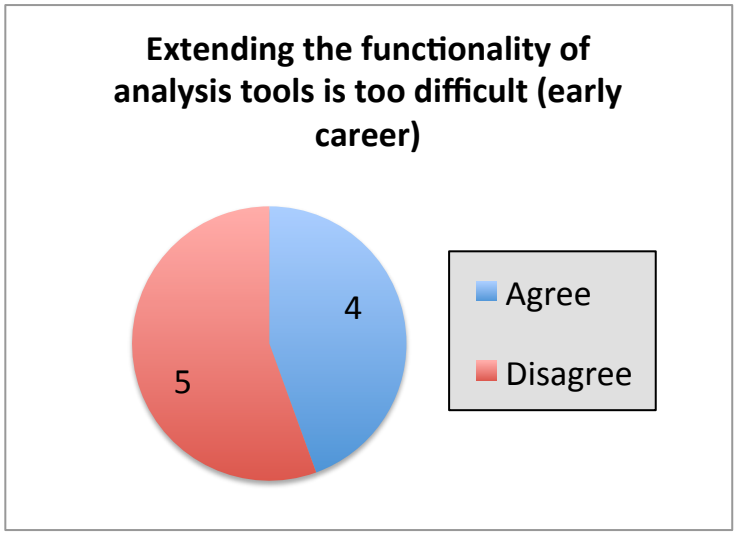


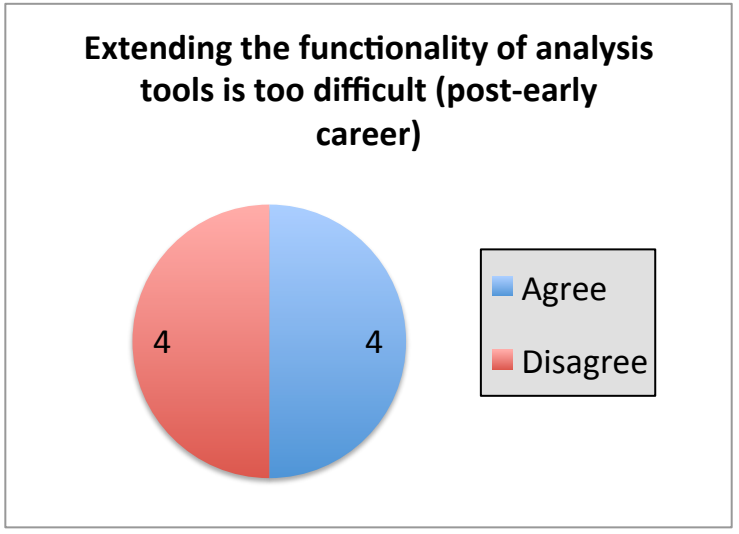


- Satisfaction levels typically at 50%-66%
- Early career responses
  - Satisfaction typically lower, around 50%

