Distributed Development:
Lessons learned by Herschel

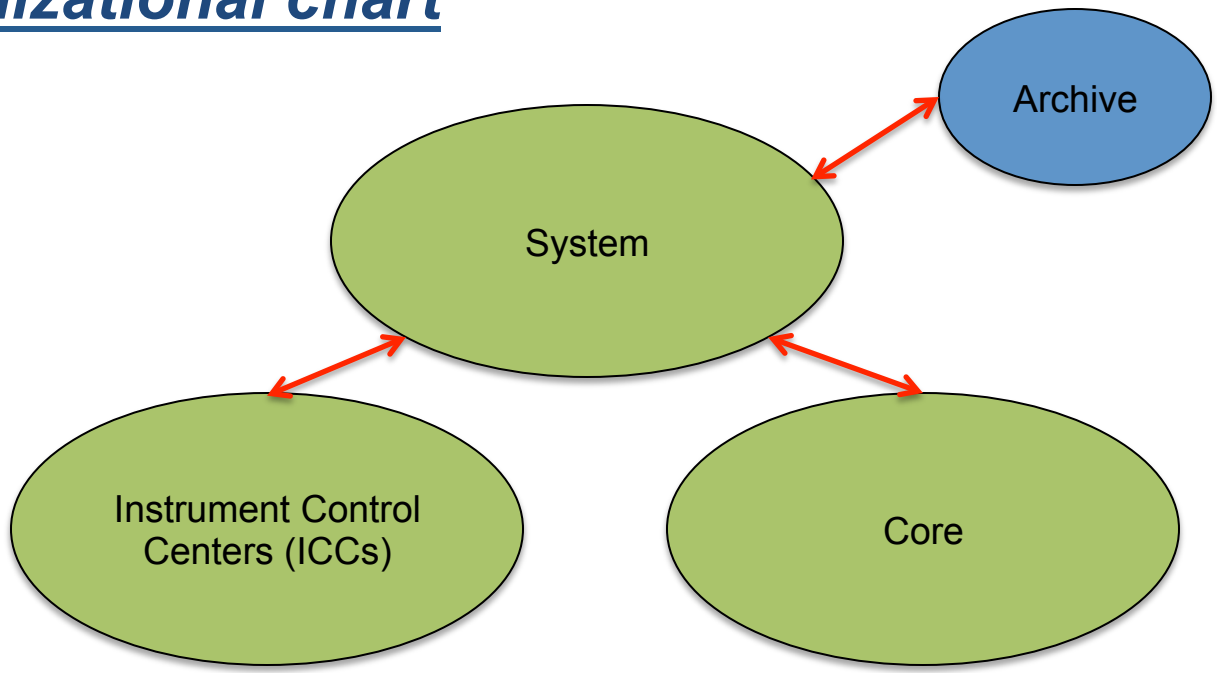
GRITS 2011, June 17

Colin Borys

Overview

- Ground Segment Organizational chart
- The Practical considerations
- General Development Infrastructure
- Inherent conflicts and how to cope
- Summary of lessons learned

Organizational chart



- Each software area manages their own CCB (configuration control board), which prioritizes work.
- Each also has their own manager and software QA.

