NORTHROP GRUMMAN

DEFINING THE FUTURE

Software Risk Simulation

SCEA National Conference 2008
June 2008

Jeff Jaekle
Eric Druker
Emily Jessen
Jackie McQueston
Brian Brophy

Northrop Grumman Corporation

Outline

- **Purpose**
- **Introduction**
- **Methodology**
- **Software Risk Factors**
  - Code Growth
  - Productivity Rates
  - Schedule Achievability
- **Monte Carlo Simulation**
- **Next Steps**
- **Applications**
- **Back Up**

Purpose

NORTHROP GRUMMAN

- Software development efforts inherently contain a certain level of risk
- When creating a cost estimate for a software development effort the way that risk is applied becomes very critical to the final cost estimate of that effort
- Additionally, customers have asked for distributions around cost estimates and more statistically based risk analysis surrounding software estimation
- This paper discusses the implementation of a software risk simulation based on historical data that derives risk factors for code growth, productivity rates, and schedule compression

Software is like entropy. It is difficult to grasp, weighs nothing and obeys the Second Laws of Thermodynamics, i.e., it always increases. [1]

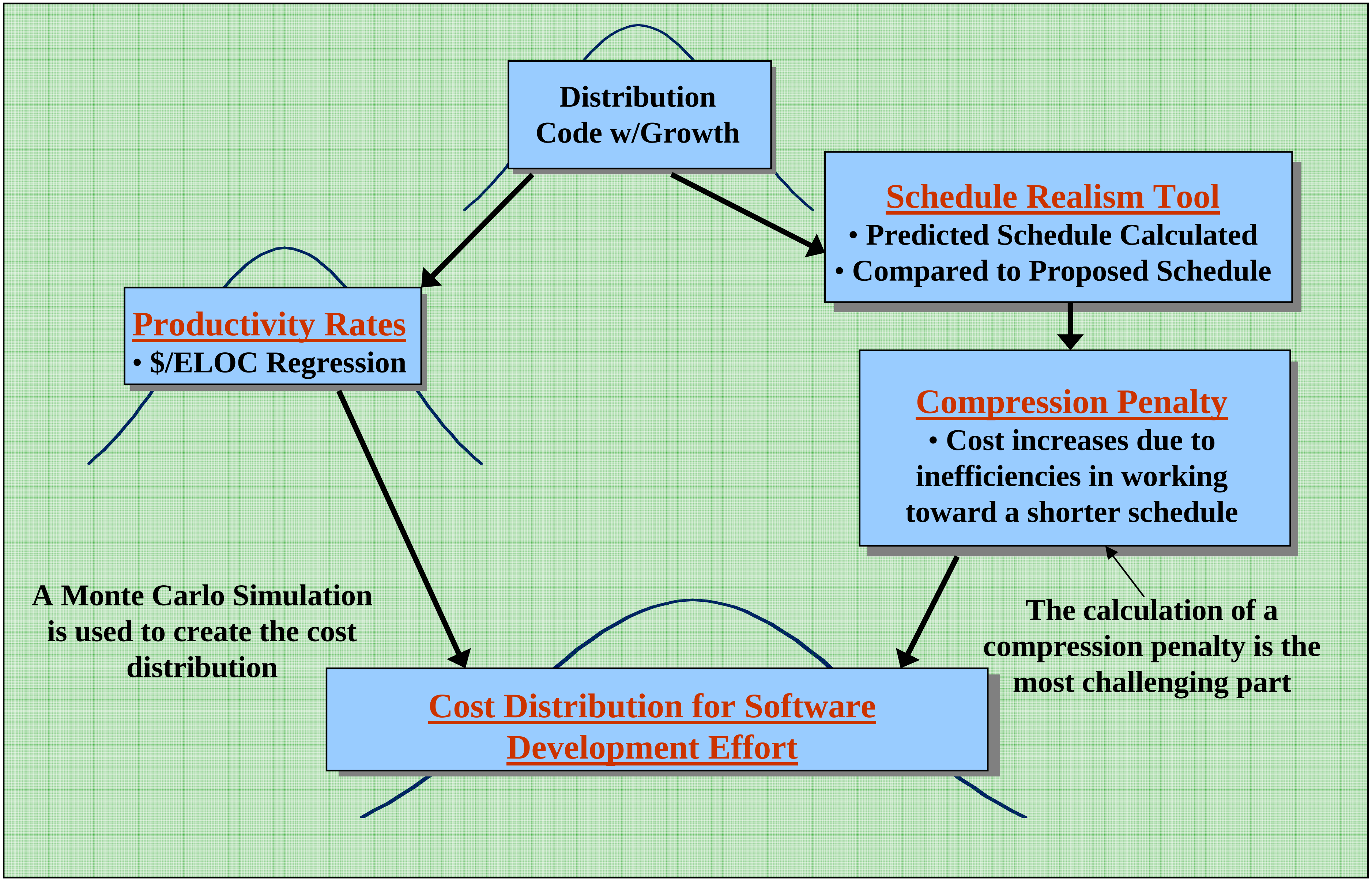
-- NORMAN R. AUGUSTINE

1. *Estimating Software Size, Cost, and Schedule: Mission Success Through Life Cycle Processes*. Jones, James E. SCEA, June 1999.

Introduction

- Software development costs and schedules are notoriously difficult to predict
  - This is due, in part, to the large number of risk factors associated with software development:
    - Code Growth – The final count for Software Lines of Code (SLOC), is on average larger than the initial estimate provided by the Contractor
