

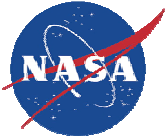


Illustrative Example of a Function Point Analysis for the NASA Crew Exploration Vehicle Guidance, Navigation & Control Flight Software

Robert Georgi /Booz | Allen | Hamilton
Terry Vogt /Booz | Allen | Hamilton
Orion Strategic Assessment Team

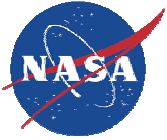
National Aeronautics & Space Administration
Lyndon B. Johnson Space Center
Houston, TX 77058

April 9, 2008



Agenda

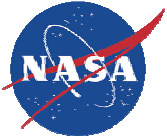
- Task Background
- Function Points Concepts
- GN&C Function Point Analysis
- GN&C SEER-SEM Estimate



Strategic Assessment Team direction



- The Strategic Assessment Team under the Orion Project Planning & Control Office (PP&C) has the charter to conduct cost estimates and schedule assessments on strategic issues facing the project
- Booz Allen supports the team through the Financial Analysis and Business Solutions contract (FABS)
- In August we were directed to prepare an independent cost estimate of the Orion Flight Software
- Decided upon developing a size estimate using function point analysis
- At the direction of the Vehicle Integration Office (VIO) Integrated Avionics & Software manager, the subsystem we examined was GNC



Software development is a significant schedule and cost risk for the project



- Currently the #6 risk on the project (1115 – CEV software development cycle)¹
- Recent studies have shown that only 35 percent of software development projects are completed successfully within the estimated schedule and budget²
- This terrible track record is due, in large part, to initial project estimates that are overly optimistic and inaccurate
- Since the cost of the development effort is directly proportional to the size of the development, estimating the size well is key
 - You cannot manage what you do not understand, you cannot understand what you cannot measure

(1) Orion Top Risks 7-17-07, Mark Kirasich, Deputy Project Manager

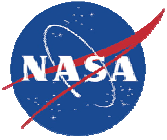
(2) 2006 Standish Group Chaos Report www.sdtimes.com/article/story-20070301-01.html



GNC



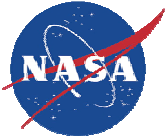
- The Orion Flight Software is composed of Computer Software Configuration Items (CSCI)
- The GNC subsystem comprises three of the CSCI:
 - Guidance
 - Navigation
 - Control
- Chosen for the analysis because it is the most critical component of the Flight Software



GNC Description and Functions

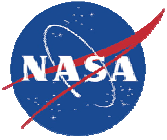


- The CEV Spacecraft GN&C performs the following functions during different modes and phases of the CEV missions:
 - Spacecraft navigation,
 - Targeting,
 - Guidance,
 - Controls,
 - Rendezvous,
 - Proximity operations,
 - Docking,
 - Undocking,
 - Solar panel and high gain antenna pointing.
- Other GNC functions include: house keeping operations such as health and status information generation and checking for the GNC sensors and effectors; performing GNC Fault Detection, Isolation and Recovery (FDIR) of GNC functionality; and maintaining flight parameter databases required for the GNC functions



Agenda

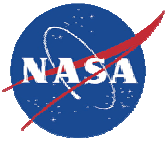
- Task Background
- Function Points Concepts
- GN&C Function Point Analysis
- GN&C SEER-SEM Estimate



Objectives of Counting Function Points



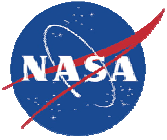
- Objectives
 - To measure functionality that the user requests and receives
 - To measure software development and maintenance independently of technology used for implementation
 - To measure software by quantifying the functionality the software provides to the user based primarily on logical design
- The process of counting function points is intended to be:
 - Simple enough to minimize the overhead of the measurement process
 - A consistent measure among various projects and organizations



What is a function point?



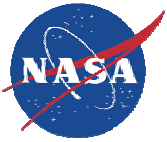
- Function Point (FP)
 - A measure which represents the functional size of application software
 - A standardized metric that describes a unit of work product suitable for quantifying software that is based on what the end-user requests and receives
- Function Point Analysis (FPA)
 - A standard method for measuring software development and maintenance from the customer's point of view
- Function Point Count (FPC)
 - The function point measurement of a particular application or project



