Vision and Status of the IOC to FOC Process

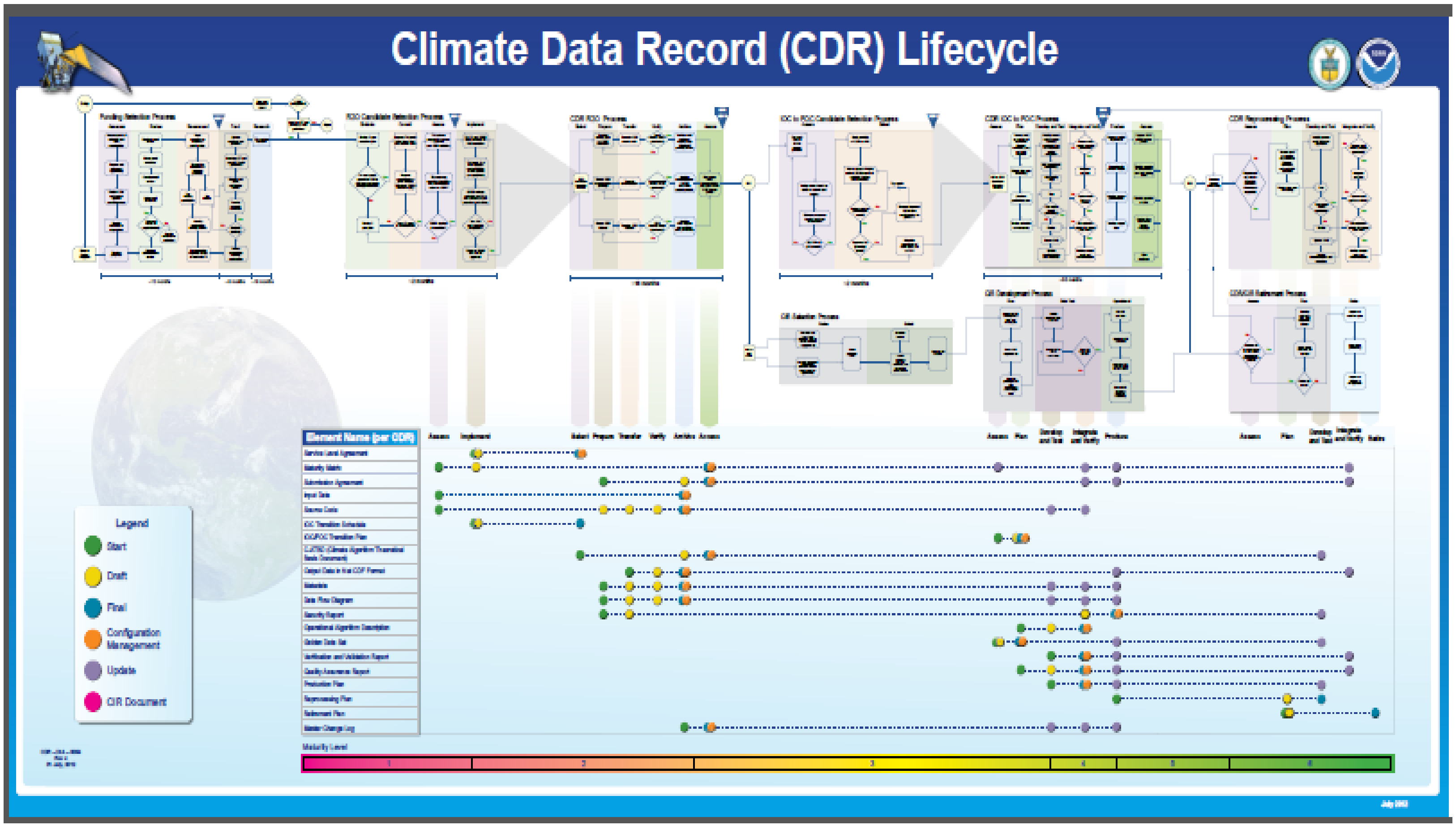
Optimizing Software Maintenance

NOAA's Annual Climate Data Record Conference

July 31 through August 2, 2012

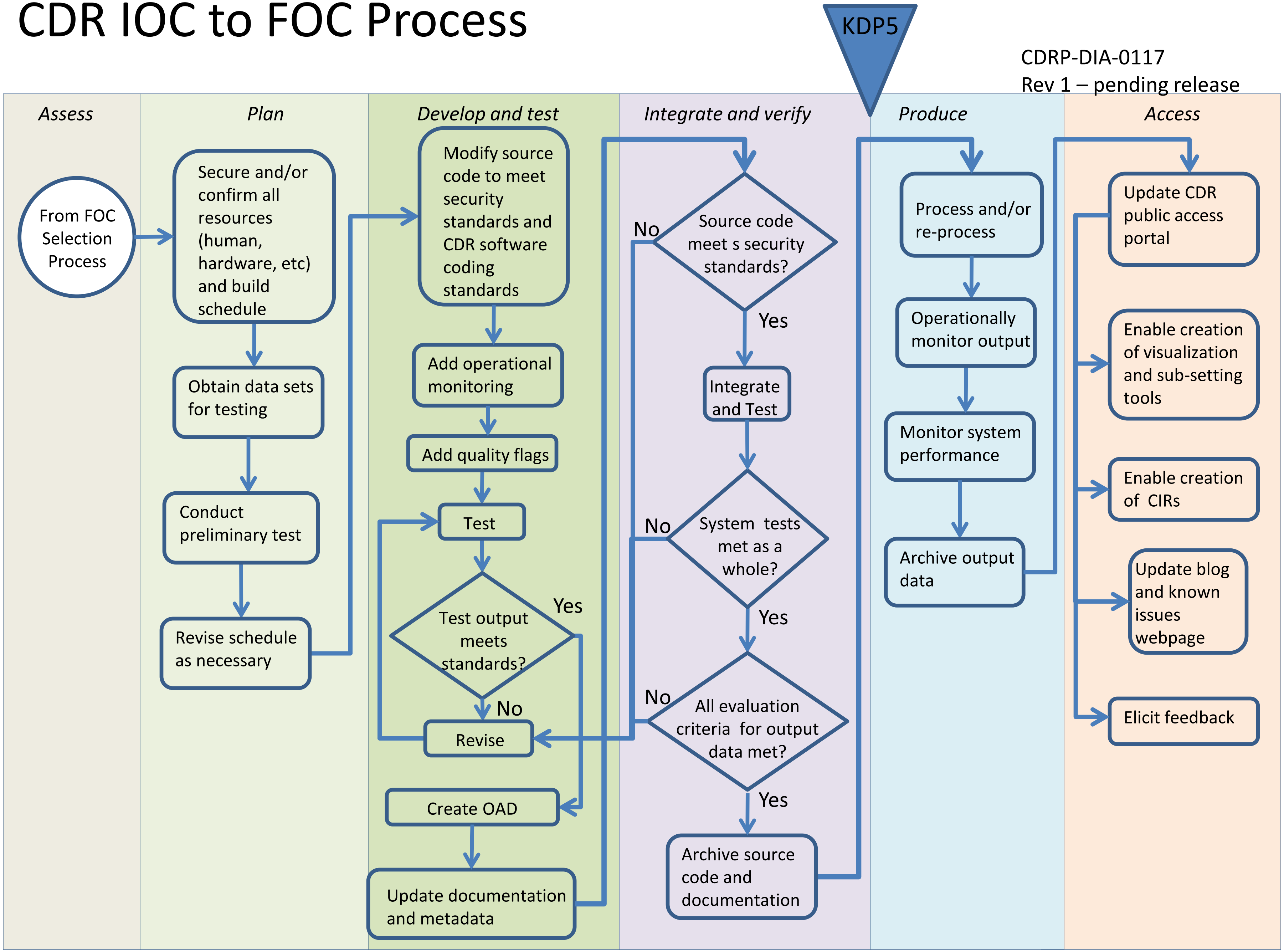
Bryant Cramer
GST Consultant

CDR Lifecycle



CDR IOC to FOC Process