Q: Extending the functionality of tools is too difficult

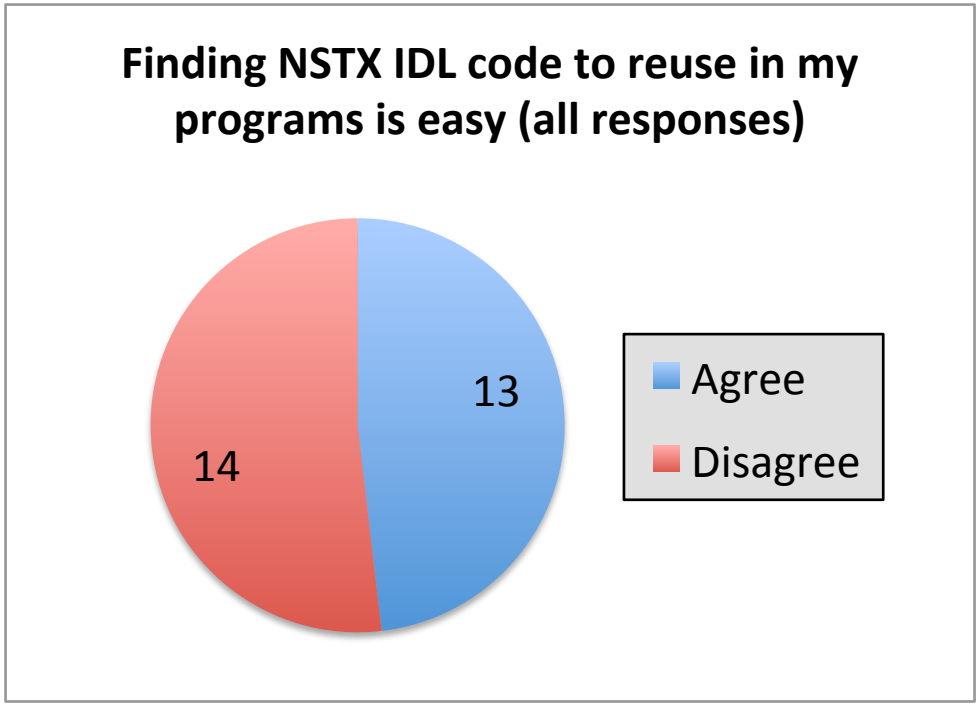


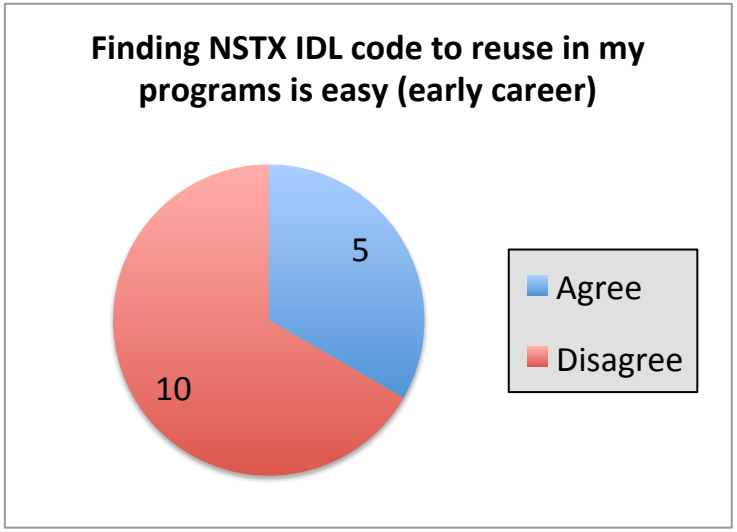


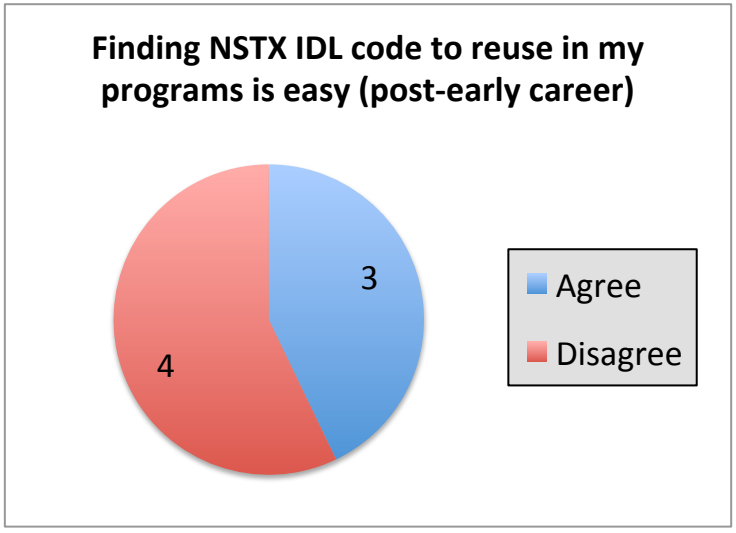


- 50% feel extending functionality of tools is difficult

Q: Finding IDL code to reuse is easy

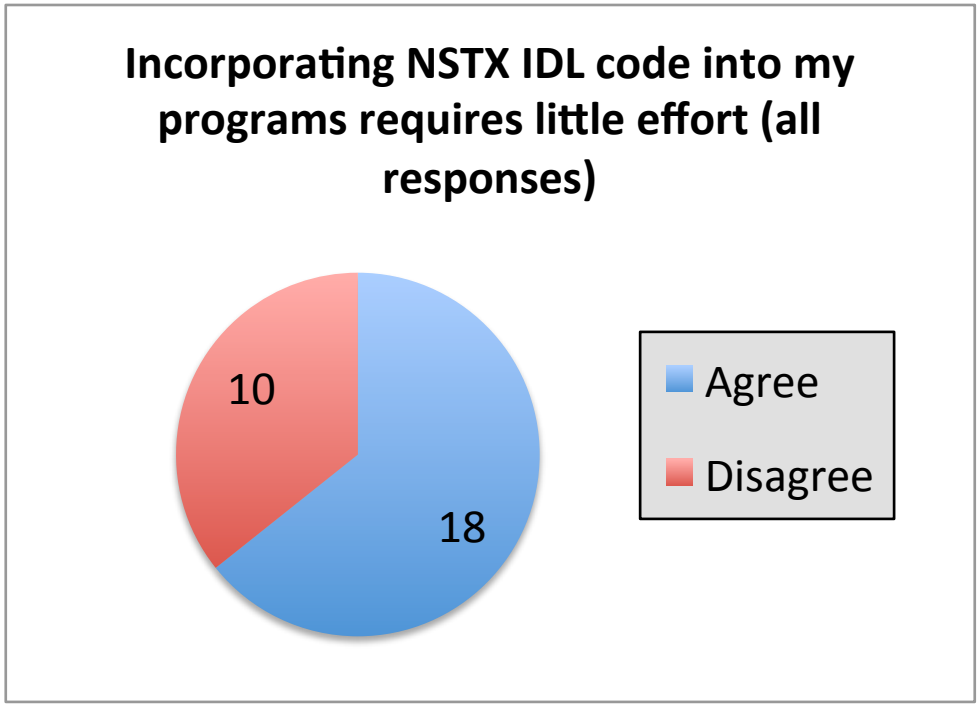


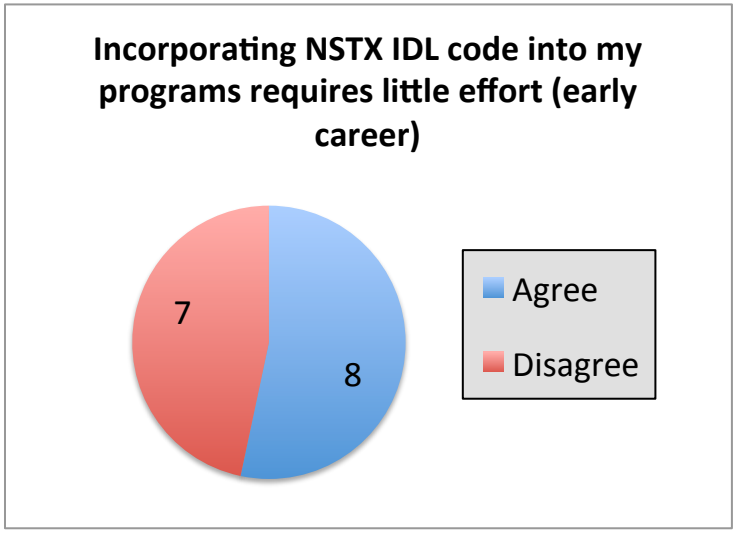


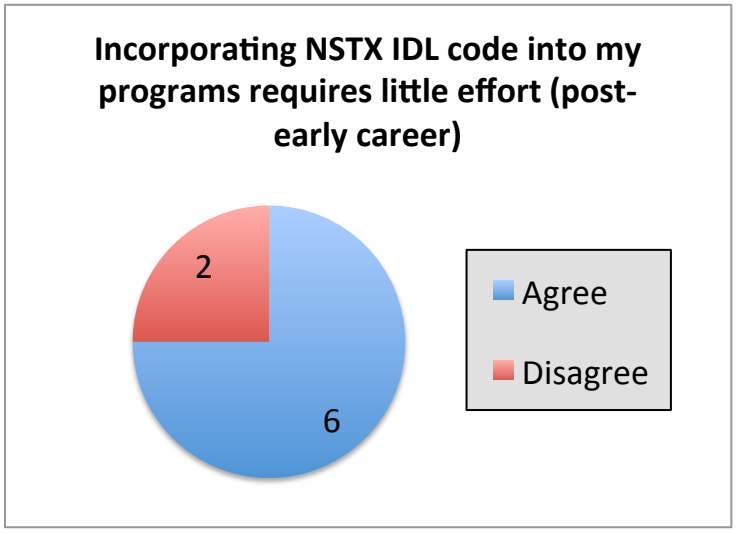