Function points compared to other conceptual measures

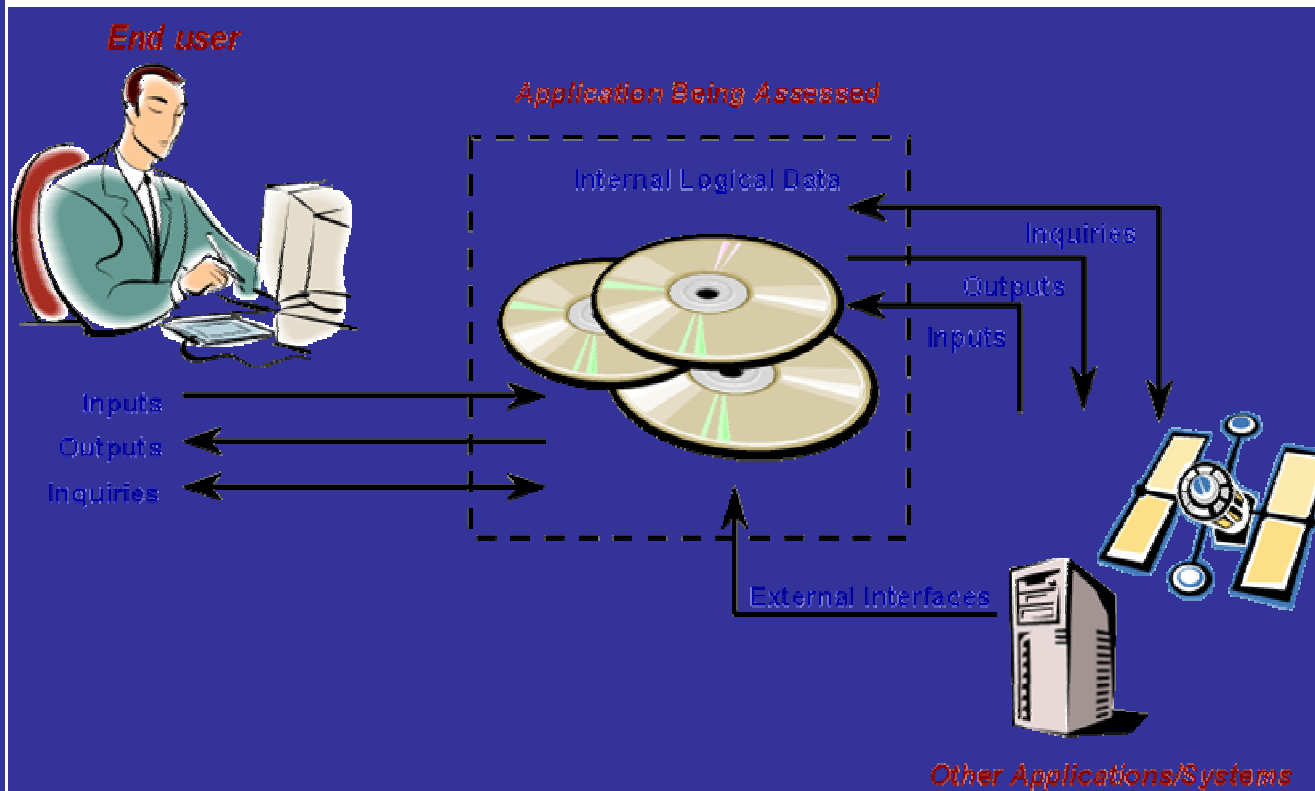


- Length: Feet, inches, yards, meters, millimeters, miles, kilometers, etc
- Time: Seconds, minutes, days, weeks, months, years, decades, centuries, millennia
- Heat Energy: Degrees of Temperature: Fahrenheit, Celsius (Centigrade), Absolute (Kelvin)
- Monetary Value: Dollar, Euro, Yen, Yuan, etc
- All are man-made concepts we accept and routinely use as standards of measurement
- Function Point: A conceptual measure of software size based on functional requirements
 - Basis is the measured proportions of effort required to produce types of functionality
 - Derived through multiple regression analyses applied to actual project results



Function Point Components

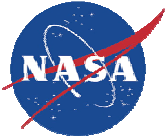
Function points measure software size based on functionality requested by and provided to the end user



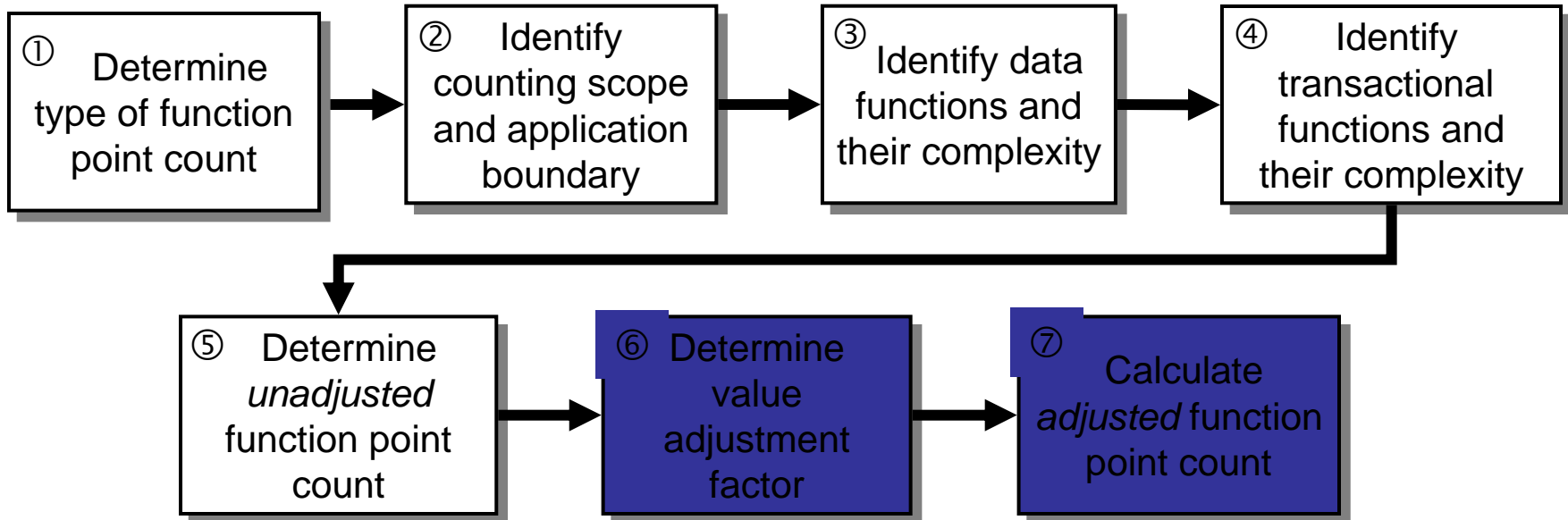
Function point counting resources

- ✓ User/analyst interviews
- ✓ Requirements documents
- ✓ Design documents
- ✓ Data dictionaries
- ✓ Use cases
- ✓ User guides
- ✓ Screen captures
- ✓ Actual software
- ✓ Entity-relationship models
- ✓ Semantic object models