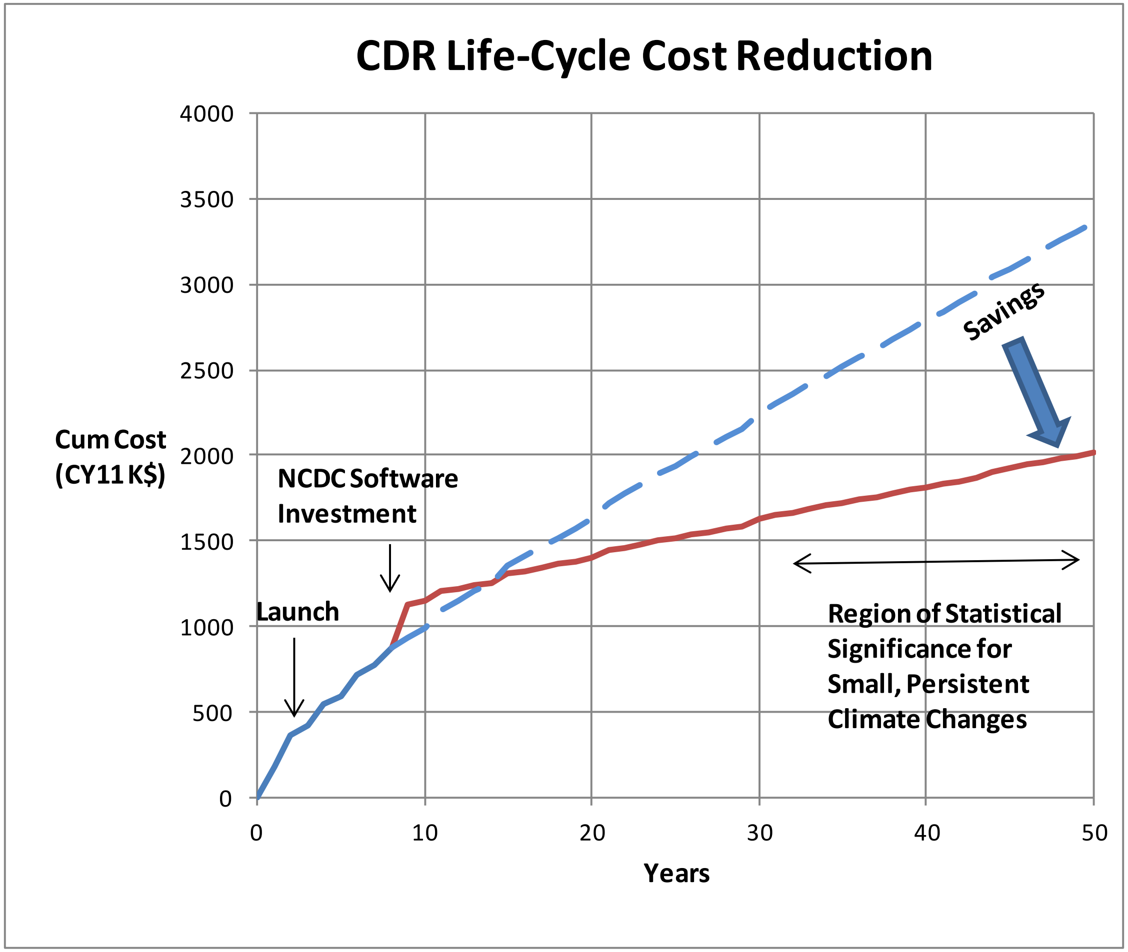
CDRP-DIA-0117
Rev 1 – pending release



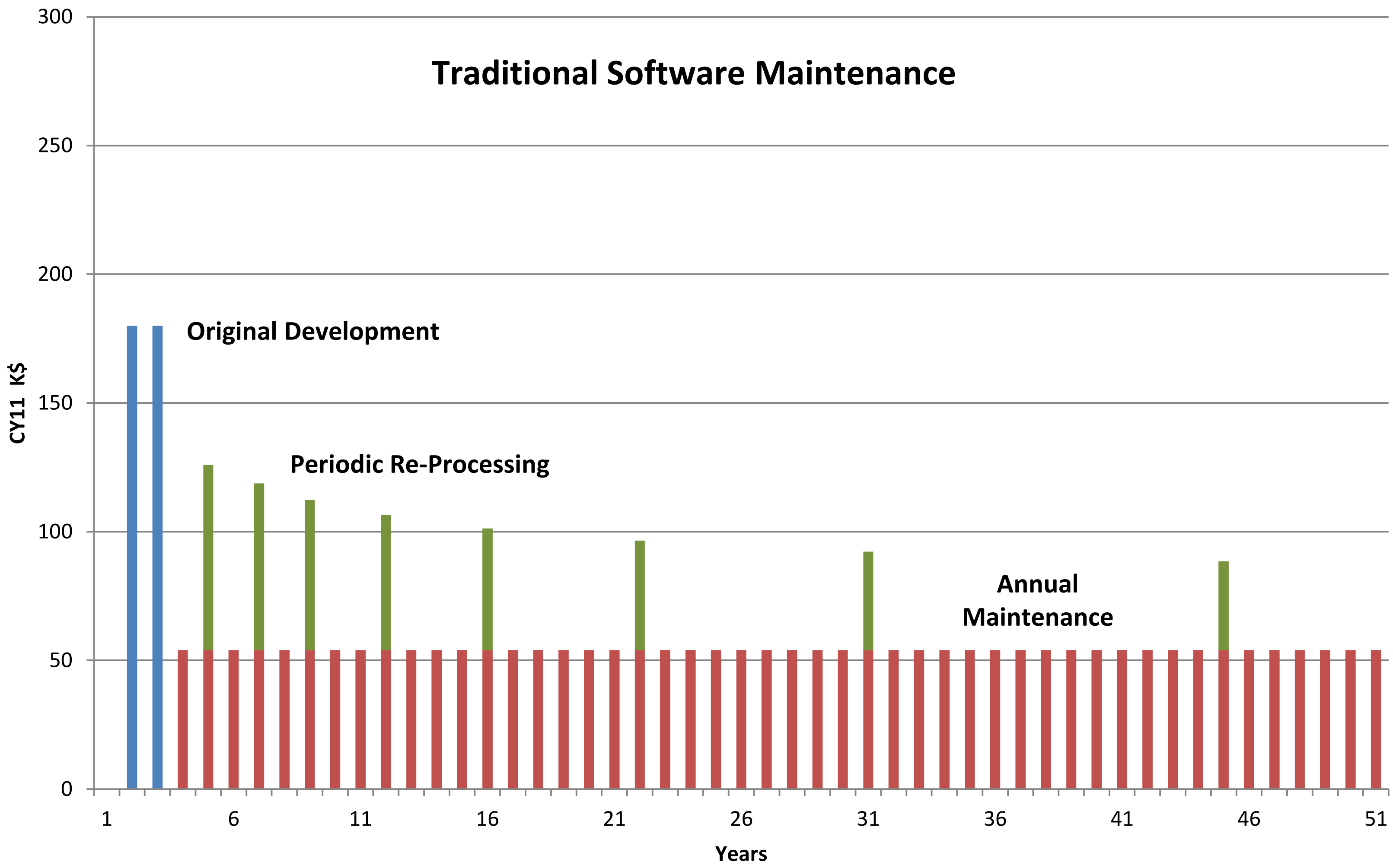
Optimizing Software Maintenance

- Small, persistent changes in *some* Climate Variables require observation over several decades to achieve statistical significance (e.g. mean daily temperature)
- For these variables the long-term cost of software maintenance is expensive and likely becomes an issue
- Would optimizing software maintenance result in meaningful savings over the long-term?
- A spreadsheet model was developed based on experiences from the banking industry