- Overall, about 50% feel it is easy
- Early career responses
  - 33% feel it is easy

Q: Incorporating IDL code requires little effort

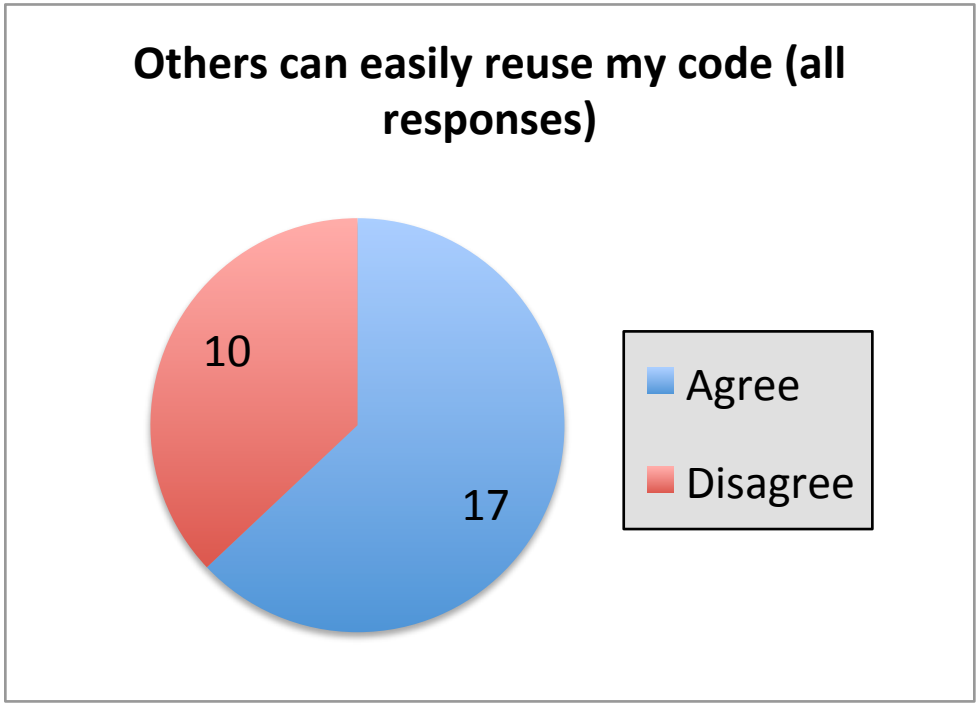


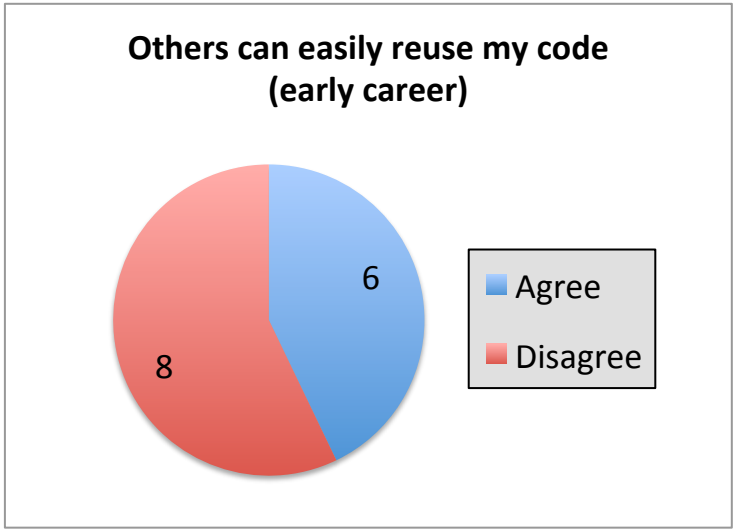


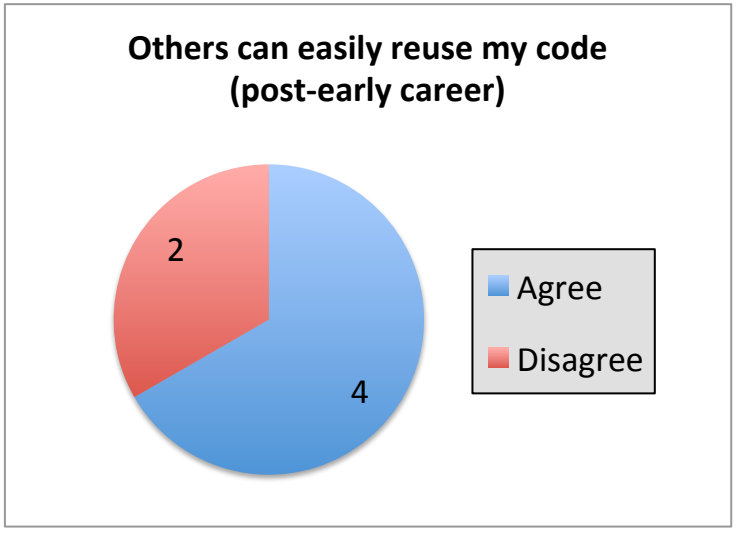


- Overall, about 66% feel incorporating IDL code requires little effort
- Early career responses
  - About 50% feel … little effort