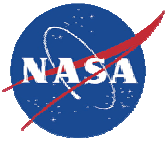
Function points represent **logical** size, as opposed to **physical** size (like SLOC or objects)



Function Point Counting Process Steps



Note: Our sizing process omits the last two steps of the official IFPUG function point analysis and ends with unadjusted function points, which are input to the SEER-SEM estimation tool along with other performance parameter values.



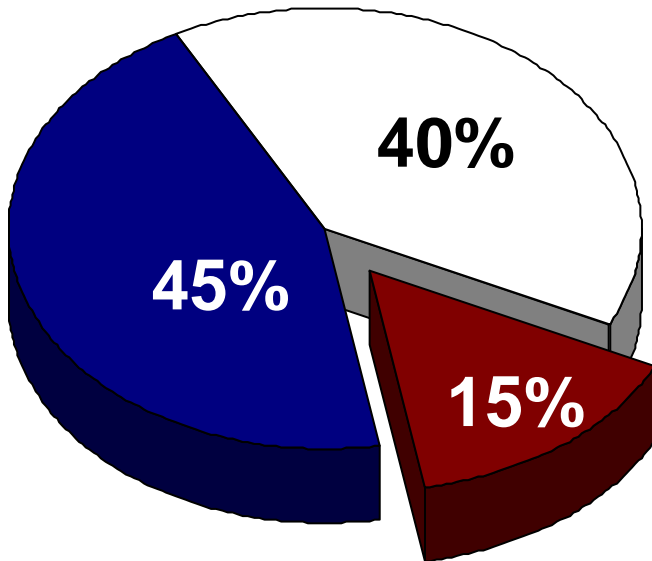
Function Point Analysis as Part of Best Practices



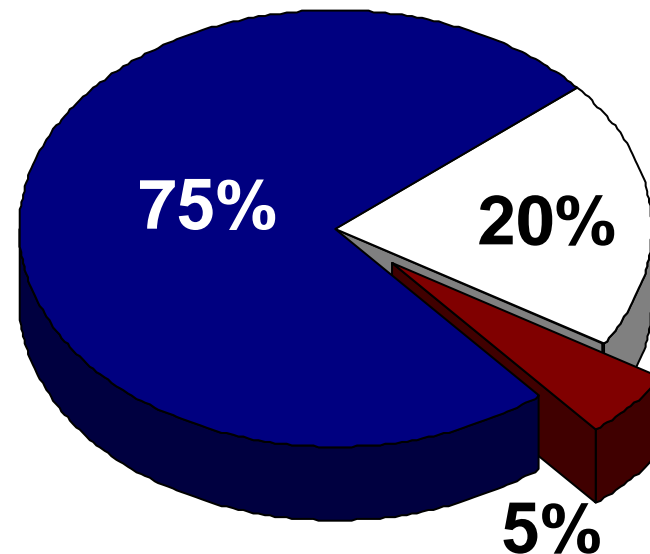
- Using FPA to help manage a software project can improve the probability of completing a project on time and within budget

Project success...

Without FPA



With FPA



■ On-time □ Late ■ Cancelled

Source: Software Productivity Research



Why Use Function Points?



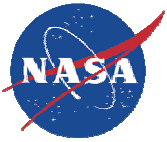
- FPA provides a consistent, documentable, repeatable measurement methodology
- Because it is linked directly to system requirements and functionality, FPA puts size analysis into terms that a client or end user can understand
 - Function points can help with communications between the end user community and the developer
 - Assists with requirements management - functional size traceable throughout entire life cycle
 - Applicable from earliest requirements stage and throughout full life-cycle
- Technology, platform, and language independent
- Advantages over lines of code: not dependent on engineering estimates, better metrics
- Provides quantitative basis for earned value management
- Can reveal gaps in functional requirements e.g. data stores without related transactions



Alternative Software Size Measure: Lines of Code



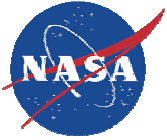
- Source Lines Of Code (SLOC)
 - A purely physical measure – measures code used to provide functionality
- Pros
 - Easy to come up with a number
 - Plenty of historical data available
 - Supported by most cost estimating tools



Alternative Software Size Measure: Lines of Code



- Cons
 - No standards - huge divergence in measured size of same material by different individuals
 - Methodologies for coding software and for counting SLOC differ between organizations
 - Cross-language estimates are inconsistent
 - Conversion factors are unreliable, code size varies widely for equivalent functionality
 - Rewards profligate design, penalizes tight design
 - No incentive for developers to reduce lines of code or associated effort or cost
 - Measures components, not completed products
 - Measures what is used to make a thing, not what is delivered
 - Measurement before coding phase is guesswork, estimates are very doubtful
 - Irrelevant to some modern development environments
 - Graphical design environments minimize use of code, wild swings in code versus functionality