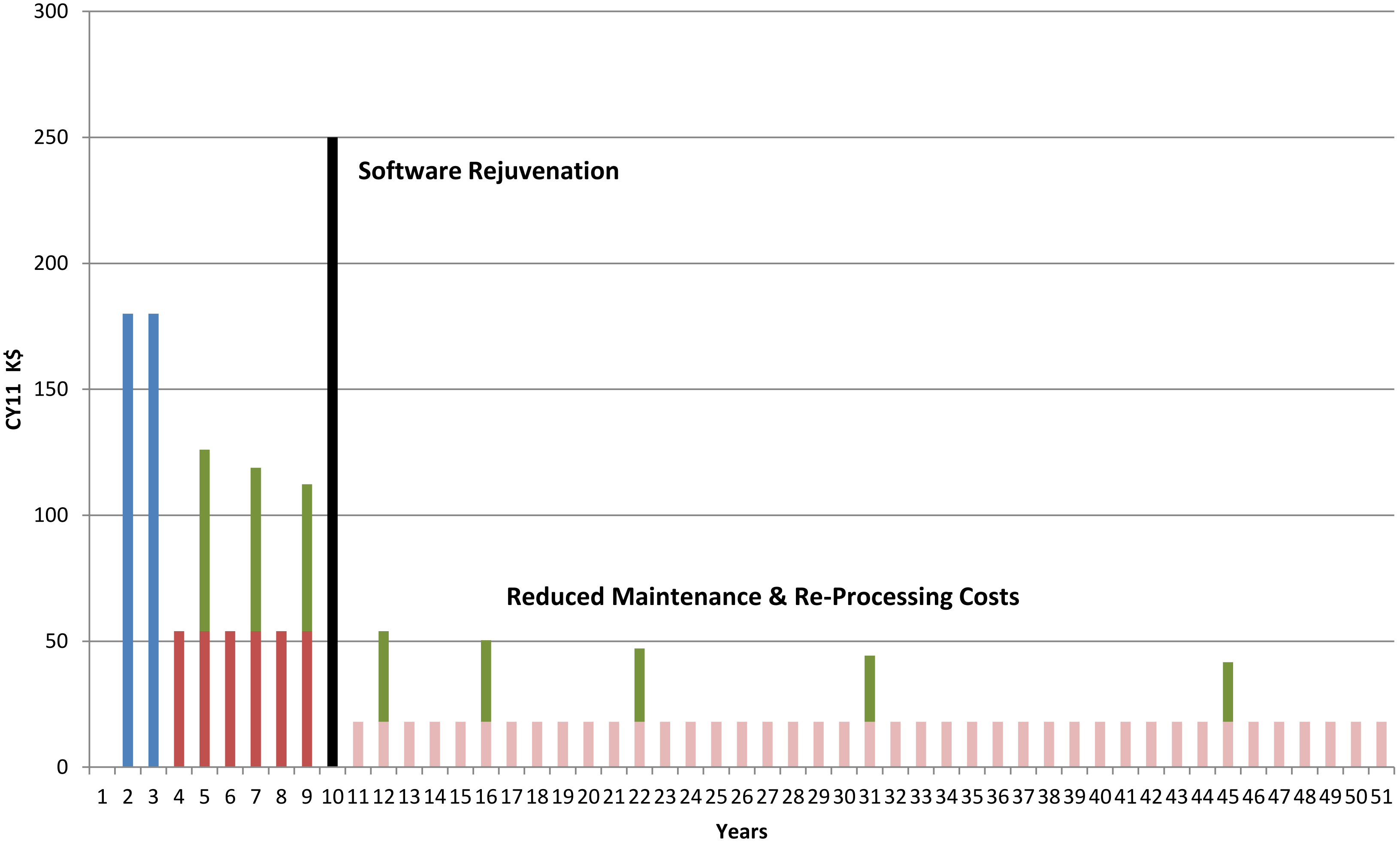
Modeling Software Maintenance



Baseline Solution



Optimized Software Maintenance



Optimizing Software Maintenance

- Our approach involves a prototyping effort to improve the maintenance of a typical CDR algorithm to test the concept
- Draft Coding Standards have been developed that reflect a high degree of software maintainability
- Software static analysis tools are used to characterize the algorithm and identify candidates for improved maintainability
- The task is coordinated with the PI's team
- Achieving good alignment with the Coding Standards potentially represents an important part of the transition from IOC to FOC

Long-term Role of the PI’s Team

- Software Maintenance Team and the PI’s Team form a *complementary* relationship
- PI’s Team continues to develop technical improvements to the algorithm
- Software Maintenance Team proceeds to improve maintainability
- The two teams continually coordinate their activities
- Software Maintenance Team adopts PI’s improvements as they mature and vice versa
- Both teams benefit from the exchange of information and products