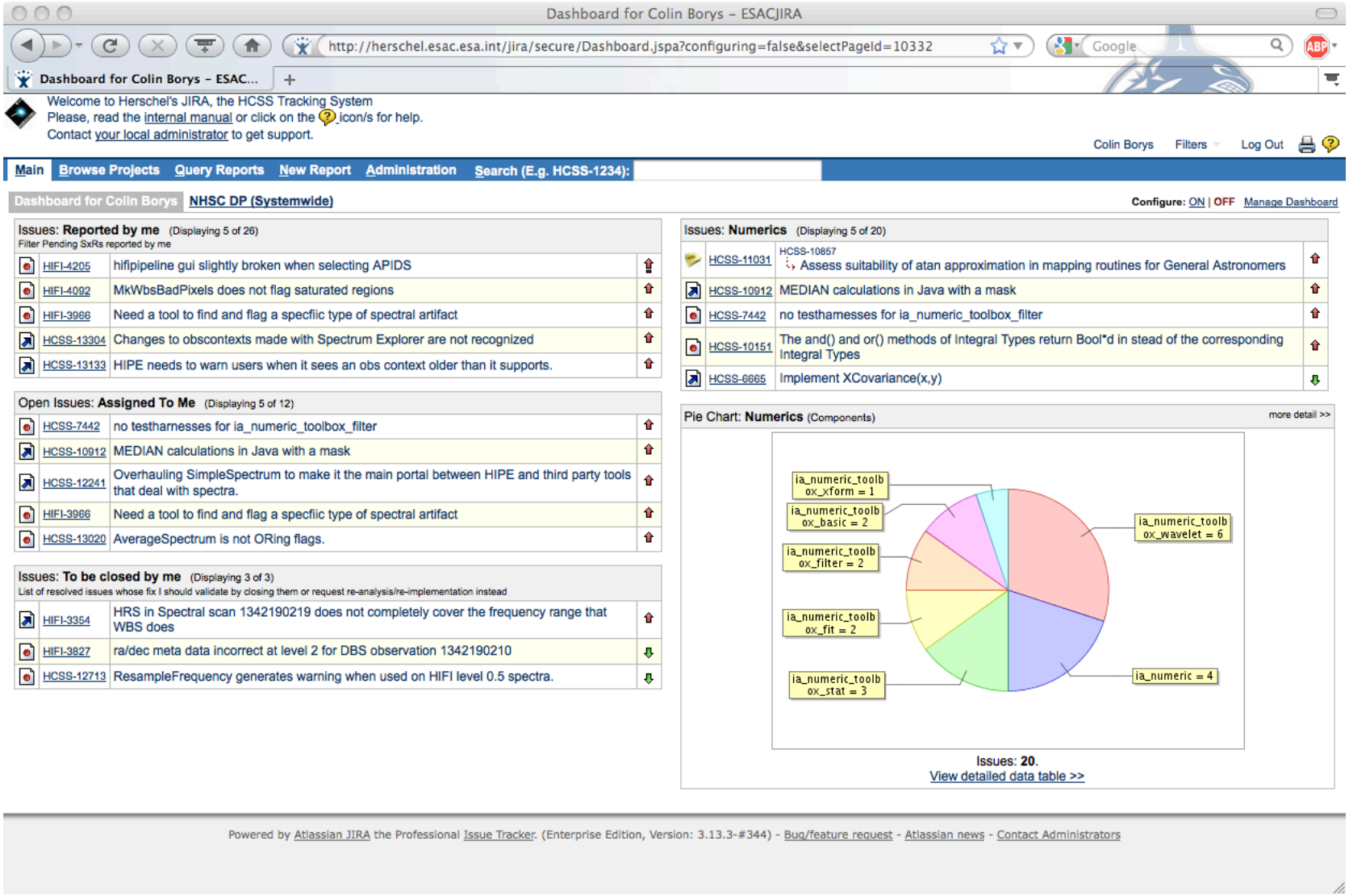
Practical Considerations

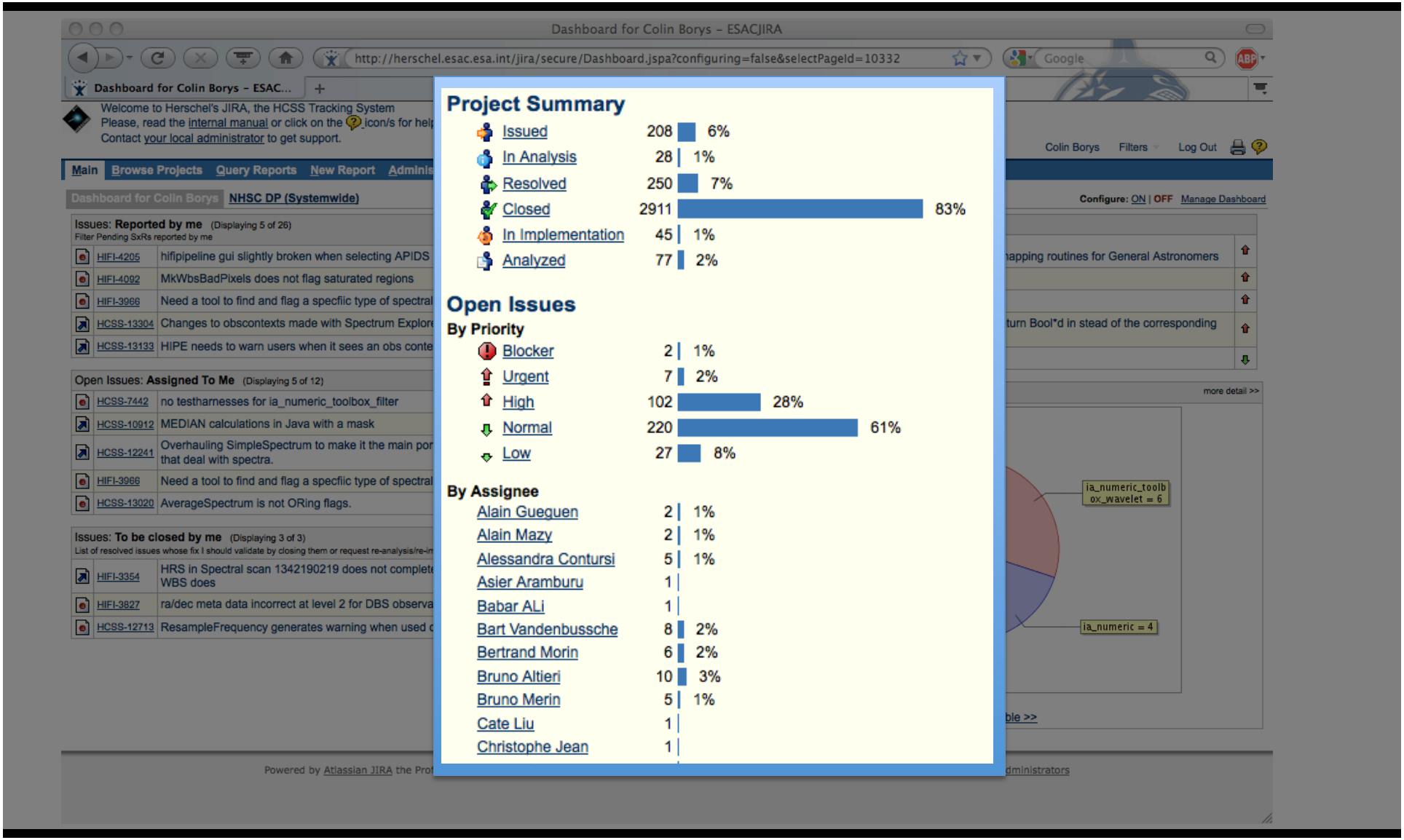
- The System Architect, QA Engineer, and top level managers define the entire development framework. Hire good ones!
- With developers spread across ~15 timezones, interaction is a challenge:
  - With Europe, we generally have telecons at their end of day/ our start of day. (6am)
  - NHSC also has a representative onsite at ESAC (Madrid) to represent us at other meetings. (David Ardilla)

- The emergence of social networking, SKYPE, and now webex are also invaluable

Development Infrastructure

- **IDE: Eclipse**. Common, powerful, and has the ability to import project-specific plug-ins to aid in development conformity
- **Code Repository: CVS**. Old, but it works.
- **Ticketing System: JIRA**. Very effective, very configurable.
- **Compilation: CIB** (Continuous Integration Build) approach.
- **Testing:** Test harnesses, nightly tester, once per release acceptance testing.