    - Productivity rates – The cost per SLOC is a measure of how productive a Contractor is when developing a release
    - Schedule Achievability – The schedule estimate provided by the Contractor is generally shorter than the final schedule, and this risk needs to be taken into account when calculating a total software development cost
- The goal is to create a risk tool for software development that encompasses all three of these risks by using the statistical distributions around the historical data
  - With a final cost distribution the customer can gauge how much risk they should budget for
  - In addition, risks can be assessed across a portfolio of programs within an agency or department to enhance the decision making process and aid in portfolio management

Methodology



Code Growth

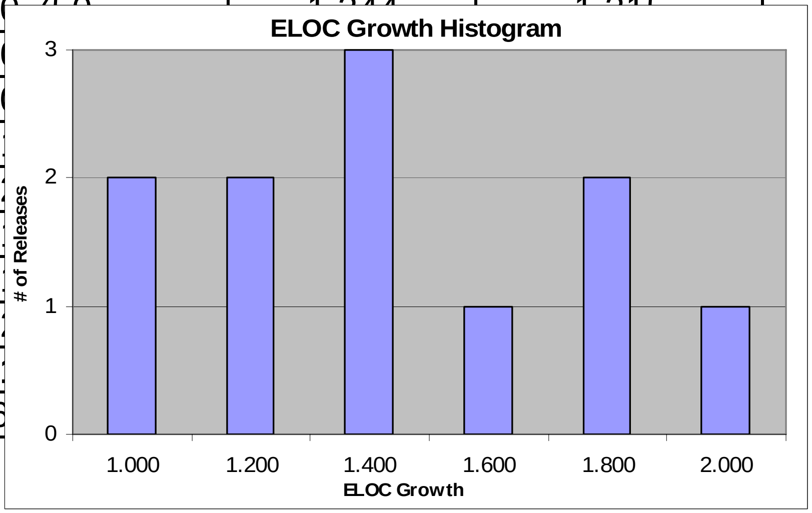
- This paper uses Equivalent Lines of Code (ELOC) as the standard measure for the size of a software effort
- Historically software efforts always have some level of code growth due to a number of different factors (e.g., under estimation of original code, additional requirements, inefficiency in coding, etc.)
- To assess expected code growth for the data set of Automated Information System AIS programs, initial SLOC estimates were compared to final SLOC counts
- Using this methodology a few difficult situations were encountered with the data
  - Determining which SLOC estimate was the absolute initial proposed estimate
  - Changes in requirements or merged releases made it difficult to compare code counts

Code Growth (cont'd)