Q: Others can easily reuse my code

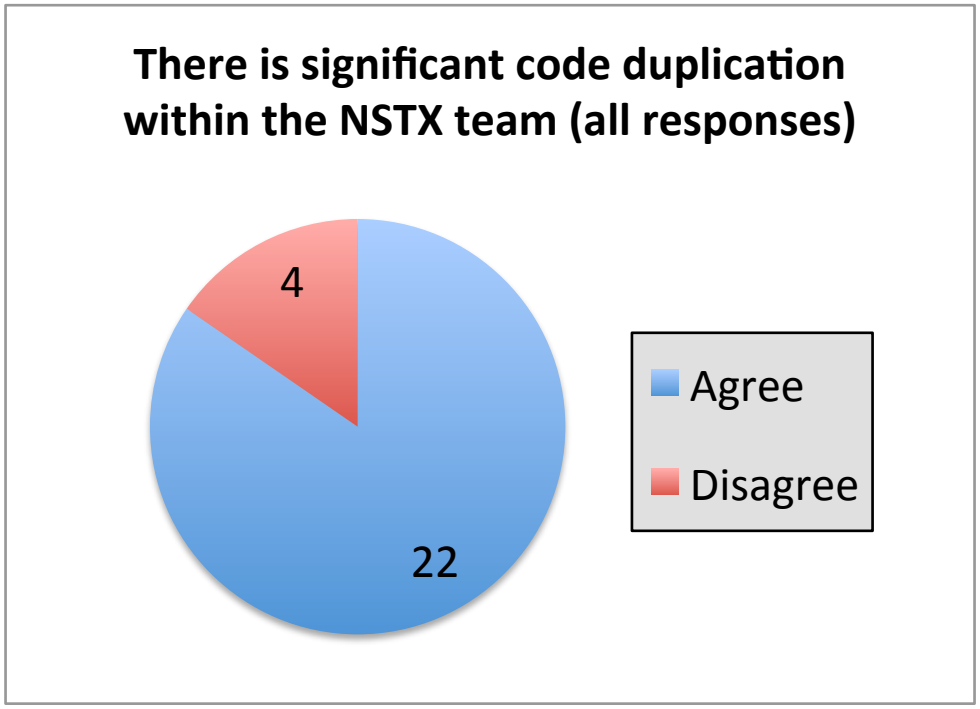


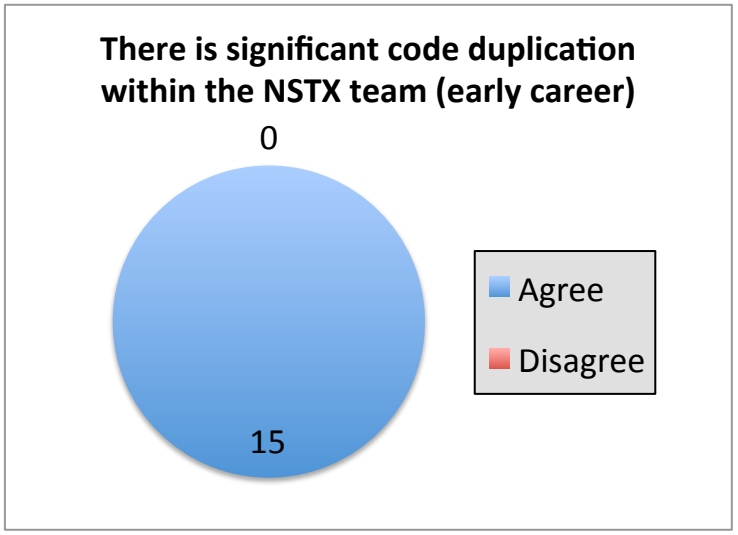


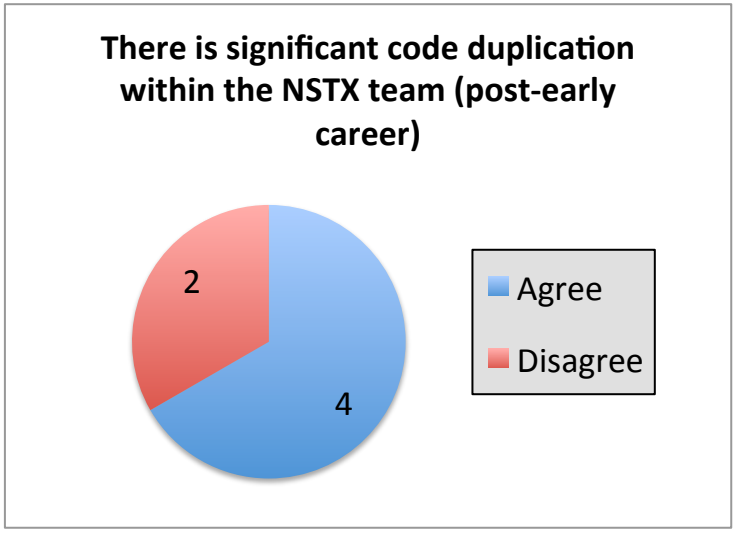


- Overall, about 66% feel their code is easily reused
- Early career responses
  - About 45% in agreement

