Cost Analysis Division Jet Propulsion Laboratory NC State University

NASA Software Cost Estimation Model: An Analogy Based Estimation Method

Jairus Hihn
Jet Propulsion Laboratory, California Institute of Technology

Tim Menzies
George Mathews
North Carolina State University

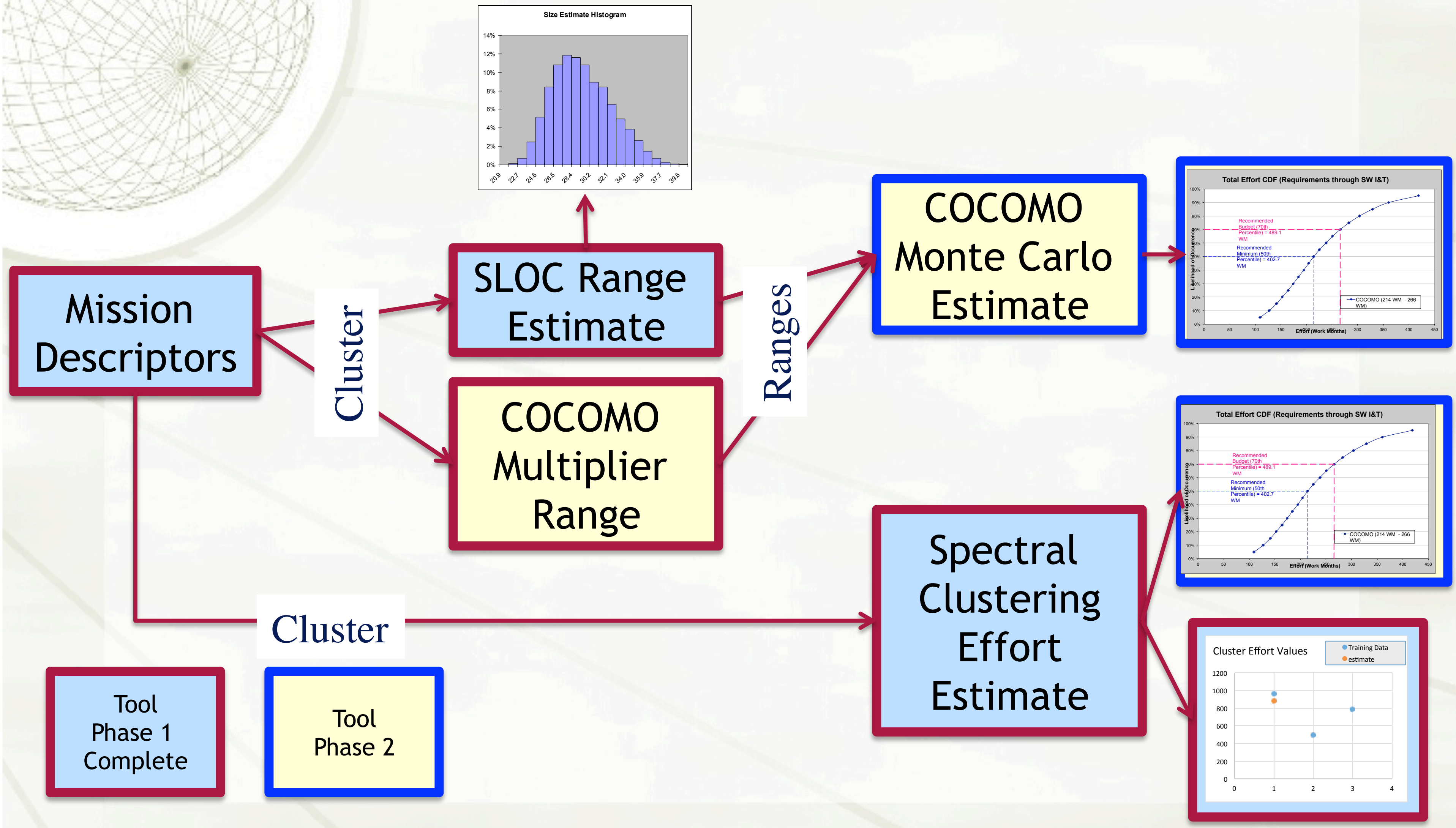
James Johnson
National Aeronautics and Space Administration