SLOC (Source Lines Of Code) metrics can be misleading



Evaluating the economics of a project:
One software project, coded in 3 different languages

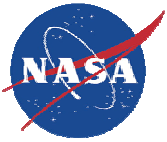
Language (notional)	Assembly	Ada 83	C++
SLOC Production (per staff month)	556	350	333
Cost per SLOC	\$30	\$47	\$50
Total SLOC in project scope			
	10,000	3,500	2,500
Total Effort (staff-months)	18.0	10.0	7.5
Total Cost (same \$ / staff month)	\$300K	\$166K	\$125K

"Best" metrics

Highest effort and cost

This is known as the Productivity Paradox

Source: Software Productivity Research



Function point metrics are comparable and logical across projects, platforms, and languages



Evaluating the economics of a project:
The same software project, coded in 3 different languages

Language (notional)	Assembly	Ada 83	C++	
FP Production (per staff month)	2.78	5.00	6.67	<i>Best metrics</i>
Cost per FP	\$6,000	\$3,320	\$2,500	
Total Function Points in project scope	50	50	50	
Total Effort (staff-months)	18.0	10.0	7.5	<i>Lowest effort and cost</i>
Total Cost (same \$ / staff month)	\$300K	\$166K	\$125K	

Function point metrics more accurately portray project economics

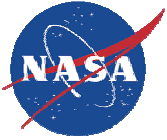
Source: Software Productivity Research
 Source: Software Productivity Research



Function Point Usage and Support



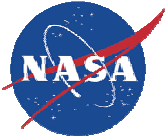
- Representative US users of Function Point Analysis include Accenture, Allied-Signal, Allstate, Dayton Hudson, DFAS, DHS, DOJ , Exelon, General Motors, IBM, IRS, JHUAPL, US Air Force, US Army, US Navy, Verizon
- Function points have been used to size and estimates numerous applications and projects in government, DoD, aerospace and embedded processes. Examples include:
 - GPS III next generation satellite system replacement for current GPS
 - Factory automation processes (Boeing, Daimler-Benz)
 - Communication systems (AT&T)
 - Avionics systems



Function Point Usage and Support

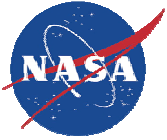


- International Function Point Users Group (IFPUG) serves over 1,200 members in more than 30 countries. The members come from every major industry segment including aerospace, automotive, banking, insurance, government, manufacturing, retail, and telecom
- IFPUG certifies individual proficiency in function point analysis by rigorous, recurrent testing: CFPS designation, which must be individually renewed every three years
- Function Point Analysis is supported by a number of tools designed specifically for documenting and reporting function point measurement
- Function point counts are directly accepted as input to major software cost estimating tools including SEER-SEM



Agenda

- Task Background
- Function Points Concepts
- GN&C Function Point Analysis
- GN&C SEER-SEM Estimate



Process



- GNC Subsystem Estimation Process
 - Used the subsystem specification developed a concept of the software in lieu of actual Software Requirement Specifications (SRS)
 - Created diagrams for each functional area
 - Functional areas are equivalent to Computer Software Components (CSC)
 - Used function point analysis to determine logical size
 - individual analysis items typed & rated and all contributions summarized
 - SEER-SEM estimation tool
 - Takes function point size as input parameter
 - Many other parameter values account for software type, development environment, etc.



Assumptions



- Assumptions
 - Requirements are described, not specified, at this time
 - Many TBDs and choices will eventually change the size of what is delivered
 - Estimate needs to be refined and updated as more information becomes available
 - This analysis, however, has established a template which could permit a quick turnaround once SRS are available
 - The initial cost estimate is for both Block 1 and Block 2 design implementation, but it does not include maintenance cost