NORTHROP GRUMMAN

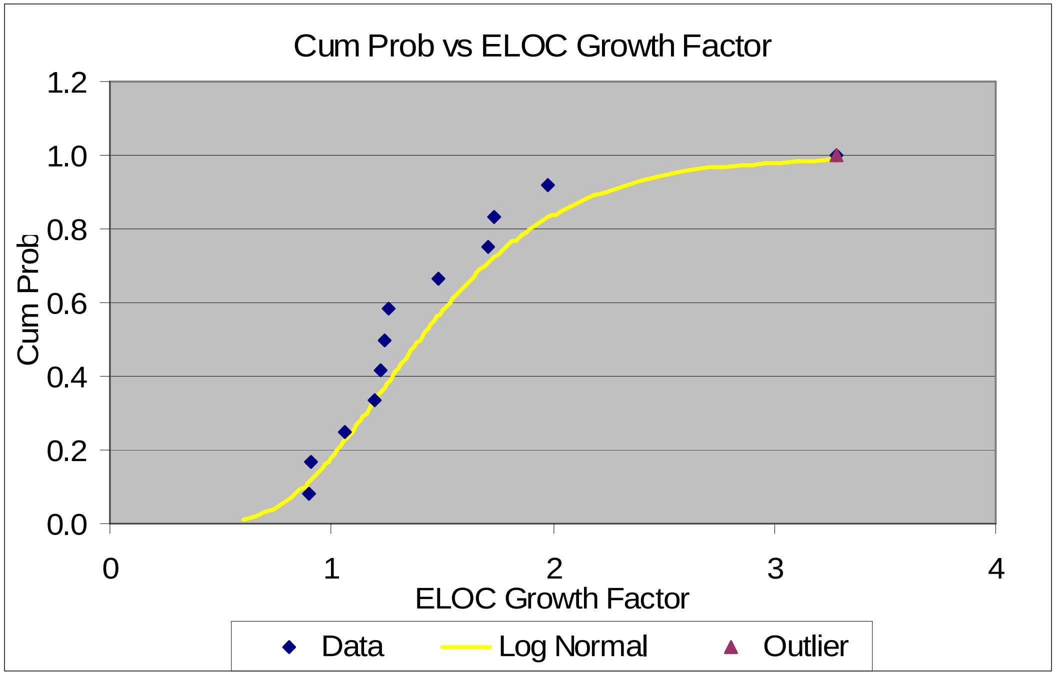
- The following table shows a subset of the average SLOC growth for one AIS program
- By using the distribution around the code growth data set, a range of possible code growth factors can be obtained
- The code growth factor is applied to the initial LOC estimate to predict the final code count (including growth) of the software release
  - This code will then be used to estimate the final cost of the release

Release	Initial ELOC	Final ELOC	ELOC Growth	New Code Growth	IR Growth	Reuse Growth
1	94,899	117,669	1.240	1.203	1.130	1.130
2	11,441	16,926	1.479	0.610	1.884	1.884
3	61,461	77,145	1.255	1.158	1.390	1.462
4	41,019	37,289	0.909	0.750	1.244	1.215
5	65,804	113,894	1.731	[illegible]	[illegible]	[illegible]
6	76,962	69,177	0.899	[illegible]	[illegible]	[illegible]
7	299,524	357,739	1.194	[illegible]	[illegible]	[illegible]
8	26,132	51,773	1.981	[illegible]	[illegible]	[illegible]
9	28,713	30,556	1.064	[illegible]	[illegible]	[illegible]
10	15,000	18,293	1.220	[illegible]	[illegible]	[illegible]
11	96,095	163,776	1.704	[illegible]	[illegible]	[illegible]
Average			**1.334**	[illegible]	[illegible]	[illegible]
St Dev			0.349992625	0.66[illegible]	[illegible]	[illegible]



Code Growth Factor Distribution

- The ELOC Growth Factor follows a lognormal distribution and passes the K-S test