Dashboard for Colin Borys - ESACJIRA
http://herschel.esac.esa.int/jira/secure/Dashboard.jspa?configuring=false&selectPageId=10332
Dashboard for Colin Borys - ESAC...
Welcome to Herschel's JIRA, the HCSS Tracking System
Please, read the internal manual or click on the icon/s for help.
Contact your local administrator to get support.
Colin Borys Filters Log Out
Main Browse Projects Query Reports New Report Administration Search (E.g. HCSS-1234):
Dashboard for Colin Borys NHSC DP (Systemwide)
Configure: ON | OFF Manage Dashboard
Issues: Reported by me (Displaying 5 of 26)
Filter Pending SxRs reported by me
HIFI-4205 hifipipeline gui slightly broken when selecting APIDS
HIFI-4092 MkWbsBadPixels does not flag saturated regions
HIFI-3966 Need a tool to find and flag a specfiic type of spectral artifact
HCSS-13304 Changes to obscontexts made with Spectrum Explorer are not recognized
HCSS-13133 HIPE needs to warn users when it sees an obs context older than it supports.
Open Issues: Assigned To Me (Displaying 5 of 12)
HCSS-7442 no testharnesses for ia_numeric_toolbox_filter
HCSS-10912 MEDIAN calculations in Java with a mask
HCSS-12241 Overhauling SimpleSpectrum to make it the main portal between HIPE and third party tools that deal with spectra.
HIFI-3966 Need a tool to find and flag a specfiic type of spectral artifact
HCSS-13020 AverageSpectrum is not ORing flags.
Issues: To be closed by me (Displaying 3 of 3)
List of resolved issues whose fix I should validate by closing them or request re-analysis/re-implementation instead
HIFI-3354 HRS in Spectral scan 1342190219 does not completely cover the frequency range that WBS does
HIFI-3827 ra/dec meta data incorrect at level 2 for DBS observation 1342190210
HCSS-12713 ResampleFrequency generates warning when used on HIFI level 0.5 spectra.
Issues: Numerics (Displaying 5 of 20)
HCSS-11031 HCSS-10857 Assess suitability of atan approximation in mapping routines for General Astronomers
HCSS-10912 MEDIAN calculations in Java with a mask
HCSS-7442 no testharnesses for ia_numeric_toolbox_filter
HCSS-10151 The and() and or() methods of Integral Types return Bool*d in stead of the corresponding Integral Types
HCSS-6665 Implement XCovariance(x,y)
Pie Chart: Numerics (Components)
more detail >>
ia_numeric_toolbox_xform = 1
ia_numeric_toolbox_basic = 2
ia_numeric_toolbox_filter = 2
ia_numeric_toolbox_fit = 2
ia_numeric_toolbox_stat = 3
ia_numeric_toolbox_wavelet = 6
ia_numeric = 4
Issues: 20.
View detailed data table >>
Powered by Atlassian JIRA the Professional Issue Tracker. (Enterprise Edition, Version: 3.13.3-#344) - Bug/feature request - Atlassian news - Contact Administrators


Dashboard for Colin Borys - ESACJIRA
http://herschel.esac.esa.int/jira/secure/Dashboard.jspa?configuring=false&selectPageId=10332
Project Summary
Issued 208 6%
In Analysis 28 1%
Resolved 250 7%
Closed 2911 83%
In Implementation 45 1%
Analyzed 77 2%
Open Issues
By Priority
Blocker 2 1%
Urgent 7 2%
High 102 28%
Normal 220 61%
Low 27 8%
By Assignee
Alain Gueguen 2 1%
Alain Mazy 2 1%
Alessandra Contursi 5 1%
Asier Aramburu 1
Babar ALi 1
Bart Vandenbussche 8 2%
Bertrand Morin 6 2%
Bruno Altieri 10 3%
Bruno Merin 5 1%
Cate Liu 1
Christophe Jean 1

JIRA workflow for a person who submits a ticket

User submits a ticket	Ticket appears on users' '*submitted by me*' panel Ticket appears on developer's '*assigned to me*' panel Ticket status is '**Assigned**'
Developer analyzes issue	Ticket status is changed to '**In Analysis**'
Developer starts implementation	Ticket status is changed to '**In Implementation**'
Developer fixes issue	Ticket status is changed to '**Resolved**' Ticket disappears from developer's panel Ticket appears on users' '*to be closed by me*' panel
User tests implementation	If test passed, user sets the ticket to '**Completed**' and the workflow is complete.