2015 NASA Cost Analysis Symposium
NASA AMES Research Center

Introduction

- In this talk we will provide an overview of our research and results in the development of an NASA SW Cost Model, which is Analogy Cost Model using data mining algorithms
  - Talk will emphasize methodology
  - TOOL Demo and mini tutorial is 8:30 Thursday in rm 105/106
- The purpose of the model is to
  - Supplement current estimation capabilities
  - *Be effective in the very early lifecycle when our knowledge is fuzzy*
    - **uses high level systems information (Symbolic Data)**
  - *Be usable by Cost Estimators, Software Engineers and Systems Engineers*
- *Methodology handles*
  - small sample sizes
  - noisy and sparse data

Model Architecture



Data Items

Data Item	Number of Projects
Total development effort in work months	28
Logical Lines of code (LOC)	
○ Delivered LOC	36
○ Equivalent LOC	36
○ Inherited LOC (Reused plus Modified reused lines)	36
○ Reused LOC (0-10% modified)	36
COCOMO model inputs (See Appendix A for the parameter definitions) - Translated from CADRE which has SEER model inputs because the SEER data items are very sparse in CADRe	19
System parameters (See Appendix B parameter definitions)	
○ Mission Type (deep-space, earth-moon, rover-lander, observatory)	39
○ Multiple element (probe, etc.)	39
○ Number of instruments	39
○ Number of deployables	39
○ Flight Computer Redundancy (Dual Warm, Dual Cold, Single String)	39
○ Software Reuse (Low, Medium, High)	36
○ Software Size (Small, Medium, Large, Very Large)	36

For detailed description of the Data see Appendix B in the paper

System Descriptor Details (Example)

System Descriptors

Mission Type	Values	Description	Example
	Earth/Lunar Orbiter	Robotic spacecraft that orbit the earth or moon conducting science measurments. These spacecraft are very similar if not identical to the many commercial satellites used for communication as well as many military satelites. They often can have have high heritage and even use production line buses from industry.	Aqua
	Telecomm Sat	Earth orbiters that support very high bandwidth and designed for very long life.	TDRS
	Observatory	Observatories are space based telescopes that support space based astronomy across a wide set of frequencies. They can be earth orbiters or earth trailing at the various lagrange points created by the garvoty fields of the earth, sun and moon.	Hubble
	Deep Space	Any robotic sapcecarft that goes beyind the moons orbit. So this category includes any misison whose destination is a planet, planetoids, any planetary satelite, comet, asteroid or the sun. These misison can be orbiters or flybys or a mixture of both.	Deep Impact
	Static Lander	A robotic spacecraft that does its science in-situ or from the surface of a soplar system body. It does not move from its original location.	Phoenix
	Rover	A robotic spacecraft that does its science in-situ or from the surface of a solar system body and has the ability to move on the surface. To date all rovers have wheels but in the future they may crawl, walk or hop.	Mars Exploration Rover (MER)

- Complete list is in the backup slides

Data Sources

- Where the data came from
  - CADRe
  - NASA 93 - Historical NASA data originally collected for ISS (1985-1990) and extended for NASA IV&V (2004-2007)
  - Contributed Center level data
  - NASA software inventory
  - Project websites and other sources for system level information if not available in CADRe

Data Summary – Selected Items

Effort, Lines of Code and Productivity by Mission Type

Mission Type	# Records	EFFORT (months)		Logical Delivered LOC		Logical Equivalent LOC		Productivity (Logical Del/month)		Productivity (Logical Equiv/month)	
		Median	S.D	Median	S.D	Median	S.D	Median	S.D	Median	S.D
Earth/Lunar Orbiter	19	579	418	92,050	40,104	56,940	41,010	265	1,366	150	711
Observatory	5	492	1,054	107,100	59,143	76,800	61,411	74	977	75	698
Deep Space	11	670	866	121,000	54,191	122,000	47,034	179	114	149	96
In Situ	4	1,408	551	246,700	164,844	199,500	220,139	215	80	178	93

Number of Deployables and Instruments by Mission Type

Mission Type	Deployables		Instruments	
	Median	Range	Median	Range
Earth/Lunar Orbiter	2	0-7	3	1-10
Observatory	2	0-4	4	1-6
Deep Space	2	1-8	3	2-12
In Situ	7	3-10	5	3-10

For a complete summary of all data see the paper.