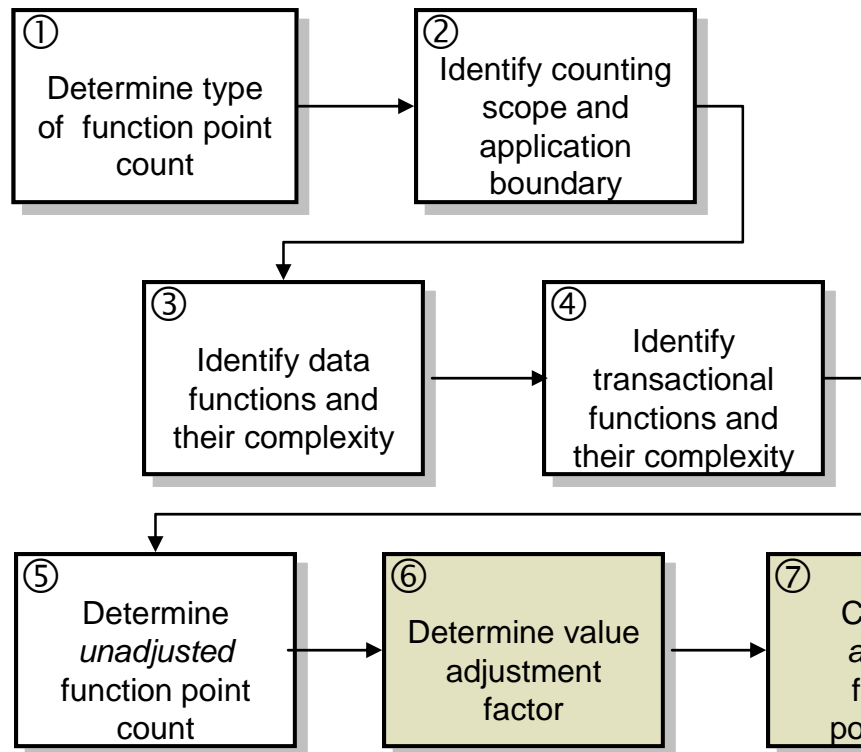


Sizing Process Steps



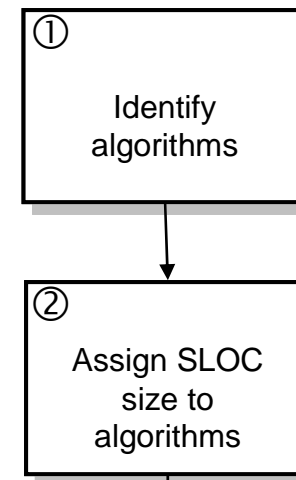
Data stores and I/O Sizing

Note: Our function point sizing process omits the steps 6 & 7 of the official IFPUG function point analysis and ends with unadjusted function points.



Algorithm Sizing

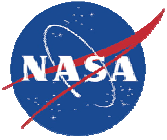
Algorithms are currently sized in SLOC with placeholder values. Later actual values for reused algorithm code will be used where appropriate.



Unadjusted function points and algorithm SLOC sizes are input to the SEER-SEM estimation tool along with other performance parameter values.



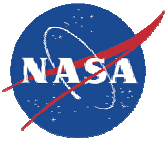
Each functional area of GN&C is separately diagrammed to clearly show related functionality and enable the illustration of individual data flows and algorithms.



List of GN&C Function Point Analysis Diagrams



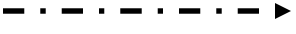
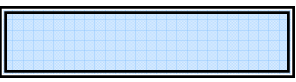

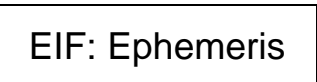




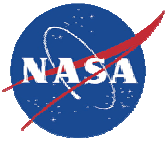
- 1A Navigation – Read Sensors
- 1B Navigation – Determine Absolute Position
- 1C Navigation – Determine Relative Position
- 2A Guidance – Compute Desired Destination
- 2B Guidance – Compute Guidance Solutions
- 3A Control – Rotational Movement
- 3B Control – Translational Movement
- 3C Control – Launch Abort System
- 4A GN&C Manager – Commands and Uploads
- 4A GN&C Manager – Perform Subsystem Test



Legend



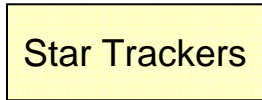
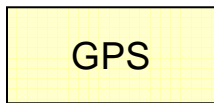
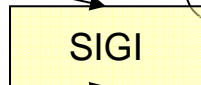
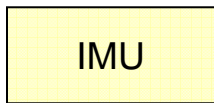
-  Data Flows within system, or Inputs to system, or Outputs from system
-  Data read by system from other systems
-  Data read from system by other systems
-  Application Boundary
-  ILF: logical data store to which the application has Read/Write access
-  EIF: logical data store to which the application has Read Only access
-  Application external to GN&C
-  External system device (sensor or effector)



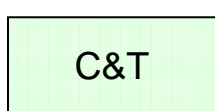
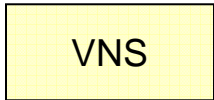
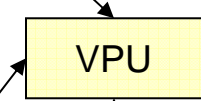
Example: Diagram 1A Navigation – Read Sensors