E-mail is sent to assignee, developer, and mentor at each step

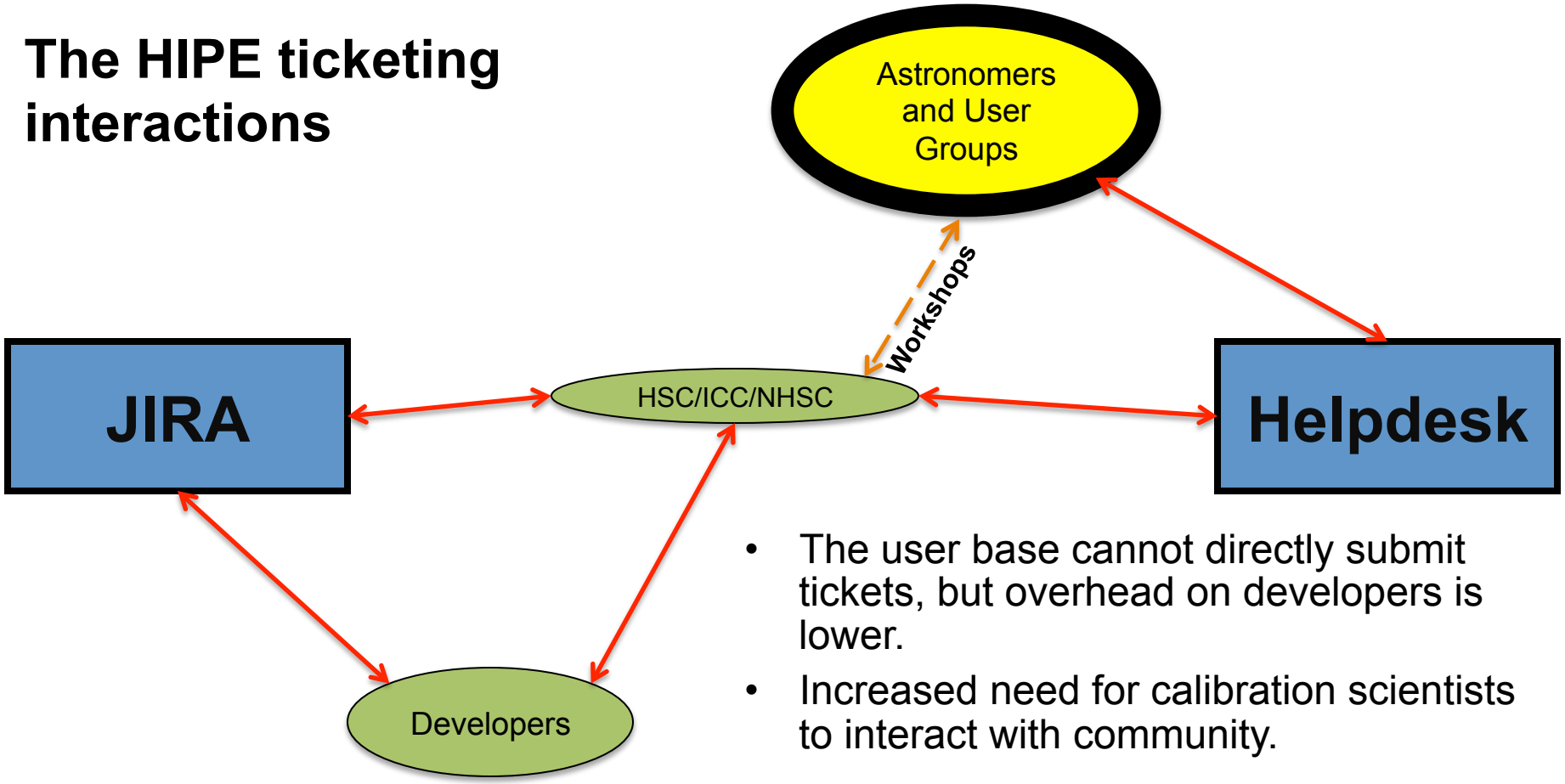
Consequences of JIRA workflow policy

- In our setup, the person who submits a bug report is also responsible for testing and closing the ticket once a developer fixes it.
- We do not release software when a ticket assigned to that version is 'resolved' but not closed.