Methodology

- Two teams were formed using different methods
- Team 1 - JPL
  - used standard statistical methods (t-test, f-test, etc.)
  - Calibrated COCOMO II
  - Linear and Ln-Linear regressions
- Team 2 - NC State, used data mining algorithms
  - Validated based on
    - leave one out validation
    - Magnitude of Relative Error (MRE) median and distribution.
  - The models/estimation methods evaluated are:
    - COCOMO II -(Out of the box)
    - COCONUT - a tuning rig for COCOMO II
    - Knn_1 - a K-nearest neighbor model
    - delLOC – a regression of total development effort on LOC
    - MED_MISSION Median effort by mission types
    - PEEKING2/PEEKER - constructs clusters of projects using spectral clustering algorithms

What We Learned - 1

- Last year we reported on findings/results in developing a prototype model
  - *Median is better measure of central tendency than the mean for much of our data*
    - *Because distributions are skewed*
  - *Should use Magnitude of Relative Error (MRE) metrics to supplement standard statistics as sometimes they are misleading*
    - *We recently experienced this directly as we started exploring the Instrument Flight Software Cost Model*
  - *When use clustering algorithms one will find some counter intuitive clustering*

What We Learned - 2

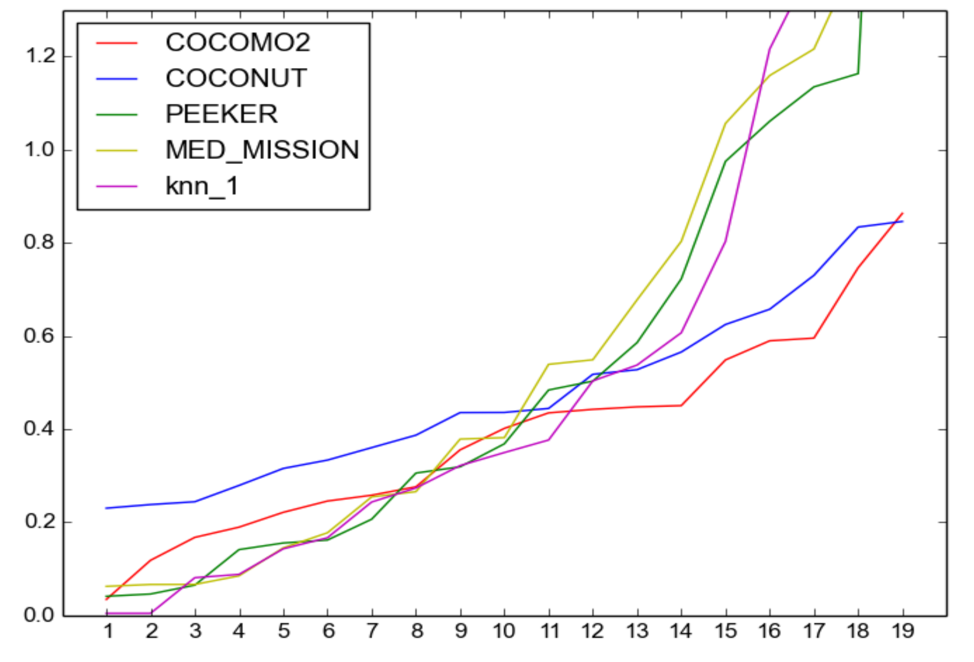
- It was not possible to derive a basic general purpose effort = f(LOC) model even though we had 26 records with both LOC and effort.
- We have been able to do this for decades on in-house JPL data
  - but when combining the data with data from other centers and from contractors as reported in the CADRe the new records appear to have added more noise than information
- The models we were able to derive violated the laws of logic
    - indicate all we need to know is the new LOC even if there is large percentage of reused code
    - or that we can actually make money by reusing code, not just reduce costs.
- Another interesting result was that the out of the box COCOMO performed better than a locally calibrated version based on comparing MRE
  - We saw this result in 2002
    - Karen Lum, John Powell, Jairus Hihn, *Validation of Spacecraft Software Cost Estimation Models for Flight and Ground Systems*, **Proceedings of the 24th Annual Conference of the International Society of Parametric Analysts (ISPA), 21-24 May, 2002,** San Diego, CA
  - Method 2 results also corroborated both results

What We Learned - 3

Estimation Model	Median MRE (MMRE)	25th Percentile	75th Percentile
knn_1 (Nearest Neighbor)	32%	14%	80%
PEEKING2 (Spectral Clustering)	32%	16%	97%
COCOMO2	36%	22%	55%
Mission Type Summary Table	38%	14%	106%
COCONUT	44%	32%	62%