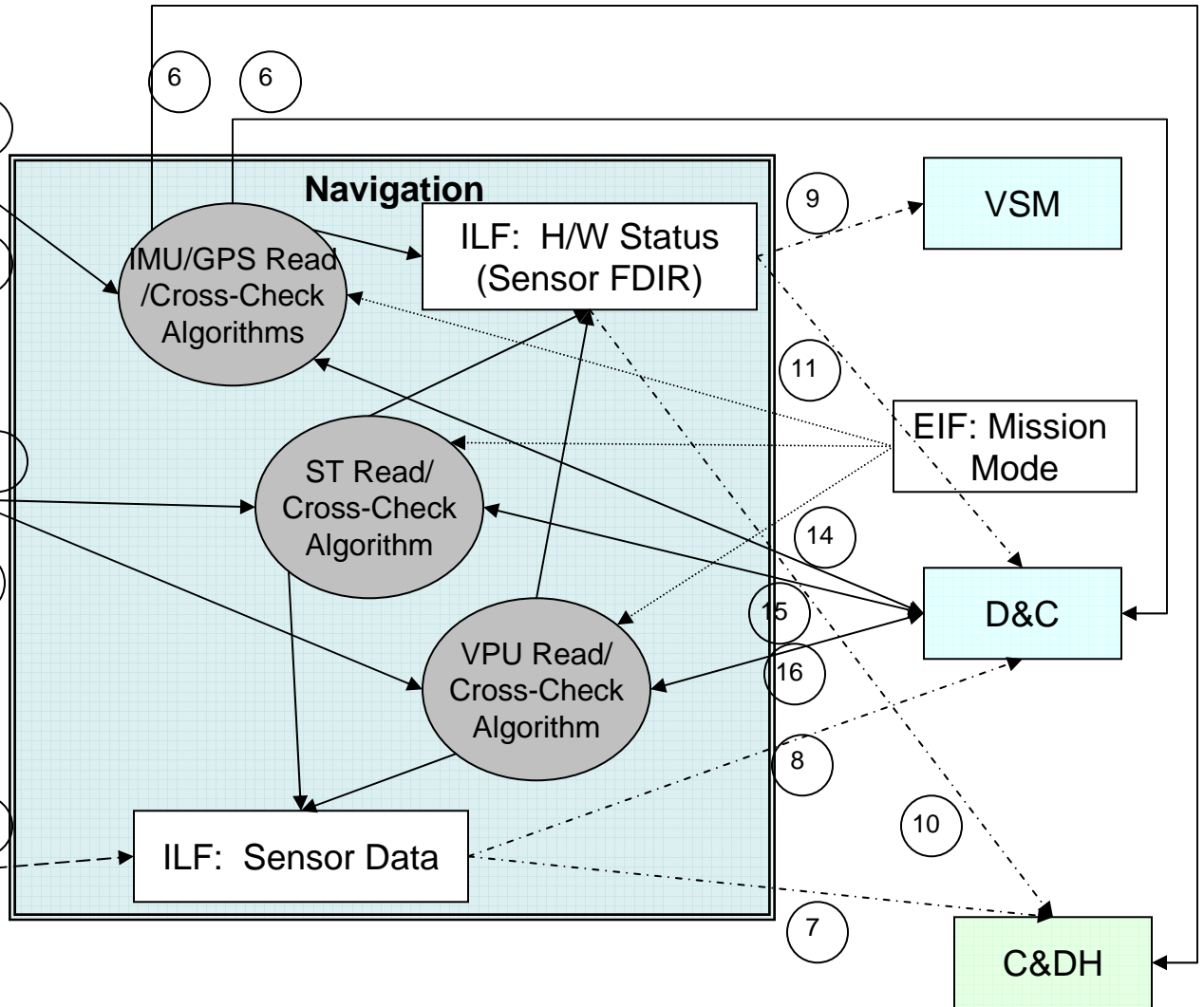
Absolute Nav Sensors

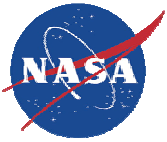


Relative Nav Sensors



Note: To clarify, labels on flows beyond VPU indicate logical transactions crossing the boundary between Navigation software and external devices





Example: Function Point Component List Diagram 1A Navigation – Read Sensors

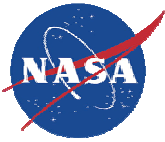


DIAGRAM 1A		NAVIGATION - Read Sensors							
TRANSACTIONAL COMPONENTS				Some functions run continuously, others can be selectively activated on demand. Activations can vary by mission phase and modes.					
Flow nbr	Flow from/to	Req Doc ref	FP type	Component name	Unique Data Flow Conditions	FP Val	Ttl.	Cond Indic.	comment
1	IMU/SIGI / IMU/GPS Read/Cross-Check Algo	SS-SC-2393; SS-SC-958	EI	IMU input:trans/rot.accel	1	4	4		when SIGI is gone, then interact directly with MIMU & GPS
1	SR/C-C Algo-SIGI/MIMU command	GNC.0468; GNC.0098;	EO	MIMU/SIGI commands	3	5	15		initialize, correction, test
2	GPS/SIGI / IMU/GPS Read/Cross-Check Algo	SS-SC-2393; SS-SC-958	EI	GPS input:position/time	1	4	4		
2	SR/C-C Algo-SIGI/GPS commands	GNC.0468; GNC.0098; GNC.0210	EO	GPS/SIGI commands	3	5	15		location, attitude,test
3	VPU/StarTracker / ST Read/Cross-Check Algo	SS-SC-2393; SS-SC-958	EI	StarTracker input:veh.attitude (update IMU)	1	4	4		
3	SR/C-C Algo-VPU/Star Tracker commands	GNC.0468; GNC.0098; GNC.0210	EO	Star Tracker commands	4	5	20		on,off, large field of view,small f.o.v.
4	VPU/Camera / VPU Read/Cross-Check Algo	SS-SC-449; SS-SC-958	EI	VPU input:range,rate,rel.att.	1	4	4		
4	SR/C-C Algo-VPU camera commands	GNC.0468; GNC.0098; GNC.0210	EO	camera commands	4	5	20		on,off,zoom in, zoom out.
6	IMU/GPS Read/Cross-Check Algo / C&DH & D&C	SS-SC-1884	EO	algo.output:Alert-loss OR restore of CEV absol.orbit determination due to GPS	2	5	10		loss-GPS,restore-GPS,loss-absol.nav.capabil.,restore-abs.nav.cap.
9	SR/C-C Algo-FDIR alert		EO	FDIR alerts	2	5	10		bad h/w,bad data
12	VPU/VNS / VPU/Read/Cross-Check Algo		EI	VPU/VNS commands	1	4	4		VPU input: range,rate,rel.att.
12	SR/C-C Algo-VPU VNS commands		EO	VNS commands	3	5	15		on,off,test
13	SR/C-C Algo-VPU processor commands		EO	VPU commands	4	5	20		on,off,test,scale on/off,
14	D&C / IMU/GPS Read/Cross-Check Algo		EI	Manual activation command (nbr TBD)	1	4	4		manual algorithm activation
15	D&C / ST Read/Cross-Check Algo		EI	Manual activation command (nbr TBD)	1	4	4		manual algorithm activation
16	D&C / VPU Read/Cross-Check Algo		EI	Manual activation command (nbr TBD)	1	4	4		manual algorithm activation
17	VPU/StarTracker / SR/C Read/Cross-Check Algo		EI	StarTracker input:veh.attitude (update IMU)	1	4	4		
Functional Total							161		