Current Prototyping Task

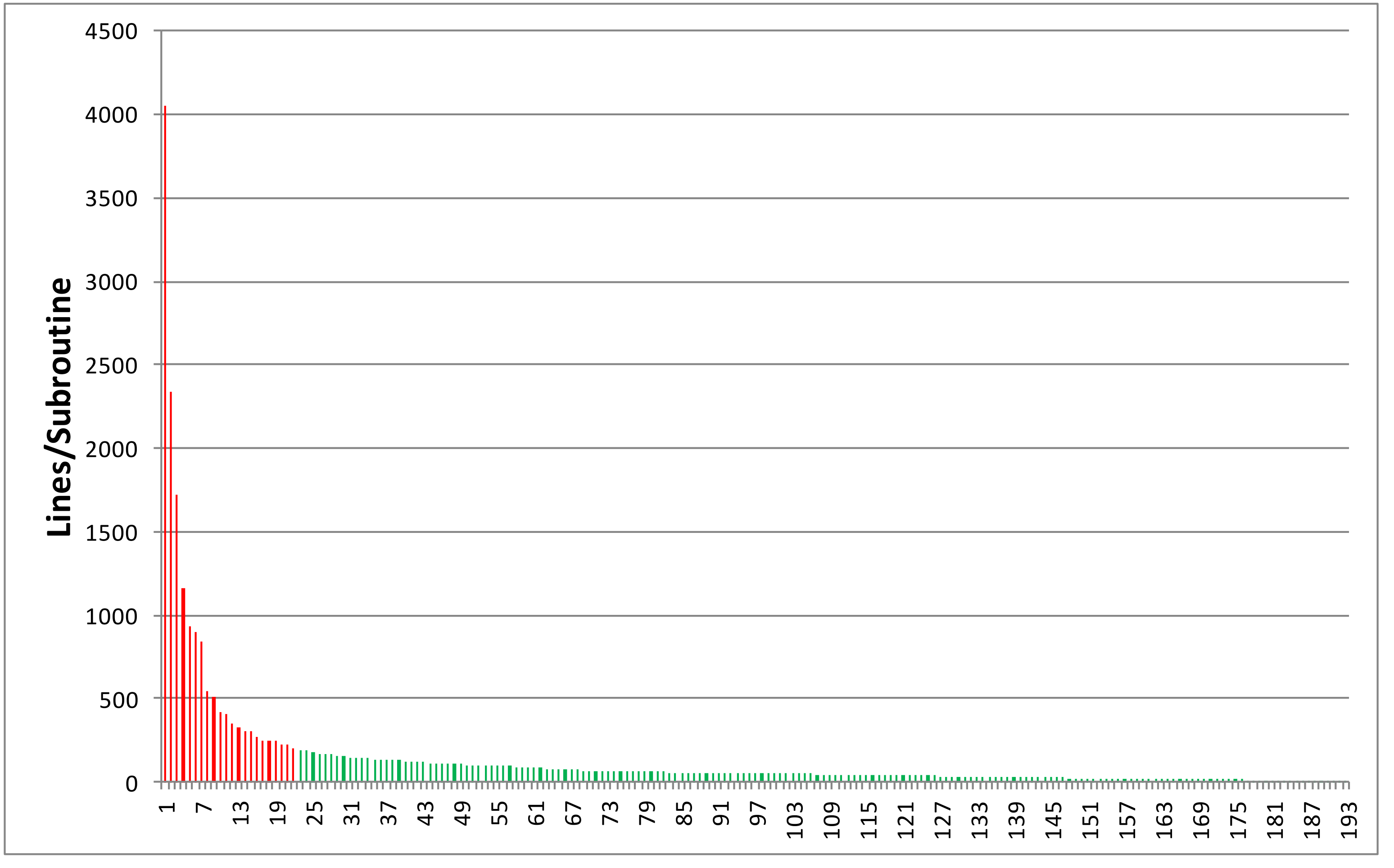
- Xuepeng (Tom) Zhao (NCDC) has developed an Aerosol Optical Thickness Climate Data Record (CDR) using the PATMOS-x Data Processing System
- PATMOS-x generates multiple products (cloud, aerosol, surface and radiometric) on the same grid using a common processing path
- PATMOS-x processes approximately 25 years of data from NOAA's Advanced Very High Resolution Radiometer (AVHRR) flown on the POES spacecraft
- The current task is to optimize the PATMOS-x algorithm for software maintenance that supports the Aerosol Optical Thickness CDR
- Current task started in February 2012 and will finish in April 2013

Initial Results of Static Analysis

- PATMOS-x version used is dated January 11, 2012
- PATMOS-x has 483 subroutines involving 42,875 lines of executable code
- With all output files turned ON (except volcanic ash), PATMOS-x utilizes 193 subroutines involving 26,395 lines of executable code
- The size, cyclomatic complexity, and execution calls have been determined for each active (called) subroutine