- It is natural for a lot of development to happen near a code freeze, thus the testing duties for reporters get compressed.

- Consequence is that people who report bugs are inherently punished and this provides some motivation to work 'outside' the system.

The HIPE ticketing interactions



- The user base cannot directly submit tickets, but overhead on developers is lower.
- Increased need for calibration scientists to interact with community.

The Developer-Mentor policy

- HIPE is made up of >100 component packages (i/o, numerical, etc.)
- For each, there is a developer (or more) and mentor assigned.
- The majority of the packages have a calibration scientist as a mentor, and it is their job to :
  - Advise developer on astronomer specific issues
  - Vet tickets that are incorrectly assigned to a package
  - Advise management on the priority of tickets in that package.
  - Aid in documentation that is directed towards users.

- Good idea but can fail in practice (over tasked, lack of expertise for shared packages with a broad user base, ignored)

Continuous Integration Builds

- HIPE takes a long time to compile, and it used to be possible for conflicts to occur on packages under heavy development.
- Was particularly problematic around a code freeze.

- With a CIB, a new minor version of the software is created every time new code in a component is checked in. Therefore changes in package Y are immune to changes in X if Y is checked in first.

- Any code that does break the build becomes ‘quarantined’, and the owners of X and Y figure out why, and fix it.

build	delivered	elapsed	hcss.common	hcss.odb	hcss.dp.core	hcss.dp.hifi	hcss.dp.pacs	hcss.dp.spire	hcss.apps	hcss.hscops	hcss.services	hcss.dp.all	hcss.icc.hifi	hcss.icc.pacs	hcss.icc.spire	hcss	comments
1153	2011-06-17 06:19:36 +0200	1:47:04	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	ia_pal_pool_lstore-1.228
1152	2011-06-17 03:20:58 +0200	0:10:33	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	spire_ia_pipeline_phot_fluxconv-1.36
1151	2011-06-17 02:11:21 +0200	0:11:39	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	spire_ia_pipeline_phot_baseline-0.31
1150	2011-06-16 21:51:18 +0200	0:08:04	✓	✓	✓	✗	✓	✓	✓	✓	✓			✓	✓		hifi_dp_tools-0.102
1149	2011-06-16 19:36:49 +0200	0:28:01	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	pacs_share-1.147
1148	2011-06-16 19:06:15 +0200	0:10:59	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	spire_ia_pipeline_scripts-2.53
1147	2011-06-16 18:53:28 +0200	0:16:12	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	ia_gui_cube-0.187
1146	2011-06-16 18:34:20 +0200	0:17:09	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	pacs_spg_spec-0.175
1145	2011-06-16 18:14:12 +0200	0:28:14	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	pacs_share-1.146
1144	2011-06-16 17:43:31 +0200	0:02:36	✓	✓	✗				✓								ia_gui_cube-0.186
1143	2011-06-16 17:40:06 +0200	0:12:57	✓	✓	✓	✓	✓	✓	✗	✓	✓		✓	✓	✓		mps-1.121
1142	2011-06-16 17:25:44 +0200	0:23:38	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	pacs_spg_common-0.157
1141	2011-06-16 17:00:33 +0200	0:14:20	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	pacs_spg_pipeline-0.202
1140	2011-06-16 16:43:33 +0200	0:32:07	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	cus-1.252
1139	2011-06-16 16:09:00 +0200	0:10:26	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	spire_ia_pipeline_hipe-2.22
1138	2011-06-16 15:55:37 +0200	1:00:04	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	ia_gui_plot-2.278
1137	2011-06-16 14:52:36 +0200	1:13:02	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	ia_pal_pool_hsa-3.53

Incoming

There are currently 0 modules in the incoming queue.