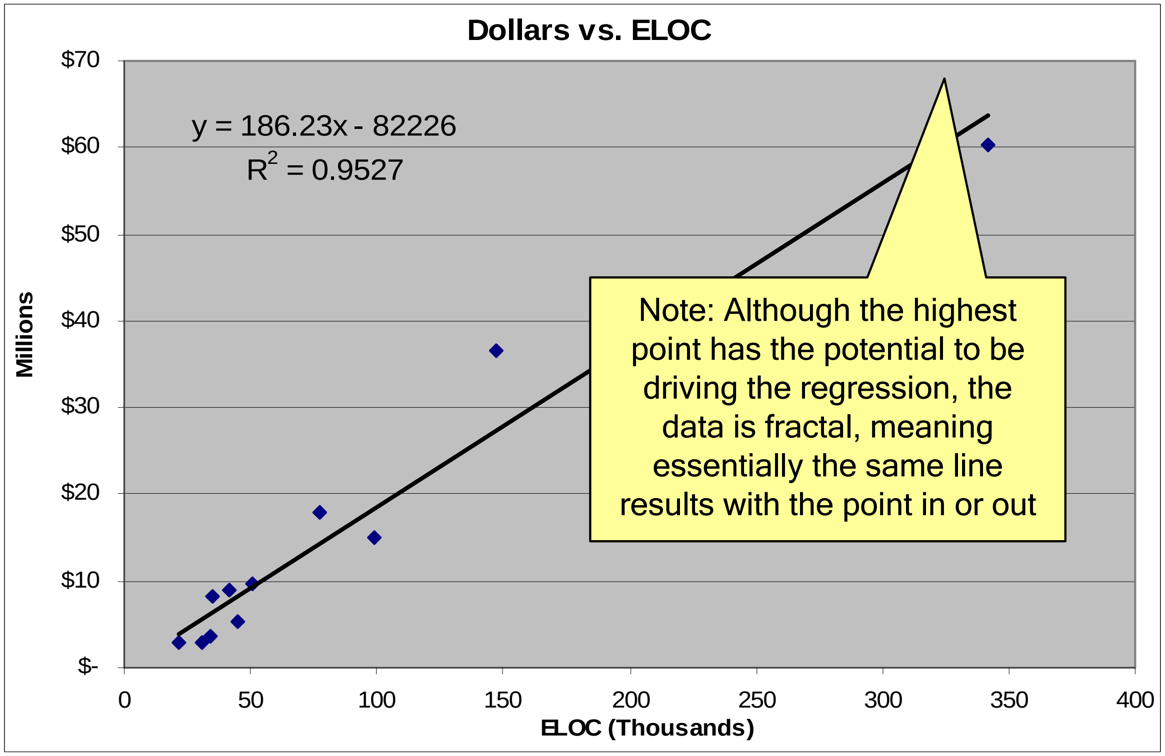


	Lognormal Distribution	Data
Mean Growth Factor	1.486	1.334
Median	1.398	1.24
Standard Deviation	0.532	0.300
CV	35.8%	22.5%

Note: The data set includes the removal of the outlier

Productivity Analysis

- The cost account data for the same program was used to calculate the productivity rates ($/ELOC)
  - These rates are calculated based on the fully burdened cost of a historical release over the total ELOC ($/ELOC)
- The regression equation obtained from the data set for a particular program is used to estimate a final cost
- By using the distribution around the regression line, a range of possible productivity rates can be obtained to support a final cost estimate, thus creating a cost distribution