Q: There is significant code duplication within the NSTX team

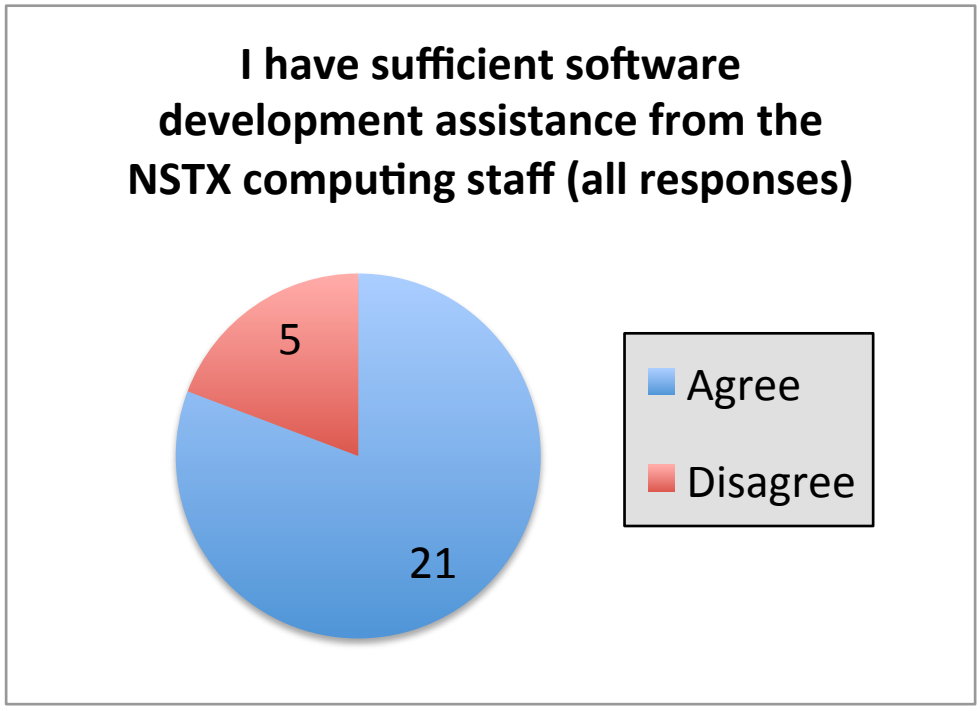


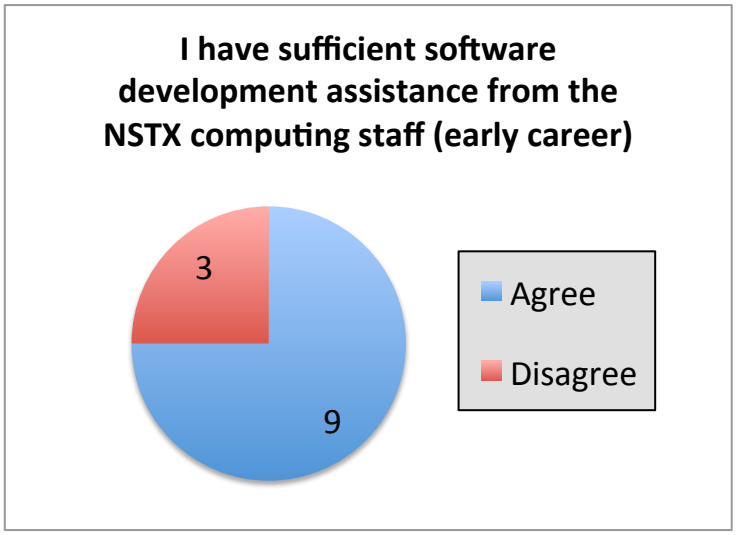


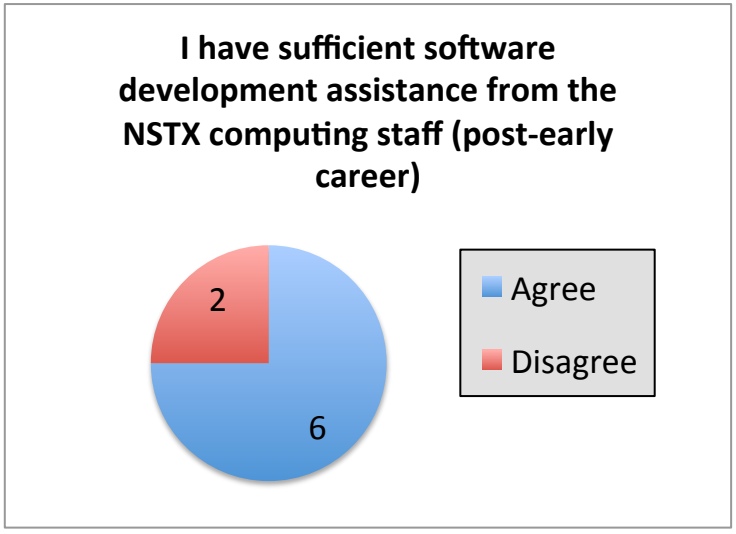


- Overall, broad agreement on code duplication
- Early career responses
  - Unanimous agreement