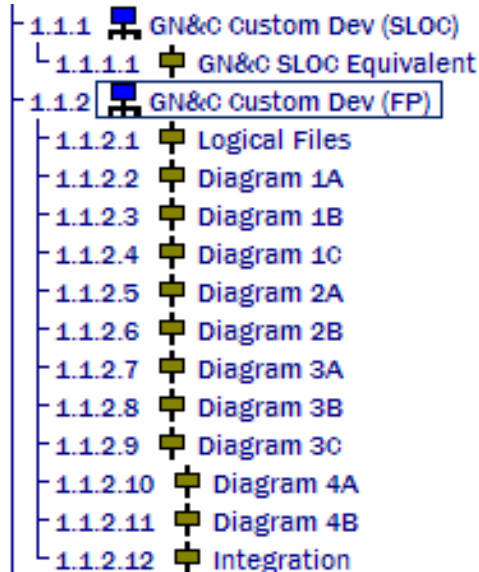


Agenda

- Task Background
- Function Points Concepts
- GN&C Function Point Analysis
- GN&C SEER-SEM Estimate

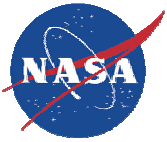


SEER-SEM GN&C Estimate Structure and Parameters



Note: Algorithms are not separately addressed by FPA. In these estimates we used a placeholder value of 2 KLOC of reused code for each of 14 major identified algorithms. SEER parameters (for real time code and other factors) account for the greater difficulty of development of this type of application and consequently result in greater effort and cost than would be the case for more ordinary software of equivalent size.

Parameters - Program: GN&C FP Custom Dev (FP)			
- PERSONNEL CAPABILITIES & EXPERIENCE			
Analyst Capabilities	Low	Nom	Hi
Analyst's Application Experience	Nom+	Hi	VHi-
Programmer Capabilities	Low	Nom	Hi
Programmer's Language Experience	VHi+	EHi	EHi
Development System Experience	Nom	Hi-	Hi+
Target System Experience	Nom	Nom	Nom
Practices & Methods Experience	Nom	Hi	VHi
- DEVELOPMENT SUPPORT ENVIRONMENT			
Modern Development Practices Use	Nom+	Hi	VHi-
Automated Tools Use	Hi-	Hi+	VHi-
Turnaround Time	VLo	VLo	Low-
Response Time	Nom+	Hi	VHi-
Multiple Site Development	Nom	VHi-	VHi+
Resource Dedication	Low-	Low-	Low+
Resource and Support Location	Hi-	Hi	VHi-
Development System Volatility	Nom	Nom	Nom
Process Volatility	Low	Low+	Nom
- PRODUCT DEVELOPMENT REQUIREMENTS			
Requirements Volatility (Change)	Hi-	Hi+	VHi
Specification Level - Reliability	Hi+	VHi	VHi
Test Level	Hi+	VHi	VHi
Quality Assurance Level	Hi+	VHi	VHi
Rehost from Development to Target	Nom	Nom+	Hi-
- PRODUCT REUSABILITY REQUIREMENTS			
Reusability Level Required	Nom	Nom	Nom
Software Impacted by Reuse	100.00%	100.00%	100.00%
- DEVELOPMENT ENVIRONMENT COMPLEXITY			
Language Type (complexity)	Nom	Nom	VHi
Development System Complexity	Nom+	Hi	Hi
Application Class Complexity	Nom	Nom+	Hi-
Process Improvement	Nom	Nom	Hi-
- TARGET ENVIRONMENT			
Special Display Requirements	Hi+	VHi	VHi+
Memory Constraints	Nom+	Hi	VHi-
Time Constraints	Hi+	VHi	EHi-
Real Time Code	VHi-	VHi	VHi+
Target System Complexity	Nom	Nom	Hi
Target System Volatility	Low	Low	Nom
Security Requirements	Nom	Nom	Nom