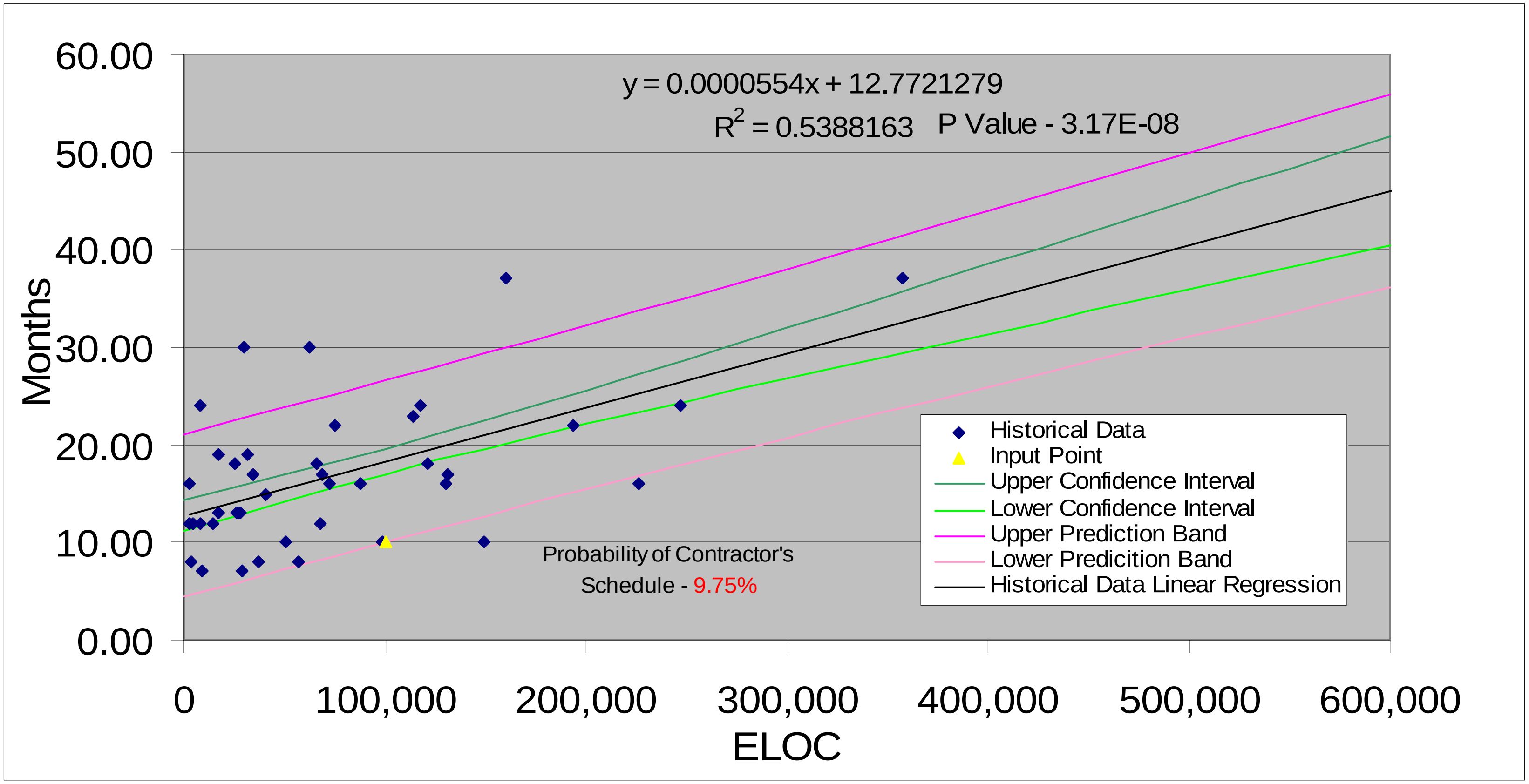


Schedule Achievability

- A schedule achievability tool was created that provides:
  - A methodology for producing schedule distributions based on the historical data set of AIS programs
  - A schedule prediction band for a proposed software development program based on ELOC
- Having this prediction band allows the user to determine:
  - A suggested schedule length
  - The probability that a proposed software development schedule will be met
  - Upside/Most Likely/Downside scenarios for the final schedule
  - A new schedule prediction as ELOC changes
- Based on historical data from 39 complete AIS releases
  - Completed releases include the 11 program specific releases used in the previous analysis
  - Final Schedule (in months) was scatter-plotted against ELOC at Complete
- Used Prediction Intervals to calculate the probability of the Contractor's Proposed Schedule being equal to the Actual Schedule

1. *Schedule Realism Prediction Band Tool.* Converse, Allison, Jaekle, Jeffrey, and Druker, Eric; SCEA, June 2007.