Quarantine

There are currently 6 modules in the quarantine queue.

Module	Version	Date of Arrival (CET)
hifi_dp_tools	0.102	2011-06-16 21:36:07
mps	1.121	2011-06-16 12:14:56
ia_dataset_gui	8.6	2011-06-13 16:14:45
hifi_cal	0.97	2011-06-09 18:07:53
ia_gui_explorer	0.73	2011-06-08 18:51:07
share_param	0.3	2011-04-08 10:48:39
Module	**Version**	**Date of Arrival (CET)**

Processed

There are currently 953 modules in the processed queue.

Module	Version	Date of Arrival (CET)
ia_pal_pool_lstore	1.228	2011-06-17 04:24:43
spire_ia_pipeline_phot_fluxconv	1.36	2011-06-17 03:04:09
spire_ia_pipeline_phot_baseline	0.31	2011-06-17 01:54:10

module		version	compile		test	coverage		size (MB)		elapsed
	status		main	test		%	#	devel	user	mm:ss
examples_hifi		0.3	✓					0.00	0.00	0:00
hifi_cal	✚ ☀	0.95	✓	✓	✓	79	11625	0.77	0.10	0:13
hifi_data		0.29		✓				526.79	0.00	0:07
hifi_dp_access		0.32	✓	✓	✓	12	13633	1.36	0.08	0:00
hifi_dp_dataflow		0.9	✓	✓	✓	15	3229	0.38	0.02	0:00
hifi_dp_dataset_spectrum		0.31	✓	✓	✓	!	!	0.61	0.01	0:04
hifi_dp_deconvolution	☀	0.74	✓	✓	✓	60	9986	5.44	0.04	0:31
hifi_dp_gui_spectrum	☀	1.7	✓	✓	✓	84	375	0.79	0.01	0:12
hifi_dp_otf	✚ ☀	5.28	✓	✓	✓	65	30908	31.82	2.64	0:26
hifi_dp_stability		0.5	✓	✓	✓	94	976	0.66	0.01	0:00
hifi_dp_standingwaves	✚ ☀	0.30	✓	✓	✓	47	7848	0.23	0.04	0:24
hifi_dp_task_spectrum		0.34	✓	✓				0.09	0.00	0:00
hifi_dp_tools		0.100	✓					0.37	0.20	0:02
hifi_fpu		0.75	✓	✓	✓	65	11814	1.12	0.08	0:00
hifi_hrs	☀	0.118	✓	✓	✓	92	13583	4.51	2.34	0:52
hifi_hrs_qla		0.6	✓	✓	✓	81	14313	4.08	0.06	0:16
hifi_manuals_pipeline		0.11	✓					2.26	1.13	0:00
hifi_manuals_shared		0.2	✓					0.01	0.00	0:00

Software testing

- Occurs at many levels, the first being the code test harnesses associated with each component. This is easily one of the most controversial areas we deal with.

- Scripts designed to run the system in many different areas are automated once per night and the output compared to an expected value. This catches bugs that don't break the build but do break the system. However does not test as much code as the test harnesses.

- Finally, every major release goes through extensive acceptance testing.

Inherent conflicts

- Developers and Astronomers often have a different view on how things should be implemented. Data access in HIPE for example is sophisticated and powerful, but until recently only expert Astronomers could actually read in data easily!

- The US mandate is to support the US Astronomer. The European one also includes development of HIPE for future ESA missions. Code quality reviews places NHSC in a difficult position.

(some) Lessons learned

- Know your colleagues, figure out who they work for, and what they are hired to do.
- Don't let developers write requirements, but don't let astronomers limit developers.
- Pair developers with calibration scientists
- Hire good people at the top
- Embrace new technologies/approaches (i.e. social media, CIB)
- Purchase good development tools.
- Monitor policy decisions…they often have unintended side-effects
- Emphasize testing at every opportunity, but be flexible.
- No amount of requirement or policy planning will prevent conflict.