Q: I have sufficient software assistance from computing staff

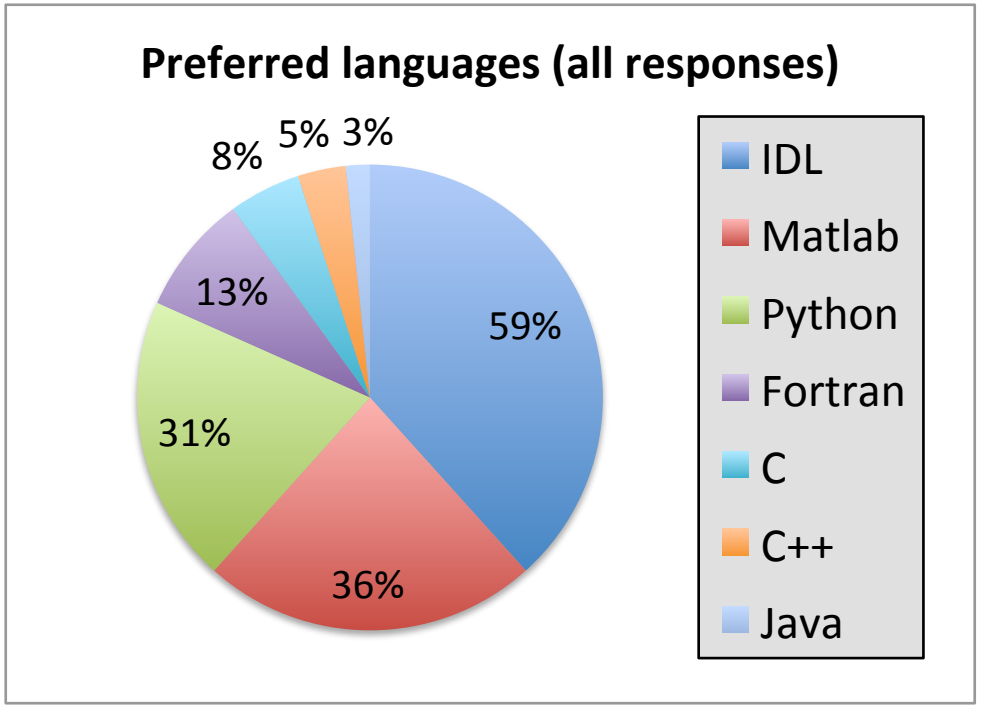


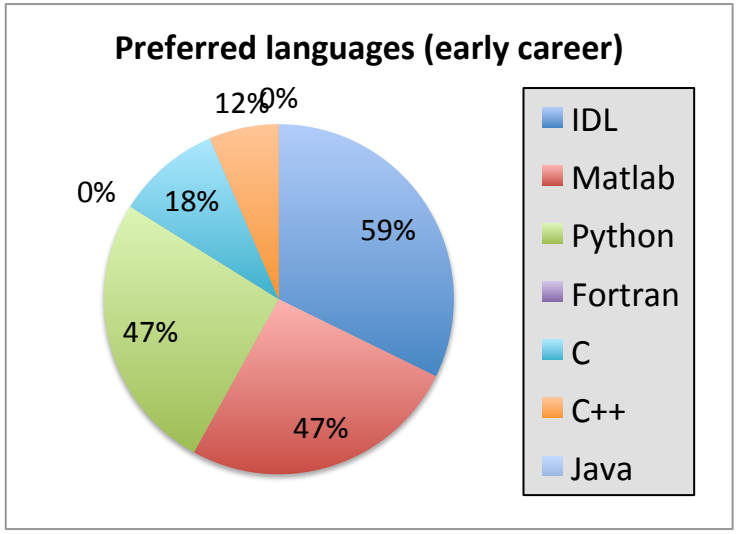


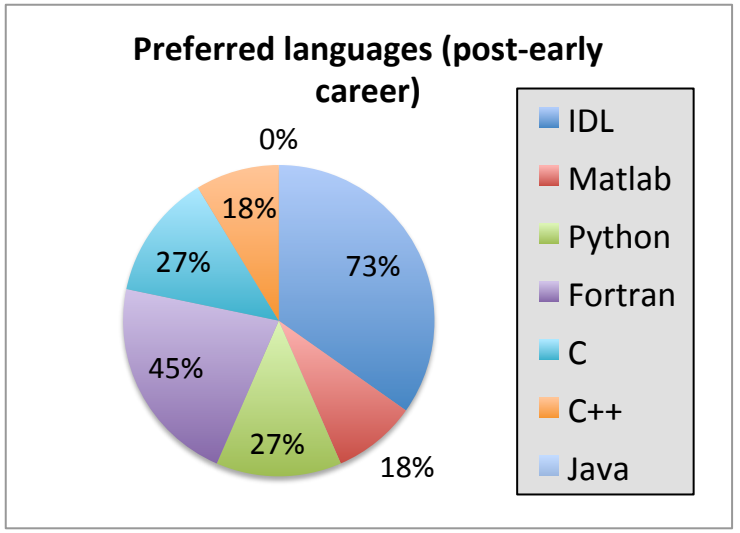


- Clear agreement that sufficient software assistance is available from computing staff

Preferred/used languages







- Overall usage
  - 59% IDL, 36% Matlab, 31% Python
- Early career responses
  - Higher usage of Matlab & Python (47%/ 47%)

Key findings

- **Data tools:** ~ 33% desire more functionality and ~ 50% feel extending functionality is difficult
  - Among early career responses, 50% desire more functionality
- **IDL:** ~ 50% feel IDL code is difficult to find, ~ 33% feel incorporating IDL code is difficult, and ~ 33% feel their code is not reusable
  - 66%, 50%, and 65%, respectively, for early career responses
- Broad agreement that significant and inefficient **code duplication** exists within the NSTX team
- **Language usage:** 60% IDL, 35% Matlab, and 30% Python
  - Among early career responses, more like 60%/50%/50%.
- The core functionality of scopes/ReviewPlus, EFITviewer, and Logbooks are vital to research activities

Challenges and a proposed solution

- The challenges include…
  - Significant code duplication
  - More functionality is desired but extending it is difficult
  - Reusing and sharing IDL code is difficult for many
- However, there is overall agreement that the computing staff provides sufficient software support
- We feel a software framework for data usage can address the identified challenges
  - Improvements in efficiency, extensibility, scalability, and collaborative development

Software framework for data usage

- Deliverables
  - A class object with data decomposed by machine, shot, and diagnostic
    - Methods for data retrieval, caching, import/export, critical or common operations (e.g. de-spiking, interpolation, power spectra), basic plotting, XP/logbook details, data introspection
    - Hide server names and node names; standardize dimensions
  - A platform for collaborative development (e.g. Github)
  - Modern documentation, PPPL/NSTX best practices, etc
- Implementation
  - Lead development in Python
    - Explore strategies to implement functionality in Matlab and IDL
- We're looking for a user community
  - Help implement diagnostics
  - User-driven development of analysis tools using the data class framework

Collaborative development of data tools can address the key findings from the survey

Strategy: software framework for data access & mgmt. + a user community contributing data tools built on the framework