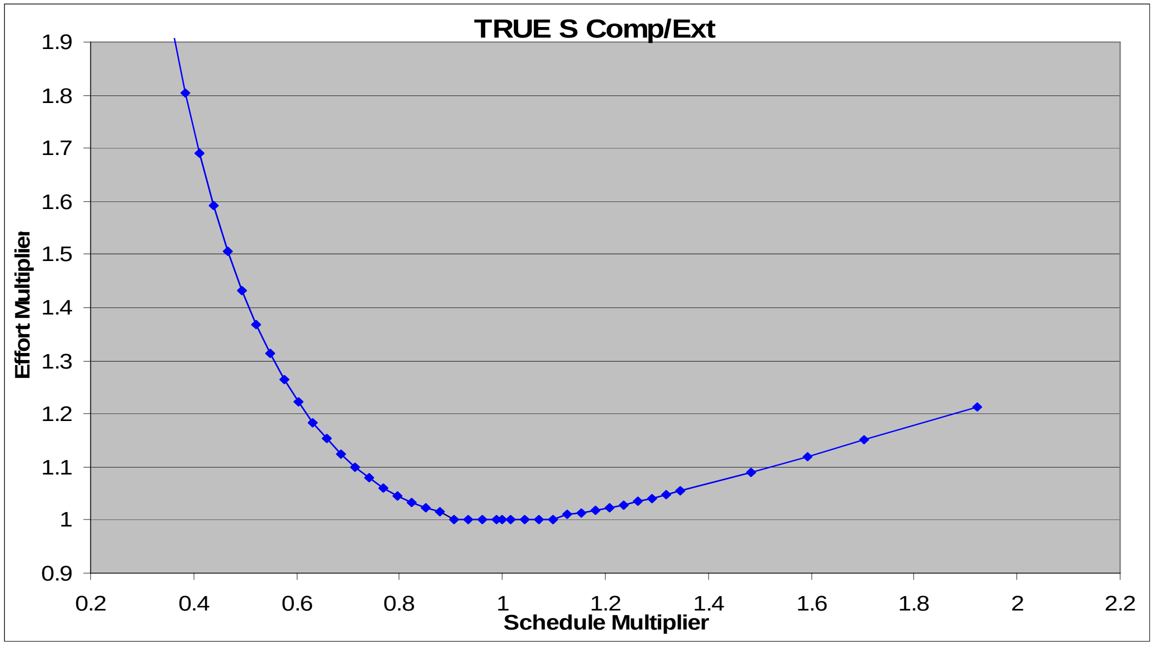
Schedule Achievability (cont'd)



- In order to relate schedule to cost, a penalty may be applied to a software release if the Contractor's proposed schedule is shorter than what the historical data predicts
- By comparing the Contractors schedule to the schedule predicted by the tool a compression penalty may be applied to the final cost

Compression Calculation

- Schedule compression is applied to the cost based on how much shorter the proposed schedule is compared to the predicted schedule: the compression penalty is based on a derived curve from TRUE S
- The cost will increase due to the reduced time the software engineers have to code, resulting in an increase in the number of errors as well as an increase in the amount of testing that will need to be performed. Overall, the entire process becomes less efficient and more costly.
- Example: If the predicted schedule from the schedule realism tool is 12 months and the Contractor proposed schedule is 6 months there would be a 0.5 schedule multiplier which would equate to a 1.4 effort multiplier