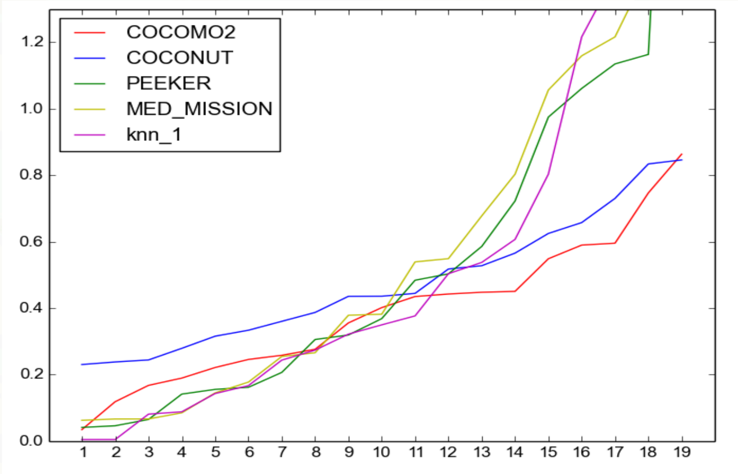
- The non-parametric models have a slightly lower MMRE

- Median effort by mission type is in the running based on MMRE
- The COCOMO models handle outliers better
- Local calibration does not improve performance
- When non-parametric models are inaccurate they tend to be extremely inaccurate



What We Learned - 4

Estimation Model	Median MRE (MMRE)	25th Percentile	75th Percentile
knn_1 (Nearest Neighbor)	32%	14%	80%
PEEKING2 (Spectral Clustering)	32%	16%	97%
COCOMO2	36%	22%	55%
Mission Type Summary Table	38%	14%	106%
COCONUT	44%	32%	62%



Conclusion

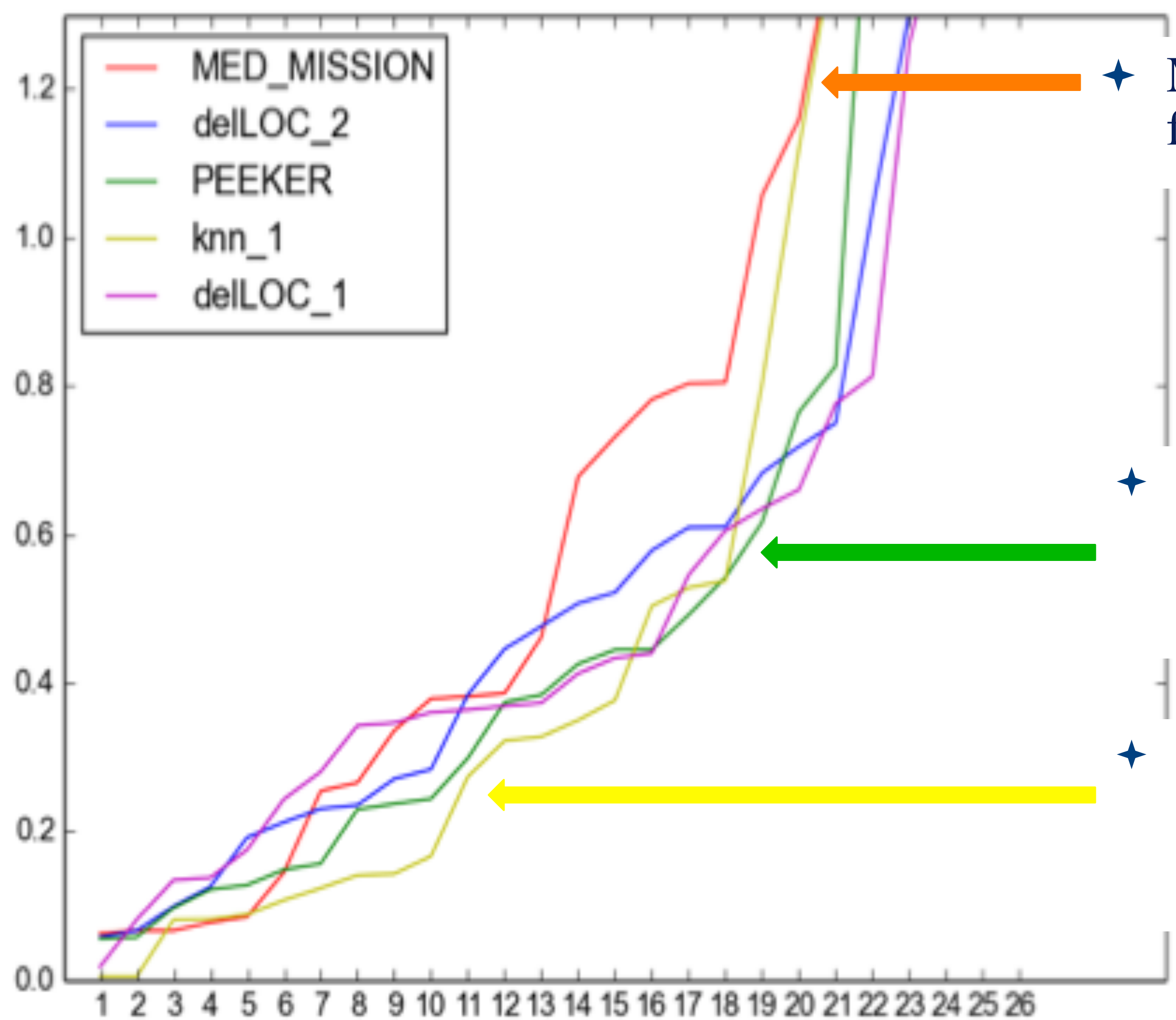
- ✓ Use parametric model if have sufficient information
- ✓ Use Non-parametric models when do not have sufficient information
  - Based on overall MRE performance is good especially for worst cases
  - MMRE only slightly worse
- ✓ For NASA FSW can use COCOMO II out of the box