- Software framework (API) for data access and management
  - Data class objects
    - Import/export/cache datasets (also addresses data publication)
    - Hide node-name details and standardize 2D data
    - Built-in default functionality for plotting and signal analysis
  - Environment/usage flexibility
    - Platform independent: local Mac/PC, remote Linux, Python, Matlab
    - Easy-to-advanced usage: easy GUIs, intermediate command-line scripting, advanced programming
- Data tools/GUIs built on the framework by a user community
  - Collaborative development: code repository, version control, bug tracker, feature requests, authoritative documentation
  - Auto-documentation, unit tests, assertions, exception handling
  - Initially target key research functionality (e.g. scopes, EFITviewer, logs)

Fusion Data Framework targets ease of use and expandability

```
In [1]: from fdf import Machine
In [2]: shot = Machine('nstx').s141398
In [3]: shot.mpts.plot()
In [4]: new_Te = s.te.interp(s.ti)
```

streamlined interface for simplified data access (Python, Matlab, IDL?)

diagnostics defined using template markup language

```
<container name='mpts' tree='activespec' path='\\top.mpts.output_data.best'>
    <node name='radius' units='cm'>fit_radii</node>
    <node name='dr' units='cm'>fit_r_width</node>
    <node name='time' units='s'>ts_times</node>
    <signal name='te' units='keV' axis='radius, time'>
        <node name='data'>fit_te</node>
        <node name='error'>fit_te_err</node>
        <method name='plot'>te_plot</method>
    </signal>
    <container name='spline'>
        <node name='radius' units='cm'>spline_radii</node>
        <signal name='te' units='keV' axis='radius, time'>
            <node name='data'>spline_te</node>
.......
```

'Signal' object: data, units, axis, error, methods (plot, interp, map)

can override default methods

Comments from survey responses (9 pages)

Q: Are there other tools you use?

- Tom **Osborne's** python-based pedestal analysis tools
- **Osborne's** Profile analysis written in Python
- **Osborne** pedestal and ELM Python tools – routinely. **OMFIT** - occasionally for now
- I have recently attended the **OMFIT** Demos by O. Meneghini and Brian Grierson and I have taken a quick look at it since it is on portal. It could be a good starting point for some discussion.
- **LRDFIT**, **TRANSP**
- **LRDFIT**, **TRANSP**, **NOVA**-K, **ORBIT**, HYM
- **TRANSP**, **NOVA**, **ORBIT**, FORTRAN
- pecomp
- Toksys
- ImageJ
- Mathematica & other desktop applications

Q: What new functionality or features do you want? (I)

- Access to **rtEFIT data** on eng_test and eng_dev trees
- **ReviewPlus** is too hard to use - too many capabilities that are hard to master. **WebTools** has a lot of potential, but it is difficult to add configuration files. LOCUS interface is ancient. **RPLOT** has limited graphics capability.
- It would be great if **WebTools** was able to **filter by engineering notes** commonly put in **Logbook** (e.g. Li used) or by shot quality. Also, an easily accessible **list of common** MDS+ tags for overview plotting would be useful.
- **RPLOT** should be definitely updated
- In **review plus**, I'd like to have spectrograms (contour plots of time dependent spectra). I've made an initial effort to add this capability, but it needs a lot of work. For **WebTools**, I'd like capabilities for creating, saving, editing and managing lists of signals (or better yet records of settings) for the multi signal plotting tools in order to routinely reuse them, much like scope files. I've talked to Bill Davis about this and large parts of this capability already exists, but it's not as friction free as I'd like. I'd also like the **mdsplus signal list page** to offer listings of signals from earlier times. The tree structures evolve over time, which has never been more apparent than in the last few years as the structure of many trees have evolved in anticipation of NSTX-U. Some signals that are still available for NSTX data are no longer included in the list, making it hard to find data that exists for NSTX.

Q: What new functionality or features do you want? (II)

- I am new user of the NSTX database. It would be nice to have an overview about the different options. Setting up **remote access** has taken me quite some time so far. But I finally got idl running, after setting up VPN, various accounts and privileges.
- Ability to plot **multiple shots**.
- For **EFITviewer**, greater control of flux surfaces displayed.
- A more robust **jScope**/shot data viewer. Easier access to MDS+ tag names in **Webtools** (many tags aren't listed or some that exist have been abandoned)
- Tools to create databases?
- **EFITviewer** with q=1,3/2,2,2.5,3 etc **diagnostic for overlay on flux cross section** plot. Diagnostic showing major poloidal probes and saddle loops.
- **EFITviewer**: ability to get basic field quantities (density, Bfield, pitch angle, ect) at the mouse location
- I like **ReviewPlus**, but this program tends to be unstable at PPPL. I use **DWScope** due to its stability, but wish it had the **functionality of ReviewPlus**, such as plotting two signals on one plot, controlling color, etc. **jScope** is nice for plotting multiple shots, but also has stability issues. **EFITviewer** should include all diagnostic overplots and profile plotting (similar to GA version).

Q: What new functionality or features do you want? (III)

- 1) Make **ReviewPlus** more stable. Try to keep it updated with the GA version. 2) **WebTools** is good, but perhaps the web page could be improved to make things easier to find. 3) The ones I have marked as "no experience" should be described under WebTools. A short paragraph of what it does, how to access it and use it.
- "In part of my research I need **GEQDSK** files and kinetic profiles but I have different ways to get them: (i) via TRANSP, (ii) with "trxpl" script (still basically TRANSP, I think) which generates GEQDSK and plasma state for a given shot and time, (iii) via idl routine by J. Menard ("get_hhfw_data"). I do not know (I've never checked them) the differences behind these approaches. This is just an example. Perhaps, there are other clever way to get them but should not be better to have a "standard/common" approach? Perhaps the possibility to save an GEQDSK file directly from a EFITviewer could be useful...just a thought."
- **jScope** - need a smooth installation. **ReviewPlus** - ready to discuss additional functionality.