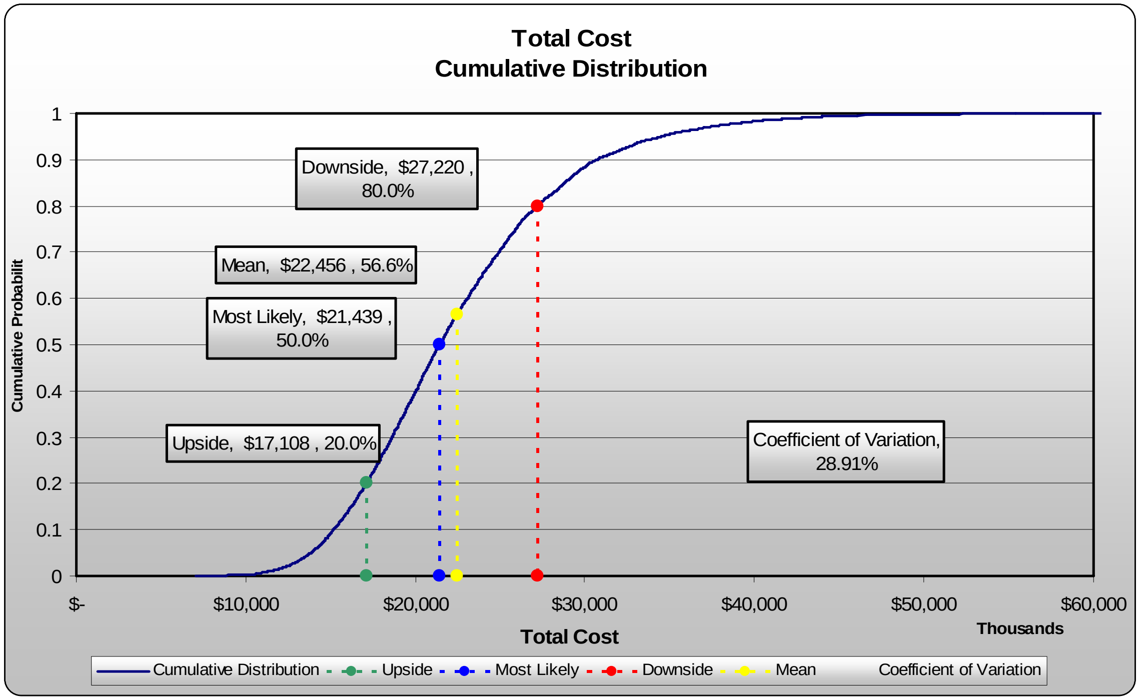


Presented at the 2008 SCEA-ISPA Joint Annual Conference and Training Workshop - www.iceaaonline.com

Monte Carlo Simulation

NORTHROP GRUMMAN

- A Monte Carlo Simulation is a tool used to imitate the events that could occur over the life of a program
  - Potential "real life" events are assigned a probability distribution
  - Using a random number generator and the inverse Cumulative Distribution Function (CDF) technique, outcomes are assigned to these events
- The software risk simulation will use the distributions around the code growth, productivity factors, and schedule prediction in addition to the potential compression penalty to obtain a final cost distribution for any future release



Next Steps

NORTHROP GRUMMAN

- Code Growth
  - ELOC growth factor
    - Look for ELOC growth distributions for different programs or families
    - Investigate size effects for the ELOC growth factor
  - New, Modified, and Reuse code growth factors
    - Look for growth distributions based on New, Modified, and Reuse code
    - Look for relationships between families of code growth
- Productivity Rates
  - Break cost verses ELOC CER using families (New, Modified, Reuse) of code rather than only ELOC
- Schedule Realism/Compression Penalties
  - Look into adjusting the schedule realism model for different programs
  - In the future, initial release schedules will be compared to final schedules to calculate a statistically based compression/extension factor
    - The data is still being collected to support that effort

Application

- The Monte Carlo simulation used to create the S-curve for the cost distribution will provide decision makers a range of potential costs and an assessment of how much risk they should budget for
- A refined portfolio management process can be used to examine the cost distributions across multiple programs to assess which programs have the greatest cost, productivity and schedule risk thereby allowing a decision-makers to determine the programs probability of success.
- A risk simulation can be expanded to capture the risks associated with other data sets, such as data ingest, export and import rates, requirements definition, etc.

Back Up

Example Inputs

NORTHROP GRUMMAN

Release 1	
New Code	100000
Modified	
Internal Reuse	
External Reuse	
Growth	0.525
ELOC	100000
ELOC with grow	152499
Contractor Schedule	20
Predicted Schedule	20.139824
Schedule Deviation	0.993057
Compression/Extension Penalty	1.00
Compression & Growth	1.52
Random Number	0.737350198
Historical Code Growth	1.524990696
Random Number	0.472272555
Schedule	20.13982422
Random Number	0.18844126
Cost	$ 24,209,878
Final Cost	**$ 24,209,877.56**