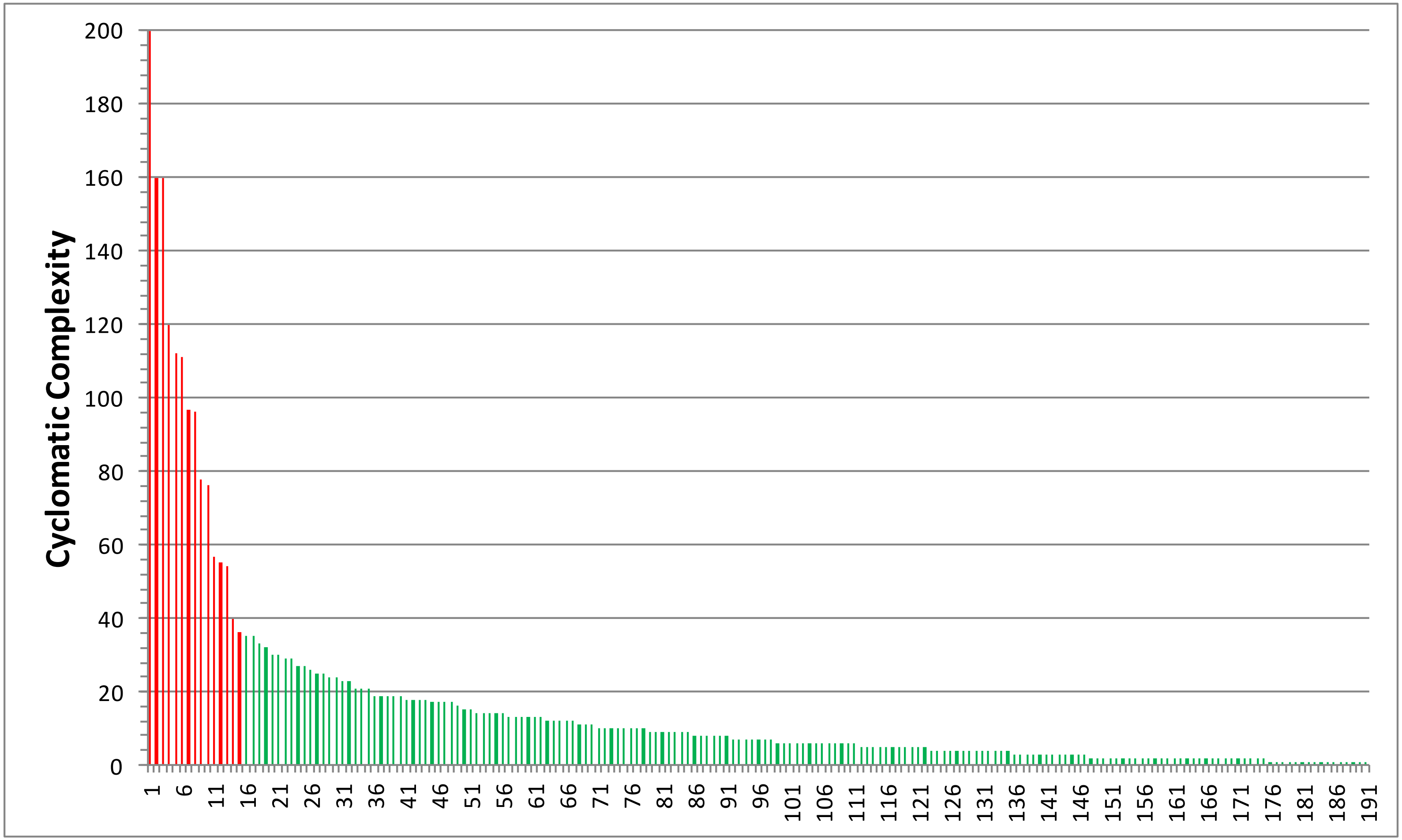
PATMOS Subroutines

Distributed by Size (Lines of Code)