Methodology 2 Results Part 2b

Estimation Model	Median MRE (MMRE)	25th Percentile	75th Percentile
knn_1 (Nearest Neighbor)	33%	12%	112%
LSR on LOC new	37%	28%	66%
PEEKING2 (Spectral Clustering)	38%	16%	76%
Mission Type Summary Table	46%	25%	116%
LSR on LOC new and reused	48%	23%	72%

- MMRE indicates Non-parametric models perform as well or better than regression methods.
- The LSR models are rejected on first principles
  - The LSR results reconfirmed the Method 1 results by producing similar illogical results that violate common sense.
- Based on MMRE Nearest Neighbor appear to outperform Spectral Clustering.
  - However, has much more significant outliers than Spectral Clustering.

Methodology 2 Results Part 2c



- Median Mission Type Model falls apart
- Spectral Clustering has its day and blows up more slowly than Nearest Neighbor
- Nearest Neighbor does best when within +/- 40% but then becomes one of the worst estimators

Conclusions

- There are a variety of models whose performance are hard to distinguish (given currently available data) but some models are better than others
- If one has sufficient data to run COCOMO or a comparable parametric model then the best model is the parametric model
- When insufficient information exists then a model using only system parameters can be used to estimate software costs with relatively small reduction in accuracy. The main weakness is the possibility of occasional very large estimation errors which the parametric model does not exhibit.
- A major strength of the nearest neighbor and spectral clustering methods is the ability to work with a combination of symbolic and numerical data
- While a nearest neighbor model performs as well or better as spectral clustering based on MMRE, spectral clustering handles outliers better and provides a structured model with more capability
- Contact me if you want a copy of the Paper (Presented at ICEAA) that has a detailed description of the results described here

TOOL Demo and mini-tutorial is 8:30 Thursday in rm 105/106

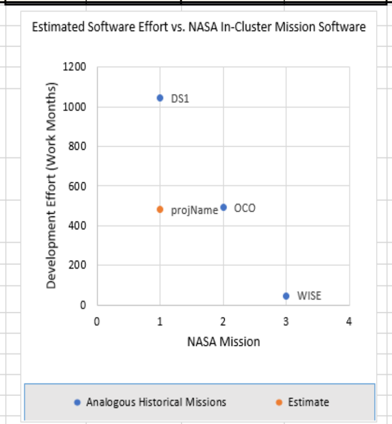
STEP 1) Enter data for the SW project you would like to estimate	Name: **Must enter name	projName
	Software Size Category:	s
	Inheritance:	Very Low to None
	Mission Type:	Deep Space
	Secondary Element:	Impactor/ Probe
	Number of Instruments:	0
	Flight Comp. Redundancy:	Single String
	Total Deployables:	5
STEP 2) Once you have entered data, press the "Add Inputs" button.	Prepare to wait ~ 10 seconds for the model to run!	
STEP 2) Run Estimate		
	Effort Estimate:	481.13
STEP 3) Reset tool before calculating new estimate		Reset Tool

TRAINING CLUSTER DATA SUMMARY

Effort Values		Equivalent Lines of code (KLOC)	
Minimum:	47	Minimum:	76.8
Maximum	1042.8	Maximum:	160
Median:	492	Median:	120.3

Click "Restart" if an error occurs

Restart



TRAINING CLUSTER MEMBERS	Name	Software Size Category	Equivalent LOC	Inheritance	Mission Type	Secondary Element	Number of Instruments	Flight Comp. Redundancy	Total Deployables	Actual Effort
	DS1	l	160	medium	deep space	none	2	single string	1	1042.8
	OCO	l	120.3	high	earth/lunar	none	1	single string	1	492
	WISE	m	76.8		observatory	none	1	single string	1	47