Run Simulation

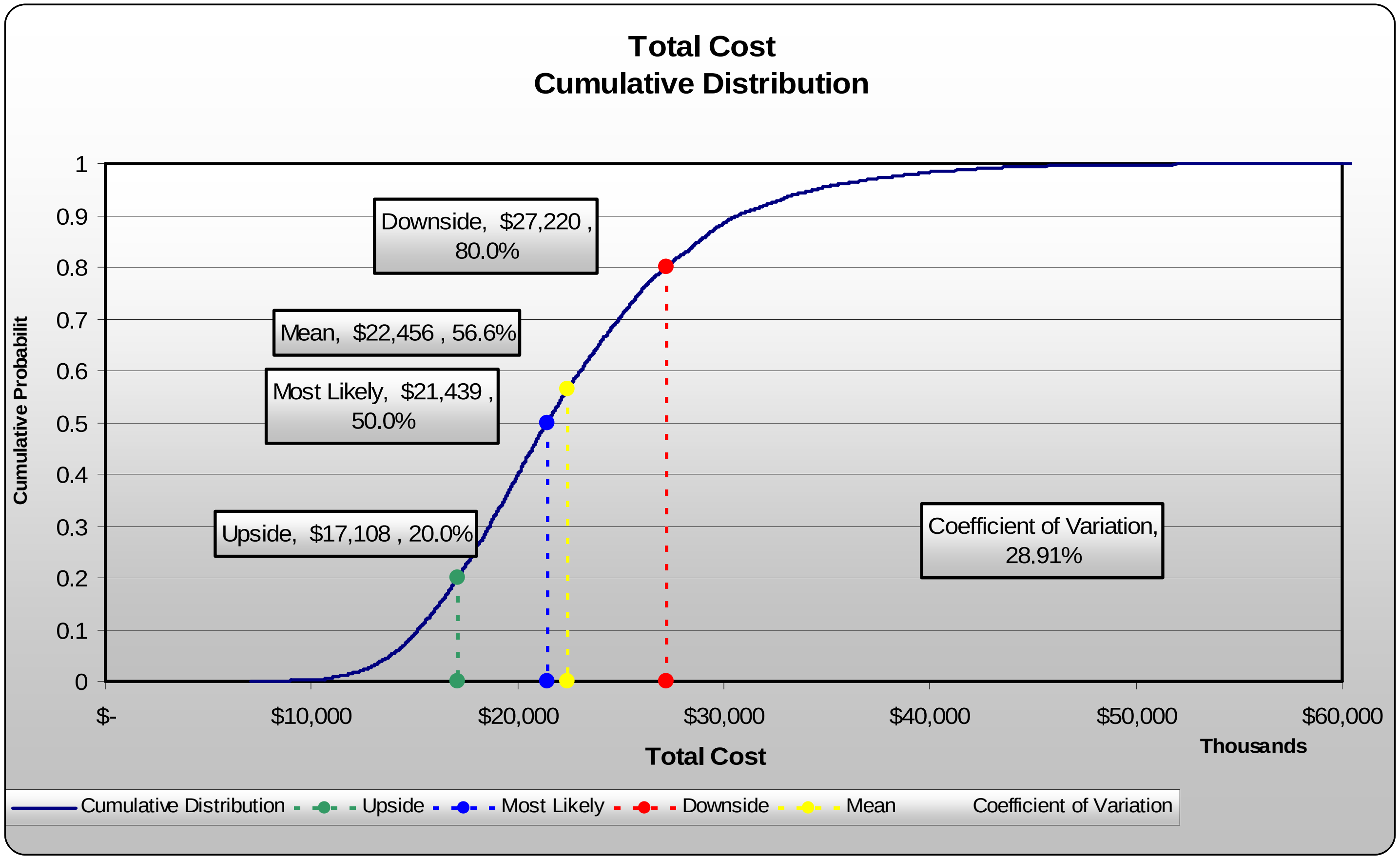
The inputs needed to run the simulation include the predicted code for new, modified, internal reuse, and external reuse

The proposed contractor schedule is also needed as an input

Each factor (code growth, schedule, and productivity rate) uses a different random number to predict the final value

Presented at the 2008 SCEA-ISPA Joint Annual Conference and Training Workshop - www.iceaaonline.com

Example Outputs



	Upside	Most Likely	Downside	Mean	CV
	20.0%	50.0%	80.0%	56.6%	
Final Cost	$ 17,108,267	$ 21,439,470	$ 27,219,751	$ 22,456,381	28.91%