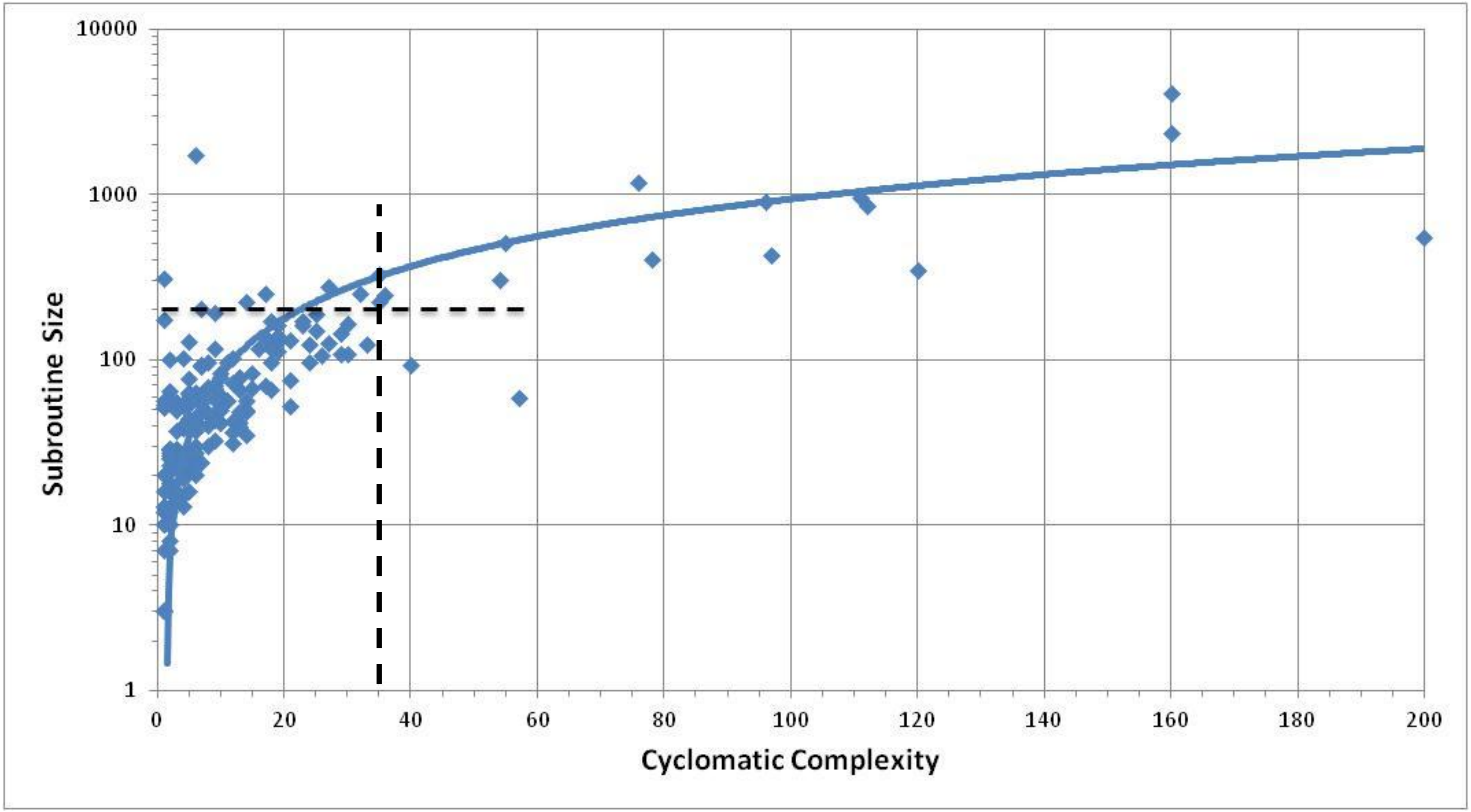


PATMOS Subroutines

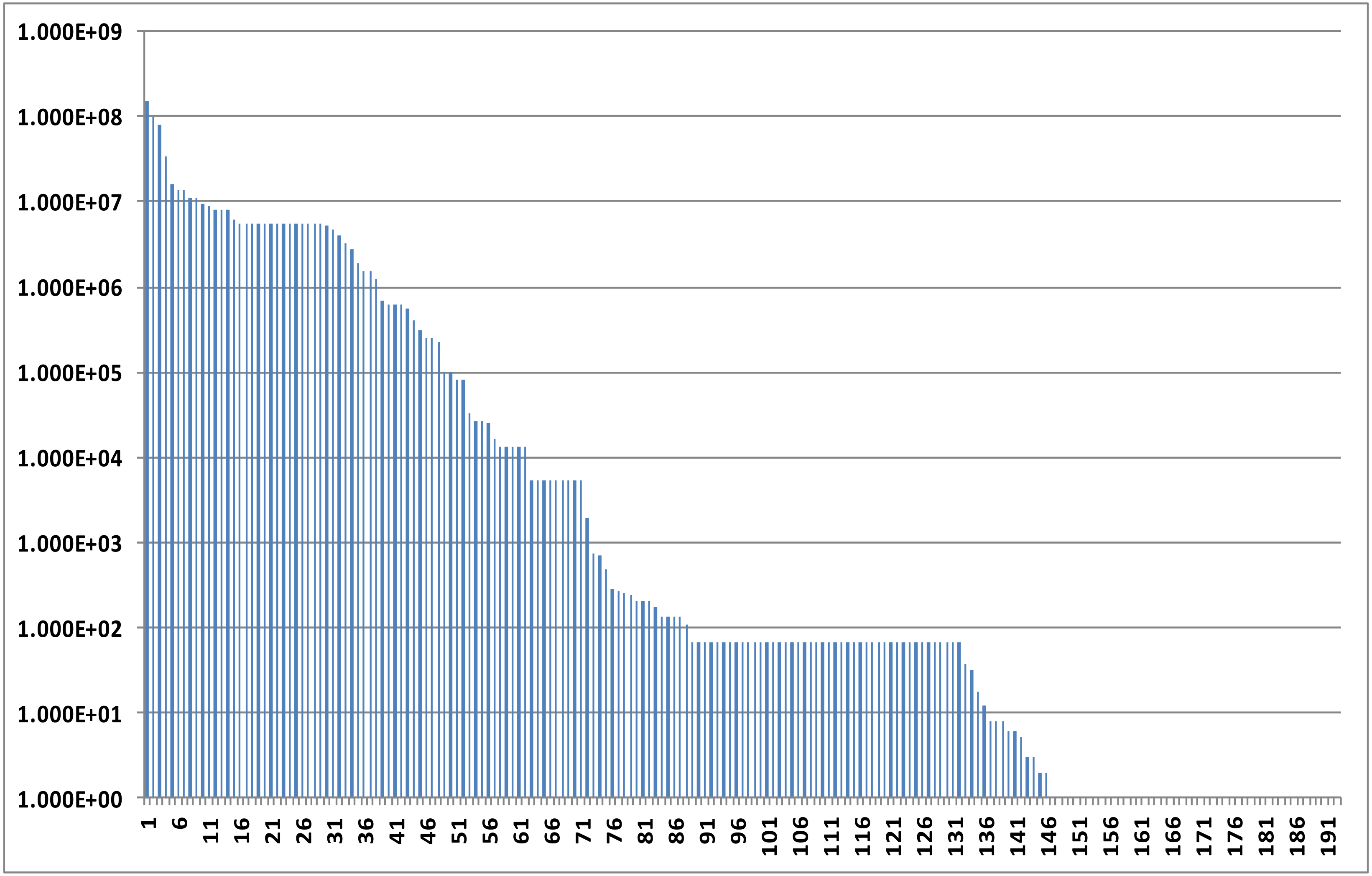
Distributed by Cyclomatic Complexity



PATMOS Subroutines
Size vs Cyclomatic Complexity



PATMOS Subroutines
Execution Calls/Orbit Calculation



Early Results of Static Analysis

PATMOS All Outputs Turned ON (Except Volcanic Ash)								
Subroutines Exceeding Complexity and/or Size								
Subroutine Name			Cyclomatic Complexity	Lines of Code	Execution Calls	Sorting Codes		
Called Subroutines that exceed BOTH Complexity and Size:								
1	325	PIXEL_COMMON:DESTROY_PIXEL_ARRAYS	200	546	1	1	1	2
2	366	LEVEL2_ROUTINES:WRITE_PIXEL_HDF_RECORDS	160	2337	68	1	1	2
3	313	LEVEL2_ROUTINES:DEFINE_HDF_FILE_STRUCTURES	160	4047	1	1	1	2
4	299	AVHRR_CALNAV_ROUTINES:READ_CLAVRXORB_COMMANDLINE_OPTIONS	120	344	1	1	1	2
5	351	COMPILE_ASC_DES	112	837	38	1	1	2
6	356	AWG_CLOUD_HEIGHT:AWG_CLOUD_HEIGHT_ALGORITHM	111	938	68	1	1	2
7	368	NAIVE_BAYESIAN_CLOUD_MASK:CLOUD_MASK_NAIVE_BAYES	97	421	68	1	1	2
8	393	RT_UTILITIES:GET_PIXEL_NWP_RTM	96	895	68	1	1	2
9	367	LEVEL3_ROUTINES:COMPILE_GRIDCELL_ARRAYS	78	404	68	1	1	2
10	324	PIXEL_COMMON:CREATE_PIXEL_ARRAYS	76	1161	1	1	1	2
11	314	LEVEL3_ROUTINES:COMPUTE_GRIDCELL_ARRAYS	55	505	1	1	1	2
12	373	PIXEL_COMMON:RESET_PIXEL_ARRAYS_TO_MISSING	54	299	68	1	1	2
13	449	Conver	36	245	1892710	1	1	2