Q: Additional comments? (I)

- **Python** is the language of the future, but there is no simple way for me to pick it up. I was encouraged by the discussion of making EFIT available to the common man via **OMFIT**. **TRANSP** results data access really needs to be developed so it is more accessible. We can't afford to loose **IDL** and **MATLAB**.
- beyond just "my personal codebase", the loss of **LRDFIT**, if **IDL** were to vanish, would be a major hit to the team. Same could be said for idl ISOLVER. Also, I am concerned that this panacea of "idl libraries for all" will not work well given that nobody is committing to maintain/upgrade/enhance codes. Sure, I can provide you with profile fitting routines, but I'm probably not gonna update them for you unless I am personally motivated...
- The **shot summaries** at DIII-D, containing time traces of common parameters (Ip, Pnbi, Ne, Dalpha) and select logbook notes, are a very useful first step for understanding data for an **outsider** who wasn't part of the XP, and thus doesn't know the XP plan or which shots performed well. An **archived, scrollable list** of these would be a welcome addition to WebTools.
- More examples in a single easily accessible location, web preferred for remote collaborators.

Q: Additional comments? (II)

- It would be a big step forward to extend the **documentation** on available "standard" analysis tools, especially for **IDL** routines. This is especially important for collaborators, who may not know who is the "unofficial" contact person to ask for specific types of analysis. A **centralized Wiki** would do it, with the obvious drawback that someone has to maintain it… At present, I have the impression that there are **too many "similar" analysis tools** that nominally do the same thing, but results may differ. At least for the more common analyses, there should be only one "officially blessed" tool, so that different people can report the same results when looking (or publishing!) data from the same shot.
- I've been writing analysis routines in idl and matlab for NSTX data. It would be difficult to overstate how big the loss would be if they were suddenly removed. However in my 25 years of experience with idl, I know that really **old code is often forgotten** so if it were necessary to end the use idl and matlab in favor of something else, I figure a five to ten year period where idl and matlab are officially deprecated (and NOT updated) would give me plenty of time to rewrite my most frequently used and most important software. Also, steady advances in the capabilities of the new language, combined with a lack of updates in matlab/idl would **incentivize the transition**, because I would want the new capabilities.

Q: Additional comments? (III)

- Software tools are fine if you are focused on your own diagnostics for your own XP. What is difficult is to get global quantities from past years of data. For example, to establish whether Zeff in the past was higher or lower with boronization than with lithium. The prospect of finding out feels like **exploring the Pacific Ocean in a row boat** and no one has time for that. If there is a database that can do this I would like to hear about it. But maybe it needs a Google-like or Siri-like interface that understands questions in plain English. Or maybe a short course **overview of all the existing software**.
- I readily admit I probably have less experience than most in NSTX data analysis, but I do wish there were more **uniform standards for analysis**. For example, I would be very interested in there being a (limited) set of agreed upon standards/ routines for: [1] **profile fitting** (how does one treat data mapping, poloidal asymmetries, simultaneously good pedestal and core fitting, etc...), [2] **equilibrium** reconstruction (good kinetic fits, iteration with TRANSP/NUBEAM fast ion pressure, including centrifugal effects, etc...). These two things are needed for a huge amount of more sophisticated analyses, but it seems like everyone does it differently (EFIT, LRDFIT, Osborne profile fits+kEFIT, ...). In the spirit of developing more integrated XPs (numerous people and topical groups using the same shots for various analyses), it would be good to standardize these things as best possible. I know OMFIT was recently installed at PPPL, so perhaps this is a path forward to help standardize some of these issues.

Q: Additional comments? (IV)

- It would be nice if a terminal emulator with the functionality of VersaTerm Pro could be found and supported. **Easy save of graphics** to pict files (Intaglio doesn't open IDL postscript files), saves last 20-30 graphics plots. XTC + X11 fulfills some of those needs, but **saving graphics files** to my Mac is a tedious process. Write ps files, scp file to mac, translate ps file to pict or pdf.
- **Adopting a standard** edge profile analysis tool (such as **GAProfiles**, **Osborne's** tools or **OMFIT**) that is used by the team would enable easier processing and sharing of results within NSTX and the world-wide community.
- If **IDL** disappears, a set of small routines should be written by the computing staff, that allows the user to begin using the new language without spending too much time. It would also be great is similar routines are written for **Matlab**. For example, to plot a bunch of traces from the NSTX data. Egemen may know how to do this because he has been doing this at D3D, but indicated the procedure may be more difficult for NSTX/NSTX-U?
- I have built an **IDL/python bridge** that solves most of my most significant transitional issues, however more work is needed to make this an easily usable public interface (it is public, but not so polished)

Q: Additional comments? (V)

- 1) for a specific data analysis we should have a **common/standard numerical tool** (already tested and with documentation). 2) moving towards open-source language, e.g., **Python** (although I am not yet so familiar with it and, I think, NSTX team is more IDL (a bit MATLAB) oriented, it could be a good and useful step to make. 3) **standard routines** (already tested) for ""basic purposes"" should be **common tools** (I know that basically some of them - or perhaps all - already exist). Just few examples: - get kinetic profiles (with different fitting choices) as a function of rho_pol, rho_tor, R, etc. - get an **EQDSK** file for specific shot and time- - get the components of the magnetic field - get all plasma and physics quantities for 2D/3D plotting 3) some ""**modern/user friendly**"" language/tools can be useful for our own research and for **new collaborations** and they can have a positive impact/impression when someone works with **young students**.
- No need to reinvent the wheel. Suggest porting most/all tools from GA/DIII-D. This is especially useful when one machine is not running and researchers go on research assignment to the other machine. Ideally, most **tools should be standardized and unified between NSTX-U and DIII-D.**

Q: Additional comments? (VI)

- Any help will be appreciated in writing an **interface program for importing NSTX-U data** directly into codes written in the language of choice, Mathematica in my case, instead of having to first generate numerical files, which will make data analysis much more efficient.
- Just a note that **Nomachine** is invaluable for offsite collaborators.