GN&C Estimate: Illustrative Outcome

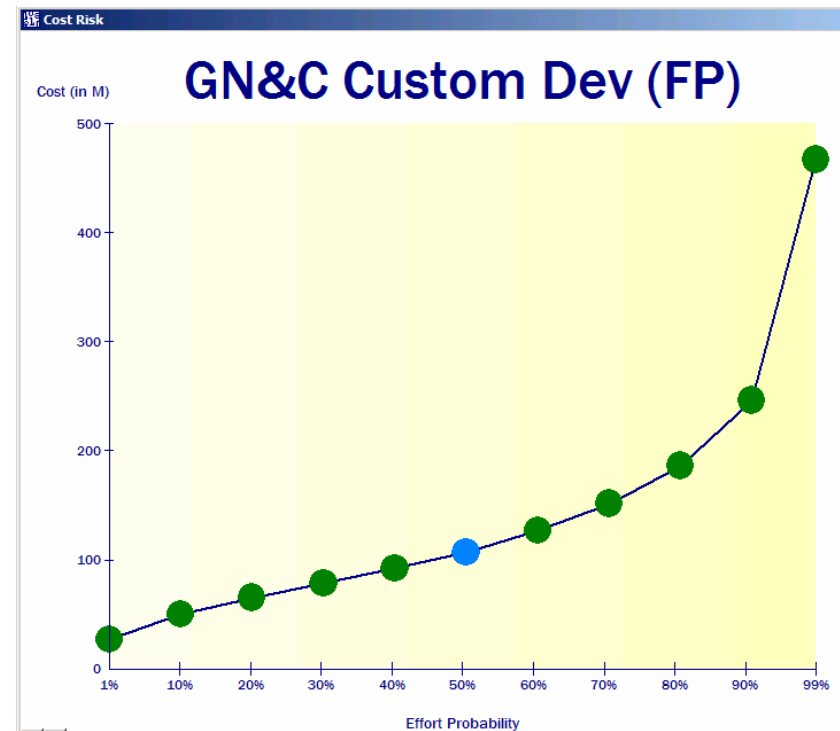


Scenario – Total custom software development

Point Estimate (50% CF):

Cost Risk Outcome:

Quick Estimate	
Program: GN&C Custom Dev (FP)	
	Estimate
Development Schedule Months	75.94
Development Effort Months	4,358.86
Development Effort Hours	662,547
Development Base Year Cost	106,892,400
Defect Prediction	68
Constraints	MIN TIME





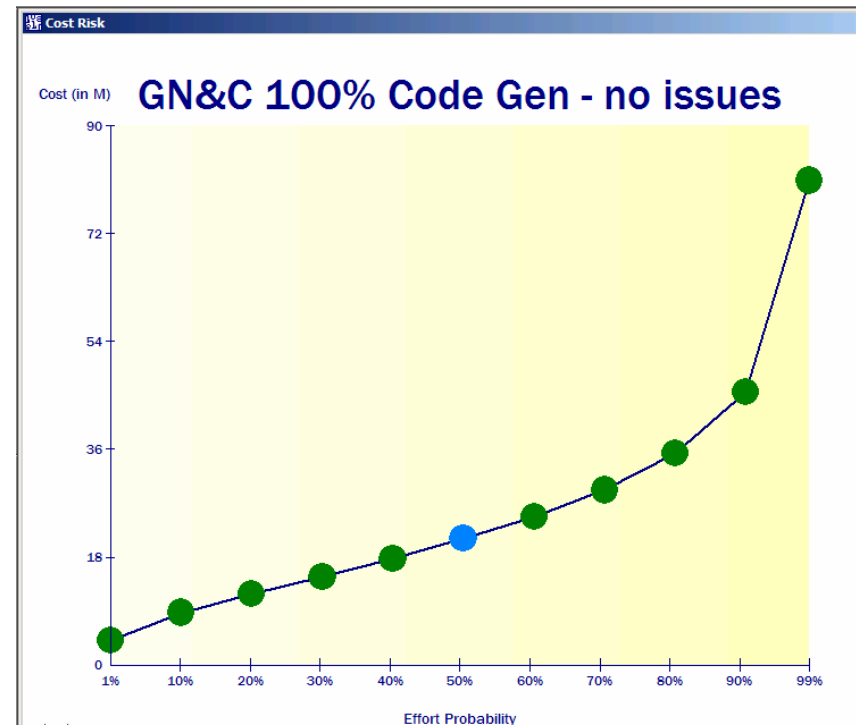
GN&C Estimate: Illustrative Outcome

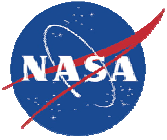
Scenario – Total code generation (maximally optimistic outcome)

Cost Risk Outcome:

Point Estimate (50% CF):

Quick Estimate	
Program: GN&C 100% Code Gen - no issues	
	Estimate
Development Schedule Months	44.20
Development Effort Months	859.49
Development Effort Hours	130,642
Development Base Year Cost	21,077,246
Defect Prediction	50
Constraints	MIN TIME

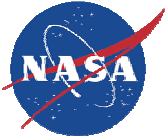




GN&C Estimate: Function Point to SLOC Equivalence



- There is no reliable translation table or method of crosswalk between SLOC and function points
- Software estimates here are sized in function points and then combined with additional parameters specific to the application type, environment, etc. to produce a cost estimate
- To determine an equivalent representation of cost based on SLOC we reverse engineered the function point solution to determine the equivalent size in SLOC that could be created with the same cost and effort.
- According to SEER in the scenario of a solution based on custom code the development of 112.5K to 150K SLOC would require effort equivalent to the development of the current estimate of 939 function points for GN&C (including expected incorporation of reused code for algorithms, currently a placeholder value)



Forward work and an Overview of the rest of the FSW



- Booz Allen did not have sufficient funding to continue developing a FP based size estimate for the CSCI beyond the work done for the GNC illustration
- We continue to work on developing a SEER- SEM estimate for the total FSW architecture using SLOC counts provided by the Prime and its teammates
- The FSW consists of flight software which includes GNC, Electronic Ground Support Equipment (EGSE) (ground software), Data Service, Firmware and Test and Simulation software
- NASA held the Software Requirements Review in March and we are incorporating that data in the estimate with the assistance of Gary Constantine, Galorath senior consultant