Called Subroutines that exceed EITHER Complexity or Size:								
14	322	NWP_COMMON:DESTROY_NWP_ARRAYS	57	58	1	1	0	1
15	390	PIXEL_ROUTINES:NORMALIZE_REFLECTANCES	40	93	68	1	0	1
16	355	AWG_CLOUD_DCOMP:AWG_CLOUD_DCOMP_ALGORITHM	35	327	68	0	1	1
17	320	NCEP_REANALYSIS:READ_NCEP_REANALYSIS_DATA	35	220	1	0	1	1
18	424	AVHRR_CALNAV_ROUTINES:COMPUTE_NEW_THERM_CAL_COEF	32	248	13514	0	1	1
19	465	RT_UTILITIES:INTERP_RTM_KNOWING_Z_NEW	27	272	5519864	0	1	1
20	387	PIXEL_ROUTINES:COMPUTE_SPATIAL_UNIFORMITY	17	247	68	0	1	1
21	470	AWG_CLOUD_HEIGHT:COMPUTE_FORWARD_MODEL_AND_KERNEL	14	224	8038080	0	1	1
22	460	AWG_CLOUD_HEIGHT:SET_CLEAR_SKY_COVARIANCE_TERMS	7	201	5519864	0	1	1
23	305	CELL_HDF_ROUTINES_1:WRITE_GRIDCELL_DATA	6	1724	1	0	1	1
24	315	LEVEL3_ROUTINES:CREATE_GRIDCELL_ARRAYS	1	308	1	0	1	1

Next Steps

- Identify the PATMOS-x subroutines required to support the Aerosol Optical Thickness algorithm
- Examine these subroutines for variances from the CDRP Coding Standards
- Identify the software changes required to meet the Coding Standards
- Prioritize these changes on the basis of the most cost-effective contributions to long-term software maintenance
- Consider architectural changes to subroutines to reduce their size and to run more efficiently
- Prioritize all of these proposed changes into a single list
- Implementing these changes becomes the FY13 effort